**To what extent does the number of neurons affect the accuracy of an artificial neural network?**

*Computer Science*

Candidate Number: 000251-0171

Word Count: 3857

**Table of Contents**

1. **Introduction**…………………………………………………………………………... p3
   1. The Importance of Machine learning………………………………………….. p3
   2. The Research Question………………………………………………………… p4
2. **Overview of artificial neural networks**……………………………………………… p5
   1. General structure……………………………………………………………...... p5
   2. Evaluation of Neuron Processing…………………..…………………………... p9
3. **Experiment**……………..……………………………………………………………... p10
   1. Dataset………………………………………………………………………….. p11
   2. Experimental Constants………………………………………………………... p12
   3. Experimental Setup…………………………………………………………….. p13
   4. Hypothesis……………………………………………………………………... p14
   5. Results and graphs……………………………………………………………... p15
4. **Interpretation of Results**……………………………………………………………... p17
   1. Analysis………………………………………………………………………... p17
   2. Limitations of the Investigation………………………………………………... p18
5. **Methods to Counter Overfitting**………………………………………………........... p20
6. **Conclusion**…………………………………………………………………………...... p21
7. **Annotated Bibliography**……………………………………………………………… p23
8. **Appendix**………………………………………………………………………………. p26

**1. Introduction**

The 20th century marked a new age for technology,[[1]](#footnote-1) with the digital revolution taking hold and a new emphasis on computing technology. These developments not only allowed for greater access to technology, but also led to an influx of digital data. Large quantities of information could now be stored efficiently on electronic devices— a utility that the world easily took advantage of. Since then, our manipulation of digital data has only become more paramount to the future of computing. It is predicted that 90% of all available data has been created within the last two years,[[2]](#footnote-2) testifying to the mounting importance of data analytics to modern society. In 2005, the term “big data” was first coined, referring to extremely large data sets that can be computationally analyzed to reveal patterns.[[3]](#footnote-3) This great availability of data can prove beneficial for many, allowing for scientists to predict future trends and for companies to make estimates about their products. However, with this increasing quantity of data comes a predicament: how can we study this data in an efficient manner? A solution that has recently been growing in popularity is machine learning.

* 1. **The Importance of Machine learning**

Machine learning is a form of artificial intelligence wherein a program “learns” from analyzing data rather than from being explicitly programmed. More specifically, it is an approach to data analysis where a computer program studies the patterns of a dataset and progressively improves its algorithm to better fit said data.[[4]](#footnote-4) A common example of machine learning algorithms is in major search engines, such as Google or Bing. Machine learning is used to both understand search queries, such as with sentence prediction and recommended searches, and to rank search results.[[5]](#footnote-5) Data of what previous users have searched and how often links are clicked are sent through machine learning algorithms to create the most effective search engine.

Machine learning has become so pervasive today that implementations of it can be found almost anywhere. In the past decade, the increasing focus on machine learning has yielded self-driving cars, practical speech recognition, online shopping recommendations, and an improved understanding of the human genome.[[6]](#footnote-6) Machine learning algorithms are able to study large amounts of data both faster and more accurately than humans could, leading to their growing strength in data analysis. Many researches even believe that machine learning is the best route to take to progress towards human-level artificial intelligence[[7]](#footnote-7)— a long term goal that has been steadily growing closer.

**1.2 The Research Question**

Artificial neural networks are a specific type of machine learning algorithm that can effectively model nonlinear and complex datasets. This makes them imperative in understanding real-life data, which often exhibits such characteristics. For this reason, artificial neural networks have grown to dominate the field of machine learning, being a flexible and powerful tool for evaluating big data. The growing importance of artificial neural networks to the future of computing has led me to investigate the topic: *“To what extent does the number of neurons impact the accuracy of an artificial neural network?”* In exploring this issue, a greater understanding of artificial neural networks, and thus a greater appreciation for machine learning algorithms, will be produced.

**2 Overview of Artificial Neural Networks**

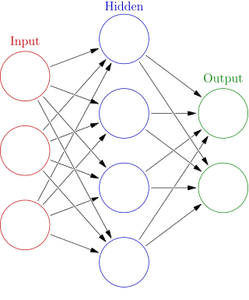
Due to the topic of this investigation, a concrete understanding of neural networks must first be established. The simplest definition of an artificial neural network is given by Dr. Robert Hecht-Nielsen— the inventor of one of the first neurocomputers. He describes a neural network as "a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs”.[[8]](#footnote-8) In essence, a neural network is a machine learning algorithm constructed of a multitude of these “highly interconnected processing elements,” otherwise known as neurons.

**2.1 General Structure**

A neuron is the most fundamental unit of a neural network that individually processes inputs to create and send outputs. Neural networks can be compromised of anything between a few dozen neurons to a couple thousand depending on the required complexity of the program. These units are designed to analyze the various inputs they receive to recognize patterns within a dataset, allowing the neural network to learn.[[9]](#footnote-9)

As previously stated, a neural network consists of many interconnected neurons that process inputs and send computed outputs to other neurons. Information is passed down between the connections between neurons, with one neuron’s output acting as another neuron’s input. The data is passed down and processed by neurons in the neural network until the final group of neurons is reached, who return the neural network’s predicted output for the inputted data from the dataset.

Neurons can be categorized into groups called “layers” based on their location within the neural network, with data being passed from layer to layer.[[10]](#footnote-10) The diagram of a neural network below displays this relationship, with each circle representing a neuron and every column of colored neurons representing a layer. The black arrows indicate the flow of data in the neural network.



**Figure 1: A diagram of a basic neural network[[11]](#footnote-11)**

The neural network can be further subdivided by the type of layer. The first and leftmost layer is the input layer of the neural network. There can only be one input layer, and the size of the input layer must match the number of inputs to the program— each neuron in this layer corresponds to a specific input variable.[[12]](#footnote-12) The last and rightmost layer is the output layer of the neural network. Like input layer, the size of the output layer must match the number of program outputs.[[13]](#footnote-13) Sandwiched between these two main layers is a “hidden layer,” dubbed as such because, unlike the input and output layers, it is not defined by the dataset. Most of the major data processing occurs in this hidden layer, making it imperative to a neural network.[[14]](#footnote-14) Data is processed in and passed through the layers in a neural network to produce an output.

To train the algorithm, the neural network runs through the training data and returns the predicted outputs. These outputs are compared with the real outputs in the dataset to find the neural network’s error. Using the error, the neural network adjusts the neurons’ calculations accordingly in a process called backpropagation.[[15]](#footnote-15) This process alters internal neuron variables to improve the neurons’ processing and the resultant neural network outputs. Backpropagation happens for a set amount of iterations through the dataset, effectively training the neural network to fit the data. After having been trained on data, the neural network is able to model the dataset and predict new inputs.[[16]](#footnote-16)

With the proper structure and training, a neural network can become highly capable in predicting complex data. They do not, however, come without some considerable problems. One of the most formidable challenges that arises when using an artificial neural network is how to structure the hidden layer. Despite the significance of this major processing layer to the neural network, the hidden layer is somewhat of a gray area as it must be defined by the programmer rather than by the data. It therefore becomes difficult to structure the neural network program and determine how many neurons to use. Such ambiguity in regards to this essential feature led me to investigate the effect of the number of neurons to the accuracy of a neural network.

