Draft version April 12, 2022
Typeset using LaTeX default style in AASTeX631

Artificial Neural Networks

Kevin B. Moposita[1]

[1] *Villanova University*
800 Lancaster Avenue
Villanova, PA 19085, USA

1. INTRODUCTION

When talking about data processing, nothing is better than the human brain. Barring any abnormalities/disabilities, the human brain is one of the most complex organs within the human body and has been evolved such that it is able to receive, process, and relay information in little to no time. For example, when looking at a picture of a fruit, say a kiwi, there is an immediate recognition of what is being observed. In some cases, not only will a person be able to recall what the fruit is, but also have other senses jump in as well. Besides sight, some might go as far as being able to recall how a kiwi tastes or how it smells. It is because of this ability that the brain is supreme in data processing.

While the brain might seem to be able to do it all, it is only able to do this because of the information that is has stored as a result of constant exposure. A person who is able to recognize a kiwi is only able to do so because they have seen one before and can formulate how it tastes due to eating one before. What if a person has never seen a kiwi before? If this tragic scenario were to occur (being deprived of the best fruit on Earth), then when this person were to see a kiwi fr the first time, there would be no recognition at all. Not only would there be no recognition, there would also be no involvement of the other senses. How could there be if the person has never seen a kiwi before? It is only through constant exposure that the brain will learn what something is and potentially stimulate the other senses. This information will get stored and accessed the next time a kiwi is encountered.

Similar to this idea, a computer is able to learn through constant exposure. Though not at the level of the human brain, a computer is capable of storing information about an object and accessing it whenever needed and this is a result of Artificial Neural Networks (A.N.N). With ANNs, a computer is capable of recreating similar functions that a brain does to produce a desired output, the process which is called non-linear mapping. Non-linear mapping involves a computer receiving an input, processing such input, and making a desired determination. Within the processing step, there is a "hidden layer" in which multiple units are found. The connections between the units of the "hidden layers" are called weights.

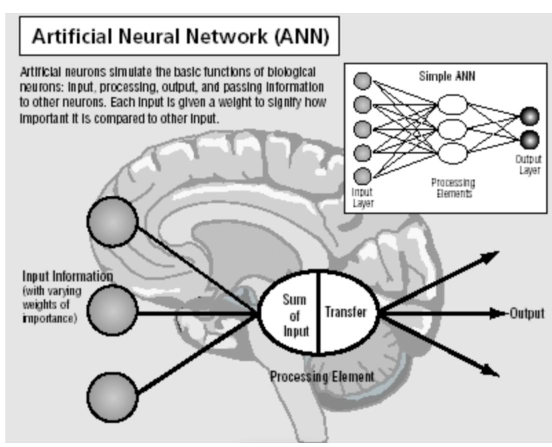

Figure 1. A comparison of machine learning to how the human brain processes information.

2. TRAINING SET

For this section, we will create and train an ANN to be able to receive a hand-drawn number as an input and tell us what number it is. Utilizing a training set from *MNIST*, 60000 images of numbers 0 to 9 were used to 'train' the ANN to be able learn (process and store information) and correctly differentiate the numbers between this range. *Figures 2 and 3* are examples of a random number from this training set. From the same site, a testing data set of 10000 images were then used as inputs.

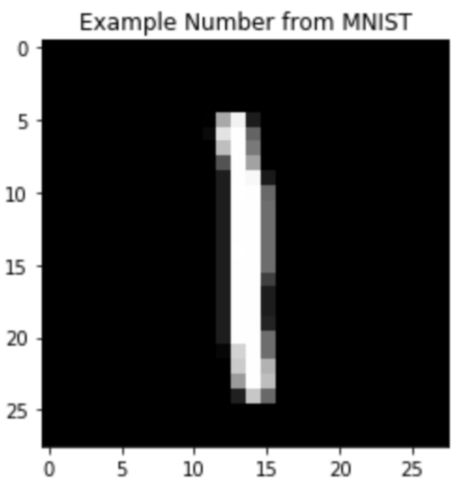

Figure 2. An example number from the training set. This is number 1.

