# Summary Report: Hyperparameter Tuning for IMDB

This report outlines the findings of experiments carried out to enhance a neural network model using the IMDB dataset. The goal is to fine-tune a range of hyperparameters for the model. We explored multiple configurations, incorporating variations in the number of layers, the number of units, loss and activation functions, and regularization techniques, in order to see how they impacted the model's performance.

**Dataset Formation and Processing**

The IMDB dataset was imported, featuring a vocabulary that consists of 10,000 words. The dataset was divided into training, validation and test sets. 50,000 samples divided into 25,000 training and 25,000 test samples, again training dataset was further divided into 15,000 training and 10,000 validation datasets.

**Experiments and Results**

1. **Single Layer Model with Embedding Layer**
   This model included an embedding layer, then a flattening layer, a dense hidden layer that used ReLU activation, and finally a sigmoid output layer.

- Optimizer: Adam
- Loss Function: Binary cross-entropy
- Epochs: 30
- Batch size: 64

2. **32 and 64 Units**
   Two different models were developed: one featuring 32 units in each hidden layer and the other with 64 units in each hidden layer. Both models included two hidden layers that used ReLU activation, and finally a sigmoid output layer.

- Optimizer: RMSprop
- Loss Function: Binary cross-entropy
- Epochs: 20
- Batch size: 512

3. **MSE Loss Function**
   This model included two hidden layers with 16 units and ReLU activation, and a final sigmoid output layer.

- Optimizer: RMSprop
- Loss Function: MSE
- Epochs: 20
- Batch size: 512

4. **Using Tanh Activation**

This model included two hidden layers with 16 units and tanh activation, and a final sigmoid output layer.

- Optimizer: RMSprop
- Loss Function: Binary cross-entropy
- Epochs: 20
- Batch size: 512

5. **Regularization and Dropout**:
   This model included Two hidden layers with 16 units and ReLU activation, L2 regularization (0.001), and Dropout layers (0.5), and a final sigmoid output layer.

- Optimizer: RMSprop
- Loss Function: Binary cross-entropy
- Epochs: 20
- Batch size: 512

**Results (Summary Table)**

| Experiment Model | Accuracy |
|---|---|
| Single Layer with Embedding | 86.10 % |
| Model with 32 hidden units | 49.52 % |
| Model with 64 hidden units | 50.28 % |
| Model with MSE loss function | 50.30 % |
| Model with tanh validation | 50.41 % |
| Model with Regularized and dropout | 50.06 % |

- The single layer model with embedding layer achieved the highest accuracy (86.10) among all models.
- Also, models with different units, MSE loss, Tanh activation, regularization, and dropout doesn’t perform well compared to single layer model.
- Binary cross-entropy loss function works better than the MSE.
- Along with this, ReLU activation performs better than the tanh activation.

**Conclusion**

The single layer model with embedding layer outperformed all other model. This highlights the importance of having an embedding layer for text data, as it effectively converts the input word index values into a more compact vector format. If there are no embedding layer, the model performance is declining. Binary cross-entropy loss combined with ReLU activation works best in this scenario. To enhance performance, more testing is required to fine tune the hyperparameters.