**2.2 Evaluation of Neuron Processing**

Specific operations are performed by neurons to process the inputted data. Using the input values, internal variables called weights and biases, and an activation function, each neuron computes an output to send to the next layer. These processes allow the neuron to “learn” how to use the input data to generate the proper outputs.

A “weight” is a number that indicates the significance of the input to their respective output. Conversely, a “bias” is a number that shifts the input value to better match the output. These variables are used to interpret the inputted values. Within each neuron, every input in multiplied by a weight and added with a bias, as shown by the equation[[17]](#footnote-17):

*y = wx + b*

Where *x* is the input value, *w* is the weight, and *b* is the bias.

This equation is applied to all of the inputs received by the neuron. The resultant values are then summed up, as follows[[18]](#footnote-18):

Where *s* is the sum and *N* is the total number of inputs received by the neuron

To calculate the neuron output, the neuron passes this sum into an “activation function,” which is a non-linear mathematical equation. Activation functions are significant because they allow neurons to generate a non-linear output, unlike the linear *y = wx + b* operation. Most real-world datasets cannot be perfectly represented by a linear model, making this a necessary function.

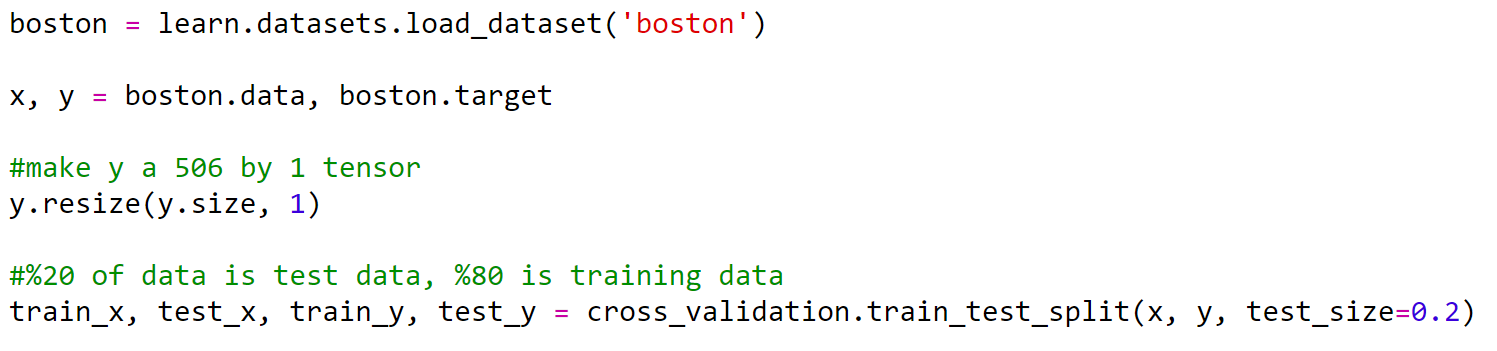
Based on these equations, it can be inferred that the increasing the number of neurons in a neural network increases the program’s analysis of the data and the program’s ability to model complex data. Incrementing the number of neurons means that the data is passed through more neurons and is evaluated further. It also means that the neural network become more capable of recognizing the specific characteristics of the dataset, as each neuron aims to learn the relationship between the input and output data. Incrementing the number of neurons therefore means that the neural network becomes more capable of learning specific details of the dataset.

**3 Experiment**

The focus of this investigation is the impact of the number of neurons on the accuracy of a neural network’s predictions. In order to investigate the research question, a python program (see appendix) will be used to test a neural network with a variable number of neurons in the hidden layer. Two python machine learning libraries are used to accomplish this: the TensorFlow library and the Scikit-Learn library. Additionally, the python Metrics library will be used for performing mathematical operations, such as calculating error and the coefficient of determination, r2.

**3.1 Dataset**

The dataset that is being used in this program is the “Boston Housing Data Set” from the UCI Machine Learning Repository[[19]](#footnote-19). This dataset is a collection of 506 data points of Boston housing prices, with 13 numeric input variables and 1 numeric output variable. The size and complexity of this dataset makes it suitable for a neural network program, which is built to analyze large, convoluted data. Although this dataset is not large enough to be considered “big data,” it still contains a relatively large amount of data and is small enough to be processed within a reasonable amount of time.



**Figure 1: The implementation of the dataset into the program**

Before creating the neural network, the data set is loaded into the program via the above Scikit-Learn function. The inputs are assigned to the matrix variable x, while the outputs are assigned to the matrix variable y.

The data is then split up into training and testing data. Eighty percent of the dataset, or 405 data points, will be used to train the neural network. The remaining twenty percent of data, or 101 data points, will be used to evaluate the accuracy of the trained neural network. It is imperative that the neural network has copious amounts of data to train on so that the trained neural network will be flexible on a variety of data.

**3.2 Experimental Constants**

As with the number of neurons, there are certain aspects of a neural network that must be decided beforehand by the programmer. To make the results of this experiment as consistent and general as possible, this neural network will use fundamental neural network algorithms and constant values for variables.

The activation function being implemented is the sigmoid activation function. It is defined as[[20]](#footnote-20):

Where *g(x)* is the activation function and *s* is the neuron sum

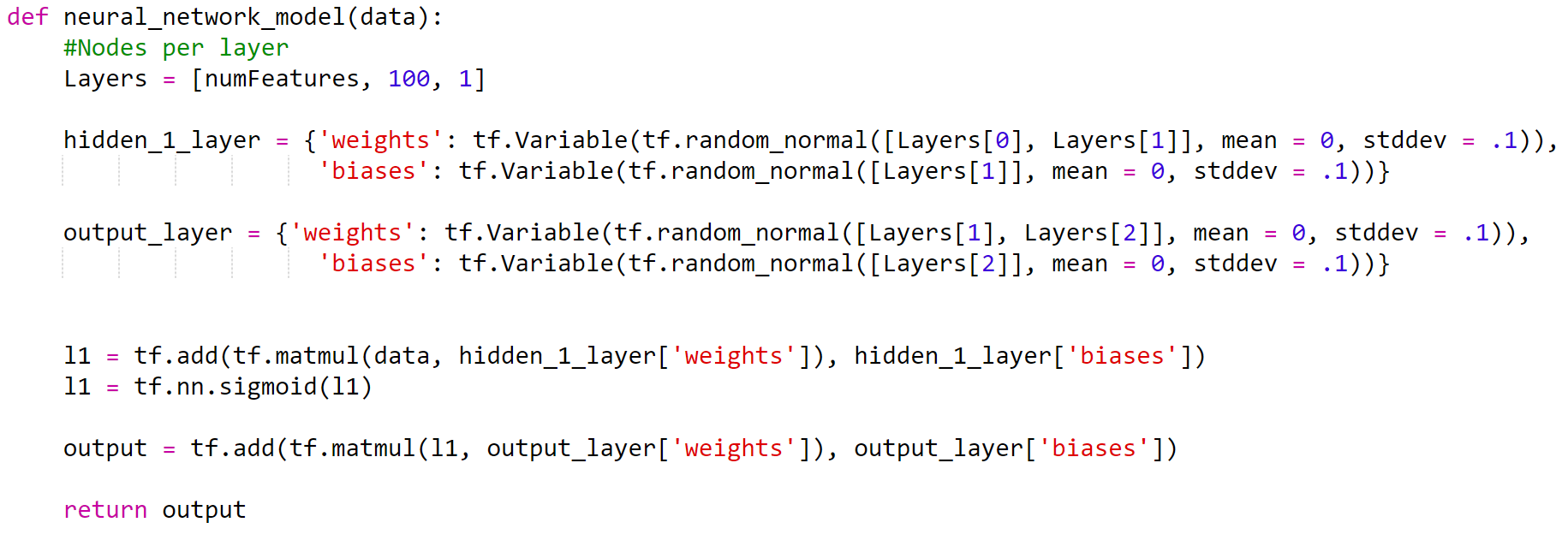
This is a non-linear function, making it an effective activation function. It is also a differentiable function, which is required for backpropagation.