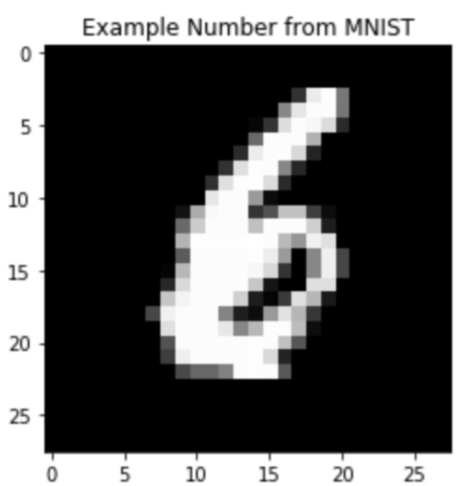

Figure 3. Another example of a random number. This is number 6.

To ensure that the training data sets were indeed randomized, two random sets of numbers were extracted. *Table 1* displays the two extracted data sets and as expected, they are indeed random and therefore excellent for training the ANN.

Table 1. Numbers Extracted from Data Set

First Training Set	Second Training Set
0	8
4	4
1	0
6	3
3	1
3	7
2	5
9	5

To further learn more about the training set, *Figure 4* displays the frequency of each number to ensure that they are being displayed at approximately the same frequency.

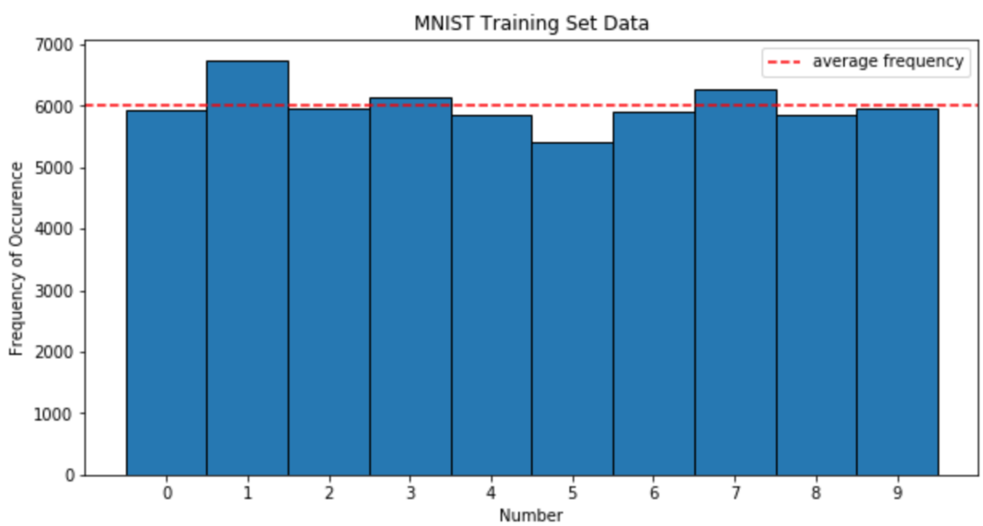

Figure 4. Number Frequency Within the Data Set

As *Figure 4* demonstrates, the frequency in which numbers 0 to 9 appear within the data set is about equal and critical for the ANN to be able to store expansive information on all numbers within the range. With this in mind, the training of my beautiful ANN can begin.

3. 3-LAYER NETWORK

Within this section, the training data set that was previously mentioned is utilized to train my ANN to become the ANN I know she can become. With 3-layers and approximately 20-30 hidden units, the training begun. To see how the training fared, a plot (*Figure 5* depicting the ANNs learning curve and recognition convergence for the training set was created.

