**ASSIGNMENT 1 – NEURAL NETWORKS**

**REPORT**

**STEPS THAT ARE DONE BEFORE BUILDING UP THE MODEL:**

The IMDB dataset is being loaded from TensorFlow. With a 10,000 word vocabulary, the dataset is loaded using the `imdb.load\_data()` function. As a result, training efficiency is increased and noise is decreased because only the 10,000 words that appear the most frequently in the dataset are kept.   
  
Training and test sets are separated from the dataset. Accessing the first movie review in the training set displays a string of integers that each correspond to a word's index in the dataset dictionary.  
  
It is determined that the dataset's maximum index value is 9999, indicating that the vocabulary size restriction has been appropriately enforced. It aids in determining the size of the neural network model's input layer.

In order to transform each integer sequence into a vector representation that can be fed into the neural network for training, multi-hot encoding is used to encode the sequences.

**BUILDING THE MODEL:**

There are 64 neurons in each of the model's four completely connected layers. Whereas the hidden layers employ the "tanh" activation function, the output layer employs the "sigmoid" activation function. Adam was selected as the optimizer for training, and "mse" was selected as the loss function.   
  
A batch size of 512 is used to train the model over four epochs. Model performance is evaluated using accuracy and loss as assessment criteria.  
  
To stop overfitting, L2 regularization and dropout layers are used. The model's ability to generalize to new data is ensured by the dropout layers' dropout rate of 0.5.   
  
The optimizer, loss function, and evaluation metrics must be defined before the model can be assembled. In order to evaluate model performance and identify any overfitting, the validation accuracy and loss are tracked throughout training.

A graph with green lines and red dots

AI-generated content may be incorrect.

The neural network's training and validation loss over epochs is displayed in the graph above. Epochs 1–20 are represented by the X-axis, and loss values are displayed on the Y-axis. Effective learning is indicated by a steady decline in the red line, or training loss. After epoch 10, the green dashed line (validation loss) varies after initially dropping, indicating overfitting, a situation in which the model retains training data but finds it difficult to generalize.  
  
Techniques like as L2 regularization, dropout, and early halting can be used to enhance generalization. Increasing data or simplifying the model can be beneficial. To avoid needless training and improve model performance on unknown data, validation loss monitoring is essential.

A graph of a graph

AI-generated content may be incorrect.

This graph shows the neural network's training and validation accuracy over epochs. Epochs (1–20) are indicated by the X-axis, and accuracy is displayed on the Y-axis. The model is learning effectively when the solid green line (training accuracy) climbs gradually. Validation accuracy, represented by the dashed yellow line, climbs at first but then varies across a few epochs.  
  
The discrepancy between validation and training accuracy points to overfitting, a situation in which the model does well on training data but poorly on unknown data. Techniques like data augmentation, early halting, and dropout can be applied to enhance generalization. Performance optimization and the avoidance of overtraining are aided by validation accuracy monitoring.

**SUMMARY:**

The training accuracy exhibits a comparable improvement, while the validation accuracy rises from an initial value to a final accuracy of almost 88%. That means the model is learning well and isn't overfitting.

The validation loss shows that the model's predictions are becoming more accurate and confident with time.  
  
A final test accuracy of about 87% was attained, with a test loss of about 0.13. Improvements in performance and stable training were facilitated by the Adam optimizer.   
  
When compared to the RMSprop optimizer, which had a marginally lower accuracy of 86%, Adam's efficacy for this task was evident.

**SUMMARY OF VALIDATION LOSSES IN TABULATED FORM:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nodes | Layers | Activation | Loss | Regularization | Validation Loss |
| 16 | 3 | tanh | MSE | Yes | 0.12 |
| 32 | 3 | ReLU | MSE | Yes | 0.13 |
| 64 | 3 | tanh | MSE | Yes | 0.10 |

**VALIDATION AND TRAINING ACCURACY:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Activation | Epochs | Test Accuracy | Test Loss |
| Nodes=64, Layers=3, tanh | tanh | 4 | 88% | 16.32 |
| Nodes=64, Layers=3, ReLU | ReLU | 4 | 86% | 21.25 |