Activation Functions as Parameters for Improved Deep Learning

by

James M. Locke

A dissertation proposal submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 1, 2021

Keywords: activation function, deep learning, artificial intelligence, machine learning, bias

Approved by

David Paradice, Chair, Harbert Eminent Scholar in Information Systems & Analytics
R. Kelly Rainer, George Phillips Privett Professor of Management Information Systems
Casey Cegielski, Professor of Management Information Systems
Pei Xu, Professor of Management Information Systems

Abstract

Neural networks have evolved into strong and dependable machine learning systems. Practitioners and researchers using them have at their disposal many tuning levers for achieving successful learning. A key task is selecting optimal activation functions (AF) for the hidden and output layers. While the recipe for selecting the output layer activation function is determined by the type of data, hidden layer AFs largely determine learning success. Practitioners, being human, approach problems with an inference bias that is driven by goals, problem interpretation, and previous experiences. Activation functions are hyperparameters; they cannot be adjusted during learning. This leads to repeated trials as users search for the optimal combination of activation functions to assign among the hidden layers. Swapping activation functions during training offers solutions to both the user bias problem and the risk of suboptimal learning. The key idea is for the loss function to sense learning decline and, at an appropriate point, discard the underperforming activation function in favor of another one. This process is repeated every time the low learning threshold is reached and continues until convergence. In this study I show that designing a neural network with the ability to manage activation functions as parameters (they may be changed by the system in response to a training success threshold) improves performance and efficiency while mitigating user bias.

Acknowledgments

There is no one I can thank more than my dear wife, Diane, for her unwavering support, encouragement, and devotion. Her character and steadiness made it possible to pursue and enjoy this long-awaited personal calling.

My sons, John and Michael, have inspired me to seek new challenges, to never settle, and to think about everything from new perspectives. I am grateful for their examples of youthfulness—even as they grow into fine men.

I hope to achieve my father's easy self-discipline, depth of character, and love for teaching while exemplifying my mother's loving and personal touch. They are missed. I hope my life would have made them proud.

I am so thankful for my dear relative, NLM, for switching her old black-and-white television from Saturday morning cartoons to *Leonard Bernstein's Young People's Concerts* on the local "educational TV" channel. Years later, I still tear-up at Beethoven's 9th. I thank her for that slip of paper she handed me not long before her passing that read, "Always show less than you have, always speak less than you know."

The four scholars' names that appear on the cover page have titles. They are each and together far more important to me that those titles. Their dedication to my education is deeply and profoundly appreciated. Some things are so valuable they cannot be repaid.

Thank you, all.

Table of Contents

Abstract 2

Acknowledgments 3

List of Tables 6

List of Figures 7

List of Abbreviations 9

Chapter 1 Introduction 10

Chapter 2 Literature and previous research 14

2.1 Research in Management Information Systems 14

2.2 Biases 21

2.3 NN Design and Implementation 24

Chapter 3 Method 32

Chapter 4 Results 46

4.1 Initial Test Results 48

4.2 Leaky ReLU, MNIST Data Set, Kolmogorov-Smirnov Tests 50

4.3 ReLU, MNIST Data Set, Kolmogorov-Smirnov Tests 52

4.4 Sigmoid, MNIST Data Set, Kolmogorov-Smirnov Tests 56

4.5 Softmax, MNIST Data Set, Kolmogorov-Smirnov Tests 60

4.6 TanH, MNIST Data Set, Kolmogorov-Smirnov Tests 64

4.7 MNIST Data Set, 30,000 epochs 68

4.8 Iris Data Set, 300 epochs 70

4.9 Summary 72

Chapter 5 Discussion 75

5.1 Overview 76

5.2 Contribution 77

5.3 Limitations 78

5.4 Future Research 78

References 80

Appendix 1 Typical Activation Functions 86

Appendix 2 Code for a neural network binary classifier to test various AF combinations 87

Appendix 3 R code of Two-sample Kolmogorov-Smirnov test 98

Appendix 4 Training results 105

List of Tables

Table 3.1 Definitions 33

Table 3.2 Neural network architecture and hyperparameter settings 43

Table 3.3 Neural network architecture and hyperparameter settings for Iris data set 44

Table 3.4 Platform configuration for MNIST studies.. 45

Table 4.1 Performance summary 74

List of Figures

Figure 2.1 Wolves vs Huskies 19

Figure 2.2 Dynamic Neuron Model 27

Figure 2.3 Majetic, et. al., Glass-Mackey time series 28

Figure 2.4 Majetic, et. al., dynamic NN results 29

Figure 2.5 CSN base and meta learner model 30

Figure 3.1 Preliminary AF combination results of coded NN; MNIST Data Set 35

Figure 3.2 Fraud detection data accuracy with various AFs 36

Figure 3.3 Activation functions used in the study 38

Figure 3.4 Entropy in the case of two possibilities with probabilities p and $(1-p)$ 40

Figure 3.5 Shannon Entropy Equation 41

Figure 3.6 Shannon Entropy Function as activation function choice algorithm 41

Figure 3.7 Example of hand-written digits of MNIST data set 42

Figure 3.8 Example of the Iris data set 44

Figure 4.1 Plots from initial test results 49

Figure 4.2 Leaky ReLU over 300 epochs with K-S Tests 52

Figure 4.3 ReLU over 300 epochs with K-S Tests 56

Figure 4.4 Sigmoid over 300 epochs with K-S Tests 60

Figure 4.5 Softmax over 300 epochs with K-S Tests 64

Figure 4.6 TanH over 300 epochs with K-S Tests 68

Figure 4.7 TanH, 30,000 epochs 70

Figure 4.8 Iris data set, 300 epochs 72

List of Abbreviations

AF	Activation Function
AI	Artificial Intelligence
CSN	Conditionally Shifted Neurons
DBMS	Data Base Management System
DIKW	Data, Information, Knowledge, Wisdom
DSS	Decision Support System
ELU	Exponential Linear Unit
GPU	Graphics Processing Unit
LReLU	Leaky Rectified Linear Unit
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology
NN	Neural Network
PReLU	Parametric Rectified Linear Unit
ReLU	Rectified Linear Unit
TanH	Hyperbolic Tangent Function

Chapter 1

Introduction

Neural networks (NN) are a widely used artificial intelligence (AI) / machine learning (ML) deep learning technique. They are often referred to as "black boxes" due to the fuzzy understanding of the assumed complexities attributed to the computations and algorithms that are executed between the input and output stages of use. Integral to the black box perception are the levers of control (hyperparameters) available to users for processing the data used in training a neural network [2]. To optimize research value, then, we must question how hyperparameter selection effects shape machine learning and how accepted practices, or ignorance thereof [2], impact the overall use of the technology for any specific learning event. In short, any bias introduced through the selection of a specific hyperparameter value is unknown and thus has unknown consequences for the subsequent NN use, an artifact that demands rigorous investigation.

A second issue involves the dynamic nature of business environments. Training versus prediction issues arise and can easily go unnoticed from such a high level of algorithm training as users confidently assess and improve their algorithm's performance based on *status quo* accuracy measures of the training data. While that approach may be reasonable for data that will not fundamentally change over time, such as the traditional NN examples of classifying cats or the hand-written numbers of the MNIST data set, it is unreasonable for most business use cases. Predicting recidivism or home sales, both subject to unforeseen events that may not have been captured in a training set, will likely be an unreliable use of a NN. While characteristics of data sets, such as means and standard deviations, often remain constant in both training and application environments in low-noise or low-context NN applications, such cannot be assumed in many highly contextual business applications.

The inherent problem in predicting recidivism versus classifying hand-written numbers is one of context. Predicting recidivism requires context. Repeatedly training algorithms on hand-written digits results in only a slight wavering of means and standard deviations among thousands of similar digits. The algorithm eventually behaves like a regression exhibiting a high power: It has a strong ability to detect a true effect where it exists. The low context environment contributes to the learning power. In a high context environment, on the other hand, such as that found when training a NN to predict recidivism, results exhibit significantly larger standard errors. NNs self-adjust parameters to better train across a broad range of features. This standard design feature is an effort to gain context but the activation function found in every node central to the learning paths cannot self-adjust; i.e., they are hyperparameters.

The use of NNs in business reflects this common scenario. Though experiencing rapid and widespread growth, presently, only about 8% of firms engage in core practices that support widespread adoption of AI or advanced analytics [21]. While the overall growth of this information processing domain is exponential, this conservative approach to mission-critical systems deployment of AI signals an uneasiness about the efficacy and reliability of AI technology.

As deep learning neural networks become more pervasive, generally, and are being employed to inform users and decision makers in more critical business and societal roles, specifically, IS researchers are uniquely situated to examine how such AI and ML is being developed and utilized as it is put into practice. Though the study of artificial intelligence is often thought to reside in the domain of computer science, a branch of engineering, machine learning (ML) and AI are not yet engineering [7].

With rapid and increasing motivation for adoption [23] of Artificial Intelligence (AI) and Machine Learning (ML) techniques throughout industry, public service, professional and academic communities [22] to assure unbiased and reliable systems, information systems researchers must examine the perceptions and assumptions being made at all levels of deployment. This study examines the effects of bias in artificial intelligence systems and considers ways MIS researchers can contribute to the underlying design and function of our learning systems, particularly focusing on activation functions in neural networks, to eliminate such biases. For reliable and unbiased predictions, it is crucial to understand not only what goes on inside the "black boxes" of AI, but also user perceptions that drive data selection and preparation, hyperparameter adjustments and evaluations of results. AI systems do not learn well in context, nor do they generalize reliably [20]. Given their narrow limitations for "learning"—they may be better described as powerful correlation machines—the training process is largely dependent on user ability for data preparation and hyperparameter design [24, 27]. Also, to understand how bias affects outcomes, IS researchers must employ their broad systems perspective to examine not only how practitioners skillfully apply technology but also how they perceive and approach both the problems they anticipate solving and the opportunities they expect to exploit with AI applications [28].

This study examines the impact of static activation functions, i.e., AFs as hyperparameters, to determine whether they restrict model learning ability, performance and generalizability. The study also explores whether dynamically setting AFs as parameters, can improve model performance in light of the predictive challenges outlined herein and, thus, a dynamic activation function's possible ability to generalize.

The central research question of this study is:

> Will changing activation functions at the node layer level during learning improve neural network accuracy and discrimination performance?

Executing this study will result in several outcomes. One, the study will result in greater understanding of the mechanisms underlying NN processing. Two, the study will demonstrate how computationally efficient and flexible AF selection affects NN performance in business contexts. Three, research into algorithm bias and the possibilities of more generalizable systems may lead to useful insight into what is contributing to the slow adoption of these powerful technologies for mission-critical applications [27].

Chapter 2

Literature and previous research

The proposed study draws on and extends foundational concepts in the Management Information Systems (MIS) field, prior studies of bias and problem formulation in MIS, and the design and implementation of neural networks. Each of these areas will be reviewed.

2.1 Research in Management Information Systems

A widely accepted and useful classification or representation of the functional relationships between data, information, knowledge and wisdom is the DIKW continuum model: data – information – knowledge – wisdom [31]. Along this continuum, data is combined and processed to become useful information. Uncovering patterns in information becomes knowledge. Wisdom is considered an "appreciation of why" and an "elevated understanding" of knowledge [31].

Typical human thought and awareness hovers at the information level. Humans are not databases. Our brains are very limited as to the amount of discrete data we can hold. We need information, i.e., connected data—a phone number and an associated name—to form the neural connections that result in recall and long-term memory. We consciously, unconsciously, and habitually combine the data, our sensory perceptions and experiences, to quickly and reliably derive information. We intuitively understand that data is valuable to information and, thus, invest in its storage, management, and processing. The result is reliable information we can assess and use to improve our condition by recognizing patterns in information, assimilating and synthesizing it to produce knowledge. We can identify data sources and design processing systems to turn the data into valuable information. We have developed tools to extend our intellectual abilities. Placing thought and awareness in the context of MIS (the study of a specific class of intellectual tools), the early paradigm was concerned with how data is efficiently organized and processed to provide information. This work was and continues to be extremely useful and valuable with the use and value now increasing exponentially.

The tools to accomplish the early goals—database management systems, algorithmic programming languages, systems analysis and design principles—could be constructed and improved with an intellectual honesty that generally avoided bias. For example, problems in data storage or management were revealed as poor design (inappropriate normalization) or a "bug" in the database software and quickly resolved. Moving from managing data and processing information to decision support systems (DSS) meant practitioners, working from their information level perspectives, were no longer just looking down the continuum into data but were designing systems to move up the continuum toward knowledge production. That new effort—enabled by accelerating advances in data storage capacity, expanding access to external data, and exponentially increasing processing capacity—has evolved into computing machines that can artificially learn by pattern recognition. Pushing the paradigm up from our human level of information-ness requires information management tools that, unlike a DBMS or a coding language, are open to bias in how the systems are built and parameterized.

This research is central to MIS. Though artificial intelligence (AI) is often thought to reside in the domain of computer science, a branch of engineering, machine learning (ML) and AI are not yet engineering [7]. They exist in a realm where the recent burst of research and commercial applications activity are happening side-by-side, largely due to advances in processing capability offered by graphics processing units (GPUs), fast RAM, massive data storage capabilities, the prevalence of available data, and cloud computing. Managers and decision makers across numerous organizations are confidently making decisions based on results from ML and AI algorithms [21, 27]. Though only about 8% of AI and ML applications are core IT functions, many across commerce, professions and academia are basing decisions in critical areas such as disease diagnosis and care plans, engineering assessments, cyber security protocols, investment opportunities, driverless transportation and the like on this new yet increasingly user-friendly technology [27]. Routine and mundane decisions and planning are also being shifted more and more towards AI [21, 23, 27]. Given the proliferation of the applications and new development, and with the model building and programming being more of an art based on experience [7] and not yet rising to that of an engineering discipline, research into its use and application fits squarely in the purview of MIS. Moreover, MIS researchers have an obligation to understand the challenges for research, didactics and practice while contributing to the paradigm as it evolves [40].

While AI and ML are thought to be engineering disciplines, they are actually more suited to MIS paradigms. The example Jordan [7] uses to convey this idea is seminal. For centuries, the practice of bridge building was a situational necessity and not an engineering practice. Bridges were often built to allow passage over or through a challenging area to traverse. The focus of the bridge construction was on its usefulness for the moment, and possibly near future, and not for a generalizable structure or lasting appropriateness for the terrain. Thus, the engineering requirements of reproducible and reliable design, terrain matching, structure failure testing, and so on, were not met. Bridge building was, therefore, not a domain of engineering. It was not until all the requirements of objective engineering disciplines were brought to bear on design and construction were bridges then deemed safe for repeated passage and verifiable for adding knowledge and practices to the engineering discipline. Because we do not have an objectively verifiable discipline for AI and ML that produces repeatedly reliable outcomes, we cannot consider AI and ML to be purely engineering. Thus, AI and ML research stretches toward philosophical questions spawned by new technical systems and how those engage in a social organization. MIS is the logical paradigm for this research activity [29]. The editor of *MIS Quarterly* stated it thus:

> *"Research in the information systems field examines more than just the technical system, or just the social system or even the two side-by-side; in addition, it investigates the phenomena that emerge when the two intersect. This embodies both a research perspective and a subject matter that differentiate the academic field of information systems from other disciplines."* [30]

MIS theorizes about the use of management information tools to improve understanding and practices where data and humans and policy meet [40]. Fundamental to MIS is theory about how information tools manage and combine data to form information [40]. New theory for tools to discover patterns in information to form knowledge is just as fundamental. The discipline considers the value of normalization of data for efficient and reliable data storage. It considers the subtle but critical difference between "for" and "while" loops in programming languages to assure reliable and efficient data processing. Data storage and programming languages are, at their fundamental levels, engineering pursuits. Engineering assures the core reliabilities and provides a solid foundation for applications. While MIS provides the theory for the application of these tools, the discipline cannot avoid the engineering fundamentals as a requisite for understanding. The same holds true as the tools evolve from a focus on data management to information production to information pattern discovery to yield knowledge. Theory to understand how practitioners perceive, understand, and interact using information tools and how those tools evaluate pattern discovery is, thus, a domain of MIS. A robust information tool for pattern discovery is a neural network. And central to how patterns are discovered by NNs, similar to how pointers are created for data storage or how data flows through code, is how NN activation functions, selected by practitioners, affect the broader artificial learning ability of the system.

We routinely use products to solve problems for us where we do not understand or fully comprehend the mechanisms of how those products work [27]. While this practice is commonly accepted and harmless for most users and uses, there are examples where it occurs in critical situations. The most notable example is in drug development and use. The inserts that accompany all prescription medications outline the chemical structure of the drug, its side-effects, and its use. Also included is the drug's mechanism. On most of those inserts it is reported that the mechanism is unknown. Yet, because the drug alleviates some suffering or prevents some malady, we take it anyway. While many patients experience side-effects that are at most inconvenient or irritating, danger can increase when drugs interact. Not knowing the mechanisms of drugs means producers do not understand, and cannot predict, what may happen when those chemicals react with each other in the similar but specifically unique physiology of individuals. The same holds true for NNs. We depend on them to learn from data and then to help us make decisions, sometimes critical decisions, based on algorithms we do not fully understand, while producing outcomes from often massive amounts of data. Not understanding the mechanisms of effect sizes among data having limitless features and ranges prevents us from quantifying an objective quality coefficient of our outcomes and predictions.

AI and ML, and in particular NNs, produce outcomes that are constrained by the data used to train the particular system. Best results, defined by accuracy of prediction, are when test data sets are randomly selected from the overall collected data with the system being trained by the residual data and used to verify output. While this approach can produce impressive results and be quite useful, it fails to generalize. For example, consider the following:

Figure 2.1. Wolves vs Huskies [53]

The images in Figure 2.1, are from an experiment where a NN was trained to classify the difference between wolves and Husky dogs [37, 53]. Numerous examples of photographs of each were used to train the system and it eventually achieved an accuracy rate of around 90%. Then, when the researchers ran algorithms to reveal the inner-workings of the NN and focused on what features were driving the classification, they discovered that the system had learned based on the image backgrounds—wolves were generally photographed on snow while the dogs were photographed on grass. The excellent performance of the NN would likely meet the threshold of practitioners looking for confirmation for a use yet would be a unreliable system for generalizing; i.e., should a test photograph contain a dog on a snowy background or a wolf standing in a grassy field.

This simple illustration reveals how a NN can underperform as a generalizing classifier. Understanding why the NN chose the image backgrounds as the primary feature for classification is key to understanding how to better design and deploy AI and ML. Discovering it after the fact, perhaps after a practitioner has deployed the NN in production, is a suboptimal outcome. A problem that can be discovered should be discovered during training and corrected—particularly in systems where training already involves dynamic adjustments. To solve these problems researchers can explore questions surrounding the role of activation functions as they are central to how a system "learns," such as whether dynamically changing activations by selecting new ones from a discrete list as training progresses in response to some signal from the loss (cost) function improves results. These fundamental issues of NN architecture, data flow, adjustment and refinement also offer possible solution inroads to learning in context, i.e., the wolf versus Husky example. AF selection as a dynamic refinement during training may hold the solution for smoothing progress toward convergence while retaining context yet not diminishing the object of the learning. These approaches are being explored and implemented as self-learning systems, yet they are primarily directing adjustments at hyperparameters as opposed to dynamic parameter adjustment during training [28].

While the development of this capability may be a challenging task, the notion of learning in context is an important contribution to achieving artificial intelligence. Anything short of that is development of artificial memorization, or a good correlation machine. The fact that all NNs adjust AFs with weights and biases suggests that there is an inefficient and perhaps inappropriate use of activation functions in presently deployed neural networks. If the output of a nodal AF needs a shift, a bias implemented to best inform the next node, then perhaps it is possible to adjust the nodal AF itself during back propagation; i.e., to treat it as a parameter, to provide a value that is a better representative of the local contribution to learning.

When linear progress was the norm, over the years leading up to the early 1990s, the risks of some new technology could be considered and resolved in the same contextual time frame as the development of the new technology. More recently, with technology capacity and growth becoming exponential, the risks associated with adoption and use are becoming greater as well [33]. Our ability to recognize and reconcile risks is lagging our adoption of new technology. This sets up a dangerous conflict as managers eagerly accept and adopt new information technologies [32] as conveyed by Buckminster Fuller:

> *I am aware that humanity*
> *Is approaching a crisis*
> *In which its residual ignorance, shortsightedness*
> *And circumstance-biased viewpoints*
> *May dominate*
> *Thus carrying humanity*
> *Beyond the point of no return*
>
> R. Buckminster Fuller

2.2 Biases

Enid Mumford anticipated the challenges of computational progress accelerating past recognition, comprehension, and resolution of the associated risks and, in 1983, offered a framework for creating "humanistic and friendly" as well as "efficient and effective" systems [34]. Her framework is defined in the acronym, ETHICS: Effective Technical and Human Implementation of Computer-based Systems. It is a socio-technical approach defined by Mumford as "one which recognizes the interaction of technology and people and produces work systems which are both technically efficient and have social characteristics which lead to high job satisfaction [34]."

Practitioner bias contributes to model design, data selection and preparation and, in turn, unreliable and unreproducible outputs. Unintentional bias is a compelling and difficult to control feature of AI and ML practices. Consider the following series of numbers:

4, 9, 3, 6, 5

What is the next number in the series?

Asking this question to several people will result in different answers. After a while, the answers will begin to fall into categories based on how the respondents approached the problem. Some respondents begin by subtracting one number from the next, or exploring some arithmetical approach, and, in doing so, arrive at some eventual value that, added to the last number, produces the number in question. Others, typically those familiar and comfortable with statistics, mentally compute a regression and suggest an answer based on that. Others see no pattern and believe some random number, usually in the range of the stated numbers, is the best answer. Still others mentally compute weighted or moving averages to get an answer.

This is the work of problem-solvers: A commitment to solving the problem and arriving at a solution, the "right" solution. The motivation to solve, to be right, along with the assumption that there must exist a solution that at least approaches an objectively confirming right answer, creates a momentum for solving that can ignore a problem-approach bias.

The problem-approach bias, among other types of human biases, are not new when evaluating intellectual tools and those who use them. While Vasarhelyi [38] examined the analytic-heuristic dimensions of individuals and observed, based on experimentation, that, "analytic type people tend to use computers and other analytic tools more in planning than do heuristic types," Sage identified twenty-seven biases commonly found to be present in designing and creating analytic and decision support systems [37]. Among them are examples that are being identified as problematic in the high-data volume, fast-processing domain of AI and ML. Chief among those are: The desire for self-fulfilling prophecies, expectations, fact-value confusion, representativeness, habit, hindsight, illusion of correlation, order effects, outcome irrelevant learning system, overconfidence, redundancy, representativeness, selective perceptions and wishful thinking [37]. There are, conversely, a number of biases outlined by Sage that can be minimized or eliminated by newer prediction and analytical systems. It is not in the scope of this study to assess specific biases in detail; however, it is imperative to consider those that seem to persist among the use of generational intellectual tools and evaluate how they impact the exponential deployment of AI and ML. A summarized general view of human bias, whether designing an AI or ML system or solving problems such as the number series question above, can be described as one's focus on how well the calculation predicts future values versus how well it fits past values.

Though the numeric answers to the simple puzzle differ, the compelling insight is not the number quoted as the answer but, rather, the approach to solving the problem. The varying approaches represent practitioner bias. There is no objective answer as there is with two-plus-two equals four; it is merely a random set of integers where one tries to predict the next integer by discerning some pattern or relationship. That is exactly what a NN does with the data it is fed. But the approach to the problem, the effort to remove the randomness, or to identify a pattern in the randomness, affects the outcome. Deeming that the problem has been solved "accurately" reveals a connection between the approach to the problem and the outcome that is at least as important as the data itself. While NNs are efficient and do an excellent job at identifying complex and hidden patterns in data and, then, provide us predictions or classifications that satisfy our accuracy definitions, we must be aware that our notions of accuracy are *a priori* constructs that are tied to our approach to problem formulation. These notions are the domain of MIS, not engineering. Problem formulation as it affects decision making has been studied extensively in MIS [41, 42, 43, 44, 45, 46].

2.3 NN Design and Implementation

The performance of neural networks, and artificial intelligence and machine learning algorithms in general, is heavily influenced by the individuals who design and implement them [24]. The programmers typically begin the process with pre-processing the data set, then selecting a neural network (NN) design based on the data (i.e., a convolutional NN for image processing, recurrent NNs for sequence problems, etc.), setting the hyperparameters, choosing the number of layers and nodes, determining optimal training and testing data splits. Based on some result, normally an accuracy reported as a percentage of true positives during the testing phase and Receiver Operator Curve (ROC) results, parameters and structure of the NN are adjusted until the results produce an "acceptable result." This "acceptable result" should be verifiable that it is indeed accurate and represents a true and independently supported outcome. Of course, there are significant challenges with verifiability if AI and ML are to be used to solve problems that have no prior solution set. Thus, we are left with the critical task of assuring that all aspects of the approach to the problem are in order: input data, design of the AI, integrity of the system, and verifiability of subsets of outcomes as possible.

While this summarized view of the procedures of training a NN reveals how it is available to anyone, there are additional details to consider for understanding the scope of the activity. During the NN training process, which is normally highly iterative, the practitioner may adjust a variety of parameters and structural options of the NN such as controlling the batch size, the number of epochs to run, the learning rate, the number of hidden layers, the number of nodes of the various (input, hidden, output) layers, selection of the loss function, and choice of the activation functions (AF). Some NN applications constrain users to select an activation function for the entire NN while most have evolved to allow setting AFs for specific layers. During learning, the NN itself makes certain adjustments during the learning process to improve its performance. These automatic, internal adjustments to improve model learning occur during both feed-forward and back propagation phases. These adjustments result from measuring its own performance using a loss or cost function which compares nodes at the output layer with the data labels and generally taking one or more of the following three actions: 1) adjusting the weights of various edges connecting nodes throughout the NN, 2) adjusting the biases that control the range of output, or effect, of each node's activation function, and 3) adjusting the learning rate during gradient descent.

While these processes of data preparation, parameter setting, and NN topology design allow substantial control of how the artificial learning process computes and approaches conclusions, what is essentially happening is the creation of a powerful correlation model. This approach mimics artificial *memorization* rather than artificial *intelligence* as the broader model adjustments are performed by the designer/programmer and the useful scope of the result is tightly constrained by an interpretation of balancing overfitting versus generalizability of the specific data. Thus, the resulting algorithm is only practically useful for producing predictions from very similar data sets and not suited to generalization [25].

These two automatic adjustments (adjusting weights and biases), combined with, in most NN algorithms, dynamic learning rate adjustment, are what drives NN performance during training and indeed are the only adjustments involved over the epochs of a learning session. The network continually adjusts itself until a learning threshold; for example, a gradient descent is at zero slope, is reached or the data is exhausted.

It is plausible that activation function selection is a limiting factor, perhaps a critically limiting factor, in NN performance. Evaluations of current practices point to such:

1. Performance metrics can vary widely if no parameters of a NN model are allowed to change except for activation function selection.
2. Activation functions are created to best model certain types of decision requirements and are thus better suited to particular data types than others. Too, data types can vary within a data set which forces data to be evaluated by AFs regardless of suitability.
3. NNs are evolving to allow researchers to specify activation functions for particular layers and, with some algorithms, to swap to a different AF in response to learning, a notion that static, model wide activation functions do not produce the best results [47].
4. Researchers have recognized the key role of activation functions and how they may be automatically adjusted during training to optimize output [1, 6, 13, 14].
5. Bias is one parameter a NN changes during training and its purpose is to adjust the output of an activation function. This raises the question of whether the NN dynamically changing the AF itself could be a more effective approach.

While there exists strong, peer-reviewed literature on the technical construction and deployment of neural networks for deep learning, much of the recent advances are reported in conferences, meetings of researchers in numerous fields populated by those who research the technology itself and those who use it for applications, by commercial enterprises and research firms deeply engaged in AI as a core organizational function, and through open sharing of papers. Too, information about practices, ethical concerns, commercial application outcomes, usage perspectives and similar subjective insights are shared through numerous news outlets, practitioner journals and various media resources.

At the core of this inquiry is a 2004 paper by Majetic, et. al. [13] that describes how researchers achieved some success in their "dynamic neuron model" using an "adaptive Gauss activation function". Figure 2.2 is the researchers' proposed Dynamic Neuron Model.

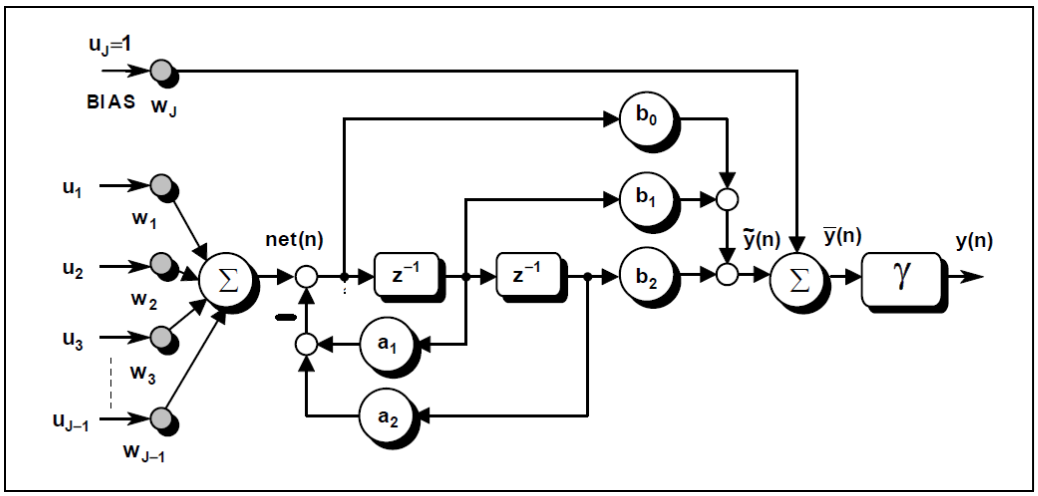


Figure 2.2. Dynamic Neuron Model

"The basic idea of proposed (sic) dynamic neuron concept is to introduce some dynamics to the neuron transfer function, such that the neuron activity depends on the internal neuron states." [13] The goal of Majetic and his team was to create an "infinite impulse response filter" so the nodal neuron would process its own past activity and new input signals. This design is common in electrical engineering digital filters and is distinguished by having impulse responses that, unlike finite response filters, do not become zero at some time past some predetermined time [39]. This property, then, provides the ability to dynamically adjust neuron values through its impulse filtering mechanism as opposed to adjusting the nodal bias and connecting edge weights.

Majetic et al. chose to use a chaotic Glass-Mackey Time Series [13] as test data due to its simplicity to implement and difficulty to predict. Their Glass-Mackey time series, with 1,000 time steps, is shown in Figure 2.3.

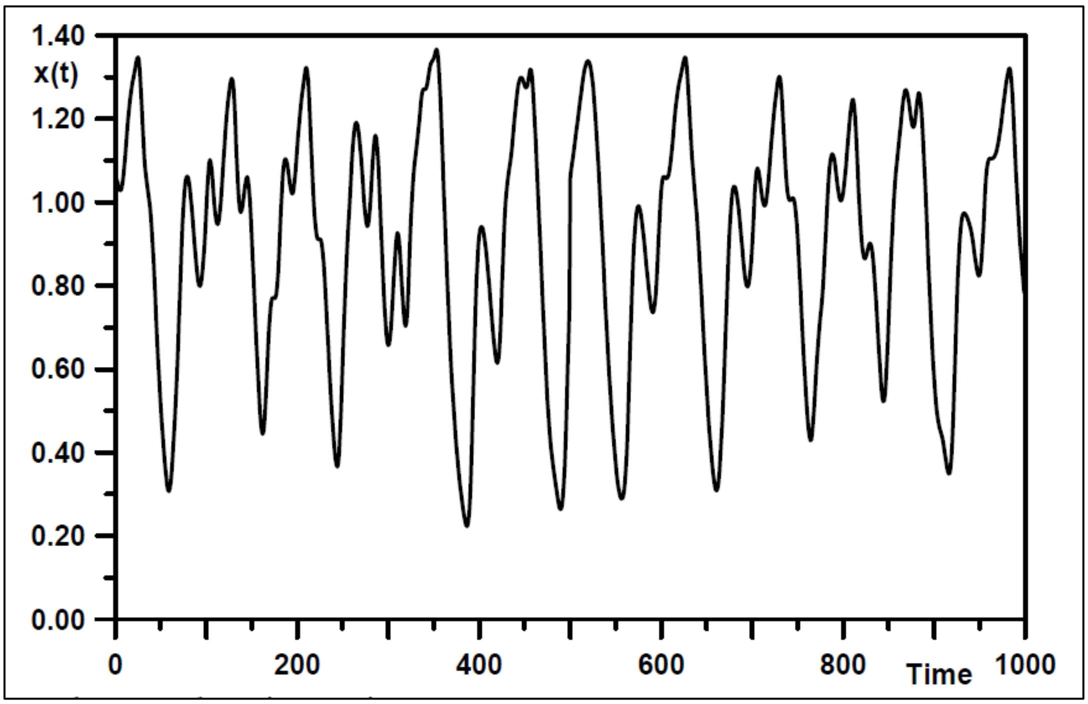


Figure 2.3. Majetic, et. al., Glass-Mackey time series

The goal of Majetic's dynamic neural network, consisting of three layers (input, output and hidden layer with only five or ten nodes), was to predict values *x(t+P)* at some future point *P* after being trained on values up to the point *x(t)*. The results are shown in Figure 2.4.

Neuron AF	Unipolar Sigmoid		Adaptive Gauss	
Network Topology	5-10-1	5-5-1	5-10-1	5-5-1
Learning Epoch's	70000	80000	35000	50000
Learning (NRMS)	0,069	0,053	0,027	0,057
Test 1. (NRMS)	0,069	0,071	0,048	0,043
Test 2. (NRMS)	0,071	0,067	0,052	0,058
Test 3. (NRMS)	0,073	0,078	0,050	0,052

Figure 2.4. Majetic, et. al., dynamic NN results

The researchers' Adaptive Gauss neuron model, with similar loss learning rates, achieved fewer errors (normalized root mean square error, or NRMS error) in about half the number of required learning iterations.

While their model required half the learning steps in the same time as a typical model and achieved better generalization, there has been little follow-up in the academic community to that paper. Rather, the focus has largely been on model-construction, generally, and improving hyperparameter efficiencies. While the paper was updated in 2007 [14] the overall insights into a Gaussian activation function for a dynamic neuron were essentially the same. The researchers concluded, based on their experimentation, that though their proposed adaptive Gaussian activation function required nearly twice the learning iterations, it resulted in faster learning and better generalization that the baseline sigmoid AF.

A different approach to a dynamic NN was proposed by Munkhdalai et al., [16] in their 2018 paper on Conditionally Shifted Neurons (CSN). Their CSN model requires splitting the learning into two subfunctions: A *base learner* and a *meta learner*. The architecture is shown in Figure 2.5.

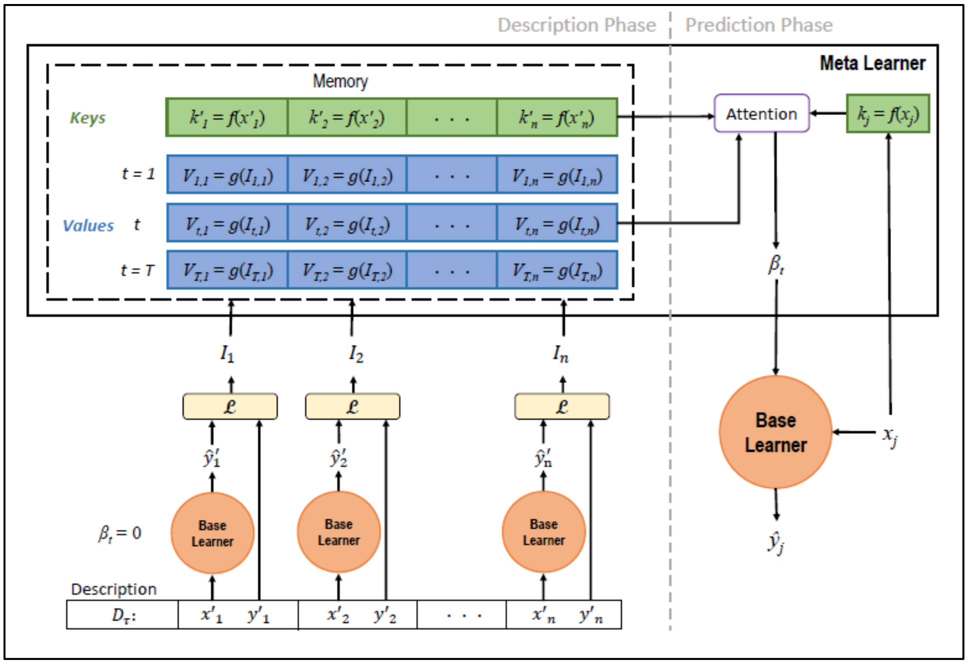


Figure 2.5. CSN base and meta learner model

The base learner holds the learned values and makes predictions on inputs for each node while the meta learner is used to store and retrieve the activation shifts to accelerate learning. While this NN architecture is interesting, the value to this study is in what can be learned from how Munkhdalai utilized two variants of conditioning (*meta learner* to *base learner* connection), one based on commonly used error gradients and the other based on feedback alignment methods. The author reports that the latter, feedback alignment, proved more efficient as it does not require a sequential backpropagation procedure. If, then, neurons can learn “from themselves” based on some combination or *meta learning* and the above described “infinite impulse response filter” [13], a path to dynamic activation functions, based on previous and verifiable research, has been established and can be evaluated by classical, proven statistical techniques as described by Hutton [6].

To study the effectiveness of competing neural network designs, it is imperative to select a standard by which to measure results and evaluate each. Hutton's 1992 paper addressed the evolving and growing NN research field [6] and reviewed common statistical techniques for evaluating NN performance such as the Receiver Operating Curve (ROC) and the Root Mean Square (RMS) error. While these are beneficial for improving a particular NN, Hutton offers guidance on using statistical techniques for evaluating NN methods beyond a results-focused approach and broadens the scope to data preprocessing steps, training and testing splitting, noise, and overtraining. These broader considerations will be necessary for a fair evaluation of any NN systems relative performance as the implementation of any NN necessitates consideration beyond its topology, activation function selection, parameters and hyperparameters. When comparing NN efficacy among models, it will be imperative to control for factors that can introduce bias. Thus, Hutton provides a broad framework of considerations for selecting appropriate statistical models to reveal possible bias and control for unavoidable circumstances.

In a 2000 paper, Pizarro, et al. [48], proposed a statistical methodology for evaluating neural networks based on a multiple resampling technique rather than the typically employed selection where several methods are employed and the one with the "best" score is selected. While the procedures proposed are novel, they are based, generally, on accepted ANOVA techniques. The benefit to the proposed study is that the Pizarro implementation evaluated NNs to provide insight into improving the design topology of the nodes contained and positioned in the NN. That, in turn, can likely provide insight into guiding the backpropagation for dynamically adjusting activation functions to improve learning and accuracy.

Chapter 3

Method

This chapter will refer to several terms that, for clarity, are defined in the following table:

Accuracy	The number of true positives and true negatives divided by the number of data points. The number of correctly predicted datapoints out of all data points.
Activation function	A function residing at nodes in neural networks that converts input values to slightly different output values to prevent linearity in the overall network and, eventually, through the learning process, result in an operator that contributes to an ability to predict with accuracy.
Batch	A subset of the training data set selected to initiate backpropagation upon completion. Each batch trained completes one iteration.
Backpropagation	An algorithm that effects small changes to the various weights and biases, and sometimes learning rate, with the degree of change determined by the loss function, to enable convergence.
Bias	A parameter function that slightly shifts values entering a node to effect a direction or value for processing by the activation function. This is distinctly different from bias as a prejudice. Herein, biases exhibited by users will be referenced as user bias. Any other use of bias will refer to the bias function of a node.
Boosting	An ensemble method for improving learning.
Convergence	When loss moves towards a minima or exhibits a decreasing trend.
Cost (loss) function	A function that compares output layer values to actual values and computes the amount of "penalty" for adjusting weights and biases for backpropagation.
Dropout	A regularization technique where activation functions are randomly set to zero (along with their connections) to force learning along another pathway. Dropout is routinely used in neural networks to combat overfitting.
Epoch	One complete cycle of the training data set. May be completed in batches or as one single batch.
Feed-forward	Feature data progressing through the input, hidden and output layers of nodes with the various activation functions, biases and weights adjusting the values along the various feature paths.
Hyperparameter	A value set by neural network users to control the learning process.
Iteration	One batch of training data. Backpropagation is initiated at the conclusion of an iteration.

Learning	The process of a neural network iteratively adjusting feature data values, one instance at a time, through the feed-forward and backpropagation cycles to achieve convergence.
Parameter	A configuration variable, internal to the NN model, that can be adjusted by the feature data through the loss function.
Regularization	A collection of processes (L1, L2, dropout) to prevent over-fitting.
Weight	A parameter value that adjusts feature values leaving a node for processing at the next node along the feature value's pathway through the feed-forward process.

Table 3.1. Definitions

Deep learning neural network algorithms that allow changing the activation functions at the node level during training are not available. Most provide the ability to set activation functions at the input, hidden and output layers, as hyperparameters, prior to training but they cannot be changed during training. Given these limits, the proposed study requires a NN simulator. This simulator was built and tested. Verification of the algorithm fundamental to the NN—an input feature value adjusted by all weights, biases and activation functions through the output layer—were manually computed. In the first phase of the study, a simple hand calculated NN using typical functions for cost, weights, and biases along with an activation function selected from a vector of activation functions was examined for likelihood of effect. In the second phase, a computer simulation that embodies the same method but on a larger scale provided results to reveal how user choices in setting AF hyperparameters affect training over a large, controlled data set.

The results of the manual NN calculations were evaluated qualitatively to investigate whether AF selections as hyperparameters introduced user bias into the system and whether dynamically changing AFs during training improves outcomes in terms of convergence. The results of the larger, automated NN runs were compared using the built-in measures of accuracy measurement during training to determine whether dynamic adjustments to nodal AFs, i.e., treating a nodal AF as a parameter, smooths and accelerates convergence and whether it can mitigate bias. These results were assessed using the Kolmogorov-Smirnov test (K-S Test) to determine whether the results obtained from the manual calculations and the automated calculations were significantly different.

To exhibit the scope of the issue being addressed, a series of NN training and testing sessions were run using an array of activation functions and results recorded for statistical analysis [6]. Previous work with fraud detection NNs [26] has revealed curious training accuracy patterns (Figure 3.2) among various and commonly used activation functions. These results show an interesting contrast with other results, such as learning progress of commonly used AFs during training of the MNIST data set (Figure 3.1). Such differences may suggest that activations functions have been designed, and are chosen, based on a training performance and convenience bias. This study used the preliminary work (Appendix) as a baseline for assessing NN learning where AFs can be dynamically assigned as proposed by the research.

The preliminary work included coding a neural network designed to record accuracy during training of various AF combinations. While some combinations showed a fast learning rate with a smooth learning curve, those (relu/relu, relu/tanh, Figure 3.1) did not result in the highest accuracy. Two other combinations (tanh/tanh, tanh/relu, Figure 3.1) exhibited slightly slower learning, and both with a distinct bend in the training progress at around 400 iterations, but with significantly higher accuracy. These preliminary results, combined with those revealed in the fraud study [26], invited a closer inspection into how activation functions may be utilized as parameters to improve learning rate and accuracy among a wide variety of data sets.

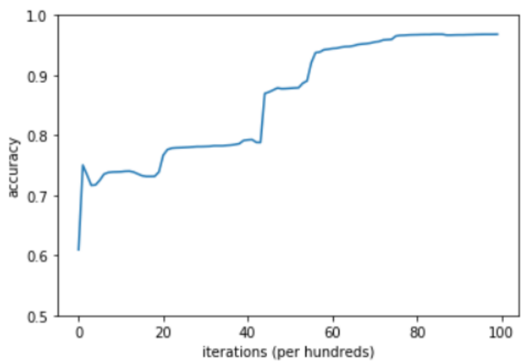
forward/ back of hidden layer Iteraton	relu /relu loss	relu / sigmoid loss	tanh / relu loss	tanh / tanh loss	relu / tanh loss
0	0.692120	0.692119	0.692608	0.692608	0.692119
100	0.572353	0.587775	0.527395	0.571761	0.571426
200	0.408191	0.509407	0.441207	0.440472	0.428149
300	0.311967	0.444694	0.359892	0.358198	0.340257
400	0.261004	0.391205	0.325459	0.317150	0.295625
500	0.217883	0.351961	0.300846	0.274086	0.254280
600	0.194391	0.324997	0.268156	0.220072	0.232096
700	0.180031	0.304279	0.235724	0.179712	0.214843
800	0.170005	0.284999	0.210881	0.152189	0.201363
900	0.162347	0.265844	0.191115	0.132651	0.191248
1000	0.156166	0.250050	0.174751	0.118280	0.183407
1100	0.150990	0.240214	0.160750	0.107340	0.177082
1200	0.146531	0.234508	0.148711	0.098754	0.171793
1300	0.142580	0.230948	0.138255	0.091837	0.167241
1400	0.139023	0.228119	0.129376	0.086143	0.163220
1500	0.135750	0.225396	0.121732	0.081368	0.159601
1600	0.132743	0.222593	0.115133	0.077302	0.156292
1700	0.129900	0.219722	0.109335	0.073797	0.153228
1800	0.127303	0.216817	0.104194	0.070741	0.150365
1900	0.124824	0.213923	0.099638	0.068052	0.147670
2000	0.122480	0.211063	0.095604	0.065668	0.145119
2100	0.120253	0.208262	0.091994	0.063537	0.142690
2200	0.118129	0.205530	0.088705	0.061622	0.140368
2300	0.116090	0.202871	0.085724	0.059891	0.138141
2400	0.114150	0.200286	0.083055	0.058319	0.135999
Train accuracy	90.13%	90.13%	98.11%	96.18%	90.13%
Test accuracy	90.20%	90.20%	98.23%	96.06%	90.20%

Figure 3.1. Preliminary AF combination results of coded NN; MNIST Data Set

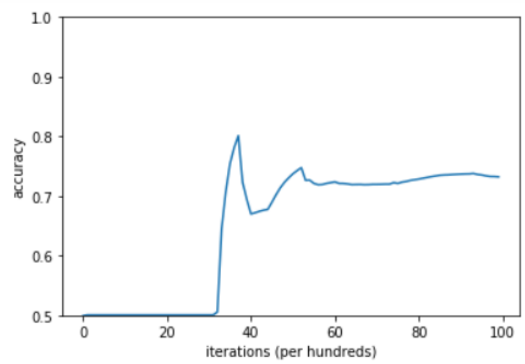
In the introduction of this study three outcomes were outlined as goals for this investigation. The process for achieving those goals is described next.

First, a greater understanding of the mechanisms underlying NN processing was sought. To achieve this, the internal accuracy measurements (the difference between output node value and the label value for the iteration) over learning iterations from different activation function combinations (Figure 3.2), "programmed to allow maximum flexibility, as allowed by the sklearn library" [26], collected during an investigation into fraud detection [26], were taken to form the methodology basis for an expanded and AF focused inquiry. The results provided insight into how various activation functions impact the learning process (in terms of loss over iterations and test data accuracy) over common data and hyperparameter selections, which are largely based on experience or trial-and-error [50], providing a baseline for this broader study. The study used the MNIST data set for testing as it provides a sufficiently large yet manageable (given AI platforms available for this study) set of data and has a history of referenceable and verifiable performance over a wide variety of processing environments. The hyperparameters (learning rate, cost functions, initial weights and biases) and computer platform were held constant throughout all training sessions to assure the measured effects would be influenced only by the activation functions.

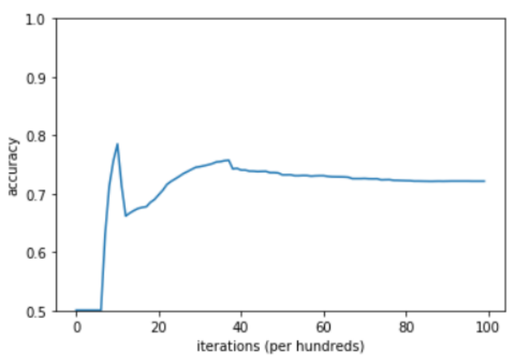
Below is a NN to report accuracy at intervals over numerous training iterations, using various combinations of feed-forward and backpropagation AFs. Figure 3.2 shows results from training runs that were used for a study [26] of fraud data using various AFs as an example:



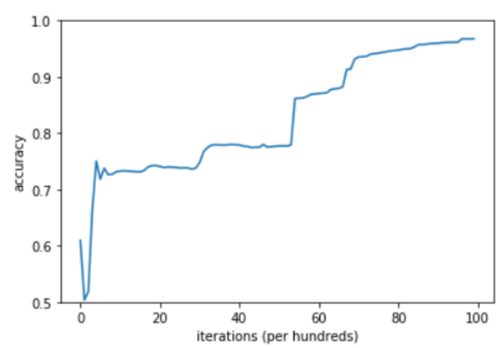
a. Forward hidden/output: ReLU/ReLU
Backprop hidden/output: ReLU/ReLU



b. Forward hidden/output: sigmoid/sigmoid
Backprop hidden/output: sigmoid/sigmoid



c. Forward hidden/output: sigmoid/tanh
Backprop hidden/output: sigmoid/tanh



d. Forward hidden/output: ReLU/sigmoid
Backprop hidden/output: tanh/sigmoid

Figure 3.2. Fraud detection data accuracy with various AFs

There are 10,000 training iterations for each of these runs and accuracy was measured at intervals of 100 iterations. Sampling training accuracy during the runs revealed that training is complicated by spikes and saddles that, while eventually, over enough iterations, tends to converge, there is risk of declaring training success at some point during the process. The plots in Figure 3.2 reveal these complexities; because other hyperparameters were controlled, the variance in accuracies is the result of choosing different combinations of activation functions as hyperparameters.

The code created for this study (Appendix 3) allows changes to AFs during backpropagation based on simple, codable parameters and can measure change in accuracy and cost at selected iteration steps as training progresses. It measured improvements in accuracy when training on the MNIST data set where each instance of a digit is treated as a vector with 784 features. After 2,400 training iterations, and compared with an initial training loss of 0.692, relu/sigmoid had a measured loss of 0.200 while the measured loss for tanh/tanh was 0.058, for a 28.9% improvement in loss resulting in an 8.9% improvement in accuracy (Figure 3.2) when changing AFs during training.

Two, the study was designed to demonstrate how computationally efficient and flexible activation function selection affects NN performance in business contexts. To do this, the experimental neural network was fitted with loss and accuracy recording points to save progress through the learning processes among various activation functions. These detailed performance catalogues provided rich resources for evaluating the convergence progress among AFs in a controlled environment common to typical business demands.

Each experiment to establish a measurement baseline began with setting the learning rate, initial weights and initial biases hyperparameters, along with the processing environment (platform, available memory and GPU), and with a pre-split data set of training, validating and testing instances from the MNIST data set. Activation functions were chosen for input, hidden layer and output nodes from among Leaky ReLU, ReLU, Sigmoid, Softmax and TanH (Figure 3.3). These five were chosen due to their common use in business applications. The loss values collected during the training iterations will be recorded and saved along with accuracy values from the testing data set.

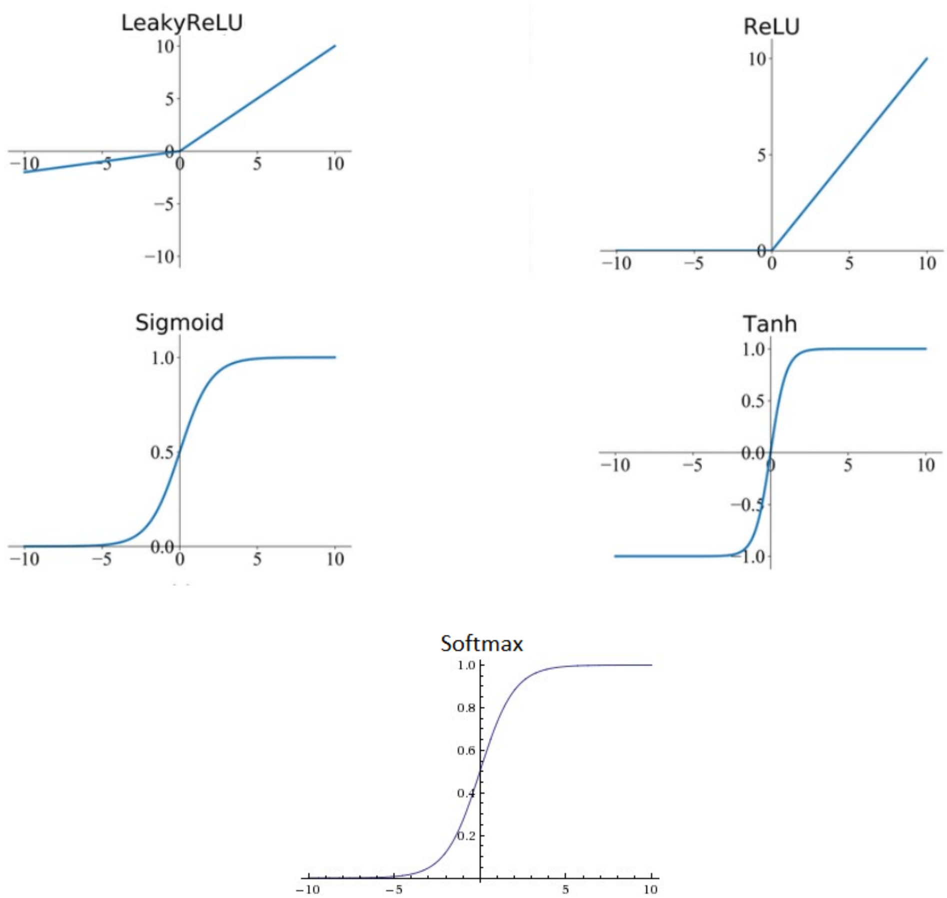


Figure 3.3. Activation functions used in the study

The data from these studies was evaluated using the Kolmogorov-Smirnov Test (K-S Test), a nonparametric test used to compare the equality of continuous distributions, to determine if a reference distribution of loss progress through training could be established. If an insignificant difference between loss distributions was established, thus resulting in a reference distribution, then a baseline learning curve was established. If not, trials were rejected until a baseline was established and, as a result, narrowed the field of available activation functions to those contributing to the baseline during traditional learning as hyperparameters.

The final step was switching the NN algorithm to allow dynamic switching of activation functions, selectable among those contributing to the baseline, during training to evaluate if loss could be further reduced and the learning rate and accuracy could be improved. The same environment and hyperparameters will be set and then trained on the identical training data used to establish the baseline. Loss measurements were taken at the same iteration points and recorded. The measured accuracy of the testing set was recorded. The K-S Test was used to statistically evaluate the results against the baseline reference distribution for significance.

Dynamically assigning AFs during the process of training raises the question of how the assignments will be determined. The decision function found in the machine learning method of classification trees provided a repeatable and reliable solution for this initial investigation. The function is based on entropy and information gain. Entropy, from the field of information theory, is a measure of uncertainty. Thresholds can be set across the spectrum of available AFs and used for selecting a particular AF based on the quality of knowledge given the cost function feedback.

