Assignment 1

Jacob Joy

Kent State University

Dr. CJ Wu

2/16/2025

Summary

Setup

Using the example code from the textbook for classification of sentiment analysis using the IMDB data set the initial code was set up. An attempt was made to incorporate a seed value of 113 as this is the value set by default in the IMDB. Unfortunately, due to the nature of running code on GPUs the seed only provides some stability in reproducing results. A batch size of 512 was used for all experiments.

Procedure

A total of 27 experiments were conducted with results recorded in the table below. Each trial followed the same procedure.

1. Select Model

2. Compile Model

3. Train Model on part of the training data at 20 epochs

4. Graph the Training and Validation Loss

5. Graph the Training and Validation Accuracy

6. Using the above graphs and information a lower epoch value was used in order to limit overfitting.

7. A new model was created using the same layers as in step 1.

8. Compile the model from step 7 using the same parameters as in step 2

9. Train this fresh model on all training data and using the epoch discovered in step 6

10. Evaluate the model's predictions on test, training, and validation vs given values of each.

Through my experiments it was discovered that steps 7, 8, and 9 must be done or the evaluations will have higher results.

Summary

Starting the experiments the activation used was relu for the hidden layers, RMSprop for the optimizer and binary cross entropy for the loss function. Starting with 1 hidden layer the units were increased by multiples of 8. The highest validation accuracy was 95.28% with a unit value of 64. Additional layers were added and tested with the results listed in the table below. The highest validation accuracy was a result of 4 hidden layers of 64 units each at an accuracy of 96.57%. With this information additional experiments were conducted using a base of 4 hidden layers and 64 units. The first experiment with this base was to try using MSE as the loss metric. This resulted in a decrease in both training and valuation accuracy. This makes sense as MSE is more acceptable to outliers and is best used with linear regression. The next experiment was to try a different optimizer, Adam was selected and produced a validation accuracy of 96.44% which is very similar to our original model. Next dropout was added to improve overfitting values of 0.2, 0.3, and 0.5 were used and it was found that 0.3 produced the highest validation accuracy of 97.14% and the training data accuracy also increased to 97.12%. Next, L2 regularization was used on our base model using the following values 0.00001, 0.0001, and 0.001. The highest validation accuracy achieved with L2 regularization at a value of 0.00001 was 94.33%. One final experiment was conducted using both dropout of 0.3 and L2 regularization of 0.00001. These values were chosen because on their own they produced the best results in comparison to other values used. This new model produced a validation accuracy of 94.79%. After all, the experiments the best result based on validation accuracy as requested in the instructions was using only dropout with a value of 0.3.

Lab Conclusion

There are large amounts of experimentation required to find the best models for a neural network. It is also difficult to compare results when reproducing the same results is challenging due to the nature of GPUs and cloud computing. My experimentation could have changed routes at any time due to this issue and upon retesting of my conclusion I have had validation accuracy ranging from 94% - 97.14%. While it is possible for me to achieve 100% validation accuracy by manipulating parameters such as the batch size, that would not make a useful model and is why I settled on 97.14%.

Result Summary Table

Experiment	Hidden Layers	Units	loss metric	Activation	Optimizer	dropout	L2 Regularization	Training Acc	Training Loss	Val Acc	Delta Training Val	Val Loss	Test Acc	Test Loss
1	1	8	binary_crossentropy	relu	RMSprop			0.9511	0.1583	0.9524	-0.0013	0.15855	0.8875	0.2816
2	1	16	binary_crossentropy	relu	RMSprop			0.9387	0.1873	0.9388	-1E-04	0.1872	0.8889	0.2784
3	1	64	binary_crossentropy	relu	RMSprop			0.9516	0.1489	0.9528	-0.0012	0.1492	0.8836	0.2886
4	1	128	binary_crossentropy	relu	RMSprop			0.9466	0.1598	0.9484	-0.0018	0.1594	0.8874	0.2798
5	2	8	binary_crossentropy	relu	RMSprop			0.9599	0.1309	0.9616	-0.0017	0.1309	0.8844	0.2981
6	2	16	binary_crossentropy	relu	RMSprop			0.9521	0.149	0.9531	-0.001	0.1485	0.8876	0.2852
7	2	32	binary_crossentropy	relu	RMSprop			0.9492	0.1548	0.9507	-0.0015	0.1551	0.8878	0.2815
8	2	64	binary_crossentropy	relu	RMSprop			0.9475	0.1535	0.9481	-0.0006	0.1538	0.8863	0.2833
9	2	128	binary_crossentropy	relu	RMSprop			0.9436	0.1434	0.9436	0	0.1441	0.872	0.3294
10	2	256	binary_crossentropy	relu	RMSprop			0.942	0.1575	0.9396	0.0024	0.1579	0.8715	0.3074
11	3	8	binary_crossentropy	relu	RMSprop			0.9559	0.1431	0.9572	-0.0013	0.142	0.885	0.2917
12	3	16	binary_crossentropy	relu	RMSprop			0.9277	0.209	0.9279	-0.0002	0.2084	0.8794	0.2992
13	3	32	binary_crossentropy	relu	RMSprop			0.9679	0.1063	0.969	-0.0011	0.1061	0.8792	0.3227
14	3	64	binary_crossentropy	relu	RMSprop			0.9578	0.1326	0.9593	-0.0015	0.1324	0.8848	0.288
15	3	128	binary_crossentropy	relu	RMSprop			0.9462	0.1411	0.9462	0	0.143	0.858	0.3499
16	4	64	binary_crossentropy	relu	RMSprop			0.9652	0.1032	0.9657	-0.0005	0.1038	0.8763	0.3244
17	4	128	binary_crossentropy	relu	RMSprop			0.9324	0.1709	0.9293	0.0031	0.1724	0.8564	0.35
18	4	64	binary_crossentropy	tanh	RMSprop			0.9508	0.1425	0.9498	0.001	0.143	0.8746	0.3187
19	4	64	MSE	relu	RMSprop			0.9401	0.0472	0.943	-0.0029	0.0457	0.8682	0.1013
20	4	64	binary_crossentropy	relu	adam			0.9647	0.1113	0.9644	0.0003	0.1125	0.8793	0.3209
21	4	64	binary_crossentropy	relu	RMSprop	0.2		0.9283	0.1762	0.9281	0.0002	0.1759	0.8538	0.4504
22	4	64	binary_crossentropy	relu	RMSprop	0.3		0.9712	0.0879	0.9714	-0.0002	0.0881	0.8844	0.3267
23	4	64	binary_crossentropy	relu	RMSprop	0.5		0.9563	0.1324	0.9557	0.0006	0.1329	0.8806	0.3172
24	4	64	binary_crossentropy	relu	RMSprop		0.00001	0.9425	0.1714	0.9433	-0.0008	0.1714	0.8887	0.2829
25	4	64	binary_crossentropy	relu	RMSprop		0.0001	0.898	0.3117	0.8975	0.0005	0.3144	0.8637	0.3691
26	4	64	binary_crossentropy	relu	RMSprop		0.001	0.937	0.3684	0.9374	-0.0004	0.3682	0.8871	0.4629
27	4	64	binary_crossentropy	relu	RMSprop	0.3	0.00001	0.9478	0.1464	0.9479	-1E-04	0.1456	0.8756	0.3278