The backpropagation algorithm that will be used is gradient descent. This is a traditional backpropagation algorithm that calculates the changes in the neuron weights and biases to minimize the neural network’s error[[21]](#footnote-21). The gradient descent calculation will be handled by the TensorFlow library due to its complexity. Essentially, this algorithm uses a neuron’s current weights, current biases, output, and the derivative of the activation function to calculate the changes for a neuron.

For every trial, the neural network will train for 50,000 iterations on the training data, allowing the neural network to learn the data thoroughly.

**3.3 Experimental Setup**

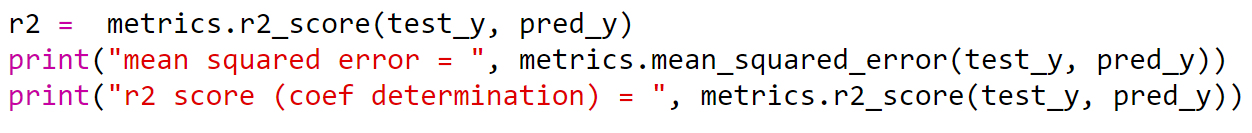
A python function will be created to create the neural network to train and test on the Boston Housing Data. This function, named “neural\_network\_model,” will take in data and run the neural network on it.



**Figure 2: The neural network python function**

This function holds an array of integers for the number of neurons in each hidden layer. Using the values in this array, python dictionaries are created for each hidden layer. Each dictionary initializes a vector of weights and biases for every neuron in the layer. The weights and biases are initialized to [-1, 1] rather than to 0 so that each neuron behaves differently and sends a unique output[[22]](#footnote-22). Since these initial values are so close to zero, this randomization should have a minimal impact on the results of this investigation. The weights and biases will eventually be altered as the neural network learns. The Layers array defines the number of neurons in each layer (see figure 2). The value of Layers[1] will be altered in this experiment to change the number of neurons in the hidden layer between trials.

Once the neural network has been trained, the accuracy of the neural network will be calculated using the test data. For each trial, the neural network error and the coefficient of determination (r2) will be calculated using functions from the Metrics library, as displayed in figure 3 below. The neural network error indicates how far away the predicted outputs are from the actual data outputs. The coefficient of determination is a statistical measure for how well a model fits data. These two values will signify the accuracy of the neural network on the testing data.



**Figure 3: Python calculations using the Metrics library**

**3.4 Hypothesis**

For this investigation, I hypothesize that as the number of neurons increases, the neural network’s accuracy will improve. This hypothesis is based on the belief that increasing the number of neurons allows the neural network to analyze the training data better.

**3.5 Results and Graphs**

|  |  |  |
| --- | --- | --- |
| Number of Neurons | Testing error | Coefficient of Determination (r2) |
| 100 | 13.0690 | .7821 |
| 200 | 22.8390 | .6822 |
| 300 | 15.8126 | .7855 |
| 400 | 20.2276 | .7302 |
| 500 | 13.1374 | .8588 |
| 600 | 9.1950 | .8947 |
| 700 | 18.8916 | .8149 |
| 800 | 14.3798 | .8073 |
| 900 | 18.0792 | .7435 |
| 1000 | 19.5112 | .7727 |

**Table 1: Number of neurons vs testing error and r2**

**Graph 1: Number of Neurons graphed against the testing error**

**Graph 2: Number of neurons graphed against the coefficient of determination**

The above data was produces by creating a neural network with one hidden layer and augmenting the hidden layer’s size. The number of neurons was increased by 100 neurons between each trial.

As predicted by the hypothesis, the accuracy of the neural network initially improves its accuracy as more neurons are added. From 100 to 600 neurons, the neural network had an increasing coefficient of determination and a decreasing testing error. Both of these characteristics indicate that the neural network is fitting the test data and modeling the dataset better. The neural network, however, does not seem consistently improve. The accuracy peaks at 600 neurons then declines as more neurons are added, contradicting the hypothesis. The accuracy also seems to fluctuate between each point, suggesting a nonlinear relationship between the number of neurons and the neural network accuracy.

**4 Interpretation of Results**

**4.1 Analysis**

Despite my initial hypothesis that increasing the number of neurons would indefinitely improve the accuracy of the neural network, the data indicates that the accuracy peaks at a certain point then declines. Intuitively, I believed that if the neural network analyzed the training data more, then the neural network would perform better on new data that it had not been train on (i.e. the 101 testing data point). The results of the experiment, however, did not fully reflect this inference. Instead, the training accuracy increased before declining after the hidden layer reached 600 neurons. Upon further research on this unexpected pattern, I believe that the cause of my results is due to the inherent differences in “noise” between the training data and the testing data.

Real-world data can be described by two aspects: signal and noise. In data modeling, “signal” can be understood as the true underlying pattern of data that a neural network aims to learn.[[23]](#footnote-23) Conversely, “noise” refers to the inaccuracies or inherent randomness that causes data to stray from the signal.[[24]](#footnote-24) Instead of data perfectly aligning with the signal of a dataset, as is ideal, real-world data is generally noisy and has error with the signal. With regards to the results of this investigation, I believe that the accuracy began to decrease after a certain point because the neural network was fitting the training data *too well*. The noise of each data point varies, meaning that the noise of the training data is different than the noise of the testing data. When there were too many neurons, the neural network began memorizing the noise of the training data rather than modeling the signal for the entire dataset.

Underfitting occurs when the neural network’s model is too simple– the lack of model complexity renders the network unable to fully understand the data.[[25]](#footnote-25) When there were too few neurons in the neural network, the program was unable to properly learn the dataset’s signal, resulting in inaccurate predictions. As neurons are added, however, the model is able to analyze the training data more and better learn the signal. This is seen in the experimental data, as the neural network accuracy roughly increases until 600 neurons are reached.

