Molly Wirtz

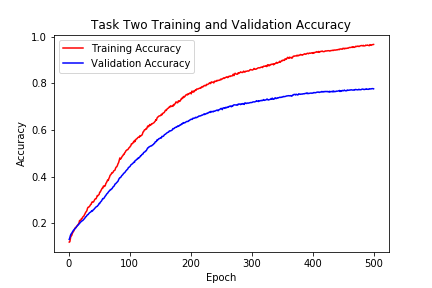
Professor Neil Heffernan

CS 4341 Introduction to Artificial Intelligence

8 September, 2019

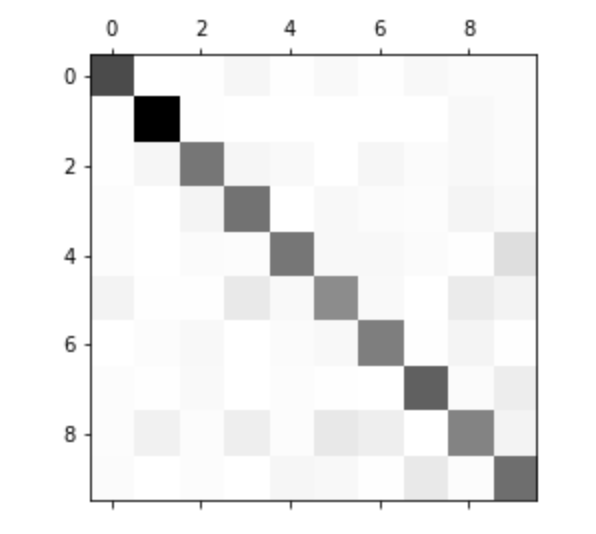
**Project Two: Artificial Neural Networks**

**Task Two:**



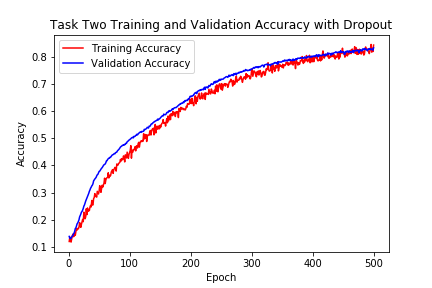
**Fig. 1: Training and Validation Accuracy Plot**

Pictured above in Figure 1 is the accuracy of the training set and validation set over epochs. Based on this plot, I can conclude that my model is roughly 80% accurate when compared to the validation data, and about 95% accurate when compared to the training data. Although this is not a bad result, running the network on the withheld test set shows a clearer picture of the accuracy. The test results produced an accuracy of 0.7646153814976032, and an error of 0.8665329990020165, represented below by the confusion matrix in Figure 2. The ANN is able to generally predict the correct answer, but has lots of room for improvement.



**Fig. 2: Confusion Matrix of Test Set**

After adding a 20% Dropout to the first layer of our model, I saw a significant difference in the closeness of my training and validation accuracy. As shown below in Figure 3, both plots now follow the same curve and trajectory. Although the training data showed a drop in accuracy, I believe the dropout result showed a much more reliable result than before. When validated against the test set, loss came to 0.6028240783397968, and accuracy measured as 0.8038461501781757, about a 5% improvement over running the neural net without dropout. The improvement in error is reflected by the confusion metric in Figure 4.



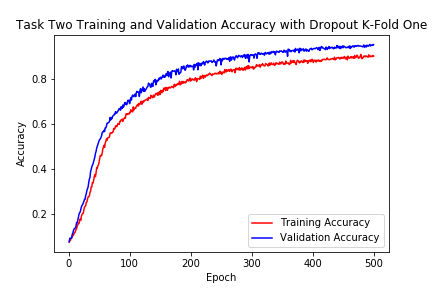
**Fig. 3: Training and Validation Accuracy Plot with 20% Dropout**



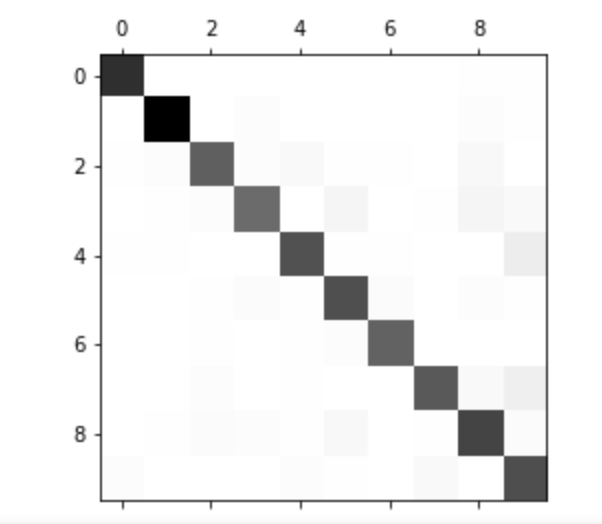
**Fig. 4: Confusion Matrix of Test Set with 20% Dropout**