Figure 3.4 below depicts the relationship between entropy and probability. Entropy is zero when an event is certain not to occur (at zero on the horizontal axis) or when it is certain to occur (at one on the horizontal axis). Entropy is maximized when the event is as likely to occur as it is not, a situation often described as "fifty–fifty" (when the probability of the event is at 0.5 on the horizontal axis).

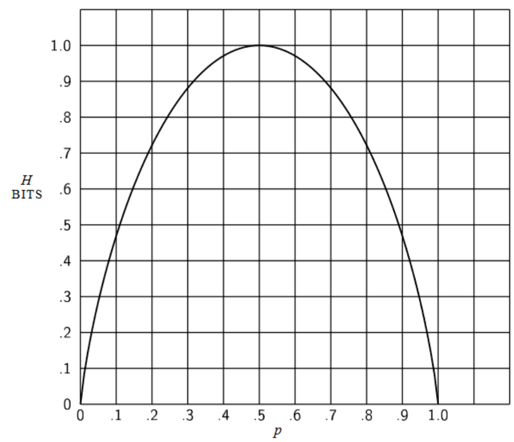


Figure 3.4. Entropy in the case of two possibilities with probabilities *p* and (1 – *p*) [49]

Figure 3.4 depicts entropy as certainty of information given two possible outcomes and probabilities, *p* and 1-*p*. Information certainty moves right across the *X* axis from zero uncertainty through perfect uncertainty, or maximum entropy, where H(x) = 1, and back to zero uncertainty. Thresholds may be set along the curve to appropriately correspond to particular AFs for assignment to nodes. The "information" in this case will be derived from the cost function, weights and biases to result in a quantity between 0 and 1.

While Figure 3.4 is an entropy model for a two-outcome event, such as a coin flip, it may be applied to multiple possible outcomes. The specific model used in information theory is that developed for multiple outcomes by Shannon [49] and summarized in Figure 3.5.

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$

Figure 3.5. Shannon Entropy Equation

The Shannon Entropy Equation is convenient and useful for hand calculations as well as for computing as a function embedded in a neural network. It will be used in this investigation to convert loss function values to entropy values for selecting nodal activation functions during backpropagation training iterations, as diagrammed in Figure 3.6.

Figure 3.6. Shannon Entropy Function as activation function choice algorithm

Three, study was designed to examine details of algorithm bias and the possibilities of more generalizable systems in order to determine useful insights into what contributes to the slow adoption of these powerful technologies for mission-critical applications [27]. To accomplish this, the study used examples, such as commonly deployed hyperparameters and activation functions (Figure 3.3) from the accepted design and implementation practices of neural networks [12, 21, 23]. These served as the design foundation to compare the loss and accuracy outcomes of the MNIST data set trained using different biases; i.e., activation function hyperparameters, to explore the effect of unintentional practitioner bias through AF selection. The results provided insights into whether hyperparameter settings unintentionally skew learning, as measured by learning as loss and accuracy values resulting from using different activation functions, and can be demonstrated to measurably affect outcomes. Dynamic changes to activation functions based on local evaluations of data, similar to localizing loss functions for specific pathways, were explored to determine whether such changes improved learning efficiency, accuracy and, in particular, in-context classification and bias reduction through parameterizing AF selection as measured by improved training loss and test data accuracy.

Required for this investigation was a neural network designed for experimentation rather than production. It needed to be capable of reporting data such as accuracy, cost, and weights and biases at defined learning progression steps for analysis. It also needed to be capable of changing activation functions during training based on cost and/or accuracy. With these requirements not being typical functions of readily available NNs, one was coded using Python (Appendix C) so hyperparameters could be set and managed along with the other required functionality and reporting capabilities.

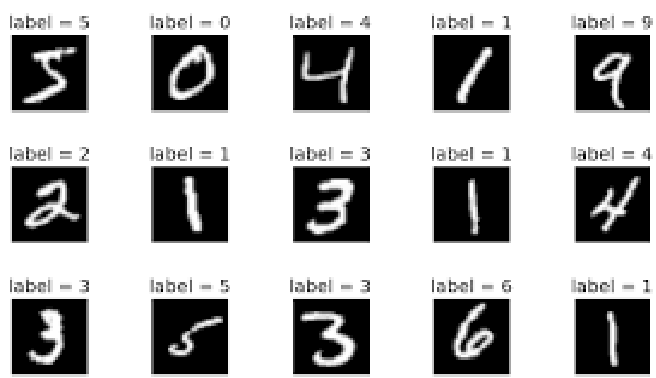


Figure 3.7. Example of hand-written digits of MNIST data set

To train the NN using the MNIST data set, Figure 3.7, as a vector requires 784 input nodes, reflective of the 28 x 28 pixel space that each digit occupies. A single hidden layer was chosen to force the activation function to have maximum effect on learning accuracy with minimal processing overhead and accommodation. This approach mirrors the previous work by Majetic, et. al., in their 2004 dynamic NN experiments (13) with the Gauss-Mackey time series data.

The ten-node output layer is standard for MNIST neural network architecture to match the ten possible outcomes and typically uses a Softmax activation function for the output layer. Softmax classifies the prediction by computing a probability for each node, each of which corresponds to one of the ten digits. The probabilities of the ten nodes sum to 1.0, with the node having the greatest probability value indicating the predicted digit. The architecture and hyperparameter settings of the neural network, for all baseline and testing, were as shown in Table 3.2.

Input layer nodes	784
Hidden layer nodes	64
Output layer nodes	10
AF, hidden layer	Leaky ReLU, ReLU, TanH, Sigmoid, Softmax
AF, output layer	Softmax
Initial learning rate	0.1
Learning rate	Initial x 1 / (1 + 0.01 x iteration)
Initial weights and biases	Randomized but constant over all sessions
Epochs	300 and 30,000

Table 3.2. Neural network architecture and hyperparameter settings

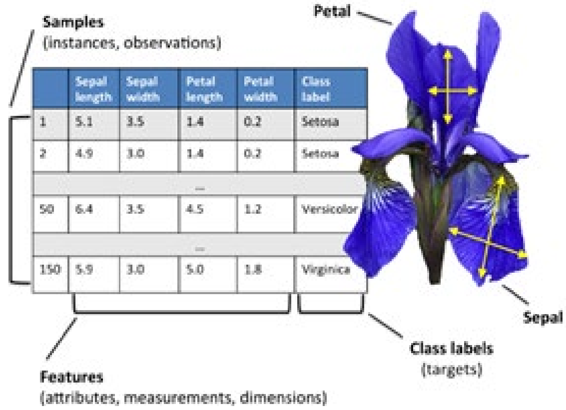


Figure 3.8. Example of the Iris data set

Training the NN using the Iris data set, Figure 3.8, required 4 input nodes, one for each of the four features: sepal width, sepal length, petal width and petal length. A single hidden layer was chosen to force the activation function to have maximum effect on learning accuracy with minimal processing overhead and accommodation.

A single-node output layer is standard for an Iris neural network to classify from among the three labels and the Softmax activation function was chosen for the output layer. The architecture and hyperparameter settings of the neural network, for all baseline and testing, were as shown in Table 3.3.

Input layer nodes	4
Hidden layer nodes	6
Output layer nodes	1
AF, hidden layer	Leaky ReLU, ReLU, TanH, Sigmoid, Softmax
AF, output layer	Softmax
Initial learning rate	0.1
Learning rate	Initial x 1 / (1 + 0.01 x iteration)
Initial weights and biases	Randomized but constant over all sessions
Epochs	300

Table 3.3. Neural network architecture and hyperparameter settings for Iris data set

All baseline and testing data gathering was conducted in single Python Jupyter Notebook sessions for each data set over a period of several days. The single session approach assured that the randomly generated values for initializing weights and biases remained the same, that the random seed for the selection of AFs remained constant, that the random training and testing data split did not change and that the sequence the instances were fed to the neural network did not change.

The hardware platform for this study was conducted on a Samsung laptop running MS Windows and Linux Ubuntu configured as shown in Table 3.4.

Computer	Samsung laptop
CPU	Intel Core i7-6500U @ 2.5 GHz, 2.59 GHz
RAM	12 GB
System	64 bit, x64-based processor
OS	MS Windows 10
Application	Jupyter Notebook, Python

a. Configuration and OS for MNIST studies

Computer	Samsung laptop
CPU	Intel Core i7-6500U @ 2.5 GHz, 2.59 GHz
RAM	12 GB
System	64 bit, x64-based processor
OS	Linux Ubuntu v20.04
Application	Jupyter Notebook, Python

b. Configuration and OS for Iris studies

Table 3.4. Platform configuration for MNIST studies

Chapter 4

Results

Training sessions were conducted with cost and accuracy values recorded. Basing AF swaps on the entropy function as originally coded (Figure 4.1a), i.e., allowing an AF swap after each epoch, produced results impossible to interpret in terms of the research question. Thus, the entropy AF selection function was changed to swap AFs when the cost value in the current epoch was greater than the cost value in the fifth previous epoch (Figure 4.1c), indicating increasing an error rate. In a process of fine-tuning the entropy test, the entropy function was further adjusted to swap AFs based on a change in cost of greater than 0.3 (Figure 4.1d). Additionally, new techniques were implemented to randomly select new AFs when the accuracy rate was zero or negative after an epoch of training (Figure 4.1b).

As shown in the figures below, results were captured at intervals of five epochs except where otherwise indicated. Baseline performances for each of the five activation functions were determined with training runs of the same data set initiated with the same hyperparameters over 300 epochs. The selection of sixty-four nodes in the single hidden layer was chosen to intentionally produce baseline performances of slightly better than random results to provide a large improvement space for evaluating change due to AF swapping. The output layer for each consisted of ten Softmax nodes. The baseline accuracy and cost performance for each AF is plotted in Figures 4.2a, 4.3a, 4.4a, 4.5a and 4.6a, below, in increments of five epochs.

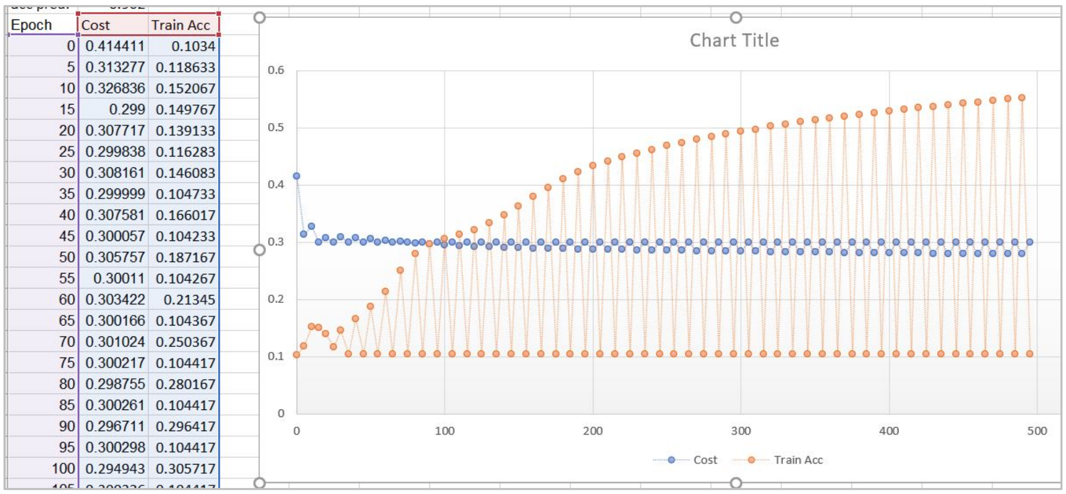
After the baseline performances were established the training runs for evaluating performance with AF swapping were begun. Training run sessions were performed that began with each of the five activation functions and were allowed to proceed with that AF through five epochs. This was to establish each as the initial learning function and to form the basis for randomly swapping to another AF should accuracy fail to improve.

From the baseline performance evaluations, it was clear that the AF producing the greatest accuracy was Leaky ReLU. Because of that uniqueness, the AF swapping runs were separated into two sessions where for two runs Leaky ReLU was available to the random swapping algorithm for selection and for two runs where it was not available (with the exception of the sessions where Leaky ReLU was the initial AF). The performance plots for the various AF swapping runs are shown in Figures 4.2 through 4.6, below. Changes in cost or accuracy, depending on the AF swapping function decision function, may not be apparent because they may have occurred between the epochs when data was captured.

Several instances of curious activity toward the end of the 300 epoch runs, seen in Figures 4.2b, 4.3e, 4.3f and 4.6f, below, suggested the likelihood of convergence issues with the AF swapping algorithm. To test this, a low-performing AF, TanH, was selected for a session of 30,000 epoch training runs. First, baseline performance over 30,000 epochs was established (Figure 4.7a) and then training runs of 30,000 epochs with AF swapping were conducted with and without Leaky ReLU availability. Those results are plotted in Figures 4.7b and 4.7c, respectively, below. In all three runs accuracy and cost are recorded after each 100 epochs.

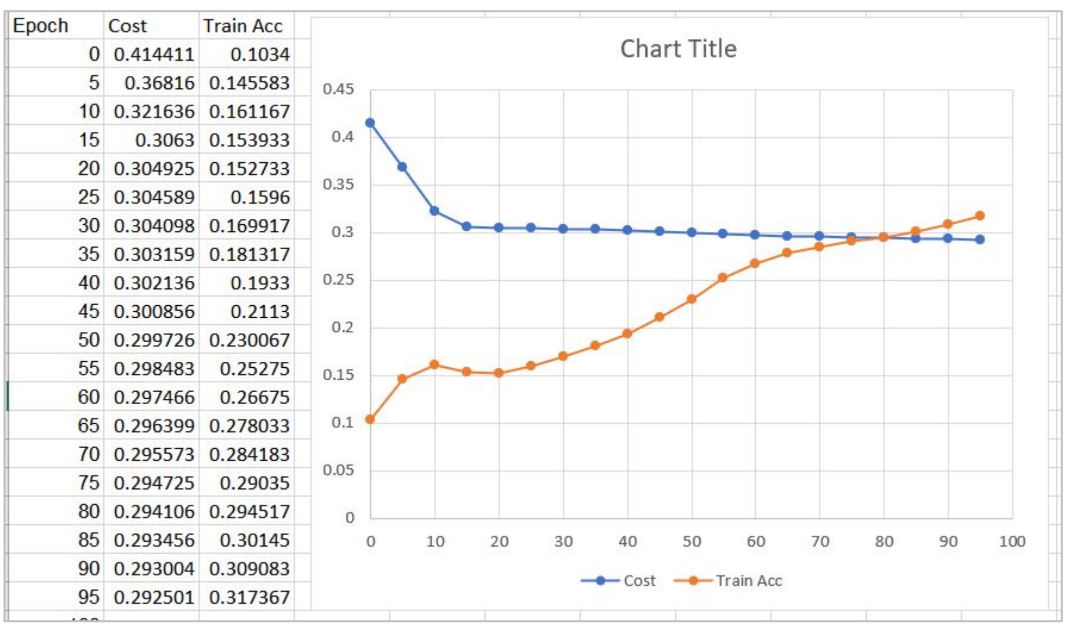
Lastly, a series of tests were performed on the commonly used Iris data set. The same set-up conditions applied; i.e., same computer platform and hyperparameter selections. Baselines were performed for each activation function and, then, sessions where all AFs were available for random swapping were performed with each AF initiating the runs for the first five epochs. The results are plotted in Figure 4.8 with cost and accuracy reported at five epoch intervals.

4.1 Initial Test Results



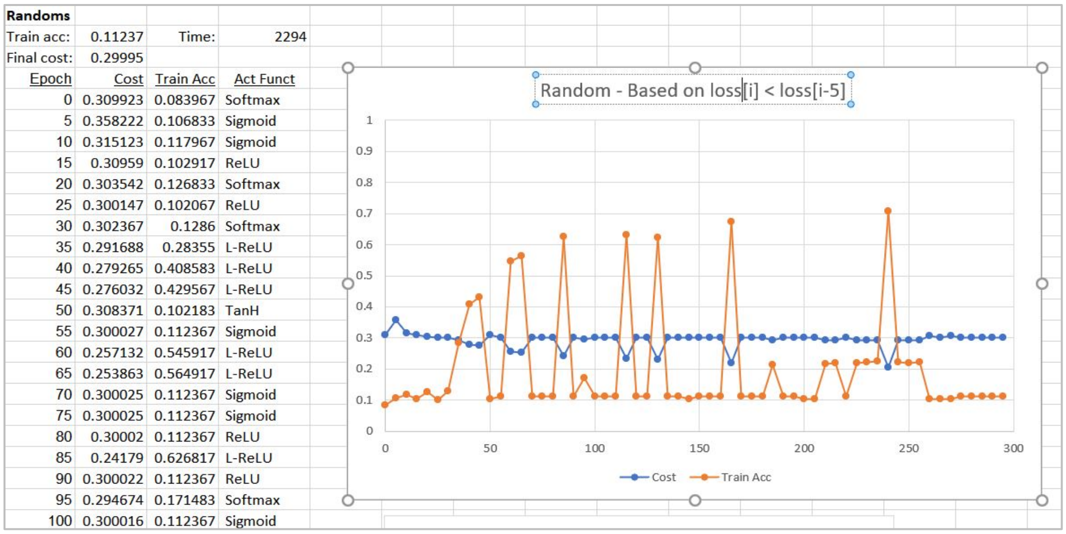
Epoch	Cost	Train Acc
0	0.414411	0.1034
5	0.313277	0.118633
10	0.326836	0.152067
15	0.299	0.149767
20	0.307717	0.139133
25	0.299838	0.116283
30	0.308161	0.146083
35	0.299999	0.104733
40	0.307581	0.166017
45	0.300057	0.104233
50	0.305757	0.187167
55	0.30011	0.104267
60	0.303422	0.21345
65	0.300166	0.104367
70	0.301024	0.250367
75	0.300217	0.104417
80	0.298755	0.280167
85	0.300261	0.104417
90	0.296711	0.296417
95	0.300298	0.104417
100	0.294943	0.305717

4.1a. Cost and accuracy plots with AF swapping by entropy function per epoch



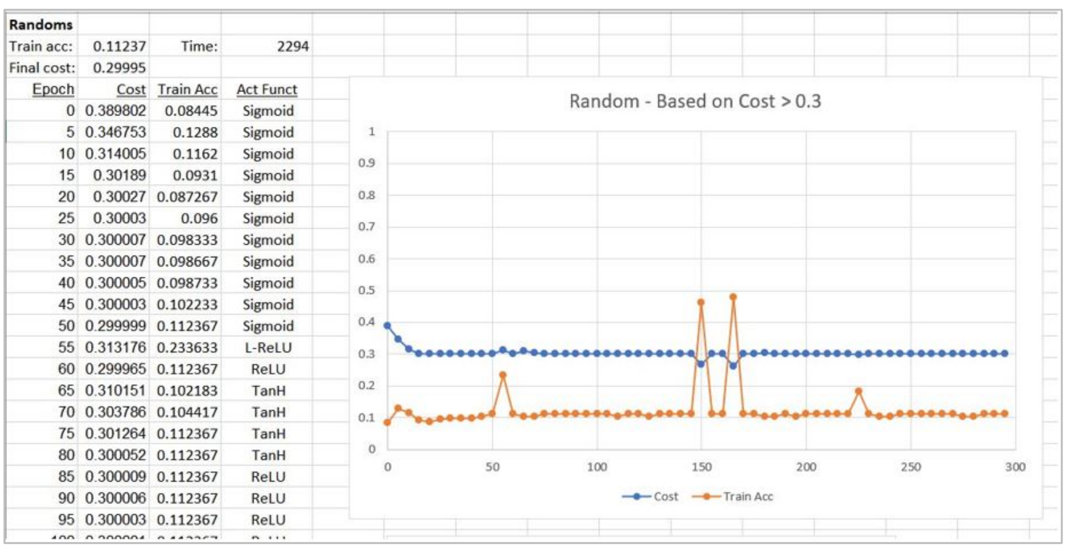
Epoch	Cost	Train Acc
0	0.414411	0.1034
5	0.36816	0.145583
10	0.321636	0.161167
15	0.3063	0.153933
20	0.304925	0.152733
25	0.304589	0.1596
30	0.304098	0.169917
35	0.303159	0.181317
40	0.302136	0.1933
45	0.300856	0.2113
50	0.299726	0.230067
55	0.298483	0.25275
60	0.297466	0.26675
65	0.296399	0.278033
70	0.295573	0.284183
75	0.294725	0.29035
80	0.294106	0.294517
85	0.293456	0.30145
90	0.293004	0.309083
95	0.292501	0.317367

4.1b. AF swap based on improving accuracy with random selection, 100 epochs



Randoms			
Train acc:	0.11237	Time:	2294
Final cost:	0.29995		
Epoch	Cost	Train Acc	Act Funct
0	0.309923	0.083967	Softmax
5	0.358222	0.106833	Sigmoid
10	0.315123	0.117967	Sigmoid
15	0.30959	0.102917	ReLU
20	0.303542	0.126833	Softmax
25	0.300147	0.102067	ReLU
30	0.302367	0.1286	Softmax
35	0.291688	0.28355	L-ReLU
40	0.279265	0.408583	L-ReLU
45	0.276032	0.429567	L-ReLU
50	0.308371	0.102183	TanH
55	0.300027	0.112367	Sigmoid
60	0.257132	0.545917	L-ReLU
65	0.253863	0.564917	L-ReLU
70	0.300025	0.112367	Sigmoid
75	0.300025	0.112367	Sigmoid
80	0.30002	0.112367	ReLU
85	0.24179	0.626817	L-ReLU
90	0.300022	0.112367	ReLU
95	0.294674	0.171483	Softmax
100	0.300016	0.112367	Sigmoid

4.1c. Entropy adjusted to compare cost this epoch vs cost at 5th prior epoch



Randoms			
Train acc:	0.11237	Time:	2294
Final cost:	0.29995		
Epoch	Cost	Train Acc	Act Funct
0	0.389802	0.08445	Sigmoid
5	0.346753	0.1288	Sigmoid
10	0.314005	0.1162	Sigmoid
15	0.30189	0.0931	Sigmoid
20	0.30027	0.087267	Sigmoid
25	0.30003	0.096	Sigmoid
30	0.300007	0.098333	Sigmoid
35	0.300007	0.098667	Sigmoid
40	0.300005	0.098733	Sigmoid
45	0.300003	0.102233	Sigmoid
50	0.299999	0.112367	Sigmoid
55	0.313176	0.233633	L-ReLU
60	0.299965	0.112367	ReLU
65	0.310151	0.102183	TanH
70	0.303786	0.104417	TanH
75	0.301264	0.112367	TanH
80	0.300052	0.112367	TanH
85	0.300009	0.112367	ReLU
90	0.300006	0.112367	ReLU
95	0.300003	0.112367	ReLU

4.1d. Entropy adjusted to swap AF based on cost greater than 0.3

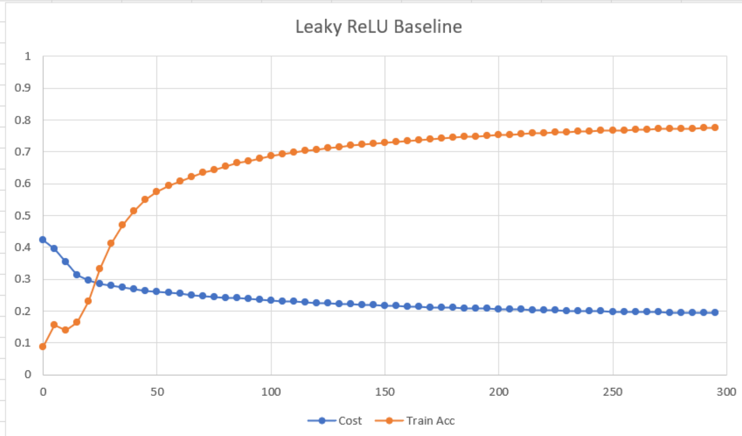
Figure 4.1. Plots from initial test results

4.2 Leaky ReLU, MNIST Data Set, Kolmogorov-Smirnov Tests

The MNIST data set was trained (using 60,000 images as training data) using the Leaky ReLU activation function as the primary or initial AF. Three training sessions of 300 epochs each were run with cost and accuracy results reported at intervals of five epochs. First, a baseline run was conducted to establish cost and accuracy for the Leaky ReLU activation function (Figure 4.2a). Then, two sessions were run that began with Leaky ReLU (Figures 4.2b and 4.2c) but with freedom to randomly switch to either the Leaky ReLU, ReLU, Sigmoid, Softmax or TanH activation function. In this specific case,, with Leaky ReLU being the initial AF, two swapping runs were conducted where all five AFs were available for random replacement. The two additional runs where Leaky ReLU was suppressed were not conducted because Leaky ReLU was the initial AF and suppressing it later in the training risked spurious results; i.e., removing the availability of an initially available and used AF is not realistic. In each case, random switching did not begin until after the initial five epochs to assure initiation of training with the primary activation function. After the initial five epochs, random AF switching was triggered after an epoch where accuracy was less than the previous epoch.

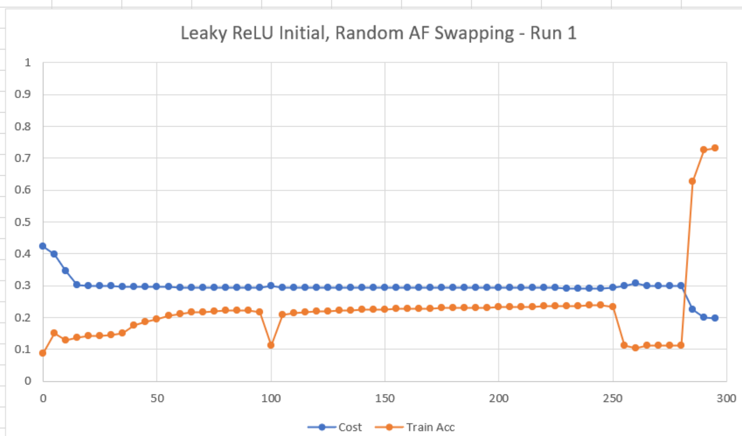
The Kolmogorov-Smirnoff Test (K-S Test), a nonparametric test, compared the accuracy curves of the two random AF swapping runs with the accuracy curve of the baseline run. In each case the accuracy difference was significant (Figure 4.2d) though slightly declining.

Leaky ReLU Baseline			
Train acc:	0.7748	Time:	138
Final cost:	0.1935		
Epoch	Cost	Train Acc	Act Funct
0	0.423105	0.086817	L-ReLU
5	0.395748	0.157433	L-ReLU
10	0.354247	0.138883	L-ReLU
15	0.3133	0.165133	L-ReLU
20	0.295669	0.23015	L-ReLU
25	0.28656	0.3314	L-ReLU
30	0.279657	0.411917	L-ReLU
35	0.273571	0.469067	L-ReLU
40	0.268393	0.5149	L-ReLU
45	0.26416	0.548883	L-ReLU
50	0.260492	0.573417	L-ReLU
55	0.257098	0.5931	L-ReLU
60	0.253895	0.608483	L-ReLU
65	0.250861	0.622417	L-ReLU
70	0.247984	0.6345	L-ReLU
75	0.245253	0.6443	L-ReLU
80	0.242658	0.65495	L-ReLU
85	0.240193	0.6638	L-ReLU
90	0.237848	0.67175	L-ReLU
95	0.235618	0.678967	L-ReLU



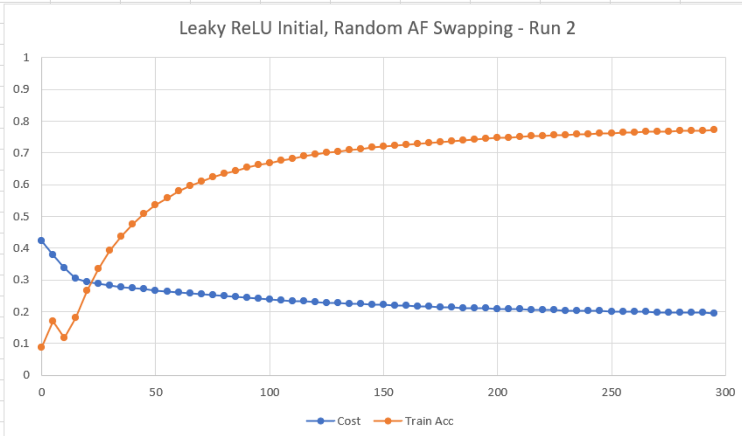
4.2a. Leaky ReLU, baseline, 300 epochs

Leaky ReLU Initial			
Train acc:	0.7299	Time:	138
Final cost:	0.1983		
Epoch	Cost	Train Acc	Act Funct
0	0.423105	0.086817	L-ReLU
5	0.396981	0.1516	ReLU
10	0.346363	0.128217	ReLU
15	0.300922	0.135717	Softmax
20	0.299773	0.141	Softmax
25	0.298809	0.142383	Softmax
30	0.297943	0.143683	Softmax
35	0.297168	0.1513	Softmax
40	0.296491	0.17645	Softmax
45	0.295919	0.186267	Softmax
50	0.295446	0.193883	Softmax
55	0.295048	0.204633	Softmax
60	0.294714	0.211417	Softmax
65	0.294428	0.215933	Softmax
70	0.29419	0.21785	Softmax
75	0.294	0.219817	Softmax
80	0.293845	0.221017	Softmax
85	0.293717	0.221433	Softmax
90	0.293611	0.221383	Softmax
95	0.293738	0.216967	Softmax



4.2b. Leaky ReLU, first random AF swapping run, 300 epochs

Leaky ReLU Initial			
Train acc:	0.7718	Time:	144
Final cost:	0.1956		
Epoch	Cost	Train Acc	Act Funct
0	0.423105	0.086817	L-ReLU
5	0.378845	0.169767	TanH
10	0.337692	0.1178	TanH
15	0.305091	0.179483	L-ReLU
20	0.293226	0.264883	L-ReLU
25	0.287536	0.3359	L-ReLU
30	0.282766	0.392283	L-ReLU
35	0.278509	0.436467	L-ReLU
40	0.274544	0.474967	L-ReLU
45	0.270759	0.507633	L-ReLU
50	0.267123	0.535383	L-ReLU
55	0.263639	0.5589	L-ReLU
60	0.260309	0.578733	L-ReLU
65	0.25713	0.596417	L-ReLU
70	0.254098	0.610017	L-ReLU
75	0.251208	0.62325	L-ReLU
80	0.248455	0.63475	L-ReLU
85	0.245831	0.6441	L-ReLU
90	0.243329	0.653117	L-ReLU
95	0.240945	0.661033	L-ReLU



4.2c. Leaky ReLU, second random AF swapping run, 300 epochs

Leaky ReLU x1 = Baseline x2 = Random AF swapping, run 1	Two-sample Kolmogorov-Smirnov test D = 0.83333, p-value < 2.2e-16 alternative hypothesis: two-sided
Leaky ReLU x1 = Baseline x2 = Random AF swapping, run 2	Two-sample Kolmogorov-Smirnov test D = 0.96667, p-value < 2.2e-16 alternative hypothesis: two-sided

4.2d. K-S Tests, Leaky ReLU baseline ~ first and second random runs

Figure 4.2. Leaky ReLU over 300 epochs with K-S Tests

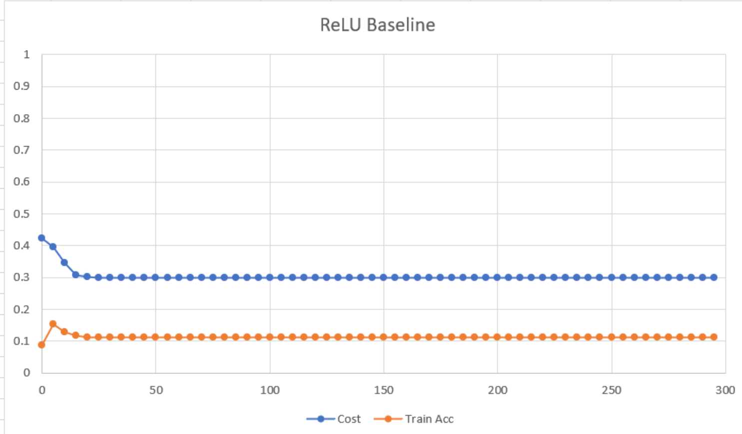
4.3 ReLU, MNIST Data Set, Kolmogorov-Smirnov Tests

The MNIST data set was trained (using 60,000 images as training data) using the ReLU activation function as the initial AF. ReLu was chosen as the initial activation function to establish it as the initial learning function with the freedom to continue until convergence or leveling of accuracy to zero or below. This allowed a training minimum performance based on ReLU to be established for the anticipated AF swapping and measuring performance due to changing activation functions.   Five training sessions of 300 epochs each were run with cost and accuracy results reported at intervals of five epochs.

First, a baseline run was conducted to establish cost and accuracy for the ReLU activation function (Figure 4.3a). Next, two sessions were run that began with ReLU (Figures 4.3b and 4.3c) but with freedom to randomly switch to either the Leaky ReLU, ReLU, Sigmoid, Softmax or TanH activation function. Then, two training sessions were run (Figures 4.3e and 4.3f) that began with ReLU and allowed random switching among ReLU, Sigmoid, Softmax or TanH. In this test the Leaky ReLU activation function was excluded because its substantially better performance on the MNIST data set relative to the other four AFs, as shown by its baseline performance, represented something of a performance outlier and would, if selected by the random AF algorithm, perform to a degree that prevented further swapping over the remaining epochs.…. To assure initiation of training with the primary activation function, random switching did not begin until after the initial five epochs. Then, random AF switching was triggered after an epoch where accuracy was less than the previous epoch.

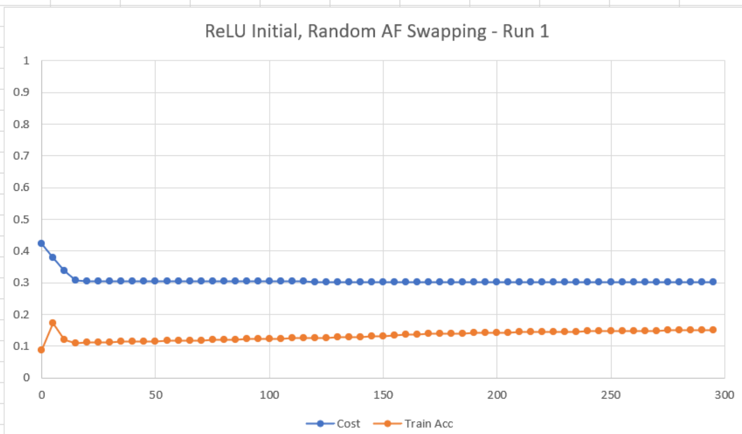
The K-S Test was used to compare the accuracy curves of the four random AF swapping runs with the accuracy curve of the baseline run. In each case the accuracy difference significantly increased (Figures 4.3d and 4.3g) in each case.

ReLU Baseline			
Train acc:	0.1124	Time:	122
Final cost:	0.2999		
Epoch	Cost	Train Acc	Act Funct
0	0.423081	0.08705	ReLU
5	0.396533	0.153617	ReLU
10	0.346294	0.128483	ReLU
15	0.306873	0.117617	ReLU
20	0.300918	0.11215	ReLU
25	0.300118	0.111983	ReLU
30	0.300001	0.112483	ReLU
35	0.299978	0.112383	ReLU
40	0.299971	0.112383	ReLU
45	0.299972	0.112383	ReLU
50	0.299971	0.112383	ReLU
55	0.299971	0.112367	ReLU
60	0.29997	0.112367	ReLU
65	0.299968	0.112367	ReLU
70	0.299968	0.112367	ReLU
75	0.299967	0.112367	ReLU
80	0.299966	0.112367	ReLU
85	0.299965	0.112367	ReLU
90	0.299964	0.112367	ReLU
95	0.299964	0.112367	ReLU



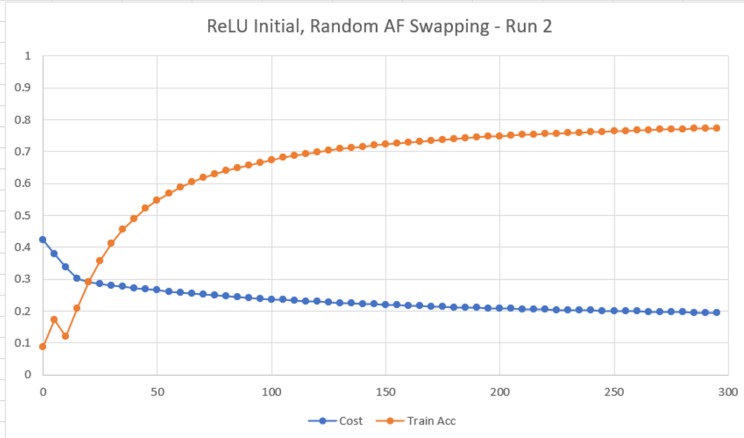
4.3a. ReLU, baseline, 300 epochs

ReLU Initial			
Train acc:	0.1124	Time:	196
Final cost:	0.2999		
Epoch	Cost	Train Acc	Act Funct
0	0.423081	0.08705	ReLU
5	0.378508	0.171333	TanH
10	0.337886	0.118917	TanH
15	0.306261	0.1098	Softmax
20	0.306005	0.110783	Softmax
25	0.305771	0.11175	Softmax
30	0.305555	0.112517	Softmax
35	0.305356	0.113417	Softmax
40	0.30517	0.1138	Softmax
45	0.304996	0.114617	Softmax
50	0.304832	0.115683	Softmax
55	0.304678	0.116583	Softmax
60	0.304533	0.117483	Softmax
65	0.304395	0.118183	Softmax
70	0.304263	0.1188	Softmax
75	0.304138	0.119467	Softmax
80	0.304019	0.120517	Softmax
85	0.303904	0.1213	Softmax
90	0.303795	0.122083	Softmax
95	0.303689	0.12285	Softmax



4.3b. ReLU initial, first run with random AF swapping, 300 epochs

ReLU Initial			
Train acc:	0.7729	Time:	129
Final cost:	0.1945		
Epoch	Cost	Train Acc	Act Funct
0	0.423081	0.08705	ReLU
5	0.378508	0.171333	TanH
10	0.337886	0.118917	TanH
15	0.300777	0.208767	L-ReLU
20	0.291179	0.29	L-ReLU
25	0.2857	0.357817	L-ReLU
30	0.280936	0.411933	L-ReLU
35	0.276673	0.45475	L-ReLU
40	0.27271	0.490117	L-ReLU
45	0.268935	0.521917	L-ReLU
50	0.265318	0.547883	L-ReLU
55	0.261858	0.569333	L-ReLU
60	0.258557	0.588417	L-ReLU
65	0.255409	0.6038	L-ReLU
70	0.252411	0.617683	L-ReLU
75	0.249556	0.629783	L-ReLU
80	0.246837	0.640233	L-ReLU
85	0.244248	0.649633	L-ReLU
90	0.241782	0.658133	L-ReLU
95	0.239433	0.666183	L-ReLU

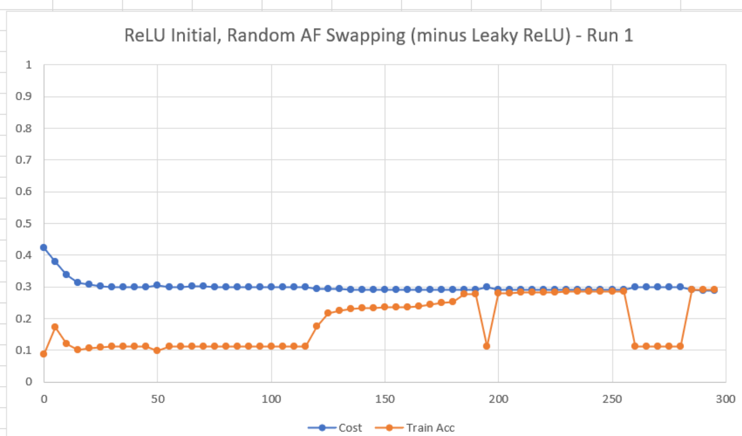


4.3c. ReLU initial, second run with random AF swapping, 300 epochs

ReLU x1 = Baseline x2 = Random AF swapping, run 1	Two-sample Kolmogorov-Smirnov test D = 0.88333, p-value < 2.2e-16 alternative hypothesis: two-sided
ReLU x1 = Baseline x2 = Random AF swapping, run 2	Two-sample Kolmogorov-Smirnov test D = 0.96667, p-value < 2.2e-16 alternative hypothesis: two-sided

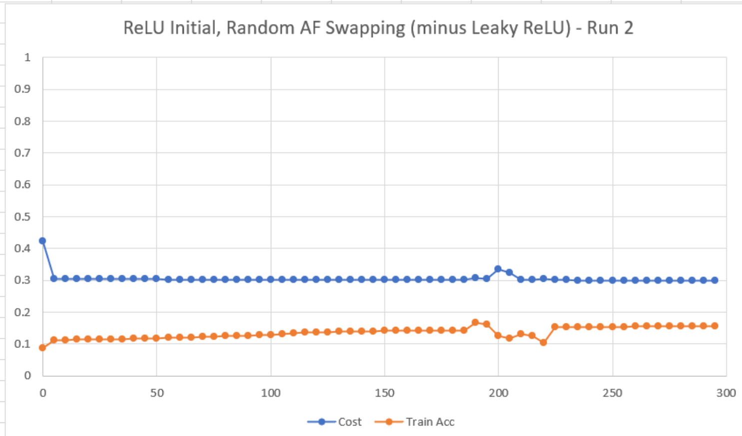
4.3d. K-S Tests, ReLU baseline ~ first and second random runs

ReLU Initial - Best performing Leaky ReLU AF not available			
Train acc:	0.2906	Time:	168
Final cost:	0.2894		
Epoch	Cost	Train Acc	Act Funct
0	0.423081	0.08705	ReLU
5	0.378508	0.171333	TanH
10	0.337886	0.118917	TanH
15	0.314252	0.10215	ReLU
20	0.306164	0.106333	ReLU
25	0.300732	0.10965	ReLU
30	0.300085	0.11145	ReLU
35	0.299978	0.112267	ReLU
40	0.299968	0.112383	ReLU
45	0.299968	0.112367	ReLU
50	0.304792	0.097533	TanH
55	0.299977	0.112367	Sigmoid
60	0.299975	0.112367	Sigmoid
65	0.302549	0.112367	TanH
70	0.301406	0.112367	TanH
75	0.299985	0.112367	ReLU
80	0.299985	0.112367	ReLU
85	0.299984	0.112367	ReLU
90	0.300023	0.112367	TanH
95	0.299981	0.112367	Sigmoid



4.3e. ReLU initial, first run with AF swapping, L-ReLU suppressed, 300 epochs

ReLU Initial - Best performing Leaky ReLU AF not available			
Train acc:	0.1571	Time:	174
Final cost:	0.2999		
Epoch	Cost	Train Acc	Act Funct
0	0.423081	0.08705	ReLU
5	0.304662	0.111683	Softmax
10	0.304473	0.11265	Softmax
15	0.304296	0.113633	Softmax
20	0.304131	0.114167	Softmax
25	0.303974	0.114733	Softmax
30	0.303826	0.1154	Softmax
35	0.303686	0.116133	Softmax
40	0.303553	0.117267	Softmax
45	0.303426	0.117883	Softmax
50	0.303305	0.118867	Softmax
55	0.30319	0.119767	Softmax
60	0.303079	0.120483	Softmax
65	0.302972	0.12135	Softmax
70	0.30287	0.122417	Softmax
75	0.302772	0.123433	Softmax
80	0.302677	0.124767	Softmax
85	0.302586	0.125767	Softmax
90	0.302498	0.126983	Softmax
95	0.302413	0.12855	Softmax



4.3f. ReLU initial, second run with AF swapping, L-ReLU suppressed, 300 epochs

ReLU x1 = Baseline x2 = Random AF swapping, minus L-ReLU, run 1	Two-sample Kolmogorov-Smirnov test D = 0.51667, p-value = 2.214e-07 alternative hypothesis: two-sided
ReLU x1 = Baseline x2 = Random AF swapping, minus L-ReLU, run 2	Two-sample Kolmogorov-Smirnov test D = 0.9, p-value < 2.2e-16 alternative hypothesis: two-sided

4.3g. K-S Tests, ReLU baseline ~ first and second random runs, L-ReLU suppressed

Figure 4.3. ReLU over 300 epochs with K-S Tests

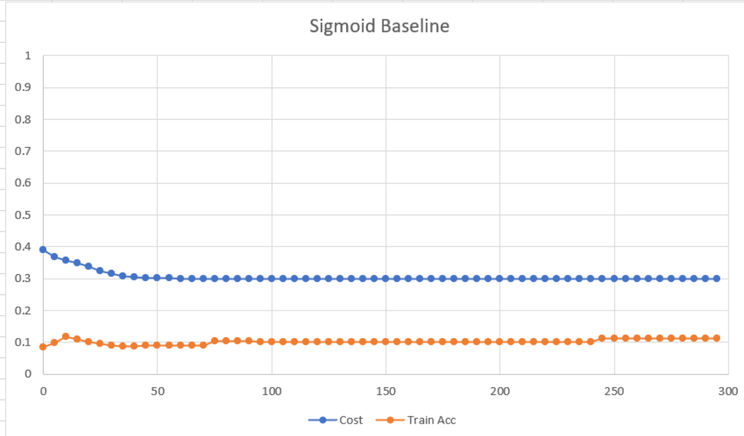
4.4 Sigmoid, MNIST Data Set Plots, Kolmogorov-Smirnov Tests

The MNIST data set was trained (using 60,000 images as training data) using the Sigmoid activation function as the initial AF. Sigmoid was chosen as the initial activation function to establish it as the initial learning function with the freedom to continue until convergence or leveling of accuracy to zero or below. This allowed a training minimum performance based on the Sigmoid AF to be established for the anticipated random AF swapping and measuring performance due to changing activation functions. Five training sessions of 300 epochs each were run with cost and accuracy results reported at intervals of five epochs.

First, a baseline run was conducted to establish cost and accuracy for the Sigmoid activation function (Figure 4.4a). Next, two sessions were run that began with the Sigmoid activation function (Figures 4.4b and 4.4c) but with freedom to randomly switch to either the Leaky ReLU, ReLU, Sigmoid, Softmax or TanH activation function. Then, two training sessions (Figures 4.4e and 4.4f) were run that began with Sigmoid and allowed random switching among ReLU, Sigmoid, Softmax or TanH with Leaky ReLU, the best performing AF, excluded. The Leaky ReLU activation function was excluded because its substantially better performance on the MNIST data set relative to the other four AFs, as shown by its baseline performance, represented something of a performance outlier and would, if selected by the random AF algorithm, perform to a degree that prevented further swapping over the remaining epochs. To assure initiation of training with the primary activation function, random switching did not begin until after the initial five epochs. Then, random AF switching was triggered after an epoch where accuracy was less than the previous epoch.

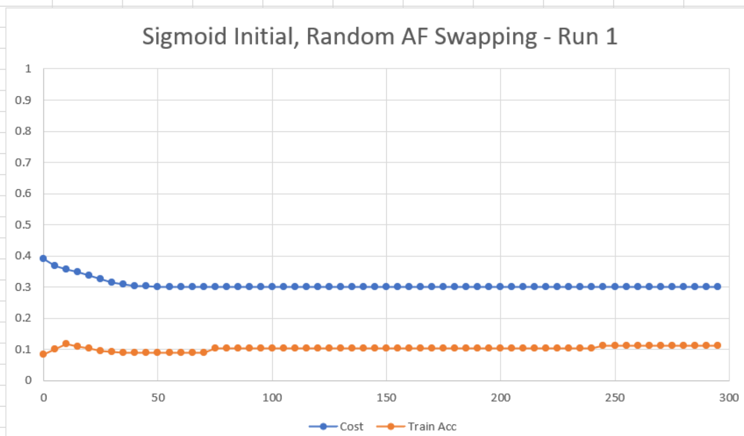
The K-S Test was used to compare the accuracy curves of the four random swapping runs with the accuracy curve of the baseline run. One run (Figure 4.4b) resulted in an exact copy of the baseline session with a p-value of one and a D-statistic of zero. In every other case the accuracy difference increased significantly (Figures 4.4c, 4.4e and 4.4f).

Sigmoid Baseline			
Train acc:	0.1124	Time:	200
Final cost:	0.2999		
Epoch	Cost	Train Acc	Act Funct
0	0.389802	0.08445	Sigmoid
5	0.367211	0.099433	Sigmoid
10	0.35641	0.11695	Sigmoid
15	0.347985	0.110033	Sigmoid
20	0.336465	0.102067	Sigmoid
25	0.324407	0.094283	Sigmoid
30	0.314465	0.091233	Sigmoid
35	0.307775	0.088333	Sigmoid
40	0.304123	0.0882	Sigmoid
45	0.302183	0.089383	Sigmoid
50	0.301177	0.08915	Sigmoid
55	0.300687	0.089167	Sigmoid
60	0.300448	0.089217	Sigmoid
65	0.300327	0.089567	Sigmoid
70	0.30026	0.0894	Sigmoid
75	0.300218	0.104367	Sigmoid
80	0.300188	0.103083	Sigmoid
85	0.300165	0.102667	Sigmoid
90	0.300145	0.1025	Sigmoid
95	0.300128	0.102367	Sigmoid



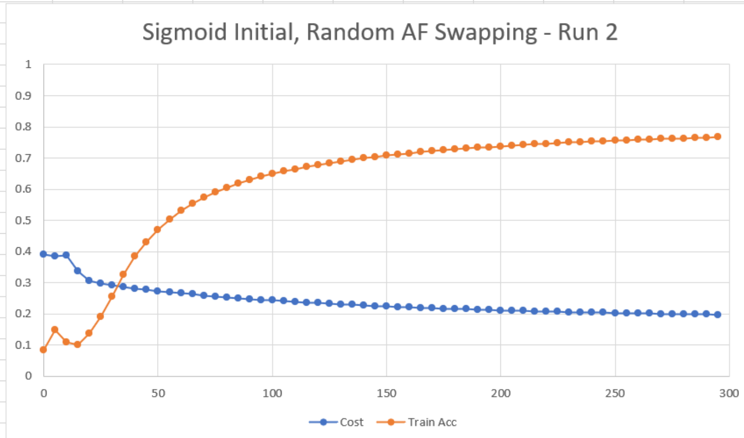
4.4a. Sigmoid, baseline, 300 epochs

Sigmoid Initial			
Train acc:	0.1124	Time:	200
Final cost:	0.2999		
Epoch	Cost	Train Acc	Act Funct
0	0.389802	0.08445	Sigmoid
5	0.367211	0.099433	Sigmoid
10	0.35641	0.11695	Sigmoid
15	0.347985	0.110033	Sigmoid
20	0.336465	0.102067	Sigmoid
25	0.324407	0.094283	Sigmoid
30	0.314465	0.091233	Sigmoid
35	0.307775	0.088333	Sigmoid
40	0.304123	0.0882	Sigmoid
45	0.302183	0.089383	Sigmoid
50	0.301177	0.08915	Sigmoid
55	0.300687	0.089167	Sigmoid
60	0.300448	0.089217	Sigmoid
65	0.300327	0.089567	Sigmoid
70	0.30026	0.0894	Sigmoid
75	0.300218	0.104367	Sigmoid
80	0.300188	0.103083	Sigmoid
85	0.300165	0.102667	Sigmoid
90	0.300145	0.1025	Sigmoid
95	0.300128	0.102367	Sigmoid



4.4b. Sigmoid initial, first run with random AF swapping, 300 epochs

Sigmoid Initial			
Train acc:	0.7667	Time:	131
Final cost:	0.1971		
Epoch	Cost	Train Acc	Act Funct
0	0.389802	0.08445	Sigmoid
5	0.383694	0.148033	TanH
10	0.38828	0.109983	TanH
15	0.335597	0.10155	L-ReLU
20	0.307109	0.138133	L-ReLU
25	0.297508	0.190483	L-ReLU
30	0.291094	0.256283	L-ReLU
35	0.285742	0.32605	L-ReLU
40	0.281128	0.383667	L-ReLU
45	0.276997	0.4285	L-ReLU
50	0.27315	0.468167	L-ReLU
55	0.269482	0.503733	L-ReLU
60	0.265971	0.531617	L-ReLU
65	0.262611	0.554633	L-ReLU
70	0.259401	0.573467	L-ReLU
75	0.256336	0.590883	L-ReLU
80	0.25341	0.6048	L-ReLU
85	0.250617	0.618167	L-ReLU
90	0.247953	0.629517	L-ReLU
95	0.24541	0.6398	L-ReLU

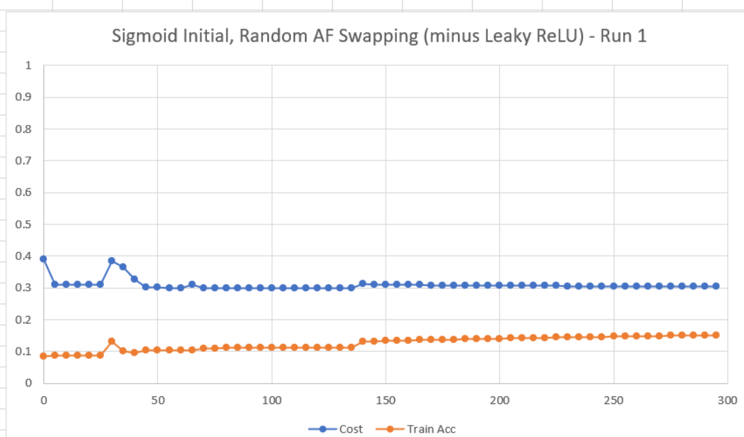


4.4c. Sigmoid initial, second run with random AF swapping, 300 epochs

Sigmoid x1 = Baseline x2 = Random AF swapping, run 1	Two-sample Kolmogorov-Smirnov test D = 0, p-value = 1 alternative hypothesis: two-sided
Sigmoid x1 = Baseline x2= Random AF swapping, run 2	Two-sample Kolmogorov-Smirnov test D = 0.95, p-value < 2.2e-16 alternative hypothesis: two-sided

4.4d. K-S Tests, Sigmoid baseline ~ first and second random runs

Sigmoid Initial - Best performing Leaky ReLU AF not available			
Train acc:	0.1512	Time:	180
Final cost:	0.3045		
Epoch	Cost	Train Acc	Act Funct
0	0.389802	0.08445	Sigmoid
5	0.310127	0.086683	Softmax
10	0.30981	0.08675	Softmax
15	0.309524	0.08675	Softmax
20	0.309629	0.087317	Softmax
25	0.309381	0.087167	Softmax
30	0.385678	0.130567	TanH
35	0.364062	0.1002	TanH
40	0.326329	0.096	ReLU
45	0.301718	0.102683	Sigmoid
50	0.300843	0.103783	Sigmoid
55	0.300423	0.10415	Sigmoid
60	0.300221	0.10345	Sigmoid
65	0.309405	0.10295	TanH
70	0.300013	0.10945	Sigmoid
75	0.299999	0.110517	Sigmoid
80	0.299988	0.1111	Sigmoid
85	0.29998	0.111417	Sigmoid
90	0.299974	0.111717	Sigmoid
95	0.29997	0.111917	Sigmoid



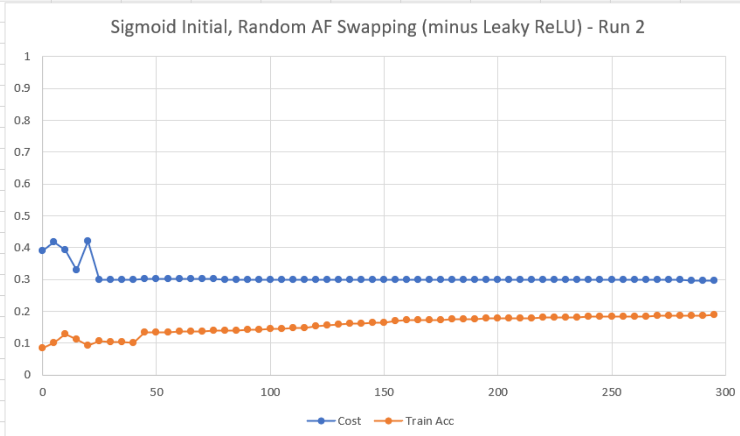
4.4e. Sigmoid initial, first run with AF swapping, L-ReLU suppressed, 300 epochs

Sigmoid Initial - Best performing Leaky ReLU AF not available

Train acc: 0.1877 Time: 175

Final cost: 0.2977

Epoch	Cost	Train Acc	Act Funct
0	0.389802	0.08445	Sigmoid
5	0.417693	0.100833	ReLU
10	0.393785	0.127233	ReLU
15	0.329179	0.11215	ReLU
20	0.420862	0.0922	TanH
25	0.300265	0.107633	Sigmoid
30	0.300127	0.1036	Sigmoid
35	0.300051	0.10295	Sigmoid
40	0.300137	0.1022	ReLU
45	0.301831	0.132967	Softmax
50	0.301593	0.133867	Softmax
55	0.301377	0.134933	Softmax
60	0.301179	0.135933	Softmax
65	0.300996	0.1372	Softmax
70	0.300827	0.138117	Softmax
75	0.30067	0.138933	Softmax
80	0.300523	0.139933	Softmax
85	0.300386	0.140867	Softmax
90	0.300257	0.14185	Softmax
95	0.300135	0.14285	Softmax



4.4f. Sigmoid initial, second run with AF swapping, L-ReLU suppressed, 300 epochs

Sigmoid x1 = Baseline x2 = Random AF swapping, minus L-ReLU, run 1	Two-sample Kolmogorov-Smirnov test D = 0.61667, p-value = 2.465e-10 alternative hypothesis: two-sided
Sigmoid x1 = Baseline x2 = Random AF swapping, minus L-ReLU, run 2	Two-sample Kolmogorov-Smirnov test D = 0.86667, p-value < 2.2e-16 alternative hypothesis: two-sided

4.4g. K-S Tests, Sigmoid baseline ~ first and second random runs, L-ReLU suppressed

Figure 4.4. Sigmoid over 300 epochs with K-S Tests

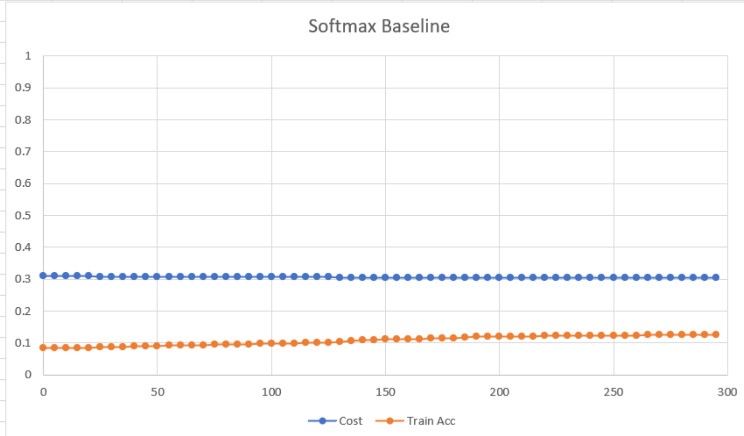
4.5 Softmax, MNIST Data Set Plots, Kolmogorov-Smirnov Tests

The MNIST data set was trained (using 60,000 images as training data) using the Softmax activation function as the primary or initial AF. The Softmax AF was chosen as the initial activation function to establish it as the initial learning function with the freedom to continue until convergence or leveling of accuracy to zero or below. This allowed a training minimum performance based on Softmax to be established for the anticipated AF swapping and measuring performance due to changing activation functions. Five training sessions of 300 epochs each were run with cost and accuracy results reported at intervals of five epochs.