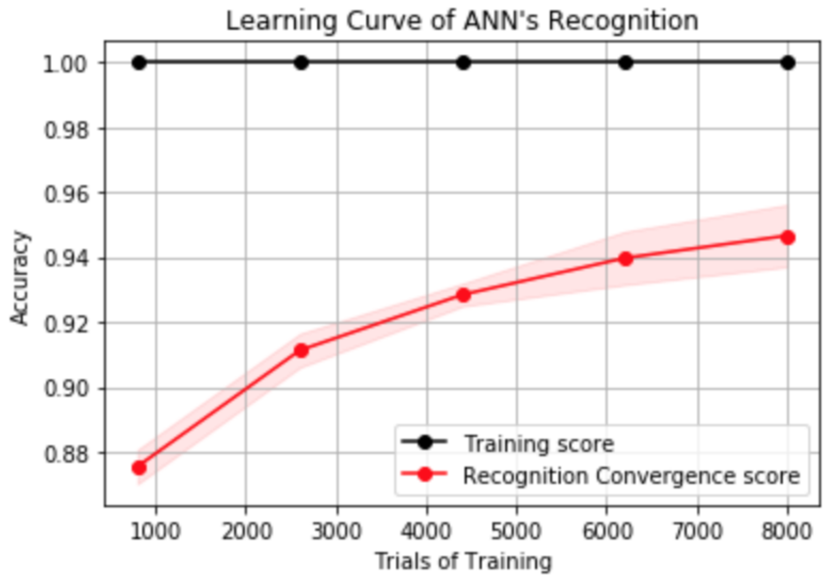

Figure 5. Line plots of the training score compared to the recognition convergence plot.

As *Figure 5* demonstrates, the training score of the ANN remained constant to 1, which is tempting to label it as perfect. The ANNs recognition convergence score steadily increased as the number of training trials increased, which was expected. The recognition convergence was measured to be approximately 97%. The faint shaded regions around this plot depict the error bars, which is accurate to 1-σ.

4. ANN VS THE WORLD

Seeing how the ANN was able to perform relatively well with the data set from *MNIST*, it would be interesting to see how it would fare against a different type of data-set. To do so, the author has contacted individuals with the most interesting handwriting (ranging from neat to horrid) to have their numbers utilized as inputs. Due to *HIPAA* compliance, these individuals will remain anonymous. Similar to the data set, numbers ranging from 0 to 9 were utilized once again. Before the test, there are numbers that the ANN is expected to be confused with. It is expected that the ANN confuse (1,7), (2,5,6), and (3,8). *Figure 6* tracks the loss of the ANN, through training loss and recognition loss.

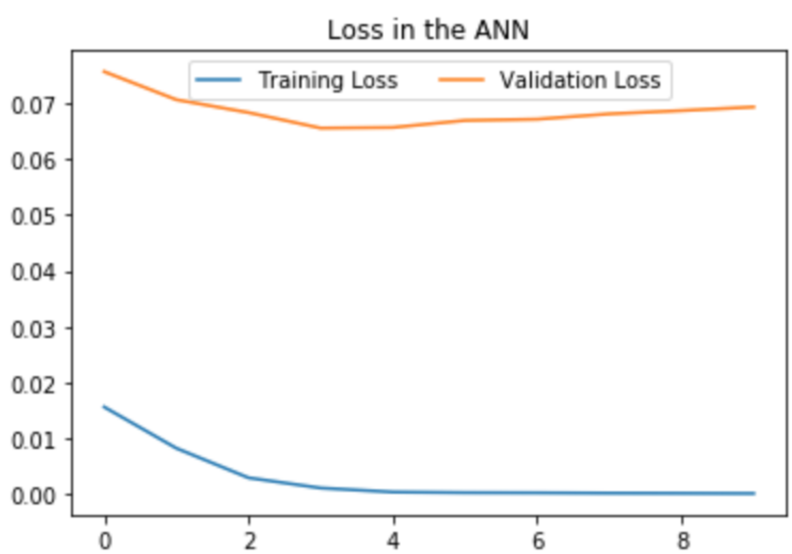

Figure 6. Tracking the training and recognition loss of the ANN.

As observed in *Figure 6* the training loss was much lower than the recognition loss. Despite the recognition loss being higher than the training loss, the plot converged at approximately 0.07, which is much better than expected. With more epochs, the plot would be much better.

Along with the tracking loss, *Figure 7* was created to track accuracy within the ANN, both from training and recognition.