**Task Three:**

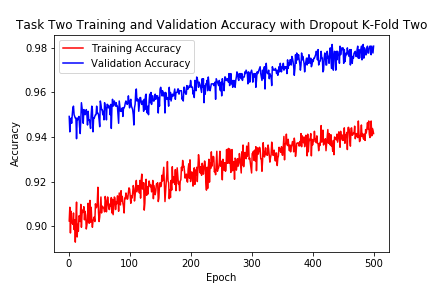
After implementing k-fold, the training, validation, and testing accuracy all improved. As shown by the plots in Figures 5, 7, and 9, training and validation now approaches 90% in k-fold one, and 100% in folds two and three. The average error rate between the three folds is 0.3118567, and the average accuracy is 0.9164, a 10% increase from the single-fold dropout neural network. The confusion matrices of all the folds are pictured below in Figures 6, 8, and 10.



**Fig. 5: Training and Validation Accuracy Plot with 20% Dropout, Fold One out of Three**



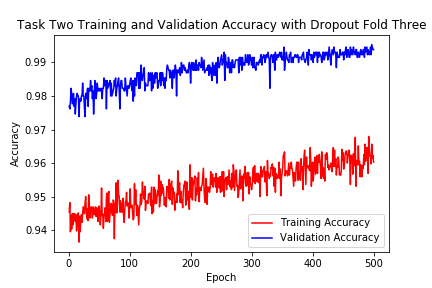
**Fig. 6: Confusion Matrix of Test Set with 20% Dropout, Fold One out of Three**



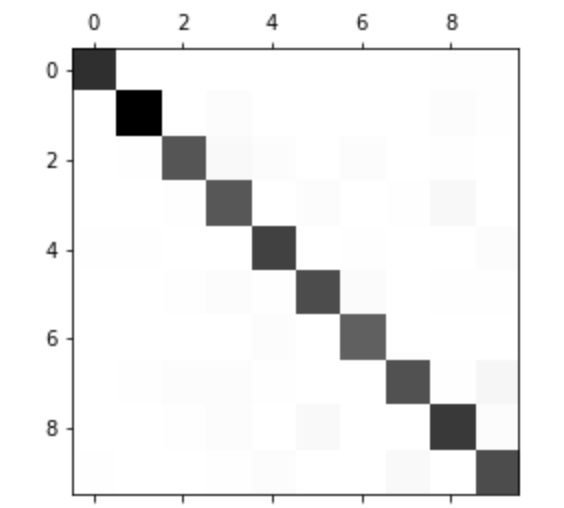
**Fig. 7: Training and Validation Accuracy Plot with 20% Dropout, Fold Two out of Three**



**Fig. 8: Confusion Matrix of Test Set with 20% Dropout, Fold Two out of Three**



**Fig. 9: Training and Validation Accuracy Plot with 20% Dropout, Fold Three out of Three**



**Fig. 10: Confusion Matrix of Test Set with 20% Dropout, Fold Three out of Three**

**Task Four:**

After experimenting with a 3-fold neural network with 20% dropout in the first layer and multiple sizes of hidden layers, I concluded that the more hidden layers, the more accurate the performance of the neural network on the test set. Figure 11 shows the data table of the error and accuracy rate for each fold. A network with a more extensive number of hidden layers outperforms networks with less layers because the more layers there are, the more opportunity there is for a more accurate computation. More layers mean that weights are distributed and calculated in many, many nodes, which makes it more accurate and robust when compared to a network with only a small number of hidden layers.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Fold One | Fold Two | Fold Three |
| **One Hidden Layer** | | | |
| Loss | 5.296781830420861 | 5.26376404102032 | 5.216514541919415 |
| Accuracy | 0.6676923113602858 | 0.6707692344372089 | 0.673846157514132 |
| **Two Hidden Layers** | | | |
| Loss | 3.918051929473877 | 3.857592313473041 | 3.7757817393082838 |
| Accuracy | 0.7507692307692307 | 0.7538461538461538 | 0.7553846152012165 |
| **Ten Hidden Layers** | | | |
| Loss | 0.34466708100759064 | 0.29626051939450776 | 0.2947371822137099 |
| Accuracy | 0.902307696892665 | 0.9200000036679782 | 0.9269230800408583 |

