**Artificial Neural Network**

**Base Case**

Confusion Matrix:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Predicted Labels | | | | | | | | | | | |
| Real Labels |  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 163 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 131 | 51 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 159 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 168 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 164 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **5** | 142 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **6** | 166 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **7** | 172 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **8** | 150 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9** | 161 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Total Accuracy:** 0.131530424093

**Experimentation**

After experimentation with numerous different aspects of the ANN, changing the amount of layers, neurons per layer, number of epochs, and the batch size, I found a combination of characteristics that successfully matched the majority of the provided images with their respective labels. In the end, I had developed an ANN with the following characteristics:

* **5 Total Layers**
* **Layer 1**: 500 Neurons
* **Layer 2:** 300 Neurons
* **Layer 3:** 100 Neurons
* **Layer 4:** 50 Neurons
* **Layer 5:** 10 Neurons
* **Number of Epochs:** 200
* **Batch Size:** 1024

**Process:**

In order to get these results, I started with the original Template that had been provided in the assignment, which only had 2 Layers and 10 epochs.

The first experiments involved increasing the amount of epochs that the training set would go through, so I had increased it from 10 to 100, then 1,000, then as high as 10,000. Despite these increases there was only a marginal increase in performance, while still suffering from wildly varying results, going as high as 80 percent accuracy and then dropping to as low as 40 percent. Since I was not seeing any definitive improvements, I decided that having such large epoch sizes was unnecessary, and reduced the size down to 200, providing me with an acceptable edge in performance without hindering runtime exponentially.

The next part of the experiment was to increase the number of layers, as well as the number of neurons that were in each layer. I started by adding another layer with 100 neurons to the algorithm, and noticed a slight increase in performance. I then added another layer of 300 neurons, and noticing a larger increase, put in another layer of 500, as well as a layer of fifty neurons in between the 10 and 100 layers. All together, this added a large amount of improvement to the ANN, bringing up accuracy to around 80 percent, and only fluctuating mildly for most cases. The results from my experimentation are shown below. Note: It is important to note that there are some cases that fail badly, going as low as 10 Percent. However, I felt that these cases were too few to consider true measures of the ANN’s success.

**Results:**

Confusion Matrix:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Predicted Labels | | | | | | | | | | | |
| Real Labels |  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 163 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 19 | 163 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 35 | 0 | 117 | 0 | 1 | 0 | 1 | 2 | 3 | 0 |
| **3** | 32 | 0 | 0 | 133 | 0 | 0 | 0 | 2 | 0 | 1 |
| **4** | 39 | 1 | 0 | 0 | 120 | 0 | 1 | 1 | 0 | 2 |
| **5** | 40 | 0 | 0 | 0 | 0 | 98 | 1 | 1 | 1 | 1 |
| **6** | 18 | 0 | 0 | 1 | 1 | 1 | 144 | 1 | 0 | 0 |
| **7** | 25 | 0 | 0 | 0 | 2 | 0 | 0 | 143 | 0 | 2 |
| **8** | 46 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 101 | 0 |
| **9** | 37 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 2 | 118 |

**Total Accuracy**: 0.79901659496

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Label** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **Precision** | 0.36 | 0.99 | 1.00 | 0.98 | 0.96 | 0.98 | 0.97 | 0.94 | 0.94 | 0.95 |
| **Recall** | 1.00 | 0.90 | 0.74 | 0.79 | 0.73 | 0.69 | 0.87 | 0.83 | 0.67 | 0.73 |

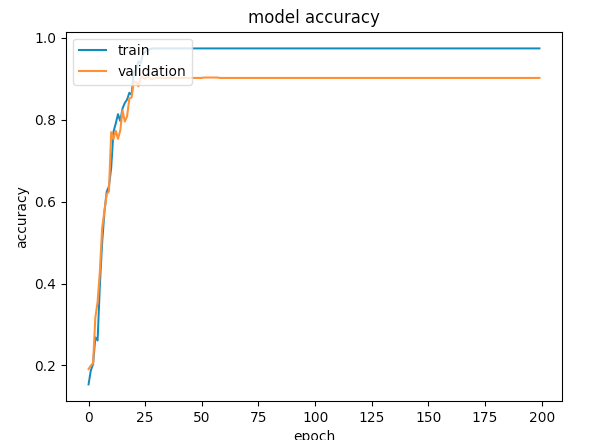
Overall precision = 91%

Overall Recall = 80%

From the above matrix, we can tell that the worst performing class in the list was 0, which usually had the most false positives amongst the classes. The cause for this could be anything from too few zeros in the training area, to too few of the other classes. However, since the proportions for the training set were preset, this could not be changed.

**Graph for Model Accuracy**

Below, we see that Accuracy increases exponentially as more epochs are initiated. We see that the validation and training set both proceed to a high level within the first 25 epochs, and then remain at a high level of accuracy for the remainder of the epocks.



**Missed Images for ANN**

While the most common mistakes occurred do to the other classes being compared to the zero class, we can see that there are still some other errors within the matrix as well. These three provide more examples:

****

**Real 2, Predict 8**

****

**Real 6, Predict 5**

****

**Real 9, Predict 7**