First, a baseline run was conducted to establish cost and accuracy for the Softmax activation function (Figure 4.5a). Next, two sessions were run that began with ReLU (Figures 4.5b and 4.5c) but with freedom to randomly switch to another activation function from among Leaky ReLU, ReLU, Sigmoid, Softmax or TanH activation functions. Then, two training sessions were run (Figures 4.5e and 4.5f) that began with Softmax and allowed random switching among ReLU, Sigmoid, Softmax or TanH with Leaky ReLU, the best performing AF, excluded. In this test the Leaky ReLU activation function was excluded because its substantially better performance on the MNIST data set relative to the other four AFs, as shown by its baseline performance, represented something of a performance outlier and would, if selected by the random AF algorithm, perform to a degree that prevented further swapping over the remaining epochs. To assure initiation of training with the primary activation function, random switching did not begin until after the initial five epochs. Then, random AF switching was triggered after an epoch where accuracy was less than the previous epoch.

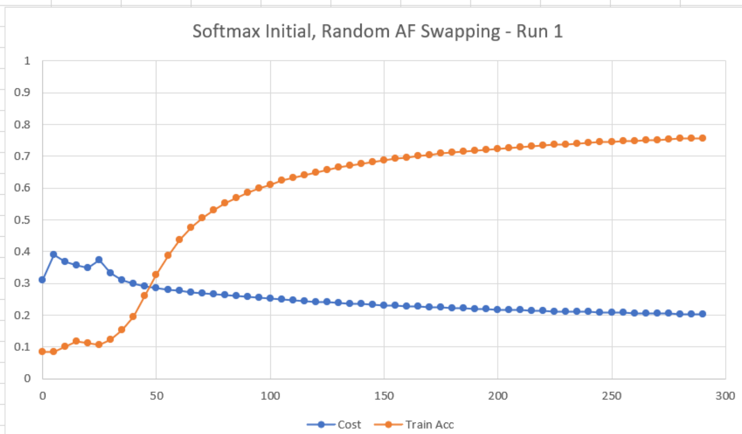
The K-S Test was used to compare the accuracy curves of the four random AF swapping runs with the accuracy curve of the baseline run. In each case the accuracy difference was significantly increased (Figures 4.5d and 4.5g).

Softmax Baseline			
Train acc:	0.1255	Time:	174
Final cost:	0.3041		
Epoch	Cost	Train Acc	Act Funct
0	0.309923	0.083967	Softmax
5	0.309614	0.084267	Softmax
10	0.309336	0.084433	Softmax
15	0.309083	0.08515	Softmax
20	0.308851	0.0855	Softmax
25	0.308637	0.086417	Softmax
30	0.308438	0.0873	Softmax
35	0.308252	0.08835	Softmax
40	0.308078	0.0897	Softmax
45	0.307913	0.090433	Softmax
50	0.307758	0.091067	Softmax
55	0.307611	0.091583	Softmax
60	0.30747	0.092467	Softmax
65	0.307337	0.093517	Softmax
70	0.307209	0.094083	Softmax
75	0.307087	0.095017	Softmax
80	0.306969	0.095167	Softmax
85	0.306857	0.095817	Softmax
90	0.306748	0.096683	Softmax
95	0.306644	0.0973	Softmax



4.5a. Softmax, baseline, 300 epochs

Softmax Initial			
Train acc:	0.7573	Time:	137
Final cost:	0.2023		
Epoch	Cost	Train Acc	Act Funct
0	0.309923	0.083967	Softmax
5	0.38914	0.084133	Sigmoid
10	0.3672	0.100117	Sigmoid
15	0.356499	0.11685	Sigmoid
20	0.348261	0.110867	Sigmoid
25	0.374254	0.1056	L-ReLU
30	0.331879	0.1242	L-ReLU
35	0.309584	0.151983	L-ReLU
40	0.298799	0.195533	L-ReLU
45	0.291604	0.259533	L-ReLU
50	0.285741	0.3254	L-ReLU
55	0.280632	0.386883	L-ReLU
60	0.276236	0.435383	L-ReLU
65	0.272468	0.474117	L-ReLU
70	0.269072	0.5053	L-ReLU
75	0.265875	0.530733	L-ReLU
80	0.262823	0.55185	L-ReLU
85	0.2599	0.569633	L-ReLU
90	0.257101	0.58545	L-ReLU
95	0.254419	0.599617	L-ReLU

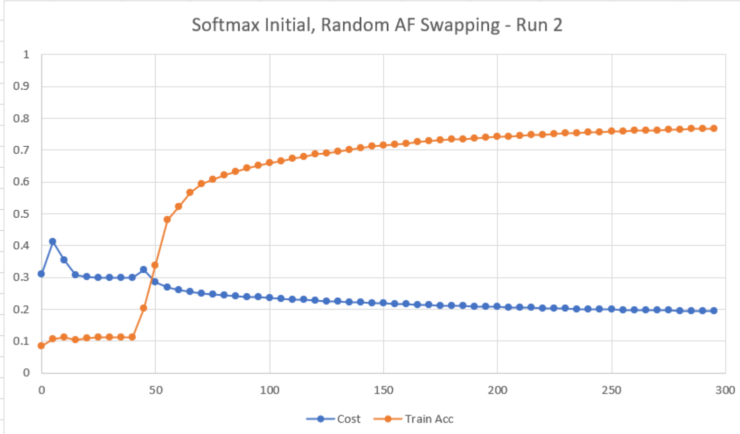


4.5b. Softmax, initial, first run with random AF swapping, 300 epochs

Softmax Initial

Train acc:	0.768	Time:	245
Final cost:	0.1942		

Epoch	Cost	Train Acc	Act Funct
0	0.309923	0.083967	Softmax
5	0.411599	0.105817	ReLU
10	0.353546	0.1119	ReLU
15	0.308239	0.104033	ReLU
20	0.301027	0.110183	ReLU
25	0.300113	0.11155	ReLU
30	0.299964	0.112267	ReLU
35	0.299957	0.112383	ReLU
40	0.299961	0.112367	ReLU
45	0.324708	0.204167	L-ReLU
50	0.285035	0.33685	L-ReLU
55	0.269178	0.4797	L-ReLU
60	0.260913	0.52125	L-ReLU
65	0.253859	0.566717	L-ReLU
70	0.25051	0.594283	L-ReLU
75	0.247696	0.608667	L-ReLU
80	0.24498	0.6212	L-ReLU
85	0.242397	0.632183	L-ReLU
90	0.239942	0.64195	L-ReLU
95	0.23761	0.651033	L-ReLU



4.5c. Softmax initial, second run with random AF swapping, 300 epochs

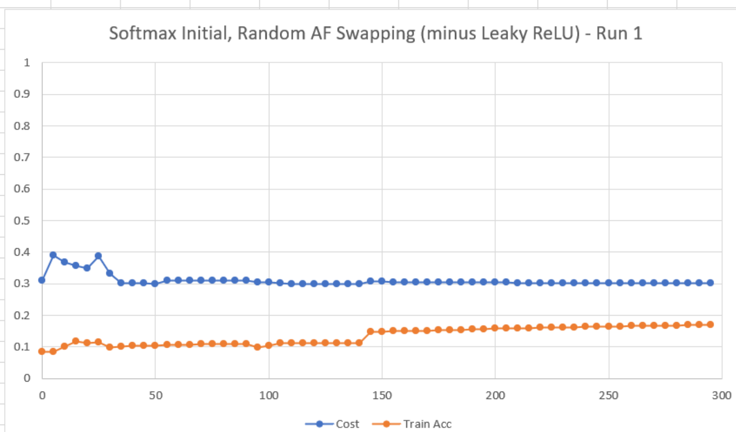
Softmax x1 = Baseline x2 = Random AF swapping, run 1	Two-sample Kolmogorov-Smirnov test D = 0.88136, p-value < 2.2e-16 alternative hypothesis: two-sided
Softmax x1 = Baseline x2 = Random AF swapping, run 2	Two-sample Kolmogorov-Smirnov test D = 0.85, p-value < 2.2e-16 alternative hypothesis: two-sided

4.5d. K-S Tests, Softmax baseline ~ first and second random runs, L-ReLU suppressed

Softmax Initial - Best performing Leaky ReLU AF not available

Train acc:	0.1696	Time:	177
Final cost:	0.3011		

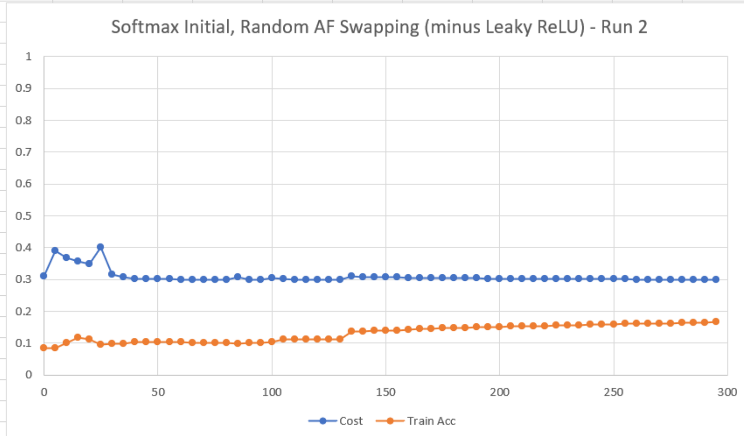
Epoch	Cost	Train Acc	Act Funct
0	0.309923	0.083967	Softmax
5	0.38914	0.084133	Sigmoid
10	0.3672	0.100117	Sigmoid
15	0.356499	0.11685	Sigmoid
20	0.348261	0.110867	Sigmoid
25	0.388266	0.1152	TanH
30	0.33247	0.097417	TanH
35	0.302687	0.101517	Sigmoid
40	0.301301	0.102517	Sigmoid
45	0.300639	0.1036	Sigmoid
50	0.300335	0.103017	Sigmoid
55	0.311375	0.107017	Softmax
60	0.310931	0.107317	Softmax
65	0.310542	0.107717	Softmax
70	0.310196	0.108317	Softmax
75	0.309887	0.109133	Softmax
80	0.309608	0.10955	Softmax
85	0.309354	0.109783	Softmax
90	0.309121	0.10975	Softmax
95	0.305853	0.098567	TanH



4.5e. Softmax initial, first run with AF swapping, L-ReLU suppressed, 300 epochs

Softmax Initial - Best performing Leaky ReLU AF not available

Train acc:	0.1661	Time:	176
Final cost:	0.2995		
Epoch	Cost	Train Acc	Act Funct
0	0.309923	0.083967	Softmax
5	0.38914	0.084133	Sigmoid
10	0.3672	0.100117	Sigmoid
15	0.356499	0.11685	Sigmoid
20	0.348261	0.110867	Sigmoid
25	0.400699	0.0961	ReLU
30	0.314382	0.099017	Sigmoid
35	0.307822	0.09775	Sigmoid
40	0.303059	0.103217	Sigmoid
45	0.301575	0.103533	Sigmoid
50	0.300817	0.103417	Sigmoid
55	0.30108	0.102683	ReLU
60	0.30016	0.102683	Sigmoid
65	0.300224	0.102283	ReLU
70	0.300048	0.102317	Sigmoid
75	0.300036	0.102283	Sigmoid
80	0.300019	0.102183	ReLU
85	0.306231	0.09865	TanH
90	0.299997	0.102183	ReLU
95	0.299994	0.102183	ReLU



4.5f. Softmax initial, second run with AF swapping, L-ReLU suppressed, 300 epochs

Softmax x1 = Baseline x2 = Random AF swapping, minus L-ReLU, run 1	Two-sample Kolmogorov-Smirnov test D = 0.51667, p-value = 2.214e-07 alternative hypothesis: two-sided
Softmax x1 = Baseline x2 = Random AF swapping, minus L-ReLU, run 2	Two-sample Kolmogorov-Smirnov test D = 0.55, p-value = 2.622e-08 alternative hypothesis: two-sided

4.5g. K-S Tests, Softmax baseline ~ first and second random runs, L-ReLU suppressed

Figure 4.5. Softmax over 300 epochs with K-S Tests

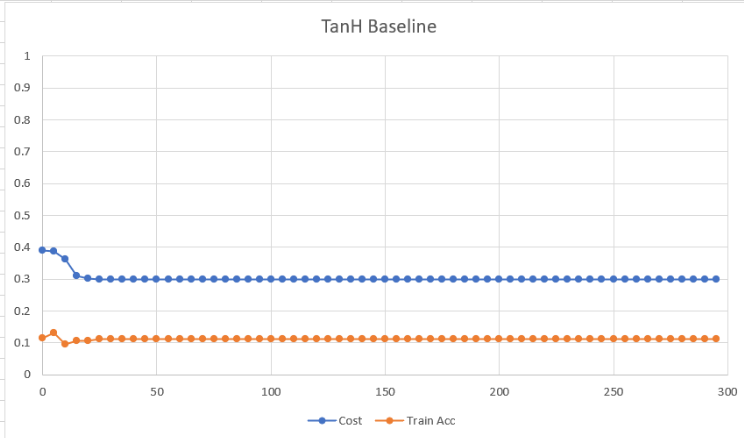
4.6 TanH, MNIST Data Set Plots, Kolmogorov-Smirnov Tests

The MNIST data set was trained (using 60,000 images as training data) using the TanH activation function as the primary or initial AF. TanH was chosen as the initial activation function to establish it as the initial learning function with the freedom to continue until convergence or leveling of accuracy to zero or below. This allowed a training minimum performance based on TanH to be established for the anticipated AF swapping and measuring performance due to changing activation functions. Five training sessions of 300 epochs each were run with cost and accuracy results reported at intervals of five epochs.

First, a baseline run was conducted to establish cost and accuracy for the TanH activation function (Figure 4.6a). Next, two sessions were run that began with TanH (Figures 4.6b and 4.6c) but with freedom to randomly switch to another activation function from Leaky ReLU, ReLU, Sigmoid, Softmax or TanH. Then, two training sessions were run (Figures 4.6e and 4.6f) that began with TanH and allowed random switching among ReLU, Sigmoid, Softmax or TanH with Leaky ReLU, the best performing AF, excluded. In this test the Leaky ReLU activation function was excluded because its substantially better performance on the MNIST data set relative to the other four AFs, as shown by its baseline performance, represented something of a performance outlier and would, if selected by the random AF algorithm, perform to a degree that prevented further swapping over the remaining epochs. To assure initiation of training with the primary activation function, random switching did not begin until after the initial five epochs. Then, random AF switching was triggered after an epoch where accuracy was less than the previous epoch.

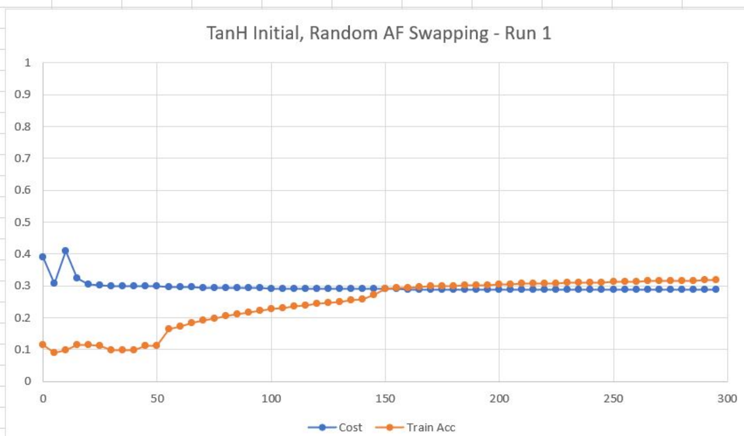
The K-S Test was used to compare the accuracy curves of the four random AF swapping runs with the accuracy curve of the baseline run. In each case the accuracy difference was significant (Figures 4.6d and 4.6g). In three of the four AF swapping runs the accuracy increased significantly. In the fourth, accuracy decreased slightly and to the same degree of decline seen in the baseline run.

TanH Baseline			
Train acc:	0.1124	Time:	214
Final cost:	0.2999		
Epoch	Cost	Train Acc	Act Funct
0	0.390736	0.113733	TanH
5	0.387043	0.130667	TanH
10	0.361815	0.096067	TanH
15	0.310578	0.106267	TanH
20	0.301881	0.107783	TanH
25	0.300134	0.110717	TanH
30	0.29998	0.111883	TanH
35	0.299957	0.11235	TanH
40	0.299953	0.112367	TanH
45	0.299952	0.112367	TanH
50	0.299952	0.112367	TanH
55	0.299952	0.112367	TanH
60	0.299952	0.112367	TanH
65	0.299952	0.112367	TanH
70	0.299952	0.112367	TanH
75	0.299952	0.112367	TanH
80	0.299952	0.112367	TanH
85	0.299952	0.112367	TanH
90	0.299952	0.112367	TanH
95	0.299952	0.112367	TanH



4.6a. TanH, baseline, 300 epochs

TanH Initial			
Train acc:	0.3182	Time:	266
Final cost:	0.2869		
Iteration	Cost	Train Acc	Act Funct
0	0.390736	0.113733	TanH
5	0.308284	0.08975	Softmax
10	0.409017	0.0971	TanH
15	0.324257	0.1154	TanH
20	0.304702	0.114133	TanH
25	0.302657	0.110983	TanH
30	0.300053	0.099167	ReLU
35	0.300023	0.09925	Sigmoid
40	0.30002	0.099217	Sigmoid
45	0.300019	0.112367	Sigmoid
50	0.300015	0.112367	Sigmoid
55	0.296658	0.163467	Softmax
60	0.295918	0.172683	Softmax
65	0.29527	0.182583	Softmax
70	0.294693	0.1908	Softmax
75	0.294168	0.198317	Softmax
80	0.293689	0.204883	Softmax
85	0.29325	0.21105	Softmax
90	0.292845	0.2169	Softmax
95	0.292474	0.221683	Softmax



4.6b. TanH initial, first run with random AF swapping, 300 epochs

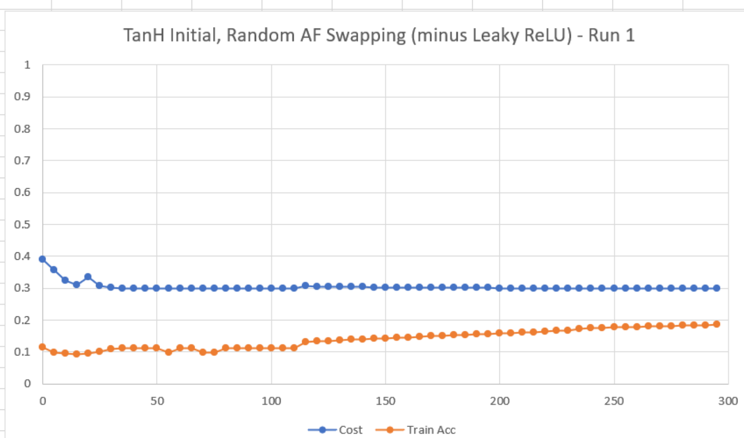
TanH Initial			
Train acc:	0.7669	Time:	133
Final cost:	0.1968		
Epoch	Cost	Train Acc	Act Funct
0	0.390736	0.113733	TanH
5	0.310951	0.0856	Softmax
10	0.38247	0.098133	L-ReLU
15	0.325669	0.114683	L-ReLU
20	0.30588	0.153683	L-ReLU
25	0.29724	0.211017	L-ReLU
30	0.290441	0.281717	L-ReLU
35	0.284534	0.343467	L-ReLU
40	0.279499	0.392317	L-ReLU
45	0.275226	0.44045	L-ReLU
50	0.2714	0.481183	L-ReLU
55	0.267783	0.513417	L-ReLU
60	0.26432	0.540217	L-ReLU
65	0.261008	0.56125	L-ReLU
70	0.257846	0.58045	L-ReLU
75	0.254831	0.5961	L-ReLU
80	0.251955	0.610483	L-ReLU
85	0.249213	0.622883	L-ReLU
90	0.246598	0.633283	L-ReLU
95	0.244105	0.6429	L-ReLU

4.6c. TanH initial, second run with random AF swapping, 300 epochs

TanH x1 = Baseline x2 = Random AF swapping, run 1	Two-sample Kolmogorov-Smirnov test D = 0.83333, p-value < 2.2e-16 alternative hypothesis: two-sided
TanH x1 = Baseline x2 = Random AF swapping, run 2	Two-sample Kolmogorov-Smirnov test D = 0.93333, p-value < 2.2e-16 alternative hypothesis: two-sided

4.6d. K-S Tests, TanH baseline ~ first and second random runs

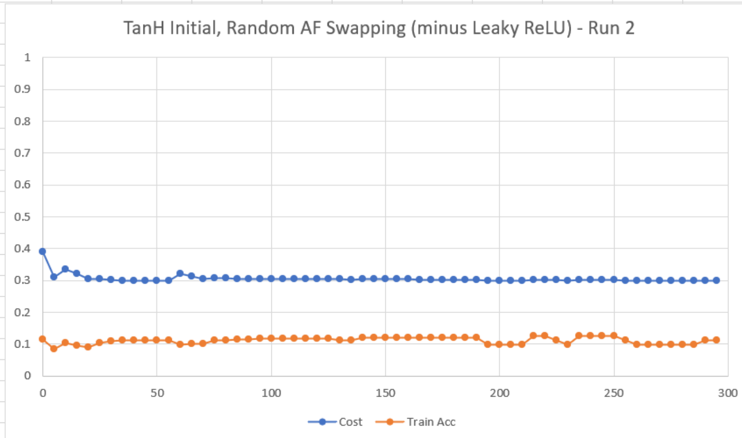
TanH Initial - Best performing Leaky ReLU AF not available			
Train acc:	0.1851	Time:	191
Final cost:	0.2879		
Epoch	Cost	Train Acc	Act Funct
0	0.390736	0.113733	TanH
5	0.356247	0.099217	Sigmoid
10	0.32274	0.095967	Sigmoid
15	0.310244	0.093633	Softmax
20	0.334908	0.096683	TanH
25	0.306624	0.102067	TanH
30	0.30205	0.109817	TanH
35	0.300209	0.111417	TanH
40	0.299962	0.112183	TanH
45	0.299954	0.112367	TanH
50	0.299952	0.112367	TanH
55	0.300074	0.09915	ReLU
60	0.299952	0.112367	TanH
65	0.299952	0.112367	TanH
70	0.300052	0.09915	ReLU
75	0.300044	0.09915	ReLU
80	0.299952	0.112367	TanH
85	0.299952	0.112367	TanH
90	0.299952	0.112367	TanH
95	0.300026	0.112367	ReLU



4.6e. TanH initial, first run with random AF swapping, L-ReLU suppressed, 300 epochs

TanH Initial - Best performing Leaky ReLU AF not available

Train acc:	0.1127	Time:	173
Final cost:	0.2999		
Epoch	Cost	Train Acc	Act Funct
0	0.390736	0.113733	TanH
5	0.310951	0.0856	Softmax
10	0.3354	0.102983	Sigmoid
15	0.321279	0.0951	Sigmoid
20	0.305589	0.08885	Sigmoid
25	0.305116	0.10335	ReLU
30	0.300738	0.108883	ReLU
35	0.300133	0.111083	ReLU
40	0.299985	0.11215	ReLU
45	0.299969	0.11235	ReLU
50	0.299969	0.112367	ReLU
55	0.299968	0.112367	ReLU
60	0.321663	0.098633	TanH
65	0.31398	0.102183	TanH
70	0.305137	0.102183	TanH
75	0.306874	0.1116	Softmax
80	0.306414	0.111817	Softmax
85	0.306015	0.11555	Softmax
90	0.305665	0.116117	Softmax
95	0.305356	0.11655	Softmax
100	0.305081	0.116867	Softmax



4.6f. TanH initial, second run with random AF swapping, L-ReLU suppressed, 300 epochs

TanH x1 = Baseline x2 = Random AF swapping, minus L-ReLU, run 1	Two-sample Kolmogorov-Smirnov test D = 0.61667, p-value = 2.465e-10 alternative hypothesis: two-sided
TanH x1 = Baseline x2 = Random AF swapping, minus L-ReLU, run 2	Two-sample Kolmogorov-Smirnov test D = 0.41667, p-value = 5.986e-05 alternative hypothesis: two-sided

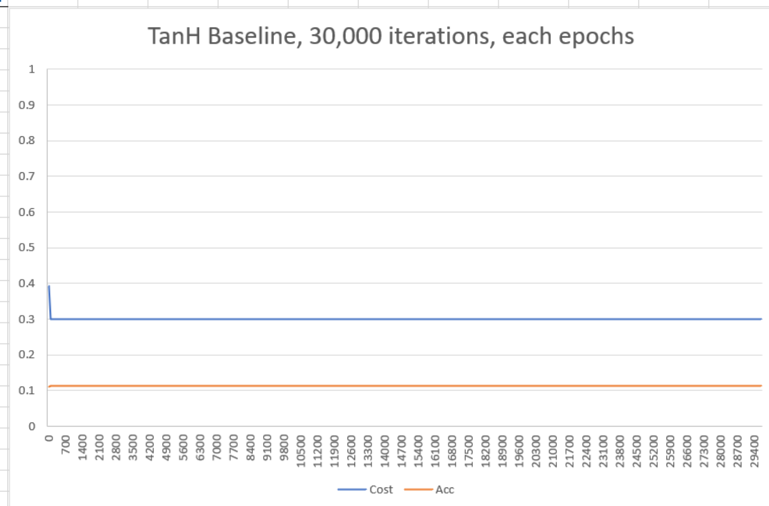
4.6g. K-S Tests, TanH baseline ~ first and second random runs, L-ReLU suppressed

Figure 4.6. TanH over 300 epochs with K-S Tests

4.7 MNIST Data Set, 30,000 epochs

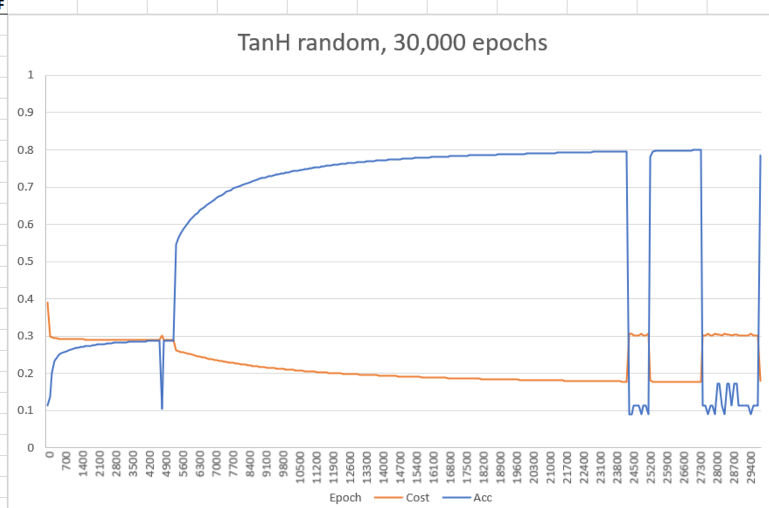
The fluctuations observed in several of the random AF swapping accuracies late in the 300 epoch cycles led to investigating what may happen with training accuracies even later in training sessions. A low-performing activation function, TanH, was selected to initiate baseline, random swapping with all AFs available and random swapping with the highest performer, Leaky ReLU, suppressed, and trained over 30,000 epochs. The results are reported in Figure 4.7 with cost and accuracy recorded at 100 epoch intervals.

Epoch	Cost	Acc	AF
0	0.394715	0.1115	TanH
100	0.299952	0.112367	TanH
200	0.299952	0.112367	TanH
300	0.299952	0.112367	TanH
400	0.299952	0.112367	TanH
500	0.299952	0.112367	TanH
600	0.299952	0.112367	TanH
700	0.299952	0.112367	TanH
800	0.299952	0.112367	TanH
900	0.299952	0.112367	TanH
1000	0.299952	0.112367	TanH
1100	0.299952	0.112367	TanH
1200	0.299952	0.112367	TanH
1300	0.299952	0.112367	TanH
1400	0.299952	0.112367	TanH
1500	0.299952	0.112367	TanH
1600	0.299952	0.112367	TanH
1700	0.299952	0.112367	TanH
1800	0.299952	0.112367	TanH
1900	0.299952	0.112367	TanH
2000	0.299952	0.112367	TanH
2100	0.299952	0.112367	TanH
2200	0.299952	0.112367	TanH
2300	0.299952	0.112367	TanH



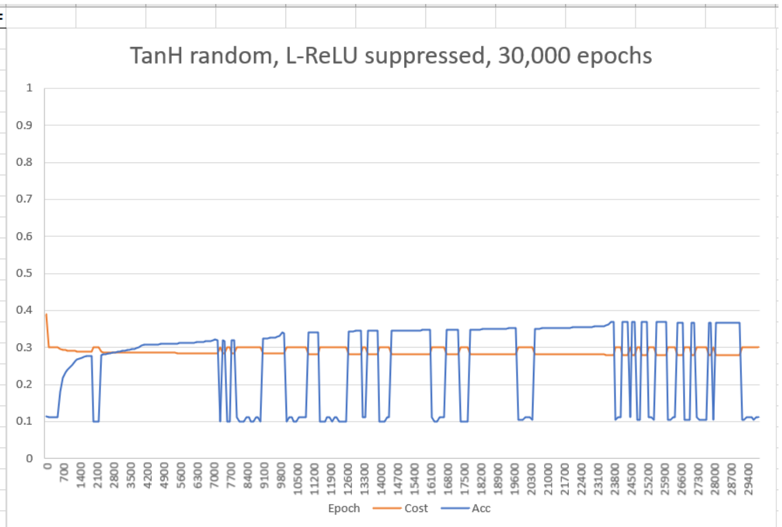
4.7a. TanH baseline, 30,000

Epoch	Cost	Acc	AF
0	0.390736	0.113733	TanH
100	0.299384	0.136883	Softmax
200	0.295722	0.199583	Softmax
300	0.294328	0.234417	Softmax
400	0.293529	0.24325	Softmax
500	0.292986	0.249067	Softmax
600	0.29258	0.25385	Softmax
700	0.29226	0.25745	Softmax
800	0.291999	0.260217	Softmax
900	0.291779	0.262667	Softmax
1000	0.291592	0.264967	Softmax
1100	0.291429	0.267017	Softmax
1200	0.291286	0.268383	Softmax
1300	0.291159	0.2693	Softmax
1400	0.291045	0.2702	Softmax
1500	0.290942	0.271167	Softmax
1600	0.290848	0.272383	Softmax
1700	0.290763	0.2733	Softmax
1800	0.290683	0.274233	Softmax
1900	0.29061	0.275067	Softmax
2000	0.290541	0.276017	Softmax
2100	0.290477	0.276867	Softmax
2200	0.290416	0.277433	Softmax
2300	0.290359	0.2784	Softmax



4.7b. TanH initial, random AF swapping, 30,000 epochs

Epoch	Cost	Acc	AF
0	0.390736	0.113733	TanH
100	0.30003	0.112367	Sigmoid
200	0.300034	0.112367	ReLU
300	0.300055	0.112367	ReLU
400	0.299952	0.112367	TanH
500	0.300093	0.112367	ReLU
600	0.296156	0.180783	Softmax
700	0.294181	0.217583	Softmax
800	0.29285	0.2313	Softmax
900	0.291895	0.240033	Softmax
1000	0.291175	0.246467	Softmax
1100	0.290611	0.252517	Softmax
1200	0.290154	0.263817	Softmax
1300	0.289776	0.268517	Softmax
1400	0.289455	0.270717	Softmax
1500	0.28918	0.272967	Softmax
1600	0.288941	0.275317	Softmax
1700	0.28873	0.276767	Softmax
1800	0.288542	0.277683	Softmax
1900	0.288373	0.277633	Softmax
2000	0.300991	0.098633	ReLU
2100	0.300959	0.098633	Sigmoid
2200	0.300923	0.098633	ReLU
2300	0.287583	0.28045	Softmax

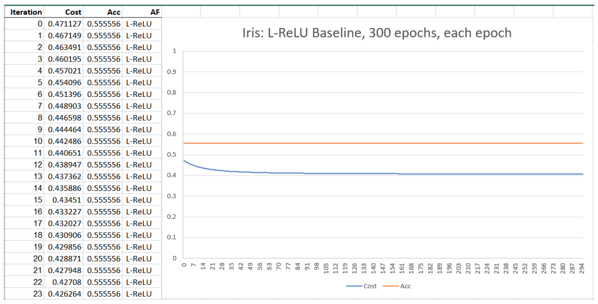


4.7c. TanH initial, random AF swapping run, L-ReLU suppressed, 30,000 epochs

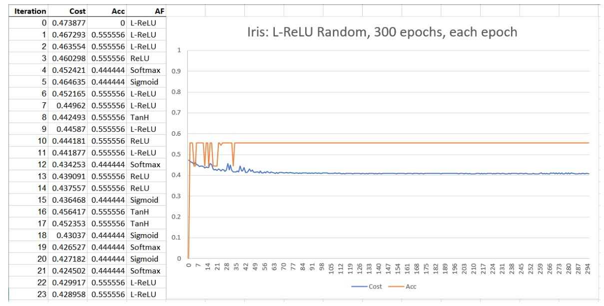
Figure 4.7. TanH, 30,000 epochs

4.8 Iris Data Set, 300 epochs

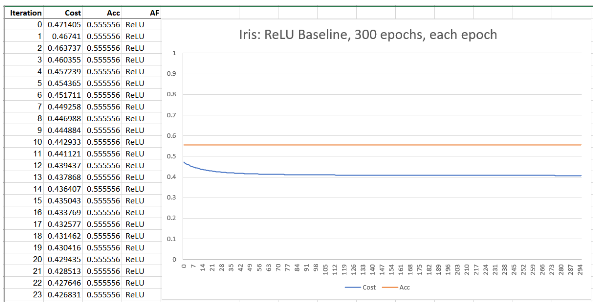
Given the results observed with the MNIST data set, another commonly used data set, Iris, was selected to performs the same study and compare results. The Iris data is a small set consisting of one label, four features and 150 instances. A baseline training session of 300 epochs was conducted for each of the five activation functions studied in this paper. A training session was then initiated with the AF from each baseline and then allowed to randomly swap to another AF after five epochs and a negative accuracy value. In each training where random swapping was allowed, all five activation functions included in the study were available to the swapping algorithm. Data was captured after each epoch. The baseline and random swapping results are shown in Figure 4.8.



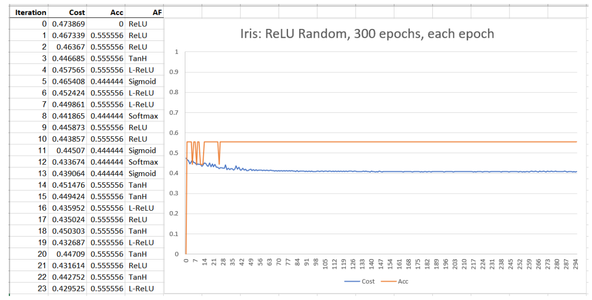
a. Leaky ReLU, baseline, 300 epochs



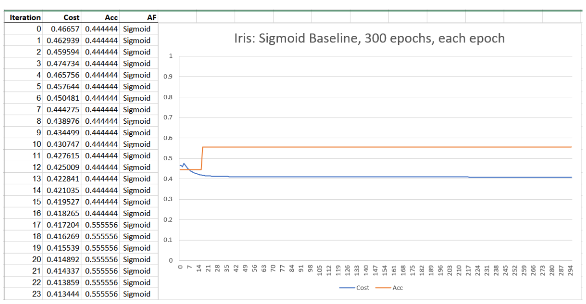
b. Leaky ReLU, random, 300 epochs



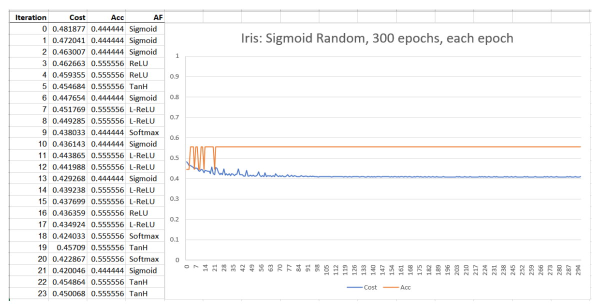
c. ReLU baseline, 300 epochs



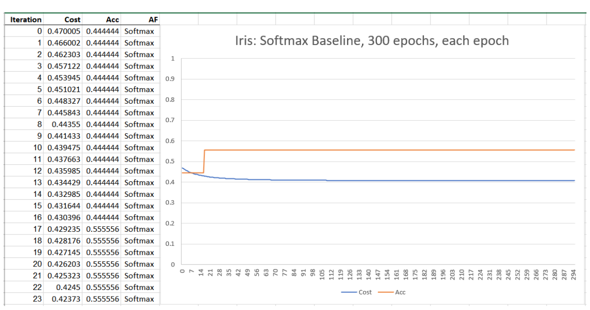
d. ReLU random, 300 epochs



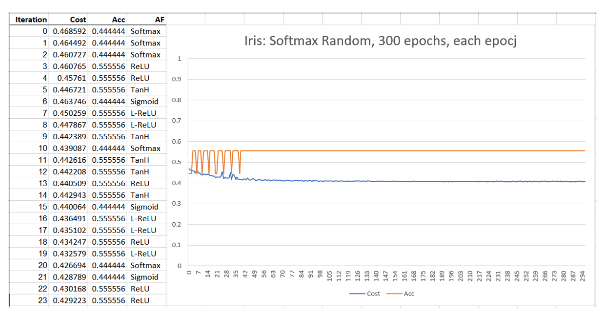
e. Sigmoid baseline, 300 epochs



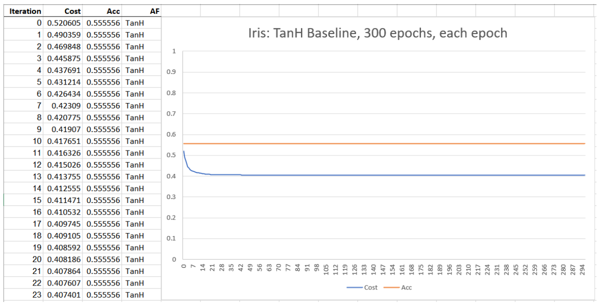
f. Sigmoid random, 300 epochs



g. Softmax baseline, 300 epochs

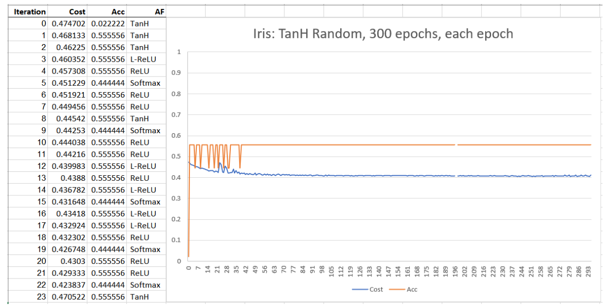


h. Softmax random, 300 epochs



Iteration	Cost	Acc	AF
0	0.520605	0.555556	TanH
1	0.490359	0.555556	TanH
2	0.469848	0.555556	TanH
3	0.445875	0.555556	TanH
4	0.437691	0.555556	TanH
5	0.431214	0.555556	TanH
6	0.426434	0.555556	TanH
7	0.42309	0.555556	TanH
8	0.420775	0.555556	TanH
9	0.41907	0.555556	TanH
10	0.417651	0.555556	TanH
11	0.416326	0.555556	TanH
12	0.415026	0.555556	TanH
13	0.413755	0.555556	TanH
14	0.412555	0.555556	TanH
15	0.411471	0.555556	TanH
16	0.410532	0.555556	TanH
17	0.409745	0.555556	TanH
18	0.409105	0.555556	TanH
19	0.408592	0.555556	TanH
20	0.408186	0.555556	TanH
21	0.407864	0.555556	TanH
22	0.407607	0.555556	TanH
23	0.407401	0.555556	TanH

i. TanH baseline, 300 epochs



Iteration	Cost	Acc	AF
0	0.474702	0.022222	TanH
1	0.468133	0.555556	TanH
2	0.46225	0.555556	TanH
3	0.460352	0.555556	L-ReLU
4	0.457308	0.555556	ReLU
5	0.451229	0.444444	Softmax
6	0.451921	0.555556	ReLU
7	0.449456	0.555556	ReLU
8	0.44542	0.555556	TanH
9	0.44253	0.444444	Softmax
10	0.444038	0.555556	ReLU
11	0.44216	0.555556	ReLU
12	0.439983	0.555556	L-ReLU
13	0.4388	0.555556	ReLU
14	0.436782	0.555556	L-ReLU
15	0.431648	0.444444	Softmax
16	0.43418	0.555556	L-ReLU
17	0.432924	0.555556	L-ReLU
18	0.432302	0.555556	ReLU
19	0.426748	0.444444	Softmax
20	0.4303	0.555556	ReLU
21	0.429333	0.555556	ReLU
22	0.423837	0.444444	Softmax
23	0.470522	0.555556	TanH

j. TanH baseline, 300 epochs

Figure 4.8. Iris data set, 300 epochs

4.9 Summary

The investigation was conducted by first establishing baseline training runs for each of the five chosen activations functions, Leaky ReLU, ReLU, Sigmoid, Softmax and TanH, recording the respective accuracy and cost data points. The subsequent efforts to train using the entropy-driven activation function swapping algorithm produced results unusable for testing the hypothesis. The swapping function approach was discarded in favor of one that would randomly swap activation functions when the accuracy after an epoch became zero or less.

Results for baseline and AF swapping runs for the MNIST data set were captured at intervals of five epochs. Baseline and AF swapping runs were conducted using the same hyperparameters and computing environment over 300 epochs. Sixty-four nodes in the single hidden layer were selected as that arrangement produced slightly better than random results yielding a large improvement space for evaluating changes due to AF swapping. The output layer for each consisted of ten Softmax nodes.

Training run sessions began with each of the five activation functions and were allowed to proceed without swapping AFs through the first five epochs. This was to establish each AF as the initial learning function and to provide a basis for randomly swapping to another AF should accuracy fail to improve at any point during training.

The baseline performance evaluations revealed Leaky ReLU as the best AF by a large performance margin. Two AF swapping runs with all five activation functions available to the swapping algorithm were conducted and, then, to examine results among the four similar lesser performing activation functions, two additional runs where Leaky ReLU was suppressed were run. The performance of each run are summarized in Table 4.1 and show that Leaky ReLU was the best performing activation function when comparing its accuracy rate versus the other four in the baseline training sessions. Among the eighteen runs where AF swapping was allowed, accuracy increased significantly in sixteen of the runs while in one run it perfectly followed the baseline and in another it declined the same as the baseline.

Curious accuracy plots near the end of the 300 epoch runs suggested possible convergence issues with the AF swapping algorithm. To examine if this artifact might be exhibited deeper into training, a low-performing AF, TanH, was selected for a session of 30,000 epoch training runs. A 30,000-epoch baseline was established and then training runs of 30,000 epochs with AF swapping were conducted with and without Leaky ReLU availability. In all three runs accuracy and cost are recorded after each 100 epochs. The plots indicate that over numerous epochs resulting in convergence where accuracy goes to zero (or below), initiating an activation swap produces similar artifacts as observed with the original training sessions.

To conclude, the study was applied to the common Iris data set over a 300-epoch series of runs. Baseline performance values were determined for each activation function. Then, sessions where all AFs were available for random swapping were performed with each AF initiating a run for the first five epochs. The results showed rapid convergence. In some cases substantial AF swapping effects were observed during training and prior to convergence.

MNIST					
Leaky ReLU	**Baseline**	**Run 1**	**Run 2**	**Run 3**	**Run 4**
Accuracy Δ	0.6880	0.6431	0.6850	N/A	N/A
Accuracy % Δ	793%	741%	789%	N/A	N/A
Cost Δ	-0.2296	-0.2248	-0.2275	N/A	N/A
ReLU					
Accuracy Δ	0.0253	0.0644	0.6858	0.2035	0.0721
Accuracy % Δ	29%	74%	787%	234%	83%
Cost Δ	-0.1232	-0.1219	-0.2282	-0.1337	-0.1232
Sigmoid					
Accuracy Δ	0.0279	0.0279	0.6822	0.0668	0.1033
Accuracy % Δ	33%	33%	807%	79%	122%
Cost Δ	-0.0899	-0.0899	-0.1927	-0.0853	-0.0921
Softmax					
Accuracy Δ	0.0416	0.6734	0.6841	0.0857	0.0822
Accuracy % Δ	50%	803%	815%	102%	98%
Cost Δ	-0.0058	-0.1076	-0.1157	-0.0088	-0.0104
TanH					
Accuracy Δ	-0.0013	0.2675	0.6533	0.0715	-0.0013
Accuracy % Δ	-1%	235%	575%	63%	-1%
Cost Δ	-0.0908	-0.1038	-0.1939	-0.0928	-0.0908

Table 4.1 Performance summary

Chapter 5

Discussion

5.1 Overview

The research question guiding this study asked, *will changing activation functions at the node layer level during learning improve neural network accuracy and discrimination performance?* After strictly controlling conditions for data preparation, computing platform, environment, and hyperparameter settings, the results as reported above show statistically significant accuracy improvements during training sessions exhibiting acceptable cost reductions in sixteen of the eighteen 300-epoch MNIST runs. Additionally, the sessions where the poorly performing TanH activation function was selected to investigate post-convergence accuracy and cost spiking as artifacts of the AF swapping algorithm confirmed that AF replacement did indeed result in such spikes over 30,000 training epochs. Lastly, the common Iris data set results suggest that much smaller data sets converge rapidly, are not adversely affected by AF swapping but are prone to post-convergence spiking with AF swapping controlled by an unsophisticated algorithm. The conclusion is that parameterizing activation functions to change at the layer level in response to flattening accuracies during training can improve neural network accuracy and achieve convergence.

Evaluating the study through the lens of the three goals outlined in the introduction, (1) this investigation increased understanding of how neural networks are affected during training by parameterizing activation functions. The effects of AF swapping appear similar to boosting algorithms commonly used in other machine learning algorithms. The study also demonstrates how typical cost functions can be extended with an additional algorithm layer to control a parameter, activation functions during training, that is central to learning accuracy. Also (2), swapping activation functions during training can be as, or nearly as, efficient as static (hyperparameter) AFs as can be seen by comparing learning progress among the different AFs over the first twenty to fifty epochs in the MNIST run plots above. One activation function, Leaky ReLU greatly outperformed the other four in the baseline tests. L-ReLU's accuracy increased by 793% while the others (ReLU, Sigmoid, Softmax and TanH), improved by 29%, 30%, 50% and -1%, respectively. The impact of swapping activation functions when initializing training runs with L-ReLU resulted in slightly reduced accuracy performances of 741% and 789%, suggesting that there might be a minor penalty, or a requirement of longer training runs to achieve convergence, if at least one of the available AFs is a strong performer and initiates the training on the data set. Among the four similar and lower-performing activation functions, no swapping runs performed worse than their respective baselines although two of the sixteen runs matched the baseline results. Considering that in every MNIST case except the two that matched the baseline results, with accuracy improvement ranging between 33% and 815%, the efficiency improvement is confirmed. Lastly (3), potential practitioner bias relating to AF selection can be reduced or eliminated by parameterizing activation functions. Providing a suite of available activation functions that support the varied functional shifts during feed-forward and, likewise, the derivative adjustments during backpropagation while allowing the neural network to select the best performer from among them during training removes user bias and frees the NN to learn efficiently and agnostically.

The study began with an entropy-based algorithm designed to interpret changes in loss value magnitudes and select another of the five available activation functions to improve learning performance. The algorithm had unanticipated effects on the learning progress that resulted in large swings and spiking in both accuracies and costs (Figure 4.1a). Adjustments to the algorithm resulted in some improvements (Figure 4.1c, 4.1d) but failed to result in learning improvement.

A new approach was invoked that randomly swapped activation functions based on zero or negative accuracy improvement values. The algorithm provided for training to progress through five epochs with the selected initial AF to establish training. Then, if accuracy fell to zero or below, the algorithm would randomly swap a new activation function into the hidden layer nodes and training would continue. This random selection method has a strong precedent in neural network design. It is the method used to select nodes to be rendered zero—to drop out—when regularizing NNs using the dropout technique [51].

5.2 Contributions

The study confirmed that in a controlled laboratory environment, with stable and common data sets, parameterization of activation functions to improve learning performance is possible and computationally efficient. The implication is that a source of practitioner bias—the selection of activation functions for the hidden layers of a neural network—can be substantially or completely eliminated if a broad enough spectrum of AFs are available for swapping as training progresses to convergence.

Additionally, the study contributes a robust laboratory neural network model that can be utilized to expand investigations to other data sets, to increase the spectrum of available activation functions for study, and to increase the number of hidden layers for AF swapping. The system will, with some modification, form the basis for swapping AFs at the single node level. The model is stable and performs on a variety of computing platforms, including CPU-based systems, in the Windows, MacOS and Linux operating systems, and requires memory only as required by the data set being studied. The NN model is easily programmable to extend data capture to include values of weights, biases, the learning rate and processing times at any iteration.

5.3 Limitations

This study is narrowly focused on one of the several parameterizations (hyperparameters and parameters) common to neural network training: The activation function and how its selection results in significant differences in training results. While training over enough epochs will, as defined by the Central Limit Theorem, result in convergence (unless there exists hyperparameter problems such as too large of a learning rate, or bad data), various AFs used on a common training data set produce very different progressions to convergence and open the possibility of prematurely claiming convergence. Because the numerous hyperparameter options and ranges are decided by users, combined with data preparation decisions by users, affect training outcomes, this study cannot encompass all possibilities. It is limited to investigating the outcomes of AF choices while controlling for other hyperparameters, determining if bias is significant in hyperparameter choices and exploring dynamic AF selection; i.e., AF as a parameter, for bias mitigation.

5.4 Future Research

Three specific areas for future research arise immediately from this study. First, the swapping of activation functions to increase accuracy loosely mimics boosting. With boosting known to often produce overfitting, studies to evaluate possible overfitting should be conducted. In addition to testing for overfitting using the parameters and data from this study, other robust and challenging data sets and multiple layers should be studied as increasing layers generally increases the possibility of overfitting [51]. Too, expanding layers and individual node access to AF swapping should be investigated as an ensemble approach to mimic learning improvement through techniques common to other machine learning approaches. Second, investigations into a more robust AF swapping algorithm will be key to producing a better performing NN with smoother cost and accuracy curves. Based on the failures encountered early in this study with the Shannon entropy-based algorithm, expanding the algorithm to a cross-entropy function may produce more predictable results. Current research in this area [52] is promising and may prove useful for managing activation function swapping. Third, while this study benefitted from using the predictable and stable MNIST data set as its primary test subject, establishing the value of this research will require AF swapping to be studied using a variety of data sources to test its generalizability and resiliency. Immediate opportunities exist with pixel-shifted MNIST examples, Kannada-MNIST, CIFAR-10 and similar data sets that can provide researchers a varied series of testbeds. This research is a first step, after the early work by Agostinelli, Majetic and others cited herein, and the work to establish a viable neural network with parameterized activation functions is promising.

References

[1] Agostinelli, F., Hoffman, M., et. al., (2015). Learning Activation Functions to Improve Deep Neural Networks. ICLR 2015, International Conference on Learning Representations, 2015. The Hilton San Diego Resort and Spa, San Diego, CA, May 7-9, 2015.

[2] Arpit, D.; Jastrzebski, S., et. al., (2017, July 1), A Closer Look at Memorization in Deep Networks. Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

[3] Fawcett, T. (2005, December 19). An Introduction to ROC Analysis. Patter Recognition Letters, Volume 27, Issue 8, June 2006, pages 861-874.

[4] Glorot, X., Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR, 2010.

[5] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.

