ADVANCED MACHINE LEARNING ASSIGNMENT -2

PROJECT REPORT

1. Introduction

The purpose of this assignment was to test different neural network designs on the IMDB movie review dataset from keras library. The task is to classify reviews as either positive or negative. To get the best performance, I experimented with changing the number of layers, hidden units, loss functions, activation functions, and regularization methods.

1.1 About the Dataset

I used the IMDB movie review dataset, which is a collection of reviews already labeled as either positive or negative. My goal was to train a model that could automatically classify these reviews. I loaded the data into training and testing sets. I also split the training data into a smaller 15,000 training set and a 10,000 validation set for my experiments so I could check the model's performance during training.

1.2 Preprocessing the Data

Since a computer cannot process words directly, I needed to turn the words in the reviews to numbers before I could train.

- **Limiting Words:**
  I only used the 10,000 most common words in the reviews to to simplify the task and make the model more efficient.
- **Turning Reviews into Vectors:**
  I converted each review into a series of 0's and 1's, which is also called vectorization. Each of the 10,000 words became a feature "1" if the word occurs in the review and "0" if it does not. I also converted the positive/negative labels to a numeric form.

This preprocessing was necessary as it allowed the neural network to process with the text data and learn patterns from it.

2. Experiments

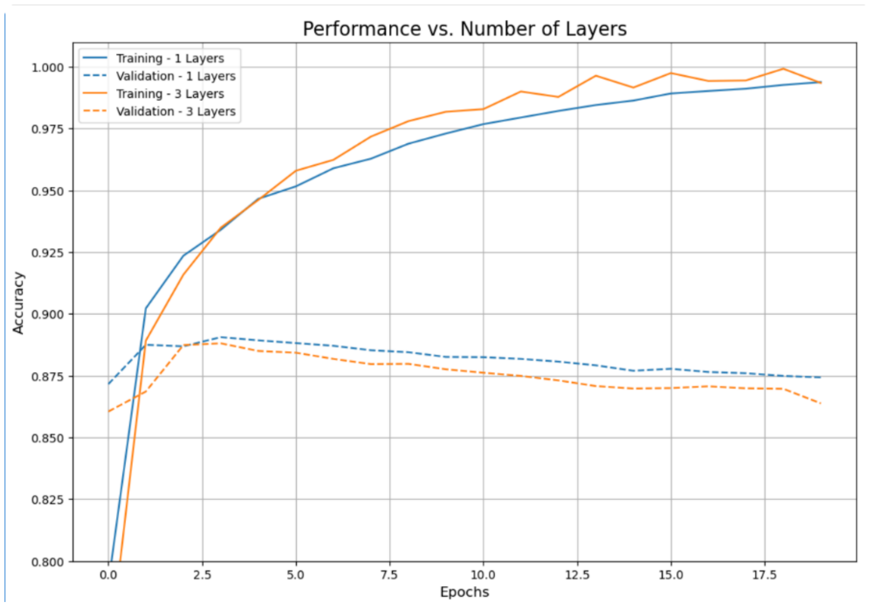
A baseline model was established with two hidden layers, using the ReLU activation function and binary_crossentropy loss.

- **ReLU (Rectified Linear Unit)**: A popular activation function that acts like a switch. It helps the network learn by turning any negative input value into zero while leaving positive values unchanged.

- **Binary Cross-Entropy**: A loss function designed for binary (yes/no) classification problems. It measures how different the model's predicted probability is from the actual label (0 or 1), guiding the model to make better predictions.

2.1 Number of Hidden Layers

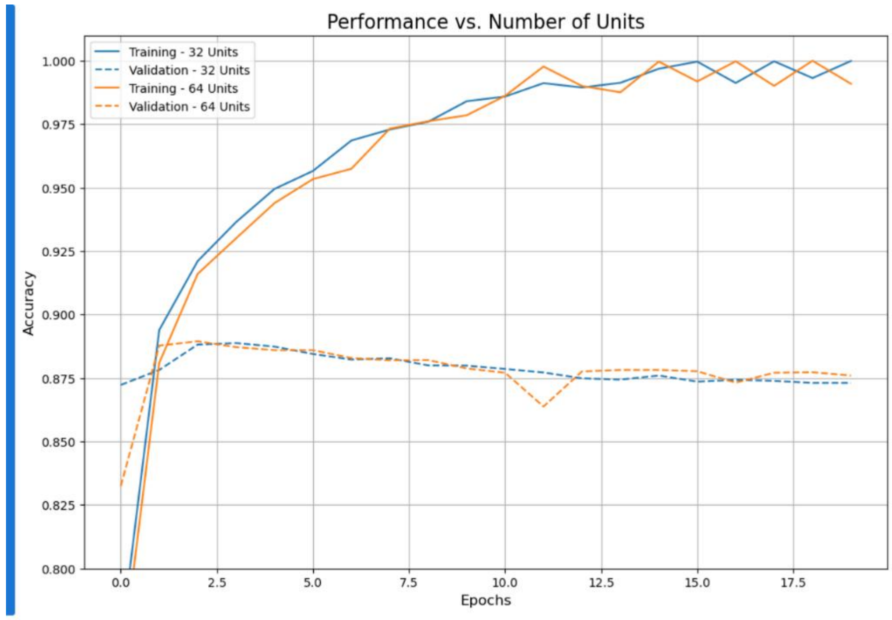
The baseline model uses two hidden layers. To see how network depth affects performance. I tested this network with one and three hidden layers.