After this peak in accuracy, however, the model overfits the dataset. The neural network begins to over-analyze the training data and creates a model that specifically matches these data points. In doing so, the neural network begins to model the specific noise of the training data. This is not ideal as noise makes the training data not completely representative of the signal. Consequently, the model begins to alter itself to predict the exact values of the training data rather than modeling the dataset’s signal. The neural network becomes unable to generalize itself to new data (such as the 101 testing data points), causing its accuracy on the test data to decline.

**4.2 Limitations of the Investigation**

After the investigation had been completed, it became apparent that there were several major limitations of the experiment that need to be considered.

The first major limitation has to do with the size of the dataset. Although most major neural network programs are trained on big data, this investigation only used a neural network that was trained on 405 data points. In comparison to the terabytes of data that modern machine learning algorithms are based on, this training data size is very small. The selected dataset was this size, however, so that I could run it on my computer. As discussed earlier, a major obstacle to machine learning is the necessary storage space and processing power to handle big data. Regardless, the size of the used dataset is a limitation in this experiment, making the results less applicable to neural networks based on big data.

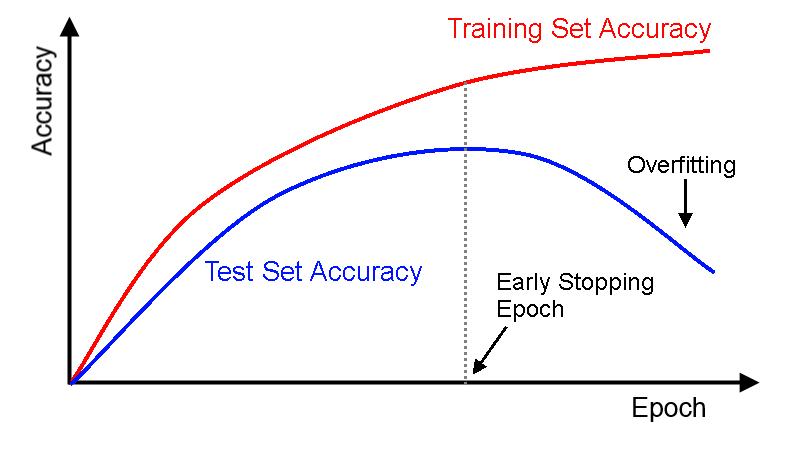
Another limitation faced during the investigation was whether the neural network’s results on a specific dataset could be reliably applied to other datasets. The chosen dataset— the Boston Housing dataset— only illustrates the behavior of the neural network for one set of inputs and outputs. The behavior of neural networks may vary based on both the complexity of the dataset and the degree of noise within the dataset. Again, this limits the extent to which the results of this experiment can extrapolated to neural networks in general.

Similarly, the experiment is inherently limited by the chosen experimental constants. There are multiple types of backpropagation algorithms and activation functions, so the results from this specific neural network may not be representative of all neural networks. To minimize this impact, fundamental neural network algorithms were used. Nevertheless, these choices still had some impact on the results.

**5 Methods to Counter Overfitting**

The experimental data made it clear that, although increasing the number of neurons can improve accuracy, it can also hinder it by overfitting the data. Unfortunately, the ideal number of neurons can currently only be found with trial and error, as the programmer must hard code in the structure of the neural network. Doing so, however, requires extensive time to repeatedly train and test the neural network— a task not always feasible in the real world. For this reason, methods to overcome overfitting in neural networks were explored. Two major solutions are dropout and early stopping.

Early stopping proposes that the neural network learns from the training data until the accuracy begins to decrease on the test data. At the end of each training iteration, the neural network is run on the test data. If the accuracy of the current iteration is lower than the accuracy of the last iteration, the neural network stops training. Otherwise, the neural network back propagates on the training data and continues learning.[[26]](#footnote-26) This method allows the neural network to continue learning until overfitting is seen. The point at which early stopping terminates training is shown in the figure below.



**Figure 2: A graph for early stopping. “Epoch” is another term for training iteration.[[27]](#footnote-27)**

An alternate way of combatting overfitting in neural networks is dropout: a technique where randomly selected neurons are ignored during training. Dropout works by giving each neuron a probability of being “ignored” during backpropagation or being “dropped out.”[[28]](#footnote-28) In doing so, dropout constrains a neural network from becoming dependent on the outputs of specific neurons. Neurons are prevented from being too reliant on each other and, instead, independently become strong processors. This neuron independence leads to better generalization on new data, and hence, less overfitting.

**6 Conclusion**

Having completed the experiment, the research question “*To what extent does the number of neurons affect the accuracy of an artificial neural network?*” was able to be considered in depth. The number of neurons in a python neural network was tested against the program’s accuracy on testing data, producing experimental data. The results of the experiment suggest a nonlinear relationship between the number of neurons and neural network accuracy, with accuracy improving as the number of neurons increased until a certain point before declining. From this, it can be determined that increasing the number of neurons can lead to improved accuracy, but too many neurons can hinder the neural network’s performance on real-world, noisy data.

The results of this experiment led to the investigation of how too many neurons can cause a neural network’s accuracy to significantly decline, as seen in the experimental results. After conducting further research, it was concluded that the neural network would overfit the training data if there were too many neurons. This means that the model fit itself to the noise in the training data rather than the signal of the dataset, making the neural network unable to generalize itself to new data.

Although the experimental results did not match the hypothesis, they were found to be logical after further research. The neural network exhibited both underfitting and overfitting in the data as reflected in the experimental data. To optimize a neural network, it follows logically that the neural network should follow a structure between the underfitting model complexity and the overfitting model complexity. Conversely, a neural network can have an excessive number of neurons but implement techniques to counter overfitting to create a robust model. Overall, it can be concluded that neural networks are a powerful data processing tool, being able to analyze lrge and complex datasets. Such potent processing capabilities explain why neural networks currently play, and will continue to play, a dominant role in the field of machine learning.

**Annotated Bibliography**

Jacobson, Ralph. "2.5 quintillion bytes of data created every day. How does CPG & Retail manage it?" IBM Consumer Products Industry Blog. April 24, 2013. Accessed March 05, 2017. https://www.ibm.com/blogs/insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/.

Jacobson reports on the significance of “big data” in the modern world and describes the four main aspects of it: Volume, Velocity, Variety, and Veracity. He continues to describe specific applications of big data within IBM, such as data analysis of customer data. Jacobson states some helpful statistics about big data, allowing the reader to grasp its growing importance. Such information aids in the introduction to this extended essay, where big data is defined and described with relevance to machine learning.

Mahanta, Jahnavi. "Introduction to Neural Networks, Advantages and Applications." Towards Data Science. July 10, 2017. Accessed October 20, 2017. https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207?gi=eb0e7aceeeb5.

Mahanta begins her description of neural networks by first describing the artificial neural networks in our brain. Our biological neurons receive inputs, process information, and send outputs to other inputs much like the neural network programs discussed in this essay. She then uses this analogy to describe the workings of neural networks in depth. Mahanta also states the advantages of neural networks in the field of machine learning, highlighting their importance. This general information on neural networks is used to introduce neural networks.