[6] Hutton, L. (1992). Using Statistics to Assess the Performance of Neural Network Classifiers. Johns Hopkins APL Technical Digest, Volume 13, Number 2 (1992).

[7] Jordan, M. (2018). Artificial Intelligence – The Revolution Hasn't Happened Yet. Medium, (2018, April 18). https://medium.com/@mijordan3/artificial-intelligence-the-revolution-hasnt-happened-yet-5e1d5812e1e7

[8] Kim, B., Wattenberg, M., et. al., (2018). Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[9] Krizhevsky, A., Sutskever, I., et. al., ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25 (NIPS 2012).

[10] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[11] LeCun, Y., @EPFL – Self-supervised learning: could machines learn like humans? Retrieved January 27, 2019.

[12] LeCun, Y., Bengio, Y., Hinton, G. 2015. Deep learning. Nature, 521:7553, 436-444.

[13] Majetic, D., Brezak, D., et. al., (2004). Dynamic neural network with adaptive Gauss neuron activation function. DAAAM Scientific Book, 2004, Vienna, Austria, 2004.

[14] Majetic, D., Brezak, D., et. al., (2007). Modified Dynamic Neuron Model. DAAAM Scientific Book, 2007, Vienna, Austria, 2007.

[15] Maruan, A., Bansal, T., et. al., (2018). Continuous adaptation via Meta-learning in Nonstationary and Competitive Environments. ICLR 2018, International Conference on Learning Representations, 2018. Vancouver Convention Center, Vancouver, BC, Canada, April 30 – May 3, 2018.

[16] Munkhdalai, T., Xingdi, Y., (2018). Rapid Adaptation with Conditionally Shifted Neurons. Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[17] Reddi, S., Kale, S., et. al., (2018, March 26). On the convergence of Adam and Beyond. International Conference on Learning Representations, 2018. Vancouver Convention Center, Vancouver, BC, Canada, April 30 – May 3, 2018.

[18] Sabour S, Frosst N, Hinton GE. Dynamic routing between capsules. InAdvances in neural information processing systems 2017 (pp. 3856-3866).

[19] Turing, Alan (October 1950), "Computing Machinery and Intelligence" (PDF), Mind, LIX (236): 433–460, doi:10.1093/mind/LIX.236.433.

[20] Zhang, C., Bengio, S., et. al. (2017, February 26). Understanding deep learning requires re-thinking generalization. ICLR 2017, International Conference on Learning Representations, Palais des Congres Neptune, Toulon, France, April 24-26, 2017.

[21] Tim Fountain, Brian McCarthy, et. al., "Building the AI-Powered Organization," Harvard Business Review, July – August 2019, pp 62 – 73.

[22] Vilma L. Patel, Edward H. Shortliffe, et. al., "The Coming Age of Artificial Intelligence in Medicine," Artif Intell Med. 2009 May; 46(1): 5-17.

[23] Yunhe Pan, "Heading toward Artificial Intelligence 2.0," Engineering, Volume 2, Issue 4, December 2016, pp 409-413.

[24] Rachel K. E. Bellamy, Kuntel Dey, et. al., "AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias," arXiv:1810.01943v1 [cs.AI] 3 October 2018.

[25] Donald A. Waterman, "Machine Learning of Hueristics," Dissertation, Computer Science Department, School of Humanities and Sciences, Stanford University, December 1968.

[26] Locke, J.M., Claiborne, J., "Neural Network Fraud Detection Dilemma: The Curious Role of Activation Functions," 2018 Pre-ICIS SIGDSA Symposium, December 12, 2018

[27] Jonathan Zittrain, “The Hidden Costs of Automated Thinking,” The New Yorker, Annals of Technology, July 23, 2019.

[28] D. Nguyen and B. Widrow, “The truck backer-upper: An example of self-learning in neural networks.” Proceedings of the International Joint Conference on Neural Networks, pages II-357-363, IEEE, June 1989.

[29] Richard G. Taylor, “The Growth of Scientific Knowledge in MIS: The MIS Paradigm.” AIS Electronic Library (AISeL), AMCIS 2005 Proceedings, January 1, 2005

[30] Lee, A. (2001) Editorial, MIS Quarterly, 25(1), iii-vii.

[31] Ackoff, R. L. (1999) Ackoff’s Best. New York: John Wiley & Sons, pp 170 – 172.

[32] Enid Mumford, “Risky ideas in the risk society,” Journal of Information Technology (1996), 11, 321-331.

[33] Giddens, A. (1990). “The consequences of modernity.” Stanford, CA: Stanford University Press.

[34] Mumford, E. (1983). “Designing Human Systems - the ETHICS Method.” Manchester Business School, Manchester, UK.

[35] Joana Porra and Rudy Hirscheim. “A Lifetime of Theory and Action on the Ethical Use of Computers: A Dialogue with Enid Mumford.” Volume 8, Issue 9, Article 2, pp. 467-478, September 2007

[36] Dingli, A. (2018). It’s magic…I owe you no explanation! Medium, (2018, November 19). https://becominghuman.ai/its-magic-i-owe-you-no-explanation-explainableai-43e798273a08

[37] Sage, A. P. (1981). “Behavioral and organizational considerations in the design and information systems and processes for planning and decision support,” IEEE Transactions on Systems, Man and Cybernetics, SMC-11(9), 1981, pp. 640-678.

[38] Vasarhelyi, M. A. (1977). "Man-machine planning systems: A cognitive style examination of interactive decision making." Journal of Accounting Res., vol 15, pp. 138-153, Spring 1977.

[39] Smith, S. W. (2006). "The Scientist and Engineer's Guide to Digital Signal Processing." San Diego, CA. California Technical Publishing, Chapter 14.

[40] Boudreau, M-C., Gefen, D., et. al. (2001). "Validation in information systems research: A state-of-the-art assessment." MIS Quarterly, Mar 2001.

[41] Paradice, D.B. and Courtney, Jr., J.F. (1986) "Controlling Bias in User Assertions in Expert Decision Support Systems for Problem Formulation," *Journal of Management Information Systems*, 3(1), pp. 52-64.

[42] Paradice, D.B. and Courtney, Jr., J.F. (1988) "Dynamic Construction of Statistical Models in Managerial DSS," *Annals of Operations Research: Approaches to Intelligent Decision Support*, 12, pp. 321-336.

[43] Courtney, Jr., J.F. and Paradice, D.B. (1993) "Studies in Managerial Problem Formulation Systems," *Decision Support Systems*, 9(3), pp. 413-423.

[44] Hall, D.J. and Paradice, D.B. (2005) "Philosophical Foundations for a Learning-Oriented Knowledge Management System for Decision Support," *Decision Support Systems*, 39(3), pp. 445-461.

[45] Chae, B., Paradice, D.B., Courtney, Jr., J.F., and Cagle, C.J. (2005) "Incorporating an Ethical Perspective into Decision Support Systems Design," *Decision Support Systems*, 40(2), pp. 197-212.

[46] Hall, D.J. and Paradice, D.B. (2007) "Investigating Value-Based Decision Bias and Mediation – Do You Do as You Think?" *Communications of the ACM,* 50(4), pp. 81-85.

[47] Cui, P. (2017) “EvoDNN: An Evolutionary Deep Neural Network with Heterogenous Activation Functions.” Thesis, School of Computing Science, University of British Columbia.

[48] Pizarro, J., Guerrero, E., et. al. (2000). “A Statistical Model Selection Strategy Applied to Neural Networks.” ESNN’2000 proceedings – European Symposium on Artificial Neural Networks, Bruges (Belgium), 26-28 April 2000, D-Facto public, ISBN 2-930307-00-5, pp. 55-60.

[49] Shannon, C.E., (1948). “A Mathematical Theory of Communication.” The Bell Systems Technical Journal, Vol. 27, pp 379-623, July, October, 1948.

[50] Zheng, M., Tang, W., Zhao, X., (2019). “Hyperparameter optimization of neural network-driven spatial models accelerated using cyber-enabled high-performance computing.” International Journal of Geographical Information Science, 33:2, 314-315, DOI: 10.1080/13658816.2018.1530355.

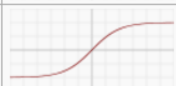
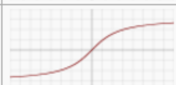
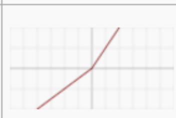
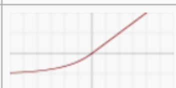
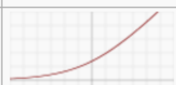
[51] Srivastava, N., Hinton, G., et. al., (2014). “Dropout: A simple way to prevent neural networks from overfitting.” The Journal of Machine Learning Research, 15 (1), 1929-1958, 2014.

[52] Barron, J., (2019). “A General and Adaptive Robust Loss Function.” arXive: 1701.03077v10 [cs.CV], 2019.

[53] Rubeiro, M., Singh, S., et. al., (2016). “’Why Should I Trust You?’ Explaining the Predictions of Any Classifier.” arXive: 1602.04938v3 [cs.LG] 9 Aug 2016.

Appendix 1: Typical Activation Functions

(source: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6)

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

Activation functions, by design, will result in different results when fed data. The motivation to continually construct new activation functions is a recognition that data and desired learning results will require targeted functions.

Appendix 2: Code for a neural network binary classifier to test various AF combinations

This is a simple neural network binary classifier, built from scratch. It consists of three layers. The first is the input layer which has the same size as the X axis. The second is the hidden layer with size user defined as n_h (64 for this study). Activation functions in the second layer may be swapped. The last layer is the output layer, its size is defined buy the number of possible predictions. The last layer uses the Softmax activation function. The network dimensions are: Input(n_x) = 784, hidden(n_h) = 64, and output(n_y) = 10.

Mr. David W. Edwards, a graduate student in computer science at Auburn University, provided helpful and hands-on assistances with the development of this algorithm.

```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": true,
    "scrolled": true
   },
   "outputs": [],
   "source": [
    "import numpy as np\n",
    "import time\n",
    "import math\n",
    "from sklearn.datasets import fetch_openml\n",
    "from sklearn.metrics import classification_report, accuracy_score\n",
    "import pandas as pd"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "# Get MNIST dataset (70,000 x 784) and split into training (60,000 x 
784) and\n",
    "# testing (10,000 x 784) data sets.  NOTE:  This cell should be executed 
only once\n",
    "# per multiple runs of the NN code cell if processing identical data for 
comparing\n",
    "# processing performance and results is desired.\n",
    "# NOTE: Must be connected to Internet\n",
    "\n",
    "X, y = fetch_openml('mnist_784', version=1, return_X_y=True)\n",
    "# prepare dataset\n",
    "X = X / 255\n",
    "\n",
    "digits = 10\n",
    "examples = y.shape[0]\n",
    "\n",
```

```
    "y = y.reshape(1, examples)\n",
    "\n",
    "Y_new = np.eye(digits)[y.astype('int32')]\n",
    "Y_new = Y_new.T.reshape(digits, examples)\n",
    "# set train test split\n",
    "f = 60000\n",
    "m_test = X.shape[0] - f\n",
    "# split dataset into train and test\n",
    "X_train, X_test = X[:f].T, X[f:].T\n",
    "Y_train, Y_test = Y_new[:,:f], Y_new[:,f:]\n",
    "np.random.seed(1)\n",
    "shuffle_index = np.random.permutation(f)\n",
    "X_train, Y_train = X_train[:, shuffle_index], Y_train[:,
shuffle_index]\n",
    "print(X_train.shape[0])\n",
    "print(X_train.shape[1])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "# Define the activactive functions to be used for feed-forward and
backpropagation\n",
    "\n",
    "# Softmax\n",
    "def softmax(X, theta = 1.0, axis = None):\n",
    "    y = np.atleast_2d(X)\n",
    "    if axis is None:\n",
    "        axis = next(j[0] for j in enumerate(y.shape) if j[1] > 1)\n",
    "    y = y * float(theta)\n",
    "    y = y - np.expand_dims(np.max(y, axis = axis), axis)\n",
    "    y = np.exp(y)\n",
    "    ax_sum = np.expand_dims(np.sum(y, axis = axis), axis)\n",
    "    p = y / ax_sum\n",
    "    if len(X.shape) == 1: p = p.flatten()\n",
    "\n",
    "    return p\n",
    "\n",
    "# ReLU\n",
    "def relu(Z1):\n",
    "    A1 = np.maximum(0,Z1)\n",
    "    return A1\n",
    "\n",
    "def relu_back(dA1):\n",
    "    dZ1 = np.greater(dA1, 0).astype(int)\n",
    "    return dZ1\n",
    "\n",
    "# TanH\n",
    "def tanh(Z1):\n",
    "    A1 = np.tanh(Z1)\n",
    "    return A1\n",
    "\n",
    "def tanh_back(dA1):\n",
    "    return 1 - np.power(tanh(dA1), 2)\n",
```

```
   "\n",
   "# Sigmoid\n",
   "def sigmoid(Z1):\n",
   "    A1 = 1/(1+np.exp(-Z1))\n",
   "    return A1\n",
   "\n",
   "def sigmoid_back(dA1):\n",
   "    dZ1 = sigmoid(dA1) * (1-sigmoid(dA1))\n",
   "    return dZ1\n",
   "\n",
   "# Leaky ReLU\n",
   "def lrelu(Z1):\n",
   "    A1 = np.maximum(Z1 * 0.01, Z1 )\n",
   "    return A1\n",
   "\n",
   "def lrelu_back(dA1):\n",
   "    dZ1 = np.ones_like(dA1)\n",
   "    dZ1[dA1 < 0] = 0.01\n",
   "    return dZ1"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "scrolled": false
  },
  "outputs": [],
  "source": [
   "### RANDOM ###\n",
   "# Use this cell to produce set or randomized initial activation\n",
   "#   function and then random AF if learning is not increasing.\n",
   "# Set number of epochs (300 is good), initial learning rate, if you\n",
   "#   want to set or randomize initial AF selection, and if you want\n",
   "#   verbose print output or CSV output for copy-and-past for Excel.\n",
   "\n",
   "# Loss function\n",
   "def compute_multiclass_loss(Y, Y_hat):\n",
   "    L = np.sqrt(((Y_hat - Y) ** 2).mean())\n",
   "\n",
   "    return L\n",
   "\n",
   "n_x = X_train.shape[0]\n",
   "\n",
   "# Value sets number of hidden layer nodes\n",
   "n_h = 64                                # 16\n",
   "\n",
   "# Set starting learning rate\n",
   "int_learning_rate = 0.1                # 0.1\n",
   "\n",
   "# Randomize starting weight, or use random seed\n",
   "np.random.seed(1)\n",
   "W1 = np.random.randn(n_h, n_x)\n",
   "b1 = np.zeros((n_h, 1))\n",
   "W2 = np.random.randn(digits, n_h)\n",
   "b2 = np.zeros((digits, 1))\n",
   "\n",
```

```
    "\n",
    "X = X_train\n",
    "Y = Y_train\n",
    "loss = []\n",
    "xs = []\n",
    "acc = []\n",
    "\n",
    "m = Y.shape[1]\n",
    "start = time.time() \n",
    "\n",
    "# Set number of iterations\n",
    "epochs = 300\n",
    "\n",
    "# For verbose results, set to 1, else 0 for copy-and-paste\n",
    "verbose_print = 0\n",
    "\n",
    "# Randomly choose initial activation function index -or-\n",
    "af_rnd = random.randint(0, 4)\n",
    "# uncomment 'af_rnd = ' line to choose initial activation function
index\n",
    "#   0 = Leaky ReLU\n",
    "#   1 = ReLUP\n",
    "#   2 = TanH\n",
    "#   3 = Sigmoid\n",
    "#   4 = Softmax\n",
    "af_rnd = 2\n",
    "    \n",
    "for i in range(epochs):\n",
    "    ##FOWARD PASS\n",
    "    #input/hidden layer\n",
    "    Z1 = np.dot(W1,X) + b1\n",
    "   \n",
    "    # Choose feed forward hidden layer activation function\n",
    "    # randomly if at least five iterations into training\n",
    "    # and accuracy is equal to or below zero\n",
    "    if (len(loss) > 5 and (acc[len(acc)-1] <= acc[len(acc)-2])):
         #  ((len(loss) > 5) and (cost < .3))\n",
    "        \n",
    "        while af_rnd == af_indx:            # Randomly choose a
different\n",
    "            af_rnd = random.randint(0, 4)   # activation function
index\n",
    "            \n",
    "        if (af_rnd == 0):\n",
    "            A1 = lrelu(Z1)\n",
    "            this_AF = \"L-ReLU\"\n",
    "        elif (af_rnd == 1):\n",
    "            A1 = relu(Z1)\n",
    "            this_AF = \"ReLU\"\n",
    "        elif (af_rnd == 2):\n",
    "            A1 = tanh(Z1)\n",
    "            this_AF = \"TanH\"\n",
    "        elif (af_rnd == 3):\n",
    "            A1 = sigmoid(Z1)\n",
    "            this_AF = \"Sigmoid\"\n",
    "        else:\n",
    "            A1 = softmax(Z1)\n",
```

```
    "            this_AF = \"Softmax\"\n",
    "        \n",
    "        af_indx = af_rnd\n",
    "\n",
    "    else:                                # The default activation
function chooser\n",
    "            \n",
    "        if (af_rnd == 0):\n",
    "            A1 = lrelu(Z1)\n",
    "            this_AF = \"L-ReLU\"\n",
    "        elif (af_rnd == 1):\n",
    "            A1 = relu(Z1)\n",
    "            this_AF = \"ReLU\"\n",
    "        elif (af_rnd == 2):\n",
    "            A1 = tanh(Z1)\n",
    "            this_AF = \"TanH\"\n",
    "        elif (af_rnd == 3):\n",
    "            A1 = sigmoid(Z1)\n",
    "            this_AF = \"Sigmoid\"\n",
    "        else:\n",
    "            A1 = softmax(Z1)\n",
    "            this_AF = \"Softmax\"\n",
    "        \n",
    "        af_indx = af_rnd\n",
    "\n",
    "    #output layer    \n",
    "    Z2 = np.dot(W2,A1) + b2\n",
    "    A2 = softmax(Z2)        # output layer activation function\n",
    "    ##COMPUTE LOSS\n",
    "    cost = compute_multiclass_loss(Y, A2)\n",
    "    loss.append(cost)\n",
    "    \n",
    "    ##BACK PASS\n",
    "    dZ2 = A2-Y\n",
    "    dW2 = (1./m) * np.dot(dZ2, A1.T)\n",
    "    db2 = (1./m) * np.sum(dZ2, axis=1, keepdims=True)\n",
    "    dA1 = np.dot(W2.T, dZ2)\n",
    "\n",
    "    # Hidden layer prime function\n",
    "    if (af_rnd == 0):\n",
    "        dZ1 = lrelu_back(dA1)\n",
    "        this_bAF = \"L-ReLU_back\"\n",
    "    elif (af_rnd == 1):\n",
    "        dZ1 = relu_back(dA1)\n",
    "        this_bAF = \"ReLU_back\"\n",
    "    elif (af_rnd == 2):\n",
    "        dZ1 = tanh_back(dA1)\n",
    "        this_bAF = \"TanH_back\"\n",
    "    elif (af_rnd == 3):\n",
    "        dZ1 = sigmoid_back(dA1)\n",
    "        this_bAF = \"Sigmoid_back\"\n",
    "    else:\n",
    "        dZ1 = softmax(dA1)\n",
    "        this_bAF = \"Softmax-back\"\n",
    "\n",
    "    dW1 = (1./m) * np.dot(dZ1, X.T)\n",
    "    db1 = (1./m) * np.sum(dZ1, axis=1, keepdims=True)\n",
```

```
    "\n",
    "    ##UPDATE WEIGHTS AND BAIS\n",
    "    learning_rate = int_learning_rate * 1/(1 + 0.01 * i)\n",
    "\n",
    "    W2 = W2 - learning_rate * dW2\n",
    "    b2 = b2 - learning_rate * db2\n",
    "    W1 = W1 - learning_rate * dW1\n",
    "    b1 = b1 - learning_rate * db1\n",
    "\n",
    "\n",
    "    predictions = np.argmax(A2, axis=0)\n",
    "    labels = np.argmax(Y_train, axis=0)\n",
    "\n",
    "    if (i % 5 == 0 and verbose_print == 1):\n",
    "        print(\"Epoch\", i, \"cost: \", cost, \": Train Accuracy:\",
accuracy_score(predictions, labels), \":\", this_AF)\n",
    "        loss.append(cost)\n",
    "        ac = accuracy_score(predictions, labels)\n",
    "        acc.append(ac)\n",
    "    elif (i % 5 == 0 and verbose_print == 0):\n",
    "        print(i, \",\", cost, \",\", accuracy_score(predictions,
labels), \",\", this_AF)\n",
    "        loss.append(cost)\n",
    "        ac = accuracy_score(predictions, labels)\n",
    "        acc.append(ac)\n",
    "\n",
    "print(\"Final cost:\", cost)\n",
    "end = time.time()\n",
    "print(\"Total Time\", end - start)\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "### BASELINE ###\n",
    "# Use this cell to select and produce activation function\n",
    "#   data for the entire run for baseline cost and accuracy data.\n",
    "# Set number of epochs (300 is good), initial learning rate,\n",
    "#   your choice of AF, and if you want verbose print output\n",
    "#   or CSV output for copy-and-past for Excel.\n",
    "\n",
    "# Loss function\n",
    "def compute_multiclass_loss(Y, Y_hat):\n",
    "    L = np.sqrt(((Y_hat - Y) ** 2).mean())\n",
    "\n",
    "    return L\n",
    "\n",
    "n_x = X_train.shape[0]\n",
    "\n",
    "# Value sets number of hidden layer nodes\n",
    "n_h = 64                                # 16\n",
    "\n",
    "# Set starting learning rate\n",
    "int_learning_rate = 0.1                 # 0.1\n",
```

```
    "\n",
    "# Randomize starting weight, or use random seed\n",
    "np.random.seed(1)\n",
    "W1 = np.random.randn(n_h, n_x)\n",
    "b1 = np.zeros((n_h, 1))\n",
    "W2 = np.random.randn(digits, n_h)\n",
    "b2 = np.zeros((digits, 1))\n",
    "\n",
    "\n",
    "X = X_train\n",
    "Y = Y_train\n",
    "loss = []\n",
    "xs = []\n",
    "acc = []\n",
    "\n",
    "m = Y.shape[1]\n",
    "start = time.time() \n",
    "\n",
    "# Set number of iterations\n",
    "epochs = 300\n",
    "\n",
    "# For verbose results, set to 1, else 0 for copy-and-paste output\n",
    "verbose_print = 0\n",
    "\n",
    "# Choose activation function index for baseline run\n",
    "#   0 = Leaky ReLU\n",
    "#   1 = ReLUP\n",
    "#   2 = TanH\n",
    "#   3 = Sigmoid\n",
    "#   4 = Softmax\n",
    "af_sel = 0\n",
    "    \n",
    "for i in range(epochs):\n",
    "    ##FOWARD PASS\n",
    "    #input/hidden layer\n",
    "    Z1 = np.dot(W1,X) + b1\n",
    "   \n",
    "    # Choose feed forward hidden layer activation function\n",
    "    # randomly if at least five iterations into training\n",
    "    # and cost is equal to or above 0.3\n",
    "    if (len(loss) > 5 and (acc[len(acc)-1] <= acc[len(acc)-2])):
#((len(loss) > 5) and (cost < .3))\n",
    "        \n",
    "        if (af_sel == 0):\n",
    "            A1 = lrelu(Z1)\n",
    "            this_AF = \"L-ReLU\"\n",
    "        elif (af_sel == 1):\n",
    "            A1 = relu(Z1)\n",
    "            this_AF = \"ReLU\"\n",
    "        elif (af_sel == 2):\n",
    "            A1 = tanh(Z1)\n",
    "            this_AF = \"TanH\"\n",
    "        elif (af_sel == 3):\n",
    "            A1 = sigmoid(Z1)\n",
    "            this_AF = \"Sigmoid\"\n",
    "        else:\n",
    "            A1 = softmax(Z1)\n",
```

```
    "            this_AF = \"Softmax\"\n",
    "        \n",
    "    # output layer    \n",
    "    Z2 = np.dot(W2,A1) + b2\n",
    "    A2 = softmax(Z2)        # output layer activation function\n",
    "    ##COMPUTE LOSS\n",
    "    cost = compute_multiclass_loss(Y, A2)\n",
    "    loss.append(cost)\n",
    "    \n",
    "    # BACK PASS\n",
    "    dZ2 = A2-Y\n",
    "    dW2 = (1./m) * np.dot(dZ2, A1.T)\n",
    "    db2 = (1./m) * np.sum(dZ2, axis=1, keepdims=True)\n",
    "    dA1 = np.dot(W2.T, dZ2)\n",
    "\n",
    "    # Hidden layer prime function\n",
    "    if (af_sel == 0):\n",
    "        dZ1 = lrelu_back(dA1)\n",
    "        this_bAF = \"L-ReLU_back\"\n",
    "    elif (af_sel == 1):\n",
    "        dZ1 = relu_back(dA1)\n",
    "        this_bAF = \"ReLU_back\"\n",
    "    elif (af_sel == 2):\n",
    "        dZ1 = tanh_back(dA1)\n",
    "        this_bAF = \"TanH_back\"\n",
    "    elif (af_sel == 3):\n",
    "        dZ1 = sigmoid_back(dA1)\n",
    "        this_bAF = \"Sigmoid_back\"\n",
    "    else:\n",
    "        dZ1 = softmax(dA1)\n",
    "        this_bAF = \"Softmax-back\"\n",
    "\n",
    "    dW1 = (1./m) * np.dot(dZ1, X.T)\n",
    "    db1 = (1./m) * np.sum(dZ1, axis=1, keepdims=True)\n",
    "\n",
    "    ## UPDATE WEIGHTS AND BAIS\n",
    "    learning_rate = int_learning_rate * 1/(1 + 0.01 * i)\n",
    "\n",
    "    W2 = W2 - learning_rate * dW2\n",
    "    b2 = b2 - learning_rate * db2\n",
    "    W1 = W1 - learning_rate * dW1\n",
    "    b1 = b1 - learning_rate * db1\n",
    "\n",
    "\n",
    "    predictions = np.argmax(A2, axis=0)\n",
    "    labels = np.argmax(Y_train, axis=0)\n",
    "\n",
    "    if (i % 5 == 0 and verbose_print == 1):\n",
    "        print(\"Epoch\", i, \"cost: \", cost, \": Train Accuracy:\", 
accuracy_score(predictions, labels), \":\", this_AF)\n",
    "        loss.append(cost)\n",
    "        ac = accuracy_score(predictions, labels)\n",
    "        acc.append(ac)\n",
    "    elif (i % 5 == 0 and verbose_print == 0):\n",
    "        print(i, \",\", cost, \",\", accuracy_score(predictions, 
labels), \",\", this_AF)\n",
    "        loss.append(cost)\n",
```

```
    "        ac = accuracy_score(predictions, labels)\n",
    "        acc.append(ac)\n",
    "\n",
    "print(\"Final cost:\", cost)\n",
    "end = time.time()\n",
    "print(\"Total Time\", end - start)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "#X_train = np.array(X_train)\n",
    "#X_train = X_train.astype('float32')\n",
    "#Z1 = np.matmul(W1, X_train) + b1\n",
    "#insert activation function\n",
    "#A1 = relu(Z1)\n",
    "#########\n",
    "#Z2 = np.matmul(W2, A1) + b2\n",
    "#A2 = softmax(Z2)\n",
    "\n",
    "#predictions = np.argmax(A2, axis=0)\n",
    "#labels = np.argmax(Y_train, axis=0)\n",
    "\n",
    "print(\"Train Accuracy\", accuracy_score(predictions, labels))"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "X_test = np.array(X_test)\n",
    "X_test = X_test.astype('float32')\n",
    "Z1 = np.matmul(W1, X_test) + b1\n",
    "#insert activation function\n",
    "A1 = relu(Z1)\n",
    "#########\n",
    "Z2 = np.matmul(W2, A1) + b2\n",
    "A2 = softmax(Z2)\n",
    "predictions = np.argmax(A2, axis=0)\n",
    "labels = np.argmax(Y_test, axis=0)\n",
    "\n",
    "print(\"Test Accuracy\", accuracy_score(predictions, labels))"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": true
   },
   "outputs": [],
   "source": [
```

```
    "import random\n",
    "\n",
    "af_rnd = 3\n",
    "af_indx = 3"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "while af_rnd == af_indx :\n",
    "    af_rnd = random.randint(0, 4)\n",
    "    \n",
    "print(af_indx, af_rnd)\n",
    " \n",
    "if (af_rnd == 0) :\n",
    "    print(\"Leaky ReLU\", af_rnd)\n",
    "elif (af_rnd == 1) :\n",
    "    print(\"ReLU\", af_rnd)\n",
    "elif (af_rnd == 2) :\n",
    "    print(\"TanH\", af_rnd)\n",
    "elif (af_rnd == 3) :\n",
    "    print(\"Sigmoid\", af_rnd)\n",
    "else :\n",
    "    print(\"Softmax\", af_rnd)\n",
    "    \n",
    "af_indx = af_rnd"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "print(cost, loss[i], loss[i-5], len(loss))\n",
    "acc_indx = len(acc) - 1\n",
    "print(ac, acc[len(acc)-1], acc[len(acc)-2], len(acc))"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "# Print randomize starting weights\n",
    "np.random.seed(1)\n",
    "W1 = np.random.randn(n_h, n_x)\n",
    "b1 = np.zeros((n_h, 1))\n",
    "W2 = np.random.randn(digits, n_h)\n",
    "b2 = np.zeros((digits, 1))\n",
    "\n",
    "print(\"W1: \", W1, \"b1: \", b1, \"W2: \", W2, \"b2: \", b2)"
   ]
```

```
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": [
    "print(A1)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": true
   },
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.6.8"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2
}
```

Appendix 3: R code of Two-sample Kolmogorov-Smirnov Test

```
# R Code for K-S Tests

# loading the required package

library(dgof)


# sample 1

# Leaky ReLU baseline

x1 <-
c(0.08705,0.153616667,0.128483333,0.117616667,0.11215,0.111983333,0.112483333,0.11238
3333,0.112383333,0.112383333,0.112383333,0.112366667,0.112366667,0.112366667,0.112366
667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.1123666
67,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.11236666
7,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667
,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,
0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0
.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.
112366667,0.112366667,0.112366667)

# ReLU baseline

x1 <-
c(0.08705,0.153616667,0.128483333,0.117616667,0.11215,0.111983333,0.112483333,0.11238
3333,0.112383333,0.112383333,0.112383333,0.112366667,0.112366667,0.112366667,0.112366
667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.1123666
67,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.11236666
7,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667
,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,
0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0
.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.
112366667,0.112366667,0.112366667)

# Sigmoid baseline

x1 <-
c(0.08445,0.099433333,0.11695,0.110033333,0.102066667,0.094283333,0.091233333,0.08833
3333,0.0882,0.089383333,0.08915,0.089166667,0.089216667,0.089566667,0.0894,0.104366667
,0.103083333,0.102666667,0.1025,0.102366667,0.102233333,0.102166667,0.102183333,0.1021
83333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.10218
3333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183
333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.1021833
33,0.102183333,0.102183333,0.102183333,0.102183333,0.112366667,0.112366667,0.11236666
```

7,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667
,0.112366667)

Softmax baseline

x1 <-
c(0.083966667,0.084266667,0.084433333,0.08515,0.0855,0.086416667,0.0873,0.08835,0.0897,
0.090433333,0.091066667,0.091583333,0.092466667,0.093516667,0.094083333,0.095016667,0
.095166667,0.095816667,0.096683333,0.0973,0.097933333,0.098283333,0.098983333,0.10006
6667,0.10105,0.102316667,0.103833333,0.105433333,0.1087,0.11025,0.111283333,0.11195,0.1
12633333,0.113283333,0.114033333,0.114816667,0.11565,0.116633333,0.11955,0.119883333,
0.120216667,0.120633333,0.121216667,0.121583333,0.121766667,0.122033333,0.1223,0.1226
33333,0.122816667,0.123183333,0.123666667,0.123933333,0.124083333,0.12445,0.124533333
,0.124633333,0.124766667,0.124933333,0.12525,0.125483333)

TanH baseline

x1 <-
c(0.113733333,0.130666667,0.096066667,0.106266667,0.107783333,0.110716667,0.11188333
3,0.11235,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.11
2366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112
366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.1123
66667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.11236
6667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366
667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.1123666
67,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.11236666
7,0.112366667,0.112366667,0.112366667)

sample 2

Leaky ReLU random 1

x2 <-
c(0.086816667,0.1516,0.128216667,0.135716667,0.141,0.142383333,0.143683333,0.1513,0.17
645,0.186266667,0.193883333,0.204633333,0.211416667,0.215933333,0.21785,0.219816667,0.
221016667,0.221433333,0.221383333,0.216966667,0.112366667,0.2077,0.212916667,0.2157,0.
218466667,0.219983333,0.2216,0.223316667,0.224733333,0.225816667,0.226183333,0.22675,
0.226816667,0.227833333,0.228366667,0.229183333,0.230733333,0.231283333,0.231466667,0
.2315,0.232266667,0.2324,0.232516667,0.233516667,0.2352,0.235833333,0.23635,0.23681666
7,0.237316667,0.237316667,0.234033333,0.112366667,0.104416667,0.112366667,0.112366667
,0.112366667,0.112366667,0.627216667,0.72695,0.729916667)

Leaky ReLU random 2

```
x2 <-
c(0.086816667,0.169766667,0.1178,0.179483333,0.264883333,0.3359,0.392283333,0.4364666
67,0.474966667,0.507633333,0.535383333,0.5589,0.578733333,0.596416667,0.610016667,0.62
325,0.63475,0.6441,0.653116667,0.661033333,0.6689,0.676216667,0.682683333,0.689316667,
0.69515,0.70025,0.704916667,0.709333333,0.7132,0.71665,0.719933333,0.72315,0.726466667,
0.729266667,0.73195,0.734933333,0.737516667,0.739716667,0.742133333,0.744516667,0.746
583333,0.7485,0.750383333,0.752033333,0.75385,0.755216667,0.756816667,0.7583,0.7597,0.7
61066667,0.762166667,0.763333333,0.764466667,0.765833333,0.76705,0.768133333,0.768966
667,0.76995,0.771066667,0.771816667)


# ReLU random 1

x2 <-
c(0.08705,0.171333333,0.118916667,0.1098,0.110783333,0.11175,0.112516667,0.113416667,0
.1138,0.114616667,0.115683333,0.116583333,0.117483333,0.118183333,0.1188,0.119466667,0
.120516667,0.1213,0.122083333,0.12285,0.123566667,0.12405,0.1248,0.125433333,0.12625,0.
12705,0.128033333,0.128566667,0.129266667,0.130283333,0.131183333,0.1328,0.13645,0.137
25,0.138283333,0.139083333,0.139766667,0.140683333,0.14145,0.142033333,0.1426,0.143166
667,0.143766667,0.14425,0.14455,0.1451,0.145483333,0.14595,0.146533333,0.147016667,0.14
7416667,0.14775,0.148416667,0.148783333,0.149166667,0.14945,0.149683333,0.150133333,0.
150866667,0.15145)

# ReLU random 2

x2 <-
c(0.08705,0.171333333,0.118916667,0.208766667,0.29,0.357816667,0.411933333,0.45475,0.4
90116667,0.521916667,0.547883333,0.569333333,0.588416667,0.6038,0.617683333,0.6297833
33,0.640233333,0.649633333,0.658133333,0.666183333,0.673916667,0.680316667,0.6871,0.69
375,0.699016667,0.7034,0.7082,0.712216667,0.715966667,0.719233333,0.722783333,0.7258,0.
729066667,0.731483333,0.734416667,0.737266667,0.739533333,0.742,0.74435,0.746566667,0.
748383333,0.750433333,0.752266667,0.75395,0.7553,0.756866667,0.758316667,0.759816667,
0.761066667,0.7623,0.76355,0.76495,0.766133333,0.7672,0.7684,0.7693,0.77025,0.771116667,
0.772116667,0.77285)

# ReLU minus L-ReLU random 1

x2 <-
c(0.08705,0.171333333,0.118916667,0.10215,0.106333333,0.10965,0.11145,0.112266667,0.11
2383333,0.112366667,0.097533333,0.112366667,0.112366667,0.112366667,0.112366667,0.112
366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.1123
66667,0.112366667,0.174933333,0.216566667,0.22575,0.229016667,0.231816667,0.233266667
,0.234783333,0.2359,0.23655,0.238333333,0.244833333,0.2483,0.25275,0.277133333,0.2771,0.
112366667,0.28045,0.281066667,0.282233333,0.282733333,0.283333333,0.28375,0.284416667
,0.2844,0.285433333,0.285716667,0.286216667,0.286216667,0.112366667,0.112366667,0.1123
66667,0.112366667,0.112366667,0.2899,0.290116667,0.290616667)

# ReLU minus L-ReLU random 2
```

x2 <-
c(0.08705,0.111683333,0.11265,0.113633333,0.114166667,0.114733333,0.1154,0.116133333,0
.117266667,0.117883333,0.118866667,0.119766667,0.120483333,0.12135,0.122416667,0.1234
33333,0.124766667,0.125766667,0.126983333,0.12855,0.1299,0.131616667,0.134183333,0.136
983333,0.137066667,0.137666667,0.1385,0.139216667,0.139833333,0.140383333,0.140983333
,0.141333333,0.141833333,0.142166667,0.1426,0.142916667,0.1434,0.143316667,0.166716667
,0.161016667,0.124466667,0.118883333,0.130016667,0.1255,0.104183333,0.151983333,0.1524
,0.152883333,0.153266667,0.153633333,0.154166667,0.154566667,0.1547,0.155083333,0.1555
66667,0.156016667,0.156233333,0.1565,0.156616667,0.157116667)

Sigmoid random 1

x2 <-
c(0.08445,0.099433333,0.11695,0.110033333,0.102066667,0.094283333,0.091233333,0.08833
3333,0.0882,0.089383333,0.08915,0.089166667,0.089216667,0.089566667,0.0894,0.104366667
,0.103083333,0.102666667,0.1025,0.102366667,0.102233333,0.102166667,0.102183333,0.1021
83333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.10218
3333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183
333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.102183333,0.1021833
33,0.102183333,0.102183333,0.102183333,0.102183333,0.112366667,0.112366667,0.11236666
7,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667
,0.112366667)

Sigmoid random 2

x2 <-
c(0.08445,0.148033333,0.109983333,0.10155,0.138133333,0.190483333,0.256283333,0.32605,
0.383666667,0.4285,0.468166667,0.503733333,0.531616667,0.554633333,0.573466667,0.5908
83333,0.6048,0.618166667,0.629516667,0.6398,0.648983333,0.656766667,0.6644,0.670566667
,0.67715,0.6837,0.68965,0.6947,0.69955,0.704216667,0.708,0.711716667,0.71515,0.718516667
,0.7218,0.724566667,0.727483333,0.729866667,0.732616667,0.73525,0.7376,0.739816667,0.74
19,0.7441,0.7459,0.747816667,0.749533333,0.75135,0.75275,0.754366667,0.75595,0.75735,0.7
58533333,0.759683333,0.760883333,0.762033333,0.76315,0.764616667,0.765583333,0.766733
333)

Sigmoid minus L-ReLU random 1

x2 <-
c(0.08445,0.086683333,0.08675,0.08675,0.087316667,0.087166667,0.130566667,0.1002,0.096,
0.102683333,0.103783333,0.10415,0.10345,0.10295,0.10945,0.110516667,0.1111,0.111416667,
0.111716667,0.111916667,0.112066667,0.112216667,0.112316667,0.112333333,0.11235,0.112
366667,0.112366667,0.112366667,0.131383333,0.132283333,0.133283333,0.133783333,0.1348
33333,0.13545,0.13605,0.136916667,0.1381,0.138516667,0.13945,0.140183333,0.140833333,0.
141533333,0.142316667,0.14255,0.143083333,0.143916667,0.144766667,0.14525,0.145716667

```
,0.1464,0.1469,0.14745,0.14795,0.148566667,0.149133333,0.149616667,0.150333333,0.1507,0.
150933333,0.151233333)

# Sigmoid minus L-ReLU random 2

x2 <-
c(0.08445,0.100833333,0.127233333,0.11215,0.0922,0.107633333,0.1036,0.10295,0.1022,0.13
2966667,0.133866667,0.134933333,0.135933333,0.1372,0.138116667,0.138933333,0.13993333
3,0.140866667,0.14185,0.14285,0.143933333,0.1449,0.146583333,0.148333333,0.153916667,0.
156633333,0.159,0.1608,0.162566667,0.164,0.16545,0.1704,0.171483333,0.172183333,0.17306
6667,0.173866667,0.17485,0.17555,0.1763,0.1771,0.1777,0.178266667,0.178483333,0.1789166
67,0.179816667,0.180583333,0.180966667,0.181733333,0.182566667,0.183233333,0.18356666
7,0.183983333,0.18445,0.1849,0.18545,0.185883333,0.186283333,0.186866667,0.187283333,0.
187716667)

# Softmax random 1

x2 <-
c(0.083966667,0.084133333,0.100116667,0.11685,0.110866667,0.1056,0.1242,0.151983333,0.
195533333,0.259533333,0.3254,0.386883333,0.435383333,0.474116667,0.5053,0.530733333,0.
55185,0.569633333,0.58545,0.599616667,0.6114,0.6232,0.632433333,0.641383333,0.64925,0.6
5705,0.663866667,0.6702,0.676766667,0.68175,0.68705,0.692283333,0.6967,0.7009,0.7047833
33,0.70835,0.7116,0.715016667,0.717916667,0.720866667,0.7237,0.72655,0.72915,0.73135,0.7
33383333,0.735483333,0.737466667,0.739633333,0.74175,0.7439,0.74555,0.747083333,0.7487
,0.750316667,0.751416667,0.753216667,0.754833333,0.756216667,0.757316667)

# Softmax random 2

x2 <-
c(0.083966667,0.105816667,0.1119,0.104033333,0.110183333,0.11155,0.112266667,0.112383
333,0.112366667,0.204166667,0.33685,0.4797,0.52125,0.566716667,0.594283333,0.608666667
,0.6212,0.632183333,0.64195,0.651033333,0.658883333,0.666066667,0.67305,0.6797,0.686,0.6
91116667,0.696766667,0.701416667,0.706,0.7106,0.714066667,0.717683333,0.721,0.72438333
3,0.727183333,0.729833333,0.7328,0.734866667,0.73695,0.738966667,0.740983333,0.74295,0.
74505,0.746783333,0.748716667,0.750066667,0.751833333,0.75355,0.755,0.756433333,0.758,
0.759233333,0.76025,0.761466667,0.7625,0.763633333,0.764783333,0.765983333,0.7669,0.76
8016667)

# SOftmax minus L-ReLU random 1

x2 <-
c(0.083966667,0.084133333,0.100116667,0.11685,0.110866667,0.1152,0.097416667,0.101516
667,0.102516667,0.1036,0.103016667,0.107016667,0.107316667,0.107716667,0.108316667,0.1
09133333,0.10955,0.109783333,0.10975,0.098566667,0.103916667,0.111516667,0.112083333,
0.112333333,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.147733333,0
```

.148316667,0.1494,0.150283333,0.151016667,0.151733333,0.1524,0.153233333,0.15425,0.155383333,0.15695,0.1582,0.158916667,0.159433333,0.160166667,0.160683333,0.161333333,0.16185,0.162283333,0.162983333,0.16415,0.164366667,0.16505,0.165716667,0.1665,0.16685,0.16755,0.168183333,0.16855,0.169066667,0.169566667)

Softmax minus L-ReLU random 2

x2 <-
c(0.083966667,0.084133333,0.100116667,0.11685,0.110866667,0.0961,0.099016667,0.09775,0.103216667,0.103533333,0.103416667,0.102683333,0.102683333,0.102283333,0.102316667,0.102283333,0.102183333,0.09865,0.102183333,0.102183333,0.104416667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.135633333,0.136933333,0.138366667,0.1392,0.140666667,0.14245,0.14475,0.14555,0.146916667,0.148216667,0.14895,0.14985,0.15055,0.151266667,0.15195,0.152883333,0.153666667,0.154133333,0.1548,0.155816667,0.156766667,0.1575,0.15835,0.159266667,0.160433333,0.161183333,0.161683333,0.162366667,0.162916667,0.163533333,0.163933333,0.16525,0.166083333)

TanH random 1

x2 <-
c(0.113733333,0.08975,0.0971,0.1154,0.114133333,0.110983333,0.099166667,0.09925,0.099216667,0.112366667,0.112366667,0.163466667,0.172683333,0.182583333,0.1908,0.198316667,0.204883333,0.21105,0.2169,0.221683333,0.22665,0.2314,0.235583333,0.2392,0.2431,0.246283333,0.249983333,0.2539,0.257316667,0.272933333,0.2908,0.292466667,0.294183333,0.295666667,0.297783333,0.29945,0.300316667,0.30135,0.301666667,0.30275,0.3037,0.305433333,0.306083333,0.306533333,0.307416667,0.3082,0.30895,0.310033333,0.310366667,0.311366667,0.312116667,0.3132,0.313866667,0.314333333,0.314883333,0.3153,0.315816667,0.3166,0.317433333,0.318216667)

TanH random 2

x2 <-
c(0.113733333,0.0856,0.098133333,0.114683333,0.153683333,0.211016667,0.281716667,0.343466667,0.392316667,0.44045,0.481183333,0.513416667,0.540216667,0.56125,0.58045,0.5961,0.610483333,0.622883333,0.633283333,0.6429,0.6522,0.659916667,0.6672,0.67415,0.680716667,0.6864,0.691933333,0.6976,0.701966667,0.706516667,0.71025,0.71375,0.717183333,0.720266667,0.723166667,0.72635,0.72895,0.731683333,0.734216667,0.736516667,0.738816667,0.741066667,0.743066667,0.745233333,0.747183333,0.7488,0.750316667,0.75215,0.753866667,0.755383333,0.756716667,0.757816667,0.7591,0.760466667,0.7619,0.763083333,0.764116667,0.765133333,0.7661,0.766983333)

TanH minus L-ReLU random 1

x2 <-
c(0.113733333,0.099216667,0.095966667,0.093633333,0.096683333,0.102066667,0.10981666

```
7,0.111416667,0.112183333,0.112366667,0.112366667,0.09915,0.112366667,0.112366667,0.09
915,0.09915,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.112366667,0.
112366667,0.132316667,0.133833333,0.135266667,0.136733333,0.138483333,0.14005,0.14153
3333,0.143216667,0.14435,0.1463,0.147983333,0.14945,0.15105,0.15225,0.153966667,0.1554,
0.15685,0.1581,0.15935,0.16075,0.1624,0.16405,0.165866667,0.168283333,0.172616667,0.175,
0.176616667,0.178033333,0.178616667,0.179416667,0.180466667,0.181416667,0.181933333,0
.182966667,0.18375,0.184516667,0.185183333)

# TanH minus L-ReLU random 2

x2 <-
c(0.113733333,0.0856,0.102983333,0.0951,0.08885,0.10335,0.108883333,0.111083333,0.1121
5,0.11235,0.112366667,0.112366667,0.098633333,0.102183333,0.102183333,0.1116,0.1118166
67,0.11555,0.116116667,0.11655,0.116866667,0.116916667,0.117116667,0.1172,0.1174,0.1173
5,0.112366667,0.112366667,0.11925,0.11935,0.11945,0.11945,0.120116667,0.120133333,0.120
25,0.120516667,0.120683333,0.120716667,0.120716667,0.098716667,0.098716667,0.09871666
7,0.098716667,0.124433333,0.124416667,0.112366667,0.098716667,0.12645,0.126533333,0.12
6683333,0.12665,0.112366667,0.098716667,0.098716667,0.098716667,0.098716667,0.0987166
67,0.098716667,0.112366667,0.112366667)


# plotting the result

# visualization

plot(ecdf(x1),

    xlim = range(c(x1, x2)),

    col = "blue")

plot(ecdf(x2),

    add = TRUE,

    lty = "dashed",

    col = "red")


# performing the K-S

# Test on x1 and x2

ks.test(x1, x2)
```

Appendix 4: Training results

Baseline – Leaky ReLU				Baseline - ReLU			
Epoch	Cost	Train Acc	Act Funct	Epoch	Cost	Train Acc	Act Funct
0	0.423105	0.086817	L-ReLU	0	0.423081	0.08705	ReLU
5	0.395748	0.157433	L-ReLU	5	0.396533	0.153617	ReLU
10	0.354247	0.138883	L-ReLU	10	0.346294	0.128483	ReLU
15	0.3133	0.165133	L-ReLU	15	0.306873	0.117617	ReLU
20	0.295669	0.23015	L-ReLU	20	0.300918	0.11215	ReLU
25	0.28656	0.3314	L-ReLU	25	0.300118	0.111983	ReLU
30	0.279657	0.411917	L-ReLU	30	0.300001	0.112483	ReLU
35	0.273571	0.469067	L-ReLU	35	0.299978	0.112383	ReLU
40	0.268393	0.5149	L-ReLU	40	0.299971	0.112383	ReLU
45	0.26416	0.548883	L-ReLU	45	0.299972	0.112383	ReLU
50	0.260492	0.573417	L-ReLU	50	0.299971	0.112383	ReLU
55	0.257098	0.5931	L-ReLU	55	0.299971	0.112367	ReLU
60	0.253895	0.608483	L-ReLU	60	0.29997	0.112367	ReLU
65	0.250861	0.622417	L-ReLU	65	0.299968	0.112367	ReLU
70	0.247984	0.6345	L-ReLU	70	0.299968	0.112367	ReLU
75	0.245253	0.6443	L-ReLU	75	0.299967	0.112367	ReLU
80	0.242658	0.65495	L-ReLU	80	0.299966	0.112367	ReLU
85	0.240193	0.6638	L-ReLU	85	0.299965	0.112367	ReLU
90	0.237848	0.67175	L-ReLU	90	0.299964	0.112367	ReLU
95	0.235618	0.678967	L-ReLU	95	0.299964	0.112367	ReLU
100	0.233496	0.6859	L-ReLU	100	0.299963	0.112367	ReLU
105	0.231474	0.69235	L-ReLU	105	0.299963	0.112367	ReLU
110	0.229547	0.697867	L-ReLU	110	0.299962	0.112367	ReLU
115	0.227709	0.702767	L-ReLU	115	0.299962	0.112367	ReLU
120	0.225955	0.707467	L-ReLU	120	0.299961	0.112367	ReLU
125	0.22428	0.71155	L-ReLU	125	0.299961	0.112367	ReLU
130	0.222679	0.715417	L-ReLU	130	0.29996	0.112367	ReLU
135	0.221148	0.719283	L-ReLU	135	0.29996	0.112367	ReLU
140	0.219684	0.72295	L-ReLU	140	0.29996	0.112367	ReLU
145	0.218281	0.725867	L-ReLU	145	0.299959	0.112367	ReLU
150	0.216938	0.729133	L-ReLU	150	0.299959	0.112367	ReLU
155	0.21565	0.731983	L-ReLU	155	0.299959	0.112367	ReLU
160	0.214414	0.73465	L-ReLU	160	0.299959	0.112367	ReLU
165	0.213227	0.737067	L-ReLU	165	0.299958	0.112367	ReLU
170	0.212087	0.739817	L-ReLU	170	0.299958	0.112367	ReLU
175	0.210991	0.742167	L-ReLU	175	0.299958	0.112367	ReLU
180	0.209937	0.744317	L-ReLU	180	0.299958	0.112367	ReLU
185	0.208923	0.7466	L-ReLU	185	0.299957	0.112367	ReLU
190	0.207946	0.748933	L-ReLU	190	0.299957	0.112367	ReLU

195	0.207005	0.751017	L-ReLU
200	0.206097	0.752817	L-ReLU
205	0.205222	0.754433	L-ReLU
210	0.204377	0.756083	L-ReLU
215	0.203562	0.757467	L-ReLU
220	0.202774	0.758867	L-ReLU
225	0.202012	0.7607	L-ReLU
230	0.201275	0.762083	L-ReLU
235	0.200562	0.763417	L-ReLU
240	0.199872	0.76465	L-ReLU
245	0.199203	0.765817	L-ReLU
250	0.198555	0.7669	L-ReLU
255	0.197927	0.768067	L-ReLU
260	0.197318	0.769017	L-ReLU
265	0.196727	0.770317	L-ReLU
270	0.196154	0.77135	L-ReLU
275	0.195597	0.772067	L-ReLU
280	0.195056	0.77265	L-ReLU
285	0.19453	0.773417	L-ReLU
290	0.194018	0.774217	L-ReLU
295	0.193521	0.774767	L-ReLU

195	0.299957	0.112367	ReLU
200	0.299957	0.112367	ReLU
205	0.299957	0.112367	ReLU
210	0.299957	0.112367	ReLU
215	0.299956	0.112367	ReLU
220	0.299956	0.112367	ReLU
225	0.299956	0.112367	ReLU
230	0.299956	0.112367	ReLU
235	0.299956	0.112367	ReLU
240	0.299956	0.112367	ReLU
245	0.299956	0.112367	ReLU
250	0.299956	0.112367	ReLU
255	0.299956	0.112367	ReLU
260	0.299955	0.112367	ReLU
265	0.299955	0.112367	ReLU
270	0.299955	0.112367	ReLU
275	0.299955	0.112367	ReLU
280	0.299955	0.112367	ReLU
285	0.299955	0.112367	ReLU
290	0.299955	0.112367	ReLU
295	0.299955	0.112367	ReLU

Baseline - Sigmoid			
Epoch	Cost	Train Acc	Act Funct
0	0.389802	0.08445	Sigmoid
5	0.367211	0.099433	Sigmoid
10	0.35641	0.11695	Sigmoid
15	0.347985	0.110033	Sigmoid
20	0.336465	0.102067	Sigmoid
25	0.324407	0.094283	Sigmoid
30	0.314465	0.091233	Sigmoid
35	0.307775	0.088333	Sigmoid
40	0.304123	0.0882	Sigmoid
45	0.302183	0.089383	Sigmoid
50	0.301177	0.08915	Sigmoid
55	0.300687	0.089167	Sigmoid
60	0.300448	0.089217	Sigmoid
65	0.300327	0.089567	Sigmoid
70	0.30026	0.0894	Sigmoid
75	0.300218	0.104367	Sigmoid
80	0.300188	0.103083	Sigmoid
85	0.300165	0.102667	Sigmoid

Baseline - Softmax			
Epoch	Cost	Train Acc	Act Funct
0	0.309923	0.083967	Softmax
5	0.309614	0.084267	Softmax
10	0.309336	0.084433	Softmax
15	0.309083	0.08515	Softmax
20	0.308851	0.0855	Softmax
25	0.308637	0.086417	Softmax
30	0.308438	0.0873	Softmax
35	0.308252	0.08835	Softmax
40	0.308078	0.0897	Softmax
45	0.307913	0.090433	Softmax
50	0.307758	0.091067	Softmax
55	0.307611	0.091583	Softmax
60	0.30747	0.092467	Softmax
65	0.307337	0.093517	Softmax
70	0.307209	0.094083	Softmax
75	0.307087	0.095017	Softmax
80	0.306969	0.095167	Softmax
85	0.306857	0.095817	Softmax