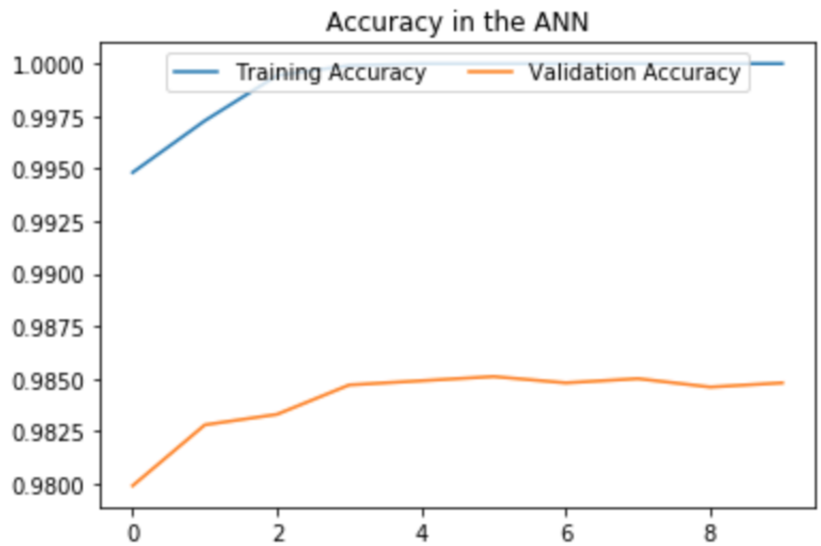

Figure 7. Tracking accuracy within my beautiful ANN.

As *Figure 7* demonstrates, the training accuracy remains supreme as it converges to 1, while the recognition accuracy converges to 0.9850. The training accuracy and recognition accuracy both appeared to converge at approximately the same time, though the training accuracy is just a tad bit steeper. Just as with the previous plot, the recognition accuracy being this high was also better than expected. An increase in the epoch number would make the plot smoother.

Table 2 displays the accurate guesses the ANN depending on the number. The high accurate guesses falls in line with the previous plots. The numbers 3 and 8 were the least accurate guesses (not by much) and confirms the expectations that were previously mentioned.

Table 2. Numbers and Their Correct Guesses

Numbers	**Correct Guesses**
0	8
1	9
2	8
3	7
4	8
5	7
6	9
7	8
8	7
9	9

5. NETWORK TOPOGRAPHY

Depending on different factors, it can affect how efficient the ANN operates. Within this section, the epochs, number of units, and the batch size were all modified to see how it would affect the ANNs efficiency.

For *Figures 8 and 9*, *Table 3* displays the parameters that were utilized. With these in hand, the ANNs loss and accuracy was tracked.

Table 3. Parameters for ANN

Training Set Images	**Epoch**	**Batch Size**
60000	10	100

As can be seen in *Figure 8*, the behavior of the plots is more linear than previous plots due to the low number of epochs introduced. The same reasoning can be said for *Figure 9*.

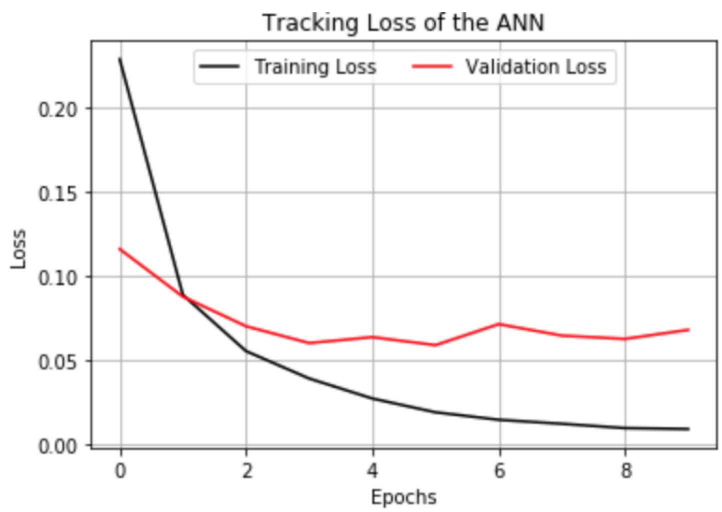

Figure 8. Tracking loss with the above parameters.