Malkevitch, Joseph. "The Digital Revolution- Barcodes: Introduction." AMS. Accessed February 10, 2017. http://www.ams.org/publicoutreach/feature-column/fcarc-barcodes1.

Malkevitch describes the rise of the digital revolution in the 20th century, focusing on the flowering of new information technologies. This article specifically focuses on the 20th century developments and how they have led to modern-day technology, such as computers and high definition television (HDTV). The article goes on to list the great variety of applications of barcodes— one result of the revolution.

Nicholson, Chris V., and Adam Gibson. "Early Stopping." Early Stopping - Deeplearning4j: Open-source, Distributed Deep Learning for the JVM. Accessed April 20, 2017. https://deeplearning4j.org/earlystopping.

In this webpage, Nicholson outlines the early stopping algorithm and how it can minimize overfitting in neural networks. At the end of each training iteration, Nicholson states, the neural network is tested on the test data to check for overfitting. This source provides a detailed, concise explanation, highlighting its general workings rather than the code implementation. This brief yet thorough description is used in the paper to give an overview of the algorithm.

Nielsen, Michael A. "Neural Networks and Deep Learning." Neural networks and deep learning. December 2017. Accessed January 13, 2018. http://neuralnetworksanddeeplearning.com/chap1.html.

In this webpage, Nielsen goes through an example of how to use a neural network to classify handwritten digits. In the process, Nielsen defines basic elements of the neural network, such as weights and biases. Using these definitions, he describes the neuron processing on inputs, providing mathematical equations for functions. Such information is paramount to the evaluation of neurons in this essay. The level of detail and the helpful examples make this a primary source to the essay.

Ng, Andrew. "Machine Learning." Coursera. Accessed January 12, 2017. https://www.coursera.org/learn/machine-learning.

This online machine learning video course thoroughly educates the view about machine learning and a multitude of its different algorithms. The course starts off with a broad overview of machine learning, describing the main uses of machine learning as well as its pervasive nature in modern society. Such descriptions, as well as Ng’s extensive explanation of neural network algorithms, are used in both the introduction and the neural network overview in this essay.

Richmond, Alan. *A Simple Neural Network*. January 31, 2018. Digital image. http://tuxar.uk/brief-introduction-artificial-neural-networks/

On Richmond’s webpage, a simple and clear diagram of neural network structure. This is a digital image accessed through the internet. This picture is used to help communicate the structure of neural networks and illustrate descriptions of neural network asspects.

Rijmenam, Mark Van. "A Short History Of Big Data." A Short History Of Big Data. January 06, 2018. Accessed February 14, 2018. https://datafloq.com/read/big-data-history/239.

In this article, Rijmenam gives a brief overview of the history of data processing. He describes the origin of data analysis with John Graunt in 1663 with mortality rates, then continues chronologically up until the 21st century and the modern era. Rijmenam then emphasizes the paramount role of big data now and its growing significance. Many dates and statistics are listed throughout the article, aiding the reader’s understanding. Some such statistics are used in this essay.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, and Yoshua Bengio. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." Journal of Machine Learning Research 15 (June 2014): 1-30. http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf.

In his paper, Srivastava evaluated the dropout algorithm in neural networks and how it can be used to minimize overfitting. Srivastava interprets the effectiveness of the algorithm through multiple tests on real world datasets, stating the accuracy of different levels of dropout. His comprehensive overview of dropout makes this paper a strong source to describe dropout. His information on dropout is used to describe methods to reduce overfitting in this essay.

Weber, Michael. "Machine learning for SEO - How to predict rankings with machine learning." SearchVIU Tools for advanced SEO. October 31, 2017. Accessed November 15, 2017. https://www.searchviu.com/en/machine-learning-seo-predicting-rankings/.

Weber focuses this article on search engine optimization— or, as he refers to it in this article, SEO. He lays out the problems of identifying words and ranking results in internet search engines and how to approach such problems via code. Weber specifically evaluates how machine learning programs can be used to rank search results, using large batches of data to predict user preferences. This discussion of machine learning applications to search engines is helpful when providing the reader with applications of machine learning,

"A Basic Introduction To Neural Networks." Computer Sciences Department. Accessed March 12, 2017. http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html.

This webpage describes the basic structure of a neural network and its uses in computing. A neural network is first explicitly defined, then the specific aspects of neural networks are explored. The webpage first describes the mathematics of neurons, then goes on to describe the process of backpropagation using calculus. Specifically, backpropagation is described with the delta rule— a calculus formula for the gradient descent optimization algorithm. The limitations of neural networks are also evaluated. Details on the limitations as a well as information about the neural network structure is incorporated into this essay.

"Boston Housing Data." Index of /ml/machine-learning-databases/housing. https://archive.ics.uci.edu/ml/machine-learning-databases/housing/.

This is the dataset that is used in the python neural network. It is a dataset of 13 inputs variables, 1 output variable, and 506 data points. The relatively large amount of data allows the neural network to be trained on a home computer and in a reasonable time.

"Machine Learning: What it is and why it matters." SAS| The Power to Know. Accessed April 05, 2017. https://www.sas.com/en\_us/insights/analytics/machine-learning.html.