90	0.300145	0.1025	Sigmoid	90	0.306748	0.096683	Softmax
95	0.300128	0.102367	Sigmoid	95	0.306644	0.0973	Softmax
100	0.300114	0.102233	Sigmoid	100	0.306543	0.097933	Softmax
105	0.300102	0.102167	Sigmoid	105	0.306445	0.098283	Softmax
110	0.300093	0.102183	Sigmoid	110	0.306351	0.098983	Softmax
115	0.300085	0.102183	Sigmoid	115	0.30626	0.100067	Softmax
120	0.300078	0.102183	Sigmoid	120	0.306171	0.10105	Softmax
125	0.300072	0.102183	Sigmoid	125	0.306085	0.102317	Softmax
130	0.300066	0.102183	Sigmoid	130	0.306002	0.103833	Softmax
135	0.300061	0.102183	Sigmoid	135	0.305921	0.105433	Softmax
140	0.300056	0.102183	Sigmoid	140	0.305842	0.1087	Softmax
145	0.300052	0.102183	Sigmoid	145	0.305766	0.11025	Softmax
150	0.300048	0.102183	Sigmoid	150	0.305691	0.111283	Softmax
155	0.300044	0.102183	Sigmoid	155	0.305618	0.11195	Softmax
160	0.30004	0.102183	Sigmoid	160	0.305547	0.112633	Softmax
165	0.300037	0.102183	Sigmoid	165	0.305478	0.113283	Softmax
170	0.300034	0.102183	Sigmoid	170	0.30541	0.114033	Softmax
175	0.300031	0.102183	Sigmoid	175	0.305344	0.114817	Softmax
180	0.300028	0.102183	Sigmoid	180	0.30528	0.11565	Softmax
185	0.300025	0.102183	Sigmoid	185	0.305217	0.116633	Softmax
190	0.300022	0.102183	Sigmoid	190	0.305155	0.11955	Softmax
195	0.30002	0.102183	Sigmoid	195	0.305095	0.119883	Softmax
200	0.300018	0.102183	Sigmoid	200	0.305036	0.120217	Softmax
205	0.300016	0.102183	Sigmoid	205	0.304978	0.120633	Softmax
210	0.300014	0.102183	Sigmoid	210	0.304921	0.121217	Softmax
215	0.300012	0.102183	Sigmoid	215	0.304865	0.121583	Softmax
220	0.30001	0.102183	Sigmoid	220	0.304811	0.121767	Softmax
225	0.300008	0.102183	Sigmoid	225	0.304757	0.122033	Softmax
230	0.300006	0.102183	Sigmoid	230	0.304705	0.1223	Softmax
235	0.300005	0.102183	Sigmoid	235	0.304653	0.122633	Softmax
240	0.300003	0.102183	Sigmoid	240	0.304603	0.122817	Softmax
245	0.300001	0.112367	Sigmoid	245	0.304553	0.123183	Softmax
250	0.3	0.112367	Sigmoid	250	0.304504	0.123667	Softmax
255	0.299999	0.112367	Sigmoid	255	0.304456	0.123933	Softmax
260	0.299997	0.112367	Sigmoid	260	0.304409	0.124083	Softmax
265	0.299996	0.112367	Sigmoid	265	0.304363	0.12445	Softmax
270	0.299995	0.112367	Sigmoid	270	0.304317	0.124533	Softmax
275	0.299994	0.112367	Sigmoid	275	0.304272	0.124633	Softmax
280	0.299993	0.112367	Sigmoid	280	0.304228	0.124767	Softmax
285	0.299992	0.112367	Sigmoid	285	0.304184	0.124933	Softmax
290	0.299991	0.112367	Sigmoid	290	0.304142	0.12525	Softmax
295	0.29999	0.112367	Sigmoid	295	0.3041	0.125483	Softmax

Baseline - TanH				Leaky ReLU – 1st random			
Epoch	Cost	Train Acc	Act Funct	Epoch	Cost	Train Acc	Act Funct
0	0.390736	0.113733	TanH	0	0.423105	0.086817	L-ReLU
5	0.387043	0.130667	TanH	5	0.396981	0.1516	ReLU
10	0.361815	0.096067	TanH	10	0.346363	0.128217	ReLU
15	0.310578	0.106267	TanH	15	0.300922	0.135717	Softmax
20	0.301881	0.107783	TanH	20	0.299773	0.141	Softmax
25	0.300134	0.110717	TanH	25	0.298809	0.142383	Softmax
30	0.29998	0.111883	TanH	30	0.297943	0.143683	Softmax
35	0.299957	0.11235	TanH	35	0.297168	0.1513	Softmax
40	0.299953	0.112367	TanH	40	0.296491	0.17645	Softmax
45	0.299952	0.112367	TanH	45	0.295919	0.186267	Softmax
50	0.299952	0.112367	TanH	50	0.295446	0.193883	Softmax
55	0.299952	0.112367	TanH	55	0.295048	0.204633	Softmax
60	0.299952	0.112367	TanH	60	0.294714	0.211417	Softmax
65	0.299952	0.112367	TanH	65	0.294428	0.215933	Softmax
70	0.299952	0.112367	TanH	70	0.29419	0.21785	Softmax
75	0.299952	0.112367	TanH	75	0.294	0.219817	Softmax
80	0.299952	0.112367	TanH	80	0.293845	0.221017	Softmax
85	0.299952	0.112367	TanH	85	0.293717	0.221433	Softmax
90	0.299952	0.112367	TanH	90	0.293611	0.221383	Softmax
95	0.299952	0.112367	TanH	95	0.293738	0.216967	Softmax
100	0.299952	0.112367	TanH	100	0.299983	0.112367	Sigmoid
105	0.299952	0.112367	TanH	105	0.294159	0.2077	Softmax
110	0.299952	0.112367	TanH	110	0.293817	0.212917	Softmax
115	0.299952	0.112367	TanH	115	0.293573	0.2157	Softmax
120	0.299952	0.112367	TanH	120	0.293387	0.218467	Softmax
125	0.299952	0.112367	TanH	125	0.293238	0.219983	Softmax
130	0.299952	0.112367	TanH	130	0.293121	0.2216	Softmax
135	0.299952	0.112367	TanH	135	0.293026	0.223317	Softmax
140	0.299952	0.112367	TanH	140	0.292947	0.224733	Softmax
145	0.299952	0.112367	TanH	145	0.29288	0.225817	Softmax
150	0.299952	0.112367	TanH	150	0.29282	0.226183	Softmax
155	0.299952	0.112367	TanH	155	0.292765	0.22675	Softmax
160	0.299952	0.112367	TanH	160	0.292716	0.226817	Softmax
165	0.299952	0.112367	TanH	165	0.292671	0.227833	Softmax
170	0.299952	0.112367	TanH	170	0.292629	0.228367	Softmax
175	0.299952	0.112367	TanH	175	0.292591	0.229183	Softmax
180	0.299952	0.112367	TanH	180	0.292556	0.230733	Softmax
185	0.299952	0.112367	TanH	185	0.292529	0.231283	Softmax
190	0.299952	0.112367	TanH	190	0.292504	0.231467	Softmax
195	0.299952	0.112367	TanH	195	0.292478	0.2315	Softmax
200	0.299952	0.112367	TanH	200	0.292449	0.232267	Softmax
205	0.299952	0.112367	TanH	205	0.292419	0.2324	Softmax

210	0.299952	0.112367	TanH	210	0.292389	0.232517	Softmax
215	0.299952	0.112367	TanH	215	0.29236	0.233517	Softmax
220	0.299952	0.112367	TanH	220	0.29233	0.2352	Softmax
225	0.299952	0.112367	TanH	225	0.292302	0.235833	Softmax
230	0.299952	0.112367	TanH	230	0.292272	0.23635	Softmax
235	0.299952	0.112367	TanH	235	0.292243	0.236817	Softmax
240	0.299952	0.112367	TanH	240	0.292209	0.237317	Softmax
245	0.299952	0.112367	TanH	245	0.292185	0.237317	Softmax
250	0.299952	0.112367	TanH	250	0.292279	0.234033	Softmax
255	0.299952	0.112367	TanH	255	0.300062	0.112367	ReLU
260	0.299952	0.112367	TanH	260	0.307578	0.104417	TanH
265	0.299952	0.112367	TanH	265	0.300066	0.112367	Sigmoid
270	0.299952	0.112367	TanH	270	0.300063	0.112367	Sigmoid
275	0.299952	0.112367	TanH	275	0.300065	0.112367	ReLU
280	0.299952	0.112367	TanH	280	0.300065	0.112367	Sigmoid
285	0.299952	0.112367	TanH	285	0.224359	0.627217	L-ReLU
290	0.299952	0.112367	TanH	290	0.199368	0.72695	L-ReLU
295	0.299952	0.112367	TanH	295	0.198297	0.729917	L-ReLU

Leaky ReLU – 2nd random				ReLU – 1st random			
Epoch	Cost	Train Acc	Act Funct	Epoch	Cost	Train Acc	Act Funct
0	0.423105	0.086817	L-ReLU	0	0.423081	0.08705	ReLU
5	0.378845	0.169767	TanH	5	0.378508	0.171333	TanH
10	0.337692	0.1178	TanH	10	0.337886	0.118917	TanH
15	0.305091	0.179483	L-ReLU	15	0.306261	0.1098	Softmax
20	0.293226	0.264883	L-ReLU	20	0.306005	0.110783	Softmax
25	0.287536	0.3359	L-ReLU	25	0.305771	0.11175	Softmax
30	0.282766	0.392283	L-ReLU	30	0.305555	0.112517	Softmax
35	0.278509	0.436467	L-ReLU	35	0.305356	0.113417	Softmax
40	0.274544	0.474967	L-ReLU	40	0.30517	0.1138	Softmax
45	0.270759	0.507633	L-ReLU	45	0.304996	0.114617	Softmax
50	0.267123	0.535383	L-ReLU	50	0.304832	0.115683	Softmax
55	0.263639	0.5589	L-ReLU	55	0.304678	0.116583	Softmax
60	0.260309	0.578733	L-ReLU	60	0.304533	0.117483	Softmax
65	0.25713	0.596417	L-ReLU	65	0.304395	0.118183	Softmax
70	0.254098	0.610017	L-ReLU	70	0.304263	0.1188	Softmax
75	0.251208	0.62325	L-ReLU	75	0.304138	0.119467	Softmax
80	0.248455	0.63475	L-ReLU	80	0.304019	0.120517	Softmax
85	0.245831	0.6441	L-ReLU	85	0.303904	0.1213	Softmax
90	0.243329	0.653117	L-ReLU	90	0.303795	0.122083	Softmax
95	0.240945	0.661033	L-ReLU	95	0.303689	0.12285	Softmax
100	0.238671	0.6689	L-ReLU	100	0.303588	0.123567	Softmax

105	0.236501	0.676217	L-ReLU	105	0.303491	0.12405	Softmax
110	0.234431	0.682683	L-ReLU	110	0.303397	0.1248	Softmax
115	0.232455	0.689317	L-ReLU	115	0.303306	0.125433	Softmax
120	0.230567	0.69515	L-ReLU	120	0.303219	0.12625	Softmax
125	0.228763	0.70025	L-ReLU	125	0.303134	0.12705	Softmax
130	0.227038	0.704917	L-ReLU	130	0.303052	0.128033	Softmax
135	0.225387	0.709333	L-ReLU	135	0.302972	0.128567	Softmax
140	0.223807	0.7132	L-ReLU	140	0.302895	0.129267	Softmax
145	0.222293	0.71665	L-ReLU	145	0.30282	0.130283	Softmax
150	0.220843	0.719933	L-ReLU	150	0.302747	0.131183	Softmax
155	0.219451	0.72315	L-ReLU	155	0.302676	0.1328	Softmax
160	0.218116	0.726467	L-ReLU	160	0.302607	0.13645	Softmax
165	0.216834	0.729267	L-ReLU	165	0.302539	0.13725	Softmax
170	0.215603	0.73195	L-ReLU	170	0.302474	0.138283	Softmax
175	0.214418	0.734933	L-ReLU	175	0.30241	0.139083	Softmax
180	0.21328	0.737517	L-ReLU	180	0.302347	0.139767	Softmax
185	0.212184	0.739717	L-ReLU	185	0.302286	0.140683	Softmax
190	0.211128	0.742133	L-ReLU	190	0.302227	0.14145	Softmax
195	0.210111	0.744517	L-ReLU	195	0.302168	0.142033	Softmax
200	0.209131	0.746583	L-ReLU	200	0.302111	0.1426	Softmax
205	0.208186	0.7485	L-ReLU	205	0.302056	0.143167	Softmax
210	0.207273	0.750383	L-ReLU	210	0.302001	0.143767	Softmax
215	0.206392	0.752033	L-ReLU	215	0.301948	0.14425	Softmax
220	0.205541	0.75385	L-ReLU	220	0.301895	0.14455	Softmax
225	0.204719	0.755217	L-ReLU	225	0.301844	0.1451	Softmax
230	0.203924	0.756817	L-ReLU	230	0.301793	0.145483	Softmax
235	0.203155	0.7583	L-ReLU	235	0.301744	0.14595	Softmax
240	0.20241	0.7597	L-ReLU	240	0.301696	0.146533	Softmax
245	0.20169	0.761067	L-ReLU	245	0.301648	0.147017	Softmax
250	0.200991	0.762167	L-ReLU	250	0.301601	0.147417	Softmax
255	0.200314	0.763333	L-ReLU	255	0.301555	0.14775	Softmax
260	0.199658	0.764467	L-ReLU	260	0.30151	0.148417	Softmax
265	0.199022	0.765833	L-ReLU	265	0.301466	0.148783	Softmax
270	0.198404	0.76705	L-ReLU	270	0.301423	0.149167	Softmax
275	0.197804	0.768133	L-ReLU	275	0.30138	0.14945	Softmax
280	0.197222	0.768967	L-ReLU	280	0.301338	0.149683	Softmax
285	0.196656	0.76995	L-ReLU	285	0.301296	0.150133	Softmax
290	0.196106	0.771067	L-ReLU	290	0.301255	0.150867	Softmax
295	0.195572	0.771817	L-ReLU	295	0.301215	0.15145	Softmax

ReLU – 2nd random				ReLU – 1st random, L-ReLU suppressed			
Epoch	Cost	Train Acc	Act Funct	Epoch	Cost	Train Acc	Act Funct
0	0.423081	0.08705	ReLU	0	0.423081	0.08705	ReLU
5	0.378508	0.171333	TanH	5	0.378508	0.171333	TanH
10	0.337886	0.118917	TanH	10	0.337886	0.118917	TanH
15	0.300777	0.208767	L-ReLU	15	0.314252	0.10215	ReLU
20	0.291179	0.29	L-ReLU	20	0.306164	0.106333	ReLU
25	0.2857	0.357817	L-ReLU	25	0.300732	0.10965	ReLU
30	0.280936	0.411933	L-ReLU	30	0.300085	0.11145	ReLU
35	0.276673	0.45475	L-ReLU	35	0.299978	0.112267	ReLU
40	0.27271	0.490117	L-ReLU	40	0.299968	0.112383	ReLU
45	0.268935	0.521917	L-ReLU	45	0.299968	0.112367	ReLU
50	0.265318	0.547883	L-ReLU	50	0.304792	0.097533	TanH
55	0.261858	0.569333	L-ReLU	55	0.299977	0.112367	Sigmoid
60	0.258557	0.588417	L-ReLU	60	0.299975	0.112367	Sigmoid
65	0.255409	0.6038	L-ReLU	65	0.302549	0.112367	TanH
70	0.252411	0.617683	L-ReLU	70	0.301406	0.112367	TanH
75	0.249556	0.629783	L-ReLU	75	0.299985	0.112367	ReLU
80	0.246837	0.640233	L-ReLU	80	0.299985	0.112367	ReLU
85	0.244248	0.649633	L-ReLU	85	0.299984	0.112367	ReLU
90	0.241782	0.658133	L-ReLU	90	0.300023	0.112367	TanH
95	0.239433	0.666183	L-ReLU	95	0.299981	0.112367	Sigmoid
100	0.237193	0.673917	L-ReLU	100	0.299969	0.112367	TanH
105	0.235058	0.680317	L-ReLU	105	0.29998	0.112367	ReLU
110	0.233021	0.6871	L-ReLU	110	0.299979	0.112367	Sigmoid
115	0.231078	0.69375	L-ReLU	115	0.299978	0.112367	ReLU
120	0.229222	0.699017	L-ReLU	120	0.292765	0.174933	Softmax
125	0.227449	0.7034	L-ReLU	125	0.292578	0.216567	Softmax
130	0.225754	0.7082	L-ReLU	130	0.292403	0.22575	Softmax
135	0.224132	0.712217	L-ReLU	135	0.292237	0.229017	Softmax
140	0.22258	0.715967	L-ReLU	140	0.29208	0.231817	Softmax
145	0.221094	0.719233	L-ReLU	145	0.291931	0.233267	Softmax
150	0.21967	0.722783	L-ReLU	150	0.291789	0.234783	Softmax
155	0.218305	0.7258	L-ReLU	155	0.291655	0.2359	Softmax
160	0.216995	0.729067	L-ReLU	160	0.291527	0.23655	Softmax
165	0.215737	0.731483	L-ReLU	165	0.291405	0.238333	Softmax
170	0.214528	0.734417	L-ReLU	170	0.291288	0.244833	Softmax
175	0.213367	0.737267	L-ReLU	175	0.291177	0.2483	Softmax
180	0.21225	0.739533	L-ReLU	180	0.291071	0.25275	Softmax
185	0.211175	0.742	L-ReLU	185	0.290969	0.277133	Softmax
190	0.21014	0.74435	L-ReLU	190	0.290871	0.2771	Softmax
195	0.209143	0.746567	L-ReLU	195	0.300042	0.112367	ReLU
200	0.208182	0.748383	L-ReLU	200	0.290662	0.28045	Softmax
205	0.207255	0.750433	L-ReLU	205	0.290568	0.281067	Softmax

210	0.20636	0.752267	L-ReLU	210	0.290479	0.282233	Softmax
215	0.205496	0.75395	L-ReLU	215	0.290392	0.282733	Softmax
220	0.204662	0.7553	L-ReLU	220	0.290309	0.283333	Softmax
225	0.203856	0.756867	L-ReLU	225	0.290229	0.28375	Softmax
230	0.203077	0.758317	L-ReLU	230	0.290152	0.284417	Softmax
235	0.202323	0.759817	L-ReLU	235	0.290077	0.2844	Softmax
240	0.201593	0.761067	L-ReLU	240	0.290007	0.285433	Softmax
245	0.200886	0.7623	L-ReLU	245	0.289935	0.285717	Softmax
250	0.200202	0.76355	L-ReLU	250	0.289865	0.286217	Softmax
255	0.199538	0.76495	L-ReLU	255	0.289798	0.286217	Softmax
260	0.198895	0.766133	L-ReLU	260	0.299952	0.112367	TanH
265	0.198271	0.7672	L-ReLU	265	0.300105	0.112367	Sigmoid
270	0.197665	0.7684	L-ReLU	270	0.300106	0.112367	Sigmoid
275	0.197077	0.7693	L-ReLU	275	0.300105	0.112367	ReLU
280	0.196506	0.77025	L-ReLU	280	0.300104	0.112367	Sigmoid
285	0.195952	0.771117	L-ReLU	285	0.289588	0.2899	Softmax
290	0.195413	0.772117	L-ReLU	290	0.289518	0.290117	Softmax
295	0.194888	0.77285	L-ReLU	295	0.289449	0.290617	Softmax

ReLU – 2^{nd} random, L-ReLU suppressed				Sigmoid – 1^{st} random			
Epoch	Cost	Train Acc	Act Funct	Epoch	Cost	Train Acc	Act Funct
0	0.423081	0.08705	ReLU	0	0.389802	0.08445	Sigmoid
5	0.304662	0.111683	Softmax	5	0.367211	0.099433	Sigmoid
10	0.304473	0.11265	Softmax	10	0.35641	0.11695	Sigmoid
15	0.304296	0.113633	Softmax	15	0.347985	0.110033	Sigmoid
20	0.304131	0.114167	Softmax	20	0.336465	0.102067	Sigmoid
25	0.303974	0.114733	Softmax	25	0.324407	0.094283	Sigmoid
30	0.303826	0.1154	Softmax	30	0.314465	0.091233	Sigmoid
35	0.303686	0.116133	Softmax	35	0.307775	0.088333	Sigmoid
40	0.303553	0.117267	Softmax	40	0.304123	0.0882	Sigmoid
45	0.303426	0.117883	Softmax	45	0.302183	0.089383	Sigmoid
50	0.303305	0.118867	Softmax	50	0.301177	0.08915	Sigmoid
55	0.30319	0.119767	Softmax	55	0.300687	0.089167	Sigmoid
60	0.303079	0.120483	Softmax	60	0.300448	0.089217	Sigmoid
65	0.302972	0.12135	Softmax	65	0.300327	0.089567	Sigmoid
70	0.30287	0.122417	Softmax	70	0.30026	0.0894	Sigmoid
75	0.302772	0.123433	Softmax	75	0.300218	0.104367	Sigmoid
80	0.302677	0.124767	Softmax	80	0.300188	0.103083	Sigmoid
85	0.302586	0.125767	Softmax	85	0.300165	0.102667	Sigmoid
90	0.302498	0.126983	Softmax	90	0.300145	0.1025	Sigmoid
95	0.302413	0.12855	Softmax	95	0.300128	0.102367	Sigmoid
100	0.30233	0.1299	Softmax	100	0.300114	0.102233	Sigmoid

105	0.30225	0.131617	Softmax	105	0.300102	0.102167	Sigmoid
110	0.302173	0.134183	Softmax	110	0.300093	0.102183	Sigmoid
115	0.302098	0.136983	Softmax	115	0.300085	0.102183	Sigmoid
120	0.302025	0.137067	Softmax	120	0.300078	0.102183	Sigmoid
125	0.301954	0.137667	Softmax	125	0.300072	0.102183	Sigmoid
130	0.301885	0.1385	Softmax	130	0.300066	0.102183	Sigmoid
135	0.301818	0.139217	Softmax	135	0.300061	0.102183	Sigmoid
140	0.301753	0.139833	Softmax	140	0.300056	0.102183	Sigmoid
145	0.301689	0.140383	Softmax	145	0.300052	0.102183	Sigmoid
150	0.301628	0.140983	Softmax	150	0.300048	0.102183	Sigmoid
155	0.301567	0.141333	Softmax	155	0.300044	0.102183	Sigmoid
160	0.301508	0.141833	Softmax	160	0.30004	0.102183	Sigmoid
165	0.301451	0.142167	Softmax	165	0.300037	0.102183	Sigmoid
170	0.301394	0.1426	Softmax	170	0.300034	0.102183	Sigmoid
175	0.301339	0.142917	Softmax	175	0.300031	0.102183	Sigmoid
180	0.301286	0.1434	Softmax	180	0.300028	0.102183	Sigmoid
185	0.301233	0.143317	Softmax	185	0.300025	0.102183	Sigmoid
190	0.30721	0.166717	Sigmoid	190	0.300022	0.102183	Sigmoid
195	0.305233	0.161017	Sigmoid	195	0.30002	0.102183	Sigmoid
200	0.335435	0.124467	ReLU	200	0.300018	0.102183	Sigmoid
205	0.324748	0.118883	ReLU	205	0.300016	0.102183	Sigmoid
210	0.300873	0.130017	Sigmoid	210	0.300014	0.102183	Sigmoid
215	0.30067	0.1255	Sigmoid	215	0.300012	0.102183	Sigmoid
220	0.303607	0.104183	ReLU	220	0.30001	0.102183	Sigmoid
225	0.300637	0.151983	Softmax	225	0.300008	0.102183	Sigmoid
230	0.300582	0.1524	Softmax	230	0.300006	0.102183	Sigmoid
235	0.300528	0.152883	Softmax	235	0.300005	0.102183	Sigmoid
240	0.300476	0.153267	Softmax	240	0.300003	0.102183	Sigmoid
245	0.300424	0.153633	Softmax	245	0.300001	0.112367	Sigmoid
250	0.300374	0.154167	Softmax	250	0.3	0.112367	Sigmoid
255	0.300324	0.154567	Softmax	255	0.299999	0.112367	Sigmoid
260	0.300276	0.1547	Softmax	260	0.299997	0.112367	Sigmoid
265	0.300229	0.155083	Softmax	265	0.299996	0.112367	Sigmoid
270	0.300183	0.155567	Softmax	270	0.299995	0.112367	Sigmoid
275	0.300137	0.156017	Softmax	275	0.299994	0.112367	Sigmoid
280	0.300093	0.156233	Softmax	280	0.299993	0.112367	Sigmoid
285	0.300049	0.1565	Softmax	285	0.299992	0.112367	Sigmoid
290	0.300006	0.156617	Softmax	290	0.299991	0.112367	Sigmoid
295	0.299964	0.157117	Softmax	295	0.29999	0.112367	Sigmoid

Sigmoid – 2nd random				Sigmoid – 1st random, L-ReLU suppressed			
Epoch	Cost	Train Acc	Act Funct	Epoch	Cost	Train Acc	Act Funct
0	0.389802	0.08445	Sigmoid	0	0.389802	0.08445	Sigmoid
5	0.383694	0.148033	TanH	5	0.310127	0.086683	Softmax
10	0.38828	0.109983	TanH	10	0.30981	0.08675	Softmax
15	0.335597	0.10155	L-ReLU	15	0.309524	0.08675	Softmax
20	0.307109	0.138133	L-ReLU	20	0.309629	0.087317	Softmax
25	0.297508	0.190483	L-ReLU	25	0.309381	0.087167	Softmax
30	0.291094	0.256283	L-ReLU	30	0.385678	0.130567	TanH
35	0.285742	0.32605	L-ReLU	35	0.364062	0.1002	TanH
40	0.281128	0.383667	L-ReLU	40	0.326329	0.096	ReLU
45	0.276997	0.4285	L-ReLU	45	0.301718	0.102683	Sigmoid
50	0.27315	0.468167	L-ReLU	50	0.300843	0.103783	Sigmoid
55	0.269482	0.503733	L-ReLU	55	0.300423	0.10415	Sigmoid
60	0.265971	0.531617	L-ReLU	60	0.300221	0.10345	Sigmoid
65	0.262611	0.554633	L-ReLU	65	0.309405	0.10295	TanH
70	0.259401	0.573467	L-ReLU	70	0.300013	0.10945	Sigmoid
75	0.256336	0.590883	L-ReLU	75	0.299999	0.110517	Sigmoid
80	0.25341	0.6048	L-ReLU	80	0.299988	0.1111	Sigmoid
85	0.250617	0.618167	L-ReLU	85	0.29998	0.111417	Sigmoid
90	0.247953	0.629517	L-ReLU	90	0.299974	0.111717	Sigmoid
95	0.24541	0.6398	L-ReLU	95	0.29997	0.111917	Sigmoid
100	0.242983	0.648983	L-ReLU	100	0.299967	0.112067	Sigmoid
105	0.240667	0.656767	L-ReLU	105	0.299966	0.112217	Sigmoid
110	0.238456	0.6644	L-ReLU	110	0.299964	0.112317	Sigmoid
115	0.236345	0.670567	L-ReLU	115	0.299963	0.112333	Sigmoid
120	0.234327	0.67715	L-ReLU	120	0.299963	0.11235	Sigmoid
125	0.232399	0.6837	L-ReLU	125	0.299962	0.112367	Sigmoid
130	0.230554	0.68965	L-ReLU	130	0.299961	0.112367	Sigmoid
135	0.22879	0.6947	L-ReLU	135	0.299961	0.112367	ReLU
140	0.227101	0.69955	L-ReLU	140	0.311575	0.131383	Softmax
145	0.225483	0.704217	L-ReLU	145	0.310987	0.132283	Softmax
150	0.223934	0.708	L-ReLU	150	0.310457	0.133283	Softmax
155	0.222448	0.711717	L-ReLU	155	0.309978	0.133783	Softmax
160	0.221023	0.71515	L-ReLU	160	0.309542	0.134833	Softmax
165	0.219655	0.718517	L-ReLU	165	0.309145	0.13545	Softmax
170	0.218341	0.7218	L-ReLU	170	0.308782	0.13605	Softmax
175	0.217078	0.724567	L-ReLU	175	0.308448	0.136917	Softmax
180	0.215865	0.727483	L-ReLU	180	0.308141	0.1381	Softmax
185	0.214697	0.729867	L-ReLU	185	0.307856	0.138517	Softmax
190	0.213574	0.732617	L-ReLU	190	0.307592	0.13945	Softmax
195	0.212491	0.73525	L-ReLU	195	0.307347	0.140183	Softmax
200	0.211449	0.7376	L-ReLU	200	0.307117	0.140833	Softmax
205	0.210444	0.739817	L-ReLU	205	0.306903	0.141533	Softmax

210	0.209475	0.7419	L-ReLU	210	0.306702	0.142317	Softmax
215	0.208539	0.7441	L-ReLU	215	0.306513	0.14255	Softmax
220	0.207636	0.7459	L-ReLU	220	0.306334	0.143083	Softmax
225	0.206763	0.747817	L-ReLU	225	0.306166	0.143917	Softmax
230	0.205919	0.749533	L-ReLU	230	0.306007	0.144767	Softmax
235	0.205104	0.75135	L-ReLU	235	0.305856	0.14525	Softmax
240	0.204315	0.75275	L-ReLU	240	0.305713	0.145717	Softmax
245	0.203551	0.754367	L-ReLU	245	0.305576	0.1464	Softmax
250	0.202811	0.75595	L-ReLU	250	0.305447	0.1469	Softmax
255	0.202095	0.75735	L-ReLU	255	0.305323	0.14745	Softmax
260	0.201401	0.758533	L-ReLU	260	0.305204	0.14795	Softmax
265	0.200728	0.759683	L-ReLU	265	0.305091	0.148567	Softmax
270	0.200075	0.760883	L-ReLU	270	0.304982	0.149133	Softmax
275	0.199441	0.762033	L-ReLU	275	0.304878	0.149617	Softmax
280	0.198826	0.76315	L-ReLU	280	0.304778	0.150333	Softmax
285	0.198229	0.764617	L-ReLU	285	0.304681	0.1507	Softmax
290	0.197648	0.765583	L-ReLU	290	0.304588	0.150933	Softmax
295	0.197084	0.766733	L-ReLU	295	0.304499	0.151233	Softmax

Sigmoid – 2nd random, L-ReLU suppressed				Softmax – 1st random			
Epoch	Cost	Train Acc	Act Funct	Epoch	Cost	Train Acc	Act Funct
0	0.389802	0.08445	Sigmoid	0	0.309923	0.083967	Softmax
5	0.417693	0.100833	ReLU	5	0.38914	0.084133	Sigmoid
10	0.393785	0.127233	ReLU	10	0.3672	0.100117	Sigmoid
15	0.329179	0.11215	ReLU	15	0.356499	0.11685	Sigmoid
20	0.420862	0.0922	TanH	20	0.348261	0.110867	Sigmoid
25	0.300265	0.107633	Sigmoid	25	0.374254	0.1056	L-ReLU
30	0.300127	0.1036	Sigmoid	30	0.331879	0.1242	L-ReLU
35	0.300051	0.10295	Sigmoid	35	0.309584	0.151983	L-ReLU
40	0.300137	0.1022	ReLU	40	0.298799	0.195533	L-ReLU
45	0.301831	0.132967	Softmax	45	0.291604	0.259533	L-ReLU
50	0.301593	0.133867	Softmax	50	0.285741	0.3254	L-ReLU
55	0.301377	0.134933	Softmax	55	0.280632	0.386883	L-ReLU
60	0.301179	0.135933	Softmax	60	0.276236	0.435383	L-ReLU
65	0.300996	0.1372	Softmax	65	0.272468	0.474117	L-ReLU
70	0.300827	0.138117	Softmax	70	0.269072	0.5053	L-ReLU
75	0.30067	0.138933	Softmax	75	0.265875	0.530733	L-ReLU
80	0.300523	0.139933	Softmax	80	0.262823	0.55185	L-ReLU
85	0.300386	0.140867	Softmax	85	0.2599	0.569633	L-ReLU
90	0.300257	0.14185	Softmax	90	0.257101	0.58545	L-ReLU
95	0.300135	0.14285	Softmax	95	0.254419	0.599617	L-ReLU
100	0.300019	0.143933	Softmax	100	0.251851	0.6114	L-ReLU

105	0.29991	0.1449	Softmax	105	0.249391	0.6232	L-ReLU
110	0.299806	0.146583	Softmax	110	0.247034	0.632433	L-ReLU
115	0.299707	0.148333	Softmax	115	0.244776	0.641383	L-ReLU
120	0.299612	0.153917	Softmax	120	0.242612	0.64925	L-ReLU
125	0.299522	0.156633	Softmax	125	0.240536	0.65705	L-ReLU
130	0.299435	0.159	Softmax	130	0.238547	0.663867	L-ReLU
135	0.299352	0.1608	Softmax	135	0.236639	0.6702	L-ReLU
140	0.299272	0.162567	Softmax	140	0.234809	0.676767	L-ReLU
145	0.299195	0.164	Softmax	145	0.233052	0.68175	L-ReLU
150	0.299121	0.16545	Softmax	150	0.231365	0.68705	L-ReLU
155	0.299049	0.1704	Softmax	155	0.229744	0.692283	L-ReLU
160	0.29898	0.171483	Softmax	160	0.228187	0.6967	L-ReLU
165	0.298913	0.172183	Softmax	165	0.22669	0.7009	L-ReLU
170	0.298848	0.173067	Softmax	170	0.22525	0.704783	L-ReLU
175	0.298785	0.173867	Softmax	175	0.223863	0.70835	L-ReLU
180	0.298724	0.17485	Softmax	180	0.222529	0.7116	L-ReLU
185	0.298665	0.17555	Softmax	185	0.221244	0.715017	L-ReLU
190	0.298607	0.1763	Softmax	190	0.220006	0.717917	L-ReLU
195	0.298551	0.1771	Softmax	195	0.218812	0.720867	L-ReLU
200	0.298497	0.1777	Softmax	200	0.21766	0.7237	L-ReLU
205	0.298444	0.178267	Softmax	205	0.216549	0.72655	L-ReLU
210	0.298392	0.178483	Softmax	210	0.215476	0.72915	L-ReLU
215	0.298342	0.178917	Softmax	215	0.21444	0.73135	L-ReLU
220	0.298293	0.179817	Softmax	220	0.213439	0.733383	L-ReLU
225	0.298244	0.180583	Softmax	225	0.212471	0.735483	L-ReLU
230	0.298198	0.180967	Softmax	230	0.211534	0.737467	L-ReLU
235	0.298152	0.181733	Softmax	235	0.210628	0.739633	L-ReLU
240	0.298107	0.182567	Softmax	240	0.209751	0.74175	L-ReLU
245	0.298063	0.183233	Softmax	245	0.208902	0.7439	L-ReLU
250	0.29802	0.183567	Softmax	250	0.208079	0.74555	L-ReLU
255	0.297978	0.183983	Softmax	255	0.207281	0.747083	L-ReLU
260	0.297937	0.18445	Softmax	260	0.206508	0.7487	L-ReLU
265	0.297896	0.1849	Softmax	265	0.205757	0.750317	L-ReLU
270	0.297857	0.18545	Softmax	270	0.205029	0.751417	L-ReLU
275	0.297818	0.185883	Softmax	275	0.204323	0.753217	L-ReLU
280	0.29778	0.186283	Softmax	280	0.203637	0.754833	L-ReLU
285	0.297743	0.186867	Softmax	285	0.20297	0.756217	L-ReLU
290	0.297706	0.187283	Softmax	290	0.202322	0.757317	L-ReLU
295	0.29767	0.187717	Softmax				

Softmax – 2nd random				Softmax – 1st random, L-ReLU suppressed			
Epoch	Cost	Train Acc	Act Funct	Epoch	Cost	Train Acc	Act Funct
0	0.309923	0.083967	Softmax	0	0.309923	0.083967	Softmax
5	0.411599	0.105817	ReLU	5	0.38914	0.084133	Sigmoid
10	0.353546	0.1119	ReLU	10	0.3672	0.100117	Sigmoid
15	0.308239	0.104033	ReLU	15	0.356499	0.11685	Sigmoid
20	0.301027	0.110183	ReLU	20	0.348261	0.110867	Sigmoid
25	0.300113	0.11155	ReLU	25	0.388266	0.1152	TanH
30	0.299964	0.112267	ReLU	30	0.33247	0.097417	TanH
35	0.299957	0.112383	ReLU	35	0.302687	0.101517	Sigmoid
40	0.299961	0.112367	ReLU	40	0.301301	0.102517	Sigmoid
45	0.324708	0.204167	L-ReLU	45	0.300639	0.1036	Sigmoid
50	0.285035	0.33685	L-ReLU	50	0.300335	0.103017	Sigmoid
55	0.269178	0.4797	L-ReLU	55	0.311375	0.107017	Softmax
60	0.260913	0.52125	L-ReLU	60	0.310931	0.107317	Softmax
65	0.253859	0.566717	L-ReLU	65	0.310542	0.107717	Softmax
70	0.25051	0.594283	L-ReLU	70	0.310196	0.108317	Softmax
75	0.247696	0.608667	L-ReLU	75	0.309887	0.109133	Softmax
80	0.24498	0.6212	L-ReLU	80	0.309608	0.10955	Softmax
85	0.242397	0.632183	L-ReLU	85	0.309354	0.109783	Softmax
90	0.239942	0.64195	L-ReLU	90	0.309121	0.10975	Softmax
95	0.23761	0.651033	L-ReLU	95	0.305853	0.098567	TanH
100	0.235393	0.658883	L-ReLU	100	0.304079	0.103917	TanH
105	0.233284	0.666067	L-ReLU	105	0.301714	0.111517	TanH
110	0.231276	0.67305	L-ReLU	110	0.300211	0.112083	TanH
115	0.229363	0.6797	L-ReLU	115	0.299966	0.112333	TanH
120	0.22754	0.686	L-ReLU	120	0.299953	0.112367	TanH
125	0.225801	0.691117	L-ReLU	125	0.299952	0.112367	TanH
130	0.224141	0.696767	L-ReLU	130	0.299988	0.112367	Sigmoid
135	0.222555	0.701417	L-ReLU	135	0.299987	0.112367	Sigmoid
140	0.221039	0.706	L-ReLU	140	0.299986	0.112367	ReLU
145	0.219589	0.7106	L-ReLU	145	0.306514	0.147733	Softmax
150	0.2182	0.714067	L-ReLU	150	0.306138	0.148317	Softmax
155	0.216871	0.717683	L-ReLU	155	0.305791	0.1494	Softmax
160	0.215596	0.721	L-ReLU	160	0.305467	0.150283	Softmax
165	0.214372	0.724383	L-ReLU	165	0.305166	0.151017	Softmax
170	0.213198	0.727183	L-ReLU	170	0.304885	0.151733	Softmax
175	0.212071	0.729833	L-ReLU	175	0.304622	0.1524	Softmax
180	0.210987	0.7328	L-ReLU	180	0.304375	0.153233	Softmax
185	0.209945	0.734867	L-ReLU	185	0.304143	0.15425	Softmax
190	0.208942	0.73695	L-ReLU	190	0.303924	0.155383	Softmax
195	0.207976	0.738967	L-ReLU	195	0.303718	0.15695	Softmax
200	0.207045	0.740983	L-ReLU	200	0.303522	0.1582	Softmax
205	0.206148	0.74295	L-ReLU	205	0.303337	0.158917	Softmax

210	0.205283	0.74505	L-ReLU	210	0.303161	0.159433	Softmax
215	0.204447	0.746783	L-ReLU	215	0.302994	0.160167	Softmax
220	0.203641	0.748717	L-ReLU	220	0.302835	0.160683	Softmax
225	0.202861	0.750067	L-ReLU	225	0.302683	0.161333	Softmax
230	0.202108	0.751833	L-ReLU	230	0.302538	0.16185	Softmax
235	0.201379	0.75355	L-ReLU	235	0.302399	0.162283	Softmax
240	0.200674	0.755	L-ReLU	240	0.302267	0.162983	Softmax
245	0.199991	0.756433	L-ReLU	245	0.302139	0.16415	Softmax
250	0.19933	0.758	L-ReLU	250	0.302017	0.164367	Softmax
255	0.198689	0.759233	L-ReLU	255	0.3019	0.16505	Softmax
260	0.198067	0.76025	L-ReLU	260	0.301788	0.165717	Softmax
265	0.197464	0.761467	L-ReLU	265	0.301679	0.1665	Softmax
270	0.19688	0.7625	L-ReLU	270	0.301575	0.16685	Softmax
275	0.196312	0.763633	L-ReLU	275	0.301474	0.16755	Softmax
280	0.195761	0.764783	L-ReLU	280	0.301377	0.168183	Softmax
285	0.195225	0.765983	L-ReLU	285	0.301283	0.16855	Softmax
290	0.194704	0.7669	L-ReLU	290	0.301192	0.169067	Softmax
295	0.194198	0.768017	L-ReLU	295	0.301104	0.169567	Softmax

Softmax – 2nd random, L-ReLU suppressed				TanH – 1st random			
Epoch	Cost	Train Acc	Act Funct	Epoch	Cost	Train Acc	Act Funct
0	0.309923	0.083967	Softmax	0	0.390736	0.113733	TanH
5	0.38914	0.084133	Sigmoid	5	0.308284	0.08975	Softmax
10	0.3672	0.100117	Sigmoid	10	0.409017	0.0971	TanH
15	0.356499	0.11685	Sigmoid	15	0.324257	0.1154	TanH
20	0.348261	0.110867	Sigmoid	20	0.304702	0.114133	TanH
25	0.400699	0.0961	ReLU	25	0.302657	0.110983	TanH
30	0.314382	0.099017	Sigmoid	30	0.300053	0.099167	ReLU
35	0.307822	0.09775	Sigmoid	35	0.300023	0.09925	Sigmoid
40	0.303059	0.103217	Sigmoid	40	0.30002	0.099217	Sigmoid
45	0.301575	0.103533	Sigmoid	45	0.300019	0.112367	Sigmoid
50	0.300817	0.103417	Sigmoid	50	0.300015	0.112367	Sigmoid
55	0.30108	0.102683	ReLU	55	0.296658	0.163467	Softmax
60	0.30016	0.102683	Sigmoid	60	0.295918	0.172683	Softmax
65	0.300224	0.102283	ReLU	65	0.29527	0.182583	Softmax
70	0.300048	0.102317	Sigmoid	70	0.294693	0.1908	Softmax
75	0.300036	0.102283	Sigmoid	75	0.294168	0.198317	Softmax
80	0.300019	0.102183	ReLU	80	0.293689	0.204883	Softmax
85	0.306231	0.09865	TanH	85	0.29325	0.21105	Softmax
90	0.299997	0.102183	ReLU	90	0.292845	0.2169	Softmax
95	0.299994	0.102183	ReLU	95	0.292474	0.221683	Softmax
100	0.303925	0.104417	TanH	100	0.292131	0.22665	Softmax

105	0.30166	0.112367	TanH	105	0.291817	0.2314	Softmax
110	0.300197	0.112367	TanH	110	0.291531	0.235583	Softmax
115	0.300096	0.112367	TanH	115	0.291267	0.2392	Softmax
120	0.299986	0.112367	ReLU	120	0.291021	0.2431	Softmax
125	0.299985	0.112367	Sigmoid	125	0.29079	0.246283	Softmax
130	0.299984	0.112367	Sigmoid	130	0.29057	0.249983	Softmax
135	0.309525	0.135633	Softmax	135	0.29034	0.2539	Softmax
140	0.308648	0.136933	Softmax	140	0.290124	0.257317	Softmax
145	0.307867	0.138367	Softmax	145	0.28992	0.272933	Softmax
150	0.307167	0.1392	Softmax	150	0.289727	0.2908	Softmax
155	0.306537	0.140667	Softmax	155	0.289545	0.292467	Softmax
160	0.305967	0.14245	Softmax	160	0.289373	0.294183	Softmax
165	0.305448	0.14475	Softmax	165	0.289216	0.295667	Softmax
170	0.304975	0.14555	Softmax	170	0.289067	0.297783	Softmax
175	0.304541	0.146917	Softmax	175	0.288925	0.29945	Softmax
180	0.304142	0.148217	Softmax	180	0.28879	0.300317	Softmax
185	0.303773	0.14895	Softmax	185	0.288662	0.30135	Softmax
190	0.303432	0.14985	Softmax	190	0.288541	0.301667	Softmax
195	0.303115	0.15055	Softmax	195	0.288427	0.30275	Softmax
200	0.30282	0.151267	Softmax	200	0.288318	0.3037	Softmax
205	0.302545	0.15195	Softmax	205	0.288214	0.305433	Softmax
210	0.302287	0.152883	Softmax	210	0.288115	0.306083	Softmax
215	0.302046	0.153667	Softmax	215	0.288021	0.306533	Softmax
220	0.301819	0.154133	Softmax	220	0.287931	0.307417	Softmax
225	0.301606	0.1548	Softmax	225	0.287844	0.3082	Softmax
230	0.301404	0.155817	Softmax	230	0.287754	0.30895	Softmax
235	0.301214	0.156767	Softmax	235	0.287664	0.310033	Softmax
240	0.301034	0.1575	Softmax	240	0.287582	0.310367	Softmax
245	0.300863	0.15835	Softmax	245	0.287505	0.311367	Softmax
250	0.300702	0.159267	Softmax	250	0.287433	0.312117	Softmax
255	0.300548	0.160433	Softmax	255	0.287365	0.3132	Softmax
260	0.300401	0.161183	Softmax	260	0.287303	0.313867	Softmax
265	0.300261	0.161683	Softmax	265	0.287244	0.314333	Softmax
270	0.300128	0.162367	Softmax	270	0.287184	0.314883	Softmax
275	0.300001	0.162917	Softmax	275	0.28712	0.3153	Softmax
280	0.299879	0.163533	Softmax	280	0.287062	0.315817	Softmax
285	0.299762	0.163933	Softmax	285	0.287008	0.3166	Softmax
290	0.29965	0.16525	Softmax	290	0.286958	0.317433	Softmax
295	0.299543	0.166083	Softmax	295	0.286912	0.318217	Softmax

TanH – 2nd random				TanH – 1st random, L-ReLU suppressed			
Epoch	Cost	Train Acc	Act Funct	Epoch	Cost	Train Acc	Act Funct
0	0.390736	0.113733	TanH	0	0.390736	0.113733	TanH
5	0.310951	0.0856	Softmax	5	0.356247	0.099217	Sigmoid
10	0.38247	0.098133	L-ReLU	10	0.32274	0.095967	Sigmoid
15	0.325669	0.114683	L-ReLU	15	0.310244	0.093633	Softmax
20	0.30588	0.153683	L-ReLU	20	0.334908	0.096683	TanH
25	0.29724	0.211017	L-ReLU	25	0.306624	0.102067	TanH
30	0.290441	0.281717	L-ReLU	30	0.30205	0.109817	TanH
35	0.284534	0.343467	L-ReLU	35	0.300209	0.111417	TanH
40	0.279499	0.392317	L-ReLU	40	0.299962	0.112183	TanH
45	0.275226	0.44045	L-ReLU	45	0.299954	0.112367	TanH
50	0.2714	0.481183	L-ReLU	50	0.299952	0.112367	TanH
55	0.267783	0.513417	L-ReLU	55	0.300074	0.09915	ReLU
60	0.26432	0.540217	L-ReLU	60	0.299952	0.112367	TanH
65	0.261008	0.56125	L-ReLU	65	0.299952	0.112367	TanH
70	0.257846	0.58045	L-ReLU	70	0.300052	0.09915	ReLU
75	0.254831	0.5961	L-ReLU	75	0.300044	0.09915	ReLU
80	0.251955	0.610483	L-ReLU	80	0.299952	0.112367	TanH
85	0.249213	0.622883	L-ReLU	85	0.299952	0.112367	TanH
90	0.246598	0.633283	L-ReLU	90	0.299952	0.112367	TanH
95	0.244105	0.6429	L-ReLU	95	0.300026	0.112367	ReLU
100	0.241727	0.6522	L-ReLU	100	0.299952	0.112367	TanH
105	0.239459	0.659917	L-ReLU	105	0.300013	0.112367	ReLU
110	0.237293	0.6672	L-ReLU	110	0.299952	0.112367	TanH
115	0.235226	0.67415	L-ReLU	115	0.306068	0.132317	Softmax
120	0.233251	0.680717	L-ReLU	120	0.305467	0.133833	Softmax
125	0.231365	0.6864	L-ReLU	125	0.304923	0.135267	Softmax
130	0.229561	0.691933	L-ReLU	130	0.30443	0.136733	Softmax
135	0.227835	0.6976	L-ReLU	135	0.303978	0.138483	Softmax
140	0.226183	0.701967	L-ReLU	140	0.303565	0.14005	Softmax
145	0.224601	0.706517	L-ReLU	145	0.303183	0.141533	Softmax
150	0.223086	0.71025	L-ReLU	150	0.30283	0.143217	Softmax
155	0.221632	0.71375	L-ReLU	155	0.302503	0.14435	Softmax
160	0.220238	0.717183	L-ReLU	160	0.302198	0.1463	Softmax
165	0.2189	0.720267	L-ReLU	165	0.301914	0.147983	Softmax
170	0.217615	0.723167	L-ReLU	170	0.301647	0.14945	Softmax
175	0.21638	0.72635	L-ReLU	175	0.301398	0.15105	Softmax
180	0.215193	0.72895	L-ReLU	180	0.301163	0.15225	Softmax
185	0.21405	0.731683	L-ReLU	185	0.300942	0.153967	Softmax
190	0.212951	0.734217	L-ReLU	190	0.300733	0.1554	Softmax
195	0.211891	0.736517	L-ReLU	195	0.300535	0.15685	Softmax
200	0.210871	0.738817	L-ReLU	200	0.300348	0.1581	Softmax
205	0.209887	0.741067	L-ReLU	205	0.300169	0.15935	Softmax

210	0.208938	0.743067	L-ReLU	210	0.3	0.16075	Softmax
215	0.208022	0.745233	L-ReLU	215	0.299838	0.1624	Softmax
220	0.207137	0.747183	L-ReLU	220	0.299683	0.16405	Softmax
225	0.206282	0.7488	L-ReLU	225	0.299535	0.165867	Softmax
230	0.205456	0.750317	L-ReLU	230	0.299394	0.168283	Softmax
235	0.204657	0.75215	L-ReLU	235	0.299258	0.172617	Softmax
240	0.203884	0.753867	L-ReLU	240	0.299127	0.175	Softmax
245	0.203135	0.755383	L-ReLU	245	0.299001	0.176617	Softmax
250	0.20241	0.756717	L-ReLU	250	0.29888	0.178033	Softmax
255	0.201708	0.757817	L-ReLU	255	0.298764	0.178617	Softmax
260	0.201027	0.7591	L-ReLU	260	0.298651	0.179417	Softmax
265	0.200367	0.760467	L-ReLU	265	0.298543	0.180467	Softmax
270	0.199726	0.7619	L-ReLU	270	0.298438	0.181417	Softmax
275	0.199105	0.763083	L-ReLU	275	0.298336	0.181933	Softmax
280	0.198501	0.764117	L-ReLU	280	0.298238	0.182967	Softmax
285	0.197915	0.765133	L-ReLU	285	0.298142	0.18375	Softmax
290	0.197345	0.7661	L-ReLU	290	0.29805	0.184517	Softmax
295	0.196792	0.766983	L-ReLU	295	0.29796	0.185183	Softmax

TanH – 2nd random, L-ReLU suppressed

Epoch	Cost	Train Acc	Act Funct
0	0.390736	0.113733	TanH
5	0.310951	0.0856	Softmax
10	0.3354	0.102983	Sigmoid
15	0.321279	0.0951	Sigmoid
20	0.305589	0.08885	Sigmoid
25	0.305116	0.10335	ReLU
30	0.300738	0.108883	ReLU
35	0.300133	0.111083	ReLU
40	0.299985	0.11215	ReLU
45	0.299969	0.11235	ReLU
50	0.299969	0.112367	ReLU
55	0.299968	0.112367	ReLU
60	0.321663	0.098633	TanH
65	0.31398	0.102183	TanH
70	0.305137	0.102183	TanH
75	0.306874	0.1116	Softmax
80	0.306414	0.111817	Softmax
85	0.306015	0.11555	Softmax
90	0.305665	0.116117	Softmax
95	0.305356	0.11655	Softmax
100	0.305081	0.116867	Softmax

105	0.304834	0.116917	Softmax
110	0.30461	0.117117	Softmax
115	0.304407	0.1172	Softmax
120	0.304222	0.1174	Softmax
125	0.304051	0.11735	Softmax
130	0.303541	0.112367	TanH
135	0.302888	0.112367	TanH
140	0.303847	0.11925	Softmax
145	0.303703	0.11935	Softmax
150	0.303568	0.11945	Softmax
155	0.303443	0.11945	Softmax
160	0.303375	0.120117	Softmax
165	0.303258	0.120133	Softmax
170	0.303148	0.12025	Softmax
175	0.303045	0.120517	Softmax
180	0.302947	0.120683	Softmax
185	0.302855	0.120717	Softmax
190	0.302767	0.120717	Softmax
195	0.300399	0.098717	Sigmoid
200	0.30039	0.098717	Sigmoid
205	0.300388	0.098717	ReLU
210	0.300389	0.098717	Sigmoid
215	0.302457	0.124433	Softmax
220	0.302379	0.124417	Softmax
225	0.300761	0.112367	TanH
230	0.300396	0.098717	Sigmoid
235	0.302256	0.12645	Softmax
240	0.302183	0.126533	Softmax
245	0.302114	0.126683	Softmax
250	0.302048	0.12665	Softmax
255	0.300315	0.112367	TanH
260	0.300446	0.098717	Sigmoid
265	0.300437	0.098717	Sigmoid
270	0.300436	0.098717	ReLU
275	0.300432	0.098717	Sigmoid
280	0.300428	0.098717	ReLU
285	0.300424	0.098717	ReLU
290	0.299993	0.112367	TanH
295	0.29996	0.112367	TanH

