Neural Networks Assignment Report – Nikhil Kumar Sampath

IMDB Dataset

Objective:

To build a neural network which can better generalize on the test set.

Model Building:

In total, we built 8 models with varied layers, nodes and other hyper tuning parameters.

Hyper Tunning Parameters:

#No.	Layers	Activation	Nodes	Regularization	Dropout	Optimizer	Loss
Model 1	2	relu	16	-	-	rmsprop	binary crossentropy
Model 2	1	tanh	64	-	-	rmsprop	mse
Model 3	3	relu	64	l2 (0.001)	-	rmsprop	binary crossentropy
Model 4	2	relu	64	-	0.5	rmsprop	binary crossentropy
Model 5	1	tanh	32	-	0.3	adam	mse
Model 6	2	relu	16	-	0.5	rmsprop	binary crossentropy
Model 7	**2**	**relu**	**16**	**-**	**0.4**	**rmsprop**	**binary crossentropy**
Model 8	2	relu	16	l2 (0.001)	-	rmsprop	binary crossentropy

#No.	Epochs	Batch Size	Optimal Epochs	Test Accuracy	Test Loss
Model 1	30	512	4	88.56%	28.78%
Model 2	50	512	3	88.37%	8.64%
Model 3	50	512	3	88.08%	42.28%
Model 4	50	256	3	88.52%	30.25%
Model 5	50	512	5	87.87%	8.85%
Model 6	30	512	7	88.71%	29.30%
Model 7	**20**	**512**	**4**	**88.85%**	**27.29%**
Model 8	20	512	6	88.60%	33.88%

Final Comments:

- Most of the models were built using the activation function as "**relu"** because they are simple, fast to compute and don't suffer from vanishing gradients like 'tanh'. relu also improves the neural network by speeding up training. Similarly, **binary crossentropy** was used as a loss function because it's the best loss (entropy) function to use when there is a classification problem.

- If noted the top models with the highest accuracy are the ones with **relu** being the activation function because they are the most widely used non-linear activation. Also, models 6 and 7 were built using 2 hidden layers, relu being the activation and dropout being added as a method to control the overfitting of the neural network.

- Model 6 was built with **0.5 being the dropout rate** i.e., out of 16 nodes only 8 nodes are forced to work in a layer which resulted in 88.71% accuracy whereas if the dropout rate was **further reduced to 0.4**, we are forcing approximately 10 nodes to work in a layer which resulted in 88.85% accuracy. So, we can think of **dropout** being an effective way to control the overfitting of a neural network, thereby leading to an increase in accuracy.

- Whereas **regularization**, another way to control the overfitting of a neural network, didn't significantly increase the accuracy of the models we built. Models 3 and 8 have been built using l2 regularization at a rate of 0.001, they didn't have much significance to increase the model's performance.

- If we are looking at the model's evaluation based on the **least loss** and **moderate accuracy** on the test data then **model 2** can be considered a better-generalized model because it had a minimal loss of 8.64% with an accuracy of 88.37%. It was built using '**tanh**' as an activation function but the loss was computed using the '**mse**' function. This can be a possible reason for getting minimal loss over the test data. But it must be noted that mse as a loss function is not suitable for "**binary sentiment classification"** whereas the IMDb dataset is all about classification.

