Phase 3

Model Selection and Evaluation

Alexander Garcia

December 2024

Contents

1 Introduction

The goal of this phase is to fit many different models and find the model that scored the highest accuracy and lowest loss on the validation set. Overall, 7 different models were fitted and validated using model checkpointing and early stopping. However, even though some models scored the highest validation accuracy they also resulted in overfitting on the training data. It is important to note that random shuffle was used so the results differ every time the model is run.

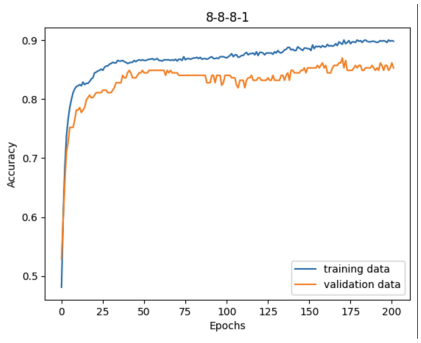
2 Model Selection

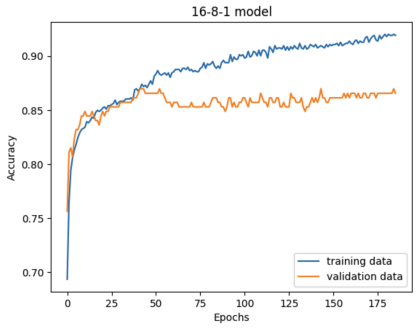
When fitting the models, model checkpointing was used based on validation loss and tracked the accuracy, precision, and recall metrics. An interesting find in the 3 and 4 layer models is that they do result in high validation accuracies, but clear signs of overfitting occur. The widening gap in the 16-8-1 model is a sign of overfitting. The 8-8-8-1 model appears to be heading toward overfitting as well while the cleanest learning curves come from the two layer networks.

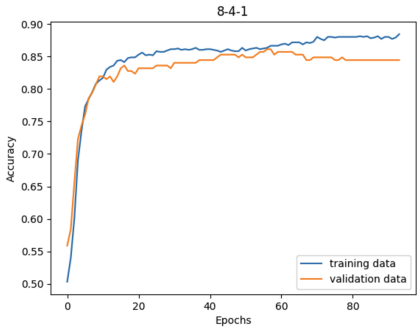
Model	Acc. Train	Acc. Valid	Prec Valid	Recall Valid
Baseline	51	56	N/A	N/A
Logistic Regression	83	84	88	82
Neural Network Model (16-8-1)	91	87	88	87
8-8-8-1	91	88	90	88
8-4-1	88	84	87	84
8-1	87	85	88	84
4-1	84	84	88	83
2-1	84	83	84	85

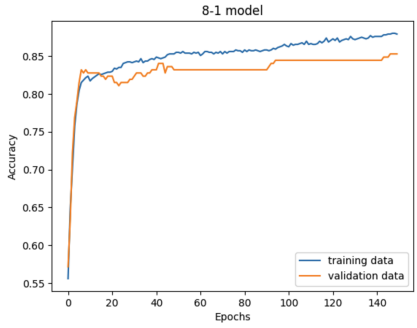
Table 1: Placeholder Caption

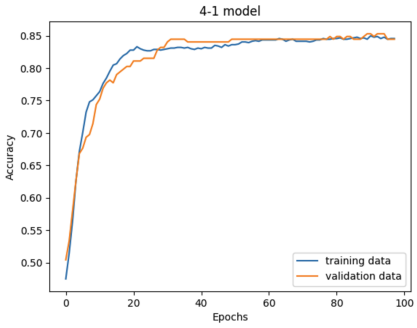
2.1 Learning curves

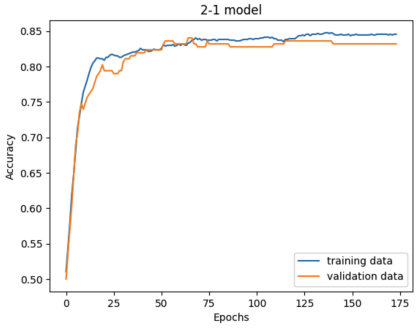


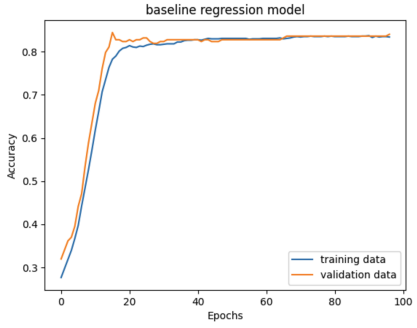






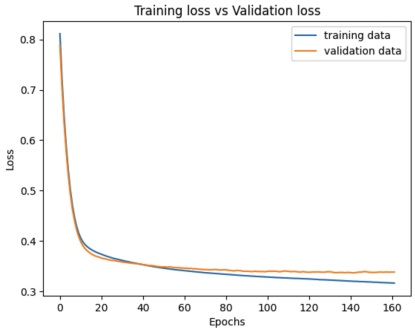






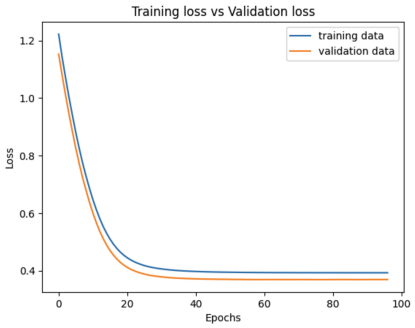
2.2 Training Loss vs Validation Loss

Here I compare the training loss versus the validation loss. As a model trains, this loss becomes important for updating the weights and biases. The model's objective is to minimize this loss and it will alter the weights and biases in a direction that minimizes the loss. It does so through a process known as backpropagation.



