ADVANCED MACHINE LEARNING ASSIGNMENT 2

IMDB REPORT

INTRODUCTION:

The project's objective is to use the neural networks model on the IMBD Example to improve performance and explain the different approaches effect the performance of the model. The models are built using different hidden layers (1,2 &3) and fewer hidden units (32,64 & so on). The evaluation is based upon the model's accuracies and loss values.

DATA DESCRIPTION:

IMDB dataset having 50K movie reviews for natural language processing or Text analytics. This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 25,000 highly polar movie reviews for training and 25,000 for testing. So, predict the number of positive and negative reviews using either classification or deep learning algorithms.

METHODOLOGY:

I started importing the required packages and dependencies to preprocess the dataset and build a neural network model. This dataset is a in built in function Keras. The data is categorized into two parts- 0 or 1 i.e the sentiment of the review.

The data is partitioned into Train and Test with six different neural networks models configuration. It starts with defining the type of model we want build for this dataset. There are two types of models available in Keras i.e Sequential model and the functional API. The input are added – hidden layers and the nodes to prevent overfitting.

The relu function and Tanh function can be used as the activation function in the model building. The relu function is used with the hidden layers as it's a good start and always gives us a satisfactory result most of the time.
Once the model is complied the Rmsprop and Adam can be used as the algorithm changes the weights and biases during the training process. I choose the binary-crossentropy and MSE as the loss and accuracy as the evaluation metrics.

The batch size specifies how many samples will be propagated through the network, and an epoch is an iteration over the entire training data set. A larger batch size results in faster training but does not always converge as quickly. A smaller batch size takes longer to train but can converge faster. This is clearly problem dependent, and you'll need to explore with a few different values. If this is your first time working with a problem, I recommend starting with a batch-size of 32, which is the standard size.

MODEL	LAYERS	NODES	ACTIVATION	OPTIMISER	LOSS FUNCTION	LOSS	ACCURACY
1	2	16	tanh	rmsprop	MSE	0.08	0.8803
2	1	64	tanh	rmsprop	MSE	0.08	0.8889
3	2	64	tanh	rmsprop	MSE	012	0.8350
4	3	64	relu	rmsprop	Binary-crossentropy	0.41	0.8872
5	3	16	relu	rmsprop	Binary-crossentropy	0.29	0.8810
6	1	32	tanh	adam	MSE	0.09	0.8706

RESULTS:

From the above table, we can see that the models have different levels of loss and accuracies.

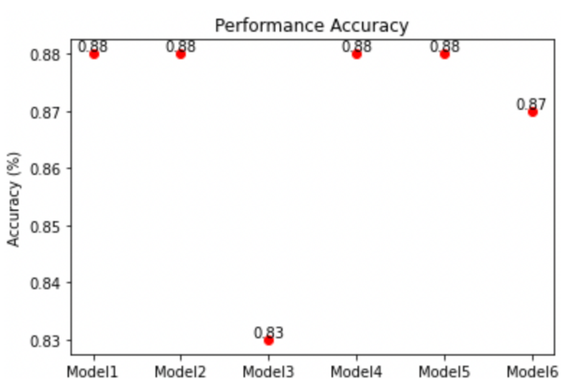
- The Model 1 and 2 have the highest accuracy of **0.8803 or 88.03 and 0.8889 or 88.89%** respectively**,** while the Model 3 has the lowest accuracy of all **0.83.50 or 83.50%.**
- The model 1 and 2 have the lowest loss of **0.08 or 8%** while the Model 4 has the highest loss of **0.41 or 41%.**

The best performing model in terms of **Accuracy** are **Model 1 and 2** but the number of nodes used in the **Model 1 are 16 nodes** in each of the 2 layers and the nodes in **Model 2 are 64** in its layer. This means the **Model 1** performed better than the Model 2 as it used fewer nodes.

Model 4 and 5 used the **relu function** as activation**, binary- cross entropy** as loss function and **rmsprop optimizer.**

The other Models (1,2,3,6) used the **relu function** as activation,**MSE as loss function** and **rmsprop optimizer**.

Overall, the best performing model is the **Model 1** as it has a lower loss value of **0.08 or 8%** and the accuracy of **0.88 or 88%.** This indicates that **Model 1** is better at minimizing the difference between its predicted outputs and the actual outputs on the training data.



CONCLUSION:

Overall, our findings emphasize the importance of carefully selecting a neural network's architecture, activation functions, and hyperparameters, as these decisions can have a significant impact on the model's performance on a given task.

Finally, we evaluated the results of six different neural network models trained on the IMDB dataset. The models were evaluated based on their loss and accuracy values. The results show that different configurations can result in varying levels of performance.