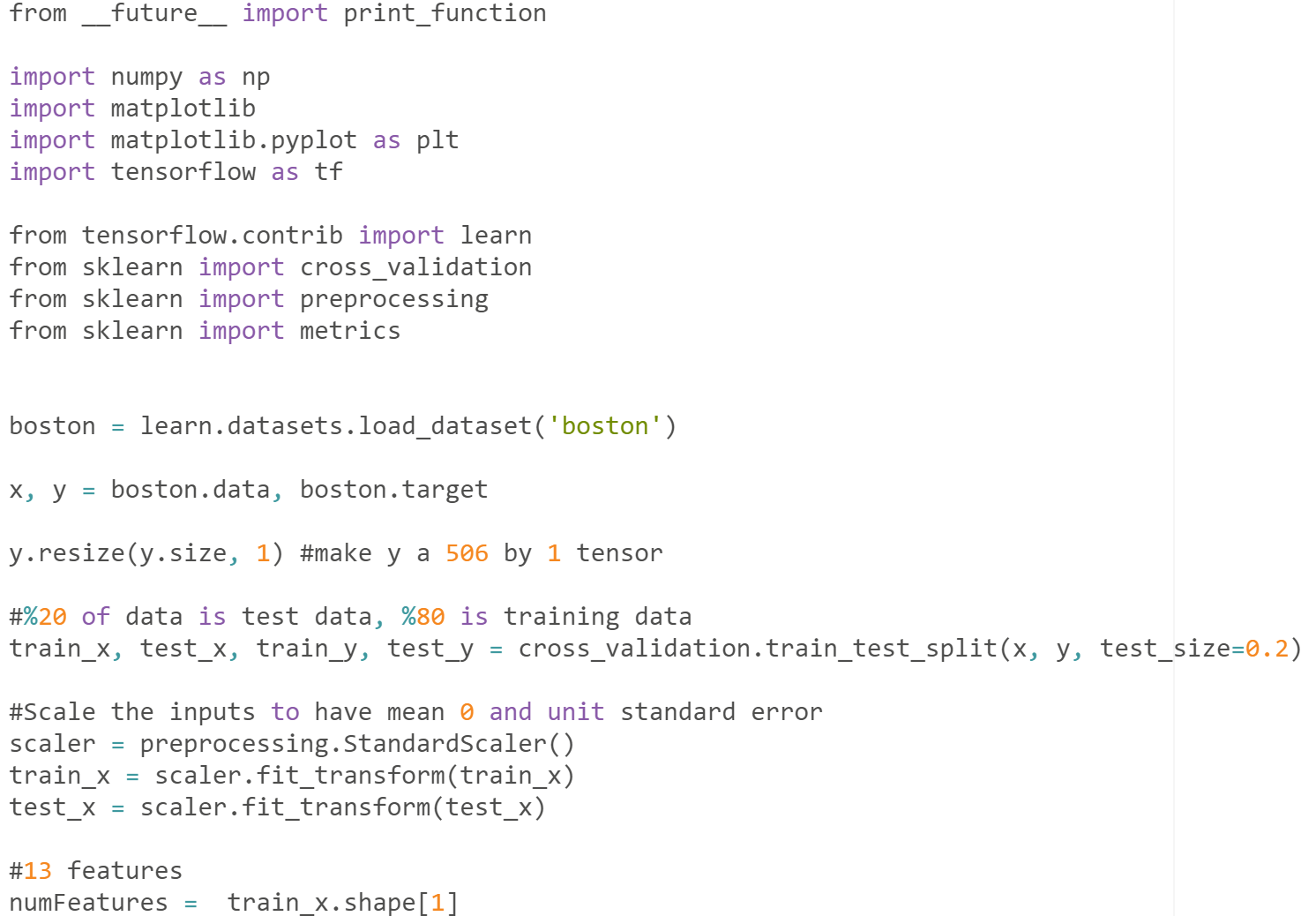
On this webpage from the SAS developer website, a short description of machine learning and its growing importance as a data analysis tool is given. The page defines machine learning, lists its numerous applications, and communicates its significance in data mining and data analysis today. These concise, general descriptions of machine learning are used in the introduction section of this essay.

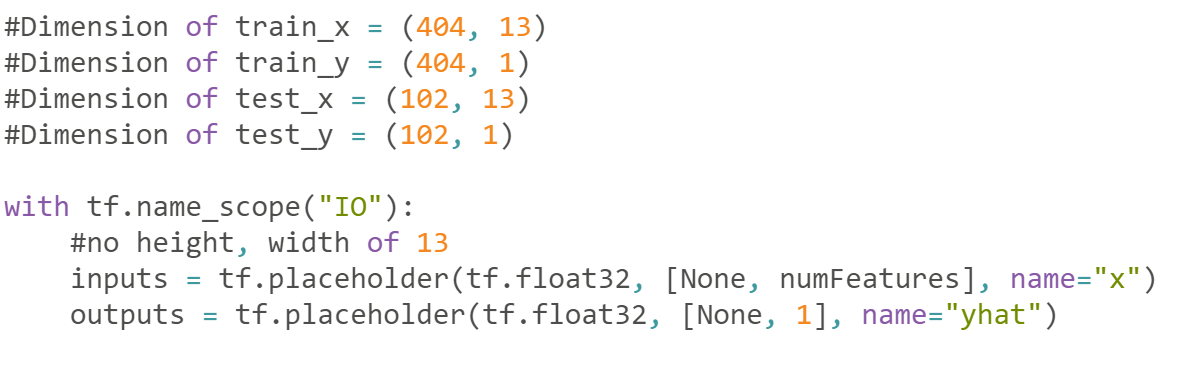
"Overfitting in Machine Learning: What It Is and How to Prevent It." EliteDataScience. February 08, 2018. https://elitedatascience.com/overfitting-in-machine-learning#signal-vs-noise.

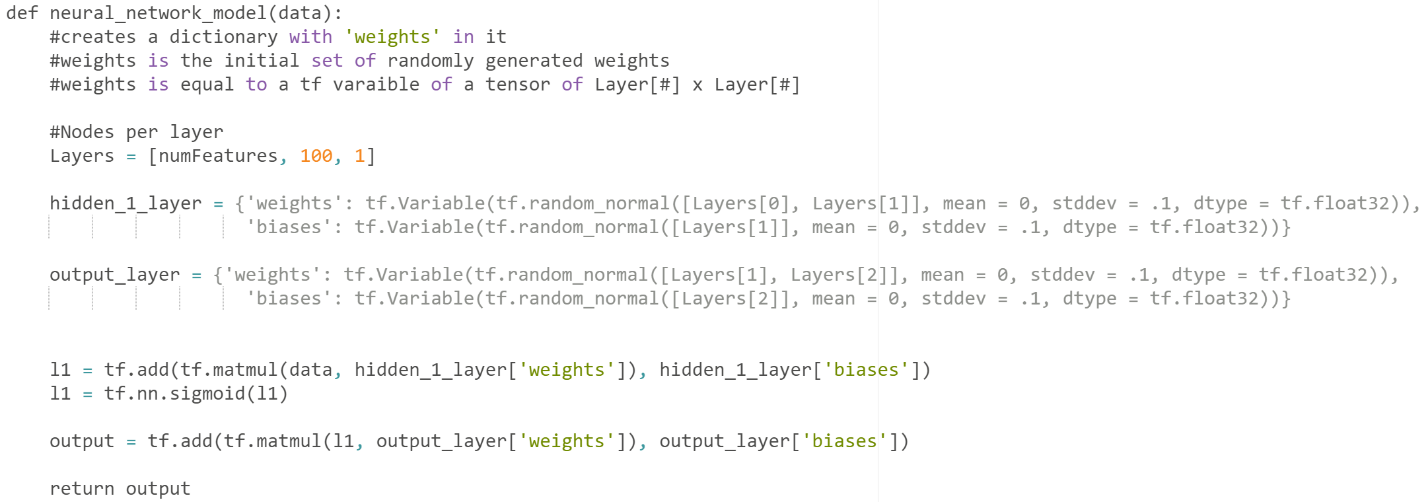
This webpage provides a thorough explanation of signal and noise in data processing and how it can lead to overfitting in neural networks. Noise, according to the article, is the inherent randomness in a data point that creates error from the signal. The descriptions and diagrams in this article illustrate how neural networks grow to overfit the training data if it is too complex. The information is used to interpret the unexpected overfitting seen in the experimental data.