TanH Baseline, 30,000 epochs				TanH – Random, 30,000 epochs			
Epoch	**Cost**	**Acc**	**AF**	**Epoch**	**Cost**	**Acc**	**AF**
0	0.394715	0.1115	TanH	0	0.390736	0.113733	TanH
100	0.299952	0.112367	TanH	100	0.299384	0.136883	Softmax
200	0.299952	0.112367	TanH	200	0.295722	0.199583	Softmax
300	0.299952	0.112367	TanH	300	0.294328	0.234417	Softmax
400	0.299952	0.112367	TanH	400	0.293529	0.24325	Softmax
500	0.299952	0.112367	TanH	500	0.292986	0.249067	Softmax
600	0.299952	0.112367	TanH	600	0.29258	0.25385	Softmax
700	0.299952	0.112367	TanH	700	0.29226	0.25745	Softmax
800	0.299952	0.112367	TanH	800	0.291999	0.260217	Softmax
900	0.299952	0.112367	TanH	900	0.291779	0.262667	Softmax
1000	0.299952	0.112367	TanH	1000	0.291592	0.264967	Softmax
1100	0.299952	0.112367	TanH	1100	0.291429	0.267017	Softmax
1200	0.299952	0.112367	TanH	1200	0.291286	0.268383	Softmax
1300	0.299952	0.112367	TanH	1300	0.291159	0.2693	Softmax
1400	0.299952	0.112367	TanH	1400	0.291045	0.2702	Softmax
1500	0.299952	0.112367	TanH	1500	0.290942	0.271167	Softmax
1600	0.299952	0.112367	TanH	1600	0.290848	0.272383	Softmax
1700	0.299952	0.112367	TanH	1700	0.290763	0.2733	Softmax
1800	0.299952	0.112367	TanH	1800	0.290683	0.274233	Softmax
1900	0.299952	0.112367	TanH	1900	0.29061	0.275067	Softmax
2000	0.299952	0.112367	TanH	2000	0.290541	0.276017	Softmax
2100	0.299952	0.112367	TanH	2100	0.290477	0.276867	Softmax
2200	0.299952	0.112367	TanH	2200	0.290416	0.277433	Softmax
2300	0.299952	0.112367	TanH	2300	0.290359	0.2784	Softmax
2400	0.299952	0.112367	TanH	2400	0.290305	0.27905	Softmax
2500	0.299952	0.112367	TanH	2500	0.290253	0.27935	Softmax
2600	0.299952	0.112367	TanH	2600	0.290204	0.27995	Softmax
2700	0.299952	0.112367	TanH	2700	0.290158	0.28105	Softmax
2800	0.299952	0.112367	TanH	2800	0.290114	0.281667	Softmax
2900	0.299952	0.112367	TanH	2900	0.290071	0.281933	Softmax
3000	0.299952	0.112367	TanH	3000	0.29003	0.282067	Softmax
3100	0.299952	0.112367	TanH	3100	0.289991	0.2827	Softmax
3200	0.299952	0.112367	TanH	3200	0.289953	0.282867	Softmax
3300	0.299952	0.112367	TanH	3300	0.289917	0.283467	Softmax
3400	0.299952	0.112367	TanH	3400	0.289882	0.284083	Softmax
3500	0.299952	0.112367	TanH	3500	0.289848	0.284817	Softmax
3600	0.299952	0.112367	TanH	3600	0.289815	0.285033	Softmax
3700	0.299952	0.112367	TanH	3700	0.289783	0.285283	Softmax
3800	0.299952	0.112367	TanH	3800	0.289753	0.285383	Softmax
3900	0.299952	0.112367	TanH	3900	0.289723	0.285667	Softmax
4000	0.299952	0.112367	TanH	4000	0.289694	0.2857	Softmax
4100	0.299952	0.112367	TanH	4100	0.289665	0.285867	Softmax

4200	0.299952	0.112367	TanH	4200	0.289638	0.286317	Softmax
4300	0.299952	0.112367	TanH	4300	0.289611	0.286617	Softmax
4400	0.299952	0.112367	TanH	4400	0.289585	0.28705	Softmax
4500	0.299952	0.112367	TanH	4500	0.28956	0.287233	Softmax
4600	0.299952	0.112367	TanH	4600	0.289535	0.2875	Softmax
4700	0.299952	0.112367	TanH	4700	0.289511	0.287467	Softmax
4800	0.299952	0.112367	TanH	4800	0.300349	0.104417	TanH
4900	0.299952	0.112367	TanH	4900	0.289102	0.2875	Softmax
5000	0.299952	0.112367	TanH	5000	0.289079	0.2876	Softmax
5100	0.299952	0.112367	TanH	5100	0.289056	0.287717	Softmax
5200	0.299952	0.112367	TanH	5200	0.289033	0.288083	Softmax
5300	0.299952	0.112367	TanH	5300	0.289011	0.288	Softmax
5400	0.299952	0.112367	TanH	5400	0.262503	0.545783	L-ReLU
5500	0.299952	0.112367	TanH	5500	0.25988	0.564483	L-ReLU
5600	0.299952	0.112367	TanH	5600	0.257984	0.57465	L-ReLU
5700	0.299952	0.112367	TanH	5700	0.256159	0.585183	L-ReLU
5800	0.299952	0.112367	TanH	5800	0.254386	0.5948	L-ReLU
5900	0.299952	0.112367	TanH	5900	0.252661	0.604533	L-ReLU
6000	0.299952	0.112367	TanH	6000	0.250984	0.61145	L-ReLU
6100	0.299952	0.112367	TanH	6100	0.249353	0.618517	L-ReLU
6200	0.299952	0.112367	TanH	6200	0.247768	0.6251	L-ReLU
6300	0.299952	0.112367	TanH	6300	0.246226	0.6314	L-ReLU
6400	0.299952	0.112367	TanH	6400	0.244727	0.636783	L-ReLU
6500	0.299952	0.112367	TanH	6500	0.24327	0.642217	L-ReLU
6600	0.299952	0.112367	TanH	6600	0.241853	0.647467	L-ReLU
6700	0.299952	0.112367	TanH	6700	0.240474	0.652733	L-ReLU
6800	0.299952	0.112367	TanH	6800	0.239134	0.657233	L-ReLU
6900	0.299952	0.112367	TanH	6900	0.237831	0.662067	L-ReLU
7000	0.299952	0.112367	TanH	7000	0.236563	0.666467	L-ReLU
7100	0.299952	0.112367	TanH	7100	0.235329	0.670767	L-ReLU
7200	0.299952	0.112367	TanH	7200	0.234129	0.67505	L-ReLU
7300	0.299952	0.112367	TanH	7300	0.232961	0.678767	L-ReLU
7400	0.299952	0.112367	TanH	7400	0.231824	0.6825	L-ReLU
7500	0.299952	0.112367	TanH	7500	0.230718	0.686067	L-ReLU
7600	0.299952	0.112367	TanH	7600	0.22964	0.689483	L-ReLU
7700	0.299952	0.112367	TanH	7700	0.228591	0.692767	L-ReLU
7800	0.299952	0.112367	TanH	7800	0.227569	0.695533	L-ReLU
7900	0.299952	0.112367	TanH	7900	0.226574	0.698483	L-ReLU
8000	0.299952	0.112367	TanH	8000	0.225604	0.7015	L-ReLU
8100	0.299952	0.112367	TanH	8100	0.224659	0.703933	L-ReLU
8200	0.299952	0.112367	TanH	8200	0.223737	0.706417	L-ReLU
8300	0.299952	0.112367	TanH	8300	0.222839	0.7087	L-ReLU
8400	0.299952	0.112367	TanH	8400	0.221963	0.711133	L-ReLU
8500	0.299952	0.112367	TanH	8500	0.221109	0.71345	L-ReLU

8600	0.299952	0.112367	TanH	8600	0.220275	0.71575	L-ReLU
8700	0.299952	0.112367	TanH	8700	0.219462	0.717983	L-ReLU
8800	0.299952	0.112367	TanH	8800	0.218668	0.72	L-ReLU
8900	0.299952	0.112367	TanH	8900	0.217893	0.72235	L-ReLU
9000	0.299952	0.112367	TanH	9000	0.217136	0.7242	L-ReLU
9100	0.299952	0.112367	TanH	9100	0.216398	0.72575	L-ReLU
9200	0.299952	0.112367	TanH	9200	0.215676	0.727167	L-ReLU
9300	0.299952	0.112367	TanH	9300	0.214971	0.728617	L-ReLU
9400	0.299952	0.112367	TanH	9400	0.214282	0.730167	L-ReLU
9500	0.299952	0.112367	TanH	9500	0.213609	0.731733	L-ReLU
9600	0.299952	0.112367	TanH	9600	0.212951	0.73315	L-ReLU
9700	0.299952	0.112367	TanH	9700	0.212308	0.734417	L-ReLU
9800	0.299952	0.112367	TanH	9800	0.211679	0.73585	L-ReLU
9900	0.299952	0.112367	TanH	9900	0.211063	0.73695	L-ReLU
10000	0.299952	0.112367	TanH	10000	0.210461	0.738167	L-ReLU
10100	0.299952	0.112367	TanH	10100	0.209872	0.739717	L-ReLU
10200	0.299952	0.112367	TanH	10200	0.209296	0.74105	L-ReLU
10300	0.299952	0.112367	TanH	10300	0.208732	0.742517	L-ReLU
10400	0.299952	0.112367	TanH	10400	0.208179	0.743717	L-ReLU
10500	0.299952	0.112367	TanH	10500	0.207638	0.744617	L-ReLU
10600	0.299952	0.112367	TanH	10600	0.207109	0.7458	L-ReLU
10700	0.299952	0.112367	TanH	10700	0.20659	0.74685	L-ReLU
10800	0.299952	0.112367	TanH	10800	0.206081	0.7479	L-ReLU
10900	0.299952	0.112367	TanH	10900	0.205583	0.748967	L-ReLU
11000	0.299952	0.112367	TanH	11000	0.205095	0.750033	L-ReLU
11100	0.299952	0.112367	TanH	11100	0.204616	0.750983	L-ReLU
11200	0.299952	0.112367	TanH	11200	0.204147	0.752083	L-ReLU
11300	0.299952	0.112367	TanH	11300	0.203687	0.753067	L-ReLU
11400	0.299952	0.112367	TanH	11400	0.203235	0.7539	L-ReLU
11500	0.299952	0.112367	TanH	11500	0.202793	0.754833	L-ReLU
11600	0.299952	0.112367	TanH	11600	0.202359	0.7557	L-ReLU
11700	0.299952	0.112367	TanH	11700	0.201932	0.756533	L-ReLU
11800	0.299952	0.112367	TanH	11800	0.201514	0.757517	L-ReLU
11900	0.299952	0.112367	TanH	11900	0.201104	0.758617	L-ReLU
12000	0.299952	0.112367	TanH	12000	0.200701	0.7595	L-ReLU
12100	0.299952	0.112367	TanH	12100	0.200305	0.760217	L-ReLU
12200	0.299952	0.112367	TanH	12200	0.199917	0.760867	L-ReLU
12300	0.299952	0.112367	TanH	12300	0.199535	0.761533	L-ReLU
12400	0.299952	0.112367	TanH	12400	0.19916	0.762183	L-ReLU
12500	0.299952	0.112367	TanH	12500	0.198792	0.762883	L-ReLU
12600	0.299952	0.112367	TanH	12600	0.19843	0.763583	L-ReLU
12700	0.299952	0.112367	TanH	12700	0.198075	0.7642	L-ReLU
12800	0.299952	0.112367	TanH	12800	0.197725	0.764883	L-ReLU
12900	0.299952	0.112367	TanH	12900	0.197382	0.765683	L-ReLU

13000	0.299952	0.112367	TanH	13000	0.197044	0.76645	L-ReLU
13100	0.299952	0.112367	TanH	13100	0.196712	0.767017	L-ReLU
13200	0.299952	0.112367	TanH	13200	0.196385	0.767683	L-ReLU
13300	0.299952	0.112367	TanH	13300	0.196064	0.768133	L-ReLU
13400	0.299952	0.112367	TanH	13400	0.195749	0.76885	L-ReLU
13500	0.299952	0.112367	TanH	13500	0.195438	0.769333	L-ReLU
13600	0.299952	0.112367	TanH	13600	0.195132	0.769783	L-ReLU
13700	0.299952	0.112367	TanH	13700	0.194832	0.77025	L-ReLU
13800	0.299952	0.112367	TanH	13800	0.194536	0.77075	L-ReLU
13900	0.299952	0.112367	TanH	13900	0.194245	0.771267	L-ReLU
14000	0.299952	0.112367	TanH	14000	0.193958	0.771733	L-ReLU
14100	0.299952	0.112367	TanH	14100	0.193676	0.772167	L-ReLU
14200	0.299952	0.112367	TanH	14200	0.193398	0.772683	L-ReLU
14300	0.299952	0.112367	TanH	14300	0.193125	0.773117	L-ReLU
14400	0.299952	0.112367	TanH	14400	0.192855	0.773683	L-ReLU
14500	0.299952	0.112367	TanH	14500	0.19259	0.774017	L-ReLU
14600	0.299952	0.112367	TanH	14600	0.192329	0.774383	L-ReLU
14700	0.299952	0.112367	TanH	14700	0.192071	0.774817	L-ReLU
14800	0.299952	0.112367	TanH	14800	0.191818	0.775283	L-ReLU
14900	0.299952	0.112367	TanH	14900	0.191568	0.775567	L-ReLU
15000	0.299952	0.112367	TanH	15000	0.191322	0.776	L-ReLU
15100	0.299952	0.112367	TanH	15100	0.191079	0.7765	L-ReLU
15200	0.299952	0.112367	TanH	15200	0.19084	0.777117	L-ReLU
15300	0.299952	0.112367	TanH	15300	0.190604	0.777417	L-ReLU
15400	0.299952	0.112367	TanH	15400	0.190372	0.777717	L-ReLU
15500	0.299952	0.112367	TanH	15500	0.190143	0.77805	L-ReLU
15600	0.299952	0.112367	TanH	15600	0.189917	0.778317	L-ReLU
15700	0.299952	0.112367	TanH	15700	0.189694	0.7786	L-ReLU
15800	0.299952	0.112367	TanH	15800	0.189474	0.7789	L-ReLU
15900	0.299952	0.112367	TanH	15900	0.189258	0.77925	L-ReLU
16000	0.299952	0.112367	TanH	16000	0.189044	0.77975	L-ReLU
16100	0.299952	0.112367	TanH	16100	0.188833	0.780133	L-ReLU
16200	0.299952	0.112367	TanH	16200	0.188625	0.7804	L-ReLU
16300	0.299952	0.112367	TanH	16300	0.18842	0.780733	L-ReLU
16400	0.299952	0.112367	TanH	16400	0.188218	0.781083	L-ReLU
16500	0.299952	0.112367	TanH	16500	0.188018	0.781333	L-ReLU
16600	0.299952	0.112367	TanH	16600	0.187821	0.781583	L-ReLU
16700	0.299952	0.112367	TanH	16700	0.187626	0.781883	L-ReLU
16800	0.299952	0.112367	TanH	16800	0.187434	0.782283	L-ReLU
16900	0.299952	0.112367	TanH	16900	0.187244	0.78265	L-ReLU
17000	0.299952	0.112367	TanH	17000	0.187057	0.782867	L-ReLU
17100	0.299952	0.112367	TanH	17100	0.186872	0.783233	L-ReLU
17200	0.299952	0.112367	TanH	17200	0.18669	0.783483	L-ReLU
17300	0.299952	0.112367	TanH	17300	0.18651	0.783717	L-ReLU

17400	0.299952	0.112367	TanH	17400	0.186332	0.784017	L-ReLU
17500	0.299952	0.112367	TanH	17500	0.186156	0.784333	L-ReLU
17600	0.299952	0.112367	TanH	17600	0.185982	0.784683	L-ReLU
17700	0.299952	0.112367	TanH	17700	0.185811	0.78505	L-ReLU
17800	0.299952	0.112367	TanH	17800	0.185642	0.785233	L-ReLU
17900	0.299952	0.112367	TanH	17900	0.185474	0.78545	L-ReLU
18000	0.299952	0.112367	TanH	18000	0.185309	0.78555	L-ReLU
18100	0.299952	0.112367	TanH	18100	0.185145	0.78575	L-ReLU
18200	0.299952	0.112367	TanH	18200	0.184984	0.785983	L-ReLU
18300	0.299952	0.112367	TanH	18300	0.184825	0.786183	L-ReLU
18400	0.299952	0.112367	TanH	18400	0.184667	0.786383	L-ReLU
18500	0.299952	0.112367	TanH	18500	0.184511	0.786483	L-ReLU
18600	0.299952	0.112367	TanH	18600	0.184357	0.786683	L-ReLU
18700	0.299952	0.112367	TanH	18700	0.184205	0.786883	L-ReLU
18800	0.299952	0.112367	TanH	18800	0.184054	0.78705	L-ReLU
18900	0.299952	0.112367	TanH	18900	0.183906	0.787317	L-ReLU
19000	0.299952	0.112367	TanH	19000	0.183758	0.787383	L-ReLU
19100	0.299952	0.112367	TanH	19100	0.183613	0.787533	L-ReLU
19200	0.299952	0.112367	TanH	19200	0.183469	0.7877	L-ReLU
19300	0.299952	0.112367	TanH	19300	0.183327	0.787883	L-ReLU
19400	0.299952	0.112367	TanH	19400	0.183186	0.788117	L-ReLU
19500	0.299952	0.112367	TanH	19500	0.183047	0.788217	L-ReLU
19600	0.299952	0.112367	TanH	19600	0.18291	0.788467	L-ReLU
19700	0.299952	0.112367	TanH	19700	0.182774	0.7886	L-ReLU
19800	0.299952	0.112367	TanH	19800	0.182639	0.788983	L-ReLU
19900	0.299952	0.112367	TanH	19900	0.182506	0.78915	L-ReLU
20000	0.299952	0.112367	TanH	20000	0.182374	0.789283	L-ReLU
20100	0.299952	0.112367	TanH	20100	0.182244	0.78945	L-ReLU
20200	0.299952	0.112367	TanH	20200	0.182115	0.789567	L-ReLU
20300	0.299952	0.112367	TanH	20300	0.181988	0.789733	L-ReLU
20400	0.299952	0.112367	TanH	20400	0.181861	0.789867	L-ReLU
20500	0.299952	0.112367	TanH	20500	0.181736	0.790083	L-ReLU
20600	0.299952	0.112367	TanH	20600	0.181613	0.7903	L-ReLU
20700	0.299952	0.112367	TanH	20700	0.181491	0.790383	L-ReLU
20800	0.299952	0.112367	TanH	20800	0.18137	0.7906	L-ReLU
20900	0.299952	0.112367	TanH	20900	0.18125	0.790783	L-ReLU
21000	0.299952	0.112367	TanH	21000	0.181131	0.791033	L-ReLU
21100	0.299952	0.112367	TanH	21100	0.181014	0.79125	L-ReLU
21200	0.299952	0.112367	TanH	21200	0.180898	0.7915	L-ReLU
21300	0.299952	0.112367	TanH	21300	0.180783	0.791667	L-ReLU
21400	0.299952	0.112367	TanH	21400	0.180669	0.791867	L-ReLU
21500	0.299952	0.112367	TanH	21500	0.180556	0.791917	L-ReLU
21600	0.299952	0.112367	TanH	21600	0.180444	0.79215	L-ReLU
21700	0.299952	0.112367	TanH	21700	0.180334	0.792433	L-ReLU

21800	0.299952	0.112367	TanH	21800	0.180224	0.7925	L-ReLU
21900	0.299952	0.112367	TanH	21900	0.180116	0.792683	L-ReLU
22000	0.299952	0.112367	TanH	22000	0.180008	0.792883	L-ReLU
22100	0.299952	0.112367	TanH	22100	0.179902	0.793017	L-ReLU
22200	0.299952	0.112367	TanH	22200	0.179797	0.793167	L-ReLU
22300	0.299952	0.112367	TanH	22300	0.179692	0.79345	L-ReLU
22400	0.299952	0.112367	TanH	22400	0.179589	0.793633	L-ReLU
22500	0.299952	0.112367	TanH	22500	0.179487	0.793717	L-ReLU
22600	0.299952	0.112367	TanH	22600	0.179385	0.7938	L-ReLU
22700	0.299952	0.112367	TanH	22700	0.179285	0.793967	L-ReLU
22800	0.299952	0.112367	TanH	22800	0.179185	0.794133	L-ReLU
22900	0.299952	0.112367	TanH	22900	0.179087	0.7943	L-ReLU
23000	0.299952	0.112367	TanH	23000	0.178989	0.7944	L-ReLU
23100	0.299952	0.112367	TanH	23100	0.178892	0.794583	L-ReLU
23200	0.299952	0.112367	TanH	23200	0.178797	0.794833	L-ReLU
23300	0.299952	0.112367	TanH	23300	0.178702	0.794967	L-ReLU
23400	0.299952	0.112367	TanH	23400	0.178607	0.79515	L-ReLU
23500	0.299952	0.112367	TanH	23500	0.178514	0.795317	L-ReLU
23600	0.299952	0.112367	TanH	23600	0.178422	0.79555	L-ReLU
23700	0.299952	0.112367	TanH	23700	0.17833	0.79575	L-ReLU
23800	0.299952	0.112367	TanH	23800	0.178239	0.795917	L-ReLU
23900	0.299952	0.112367	TanH	23900	0.178149	0.796083	L-ReLU
24000	0.299952	0.112367	TanH	24000	0.17806	0.796183	L-ReLU
24100	0.299952	0.112367	TanH	24100	0.177971	0.796317	L-ReLU
24200	0.299952	0.112367	TanH	24200	0.177883	0.796433	L-ReLU
24300	0.299952	0.112367	TanH	24300	0.177797	0.796433	L-ReLU
24400	0.299952	0.112367	TanH	24400	0.305976	0.09035	TanH
24500	0.299952	0.112367	TanH	24500	0.305827	0.09035	TanH
24600	0.299952	0.112367	TanH	24600	0.300397	0.112367	Sigmoid
24700	0.299952	0.112367	TanH	24700	0.300394	0.112367	Sigmoid
24800	0.299952	0.112367	TanH	24800	0.300392	0.112367	ReLU
24900	0.299952	0.112367	TanH	24900	0.305507	0.09035	TanH
25000	0.299952	0.112367	TanH	25000	0.300389	0.112367	ReLU
25100	0.299952	0.112367	TanH	25100	0.300386	0.112367	ReLU
25200	0.299952	0.112367	TanH	25200	0.30552	0.09035	TanH
25300	0.299952	0.112367	TanH	25300	0.181367	0.7817	L-ReLU
25400	0.299952	0.112367	TanH	25400	0.177724	0.795633	L-ReLU
25500	0.299952	0.112367	TanH	25500	0.177319	0.796867	L-ReLU
25600	0.299952	0.112367	TanH	25600	0.177175	0.797383	L-ReLU
25700	0.299952	0.112367	TanH	25700	0.177078	0.797417	L-ReLU
25800	0.299952	0.112367	TanH	25800	0.176994	0.797517	L-ReLU
25900	0.299952	0.112367	TanH	25900	0.176914	0.797567	L-ReLU
26000	0.299952	0.112367	TanH	26000	0.176837	0.7977	L-ReLU
26100	0.299952	0.112367	TanH	26100	0.17676	0.797783	L-ReLU

26200	0.299952	0.112367	TanH	26200	0.176685	0.797817	L-ReLU
26300	0.299952	0.112367	TanH	26300	0.17661	0.798017	L-ReLU
26400	0.299952	0.112367	TanH	26400	0.176535	0.798183	L-ReLU
26500	0.299952	0.112367	TanH	26500	0.176461	0.798283	L-ReLU
26600	0.299952	0.112367	TanH	26600	0.176388	0.798433	L-ReLU
26700	0.299952	0.112367	TanH	26700	0.176315	0.7985	L-ReLU
26800	0.299952	0.112367	TanH	26800	0.176243	0.798567	L-ReLU
26900	0.299952	0.112367	TanH	26900	0.176172	0.79865	L-ReLU
27000	0.299952	0.112367	TanH	27000	0.176101	0.798717	L-ReLU
27100	0.299952	0.112367	TanH	27100	0.17603	0.798883	L-ReLU
27200	0.299952	0.112367	TanH	27200	0.17596	0.798917	L-ReLU
27300	0.299952	0.112367	TanH	27300	0.175891	0.799033	L-ReLU
27400	0.299952	0.112367	TanH	27400	0.175822	0.799033	L-ReLU
27500	0.299952	0.112367	TanH	27500	0.300391	0.112367	ReLU
27600	0.299952	0.112367	TanH	27600	0.30039	0.112367	ReLU
27700	0.299952	0.112367	TanH	27700	0.306059	0.09035	TanH
27800	0.299952	0.112367	TanH	27800	0.300387	0.112367	Sigmoid
27900	0.299952	0.112367	TanH	27900	0.300384	0.112367	Sigmoid
28000	0.299952	0.112367	TanH	28000	0.30605	0.09035	TanH
28100	0.299952	0.112367	TanH	28100	0.30459	0.172117	Softmax
28200	0.299952	0.112367	TanH	28200	0.30448	0.172067	Softmax
28300	0.299952	0.112367	TanH	28300	0.300382	0.112367	Sigmoid
28400	0.299952	0.112367	TanH	28400	0.305761	0.09035	TanH
28500	0.299952	0.112367	TanH	28500	0.30444	0.171867	Softmax
28600	0.299952	0.112367	TanH	28600	0.304333	0.17175	Softmax
28700	0.299952	0.112367	TanH	28700	0.30038	0.112367	Sigmoid
28800	0.299952	0.112367	TanH	28800	0.304301	0.171767	Softmax
28900	0.299952	0.112367	TanH	28900	0.304196	0.17175	Softmax
29000	0.299952	0.112367	TanH	29000	0.30038	0.112367	Sigmoid
29100	0.299952	0.112367	TanH	29100	0.300379	0.112367	ReLU
29200	0.299952	0.112367	TanH	29200	0.300378	0.112367	Sigmoid
29300	0.299952	0.112367	TanH	29300	0.300377	0.112367	Sigmoid
29400	0.299952	0.112367	TanH	29400	0.300375	0.112367	ReLU
29500	0.299952	0.112367	TanH	29500	0.305805	0.09035	TanH
29600	0.299952	0.112367	TanH	29600	0.300373	0.112367	Sigmoid
29700	0.299952	0.112367	TanH	29700	0.30037	0.112367	Sigmoid
29800	0.299952	0.112367	TanH	29800	0.300369	0.112367	ReLU
29900	0.299952	0.112367	TanH	29900	0.179145	0.78605	L-ReLU

TanH – Random, L-ReLU suppressed, 30,000 epochs

Epoch	Cost	Acc	AF
0	0.390736	0.113733	TanH
100	0.30003	0.112367	Sigmoid
200	0.300034	0.112367	ReLU
300	0.300055	0.112367	ReLU
400	0.299952	0.112367	TanH
500	0.300093	0.112367	ReLU
600	0.296156	0.180783	Softmax
700	0.294181	0.217583	Softmax
800	0.29285	0.2313	Softmax
900	0.291895	0.240033	Softmax
1000	0.291175	0.246467	Softmax
1100	0.290611	0.252517	Softmax
1200	0.290154	0.263817	Softmax
1300	0.289776	0.268517	Softmax
1400	0.289455	0.270717	Softmax
1500	0.28918	0.272967	Softmax
1600	0.288941	0.275317	Softmax
1700	0.28873	0.276767	Softmax
1800	0.288542	0.277683	Softmax
1900	0.288373	0.277633	Softmax
2000	0.300991	0.098633	ReLU
2100	0.300959	0.098633	Sigmoid
2200	0.300923	0.098633	ReLU
2300	0.287583	0.28045	Softmax
2400	0.287423	0.281233	Softmax
2500	0.287276	0.282633	Softmax
2600	0.287141	0.2838	Softmax
2700	0.287015	0.284883	Softmax
2800	0.286898	0.286067	Softmax
2900	0.286789	0.287083	Softmax
3000	0.286686	0.28805	Softmax
3100	0.28659	0.2891	Softmax
3200	0.286499	0.290383	Softmax
3300	0.286413	0.291517	Softmax
3400	0.286331	0.292717	Softmax
3500	0.286254	0.29375	Softmax
3600	0.28618	0.295033	Softmax
3700	0.286109	0.296833	Softmax
3800	0.286042	0.298517	Softmax
3900	0.285978	0.301033	Softmax
4000	0.285916	0.30495	Softmax

4100	0.285857	0.306933	Softmax
4200	0.285799	0.307133	Softmax
4300	0.285745	0.307567	Softmax
4400	0.285692	0.308033	Softmax
4500	0.28564	0.3085	Softmax
4600	0.285591	0.308617	Softmax
4700	0.285543	0.308833	Softmax
4800	0.285497	0.309117	Softmax
4900	0.285452	0.309517	Softmax
5000	0.285408	0.309667	Softmax
5100	0.285366	0.309917	Softmax
5200	0.285325	0.310083	Softmax
5300	0.285285	0.310283	Softmax
5400	0.285246	0.310667	Softmax
5500	0.285209	0.31105	Softmax
5600	0.285172	0.311383	Softmax
5700	0.285136	0.311467	Softmax
5800	0.285101	0.311767	Softmax
5900	0.285067	0.312017	Softmax
6000	0.285034	0.312483	Softmax
6100	0.285001	0.313	Softmax
6200	0.284969	0.31325	Softmax
6300	0.284938	0.313733	Softmax
6400	0.284908	0.314367	Softmax
6500	0.284878	0.314767	Softmax
6600	0.284849	0.315517	Softmax
6700	0.284821	0.31635	Softmax
6800	0.284793	0.31705	Softmax
6900	0.284766	0.318183	Softmax
7000	0.284739	0.319817	Softmax
7100	0.284713	0.322733	Softmax
7200	0.284687	0.318567	Softmax
7300	0.301222	0.098633	Sigmoid
7400	0.284502	0.318583	Softmax
7500	0.284477	0.317867	Softmax
7600	0.301197	0.098633	Sigmoid
7700	0.301184	0.098633	Sigmoid
7800	0.28423	0.319433	Softmax
7900	0.284205	0.3193	Softmax
8000	0.299953	0.112367	TanH
8100	0.301151	0.098633	Sigmoid
8200	0.301139	0.098633	Sigmoid
8300	0.301129	0.098633	ReLU
8400	0.299953	0.112367	TanH

8500	0.299952	0.112367	TanH
8600	0.301109	0.098633	ReLU
8700	0.301102	0.098633	Sigmoid
8800	0.299952	0.112367	TanH
8900	0.299952	0.112367	TanH
9000	0.301083	0.098633	Sigmoid
9100	0.28337	0.3244	Softmax
9200	0.283335	0.324667	Softmax
9300	0.283301	0.3251	Softmax
9400	0.283267	0.325783	Softmax
9500	0.283234	0.326367	Softmax
9600	0.283202	0.327383	Softmax
9700	0.283171	0.32865	Softmax
9800	0.28314	0.33015	Softmax
9900	0.283109	0.339717	Softmax
10000	0.283079	0.338833	Softmax
10100	0.301103	0.098633	ReLU
10200	0.299953	0.112367	TanH
10300	0.299952	0.112367	TanH
10400	0.301088	0.098633	Sigmoid
10500	0.301081	0.098633	Sigmoid
10600	0.299952	0.112367	TanH
10700	0.299952	0.112367	TanH
10800	0.299952	0.112367	TanH
10900	0.299952	0.112367	TanH
11000	0.282834	0.340567	Softmax
11100	0.282797	0.340717	Softmax
11200	0.282761	0.340967	Softmax
11300	0.282725	0.34115	Softmax
11400	0.28269	0.3411	Softmax
11500	0.301068	0.098633	Sigmoid
11600	0.301062	0.098633	Sigmoid
11700	0.301055	0.098633	Sigmoid
11800	0.299953	0.112367	TanH
11900	0.299952	0.112367	TanH
12000	0.301041	0.098633	ReLU
12100	0.299952	0.112367	TanH
12200	0.299952	0.112367	TanH
12300	0.301031	0.098633	ReLU
12400	0.301026	0.098633	ReLU
12500	0.301021	0.098633	ReLU
12600	0.301016	0.098633	ReLU
12700	0.282656	0.343517	Softmax
12800	0.282614	0.344083	Softmax

12900	0.282573	0.3442	Softmax
13000	0.282533	0.3444	Softmax
13100	0.282494	0.344417	Softmax
13200	0.282455	0.344383	Softmax
13300	0.299952	0.112367	TanH
13400	0.299953	0.112367	TanH
13500	0.282409	0.344833	Softmax
13600	0.28237	0.34505	Softmax
13700	0.282333	0.345183	Softmax
13800	0.282296	0.345167	Softmax
13900	0.282288	0.34515	Softmax
14000	0.301031	0.098633	ReLU
14100	0.301026	0.098633	ReLU
14200	0.301021	0.098633	Sigmoid
14300	0.299953	0.112367	TanH
14400	0.299952	0.112367	TanH
14500	0.282328	0.344983	Softmax
14600	0.282289	0.345167	Softmax
14700	0.282251	0.345417	Softmax
14800	0.282213	0.3454	Softmax
14900	0.282195	0.345533	Softmax
15000	0.282158	0.345667	Softmax
15100	0.282122	0.345733	Softmax
15200	0.282086	0.34585	Softmax
15300	0.28205	0.346067	Softmax
15400	0.282016	0.346233	Softmax
15500	0.281981	0.346383	Softmax
15600	0.281947	0.346567	Softmax
15700	0.281914	0.346567	Softmax
15800	0.281904	0.346717	Softmax
15900	0.281871	0.3468	Softmax
16000	0.281839	0.346883	Softmax
16100	0.281807	0.3468	Softmax
16200	0.299953	0.112367	TanH
16300	0.301067	0.098633	Sigmoid
16400	0.301062	0.098633	ReLU
16500	0.299953	0.112367	TanH
16600	0.299952	0.112367	TanH
16700	0.299952	0.112367	TanH
16800	0.281845	0.34695	Softmax
16900	0.281812	0.347233	Softmax
17000	0.281779	0.347267	Softmax
17100	0.281747	0.347467	Softmax
17200	0.281715	0.347617	Softmax

17300	0.281683	0.347617	Softmax
17400	0.301071	0.098633	ReLU
17500	0.301067	0.098633	Sigmoid
17600	0.301062	0.098633	ReLU
17700	0.301058	0.098633	ReLU
17800	0.281628	0.348333	Softmax
17900	0.281597	0.348483	Softmax
18000	0.281566	0.3486	Softmax
18100	0.281536	0.348817	Softmax
18200	0.281506	0.348967	Softmax
18300	0.281477	0.349183	Softmax
18400	0.281448	0.349283	Softmax
18500	0.281419	0.349533	Softmax
18600	0.281391	0.349767	Softmax
18700	0.281363	0.349933	Softmax
18800	0.281335	0.350117	Softmax
18900	0.281307	0.3502	Softmax
19000	0.28128	0.350517	Softmax
19100	0.281253	0.350817	Softmax
19200	0.281227	0.350983	Softmax
19300	0.281201	0.351117	Softmax
19400	0.281175	0.351383	Softmax
19500	0.281149	0.3516	Softmax
19600	0.281124	0.35165	Softmax
19700	0.281099	0.351617	Softmax
19800	0.301131	0.104417	ReLU
19900	0.301127	0.104417	Sigmoid
20000	0.301123	0.104417	Sigmoid
20100	0.299953	0.112367	TanH
20200	0.299952	0.112367	TanH
20300	0.299952	0.112367	TanH
20400	0.301111	0.104417	Sigmoid
20500	0.281136	0.351017	Softmax
20600	0.28111	0.351167	Softmax
20700	0.281085	0.351283	Softmax
20800	0.281059	0.3514	Softmax
20900	0.281034	0.351767	Softmax
21000	0.28101	0.351833	Softmax
21100	0.280985	0.352083	Softmax
21200	0.280961	0.3522	Softmax
21300	0.280937	0.352267	Softmax
21400	0.280913	0.352517	Softmax
21500	0.28089	0.352633	Softmax
21600	0.280866	0.352883	Softmax

21700	0.280843	0.353067	Softmax
21800	0.28082	0.353333	Softmax
21900	0.280798	0.353333	Softmax
22000	0.28079	0.3536	Softmax
22100	0.280768	0.353817	Softmax
22200	0.280746	0.35395	Softmax
22300	0.280724	0.354	Softmax
22400	0.280702	0.354233	Softmax
22500	0.28068	0.35445	Softmax
22600	0.280659	0.354833	Softmax
22700	0.280638	0.35505	Softmax
22800	0.280617	0.355467	Softmax
22900	0.280596	0.355817	Softmax
23000	0.280576	0.356183	Softmax
23100	0.280555	0.356633	Softmax
23200	0.280535	0.357217	Softmax
23300	0.280515	0.357717	Softmax
23400	0.280495	0.3583	Softmax
23500	0.280476	0.3591	Softmax
23600	0.280456	0.361717	Softmax
23700	0.280437	0.368683	Softmax
23800	0.280418	0.36855	Softmax
23900	0.301211	0.104417	ReLU
24000	0.299953	0.112367	TanH
24100	0.299952	0.112367	TanH
24200	0.280458	0.368217	Softmax
24300	0.280438	0.368183	Softmax
24400	0.280436	0.368133	Softmax
24500	0.299952	0.112367	TanH
24600	0.280428	0.368167	Softmax
24700	0.280408	0.3681	Softmax
24800	0.301197	0.104417	Sigmoid
24900	0.301194	0.104417	Sigmoid
25000	0.280397	0.36805	Softmax
25100	0.280377	0.36815	Softmax
25200	0.280358	0.368067	Softmax
25300	0.299953	0.112367	TanH
25400	0.299953	0.112367	TanH
25500	0.301188	0.104417	Sigmoid
25600	0.280339	0.368217	Softmax
25700	0.28032	0.36825	Softmax
25800	0.280301	0.368133	Softmax
25900	0.280295	0.3682	Softmax
26000	0.280276	0.368167	Softmax

26100	0.301188	0.104417	ReLU
26200	0.301184	0.104417	Sigmoid
26300	0.299953	0.112367	TanH
26400	0.299952	0.112367	TanH
26500	0.280323	0.3678	Softmax
26600	0.280303	0.367817	Softmax
26700	0.280284	0.367817	Softmax
26800	0.301182	0.104417	ReLU
26900	0.30118	0.104417	Sigmoid
27000	0.301177	0.104417	Sigmoid
27100	0.280273	0.367567	Softmax
27200	0.280254	0.367567	Softmax
27300	0.299953	0.112367	TanH
27400	0.30117	0.104417	ReLU
27500	0.301167	0.104417	ReLU
27600	0.301164	0.104417	Sigmoid
27700	0.301161	0.104417	ReLU
27800	0.280229	0.3677	Softmax
27900	0.28021	0.3677	Softmax
28000	0.301159	0.104417	ReLU
28100	0.280205	0.367683	Softmax
28200	0.280186	0.367767	Softmax
28300	0.280168	0.367783	Softmax
28400	0.280149	0.36785	Softmax
28500	0.280131	0.367883	Softmax
28600	0.280113	0.367883	Softmax
28700	0.280107	0.36785	Softmax
28800	0.280107	0.3677	Softmax
28900	0.280105	0.367783	Softmax
29000	0.280087	0.367867	Softmax
29100	0.280069	0.367767	Softmax
29200	0.301166	0.104417	ReLU
29300	0.301164	0.104417	ReLU
29400	0.299953	0.112367	TanH
29500	0.299953	0.112367	TanH
29600	0.299953	0.112367	TanH
29700	0.301156	0.104417	ReLU
29800	0.299953	0.112367	TanH
29900	0.299952	0.112367	TanH

Iris – Baseline Leaky ReLU

Iteration	Cost	Acc	AF
0	0.471127	0.555556	L-ReLU
1	0.467149	0.555556	L-ReLU
2	0.463491	0.555556	L-ReLU
3	0.460195	0.555556	L-ReLU
4	0.457021	0.555556	L-ReLU
5	0.454096	0.555556	L-ReLU
6	0.451396	0.555556	L-ReLU
7	0.448903	0.555556	L-ReLU
8	0.446598	0.555556	L-ReLU
9	0.444464	0.555556	L-ReLU
10	0.442486	0.555556	L-ReLU
11	0.440651	0.555556	L-ReLU
12	0.438947	0.555556	L-ReLU
13	0.437362	0.555556	L-ReLU
14	0.435886	0.555556	L-ReLU
15	0.43451	0.555556	L-ReLU
16	0.433227	0.555556	L-ReLU
17	0.432027	0.555556	L-ReLU
18	0.430906	0.555556	L-ReLU
19	0.429856	0.555556	L-ReLU
20	0.428871	0.555556	L-ReLU
21	0.427948	0.555556	L-ReLU
22	0.42708	0.555556	L-ReLU
23	0.426264	0.555556	L-ReLU
24	0.425496	0.555556	L-ReLU
25	0.424773	0.555556	L-ReLU
26	0.42409	0.555556	L-ReLU
27	0.423446	0.555556	L-ReLU
28	0.422837	0.555556	L-ReLU
29	0.422262	0.555556	L-ReLU
30	0.421716	0.555556	L-ReLU
31	0.4212	0.555556	L-ReLU
32	0.42071	0.555556	L-ReLU
33	0.420245	0.555556	L-ReLU
34	0.419803	0.555556	L-ReLU
35	0.419383	0.555556	L-ReLU
36	0.418984	0.555556	L-ReLU
37	0.418603	0.555556	L-ReLU
38	0.41824	0.555556	L-ReLU
39	0.417894	0.555556	L-ReLU
40	0.417564	0.555556	L-ReLU
41	0.417249	0.555556	L-ReLU

Iris – Baseline, ReLU

Iteration	Cost	Acc	AF
0	0.471405	0.555556	ReLU
1	0.46741	0.555556	ReLU
2	0.463737	0.555556	ReLU
3	0.460355	0.555556	ReLU
4	0.457239	0.555556	ReLU
5	0.454365	0.555556	ReLU
6	0.451711	0.555556	ReLU
7	0.449258	0.555556	ReLU
8	0.446988	0.555556	ReLU
9	0.444884	0.555556	ReLU
10	0.442933	0.555556	ReLU
11	0.441121	0.555556	ReLU
12	0.439437	0.555556	ReLU
13	0.437868	0.555556	ReLU
14	0.436407	0.555556	ReLU
15	0.435043	0.555556	ReLU
16	0.433769	0.555556	ReLU
17	0.432577	0.555556	ReLU
18	0.431462	0.555556	ReLU
19	0.430416	0.555556	ReLU
20	0.429435	0.555556	ReLU
21	0.428513	0.555556	ReLU
22	0.427646	0.555556	ReLU
23	0.426831	0.555556	ReLU
24	0.426062	0.555556	ReLU
25	0.425337	0.555556	ReLU
26	0.424652	0.555556	ReLU
27	0.424005	0.555556	ReLU
28	0.423393	0.555556	ReLU
29	0.422814	0.555556	ReLU
30	0.422264	0.555556	ReLU
31	0.421743	0.555556	ReLU
32	0.421249	0.555556	ReLU
33	0.420779	0.555556	ReLU
34	0.420332	0.555556	ReLU
35	0.419907	0.555556	ReLU
36	0.419502	0.555556	ReLU
37	0.419116	0.555556	ReLU
38	0.418748	0.555556	ReLU
39	0.418397	0.555556	ReLU
40	0.418061	0.555556	ReLU
41	0.41774	0.555556	ReLU

42	0.416947	0.555556	L-ReLU	42	0.417433	0.555556	ReLU
43	0.416659	0.555556	L-ReLU	43	0.41714	0.555556	ReLU
44	0.416383	0.555556	L-ReLU	44	0.416858	0.555556	ReLU
45	0.416119	0.555556	L-ReLU	45	0.416589	0.555556	ReLU
46	0.415865	0.555556	L-ReLU	46	0.41633	0.555556	ReLU
47	0.415622	0.555556	L-ReLU	47	0.416082	0.555556	ReLU
48	0.415389	0.555556	L-ReLU	48	0.415843	0.555556	ReLU
49	0.415165	0.555556	L-ReLU	49	0.415614	0.555556	ReLU
50	0.41495	0.555556	L-ReLU	50	0.415394	0.555556	ReLU
51	0.414744	0.555556	L-ReLU	51	0.415182	0.555556	ReLU
52	0.414545	0.555556	L-ReLU	52	0.414979	0.555556	ReLU
53	0.414354	0.555556	L-ReLU	53	0.414782	0.555556	ReLU
54	0.414169	0.555556	L-ReLU	54	0.414593	0.555556	ReLU
55	0.413992	0.555556	L-ReLU	55	0.414411	0.555556	ReLU
56	0.413821	0.555556	L-ReLU	56	0.414236	0.555556	ReLU
57	0.413656	0.555556	L-ReLU	57	0.414066	0.555556	ReLU
58	0.413497	0.555556	L-ReLU	58	0.413903	0.555556	ReLU
59	0.413343	0.555556	L-ReLU	59	0.413745	0.555556	ReLU
60	0.413194	0.555556	L-ReLU	60	0.413592	0.555556	ReLU
61	0.413051	0.555556	L-ReLU	61	0.413444	0.555556	ReLU
62	0.412913	0.555556	L-ReLU	62	0.413302	0.555556	ReLU
63	0.412778	0.555556	L-ReLU	63	0.413164	0.555556	ReLU
64	0.412649	0.555556	L-ReLU	64	0.41303	0.555556	ReLU
65	0.412523	0.555556	L-ReLU	65	0.412901	0.555556	ReLU
66	0.412402	0.555556	L-ReLU	66	0.412775	0.555556	ReLU
67	0.412284	0.555556	L-ReLU	67	0.412654	0.555556	ReLU
68	0.41217	0.555556	L-ReLU	68	0.412536	0.555556	ReLU
69	0.412059	0.555556	L-ReLU	69	0.412422	0.555556	ReLU
70	0.411952	0.555556	L-ReLU	70	0.412311	0.555556	ReLU
71	0.411848	0.555556	L-ReLU	71	0.412204	0.555556	ReLU
72	0.411747	0.555556	L-ReLU	72	0.4121	0.555556	ReLU
73	0.411649	0.555556	L-ReLU	73	0.411999	0.555556	ReLU
74	0.411554	0.555556	L-ReLU	74	0.4119	0.555556	ReLU
75	0.411461	0.555556	L-ReLU	75	0.411805	0.555556	ReLU
76	0.411371	0.555556	L-ReLU	76	0.411712	0.555556	ReLU
77	0.411284	0.555556	L-ReLU	77	0.411621	0.555556	ReLU
78	0.411199	0.555556	L-ReLU	78	0.411533	0.555556	ReLU
79	0.411116	0.555556	L-ReLU	79	0.411448	0.555556	ReLU
80	0.411036	0.555556	L-ReLU	80	0.411365	0.555556	ReLU
81	0.410958	0.555556	L-ReLU	81	0.411284	0.555556	ReLU
82	0.410881	0.555556	L-ReLU	82	0.411205	0.555556	ReLU
83	0.410807	0.555556	L-ReLU	83	0.411128	0.555556	ReLU
84	0.410735	0.555556	L-ReLU	84	0.411053	0.555556	ReLU
85	0.410664	0.555556	L-ReLU	85	0.41098	0.555556	ReLU

86	0.410596	0.555556	L-ReLU	86	0.410909	0.555556	ReLU
87	0.410529	0.555556	L-ReLU	87	0.41084	0.555556	ReLU
88	0.410463	0.555556	L-ReLU	88	0.410772	0.555556	ReLU
89	0.4104	0.555556	L-ReLU	89	0.410706	0.555556	ReLU
90	0.410337	0.555556	L-ReLU	90	0.410642	0.555556	ReLU
91	0.410277	0.555556	L-ReLU	91	0.410579	0.555556	ReLU
92	0.410217	0.555556	L-ReLU	92	0.410517	0.555556	ReLU
93	0.410159	0.555556	L-ReLU	93	0.410457	0.555556	ReLU
94	0.410103	0.555556	L-ReLU	94	0.410399	0.555556	ReLU
95	0.410048	0.555556	L-ReLU	95	0.410342	0.555556	ReLU
96	0.409994	0.555556	L-ReLU	96	0.410286	0.555556	ReLU
97	0.409941	0.555556	L-ReLU	97	0.410231	0.555556	ReLU
98	0.40989	0.555556	L-ReLU	98	0.410178	0.555556	ReLU
99	0.409839	0.555556	L-ReLU	99	0.410126	0.555556	ReLU
100	0.40979	0.555556	L-ReLU	100	0.410074	0.555556	ReLU
101	0.409742	0.555556	L-ReLU	101	0.410025	0.555556	ReLU
102	0.409694	0.555556	L-ReLU	102	0.409976	0.555556	ReLU
103	0.409648	0.555556	L-ReLU	103	0.409928	0.555556	ReLU
104	0.409603	0.555556	L-ReLU	104	0.409881	0.555556	ReLU
105	0.409559	0.555556	L-ReLU	105	0.409835	0.555556	ReLU
106	0.409516	0.555556	L-ReLU	106	0.40979	0.555556	ReLU
107	0.409473	0.555556	L-ReLU	107	0.409746	0.555556	ReLU
108	0.409432	0.555556	L-ReLU	108	0.409703	0.555556	ReLU
109	0.409391	0.555556	L-ReLU	109	0.409661	0.555556	ReLU
110	0.409351	0.555556	L-ReLU	110	0.40962	0.555556	ReLU
111	0.409312	0.555556	L-ReLU	111	0.409579	0.555556	ReLU
112	0.409274	0.555556	L-ReLU	112	0.409539	0.555556	ReLU
113	0.409236	0.555556	L-ReLU	113	0.4095	0.555556	ReLU
114	0.409199	0.555556	L-ReLU	114	0.409462	0.555556	ReLU
115	0.409163	0.555556	L-ReLU	115	0.409425	0.555556	ReLU
116	0.409128	0.555556	L-ReLU	116	0.409388	0.555556	ReLU
117	0.409093	0.555556	L-ReLU	117	0.409352	0.555556	ReLU
118	0.409059	0.555556	L-ReLU	118	0.409317	0.555556	ReLU
119	0.409025	0.555556	L-ReLU	119	0.409282	0.555556	ReLU
120	0.408992	0.555556	L-ReLU	120	0.409248	0.555556	ReLU
121	0.40896	0.555556	L-ReLU	121	0.409214	0.555556	ReLU
122	0.408929	0.555556	L-ReLU	122	0.409182	0.555556	ReLU
123	0.408897	0.555556	L-ReLU	123	0.409149	0.555556	ReLU
124	0.408867	0.555556	L-ReLU	124	0.409118	0.555556	ReLU
125	0.408837	0.555556	L-ReLU	125	0.409087	0.555556	ReLU
126	0.408808	0.555556	L-ReLU	126	0.409056	0.555556	ReLU
127	0.408779	0.555556	L-ReLU	127	0.409026	0.555556	ReLU
128	0.40875	0.555556	L-ReLU	128	0.408997	0.555556	ReLU
129	0.408722	0.555556	L-ReLU	129	0.408968	0.555556	ReLU

130	0.408695	0.555556	L-ReLU	130	0.408939	0.555556	ReLU
131	0.408668	0.555556	L-ReLU	131	0.408911	0.555556	ReLU
132	0.408641	0.555556	L-ReLU	132	0.408884	0.555556	ReLU
133	0.408615	0.555556	L-ReLU	133	0.408857	0.555556	ReLU
134	0.40859	0.555556	L-ReLU	134	0.40883	0.555556	ReLU
135	0.408564	0.555556	L-ReLU	135	0.408804	0.555556	ReLU
136	0.40854	0.555556	L-ReLU	136	0.408778	0.555556	ReLU
137	0.408515	0.555556	L-ReLU	137	0.408753	0.555556	ReLU
138	0.408491	0.555556	L-ReLU	138	0.408728	0.555556	ReLU
139	0.408468	0.555556	L-ReLU	139	0.408704	0.555556	ReLU
140	0.408444	0.555556	L-ReLU	140	0.40868	0.555556	ReLU
141	0.408422	0.555556	L-ReLU	141	0.408656	0.555556	ReLU
142	0.408399	0.555556	L-ReLU	142	0.408633	0.555556	ReLU
143	0.408377	0.555556	L-ReLU	143	0.40861	0.555556	ReLU
144	0.408355	0.555556	L-ReLU	144	0.408587	0.555556	ReLU
145	0.408334	0.555556	L-ReLU	145	0.408565	0.555556	ReLU
146	0.408312	0.555556	L-ReLU	146	0.408543	0.555556	ReLU
147	0.408292	0.555556	L-ReLU	147	0.408521	0.555556	ReLU
148	0.408271	0.555556	L-ReLU	148	0.4085	0.555556	ReLU
149	0.408251	0.555556	L-ReLU	149	0.408479	0.555556	ReLU
150	0.408231	0.555556	L-ReLU	150	0.408458	0.555556	ReLU
151	0.408211	0.555556	L-ReLU	151	0.408438	0.555556	ReLU
152	0.408192	0.555556	L-ReLU	152	0.408418	0.555556	ReLU
153	0.408173	0.555556	L-ReLU	153	0.408398	0.555556	ReLU
154	0.408154	0.555556	L-ReLU	154	0.408379	0.555556	ReLU
155	0.408136	0.555556	L-ReLU	155	0.40836	0.555556	ReLU
156	0.408118	0.555556	L-ReLU	156	0.408341	0.555556	ReLU
157	0.4081	0.555556	L-ReLU	157	0.408322	0.555556	ReLU
158	0.408082	0.555556	L-ReLU	158	0.408304	0.555556	ReLU
159	0.408065	0.555556	L-ReLU	159	0.408286	0.555556	ReLU
160	0.408047	0.555556	L-ReLU	160	0.408268	0.555556	ReLU
161	0.40803	0.555556	L-ReLU	161	0.40825	0.555556	ReLU
162	0.408014	0.555556	L-ReLU	162	0.408233	0.555556	ReLU
163	0.407997	0.555556	L-ReLU	163	0.408216	0.555556	ReLU
164	0.407981	0.555556	L-ReLU	164	0.408199	0.555556	ReLU
165	0.407965	0.555556	L-ReLU	165	0.408182	0.555556	ReLU
166	0.407949	0.555556	L-ReLU	166	0.408166	0.555556	ReLU
167	0.407934	0.555556	L-ReLU	167	0.40815	0.555556	ReLU
168	0.407918	0.555556	L-ReLU	168	0.408134	0.555556	ReLU
169	0.407903	0.555556	L-ReLU	169	0.408118	0.555556	ReLU
170	0.407888	0.555556	L-ReLU	170	0.408103	0.555556	ReLU
171	0.407873	0.555556	L-ReLU	171	0.408087	0.555556	ReLU
172	0.407859	0.555556	L-ReLU	172	0.408072	0.555556	ReLU
173	0.407844	0.555556	L-ReLU	173	0.408057	0.555556	ReLU

174	0.40783	0.555556	L-ReLU	174	0.408042	0.555556	ReLU
175	0.407816	0.555556	L-ReLU	175	0.408028	0.555556	ReLU
176	0.407802	0.555556	L-ReLU	176	0.408013	0.555556	ReLU
177	0.407788	0.555556	L-ReLU	177	0.407999	0.555556	ReLU
178	0.407775	0.555556	L-ReLU	178	0.407985	0.555556	ReLU
179	0.407761	0.555556	L-ReLU	179	0.407971	0.555556	ReLU
180	0.407748	0.555556	L-ReLU	180	0.407958	0.555556	ReLU
181	0.407735	0.555556	L-ReLU	181	0.407944	0.555556	ReLU
182	0.407722	0.555556	L-ReLU	182	0.407931	0.555556	ReLU
183	0.40771	0.555556	L-ReLU	183	0.407918	0.555556	ReLU
184	0.407697	0.555556	L-ReLU	184	0.407905	0.555556	ReLU
185	0.407685	0.555556	L-ReLU	185	0.407892	0.555556	ReLU
186	0.407673	0.555556	L-ReLU	186	0.407879	0.555556	ReLU
187	0.40766	0.555556	L-ReLU	187	0.407867	0.555556	ReLU
188	0.407649	0.555556	L-ReLU	188	0.407854	0.555556	ReLU
189	0.407637	0.555556	L-ReLU	189	0.407842	0.555556	ReLU
190	0.407625	0.555556	L-ReLU	190	0.40783	0.555556	ReLU
191	0.407614	0.555556	L-ReLU	191	0.407818	0.555556	ReLU
192	0.407602	0.555556	L-ReLU	192	0.407806	0.555556	ReLU
193	0.407591	0.555556	L-ReLU	193	0.407794	0.555556	ReLU
194	0.40758	0.555556	L-ReLU	194	0.407783	0.555556	ReLU
195	0.407569	0.555556	L-ReLU	195	0.407771	0.555556	ReLU
196	0.407558	0.555556	L-ReLU	196	0.40776	0.555556	ReLU
197	0.407547	0.555556	L-ReLU	197	0.407749	0.555556	ReLU
198	0.407537	0.555556	L-ReLU	198	0.407738	0.555556	ReLU
199	0.407526	0.555556	L-ReLU	199	0.407727	0.555556	ReLU
200	0.407516	0.555556	L-ReLU	200	0.407716	0.555556	ReLU
201	0.407505	0.555556	L-ReLU	201	0.407706	0.555556	ReLU
202	0.407495	0.555556	L-ReLU	202	0.407695	0.555556	ReLU
203	0.407485	0.555556	L-ReLU	203	0.407685	0.555556	ReLU
204	0.407475	0.555556	L-ReLU	204	0.407674	0.555556	ReLU
205	0.407465	0.555556	L-ReLU	205	0.407664	0.555556	ReLU
206	0.407456	0.555556	L-ReLU	206	0.407654	0.555556	ReLU
207	0.407446	0.555556	L-ReLU	207	0.407644	0.555556	ReLU
208	0.407437	0.555556	L-ReLU	208	0.407634	0.555556	ReLU
209	0.407427	0.555556	L-ReLU	209	0.407624	0.555556	ReLU
210	0.407418	0.555556	L-ReLU	210	0.407615	0.555556	ReLU
211	0.407409	0.555556	L-ReLU	211	0.407605	0.555556	ReLU
212	0.4074	0.555556	L-ReLU	212	0.407596	0.555556	ReLU
213	0.407391	0.555556	L-ReLU	213	0.407586	0.555556	ReLU
214	0.407382	0.555556	L-ReLU	214	0.407577	0.555556	ReLU
215	0.407373	0.555556	L-ReLU	215	0.407568	0.555556	ReLU
216	0.407364	0.555556	L-ReLU	216	0.407559	0.555556	ReLU
217	0.407355	0.555556	L-ReLU	217	0.40755	0.555556	ReLU

