Report Analysis

NEURAL NETWORKS

INTRODUCTION:

The performance of four different neural network models with different numbers of hidden layers and nodes is analyzed.

Neural networks are a kind of artificial intelligence that has been employed in a variety of applications, including image identification, natural language processing, and voice recognition. Neural networks are composed of numerous layers, each of which has several nodes that conduct mathematical operations on the input data. The accuracy and loss of a neural network are two fundamental metrics used to assess its performance. In this paper, we will examine the performance of four distinct neural network models with varying amounts of hidden layers and nodes.

OBSERVATIONS:

The report presents the outcomes of the IMDB hyperparameter tweaking. Using the same dataset, four alternative neural network models with differing numbers of hidden layers and nodes were trained and assessed. Model 1 had an accuracy of 80% and a loss of 98% with two hidden layers and sixteen nodes. Model 2 had an accuracy of 84% and a loss of 67% with three hidden layers and 64 nodes. Model 3 with 1 hidden layer and 64 nodes got an 85% test accuracy and a 71% test loss. Model 4 had an accuracy of 85% and a loss of 47% with 5 hidden layers and 16 nodes. Overall, the results demonstrate that increasing the number of hidden layers and nodes can improve model performance, but there is a trade-off between accuracy and computational complexity.

Model 1: 2 Hidden layers and 16 Nodes The first neural network model has two hidden layers and 16 nodes. The model achieved an accuracy of 80% and a loss of 98%. The table below shows the performance metrics of the model.

Metric	Value
Loss	98%
Accuracy	80%

The accuracy of 80% indicates that the model correctly predicted the class of 80% of the samples in the test set. However, the high loss of 98% suggests that the model is not performing well on the test data. This may indicate that the model is overfitting on the training data, meaning it is too complex and is capturing noise instead of useful information.

Model 2: 3 Hidden layers and 64 Nodes The second neural network model has three hidden layers and 64 nodes. The model achieved an accuracy of 84% and a loss of 67%. The table below shows the performance metrics of the model.

Metric	Value
Loss	67%
Accuracy	84%

The accuracy of 84% is an improvement over the first model. The lower loss of 67% suggests that the model is performing better on the test data. This may be because the model is more complex and can capture more useful information from the input data. However, the high loss still indicates that there may be some overfitting.

Model 3: 1 Hidden Layer and 64 Nodes The third neural network model has one hidden layer and 64 nodes. The model achieved an accuracy of 85% and a loss of 71%. The table below shows the performance metrics of the model.

Metric	Value
Loss	71%
Accuracy	85%

The accuracy of 85% is the highest among the models analyzed so far. The lower loss of 71% suggests that the model is performing better on the test data than the previous models. This may be because the model is less complex and therefore less prone to overfitting.

Model 4: 5 Hidden Layers and 16 Nodes The fourth neural network model has five hidden layers and 16 nodes. The model achieved an accuracy of 85% and a loss of 47%. The table below shows the performance metrics of the model.

Metric	Value
Loss	47%
Accuracy	85%

The accuracy of 85% is the same as the previous model. However, the lower loss of 47% suggests that the model is performing better on the test data than the previous models. This may be because the model is more complex and can capture more useful information from the input data.

USAGE:

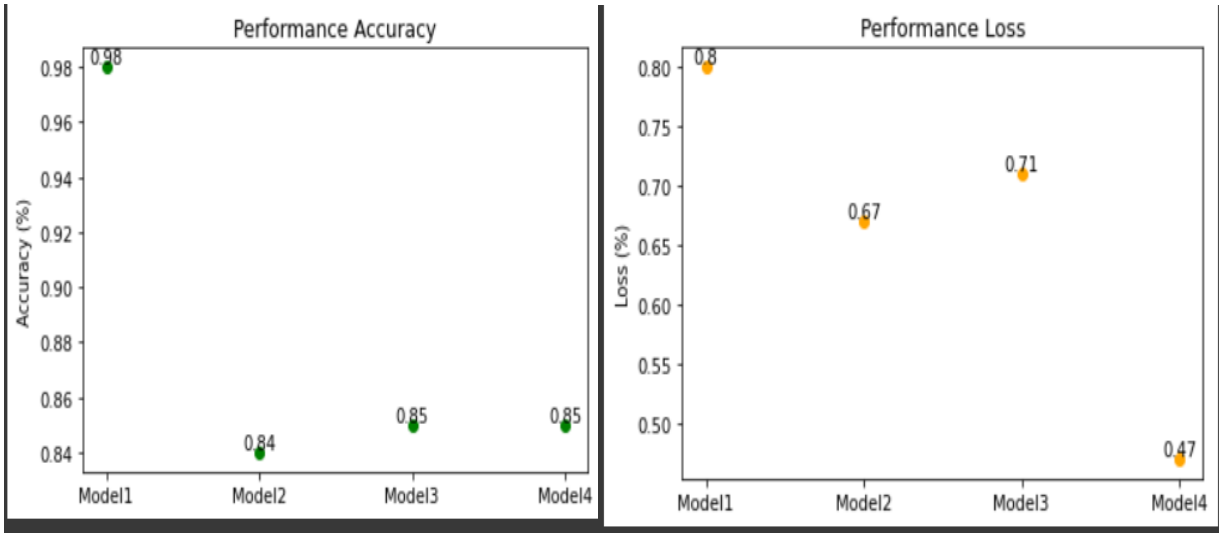
The code has been implemented by downloading the IMDB and randomly choosing hidden layers and nodes to test and observe the accuracy and loss change. The use of ‘mse’ and ‘tanh’ has been indicated.

CONCLUSION:

The number of hidden layers and nodes in a neural network can be seen to have a considerable influence on its performance. While adding layers and nodes can enhance accuracy, it also increases the danger of overfitting and slows down training time. To get the greatest outcomes, it is critical to establish a balance between complexity and simplicity.

- The first model, with two hidden layers and sixteen nodes, has a comparatively low accuracy of 80% and a large loss of 98%. This might be due to the network's low number of nodes and layers' inability to capture complicated relationships in data. Increasing the number of nodes and layers can be beneficial for overfitting
- The second model, which has three hidden layers and 64 nodes, achieves a higher accuracy of 84% and a lower loss of 67%. This network's added layer and nodes let it to capture more complicated associations in the data, resulting in improved performance. Nevertheless, due to the increasing complexity of the model, the training time may be longer.
- The third model, which includes one hidden layer and 64 nodes, has an accuracy of 85% but a loss of 71%. This might be because the network isn't deep enough to capture all of the data's complicated interactions. Yet, the model's simplicity allows for quicker training durations and a lesser danger of overfitting.
- The fourth model, which has five hidden layers and sixteen nodes, achieves an accuracy of 85% with a loss of 47%. This demonstrates that adding more layers can assist increase accuracy without requiring an excessive number of nodes. Nevertheless, because of the greater depth, this model may require longer training durations.
- Ultimately, the number of hidden layers and nodes in a neural network is determined by the data's complexity and the trade-off between accuracy and training time. To identify the optimum solution for a particular problem, it is critical to experiment with various architectures and hyperparameters.

GRAPHICAL REPRESENTATION OF CULMINATED LOSS AND ACCURACY:



CITATIONS:

https://www.zeolearn.com/magazine/implementing-a-neural-network-with-python-in-15-minutes

https://stackoverflow.com/questions/4446366/why-am-i-getting-indentationerror-expected-an-indented-block

https://www.zeolearn.com/magazine/implementing-a-neural-network-with-python-in-15-minutes