(a) 8-1 loss curve

(b) 8-8-8-1 loss curve



(c) 16-8-1 loss curve

3 Manual Prediction Function

In this portion, I wrote a manual prediction function using the weights and biases that resulted from training the model. Each layer is iterated and then using the formula from Keras docs

$$\text{output} = \text{activation}(\text{dot}(\text{weight kernel, input}) + \text{bias})$$

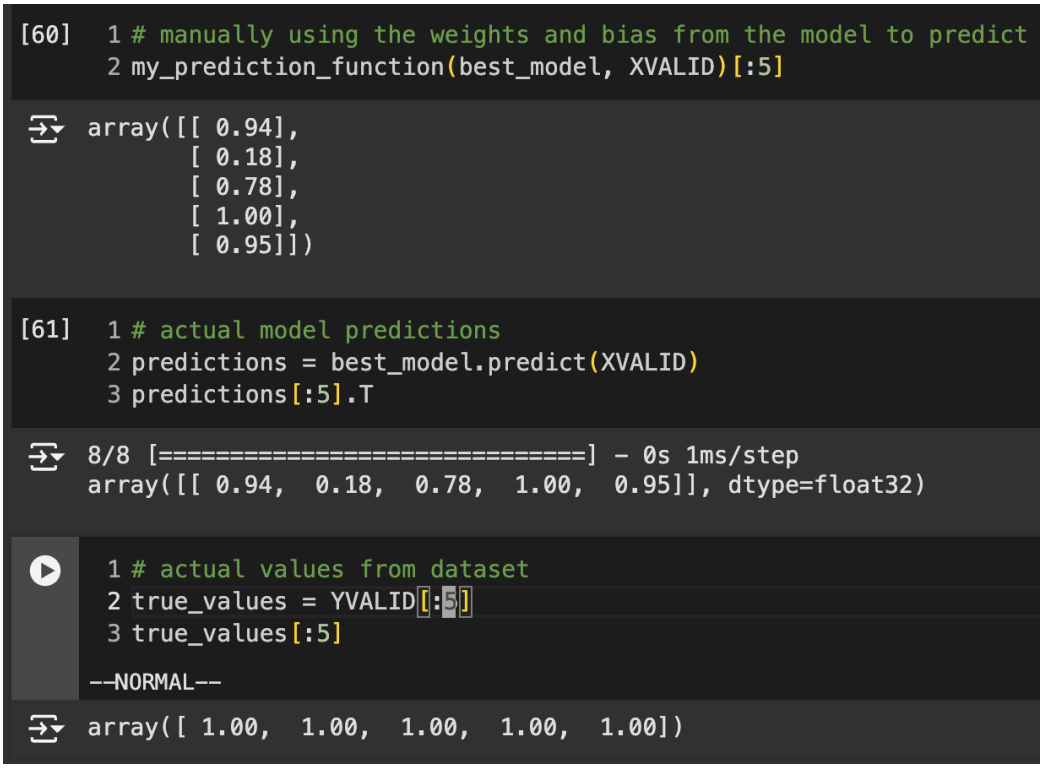
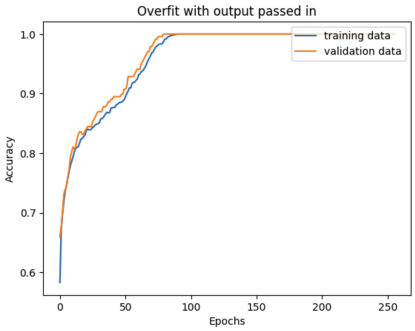


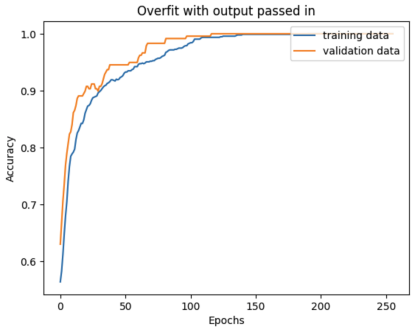
Figure 3: Data statistics

4 Overfit With Output Passed In

The purpose of this section is to test what architecture is needed to overfit the model when passing in the target value. As expected the model overfits with just one neuron at 100 epochs.



(a) 2-1 overfit with target passed in



(b) 1 neuron overfit with target passed in

5 Conclusion

In summary, I demonstrated the training and loss as the model runs over epochs. Even though the best model in terms of validation loss is 16-8-1 this model shows signs of overfitting. Some of the best curves are single layer with low amounts of neurons. When passing in the output value model appears to learn very quickly and overfits rapidly with a small architecture. This illustrates the importance of keeping the output variable out of the training data.