218	0.407347	0.555556	L-ReLU	218	0.407541	0.555556	ReLU
219	0.407338	0.555556	L-ReLU	219	0.407532	0.555556	ReLU
220	0.40733	0.555556	L-ReLU	220	0.407524	0.555556	ReLU
221	0.407322	0.555556	L-ReLU	221	0.407515	0.555556	ReLU
222	0.407314	0.555556	L-ReLU	222	0.407506	0.555556	ReLU
223	0.407305	0.555556	L-ReLU	223	0.407498	0.555556	ReLU
224	0.407297	0.555556	L-ReLU	224	0.40749	0.555556	ReLU
225	0.407289	0.555556	L-ReLU	225	0.407481	0.555556	ReLU
226	0.407282	0.555556	L-ReLU	226	0.407473	0.555556	ReLU
227	0.407274	0.555556	L-ReLU	227	0.407465	0.555556	ReLU
228	0.407266	0.555556	L-ReLU	228	0.407457	0.555556	ReLU
229	0.407258	0.555556	L-ReLU	229	0.407449	0.555556	ReLU
230	0.407251	0.555556	L-ReLU	230	0.407441	0.555556	ReLU
231	0.407243	0.555556	L-ReLU	231	0.407433	0.555556	ReLU
232	0.407236	0.555556	L-ReLU	232	0.407426	0.555556	ReLU
233	0.407229	0.555556	L-ReLU	233	0.407418	0.555556	ReLU
234	0.407221	0.555556	L-ReLU	234	0.40741	0.555556	ReLU
235	0.407214	0.555556	L-ReLU	235	0.407403	0.555556	ReLU
236	0.407207	0.555556	L-ReLU	236	0.407396	0.555556	ReLU
237	0.4072	0.555556	L-ReLU	237	0.407388	0.555556	ReLU
238	0.407193	0.555556	L-ReLU	238	0.407381	0.555556	ReLU
239	0.407186	0.555556	L-ReLU	239	0.407374	0.555556	ReLU
240	0.407179	0.555556	L-ReLU	240	0.407367	0.555556	ReLU
241	0.407172	0.555556	L-ReLU	241	0.407359	0.555556	ReLU
242	0.407165	0.555556	L-ReLU	242	0.407352	0.555556	ReLU
243	0.407159	0.555556	L-ReLU	243	0.407345	0.555556	ReLU
244	0.407152	0.555556	L-ReLU	244	0.407339	0.555556	ReLU
245	0.407146	0.555556	L-ReLU	245	0.407332	0.555556	ReLU
246	0.407139	0.555556	L-ReLU	246	0.407325	0.555556	ReLU
247	0.407133	0.555556	L-ReLU	247	0.407318	0.555556	ReLU
248	0.407126	0.555556	L-ReLU	248	0.407312	0.555556	ReLU
249	0.40712	0.555556	L-ReLU	249	0.407305	0.555556	ReLU
250	0.407114	0.555556	L-ReLU	250	0.407298	0.555556	ReLU
251	0.407107	0.555556	L-ReLU	251	0.407292	0.555556	ReLU
252	0.407101	0.555556	L-ReLU	252	0.407286	0.555556	ReLU
253	0.407095	0.555556	L-ReLU	253	0.407279	0.555556	ReLU
254	0.407089	0.555556	L-ReLU	254	0.407273	0.555556	ReLU
255	0.407083	0.555556	L-ReLU	255	0.407267	0.555556	ReLU
256	0.407077	0.555556	L-ReLU	256	0.40726	0.555556	ReLU
257	0.407071	0.555556	L-ReLU	257	0.407254	0.555556	ReLU
258	0.407065	0.555556	L-ReLU	258	0.407248	0.555556	ReLU
259	0.40706	0.555556	L-ReLU	259	0.407242	0.555556	ReLU
260	0.407054	0.555556	L-ReLU	260	0.407236	0.555556	ReLU
261	0.407048	0.555556	L-ReLU	261	0.40723	0.555556	ReLU

262	0.407042	0.555556	L-ReLU	262	0.407224	0.555556	ReLU
263	0.407037	0.555556	L-ReLU	263	0.407219	0.555556	ReLU
264	0.407031	0.555556	L-ReLU	264	0.407213	0.555556	ReLU
265	0.407026	0.555556	L-ReLU	265	0.407207	0.555556	ReLU
266	0.40702	0.555556	L-ReLU	266	0.407201	0.555556	ReLU
267	0.407015	0.555556	L-ReLU	267	0.407196	0.555556	ReLU
268	0.40701	0.555556	L-ReLU	268	0.40719	0.555556	ReLU
269	0.407004	0.555556	L-ReLU	269	0.407185	0.555556	ReLU
270	0.406999	0.555556	L-ReLU	270	0.407179	0.555556	ReLU
271	0.406994	0.555556	L-ReLU	271	0.407174	0.555556	ReLU
272	0.406988	0.555556	L-ReLU	272	0.407168	0.555556	ReLU
273	0.406983	0.555556	L-ReLU	273	0.407163	0.555556	ReLU
274	0.406978	0.555556	L-ReLU	274	0.407158	0.555556	ReLU
275	0.406973	0.555556	L-ReLU	275	0.407152	0.555556	ReLU
276	0.406968	0.555556	L-ReLU	276	0.407147	0.555556	ReLU
277	0.406963	0.555556	L-ReLU	277	0.407142	0.555556	ReLU
278	0.406958	0.555556	L-ReLU	278	0.407137	0.555556	ReLU
279	0.406953	0.555556	L-ReLU	279	0.407132	0.555556	ReLU
280	0.406948	0.555556	L-ReLU	280	0.407126	0.555556	ReLU
281	0.406944	0.555556	L-ReLU	281	0.407121	0.555556	ReLU
282	0.406939	0.555556	L-ReLU	282	0.407116	0.555556	ReLU
283	0.406934	0.555556	L-ReLU	283	0.407112	0.555556	ReLU
284	0.406929	0.555556	L-ReLU	284	0.407107	0.555556	ReLU
285	0.406925	0.555556	L-ReLU	285	0.407102	0.555556	ReLU
286	0.40692	0.555556	L-ReLU	286	0.407097	0.555556	ReLU
287	0.406915	0.555556	L-ReLU	287	0.407092	0.555556	ReLU
288	0.406911	0.555556	L-ReLU	288	0.407087	0.555556	ReLU
289	0.406906	0.555556	L-ReLU	289	0.407083	0.555556	ReLU
290	0.406902	0.555556	L-ReLU	290	0.407078	0.555556	ReLU
291	0.406897	0.555556	L-ReLU	291	0.407073	0.555556	ReLU
292	0.406893	0.555556	L-ReLU	292	0.407069	0.555556	ReLU
293	0.406889	0.555556	L-ReLU	293	0.407064	0.555556	ReLU
294	0.406884	0.555556	L-ReLU	294	0.407059	0.555556	ReLU
295	0.40688	0.555556	L-ReLU	295	0.407055	0.555556	ReLU
296	0.406876	0.555556	L-ReLU	296	0.40705	0.555556	ReLU
297	0.406871	0.555556	L-ReLU	297	0.407046	0.555556	ReLU
298	0.406867	0.555556	L-ReLU	298	0.407042	0.555556	ReLU
299	0.406863	0.555556	L-ReLU	299	0.407037	0.555556	ReLU

Iris – Baseline Sigmoid			
Iteration	**Cost**	**Acc**	**AF**
0	0.46657	0.444444	Sigmoid
1	0.462939	0.444444	Sigmoid
2	0.459594	0.444444	Sigmoid
3	0.474734	0.444444	Sigmoid
4	0.465756	0.444444	Sigmoid
5	0.457644	0.444444	Sigmoid
6	0.450481	0.444444	Sigmoid
7	0.444275	0.444444	Sigmoid
8	0.438976	0.444444	Sigmoid
9	0.434499	0.444444	Sigmoid
10	0.430747	0.444444	Sigmoid
11	0.427615	0.444444	Sigmoid
12	0.425009	0.444444	Sigmoid
13	0.422841	0.444444	Sigmoid
14	0.421035	0.444444	Sigmoid
15	0.419527	0.444444	Sigmoid
16	0.418265	0.444444	Sigmoid
17	0.417204	0.555556	Sigmoid
18	0.416269	0.555556	Sigmoid
19	0.415539	0.555556	Sigmoid
20	0.414892	0.555556	Sigmoid
21	0.414337	0.555556	Sigmoid
22	0.413859	0.555556	Sigmoid
23	0.413444	0.555556	Sigmoid
24	0.413083	0.555556	Sigmoid
25	0.412766	0.555556	Sigmoid
26	0.412488	0.555556	Sigmoid
27	0.412241	0.555556	Sigmoid
28	0.412022	0.555556	Sigmoid
29	0.411826	0.555556	Sigmoid
30	0.411651	0.555556	Sigmoid
31	0.411493	0.555556	Sigmoid
32	0.41135	0.555556	Sigmoid
33	0.41122	0.555556	Sigmoid
34	0.411102	0.555556	Sigmoid
35	0.410994	0.555556	Sigmoid
36	0.410895	0.555556	Sigmoid
37	0.410804	0.555556	Sigmoid
38	0.410721	0.555556	Sigmoid
39	0.410643	0.555556	Sigmoid
40	0.410572	0.555556	Sigmoid
41	0.410505	0.555556	Sigmoid

Iris – Baseine Softmax			
Iteration	**Cost**	**Acc**	**AF**
0	0.470005	0.444444	Softmax
1	0.466002	0.444444	Softmax
2	0.462303	0.444444	Softmax
3	0.457122	0.444444	Softmax
4	0.453945	0.444444	Softmax
5	0.451021	0.444444	Softmax
6	0.448327	0.444444	Softmax
7	0.445843	0.444444	Softmax
8	0.44355	0.444444	Softmax
9	0.441433	0.444444	Softmax
10	0.439475	0.444444	Softmax
11	0.437663	0.444444	Softmax
12	0.435985	0.444444	Softmax
13	0.434429	0.444444	Softmax
14	0.432985	0.444444	Softmax
15	0.431644	0.444444	Softmax
16	0.430396	0.444444	Softmax
17	0.429235	0.555556	Softmax
18	0.428176	0.555556	Softmax
19	0.427145	0.555556	Softmax
20	0.426203	0.555556	Softmax
21	0.425323	0.555556	Softmax
22	0.4245	0.555556	Softmax
23	0.42373	0.555556	Softmax
24	0.423007	0.555556	Softmax
25	0.42233	0.555556	Softmax
26	0.421694	0.555556	Softmax
27	0.421096	0.555556	Softmax
28	0.420533	0.555556	Softmax
29	0.420003	0.555556	Softmax
30	0.419503	0.555556	Softmax
31	0.419032	0.555556	Softmax
32	0.418587	0.555556	Softmax
33	0.418166	0.555556	Softmax
34	0.417768	0.555556	Softmax
35	0.417391	0.555556	Softmax
36	0.417034	0.555556	Softmax
37	0.416695	0.555556	Softmax
38	0.416374	0.555556	Softmax
39	0.416068	0.555556	Softmax
40	0.415777	0.555556	Softmax
41	0.415501	0.555556	Softmax

42	0.410443	0.555556	Sigmoid	42	0.415237	0.555556	Softmax
43	0.410385	0.555556	Sigmoid	43	0.414986	0.555556	Softmax
44	0.410332	0.555556	Sigmoid	44	0.414746	0.555556	Softmax
45	0.410281	0.555556	Sigmoid	45	0.414518	0.555556	Softmax
46	0.410234	0.555556	Sigmoid	46	0.414299	0.555556	Softmax
47	0.41019	0.555556	Sigmoid	47	0.41409	0.555556	Softmax
48	0.410148	0.555556	Sigmoid	48	0.413891	0.555556	Softmax
49	0.410109	0.555556	Sigmoid	49	0.413699	0.555556	Softmax
50	0.410072	0.555556	Sigmoid	50	0.413516	0.555556	Softmax
51	0.410037	0.555556	Sigmoid	51	0.41334	0.555556	Softmax
52	0.410004	0.555556	Sigmoid	52	0.413172	0.555556	Softmax
53	0.409973	0.555556	Sigmoid	53	0.41301	0.555556	Softmax
54	0.409943	0.555556	Sigmoid	54	0.412855	0.555556	Softmax
55	0.409915	0.555556	Sigmoid	55	0.412705	0.555556	Softmax
56	0.409888	0.555556	Sigmoid	56	0.412562	0.555556	Softmax
57	0.409863	0.555556	Sigmoid	57	0.412424	0.555556	Softmax
58	0.409839	0.555556	Sigmoid	58	0.412291	0.555556	Softmax
59	0.409816	0.555556	Sigmoid	59	0.412163	0.555556	Softmax
60	0.409794	0.555556	Sigmoid	60	0.412039	0.555556	Softmax
61	0.409773	0.555556	Sigmoid	61	0.41192	0.555556	Softmax
62	0.409754	0.555556	Sigmoid	62	0.411806	0.555556	Softmax
63	0.409734	0.555556	Sigmoid	63	0.411695	0.555556	Softmax
64	0.409716	0.555556	Sigmoid	64	0.411588	0.555556	Softmax
65	0.409699	0.555556	Sigmoid	65	0.411485	0.555556	Softmax
66	0.409682	0.555556	Sigmoid	66	0.411385	0.555556	Softmax
67	0.409666	0.555556	Sigmoid	67	0.411288	0.555556	Softmax
68	0.40965	0.555556	Sigmoid	68	0.411195	0.555556	Softmax
69	0.409635	0.555556	Sigmoid	69	0.411104	0.555556	Softmax
70	0.409621	0.555556	Sigmoid	70	0.411017	0.555556	Softmax
71	0.409607	0.555556	Sigmoid	71	0.410932	0.555556	Softmax
72	0.409594	0.555556	Sigmoid	72	0.410849	0.555556	Softmax
73	0.409581	0.555556	Sigmoid	73	0.41077	0.555556	Softmax
74	0.409569	0.555556	Sigmoid	74	0.410692	0.555556	Softmax
75	0.409557	0.555556	Sigmoid	75	0.410617	0.555556	Softmax
76	0.409545	0.555556	Sigmoid	76	0.410545	0.555556	Softmax
77	0.409534	0.555556	Sigmoid	77	0.410474	0.555556	Softmax
78	0.409523	0.555556	Sigmoid	78	0.410405	0.555556	Softmax
79	0.409512	0.555556	Sigmoid	79	0.410339	0.555556	Softmax
80	0.409501	0.555556	Sigmoid	80	0.410274	0.555556	Softmax
81	0.409491	0.555556	Sigmoid	81	0.410211	0.555556	Softmax
82	0.409481	0.555556	Sigmoid	82	0.410149	0.555556	Softmax
83	0.409472	0.555556	Sigmoid	83	0.41009	0.555556	Softmax
84	0.409462	0.555556	Sigmoid	84	0.410032	0.555556	Softmax
85	0.409453	0.555556	Sigmoid	85	0.409975	0.555556	Softmax

86	0.409444	0.555556	Sigmoid	86	0.40992	0.555556	Softmax
87	0.409435	0.555556	Sigmoid	87	0.409866	0.555556	Softmax
88	0.409427	0.555556	Sigmoid	88	0.409814	0.555556	Softmax
89	0.409418	0.555556	Sigmoid	89	0.409763	0.555556	Softmax
90	0.40941	0.555556	Sigmoid	90	0.409714	0.555556	Softmax
91	0.409402	0.555556	Sigmoid	91	0.409665	0.555556	Softmax
92	0.409394	0.555556	Sigmoid	92	0.409618	0.555556	Softmax
93	0.409386	0.555556	Sigmoid	93	0.409572	0.555556	Softmax
94	0.409378	0.555556	Sigmoid	94	0.409527	0.555556	Softmax
95	0.40937	0.555556	Sigmoid	95	0.409483	0.555556	Softmax
96	0.409362	0.555556	Sigmoid	96	0.40944	0.555556	Softmax
97	0.409355	0.555556	Sigmoid	97	0.409398	0.555556	Softmax
98	0.409347	0.555556	Sigmoid	98	0.409357	0.555556	Softmax
99	0.40934	0.555556	Sigmoid	99	0.409317	0.555556	Softmax
100	0.409333	0.555556	Sigmoid	100	0.409278	0.555556	Softmax
101	0.409325	0.555556	Sigmoid	101	0.40924	0.555556	Softmax
102	0.409318	0.555556	Sigmoid	102	0.409203	0.555556	Softmax
103	0.409311	0.555556	Sigmoid	103	0.409166	0.555556	Softmax
104	0.409304	0.555556	Sigmoid	104	0.40913	0.555556	Softmax
105	0.409297	0.555556	Sigmoid	105	0.409095	0.555556	Softmax
106	0.40929	0.555556	Sigmoid	106	0.409061	0.555556	Softmax
107	0.409283	0.555556	Sigmoid	107	0.409028	0.555556	Softmax
108	0.409276	0.555556	Sigmoid	108	0.408995	0.555556	Softmax
109	0.409269	0.555556	Sigmoid	109	0.408963	0.555556	Softmax
110	0.409262	0.555556	Sigmoid	110	0.408931	0.555556	Softmax
111	0.409255	0.555556	Sigmoid	111	0.408901	0.555556	Softmax
112	0.409248	0.555556	Sigmoid	112	0.40887	0.555556	Softmax
113	0.409241	0.555556	Sigmoid	113	0.408841	0.555556	Softmax
114	0.409235	0.555556	Sigmoid	114	0.408812	0.555556	Softmax
115	0.409228	0.555556	Sigmoid	115	0.408783	0.555556	Softmax
116	0.409221	0.555556	Sigmoid	116	0.408756	0.555556	Softmax
117	0.409214	0.555556	Sigmoid	117	0.408728	0.555556	Softmax
118	0.409207	0.555556	Sigmoid	118	0.408701	0.555556	Softmax
119	0.409201	0.555556	Sigmoid	119	0.408675	0.555556	Softmax
120	0.409194	0.555556	Sigmoid	120	0.408649	0.555556	Softmax
121	0.409187	0.555556	Sigmoid	121	0.408624	0.555556	Softmax
122	0.40918	0.555556	Sigmoid	122	0.408599	0.555556	Softmax
123	0.409173	0.555556	Sigmoid	123	0.408575	0.555556	Softmax
124	0.409167	0.555556	Sigmoid	124	0.408551	0.555556	Softmax
125	0.40916	0.555556	Sigmoid	125	0.408527	0.555556	Softmax
126	0.409153	0.555556	Sigmoid	126	0.408504	0.555556	Softmax
127	0.409146	0.555556	Sigmoid	127	0.408482	0.555556	Softmax
128	0.409139	0.555556	Sigmoid	128	0.408459	0.555556	Softmax
129	0.409132	0.555556	Sigmoid	129	0.408438	0.555556	Softmax

130	0.409126	0.555556	Sigmoid	130	0.408416	0.555556	Softmax
131	0.409119	0.555556	Sigmoid	131	0.408395	0.555556	Softmax
132	0.409112	0.555556	Sigmoid	132	0.408374	0.555556	Softmax
133	0.409105	0.555556	Sigmoid	133	0.408354	0.555556	Softmax
134	0.409098	0.555556	Sigmoid	134	0.408334	0.555556	Softmax
135	0.409091	0.555556	Sigmoid	135	0.408314	0.555556	Softmax
136	0.409084	0.555556	Sigmoid	136	0.408295	0.555556	Softmax
137	0.409077	0.555556	Sigmoid	137	0.408276	0.555556	Softmax
138	0.409071	0.555556	Sigmoid	138	0.408257	0.555556	Softmax
139	0.409064	0.555556	Sigmoid	139	0.408238	0.555556	Softmax
140	0.409057	0.555556	Sigmoid	140	0.40822	0.555556	Softmax
141	0.40905	0.555556	Sigmoid	141	0.408202	0.555556	Softmax
142	0.409043	0.555556	Sigmoid	142	0.408185	0.555556	Softmax
143	0.409036	0.555556	Sigmoid	143	0.408168	0.555556	Softmax
144	0.409029	0.555556	Sigmoid	144	0.408151	0.555556	Softmax
145	0.409022	0.555556	Sigmoid	145	0.408134	0.555556	Softmax
146	0.409015	0.555556	Sigmoid	146	0.408117	0.555556	Softmax
147	0.409008	0.555556	Sigmoid	147	0.408101	0.555556	Softmax
148	0.409001	0.555556	Sigmoid	148	0.408085	0.555556	Softmax
149	0.408994	0.555556	Sigmoid	149	0.408069	0.555556	Softmax
150	0.408986	0.555556	Sigmoid	150	0.408054	0.555556	Softmax
151	0.408979	0.555556	Sigmoid	151	0.408039	0.555556	Softmax
152	0.408972	0.555556	Sigmoid	152	0.408024	0.555556	Softmax
153	0.408965	0.555556	Sigmoid	153	0.408009	0.555556	Softmax
154	0.408958	0.555556	Sigmoid	154	0.407994	0.555556	Softmax
155	0.408951	0.555556	Sigmoid	155	0.40798	0.555556	Softmax
156	0.408944	0.555556	Sigmoid	156	0.407966	0.555556	Softmax
157	0.408937	0.555556	Sigmoid	157	0.407952	0.555556	Softmax
158	0.40893	0.555556	Sigmoid	158	0.407938	0.555556	Softmax
159	0.408922	0.555556	Sigmoid	159	0.407924	0.555556	Softmax
160	0.408915	0.555556	Sigmoid	160	0.407911	0.555556	Softmax
161	0.408908	0.555556	Sigmoid	161	0.407898	0.555556	Softmax
162	0.408901	0.555556	Sigmoid	162	0.407885	0.555556	Softmax
163	0.408894	0.555556	Sigmoid	163	0.407872	0.555556	Softmax
164	0.408886	0.555556	Sigmoid	164	0.407859	0.555556	Softmax
165	0.408879	0.555556	Sigmoid	165	0.407847	0.555556	Softmax
166	0.408872	0.555556	Sigmoid	166	0.407835	0.555556	Softmax
167	0.408865	0.555556	Sigmoid	167	0.407822	0.555556	Softmax
168	0.408858	0.555556	Sigmoid	168	0.40781	0.555556	Softmax
169	0.40885	0.555556	Sigmoid	169	0.407799	0.555556	Softmax
170	0.408843	0.555556	Sigmoid	170	0.407787	0.555556	Softmax
171	0.408836	0.555556	Sigmoid	171	0.407775	0.555556	Softmax
172	0.408829	0.555556	Sigmoid	172	0.407764	0.555556	Softmax
173	0.408821	0.555556	Sigmoid	173	0.407753	0.555556	Softmax

174	0.408814	0.555556	Sigmoid	174	0.407742	0.555556	Softmax
175	0.408807	0.555556	Sigmoid	175	0.407731	0.555556	Softmax
176	0.4088	0.555556	Sigmoid	176	0.40772	0.555556	Softmax
177	0.408793	0.555556	Sigmoid	177	0.407709	0.555556	Softmax
178	0.408785	0.555556	Sigmoid	178	0.407699	0.555556	Softmax
179	0.408778	0.555556	Sigmoid	179	0.407689	0.555556	Softmax
180	0.408771	0.555556	Sigmoid	180	0.407678	0.555556	Softmax
181	0.408764	0.555556	Sigmoid	181	0.407668	0.555556	Softmax
182	0.408756	0.555556	Sigmoid	182	0.407658	0.555556	Softmax
183	0.408749	0.555556	Sigmoid	183	0.407648	0.555556	Softmax
184	0.408742	0.555556	Sigmoid	184	0.407639	0.555556	Softmax
185	0.408735	0.555556	Sigmoid	185	0.407629	0.555556	Softmax
186	0.408727	0.555556	Sigmoid	186	0.40762	0.555556	Softmax
187	0.40872	0.555556	Sigmoid	187	0.40761	0.555556	Softmax
188	0.408713	0.555556	Sigmoid	188	0.407601	0.555556	Softmax
189	0.408706	0.555556	Sigmoid	189	0.407592	0.555556	Softmax
190	0.408698	0.555556	Sigmoid	190	0.407583	0.555556	Softmax
191	0.408691	0.555556	Sigmoid	191	0.407574	0.555556	Softmax
192	0.408684	0.555556	Sigmoid	192	0.407565	0.555556	Softmax
193	0.408677	0.555556	Sigmoid	193	0.407556	0.555556	Softmax
194	0.40867	0.555556	Sigmoid	194	0.407548	0.555556	Softmax
195	0.408662	0.555556	Sigmoid	195	0.407539	0.555556	Softmax
196	0.408655	0.555556	Sigmoid	196	0.407531	0.555556	Softmax
197	0.408648	0.555556	Sigmoid	197	0.407522	0.555556	Softmax
198	0.408641	0.555556	Sigmoid	198	0.407514	0.555556	Softmax
199	0.408634	0.555556	Sigmoid	199	0.407506	0.555556	Softmax
200	0.408626	0.555556	Sigmoid	200	0.407498	0.555556	Softmax
201	0.408619	0.555556	Sigmoid	201	0.40749	0.555556	Softmax
202	0.408612	0.555556	Sigmoid	202	0.407482	0.555556	Softmax
203	0.408605	0.555556	Sigmoid	203	0.407474	0.555556	Softmax
204	0.408598	0.555556	Sigmoid	204	0.407466	0.555556	Softmax
205	0.408591	0.555556	Sigmoid	205	0.407459	0.555556	Softmax
206	0.408584	0.555556	Sigmoid	206	0.407451	0.555556	Softmax
207	0.408576	0.555556	Sigmoid	207	0.407444	0.555556	Softmax
208	0.408569	0.555556	Sigmoid	208	0.407436	0.555556	Softmax
209	0.408562	0.555556	Sigmoid	209	0.407429	0.555556	Softmax
210	0.408555	0.555556	Sigmoid	210	0.407422	0.555556	Softmax
211	0.408548	0.555556	Sigmoid	211	0.407415	0.555556	Softmax
212	0.408541	0.555556	Sigmoid	212	0.407408	0.555556	Softmax
213	0.408534	0.555556	Sigmoid	213	0.407401	0.555556	Softmax
214	0.408527	0.555556	Sigmoid	214	0.407394	0.555556	Softmax
215	0.40852	0.555556	Sigmoid	215	0.407387	0.555556	Softmax
216	0.408513	0.555556	Sigmoid	216	0.40738	0.555556	Softmax
217	0.408506	0.555556	Sigmoid	217	0.407374	0.555556	Softmax

218	0.408499	0.555556	Sigmoid	218	0.407367	0.555556	Softmax
219	0.408492	0.555556	Sigmoid	219	0.40736	0.555556	Softmax
220	0.408485	0.555556	Sigmoid	220	0.407354	0.555556	Softmax
221	0.408478	0.555556	Sigmoid	221	0.407347	0.555556	Softmax
222	0.408471	0.555556	Sigmoid	222	0.407341	0.555556	Softmax
223	0.408464	0.555556	Sigmoid	223	0.407335	0.555556	Softmax
224	0.408457	0.555556	Sigmoid	224	0.407328	0.555556	Softmax
225	0.408451	0.555556	Sigmoid	225	0.407322	0.555556	Softmax
226	0.408444	0.555556	Sigmoid	226	0.407316	0.555556	Softmax
227	0.408437	0.555556	Sigmoid	227	0.40731	0.555556	Softmax
228	0.40843	0.555556	Sigmoid	228	0.407304	0.555556	Softmax
229	0.408423	0.555556	Sigmoid	229	0.407298	0.555556	Softmax
230	0.408416	0.555556	Sigmoid	230	0.407292	0.555556	Softmax
231	0.40841	0.555556	Sigmoid	231	0.407286	0.555556	Softmax
232	0.408403	0.555556	Sigmoid	232	0.407281	0.555556	Softmax
233	0.408396	0.555556	Sigmoid	233	0.407275	0.555556	Softmax
234	0.408389	0.555556	Sigmoid	234	0.407269	0.555556	Softmax
235	0.408383	0.555556	Sigmoid	235	0.407264	0.555556	Softmax
236	0.408376	0.555556	Sigmoid	236	0.407258	0.555556	Softmax
237	0.408369	0.555556	Sigmoid	237	0.407253	0.555556	Softmax
238	0.408362	0.555556	Sigmoid	238	0.407247	0.555556	Softmax
239	0.408356	0.555556	Sigmoid	239	0.407242	0.555556	Softmax
240	0.408349	0.555556	Sigmoid	240	0.407237	0.555556	Softmax
241	0.408342	0.555556	Sigmoid	241	0.407231	0.555556	Softmax
242	0.408336	0.555556	Sigmoid	242	0.407226	0.555556	Softmax
243	0.408329	0.555556	Sigmoid	243	0.407221	0.555556	Softmax
244	0.408323	0.555556	Sigmoid	244	0.407216	0.555556	Softmax
245	0.408316	0.555556	Sigmoid	245	0.407211	0.555556	Softmax
246	0.40831	0.555556	Sigmoid	246	0.407206	0.555556	Softmax
247	0.408303	0.555556	Sigmoid	247	0.407201	0.555556	Softmax
248	0.408297	0.555556	Sigmoid	248	0.407196	0.555556	Softmax
249	0.40829	0.555556	Sigmoid	249	0.407191	0.555556	Softmax
250	0.408284	0.555556	Sigmoid	250	0.407186	0.555556	Softmax
251	0.408277	0.555556	Sigmoid	251	0.407181	0.555556	Softmax
252	0.408271	0.555556	Sigmoid	252	0.407176	0.555556	Softmax
253	0.408264	0.555556	Sigmoid	253	0.407171	0.555556	Softmax
254	0.408258	0.555556	Sigmoid	254	0.407167	0.555556	Softmax
255	0.408252	0.555556	Sigmoid	255	0.407162	0.555556	Softmax
256	0.408245	0.555556	Sigmoid	256	0.407157	0.555556	Softmax
257	0.408239	0.555556	Sigmoid	257	0.407153	0.555556	Softmax
258	0.408232	0.555556	Sigmoid	258	0.407148	0.555556	Softmax
259	0.408226	0.555556	Sigmoid	259	0.407144	0.555556	Softmax
260	0.40822	0.555556	Sigmoid	260	0.407139	0.555556	Softmax
261	0.408214	0.555556	Sigmoid	261	0.407135	0.555556	Softmax

262	0.408207	0.555556	Sigmoid	262	0.407131	0.555556	Softmax
263	0.408201	0.555556	Sigmoid	263	0.407126	0.555556	Softmax
264	0.408195	0.555556	Sigmoid	264	0.407122	0.555556	Softmax
265	0.408189	0.555556	Sigmoid	265	0.407118	0.555556	Softmax
266	0.408183	0.555556	Sigmoid	266	0.407113	0.555556	Softmax
267	0.408176	0.555556	Sigmoid	267	0.407109	0.555556	Softmax
268	0.40817	0.555556	Sigmoid	268	0.407105	0.555556	Softmax
269	0.408164	0.555556	Sigmoid	269	0.407101	0.555556	Softmax
270	0.408158	0.555556	Sigmoid	270	0.407097	0.555556	Softmax
271	0.408152	0.555556	Sigmoid	271	0.407093	0.555556	Softmax
272	0.408146	0.555556	Sigmoid	272	0.407089	0.555556	Softmax
273	0.40814	0.555556	Sigmoid	273	0.407085	0.555556	Softmax
274	0.408134	0.555556	Sigmoid	274	0.407081	0.555556	Softmax
275	0.408128	0.555556	Sigmoid	275	0.407077	0.555556	Softmax
276	0.408122	0.555556	Sigmoid	276	0.407073	0.555556	Softmax
277	0.408116	0.555556	Sigmoid	277	0.407069	0.555556	Softmax
278	0.40811	0.555556	Sigmoid	278	0.407065	0.555556	Softmax
279	0.408104	0.555556	Sigmoid	279	0.407061	0.555556	Softmax
280	0.408098	0.555556	Sigmoid	280	0.407058	0.555556	Softmax
281	0.408092	0.555556	Sigmoid	281	0.407054	0.555556	Softmax
282	0.408086	0.555556	Sigmoid	282	0.40705	0.555556	Softmax
283	0.408081	0.555556	Sigmoid	283	0.407046	0.555556	Softmax
284	0.408075	0.555556	Sigmoid	284	0.407043	0.555556	Softmax
285	0.408069	0.555556	Sigmoid	285	0.407039	0.555556	Softmax
286	0.408063	0.555556	Sigmoid	286	0.407035	0.555556	Softmax
287	0.408057	0.555556	Sigmoid	287	0.407032	0.555556	Softmax
288	0.408052	0.555556	Sigmoid	288	0.407028	0.555556	Softmax
289	0.408046	0.555556	Sigmoid	289	0.407025	0.555556	Softmax
290	0.40804	0.555556	Sigmoid	290	0.407021	0.555556	Softmax
291	0.408035	0.555556	Sigmoid	291	0.407018	0.555556	Softmax
292	0.408029	0.555556	Sigmoid	292	0.407014	0.555556	Softmax
293	0.408023	0.555556	Sigmoid	293	0.407011	0.555556	Softmax
294	0.408018	0.555556	Sigmoid	294	0.407008	0.555556	Softmax
295	0.408012	0.555556	Sigmoid	295	0.407004	0.555556	Softmax
296	0.408006	0.555556	Sigmoid	296	0.407001	0.555556	Softmax
297	0.408001	0.555556	Sigmoid	297	0.406998	0.555556	Softmax
298	0.407995	0.555556	Sigmoid	298	0.406994	0.555556	Softmax
299	0.40799	0.555556	Sigmoid	299	0.406991	0.555556	Softmax

Iris – Baseline TanH				Iris – Leaky ReLU random			
Iteration	**Cost**	**Acc**	**AF**	**Iteration**	**Cost**	**Acc**	**AF**
0	0.520605	0.555556	TanH	0	0.473877	0	L-ReLU
1	0.490359	0.555556	TanH	1	0.467293	0.555556	L-ReLU
2	0.469848	0.555556	TanH	2	0.463554	0.555556	L-ReLU
3	0.445875	0.555556	TanH	3	0.460298	0.555556	ReLU
4	0.437691	0.555556	TanH	4	0.452421	0.444444	Softmax
5	0.431214	0.555556	TanH	5	0.464635	0.444444	Sigmoid
6	0.426434	0.555556	TanH	6	0.452165	0.555556	L-ReLU
7	0.42309	0.555556	TanH	7	0.44962	0.555556	L-ReLU
8	0.420775	0.555556	TanH	8	0.442493	0.555556	TanH
9	0.41907	0.555556	TanH	9	0.44587	0.555556	L-ReLU
10	0.417651	0.555556	TanH	10	0.444181	0.555556	ReLU
11	0.416326	0.555556	TanH	11	0.441877	0.555556	L-ReLU
12	0.415026	0.555556	TanH	12	0.434253	0.444444	Softmax
13	0.413755	0.555556	TanH	13	0.439091	0.555556	ReLU
14	0.412555	0.555556	TanH	14	0.437557	0.555556	ReLU
15	0.411471	0.555556	TanH	15	0.436468	0.444444	Sigmoid
16	0.410532	0.555556	TanH	16	0.456417	0.555556	TanH
17	0.409745	0.555556	TanH	17	0.452353	0.555556	TanH
18	0.409105	0.555556	TanH	18	0.43037	0.444444	Sigmoid
19	0.408592	0.555556	TanH	19	0.426527	0.444444	Softmax
20	0.408186	0.555556	TanH	20	0.427182	0.444444	Sigmoid
21	0.407864	0.555556	TanH	21	0.424502	0.444444	Softmax
22	0.407607	0.555556	TanH	22	0.429917	0.555556	L-ReLU
23	0.407401	0.555556	TanH	23	0.428958	0.555556	L-ReLU
24	0.407233	0.555556	TanH	24	0.421882	0.544444	Softmax
25	0.407093	0.555556	TanH	25	0.427376	0.555556	L-ReLU
26	0.406976	0.555556	TanH	26	0.426571	0.555556	L-ReLU
27	0.406876	0.555556	TanH	27	0.419742	0.555556	Softmax
28	0.406789	0.555556	TanH	28	0.425249	0.555556	L-ReLU
29	0.406714	0.555556	TanH	29	0.456436	0.555556	TanH
30	0.406647	0.555556	TanH	30	0.424618	0.555556	ReLU
31	0.406587	0.555556	TanH	31	0.449538	0.555556	TanH
32	0.406534	0.555556	TanH	32	0.423675	0.555556	ReLU
33	0.406485	0.555556	TanH	33	0.416781	0.444444	Sigmoid
34	0.406442	0.555556	TanH	34	0.416677	0.555556	Softmax
35	0.406402	0.555556	TanH	35	0.416309	0.555556	Softmax
36	0.406365	0.555556	TanH	36	0.421819	0.555556	ReLU
37	0.406331	0.555556	TanH	37	0.415495	0.555556	Softmax
38	0.406301	0.555556	TanH	38	0.443669	0.555556	TanH
39	0.406272	0.555556	TanH	39	0.420148	0.555556	L-ReLU
40	0.406245	0.555556	TanH	40	0.420282	0.555556	ReLU
41	0.406221	0.555556	TanH	41	0.436888	0.555556	TanH

42	0.406198	0.555556	TanH	42	0.414104	0.555556	Sigmoid
43	0.406177	0.555556	TanH	43	0.413877	0.555556	Softmax
44	0.406157	0.555556	TanH	44	0.418481	0.555556	L-ReLU
45	0.406139	0.555556	TanH	45	0.430674	0.555556	TanH
46	0.406121	0.555556	TanH	46	0.417908	0.555556	L-ReLU
47	0.406105	0.555556	TanH	47	0.425371	0.555556	TanH
48	0.40609	0.555556	TanH	48	0.417385	0.555556	L-ReLU
49	0.406076	0.555556	TanH	49	0.413051	0.555556	Sigmoid
50	0.406062	0.555556	TanH	50	0.416844	0.555556	L-ReLU
51	0.40605	0.555556	TanH	51	0.417208	0.555556	ReLU
52	0.406038	0.555556	TanH	52	0.412551	0.555556	Sigmoid
53	0.406027	0.555556	TanH	53	0.421303	0.555556	TanH
54	0.406016	0.555556	TanH	54	0.411544	0.555556	Softmax
55	0.406006	0.555556	TanH	55	0.412334	0.555556	Sigmoid
56	0.405996	0.555556	TanH	56	0.416193	0.555556	ReLU
57	0.405987	0.555556	TanH	57	0.411089	0.555556	Softmax
58	0.405979	0.555556	TanH	58	0.418078	0.555556	TanH
59	0.405971	0.555556	TanH	59	0.415017	0.555556	L-ReLU
60	0.405963	0.555556	TanH	60	0.411912	0.555556	Sigmoid
61	0.405956	0.555556	TanH	61	0.415632	0.555556	TanH
62	0.405949	0.555556	TanH	62	0.414518	0.555556	L-ReLU
63	0.405942	0.555556	TanH	63	0.411744	0.555556	Sigmoid
64	0.405936	0.555556	TanH	64	0.410302	0.555556	Softmax
65	0.40593	0.555556	TanH	65	0.413796	0.555556	TanH
66	0.405924	0.555556	TanH	66	0.410156	0.555556	Softmax
67	0.405919	0.555556	TanH	67	0.413829	0.555556	L-ReLU
68	0.405913	0.555556	TanH	68	0.414304	0.555556	ReLU
69	0.405908	0.555556	TanH	69	0.41349	0.555556	L-ReLU
70	0.405904	0.555556	TanH	70	0.412335	0.555556	TanH
71	0.405899	0.555556	TanH	71	0.41129	0.555556	Sigmoid
72	0.405894	0.555556	TanH	72	0.413113	0.555556	L-ReLU
73	0.40589	0.555556	TanH	73	0.411275	0.555556	TanH
74	0.405886	0.555556	TanH	74	0.411157	0.555556	Sigmoid
75	0.405882	0.555556	TanH	75	0.41342	0.555556	ReLU
76	0.405879	0.555556	TanH	76	0.411017	0.555556	Sigmoid
77	0.405875	0.555556	TanH	77	0.412521	0.555556	L-ReLU
78	0.405871	0.555556	TanH	78	0.409217	0.555556	Softmax
79	0.405868	0.555556	TanH	79	0.410845	0.555556	Sigmoid
80	0.405865	0.555556	TanH	80	0.412231	0.555556	L-ReLU
81	0.405862	0.555556	TanH	81	0.410331	0.555556	TanH
82	0.405859	0.555556	TanH	82	0.412676	0.555556	ReLU
83	0.405856	0.555556	TanH	83	0.408979	0.555556	Softmax
84	0.405853	0.555556	TanH	84	0.410633	0.555556	Sigmoid
85	0.40585	0.555556	TanH	85	0.4124	0.555556	ReLU

86	0.405848	0.555556	TanH	86	0.410523	0.555556	Sigmoid
87	0.405845	0.555556	TanH	87	0.412202	0.555556	ReLU
88	0.405843	0.555556	TanH	88	0.410419	0.555556	Sigmoid
89	0.405841	0.555556	TanH	89	0.412013	0.555556	ReLU
90	0.405838	0.555556	TanH	90	0.409574	0.555556	TanH
91	0.405836	0.555556	TanH	91	0.411865	0.555556	ReLU
92	0.405834	0.555556	TanH	92	0.410272	0.555556	Sigmoid
93	0.405832	0.555556	TanH	93	0.411093	0.555556	L-ReLU
94	0.40583	0.555556	TanH	94	0.411608	0.555556	ReLU
95	0.405828	0.555556	TanH	95	0.409037	0.555556	TanH
96	0.405826	0.555556	TanH	96	0.410868	0.555556	L-ReLU
97	0.405824	0.555556	TanH	97	0.410098	0.555556	Sigmoid
98	0.405822	0.555556	TanH	98	0.408555	0.555556	Softmax
99	0.405821	0.555556	TanH	99	0.410699	0.555556	L-ReLU
100	0.405819	0.555556	TanH	100	0.411219	0.555556	ReLU
101	0.405817	0.555556	TanH	101	0.410551	0.555556	L-ReLU
102	0.405816	0.555556	TanH	102	0.411071	0.555556	ReLU
103	0.405814	0.555556	TanH	103	0.408546	0.555556	TanH
104	0.405813	0.555556	TanH	104	0.410368	0.555556	L-ReLU
105	0.405811	0.555556	TanH	105	0.408259	0.555556	TanH
106	0.40581	0.555556	TanH	106	0.408544	0.555556	Softmax
107	0.405809	0.555556	TanH	107	0.41024	0.555556	L-ReLU
108	0.405807	0.555556	TanH	108	0.410773	0.555556	ReLU
109	0.405806	0.555556	TanH	109	0.408522	0.555556	Softmax
110	0.405805	0.555556	TanH	110	0.4101	0.555556	L-ReLU
111	0.405804	0.555556	TanH	111	0.410625	0.555556	ReLU
112	0.405802	0.555556	TanH	112	0.408511	0.555556	Softmax
113	0.405801	0.555556	TanH	113	0.40796	0.555556	TanH
114	0.4058	0.555556	TanH	114	0.408479	0.555556	Softmax
115	0.405799	0.555556	TanH	115	0.407786	0.555556	TanH
116	0.405798	0.555556	TanH	116	0.409618	0.555556	Sigmoid
117	0.405797	0.555556	TanH	117	0.409835	0.555556	L-ReLU
118	0.405796	0.555556	TanH	118	0.409553	0.555556	Sigmoid
119	0.405795	0.555556	TanH	119	0.410317	0.555556	ReLU
120	0.405794	0.555556	TanH	120	0.40848	0.555556	Softmax
121	0.405793	0.555556	TanH	121	0.407578	0.555556	TanH
122	0.405792	0.555556	TanH	122	0.410224	0.555556	ReLU
123	0.405791	0.555556	TanH	123	0.409597	0.555556	L-ReLU
124	0.40579	0.555556	TanH	124	0.408511	0.555556	Softmax
125	0.405789	0.555556	TanH	125	0.409397	0.555556	Sigmoid
126	0.405788	0.555556	TanH	126	0.410067	0.555556	ReLU
127	0.405788	0.555556	TanH	127	0.4074	0.555556	TanH
128	0.405787	0.555556	TanH	128	0.409995	0.555556	ReLU
129	0.405786	0.555556	TanH	129	0.408546	0.555556	Softmax

130	0.405785	0.555556	TanH	130	0.407283	0.555556	TanH
131	0.405784	0.555556	TanH	131	0.409919	0.555556	ReLU
132	0.405784	0.555556	TanH	132	0.40718	0.555556	TanH
133	0.405783	0.555556	TanH	133	0.409248	0.555556	Sigmoid
134	0.405782	0.555556	TanH	134	0.409245	0.555556	L-ReLU
135	0.405782	0.555556	TanH	135	0.409194	0.555556	Sigmoid
136	0.405781	0.555556	TanH	136	0.409173	0.555556	L-ReLU
137	0.40578	0.555556	TanH	137	0.407056	0.555556	TanH
138	0.40578	0.555556	TanH	138	0.409674	0.555556	ReLU
139	0.405779	0.555556	TanH	139	0.406976	0.555556	TanH
140	0.405778	0.555556	TanH	140	0.409615	0.555556	ReLU
141	0.405778	0.555556	TanH	141	0.409013	0.555556	L-ReLU
142	0.405777	0.555556	TanH	142	0.406893	0.555556	TanH
143	0.405776	0.555556	TanH	143	0.409521	0.555556	ReLU
144	0.405776	0.555556	TanH	144	0.40902	0.555556	Sigmoid
145	0.405775	0.555556	TanH	145	0.409449	0.555556	ReLU
146	0.405775	0.555556	TanH	146	0.409042	0.555556	Softmax
147	0.405774	0.555556	TanH	147	0.408966	0.555556	Sigmoid
148	0.405774	0.555556	TanH	148	0.408832	0.555556	L-ReLU
149	0.405773	0.555556	TanH	149	0.409347	0.555556	ReLU
150	0.405773	0.555556	TanH	150	0.40877	0.555556	L-ReLU
151	0.405772	0.555556	TanH	151	0.409108	0.555556	Softmax
152	0.405772	0.555556	TanH	152	0.409284	0.555556	ReLU
153	0.405771	0.555556	TanH	153	0.406776	0.555556	TanH
154	0.405771	0.555556	TanH	154	0.408695	0.555556	L-ReLU
155	0.40577	0.555556	TanH	155	0.409206	0.555556	ReLU
156	0.40577	0.555556	TanH	156	0.408638	0.555556	L-ReLU
157	0.405769	0.555556	TanH	157	0.409281	0.555556	Softmax
158	0.405769	0.555556	TanH	158	0.40878	0.555556	Sigmoid
159	0.405768	0.555556	TanH	159	0.40859	0.555556	L-ReLU
160	0.405768	0.555556	TanH	160	0.406694	0.555556	TanH
161	0.405768	0.555556	TanH	161	0.409081	0.555556	ReLU
162	0.405767	0.555556	TanH	162	0.409353	0.555556	Softmax
163	0.405767	0.555556	TanH	163	0.408713	0.555556	Sigmoid
164	0.405766	0.555556	TanH	164	0.409257	0.555556	Softmax
165	0.405766	0.555556	TanH	165	0.40664	0.555556	TanH
166	0.405765	0.555556	TanH	166	0.40903	0.555556	ReLU
167	0.405765	0.555556	TanH	167	0.408466	0.555556	L-ReLU
168	0.405765	0.555556	TanH	168	0.408649	0.555556	Sigmoid
169	0.405764	0.555556	TanH	169	0.409336	0.555556	Softmax
170	0.405764	0.555556	TanH	170	0.406586	0.555556	TanH
171	0.405764	0.555556	TanH	171	0.408413	0.555556	L-ReLU
172	0.405763	0.555556	TanH	172	0.409341	0.555556	Softmax
173	0.405763	0.555556	TanH	173	0.406545	0.555556	TanH

174	0.405763	0.555556	TanH	174	0.408613	0.555556	Sigmoid
175	0.405762	0.555556	TanH	175	0.409304	0.555556	Softmax
176	0.405762	0.555556	TanH	176	0.408911	0.555556	ReLU
177	0.405762	0.555556	TanH	177	0.408345	0.555556	L-ReLU
178	0.405761	0.555556	TanH	178	0.408856	0.555556	ReLU
179	0.405761	0.555556	TanH	179	0.408297	0.555556	L-ReLU
180	0.405761	0.555556	TanH	180	0.408802	0.555556	ReLU
181	0.40576	0.555556	TanH	181	0.409509	0.555556	Softmax
182	0.40576	0.555556	TanH	182	0.408261	0.555556	L-ReLU
183	0.40576	0.555556	TanH	183	0.408491	0.555556	Sigmoid
184	0.405759	0.555556	TanH	184	0.409474	0.555556	Softmax
185	0.405759	0.555556	TanH	185	0.408481	0.555556	Sigmoid
186	0.405759	0.555556	TanH	186	0.406484	0.555556	TanH
187	0.405759	0.555556	TanH	187	0.409436	0.555556	Softmax
188	0.405758	0.555556	TanH	188	0.40847	0.555556	Sigmoid
189	0.405758	0.555556	TanH	189	0.40819	0.555556	L-ReLU
190	0.405758	0.555556	TanH	190	0.408693	0.555556	ReLU
191	0.405757	0.555556	TanH	191	0.408412	0.555556	Sigmoid
192	0.405757	0.555556	TanH	192	0.408644	0.555556	ReLU
193	0.405757	0.555556	TanH	193	0.409557	0.555556	Softmax
194	0.405757	0.555556	TanH	194	0.408116	0.555556	L-ReLU
195	0.405756	0.555556	TanH	195	0.408369	0.555556	Sigmoid
196	0.405756	0.555556	TanH	196	0.409524	0.555556	Softmax
197	0.405756	0.555556	TanH	197	0.408089	0.555556	L-ReLU
198	0.405756	0.555556	TanH	198	0.408346	0.555556	Sigmoid
199	0.405755	0.555556	TanH	199	0.406437	0.555556	TanH
200	0.405755	0.555556	TanH	200	0.409548	0.555556	Softmax
201	0.405755	0.555556	TanH	201	0.408578	0.555556	ReLU
202	0.405755	0.555556	TanH	202	0.408321	0.555556	Sigmoid
203	0.405755	0.555556	TanH	203	0.408016	0.555556	L-ReLU
204	0.405754	0.555556	TanH	204	0.408286	0.555556	Sigmoid
205	0.405754	0.555556	TanH	205	0.408488	0.555556	ReLU
206	0.405754	0.555556	TanH	206	0.407959	0.555556	L-ReLU
207	0.405754	0.555556	TanH	207	0.408444	0.555556	ReLU
208	0.405753	0.555556	TanH	208	0.407922	0.555556	L-ReLU
209	0.405753	0.555556	TanH	209	0.406401	0.555556	TanH
210	0.405753	0.555556	TanH	210	0.407898	0.555556	L-ReLU
211	0.405753	0.555556	TanH	211	0.406367	0.555556	TanH
212	0.405753	0.555556	TanH	212	0.410046	0.555556	Softmax
213	0.405752	0.555556	TanH	213	0.408202	0.555556	Sigmoid
214	0.405752	0.555556	TanH	214	0.409941	0.555556	Softmax
215	0.405752	0.555556	TanH	215	0.407887	0.555556	L-ReLU
216	0.405752	0.555556	TanH	216	0.40838	0.555556	ReLU
217	0.405752	0.555556	TanH	217	0.407851	0.555556	L-ReLU

218	0.405751	0.555556	TanH	218	0.408339	0.555556	ReLU
219	0.405751	0.555556	TanH	219	0.406338	0.555556	TanH
220	0.405751	0.555556	TanH	220	0.410143	0.555556	Softmax
221	0.405751	0.555556	TanH	221	0.408154	0.555556	Sigmoid
222	0.405751	0.555556	TanH	222	0.410038	0.555556	Softmax
223	0.405751	0.555556	TanH	223	0.408155	0.555556	Sigmoid
224	0.40575	0.555556	TanH	224	0.40781	0.555556	L-ReLU
225	0.40575	0.555556	TanH	225	0.410001	0.555556	Softmax
226	0.40575	0.555556	TanH	226	0.40631	0.555556	TanH
227	0.40575	0.555556	TanH	227	0.409934	0.555556	Softmax
228	0.40575	0.555556	TanH	228	0.408163	0.555556	Sigmoid
229	0.40575	0.555556	TanH	229	0.408326	0.555556	ReLU
230	0.405749	0.555556	TanH	230	0.406286	0.555556	TanH
231	0.405749	0.555556	TanH	231	0.408302	0.555556	ReLU
232	0.405749	0.555556	TanH	232	0.41004	0.555556	Softmax
233	0.405749	0.555556	TanH	233	0.406263	0.555556	TanH
234	0.405749	0.555556	TanH	234	0.407777	0.555556	L-ReLU
235	0.405749	0.555556	TanH	235	0.406243	0.555556	TanH
236	0.405748	0.555556	TanH	236	0.410094	0.555556	Softmax
237	0.405748	0.555556	TanH	237	0.406224	0.555556	TanH
238	0.405748	0.555556	TanH	238	0.410026	0.555556	Softmax
239	0.405748	0.555556	TanH	239	0.408162	0.555556	Sigmoid
240	0.405748	0.555556	TanH	240	0.408313	0.555556	ReLU
241	0.405748	0.555556	TanH	241	0.408124	0.555556	Sigmoid
242	0.405748	0.555556	TanH	242	0.410038	0.555556	Softmax
243	0.405747	0.555556	TanH	243	0.406209	0.555556	TanH
244	0.405747	0.555556	TanH	244	0.408295	0.555556	ReLU
245	0.405747	0.555556	TanH	245	0.40811	0.555556	Sigmoid
246	0.405747	0.555556	TanH	246	0.410076	0.555556	Softmax
247	0.405747	0.555556	TanH	247	0.406194	0.555556	TanH
248	0.405747	0.555556	TanH	248	0.408117	0.555556	Sigmoid
249	0.405747	0.555556	TanH	249	0.408256	0.555556	ReLU
250	0.405746	0.555556	TanH	250	0.40618	0.555556	TanH
251	0.405746	0.555556	TanH	251	0.407691	0.555556	L-ReLU
252	0.405746	0.555556	TanH	252	0.406166	0.555556	TanH
253	0.405746	0.555556	TanH	253	0.40767	0.555556	L-ReLU
254	0.405746	0.555556	TanH	254	0.408195	0.555556	ReLU
255	0.405746	0.555556	TanH	255	0.407636	0.555556	L-ReLU
256	0.405746	0.555556	TanH	256	0.406154	0.555556	TanH
257	0.405746	0.555556	TanH	257	0.407617	0.555556	L-ReLU
258	0.405745	0.555556	TanH	258	0.406141	0.555556	TanH
259	0.405745	0.555556	TanH	259	0.408137	0.555556	ReLU
260	0.405745	0.555556	TanH	260	0.410768	0.555556	Softmax
261	0.405745	0.555556	TanH	261	0.407603	0.555556	L-ReLU

