# ADVANCED MEACHINE LEARNING
# Neural Networks
# ASSIGNMENT 1

**Yashwanth Reddy Sadala**
**811292491**

# SUMMARY:

The IMDB dataset has provided 50,000 reviews. The total word count of these reviews is 10,000. These evaluations have all already been labeled. These are the steps I used to determine which hyperparameter combination will increase test and validation accuracy.
I started by importing all the modules needed to create and run a model. Those are:

```
import os
from operator import itemgetter import numpy as np
import pandas as pd
import matplotlib.pyplot as plt import tensorflow as tf

from keras import models, regularizers, layers, optimizers, losses, metrics
from keras.models import Sequential
from keras.layers import Dense

from keras.datasets import imdb
```

Then the IMDB dataset was loaded.
The dataset is separated into two sections: training and testing. Each section contains 25000 reviews. The data is then vectorized because we cannot input integers into neural networks after that. The training set is further separated into two parts: partial training and validation set.

**MODEL BUILDING:**

To construct the first three models, 16 hidden units, the ReLU activation function, the rmsprop optimizer, and the binary cross entropy loss function were used. Each model has a modified number of hidden layers. The accuracy for training, validation, and testing is shown in the following table.

| Combination | Validation accuracy | Accuracy |
|---|---|---|
| One hidden layer | 0.8895 | 0.8824 |
| Two hidden layers | 0.8696 | 0.8582 |
| Three hidden layers | 0.8440 | 0.8326 |

We can say that one hidden layer performs better than the other two layers in the IMDB dataset because there are less data samples available. Multiple layers cause overfitting, which lowers test and validation accuracy. The next two models are constructed using a single hidden layer, the ReLU activation function, the rmsprop optimizer, and the binary cross entropy loss function. The hidden units are now 32 and 64 in number. Results are displayed in the following table.

| Combination | Validation accuracy | Accuracy |
|---|---|---|
| 16 hidden units | 0.8895 | 0.8824 |
| 32 hidden units | 0.8746 | 0.8649 |
| 64 hidden units | 0.8681 | 0.8576 |

Increasing the hidden units leads to overfitting problems similar to those with hidden layers. 16 hidden units is the ideal combination to use for maximum accuracy.

Now let's change the loss function to mse and compare the accuracy to binary cross entropy. These models feature 16 hidden units, an rmsprop optimizer, a ReLU activation function, and one hidden layer.

| Combination | Validation accuracy | Accuracy |
|---|---|---|
| Binary cross entropy | 0.8894 | 0.8824 |
| MSE | 0.8774 | 0.8685 |

It is clear that the binary cross entropy loss function is preferable than the MSE loss function for the provided dataset.
The performance of the Tanh and ReLU activation functions is compared in the next model, which is constructed using one hidden layer, sixteen hidden units, an rmsprop optimizer, and a binary cross entropy loss function.

| Combination | Validation accuracy | Accuracy |
|---|---|---|
| relu | 0.8894 | 0.8824 |
| tanh | 0.8720 | 0.8616 |

Relu has given better accuracy than tanh. So, I’m going to use relu as my activation function for the next models.

Using dropout:

I used the Drop out (0.5) in the following model. I utilized the rmsprop optimizer, 16 hidden units, one hidden layer, binary cross entropy loss function, and relu activation function in this model.

| Combination | Validation accuracy | Accuracy |
|---|---|---|
| Without drop out | 0.8894 | 0.8824 |
| With drop out (0.5) | 0.8799 | 0.8724 |

The use of dropouts is causing accuracy to decline. This could be the case because dropout is not advantageous in this scenario and the model is not overfitting.

**Using regularizers:**

One hidden layer, sixteen hidden units, the rmsprop optimizer, the binary cross entropy loss function, and the relu activation function are used in the construction of the next two models. The effectiveness of L1 and L2 regularizers is compared.

| Combination | Validation accuracy | Accuracy |
|---|---|---|
| No regularizer | 0.8894 | 0.8824 |
| L1 regularizer | 0.8760 | 0.8674 |
| L2 regularizer | 0.8748 | 0.8660 |

It is clear from the following table that regularizers are not required for this data because their use does not increase accuracy.

contrasting rmsprop and Adam Optimizer. One hidden layer, sixteen hidden units, a binary cross entropy loss function, and a relu activation function are used in the construction of the model

| Combination | Validation accuracy | Test accuracy |
|---|---|---|
| rmsprop | 0.8894 | 0.8824 |
| Adam | 0.8585 | 0.8450 |

Rmsprop has performed better than adam for the given IMDB dataset.

**Conclusion:**

The effectiveness of a model in a neural network is influenced by various aspects. The size of the data sample is the first. In order to increase accuracy, more data samples are needed. In this instance, with 25000 samples, one layer with 16 hidden units performs better than other hidden layers. Taking into account that the model benefits from the ReLU function. The accuracy decreased when I experimented with alternative optimizers and regularizers, which made me realize that we shouldn't actually utilize it until absolutely necessary.