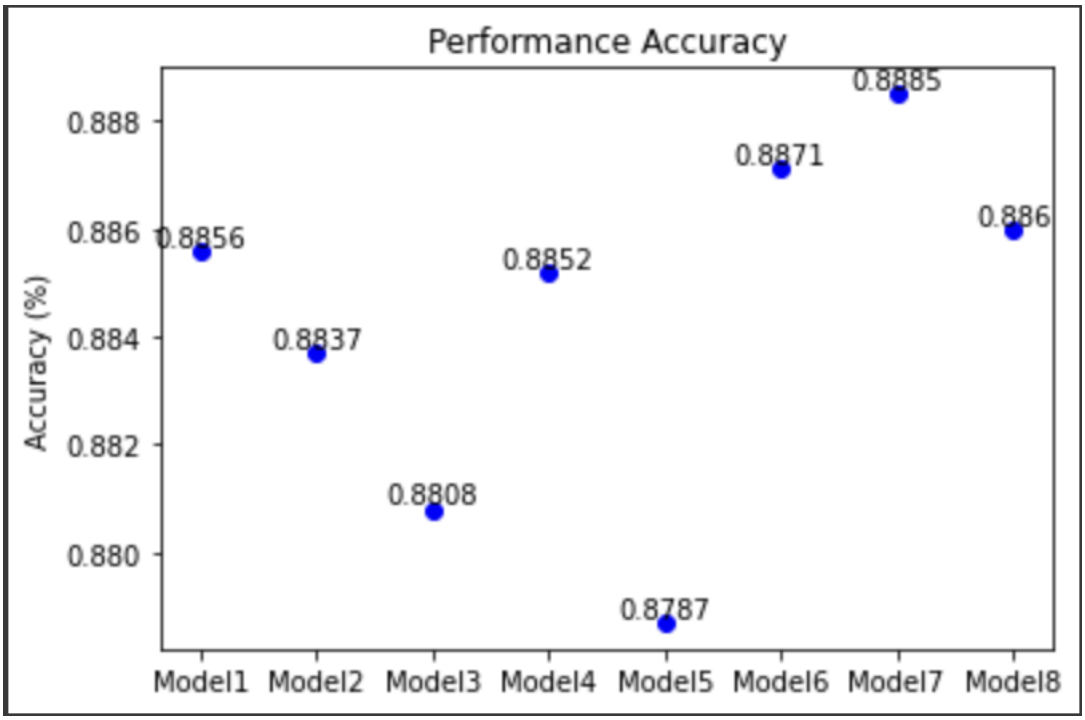
- An increase in the **hidden layers** from two to three didn't have a greater impact on the model's performance, whereas two and single-layer neural models have seemed to show greater accuracy on the test set.

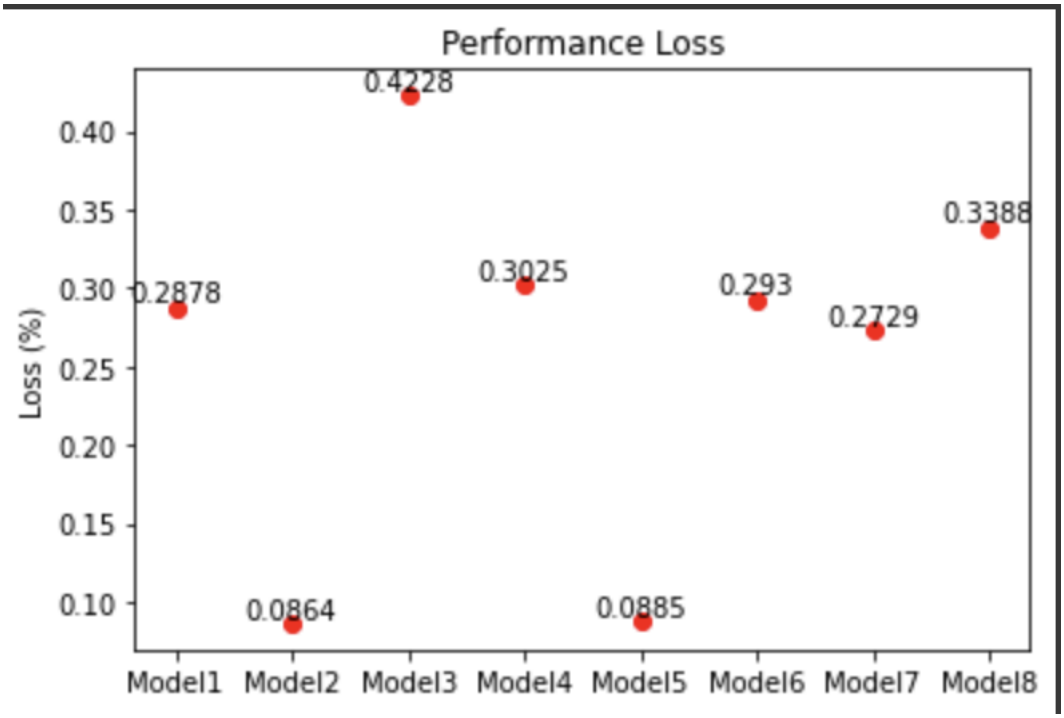
- In a similar manner setting the **epochs** to a higher count initially so that the model overfits didn't have a major impact, most of the models reached their optimal run during the initial epochs.

Conclusion:

The final model which can be considered the best model to generalize over the test set is Model 7, which was built using two hidden layers, 16 neurons per layer, a dropout rate of 0.4%, optimizer being rmsprop and loss function being binary cross entropy.

Accuracy – 88.85% and Loss – 27.29%




Performance Loss
Loss (%)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.4228
0.3388
0.3025
0.293
0.2878
0.2729
0.0864
0.0885
Model1
Model2
Model3
Model4
Model5
Model6
Model7
Model8