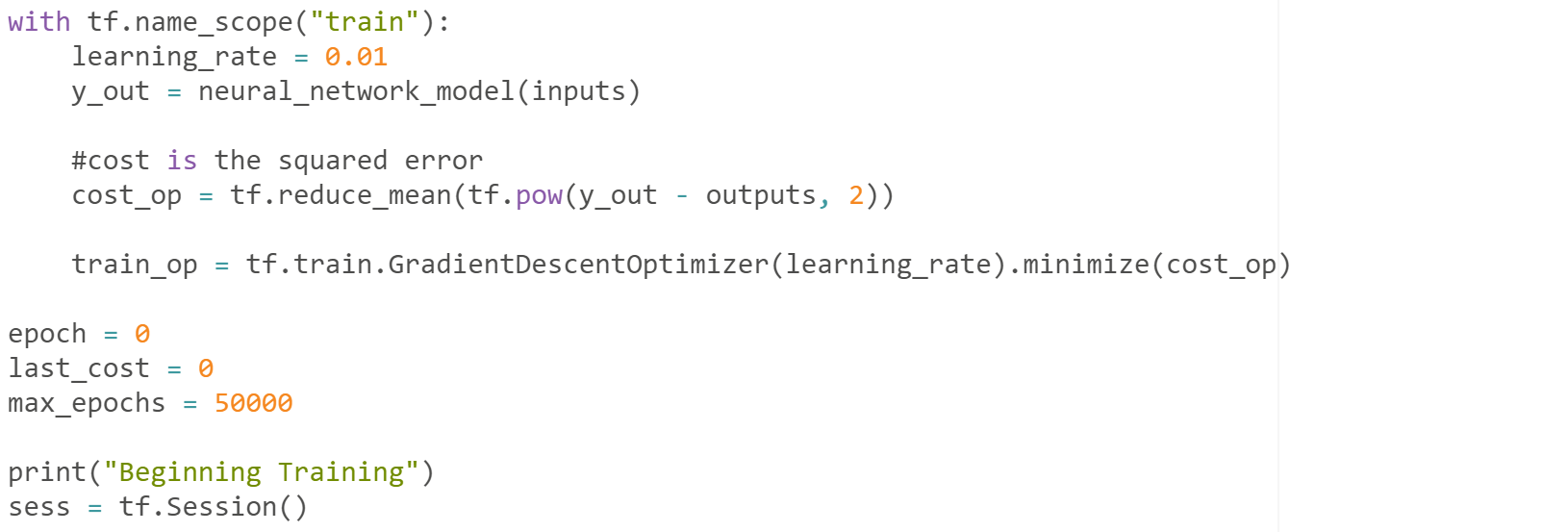
**Appendix**

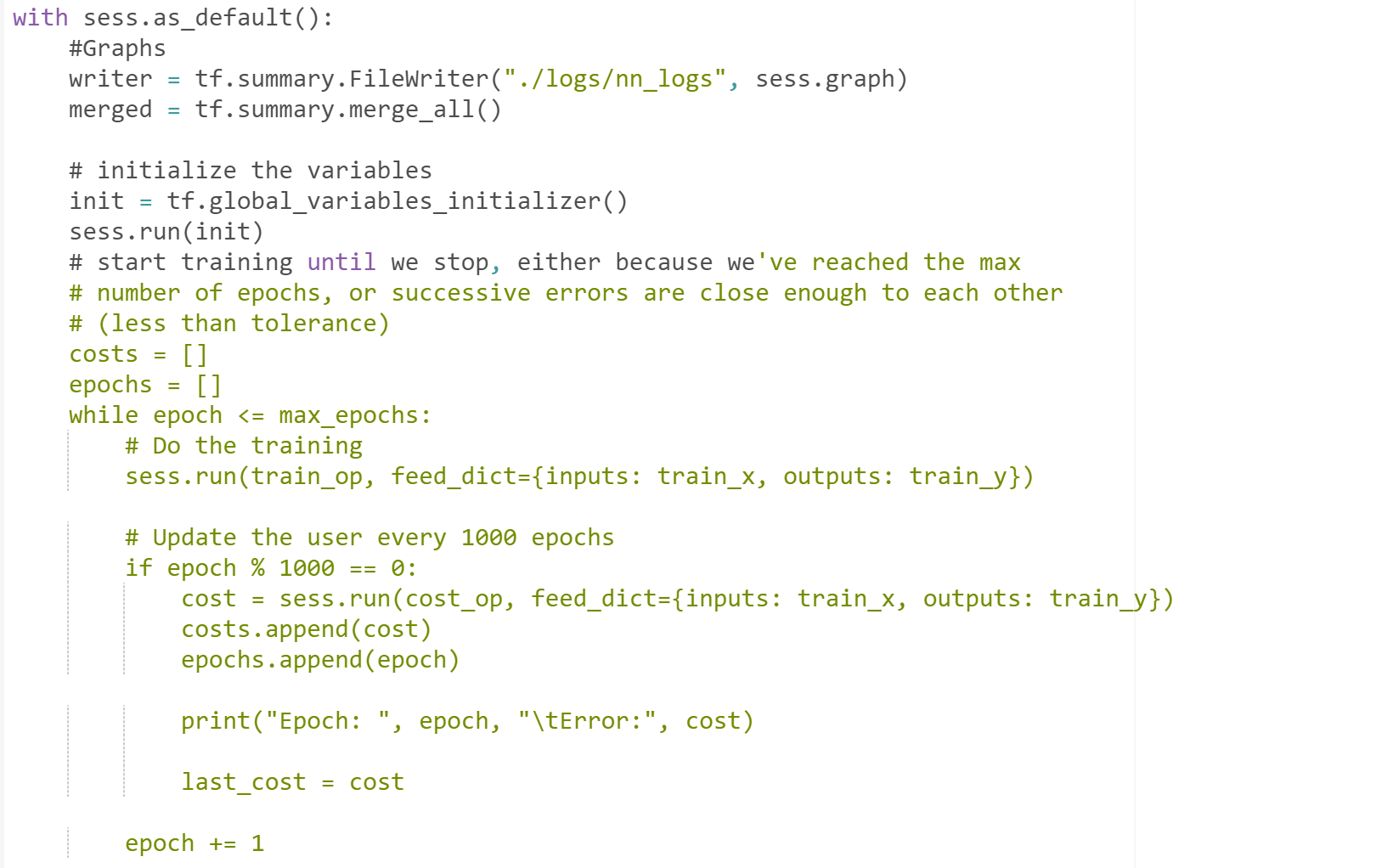
Program Code

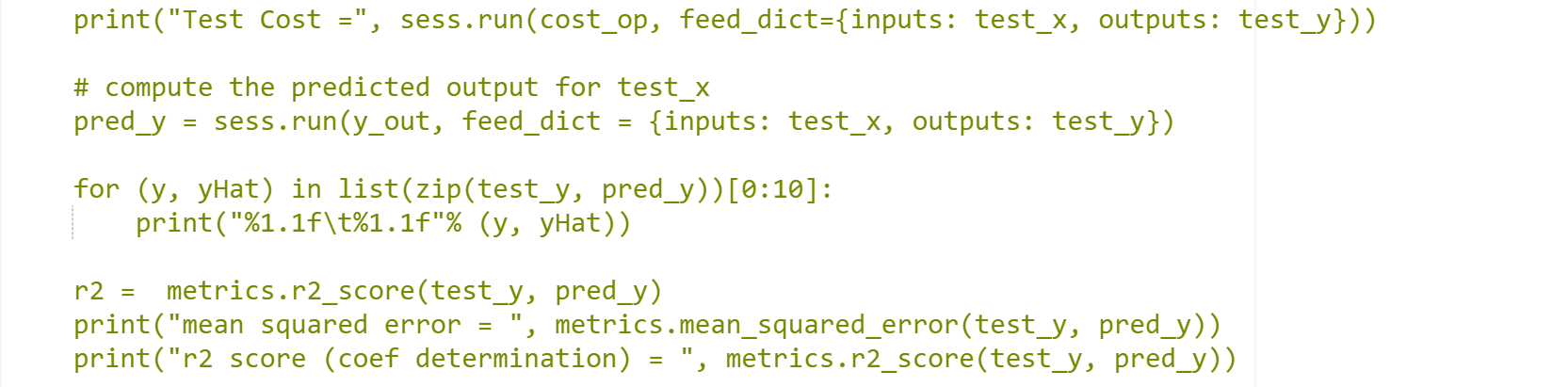












1. Malkevitch, Joseph. "The Digital Revolution- Barcodes: Introduction." AMS. Accessed February 10, 2017. http://www.ams.org/publicoutreach/feature-column/fcarc-barcodes1. [↑](#footnote-ref-1)
2. Jacobson, Ralph. "2.5 quintillion bytes of data created every day. How does CPG & Retail manage it?" IBM Consumer Products Industry Blog. April 24, 2013. Accessed March 05, 2017. https://www.ibm.com/blogs/insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/. [↑](#footnote-ref-2)
3. Rijmenam, Mark Van. "A Short History Of Big Data." A Short History Of Big Data. January 06, 2018. Accessed February 14, 2018. https://datafloq.com/read/big-data-history/239. [↑](#footnote-ref-3)
4. "Machine Learning: What it is and why it matters." SAS | The Power to Know. Accessed April 05, 2017. https://www.sas.com/en\_us/insights/analytics/machine-learning.html. [↑](#footnote-ref-4)
5. Weber, Michael. "Machine learning for SEO - How to predict rankings with machine learning." SearchVIU Tools for advanced SEO. October 31, 2017. Accessed November 15, 2017. https://www.searchviu.com/en/machine-learning-seo-predicting-rankings/. [↑](#footnote-ref-5)
6. Ng, Andrew. "Machine Learning." Coursera. Accessed January 12, 2017. https://www.coursera.org/learn/machine-learning. [↑](#footnote-ref-6)
7. Ng, Andrew. "Machine Learning." Coursera. Accessed January 12, 2017. https://www.coursera.org/learn/machine-learning. [↑](#footnote-ref-7)
8. "A Basic Introduction To Neural Networks." Computer Sciences Department. Accessed March 12, 2017. http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html. [↑](#footnote-ref-8)
9. Mahanta, Jahnavi. "Introduction to Neural Networks, Advantages and Applications." Towards Data Science. July 10, 2017. Accessed October 20, 2017. https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207?gi=eb0e7aceeeb5. [↑](#footnote-ref-9)
10. Mahanta, Jahnavi. "Introduction to Neural Networks, Advantages and Applications." Towards Data Science. July 10, 2017. Accessed October 20, 2017. https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207?gi=eb0e7aceeeb5. [↑](#footnote-ref-10)
11. Alan Richmond, *A Simple Neural Network*. January 31, 2018. Digital image. http://tuxar.uk/brief-introduction-artificial-neural-networks/ [↑](#footnote-ref-11)
12. "A Basic Introduction To Neural Networks." Computer Sciences Department. Accessed March 12, 2017. http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html. [↑](#footnote-ref-12)
13. "A Basic Introduction To Neural Networks." Computer Sciences Department. Accessed March 12, 2017. http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html. [↑](#footnote-ref-13)
14. "A Basic Introduction To Neural Networks." Computer Sciences Department. Accessed March 12, 2017. http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html. [↑](#footnote-ref-14)