**Figure 11: Loss and Accuracy of all Folds with Varied Hidden Layers**

Next, I compared the batch size of 32 to a batch size of 512, both with 20% drop out in the first layer, 10 hidden layers, and with three-fold. I found that when I lowered the batch size, time became a more significant factor: it took much longer to run than a larger batch size. The accuracy, however, improved by about 5% on the smaller batch when evaluated on the test set. There appears to be a tradeoff between accuracy and efficiency when it comes to choosing a batch size for a neural network. The final results are shown below in Figure 12.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Fold One | Fold Two | Fold Three |
| **Batch Size 32** | | | |
| Loss | 0.2819265704831252 | 0.3712456666133725 | 0.34672120205986384 |
| Accuracy | 0.9523076921242934 | 0.9446153844319857 | 0.9530769228935242 |
| **Batch Size 512** | | | |
| Loss | 0.34466708100759064 | 0.29626051939450776 | 0.2947371822137099 |
| Accuracy | 0.902307696892665 | 0.9200000036679782 | 0.9269230800408583 |

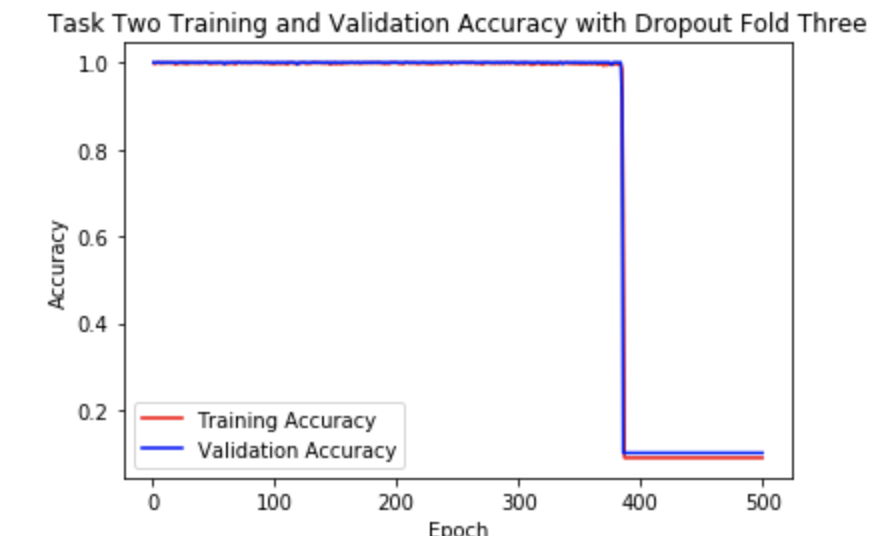
**Figure 12: Loss and Accuracy of all Folds with Varied Batch Size**

The final part of task four was to create three experiments of my own. I chose to experiment with batch size, dropout, and the number of nodes in a hidden layer.

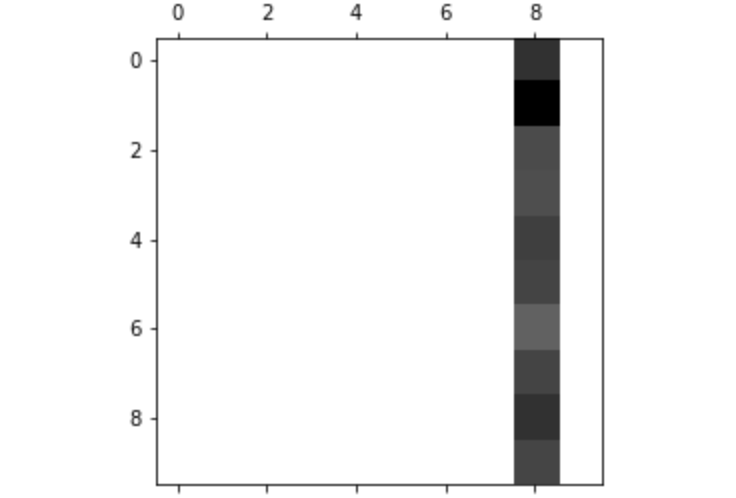
For my experiment with batch size, I ran my three-fold neural network with 20% dropout in the first layer and 10 hidden layers with a batch size of 10, 100, and then 512. On a size of 10, I did not see much significant improvement in the accuracy compared with batch size of 32 (as shown in Figure 12); the time, however, increased drastically, averaging about 13 minutes per fold. The tradeoff between time and accuracy is not feasible for a batch size of 10. A batch size of 100 showed 95% accuracy with a time of under 2 minutes per fold. A batch size of 512 has an accuracy of about 91% in under a minute. The complete data can be found in Figure 13. The loss and accuracy data for this experiment was more or less what I expected based on the varied batch size experiment above, with the exception of fold three. After getting highly abnormal results, I re-ran that batch three times, but the result never changed. The loss for fold three came to 14.40709786781898, and the accuracy plummeted to 0.10615384813684683. The confusion matrix showed that nearly every example was interpreted as value 8. Even after a close evaluation of my code, I could not figure out what could have caused this drastic outlier. Figure 14 shows the training and validation plot for fold three, and Figure 15 shows its confusion matrix. The T-test between batch size 10 and size 100 resulted in 0.441349117, and the T-test between batch size 100 and 512 was 0.026045895. This means the there was a significant difference in my data between batch sizes 10 and 100, but not 100 and 500.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Fold One | Fold Two | Fold Three |
| **Batch Size 10** | | | |
| Loss | 0.3577428076016222 | 0.41343374001396427 | 14.40709786781898 |
| Accuracy | 0.9553846079569597 | 0.9523076850634354 | 0.10615384813684683 |
| **Batch Size 100** | | | |
| Loss | 0.2799204496236948 | 0.3171409096282262 | 0.3408070782629343 |
| Accuracy | 0.9353846174020034 | 0.9453846124502329 | 0.9453846170352056 |
| **Batch Size 512** | | | |
| Loss | 0.34466708100759064 | 0.29626051939450776 | 0.2947371822137099 |
| Accuracy | 0.902307696892665 | 0.9200000036679782 | 0.9269230800408583 |

