Assignment 1: Project Report

Objective : Application of deep learning concepts using Keras and Tensorflow and To modify an existing neural network and evaluate its performance for the changes in input and output.

The IMDb dataset in Keras typically contains a large number of movie reviews. In its common form, it is split into two sets: a training set and a testing set. The training set is used to train machine learning models, while the testing set is used to evaluate their performance. Each set may contain thousands of movie reviews.

In the data set we are limiting the most frequently repeating words as it is a common preprocessing step in natural language processing (NLP). This process is typically referred to as "word frequency thresholding" The purpose of this step is to reduce the dimensionality of the dataset and focus on the most informative words while discarding less relevant or frequent ones.

Rewriting the data in 0 and 1's format as prepare text data for input into a neural network. Text data needs to be converted into a format that can be processed by a neural network.

We one-hot-encode each word in your vocabulary, representing each word as a binary vector where only one element (corresponding to the word's index) is 1, and all others are 0.

Each sequence is then represented as a matrix of one-hot vectors, which can be used as input to a neural network.

Models	**Layers**	**Neurons**	**Activation**	**Optimizer**	**dropout**	**Regularization**	**Loss Function**	**Epochs**	**Batch Size**	**Loss**	**Accuracy**
1	2	16	relu	rmsprop	0.00		Binary Crossentropy	4	512	0.2874	0.8849
2	1	64	tanh	rmsprop	0.00		mse	3	512	0.0997	0.862
3	3	64	relu	rmsprop	0.00	l1-0.001	Binary Crossentropy	3	512	0.4889	0.8521
4	2	64	relu	rmsprop	0.50		Binary Crossentropy	3	256	0.2898	0.884
5	1	32	tanh	rmsprop	0.30		Binary Crossentropy	5	512	0.0878	0.8808
6	2	16	relu	rmsprop	0.50		Binary Crossentropy	7	512	0.2978	0.8852
7	***2***	***16***	***relu***	***rmsprop***	**0.40**		***Binary Crossentropy***	***4***	***512***	***0.2729***	***0.8899***

- We are using **ReLU** is in most of the cases over tanh in most cases because it mitigates the vanishing gradient problem more effectively and is computationally efficient due to its piecewise linear nature, making it well-suited for training deep neural networks where as tanh is that it can suffer from the vanishing gradient problem, especially near its saturation points, making it less effective for training very deep neural networks.

- **ReLU** tends to be more computationally efficient than other activation functions like **sigmoid or tanh** because it simply turns off (sets to zero) all negative inputs.

- Model 2 and model 6 has the lowest Test loss of about 8-9% but IMDB problem is a Sentiment analysis problem and it often involves capturing non-linear relationships between words and sentiment. Tanh, being a symmetric activation function, may not be as effective as the rectified linear unit (ReLU) at capturing these non-linearities.

- Overfitting is typically managed through techniques like dropout, regularization, early stopping, As mentioned earlier, dropout randomly drops a fraction of neurons during training, which can help prevent overfitting by adding noise and reducing the reliance on any specific neuron.

- Increasing the number of hidden layers in the models did not consistently improve performance on the IMDb dataset. Models 1 and 5 with fewer hidden layers perform well, while models 3, 6, and 7 with more hidden layers show similar or slightly lower accuracy. Model 4 with two hidden layers and moderate dropout achieves good accuracy.

- Increasing the number of epochs generally improves model performance up to a certain point by allowing the model to learn from the training data for a longer duration, but overfits the model after a certain point and doesn't have any significant effect on the models performance.

- It appears that adding dropout to the models (except Model 2) generally improves model performance, as indicated by higher accuracy values. Dropout helps regularize the models, preventing overfitting and improving their ability to generalize to unseen data. Model 4, which has a dropout rate of 0.50, demonstrates a significant increase in accuracy, indicating the effectiveness of dropout in controlling overfitting.

Model 7 stands out as the best model with an accuracy of 0.8899, which is the highest among the listed models. It combines two hidden layers with ReLU activation, moderate dropout (0.40), and binary cross-entropy loss, demonstrating excellent performance on the task.