15. Mahanta, Jahnavi. "Introduction to Neural Networks, Advantages and Applications." Towards Data Science. July 10, 2017. Accessed October 20, 2017. https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207?gi=eb0e7aceeeb5. [↑](#footnote-ref-15)
16. Mahanta, Jahnavi. "Introduction to Neural Networks, Advantages and Applications." Towards Data Science. July 10, 2017. Accessed October 20, 2017. https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207?gi=eb0e7aceeeb5. [↑](#footnote-ref-16)
17. Nielsen, Michael A. "Neural Networks and Deep Learning." Neural networks and deep learning. December 2017. Accessed January 13, 2018. http://neuralnetworksanddeeplearning.com/chap1.html. [↑](#footnote-ref-17)
18. "A Basic Introduction To Neural Networks." Computer Sciences Department. Accessed March 12, 2017. http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html. [↑](#footnote-ref-18)
19. "Boston Housing Data." Index of /ml/machine-learning-databases/housing. https://archive.ics.uci.edu/ml/machine-learning-databases/housing/. [↑](#footnote-ref-19)
20. Nielsen, Michael A. "Neural Networks and Deep Learning." Neural networks and deep learning. December 2017. Accessed January 13, 2018. http://neuralnetworksanddeeplearning.com/chap1.html. [↑](#footnote-ref-20)
21. Nielsen, Michael A. "Neural Networks and Deep Learning." Neural networks and deep learning. December 2017. Accessed January 13, 2018. http://neuralnetworksanddeeplearning.com/chap1.html. [↑](#footnote-ref-21)
22. Ng, Andrew. "Machine Learning." Coursera. Accessed January 12, 2017. https://www.coursera.org/learn/machine-learning. [↑](#footnote-ref-22)
23. "Overfitting in Machine Learning: What It Is and How to Prevent It." EliteDataScience. February 08, 2018. https://elitedatascience.com/overfitting-in-machine-learning#signal-vs-noise. [↑](#footnote-ref-23)
24. "Overfitting in Machine Learning: What It Is and How to Prevent It." EliteDataScience. February 08, 2018. https://elitedatascience.com/overfitting-in-machine-learning#signal-vs-noise. [↑](#footnote-ref-24)
25. "Overfitting in Machine Learning: What It Is and How to Prevent It." EliteDataScience. February 08, 2018. https://elitedatascience.com/overfitting-in-machine-learning#signal-vs-noise. [↑](#footnote-ref-25)
26. Nicholson, Chris V., and Adam Gibson. "Early Stopping." Early Stopping - Deeplearning4j: Open-source, Distributed Deep Learning for the JVM. Accessed April 20, 2017. https://deeplearning4j.org/earlystopping. [↑](#footnote-ref-26)
27. Nicholson, Chris V., and Adam Gibson. "Early Stopping." Early Stopping - Deeplearning4j: Open-source, Distributed Deep Learning for the JVM. Accessed April 20, 2017. https://deeplearning4j.org/earlystopping. [↑](#footnote-ref-27)
28. Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, and Yoshua Bengio. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." Journal of Machine Learning Research 15 (June 2014): 1-30. http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf. [↑](#footnote-ref-28)