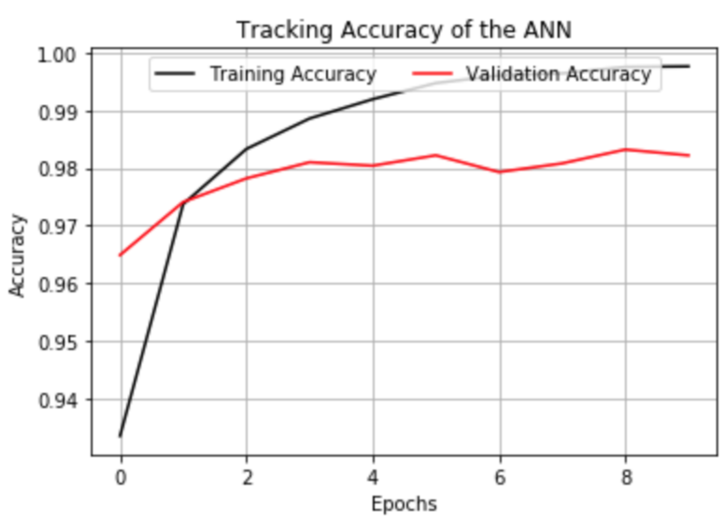

Figure 9. Tracking accuracy with the above parameters.

To verify this, the number of epochs was changed to 40, reflected through *Table 4*.

Table 4. New ANN Parameters

Training Set Images	Epoch	Batch Size
60000	40	100

Figured 10 and 11 were the resulting plots. The increase in epoch count made the plots smoother.

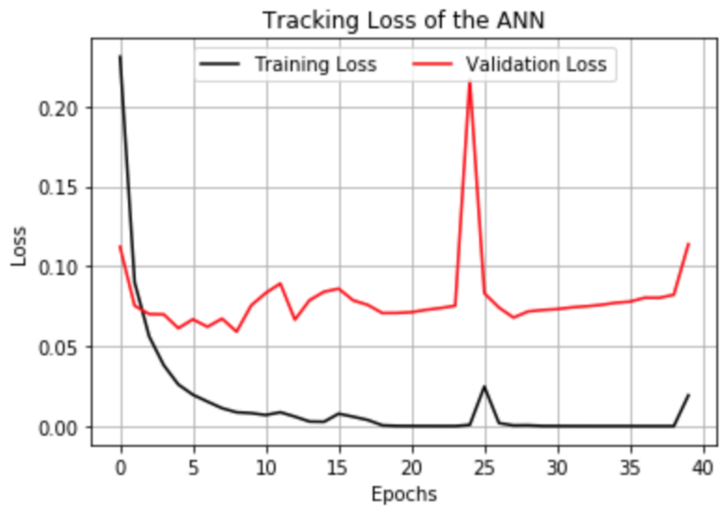

Figure 10. Resulting Plots

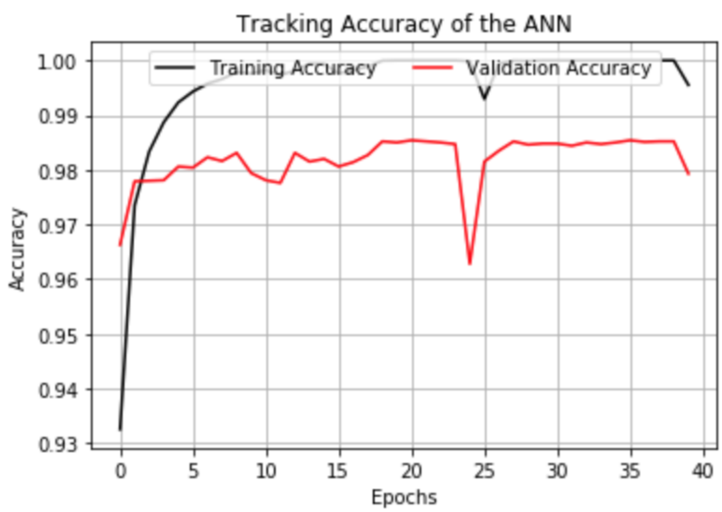

Figure 11. Resulting Plots

6. CONCLUSION

For this assignment, we created and worked with Artificial Neural Networks (ANNs) to train them through constant exposure to be able to correctly recognize the numbers that were used as the input. As this report shows, the training that the ANN underwent proved to be useful as the loss and accuracy in the plots demonstrate.