262	0.405745	0.555556	TanH	262	0.40813	0.555556	ReLU
263	0.405745	0.555556	TanH	263	0.407981	0.555556	Sigmoid
264	0.405745	0.555556	TanH	264	0.408091	0.555556	ReLU
265	0.405745	0.555556	TanH	265	0.407948	0.555556	Sigmoid
266	0.405745	0.555556	TanH	266	0.407526	0.555556	L-ReLU
267	0.405744	0.555556	TanH	267	0.410981	0.555556	Softmax
268	0.405744	0.555556	TanH	268	0.408068	0.555556	ReLU
269	0.405744	0.555556	TanH	269	0.410911	0.555556	Softmax
270	0.405744	0.555556	TanH	270	0.407539	0.555556	L-ReLU
271	0.405744	0.555556	TanH	271	0.410835	0.555556	Softmax
272	0.405744	0.555556	TanH	272	0.407547	0.555556	L-ReLU
273	0.405744	0.555556	TanH	273	0.410762	0.555556	Softmax
274	0.405744	0.555556	TanH	274	0.40614	0.555556	TanH
275	0.405744	0.555556	TanH	275	0.410681	0.555556	Softmax
276	0.405743	0.555556	TanH	276	0.406123	0.555556	TanH
277	0.405743	0.555556	TanH	277	0.408017	0.555556	Sigmoid
278	0.405743	0.555556	TanH	278	0.410638	0.555556	Softmax
279	0.405743	0.555556	TanH	279	0.408029	0.555556	Sigmoid
280	0.405743	0.555556	TanH	280	0.410541	0.555556	Softmax
281	0.405743	0.555556	TanH	281	0.408041	0.555556	Sigmoid
282	0.405743	0.555556	TanH	282	0.410447	0.555556	Softmax
283	0.405743	0.555556	TanH	283	0.408179	0.555556	ReLU
284	0.405743	0.555556	TanH	284	0.410397	0.555556	Softmax
285	0.405743	0.555556	TanH	285	0.407605	0.555556	L-ReLU
286	0.405743	0.555556	TanH	286	0.408048	0.555556	Sigmoid
287	0.405742	0.555556	TanH	287	0.406122	0.555556	TanH
288	0.405742	0.555556	TanH	288	0.410429	0.555556	Softmax
289	0.405742	0.555556	TanH	289	0.408063	0.555556	Sigmoid
290	0.405742	0.555556	TanH	290	0.410342	0.555556	Softmax
291	0.405742	0.555556	TanH	291	0.408198	0.555556	ReLU
292	0.405742	0.555556	TanH	292	0.406111	0.555556	TanH
293	0.405742	0.555556	TanH	293	0.408176	0.555556	ReLU
294	0.405742	0.555556	TanH	294	0.410424	0.555556	Softmax
295	0.405742	0.555556	TanH	295	0.407591	0.555556	L-ReLU
296	0.405742	0.555556	TanH	296	0.408169	0.555556	ReLU
297	0.405742	0.555556	TanH	297	0.406105	0.555556	TanH
298	0.405741	0.555556	TanH	298	0.407554	0.555556	L-ReLU
299	0.405741	0.555556	TanH	299	0.406092	0.555556	TanH

Iris – ReLU random			
Iteration	**Cost**	**Acc**	**AF**
0	0.473869	0	ReLU
1	0.467339	0.555556	ReLU
2	0.46367	0.555556	ReLU
3	0.446685	0.555556	TanH
4	0.457565	0.555556	L-ReLU
5	0.465408	0.444444	Sigmoid
6	0.452424	0.555556	L-ReLU
7	0.449861	0.555556	L-ReLU
8	0.441865	0.444444	Softmax
9	0.445873	0.555556	ReLU
10	0.443857	0.555556	ReLU
11	0.44507	0.444444	Sigmoid
12	0.433674	0.444444	Softmax
13	0.439064	0.444444	Sigmoid
14	0.451476	0.555556	TanH
15	0.449424	0.555556	TanH
16	0.435952	0.555556	L-ReLU
17	0.435024	0.555556	ReLU
18	0.450303	0.555556	TanH
19	0.432687	0.555556	L-ReLU
20	0.44709	0.555556	TanH
21	0.431614	0.555556	ReLU
22	0.442752	0.555556	TanH
23	0.429525	0.555556	L-ReLU
24	0.429271	0.555556	ReLU
25	0.422221	0.444444	Sigmoid
26	0.427676	0.555556	ReLU
27	0.426876	0.555556	ReLU
28	0.425424	0.555556	L-ReLU
29	0.425422	0.555556	ReLU
30	0.441801	0.555556	TanH
31	0.418437	0.555556	Softmax
32	0.42398	0.555556	ReLU
33	0.417369	0.555556	Softmax
34	0.422181	0.555556	L-ReLU
35	0.422378	0.555556	ReLU
36	0.415938	0.555556	Softmax
37	0.42146	0.555556	ReLU
38	0.436011	0.555556	TanH
39	0.420048	0.555556	L-ReLU
40	0.429293	0.555556	TanH
41	0.414478	0.555556	Softmax

Iris – Sigmoid random			
Iteration	**Cost**	**Acc**	**AF**
0	0.481877	0.444444	Sigmoid
1	0.472041	0.444444	Sigmoid
2	0.463007	0.444444	Sigmoid
3	0.462663	0.555556	ReLU
4	0.459355	0.555556	ReLU
5	0.454684	0.555556	TanH
6	0.447654	0.444444	Sigmoid
7	0.451769	0.555556	L-ReLU
8	0.449285	0.555556	L-ReLU
9	0.438033	0.444444	Softmax
10	0.436143	0.444444	Sigmoid
11	0.443865	0.555556	L-ReLU
12	0.441988	0.555556	L-ReLU
13	0.429268	0.444444	Sigmoid
14	0.439238	0.555556	L-ReLU
15	0.437699	0.555556	L-ReLU
16	0.436359	0.555556	ReLU
17	0.434924	0.555556	L-ReLU
18	0.424033	0.555556	Softmax
19	0.45709	0.555556	TanH
20	0.422867	0.555556	Softmax
21	0.420046	0.444444	Sigmoid
22	0.454864	0.555556	TanH
23	0.450068	0.555556	TanH
24	0.428319	0.555556	L-ReLU
25	0.419971	0.555556	Softmax
26	0.427083	0.555556	ReLU
27	0.418723	0.555556	Softmax
28	0.447323	0.555556	TanH
29	0.416069	0.555556	Sigmoid
30	0.424642	0.555556	ReLU
31	0.41699	0.555556	Softmax
32	0.423287	0.555556	L-ReLU
33	0.414489	0.555556	Sigmoid
34	0.422623	0.555556	ReLU
35	0.421838	0.555556	L-ReLU
36	0.413557	0.555556	Sigmoid
37	0.421039	0.555556	L-ReLU
38	0.420855	0.555556	ReLU
39	0.446486	0.555556	TanH
40	0.41995	0.555556	ReLU
41	0.419213	0.555556	L-ReLU

42	0.41466	0.555556	Sigmoid	42	0.419192	0.555556	ReLU
43	0.423912	0.555556	TanH	43	0.41232	0.555556	Sigmoid
44	0.414437	0.555556	Sigmoid	44	0.412535	0.555556	Softmax
45	0.41911	0.555556	ReLU	45	0.439986	0.555556	TanH
46	0.413209	0.555556	Softmax	46	0.412133	0.555556	Sigmoid
47	0.41377	0.555556	Sigmoid	47	0.412114	0.555556	Softmax
48	0.417452	0.555556	L-ReLU	48	0.417553	0.555556	ReLU
49	0.419986	0.555556	TanH	49	0.411735	0.555556	Softmax
50	0.41342	0.555556	Sigmoid	50	0.416711	0.555556	L-ReLU
51	0.416936	0.555556	TanH	51	0.411558	0.555556	Sigmoid
52	0.413285	0.555556	Sigmoid	52	0.411275	0.555556	Softmax
53	0.417144	0.555556	ReLU	53	0.416103	0.555556	L-ReLU
54	0.412969	0.555556	Sigmoid	54	0.433803	0.555556	TanH
55	0.415886	0.555556	L-ReLU	55	0.415944	0.555556	ReLU
56	0.416391	0.555556	ReLU	56	0.410743	0.555556	Softmax
57	0.415392	0.555556	L-ReLU	57	0.415189	0.555556	L-ReLU
58	0.412441	0.555556	Sigmoid	58	0.410517	0.555556	Softmax
59	0.414607	0.555556	TanH	59	0.428119	0.555556	TanH
60	0.41235	0.555556	Sigmoid	60	0.410369	0.555556	Softmax
61	0.4147	0.555556	L-ReLU	61	0.414484	0.555556	L-ReLU
62	0.412137	0.555556	Sigmoid	62	0.41097	0.555556	Sigmoid
63	0.412892	0.555556	TanH	63	0.414177	0.555556	L-ReLU
64	0.414978	0.555556	ReLU	64	0.409983	0.555556	Softmax
65	0.411948	0.555556	Sigmoid	65	0.414285	0.555556	ReLU
66	0.413893	0.555556	L-ReLU	66	0.409815	0.555556	Softmax
67	0.411592	0.555556	TanH	67	0.423511	0.555556	TanH
68	0.413619	0.555556	L-ReLU	68	0.413451	0.555556	L-ReLU
69	0.409825	0.555556	Softmax	69	0.409617	0.555556	Softmax
70	0.414095	0.555556	ReLU	70	0.413217	0.555556	L-ReLU
71	0.41154	0.555556	Sigmoid	71	0.409481	0.555556	Softmax
72	0.413077	0.555556	L-ReLU	72	0.413384	0.555556	ReLU
73	0.411388	0.555556	Sigmoid	73	0.410474	0.555556	Sigmoid
74	0.413526	0.555556	ReLU	74	0.409297	0.555556	Softmax
75	0.411238	0.555556	Sigmoid	75	0.412691	0.555556	L-ReLU
76	0.413263	0.555556	ReLU	76	0.419765	0.555556	TanH
77	0.411097	0.555556	Sigmoid	77	0.40915	0.555556	Softmax
78	0.413016	0.555556	ReLU	78	0.412751	0.555556	ReLU
79	0.410965	0.555556	Sigmoid	79	0.417226	0.555556	TanH
80	0.412117	0.555556	L-ReLU	80	0.410273	0.555556	Sigmoid
81	0.410847	0.555556	Sigmoid	81	0.415307	0.555556	TanH
82	0.410304	0.555556	TanH	82	0.411921	0.555556	L-ReLU
83	0.408954	0.555556	Softmax	83	0.410194	0.555556	Sigmoid
84	0.412456	0.555556	ReLU	84	0.408859	0.555556	Softmax
85	0.408871	0.555556	Softmax	85	0.410115	0.555556	Sigmoid

86	0.41065	0.555556	Sigmoid	86	0.413708	0.555556	TanH
87	0.411552	0.555556	L-ReLU	87	0.411952	0.555556	ReLU
88	0.410546	0.555556	Sigmoid	88	0.410034	0.555556	Sigmoid
89	0.409571	0.555556	TanH	89	0.412509	0.555556	TanH
90	0.408708	0.555556	Softmax	90	0.411268	0.555556	L-ReLU
91	0.410458	0.555556	Sigmoid	91	0.409957	0.555556	Sigmoid
92	0.409087	0.555556	TanH	92	0.408615	0.555556	Softmax
93	0.411148	0.555556	L-ReLU	93	0.411512	0.555556	ReLU
94	0.408695	0.555556	TanH	94	0.408554	0.555556	Softmax
95	0.410355	0.555556	Sigmoid	95	0.411499	0.555556	TanH
96	0.40857	0.555556	Softmax	96	0.40851	0.555556	Softmax
97	0.408387	0.555556	TanH	97	0.410824	0.555556	TanH
98	0.408527	0.555556	Softmax	98	0.409786	0.555556	Sigmoid
99	0.411478	0.555556	ReLU	99	0.408442	0.555556	Softmax
100	0.410198	0.555556	Sigmoid	100	0.410685	0.555556	L-ReLU
101	0.408455	0.555556	Softmax	101	0.409698	0.555556	Sigmoid
102	0.411289	0.555556	ReLU	102	0.410565	0.555556	L-ReLU
103	0.410081	0.555556	Sigmoid	103	0.409638	0.555556	Sigmoid
104	0.408077	0.555556	TanH	104	0.41045	0.555556	L-ReLU
105	0.411109	0.555556	ReLU	105	0.410812	0.555556	ReLU
106	0.410395	0.555556	L-ReLU	106	0.410324	0.555556	L-ReLU
107	0.407845	0.555556	TanH	107	0.409525	0.555556	Sigmoid
108	0.409947	0.555556	Sigmoid	108	0.41022	0.555556	L-ReLU
109	0.410874	0.555556	ReLU	109	0.408342	0.555556	Softmax
110	0.408401	0.555556	Softmax	110	0.410141	0.555556	L-ReLU
111	0.409847	0.555556	Sigmoid	111	0.409426	0.555556	Sigmoid
112	0.407641	0.555556	TanH	112	0.410044	0.555556	L-ReLU
113	0.409801	0.555556	Sigmoid	113	0.410388	0.555556	ReLU
114	0.408343	0.555556	Softmax	114	0.409348	0.555556	Sigmoid
115	0.409746	0.555556	Sigmoid	115	0.410286	0.555556	ReLU
116	0.408288	0.555556	Softmax	116	0.40985	0.555556	L-ReLU
117	0.41055	0.555556	ReLU	117	0.409273	0.555556	Sigmoid
118	0.408265	0.555556	Softmax	118	0.408315	0.555556	Softmax
119	0.409863	0.555556	L-ReLU	119	0.409645	0.555556	TanH
120	0.408242	0.555556	Softmax	120	0.409217	0.555556	Sigmoid
121	0.409597	0.555556	Sigmoid	121	0.408272	0.555556	Softmax
122	0.407441	0.555556	TanH	122	0.409173	0.555556	Sigmoid
123	0.409722	0.555556	L-ReLU	123	0.409988	0.555556	ReLU
124	0.410274	0.555556	ReLU	124	0.408234	0.555556	Softmax
125	0.409622	0.555556	L-ReLU	125	0.409551	0.555556	L-ReLU
126	0.409463	0.555556	Sigmoid	126	0.40918	0.555556	TanH
127	0.409535	0.555556	L-ReLU	127	0.409851	0.555556	ReLU
128	0.410074	0.555556	ReLU	128	0.408863	0.555556	TanH
129	0.40937	0.555556	Sigmoid	129	0.40939	0.555556	L-ReLU

130	0.407247	0.555556	TanH	130	0.4083	0.555556	Softmax
131	0.409333	0.555556	Sigmoid	131	0.409342	0.555556	L-ReLU
132	0.408351	0.555556	Softmax	132	0.408558	0.555556	TanH
133	0.409293	0.555556	Sigmoid	133	0.409269	0.555556	L-ReLU
134	0.409871	0.555556	ReLU	134	0.408328	0.555556	Softmax
135	0.409259	0.555556	L-ReLU	135	0.408321	0.555556	TanH
136	0.409782	0.555556	ReLU	136	0.409192	0.555556	L-ReLU
137	0.409179	0.555556	Sigmoid	137	0.409549	0.555556	ReLU
138	0.409147	0.555556	L-ReLU	138	0.408364	0.555556	Softmax
139	0.407088	0.555556	TanH	139	0.409503	0.555556	ReLU
140	0.409645	0.555556	ReLU	140	0.408341	0.555556	Softmax
141	0.408506	0.555556	Softmax	141	0.409073	0.555556	L-ReLU
142	0.409049	0.555556	L-ReLU	142	0.408316	0.555556	Softmax
143	0.406989	0.555556	TanH	143	0.409419	0.555556	ReLU
144	0.408514	0.555556	Softmax	144	0.408812	0.555556	Sigmoid
145	0.406918	0.555556	TanH	145	0.408972	0.555556	L-ReLU
146	0.409534	0.555556	ReLU	146	0.408326	0.555556	Softmax
147	0.408936	0.555556	L-ReLU	147	0.409314	0.555556	ReLU
148	0.406841	0.555556	TanH	148	0.408744	0.555556	Sigmoid
149	0.408608	0.555556	Softmax	149	0.408878	0.555556	L-ReLU
150	0.408995	0.555556	Sigmoid	150	0.409216	0.555556	ReLU
151	0.408855	0.555556	L-ReLU	151	0.408818	0.555556	L-ReLU
152	0.408585	0.555556	Softmax	152	0.408667	0.555556	Sigmoid
153	0.408826	0.555556	L-ReLU	153	0.407923	0.555556	TanH
154	0.409348	0.555556	ReLU	154	0.408741	0.555556	L-ReLU
155	0.408605	0.555556	Softmax	155	0.408485	0.555556	Softmax
156	0.408766	0.555556	L-ReLU	156	0.409079	0.555556	ReLU
157	0.408577	0.555556	Softmax	157	0.408597	0.555556	Sigmoid
158	0.408877	0.555556	Sigmoid	158	0.408461	0.555556	Softmax
159	0.408514	0.555556	Softmax	159	0.408574	0.555556	Sigmoid
160	0.409261	0.555556	ReLU	160	0.407732	0.555556	TanH
161	0.408684	0.555556	L-ReLU	161	0.408981	0.555556	ReLU
162	0.408807	0.555556	Sigmoid	162	0.408588	0.555556	L-ReLU
163	0.408546	0.555556	Softmax	163	0.407587	0.555556	TanH
164	0.406734	0.555556	TanH	164	0.408506	0.555556	Sigmoid
165	0.408514	0.555556	Softmax	165	0.40852	0.555556	L-ReLU
166	0.409167	0.555556	ReLU	166	0.408471	0.555556	Sigmoid
167	0.408493	0.555556	Softmax	167	0.408836	0.555556	ReLU
168	0.4086	0.555556	L-ReLU	168	0.408652	0.555556	Softmax
169	0.408745	0.555556	Sigmoid	169	0.40744	0.555556	TanH
170	0.408546	0.555556	L-ReLU	170	0.408434	0.555556	L-ReLU
171	0.406669	0.555556	TanH	171	0.408776	0.555556	ReLU
172	0.409043	0.555556	ReLU	172	0.407328	0.555556	TanH
173	0.408621	0.555556	Softmax	173	0.408737	0.555556	ReLU

174	0.40868	0.555556	Sigmoid
175	0.408559	0.555556	Softmax
176	0.408664	0.555556	Sigmoid
177	0.406615	0.555556	TanH
178	0.408967	0.555556	ReLU
179	0.406571	0.555556	TanH
180	0.408613	0.555556	Sigmoid
181	0.40837	0.555556	L-ReLU
182	0.406528	0.555556	TanH
183	0.408334	0.555556	L-ReLU
184	0.40855	0.555556	Sigmoid
185	0.406489	0.555556	TanH
186	0.408277	0.555556	L-ReLU
187	0.408785	0.555556	ReLU
188	0.40897	0.555556	Softmax
189	0.408497	0.555556	Sigmoid
190	0.408746	0.555556	ReLU
191	0.408955	0.555556	Softmax
192	0.408206	0.555556	L-ReLU
193	0.408448	0.555556	Sigmoid
194	0.408164	0.555556	L-ReLU
195	0.406444	0.555556	TanH
196	0.408409	0.555556	Sigmoid
197	0.409057	0.555556	Softmax
198	0.408648	0.555556	ReLU
199	0.408381	0.555556	Sigmoid
200	0.409042	0.555556	Softmax
201	0.408615	0.555556	ReLU
202	0.406409	0.555556	TanH
203	0.409058	0.555556	Softmax
204	0.408078	0.555556	L-ReLU
205	0.408579	0.555556	ReLU
206	0.409083	0.555556	Softmax
207	0.408572	0.555556	ReLU
208	0.409051	0.555556	Softmax
209	0.408041	0.555556	L-ReLU
210	0.409014	0.555556	Softmax
211	0.408035	0.555556	L-ReLU
212	0.408317	0.555556	Sigmoid
213	0.409001	0.555556	Softmax
214	0.40801	0.555556	L-ReLU
215	0.408512	0.555556	ReLU
216	0.408276	0.555556	Sigmoid
217	0.40638	0.555556	TanH

174	0.408803	0.555556	Softmax
175	0.408373	0.555556	Sigmoid
176	0.408695	0.555556	ReLU
177	0.408339	0.555556	Sigmoid
178	0.408289	0.555556	L-ReLU
179	0.408626	0.555556	ReLU
180	0.408247	0.555556	L-ReLU
181	0.408277	0.555556	Sigmoid
182	0.408561	0.555556	ReLU
183	0.40819	0.555556	L-ReLU
184	0.407165	0.555556	TanH
185	0.408508	0.555556	ReLU
186	0.407088	0.555556	TanH
187	0.408207	0.555556	Sigmoid
188	0.4092	0.555556	Softmax
189	0.408467	0.555556	ReLU
190	0.407012	0.555556	TanH
191	0.408436	0.555556	ReLU
192	0.406949	0.555556	TanH
193	0.409317	0.555556	Softmax
194	0.408159	0.555556	Sigmoid
195	0.408035	0.555556	L-ReLU
196	0.406884	0.555556	TanH
197	0.408006	0.555556	L-ReLU
198	0.408113	0.555556	Sigmoid
199	0.408335	0.555556	ReLU
200	0.407957	0.555556	L-ReLU
201	0.406814	0.555556	TanH
202	0.408291	0.555556	ReLU
203	0.409622	0.555556	Softmax
204	0.408061	0.555556	Sigmoid
205	0.409523	0.555556	Softmax
206	0.406762	0.555556	TanH
207	0.407902	0.555556	L-ReLU
208	0.406717	0.555556	TanH
209	0.409578	0.555556	Softmax
210	0.407884	0.555556	L-ReLU
211	0.40803	0.555556	Sigmoid
212	0.407855	0.555556	L-ReLU
213	0.409593	0.555556	Softmax
214	0.40785	0.555556	L-ReLU
215	0.406661	0.555556	TanH
216	0.407825	0.555556	L-ReLU
217	0.408188	0.555556	ReLU

218	0.407945	0.555556	L-ReLU	218	0.407973	0.555556	Sigmoid
219	0.406349	0.555556	TanH	219	0.408154	0.555556	ReLU
220	0.408241	0.555556	Sigmoid	220	0.40979	0.555556	Softmax
221	0.409225	0.555556	Softmax	221	0.406614	0.555556	TanH
222	0.407918	0.555556	L-ReLU	222	0.40973	0.555556	Softmax
223	0.408418	0.555556	ReLU	223	0.407777	0.555556	L-ReLU
224	0.406323	0.555556	TanH	224	0.409672	0.555556	Softmax
225	0.409299	0.555556	Softmax	225	0.407967	0.555556	Sigmoid
226	0.408414	0.555556	ReLU	226	0.40776	0.555556	L-ReLU
227	0.408202	0.555556	Sigmoid	227	0.409636	0.555556	Softmax
228	0.407852	0.555556	L-ReLU	228	0.407759	0.555556	L-ReLU
229	0.406298	0.555556	TanH	229	0.407943	0.555556	Sigmoid
230	0.408169	0.555556	Sigmoid	230	0.406567	0.555556	TanH
231	0.407813	0.555556	L-ReLU	231	0.409657	0.555556	Softmax
232	0.40948	0.555556	Softmax	232	0.406538	0.555556	TanH
233	0.407812	0.555556	L-ReLU	233	0.4096	0.555556	Softmax
234	0.408311	0.555556	ReLU	234	0.407742	0.555556	L-ReLU
235	0.4095	0.555556	Softmax	235	0.406509	0.555556	TanH
236	0.407793	0.555556	L-ReLU	236	0.407941	0.555556	Sigmoid
237	0.409454	0.555556	Softmax	237	0.407705	0.555556	L-ReLU
238	0.406278	0.555556	TanH	238	0.408094	0.555556	ReLU
239	0.407789	0.555556	L-ReLU	239	0.409748	0.555556	Softmax
240	0.40946	0.555556	Softmax	240	0.407916	0.555556	Sigmoid
241	0.40815	0.555556	Sigmoid	241	0.40966	0.555556	Softmax
242	0.407773	0.555556	L-ReLU	242	0.40792	0.555556	Sigmoid
243	0.409443	0.555556	Softmax	243	0.406474	0.555556	TanH
244	0.406258	0.555556	TanH	244	0.409627	0.555556	Softmax
245	0.408302	0.555556	ReLU	245	0.40645	0.555556	TanH
246	0.409456	0.555556	Softmax	246	0.407921	0.555556	Sigmoid
247	0.408304	0.555556	ReLU	247	0.409597	0.555556	Softmax
248	0.408121	0.555556	Sigmoid	248	0.40768	0.555556	L-ReLU
249	0.409449	0.555556	Softmax	249	0.406425	0.555556	TanH
250	0.407751	0.555556	L-ReLU	250	0.408084	0.555556	ReLU
251	0.406242	0.555556	TanH	251	0.407895	0.555556	Sigmoid
252	0.409455	0.555556	Softmax	252	0.407627	0.555556	L-ReLU
253	0.407751	0.555556	L-ReLU	253	0.4064	0.555556	TanH
254	0.406221	0.555556	TanH	254	0.409804	0.555556	Softmax
255	0.408272	0.555556	ReLU	255	0.407886	0.555556	Sigmoid
256	0.406202	0.555556	TanH	256	0.409719	0.555556	Softmax
257	0.4081	0.555556	Sigmoid	257	0.406379	0.555556	TanH
258	0.407691	0.555556	L-ReLU	258	0.407622	0.555556	L-ReLU
259	0.406186	0.555556	TanH	259	0.40804	0.555556	ReLU
260	0.408067	0.555556	Sigmoid	260	0.407862	0.555556	Sigmoid
261	0.40975	0.555556	Softmax	261	0.408004	0.555556	ReLU

262	0.408216	0.555556	ReLU	262	0.409881	0.555556	Softmax
263	0.408054	0.555556	Sigmoid	263	0.407854	0.555556	Sigmoid
264	0.40974	0.555556	Softmax	264	0.406356	0.555556	TanH
265	0.408201	0.555556	ReLU	265	0.40756	0.555556	L-ReLU
266	0.406175	0.555556	TanH	266	0.407826	0.555556	Sigmoid
267	0.40975	0.555556	Softmax	267	0.407532	0.555556	L-ReLU
268	0.408206	0.555556	ReLU	268	0.406335	0.555556	TanH
269	0.407642	0.555556	L-ReLU	269	0.410048	0.555556	Softmax
270	0.409773	0.555556	Softmax	270	0.407821	0.555556	Sigmoid
271	0.408059	0.555556	Sigmoid	271	0.409959	0.555556	Softmax
272	0.406163	0.555556	TanH	272	0.406317	0.555556	TanH
273	0.409745	0.555556	Softmax	273	0.407829	0.555556	Sigmoid
274	0.40765	0.555556	L-ReLU	274	0.409927	0.555556	Softmax
275	0.408052	0.555556	Sigmoid	275	0.406299	0.555556	TanH
276	0.409731	0.555556	Softmax	276	0.407837	0.555556	Sigmoid
277	0.40806	0.555556	Sigmoid	277	0.407519	0.555556	L-ReLU
278	0.408171	0.555556	ReLU	278	0.409955	0.555556	Softmax
279	0.409726	0.555556	Softmax	279	0.407976	0.555556	ReLU
280	0.408178	0.555556	ReLU	280	0.406283	0.555556	TanH
281	0.408029	0.555556	Sigmoid	281	0.407955	0.555556	ReLU
282	0.407588	0.555556	L-ReLU	282	0.40749	0.555556	L-ReLU
283	0.408118	0.555556	ReLU	283	0.407789	0.555556	Sigmoid
284	0.407555	0.555556	L-ReLU	284	0.407906	0.555556	ReLU
285	0.408081	0.555556	ReLU	285	0.406268	0.555556	TanH
286	0.409981	0.555556	Softmax	286	0.410243	0.555556	Softmax
287	0.408089	0.555556	ReLU	287	0.406252	0.555556	TanH
288	0.406177	0.555556	TanH	288	0.410176	0.555556	Softmax
289	0.407527	0.555556	L-ReLU	289	0.407809	0.555556	Sigmoid
290	0.407941	0.555556	Sigmoid	290	0.406238	0.555556	TanH
291	0.408036	0.555556	ReLU	291	0.407922	0.555556	ReLU
292	0.406163	0.555556	TanH	292	0.410212	0.555556	Softmax
293	0.408018	0.555556	ReLU	293	0.407933	0.555556	ReLU
294	0.406143	0.555556	TanH	294	0.407453	0.555556	L-ReLU
295	0.408	0.555556	ReLU	295	0.407899	0.555556	ReLU
296	0.407449	0.555556	L-ReLU	296	0.410295	0.555556	Softmax
297	0.410455	0.555556	Softmax	297	0.407445	0.555556	L-ReLU
298	0.407457	0.555556	L-ReLU	298	0.406227	0.555556	TanH
299	0.407981	0.555556	ReLU	299	0.407774	0.555556	Sigmoid

Iris – Softmax random				Iris – TanH random			
Iteration	**Cost**	**Acc**	**AF**	**Iteration**	**Cost**	**Acc**	**AF**
0	0.468592	0.444444	Softmax	0	0.474702	0.022222	TanH
1	0.464492	0.444444	Softmax	1	0.468133	0.555556	TanH
2	0.460727	0.444444	Softmax	2	0.46225	0.555556	TanH
3	0.460765	0.555556	ReLU	3	0.460352	0.555556	L-ReLU
4	0.45761	0.555556	ReLU	4	0.457308	0.555556	ReLU
5	0.446721	0.555556	TanH	5	0.451229	0.444444	Softmax
6	0.463746	0.444444	Sigmoid	6	0.451921	0.555556	ReLU
7	0.450259	0.555556	L-ReLU	7	0.449456	0.555556	ReLU
8	0.447867	0.555556	L-ReLU	8	0.44542	0.555556	TanH
9	0.442389	0.555556	TanH	9	0.44253	0.444444	Softmax
10	0.439087	0.444444	Softmax	10	0.444038	0.555556	ReLU
11	0.442616	0.555556	TanH	11	0.44216	0.555556	ReLU
12	0.442208	0.555556	TanH	12	0.439983	0.555556	L-ReLU
13	0.440509	0.555556	ReLU	13	0.4388	0.555556	ReLU
14	0.442943	0.555556	TanH	14	0.436782	0.555556	L-ReLU
15	0.440064	0.444444	Sigmoid	15	0.431648	0.444444	Softmax
16	0.436491	0.555556	L-ReLU	16	0.43418	0.555556	L-ReLU
17	0.435102	0.555556	L-ReLU	17	0.432924	0.555556	L-ReLU
18	0.434247	0.555556	ReLU	18	0.432302	0.555556	ReLU
19	0.432579	0.555556	L-ReLU	19	0.426748	0.444444	Softmax
20	0.426694	0.444444	Softmax	20	0.4303	0.555556	ReLU
21	0.428789	0.444444	Sigmoid	21	0.429333	0.555556	ReLU
22	0.430168	0.555556	ReLU	22	0.423837	0.444444	Softmax
23	0.429223	0.555556	ReLU	23	0.470522	0.555556	TanH
24	0.427872	0.555556	L-ReLU	24	0.462023	0.555556	TanH
25	0.454278	0.555556	TanH	25	0.426657	0.555556	L-ReLU
26	0.421722	0.444444	Softmax	26	0.42417	0.444444	Sigmoid
27	0.425987	0.555556	L-ReLU	27	0.454651	0.555556	TanH
28	0.425251	0.555556	L-ReLU	28	0.445796	0.555556	TanH
29	0.425101	0.555556	ReLU	29	0.421743	0.444444	Sigmoid
30	0.423888	0.555556	L-ReLU	30	0.420389	0.444444	Softmax
31	0.451199	0.555556	TanH	31	0.423748	0.555556	L-ReLU
32	0.418379	0.444444	Sigmoid	32	0.423121	0.555556	L-ReLU
33	0.443084	0.555556	TanH	33	0.439335	0.555556	TanH
34	0.435603	0.555556	TanH	34	0.422392	0.555556	L-ReLU
35	0.417435	0.555556	Softmax	35	0.432311	0.555556	TanH
36	0.429384	0.555556	TanH	36	0.421716	0.555556	L-ReLU
37	0.416858	0.555556	Softmax	37	0.4264	0.555556	TanH
38	0.416428	0.444444	Sigmoid	38	0.417374	0.444444	Sigmoid
39	0.416069	0.555556	Softmax	39	0.420652	0.555556	L-ReLU
40	0.415727	0.555556	Softmax	40	0.420175	0.555556	L-ReLU
41	0.420502	0.555556	ReLU	41	0.420571	0.555556	ReLU

42	0.41944	0.555556	L-ReLU	42	0.4151	0.555556	Softmax
43	0.414692	0.555556	Sigmoid	43	0.418944	0.555556	L-ReLU
44	0.425209	0.555556	TanH	44	0.414389	0.555556	Softmax
45	0.41413	0.555556	Softmax	45	0.419066	0.555556	ReLU
46	0.414173	0.555556	Sigmoid	46	0.417878	0.555556	L-ReLU
47	0.417985	0.555556	L-ReLU	47	0.413398	0.555556	Softmax
48	0.421238	0.555556	TanH	48	0.414063	0.555556	Sigmoid
49	0.417476	0.555556	L-ReLU	49	0.422614	0.555556	TanH
50	0.413486	0.555556	Sigmoid	50	0.412822	0.555556	Softmax
51	0.41266	0.555556	Softmax	51	0.416705	0.555556	L-ReLU
52	0.418133	0.555556	TanH	52	0.418614	0.555556	TanH
53	0.417209	0.555556	ReLU	53	0.417138	0.555556	ReLU
54	0.412974	0.555556	Sigmoid	54	0.411993	0.555556	Softmax
55	0.415753	0.555556	TanH	55	0.413168	0.555556	Sigmoid
56	0.41284	0.555556	Sigmoid	56	0.416412	0.555556	ReLU
57	0.416371	0.555556	ReLU	57	0.412844	0.555556	Sigmoid
58	0.412549	0.555556	Sigmoid	58	0.415946	0.555556	ReLU
59	0.415281	0.555556	L-ReLU	59	0.412554	0.555556	Sigmoid
60	0.415701	0.555556	ReLU	60	0.415515	0.555556	ReLU
61	0.410957	0.555556	Softmax	61	0.410764	0.555556	Softmax
62	0.412068	0.555556	Sigmoid	62	0.414426	0.555556	L-ReLU
63	0.414537	0.555556	L-ReLU	63	0.414943	0.555556	ReLU
64	0.414958	0.555556	ReLU	64	0.410346	0.555556	Softmax
65	0.411755	0.555556	Sigmoid	65	0.414608	0.555556	ReLU
66	0.41461	0.555556	ReLU	66	0.415762	0.555556	TanH
67	0.411567	0.555556	Sigmoid	67	0.413666	0.555556	L-ReLU
68	0.414285	0.555556	ReLU	68	0.409922	0.555556	Softmax
69	0.409983	0.555556	Softmax	69	0.414086	0.555556	ReLU
70	0.413428	0.555556	L-ReLU	70	0.413592	0.555556	TanH
71	0.411251	0.555556	Sigmoid	71	0.413164	0.555556	L-ReLU
72	0.409732	0.555556	Softmax	72	0.413716	0.555556	ReLU
73	0.413736	0.555556	TanH	73	0.412861	0.555556	L-ReLU
74	0.413547	0.555556	ReLU	74	0.413416	0.555556	ReLU
75	0.41283	0.555556	L-ReLU	75	0.409283	0.555556	Softmax
76	0.410989	0.555556	Sigmoid	76	0.411976	0.555556	TanH
77	0.40937	0.555556	Softmax	77	0.41314	0.555556	ReLU
78	0.41087	0.555556	Sigmoid	78	0.411098	0.555556	Sigmoid
79	0.412266	0.555556	TanH	79	0.410838	0.555556	TanH
80	0.412336	0.555556	L-ReLU	80	0.412849	0.555556	ReLU
81	0.412781	0.555556	ReLU	81	0.408948	0.555556	Softmax
82	0.410697	0.555556	Sigmoid	82	0.410917	0.555556	Sigmoid
83	0.412011	0.555556	L-ReLU	83	0.409999	0.555556	TanH
84	0.411126	0.555556	TanH	84	0.412507	0.555556	ReLU
85	0.411831	0.555556	L-ReLU	85	0.409387	0.555556	TanH

86	0.408884	0.555556	Softmax	86	0.412347	0.555556	ReLU
87	0.411673	0.555556	L-ReLU	87	0.408701	0.555556	Softmax
88	0.410296	0.555556	TanH	88	0.408915	0.555556	TanH
89	0.410452	0.555556	Sigmoid	89	0.412138	0.555556	ReLU
90	0.411998	0.555556	ReLU	90	0.411331	0.555556	L-ReLU
91	0.411337	0.555556	L-ReLU	91	0.408518	0.555556	TanH
92	0.40866	0.555556	Softmax	92	0.408562	0.555556	Softmax
93	0.41176	0.555556	ReLU	93	0.410538	0.555556	Sigmoid
94	0.411115	0.555556	L-ReLU	94	0.411071	0.555556	L-ReLU
95	0.411584	0.555556	ReLU	95	0.411677	0.555556	ReLU
96	0.410949	0.555556	L-ReLU	96	0.410375	0.555556	Sigmoid
97	0.40852	0.555556	Softmax	97	0.410825	0.555556	L-ReLU
98	0.410073	0.555556	Sigmoid	98	0.410276	0.555556	Sigmoid
99	0.41131	0.555556	ReLU	99	0.408395	0.555556	Softmax
100	0.408439	0.555556	Softmax	100	0.408132	0.555556	TanH
101	0.409465	0.555556	TanH	101	0.411286	0.555556	ReLU
102	0.410613	0.555556	L-ReLU	102	0.407894	0.555556	TanH
103	0.411085	0.555556	ReLU	103	0.411174	0.555556	ReLU
104	0.409882	0.555556	Sigmoid	104	0.410097	0.555556	Sigmoid
105	0.410418	0.555556	L-ReLU	105	0.410354	0.555556	L-ReLU
106	0.410885	0.555556	ReLU	106	0.410011	0.555556	Sigmoid
107	0.410291	0.555556	L-ReLU	107	0.410233	0.555556	L-ReLU
108	0.408894	0.555556	TanH	108	0.407637	0.555556	TanH
109	0.410722	0.555556	ReLU	109	0.409924	0.555556	Sigmoid
110	0.409703	0.555556	Sigmoid	110	0.410744	0.555556	ReLU
111	0.41008	0.555556	L-ReLU	111	0.409839	0.555556	Sigmoid
112	0.408356	0.555556	Softmax	112	0.409968	0.555556	L-ReLU
113	0.409618	0.555556	Sigmoid	113	0.409763	0.555556	Sigmoid
114	0.410473	0.555556	ReLU	114	0.407426	0.555556	TanH
115	0.409552	0.555556	Sigmoid	115	0.409833	0.555556	L-ReLU
116	0.408296	0.555556	Softmax	116	0.410426	0.555556	ReLU
117	0.408454	0.555556	TanH	117	0.409727	0.555556	L-ReLU
118	0.410319	0.555556	ReLU	118	0.409617	0.555556	Sigmoid
119	0.408283	0.555556	Softmax	119	0.408434	0.555556	Softmax
120	0.409742	0.555556	L-ReLU	120	0.409568	0.555556	Sigmoid
121	0.408165	0.555556	TanH	121	0.407245	0.555556	TanH
122	0.409414	0.555556	Sigmoid	122	0.408398	0.555556	Softmax
123	0.407947	0.555556	TanH	123	0.410174	0.555556	ReLU
124	0.410103	0.555556	ReLU	124	0.408376	0.555556	Softmax
125	0.409542	0.555556	L-ReLU	125	0.40949	0.555556	L-ReLU
126	0.410006	0.555556	ReLU	126	0.409442	0.555556	Sigmoid
127	0.409456	0.555556	L-ReLU	127	0.409406	0.555556	L-ReLU
128	0.409267	0.555556	Sigmoid	128	0.409379	0.555556	Sigmoid
129	0.4077	0.555556	TanH	129	0.407095	0.555556	TanH

130	0.40935	0.555556	L-ReLU	130	0.409342	0.555556	Sigmoid
131	0.40981	0.555556	ReLU	131	0.409868	0.555556	ReLU
132	0.408462	0.555556	Softmax	132	0.409278	0.555556	Sigmoid
133	0.409757	0.555556	ReLU	133	0.40978	0.555556	ReLU
134	0.409142	0.555556	Sigmoid	134	0.409217	0.555556	Sigmoid
135	0.409192	0.555556	L-ReLU	135	0.408488	0.555556	Softmax
136	0.407493	0.555556	TanH	136	0.409693	0.555556	ReLU
137	0.408494	0.555556	Softmax	137	0.406965	0.555556	TanH
138	0.407374	0.555556	TanH	138	0.409635	0.555556	ReLU
139	0.408463	0.555556	Softmax	139	0.409011	0.555556	L-ReLU
140	0.407273	0.555556	TanH	140	0.409094	0.555556	Sigmoid
141	0.409054	0.555556	Sigmoid	141	0.409519	0.555556	ReLU
142	0.408432	0.555556	Softmax	142	0.408633	0.555556	Softmax
143	0.409022	0.555556	Sigmoid	143	0.406863	0.555556	TanH
144	0.40837	0.555556	Softmax	144	0.409031	0.555556	Sigmoid
145	0.409499	0.555556	ReLU	145	0.408604	0.555556	Softmax
146	0.408966	0.555556	L-ReLU	146	0.408862	0.555556	L-ReLU
147	0.408386	0.555556	Softmax	147	0.408575	0.555556	Softmax
148	0.407145	0.555556	TanH	148	0.409402	0.555556	ReLU
149	0.408357	0.555556	Softmax	149	0.40678	0.555556	TanH
150	0.40893	0.555556	Sigmoid	150	0.40859	0.555556	Softmax
151	0.408301	0.555556	Softmax	151	0.409357	0.555556	ReLU
152	0.409375	0.555556	ReLU	152	0.40875	0.555556	L-ReLU
153	0.408877	0.555556	Sigmoid	153	0.409285	0.555556	ReLU
154	0.409305	0.555556	ReLU	154	0.408663	0.555556	Softmax
155	0.407039	0.555556	TanH	155	0.409255	0.555556	ReLU
156	0.408823	0.555556	Sigmoid	156	0.408848	0.555556	Sigmoid
157	0.408738	0.555556	L-ReLU	157	0.408633	0.555556	L-ReLU
158	0.408778	0.555556	Sigmoid	158	0.408695	0.555556	Softmax
159	0.408684	0.555556	L-ReLU	159	0.40861	0.555556	L-ReLU
160	0.408453	0.555556	Softmax	160	0.408783	0.555556	Sigmoid
161	0.408735	0.555556	Sigmoid	161	0.40669	0.555556	TanH
162	0.408633	0.555556	L-ReLU	162	0.408756	0.555556	Sigmoid
163	0.406927	0.555556	TanH	163	0.40873	0.555556	Softmax
164	0.408688	0.555556	Sigmoid	164	0.408736	0.555556	Sigmoid
165	0.408564	0.555556	L-ReLU	165	0.40866	0.555556	Softmax
166	0.408521	0.555556	Softmax	166	0.40663	0.555556	TanH
167	0.408542	0.555556	L-ReLU	167	0.408493	0.555556	L-ReLU
168	0.408989	0.555556	ReLU	168	0.408694	0.555556	Sigmoid
169	0.406842	0.555556	TanH	169	0.408442	0.555556	L-ReLU
170	0.408474	0.555556	L-ReLU	170	0.408739	0.555556	Softmax
171	0.408627	0.555556	Softmax	171	0.408657	0.555556	Sigmoid
172	0.408589	0.555556	Sigmoid	172	0.408942	0.555556	ReLU
173	0.408561	0.555556	Softmax	173	0.406571	0.555556	TanH

174	0.408437	0.555556	L-ReLU	174	0.408362	0.555556	L-ReLU
175	0.408553	0.555556	Sigmoid	175	0.408878	0.555556	ReLU
176	0.406767	0.555556	TanH	176	0.408882	0.555556	Softmax
177	0.408584	0.555556	Softmax	177	0.408575	0.555556	Sigmoid
178	0.40672	0.555556	TanH	178	0.408809	0.555556	Softmax
179	0.40853	0.555556	Sigmoid	179	0.408562	0.555556	Sigmoid
180	0.408823	0.555556	ReLU	180	0.40652	0.555556	TanH
181	0.408322	0.555556	L-ReLU	181	0.408275	0.555556	L-ReLU
182	0.408661	0.555556	Softmax	182	0.408788	0.555556	ReLU
183	0.408307	0.555556	L-ReLU	183	0.408497	0.555556	Sigmoid
184	0.406659	0.555556	TanH	184	0.408735	0.555556	ReLU
185	0.408272	0.555556	L-ReLU	185	0.408456	0.555556	Sigmoid
186	0.408724	0.555556	ReLU	186	0.408683	0.555556	ReLU
187	0.408419	0.555556	Sigmoid	187	0.409051	0.555556	Softmax
188	0.408207	0.555556	L-ReLU	188	0.406474	0.555556	TanH
189	0.408384	0.555556	Sigmoid	189	0.409005	0.555556	Softmax
190	0.408854	0.555556	Softmax	190	0.408681	0.555556	ReLU
191	0.406601	0.555556	TanH	191	0.406436	0.555556	TanH
192	0.40837	0.555556	Sigmoid	192	0.409022	0.555556	Softmax
193	0.408828	0.555556	Softmax	193	0.408132	0.555556	L-ReLU
194	0.406562	0.555556	TanH	194	0.408407	0.555556	Sigmoid
195	0.408789	0.555556	Softmax	195	0.40809	0.555556	L-ReLU
196	0.408366	0.555556	Sigmoid	196	0.4064	0.555556	TanH
197	0.406526	0.555556	TanH				
198	0.408345	0.555556	Sigmoid	197	0.408368	0.555556	Sigmoid
199	0.408103	0.555556	L-ReLU	198	0.406369	0.555556	TanH
200	0.408834	0.555556	Softmax	199	0.409183	0.555556	Softmax
201	0.40832	0.555556	Sigmoid	200	0.408048	0.555556	L-ReLU
202	0.408075	0.555556	L-ReLU	201	0.408559	0.555556	ReLU
203	0.408527	0.555556	ReLU	202	0.406341	0.555556	TanH
204	0.408874	0.555556	Softmax	203	0.40853	0.555556	ReLU
205	0.408045	0.555556	L-ReLU	204	0.408302	0.555556	Sigmoid
206	0.408262	0.555556	Sigmoid	205	0.409345	0.555556	Softmax
207	0.408858	0.555556	Softmax	206	0.406315	0.555556	TanH
208	0.408489	0.555556	ReLU	207	0.409293	0.555556	Softmax
209	0.40883	0.555556	Softmax	208	0.40798	0.555556	L-ReLU
210	0.408249	0.555556	Sigmoid	209	0.408496	0.555556	ReLU
211	0.408459	0.555556	ReLU	210	0.407939	0.555556	L-ReLU
212	0.408821	0.555556	Softmax	211	0.406292	0.555556	TanH
213	0.407982	0.555556	L-ReLU	212	0.40943	0.555556	Softmax
214	0.40821	0.555556	Sigmoid	213	0.408276	0.555556	Sigmoid
215	0.407946	0.555556	L-ReLU	214	0.407911	0.555556	L-ReLU
216	0.406473	0.555556	TanH	215	0.408425	0.555556	ReLU
217	0.408905	0.555556	Softmax	216	0.408221	0.555556	Sigmoid

218	0.40819	0.555556	Sigmoid	217	0.408381	0.555556	ReLU
219	0.408383	0.555556	ReLU	218	0.407836	0.555556	L-ReLU
220	0.407897	0.555556	L-ReLU	219	0.408169	0.555556	Sigmoid
221	0.408949	0.555556	Softmax	220	0.40627	0.555556	TanH
222	0.408358	0.555556	ReLU	221	0.409708	0.555556	Softmax
223	0.406439	0.555556	TanH	222	0.408169	0.555556	Sigmoid
224	0.408334	0.555556	ReLU	223	0.406247	0.555556	TanH
225	0.408118	0.555556	Sigmoid	224	0.409672	0.555556	Softmax
226	0.408292	0.555556	ReLU	225	0.406227	0.555556	TanH
227	0.408084	0.555556	Sigmoid	226	0.408169	0.555556	Sigmoid
228	0.406408	0.555556	TanH	227	0.406209	0.555556	TanH
229	0.40825	0.555556	ReLU	228	0.407778	0.555556	L-ReLU
230	0.409233	0.555556	Softmax	229	0.406193	0.555556	TanH
231	0.407787	0.555556	L-ReLU	230	0.407756	0.555556	L-ReLU
232	0.408052	0.555556	Sigmoid	231	0.408119	0.555556	Sigmoid
233	0.409212	0.555556	Softmax	232	0.409905	0.555556	Softmax
234	0.408052	0.555556	Sigmoid	233	0.407741	0.555556	L-ReLU
235	0.407754	0.555556	L-ReLU	234	0.408107	0.555556	Sigmoid
236	0.408195	0.555556	ReLU	235	0.406178	0.555556	TanH
237	0.406379	0.555556	TanH	236	0.408236	0.555556	ReLU
238	0.408174	0.555556	ReLU	237	0.410003	0.555556	Softmax
239	0.406351	0.555556	TanH	238	0.408239	0.555556	ReLU
240	0.407696	0.555556	L-ReLU	239	0.407686	0.555556	L-ReLU
241	0.406325	0.555556	TanH	240	0.40806	0.555556	Sigmoid
242	0.409519	0.555556	Softmax	241	0.406164	0.555556	TanH
243	0.408154	0.555556	ReLU	242	0.40818	0.555556	ReLU
244	0.407674	0.555556	L-ReLU	243	0.406149	0.555556	TanH
245	0.406303	0.555556	TanH	244	0.407629	0.555556	L-ReLU
246	0.409585	0.555556	Softmax	245	0.410298	0.555556	Softmax
247	0.40767	0.555556	L-ReLU	246	0.408034	0.555556	Sigmoid
248	0.409536	0.555556	Softmax	247	0.408149	0.555556	ReLU
249	0.408142	0.555556	ReLU	248	0.408001	0.555556	Sigmoid
250	0.406282	0.555556	TanH	249	0.410296	0.555556	Softmax
251	0.408121	0.555556	ReLU	250	0.408007	0.555556	Sigmoid
252	0.409611	0.555556	Softmax	251	0.40759	0.555556	L-ReLU
253	0.406263	0.555556	TanH	252	0.406139	0.555556	TanH
254	0.408122	0.555556	ReLU	253	0.408101	0.555556	ReLU
255	0.40962	0.555556	Softmax	254	0.406125	0.555556	TanH
256	0.406245	0.555556	TanH	255	0.407553	0.555556	L-ReLU
257	0.40764	0.555556	L-ReLU	256	0.407948	0.555556	Sigmoid
258	0.409623	0.555556	Softmax	257	0.407526	0.555556	L-ReLU
259	0.408131	0.555556	ReLU	258	0.40792	0.555556	Sigmoid
260	0.407967	0.555556	Sigmoid	259	0.41064	0.555556	Softmax
261	0.407611	0.555556	L-ReLU	260	0.406114	0.555556	TanH

262	0.40623	0.555556	TanH	261	0.407514	0.555556	L-ReLU
263	0.408075	0.555556	ReLU	262	0.408027	0.555556	ReLU
264	0.409786	0.555556	Softmax	263	0.4079	0.555556	Sigmoid
265	0.40808	0.555556	ReLU	264	0.407474	0.555556	L-ReLU
266	0.409743	0.555556	Softmax	265	0.407873	0.555556	Sigmoid
267	0.407949	0.555556	Sigmoid	266	0.407963	0.555556	ReLU
268	0.406215	0.555556	TanH	267	0.410898	0.555556	Softmax
269	0.408066	0.555556	ReLU	268	0.406106	0.555556	TanH
270	0.407561	0.555556	L-ReLU	269	0.410813	0.555556	Softmax
271	0.407903	0.555556	Sigmoid	270	0.407469	0.555556	L-ReLU
272	0.409866	0.555556	Softmax	271	0.407984	0.555556	ReLU
273	0.40791	0.555556	Sigmoid	272	0.407442	0.555556	L-ReLU
274	0.407537	0.555556	L-ReLU	273	0.407853	0.555556	Sigmoid
275	0.409843	0.555556	Softmax	274	0.407417	0.555556	L-ReLU
276	0.408031	0.555556	ReLU	275	0.407924	0.555556	ReLU
277	0.407525	0.555556	L-ReLU	276	0.411047	0.555556	Softmax
278	0.409859	0.555556	Softmax	277	0.407411	0.555556	L-ReLU
279	0.406207	0.555556	TanH	278	0.407921	0.555556	ReLU
280	0.408023	0.555556	ReLU	279	0.41104	0.555556	Softmax
281	0.407512	0.555556	L-ReLU	280	0.406105	0.555556	TanH
282	0.407989	0.555556	ReLU	281	0.407836	0.555556	Sigmoid
283	0.407854	0.555556	Sigmoid	282	0.407391	0.555556	L-ReLU
284	0.407954	0.555556	ReLU	283	0.40781	0.555556	Sigmoid
285	0.406197	0.555556	TanH	284	0.411082	0.555556	Softmax
286	0.407824	0.555556	Sigmoid	285	0.406093	0.555556	TanH
287	0.406179	0.555556	TanH	286	0.407916	0.555556	ReLU
288	0.407438	0.555556	L-ReLU	287	0.407371	0.555556	L-ReLU
289	0.406163	0.555556	TanH	288	0.411136	0.555556	Softmax
290	0.4078	0.555556	Sigmoid	289	0.407914	0.555556	ReLU
291	0.407891	0.555556	ReLU	290	0.407808	0.555556	Sigmoid
292	0.41043	0.555556	Softmax	291	0.411097	0.555556	Softmax
293	0.407795	0.555556	Sigmoid	292	0.40782	0.555556	Sigmoid
294	0.406152	0.555556	TanH	293	0.407361	0.555556	L-ReLU
295	0.407781	0.555556	Sigmoid	294	0.40609	0.555556	TanH
296	0.407869	0.555556	ReLU	295	0.411109	0.555556	Softmax
297	0.407375	0.555556	L-ReLU	296	0.407823	0.555556	Sigmoid
298	0.407742	0.555556	Sigmoid	297	0.410999	0.555556	Softmax
299	0.410575	0.555556	Softmax	298	0.407926	0.555556	ReLU
				299	0.40782	0.555556	Sigmoid