**Figure 13: Loss and Accuracy of all Folds with Varied Batch Size**

****

**Figure 14: Irregular Training and Validation Plot**

****

**Figure 15: Irregular Confusion Matrix**

For my experiment with dropout size, I ran my three-fold neural network with 10 hidden layers, a batch 512, and varied dropout rates. I concluded that out of 0%, 20%, and 50% dropouts, 20% yielded the best results. As we saw earlier in this assignment, having no dropout can lead to overfitting, and a lack of accuracy when compared to the training set, and my data supports this conclusion. I found that there wasn’t a large difference between 20% and 50% dropout (something I was not expecting), but that 20% did produce better results by a small margin. The final data can be found in Figure 16. The T-test between 0% and 20% dropout resulted in 0.065399645, and the T-test between 20% and 50% was 0.387759962, which means that all the results show a significant difference, and are valid.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Fold One | Fold Two | Fold Three |
| **Dropout 0%** | | | |
| Loss | 0.48394001896564776 | 0.5806185722351074 | 0.6440129841290988 |
| Accuracy | 0.8907692349874057 | 0.8938461580643287 | 0.8938461580643287 |
| **Dropout 20%** | | | |
| Loss | 0.34466708100759064 | 0.29626051939450776 | 0.2947371822137099 |
| Accuracy | 0.902307696892665 | 0.9200000036679782 | 0.9269230800408583 |
| **Dropout 50%** | | | |
| Loss | 0.41973810920348537 | 0.2882455662580637 | 0.2518555971750846 |
| Accuracy | 0.853076917208158 | 0.9169230805910551 | 0.9269230800408583 |

**Figure 16: Loss and Accuracy of all Folds with Varied Dropout Size**

For my experiment with number of nodes, I ran my 3-fold, 10-layer neural network with 20% dropout in the first layer with 20, 50, and 80 nodes per hidden layer, and concluded that the more nodes, the more accurate a result the network produced. This supports my point I made earlier about number of hidden layers. More layers (and nodes per layer) means more calculations, weights, and opportunities to fit the data accurately. My node results, however, were not as drastically different as I had hoped. My T-test between 20 nodes and 50 nodes produced 0.03316695, and my T-test between 50 nodes and 80 nodes produced 0.00481268. Neither of these values cross the value of 0.05, meaning that the difference between my results are not significant enough to be valid. My final results are shown in Figure 17.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Fold One | Fold Two | Fold Three |
| **20 Nodes Per Hidden Layer** | | | |
| Loss | 1.0022185490681574 | 0.8213117357400748 | 0.7611197931949909 |
| Accuracy | 0.8369230712377108 | 0.8323076868057251 | 0.8061538421190702 |
| **50 Nodes Per Hidden Layer** | | | |
| Loss | 0.3547477575448843 | 0.29138128739136915 | 0.2828616774082184 |
| Accuracy | 0.8992307725319496 | 0.9207692337036133 | 0.9269230796740605 |
| **80 Nodes Per Hidden Layer** | | | |
| Loss | 0.2979143240818611 | 0.2716766166687012 | 0.2771025324784792 |
| Accuracy | 0.9100000036679782 | 0.9307692337036133 | 0.935384618135599 |

**Figure 17: Loss and Accuracy of all Folds with Varied Hidden Layer Nodes**