- Both models learned the training data very well, as their training accuracy increased to nearly 100%. The 3-layer model learned slightly faster.
- Both models started to overfit after about 2-3 epochs, meaning they began memorizing the training data instead of learning general patterns.
- **The 1-layer model performed better**. Although its validation accuracy also decreased slightly over time, it was more stable and ended higher than the 3-layer model's accuracy, which dropped more significantly.

2.2 Number of Hidden Units

The capacity of the network was adjusted by testing layers with 32 and 64 hidden units.



- Both models were excellent at learning the training data and climbing towards 100% accuracy.
- The **64-unit model reached a slightly higher peak accuracy** early in the training. However, after this initial peak, both models performed very similarly.
- Both models began to overfit at a similar rate after the first few epochs. The 64-unit model showed a brief dip in performance around epoch 11 but recovered.

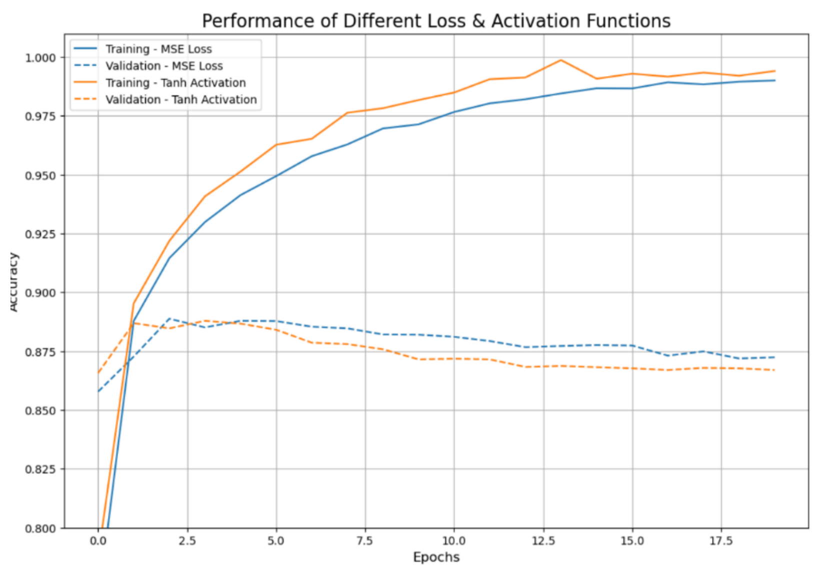
2.3 Loss Function and Activation Function

MSE (Mean Squared Error): A loss function that measures a model's average error by squaring the difference between predictions and actual values, heavily penalizing large mistakes.

- The accuracy of the MSE model's validation was ever so slightly poorer than your baseline models with binary_crossentropy.
- That's to say binary_crossentropy is the more appropriate loss function for this task, which is given that it's tailored for yes/no classification problems.

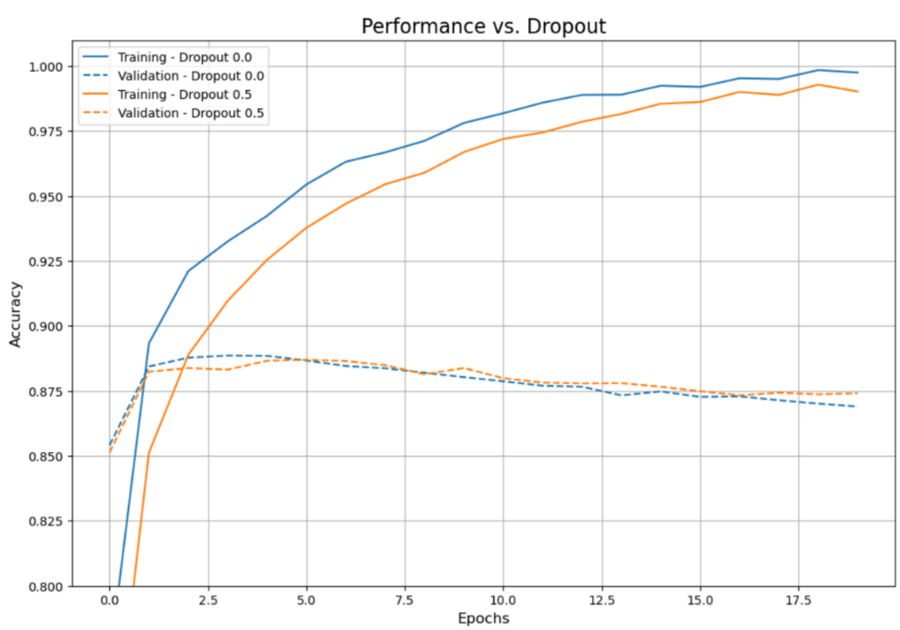
Tanh (Hyperbolic Tangent): An S-shaped activation function that squashes values in the range of -1 to 1, which helps to normalize the output of a neuron.

- Tanh model's validation accuracy (dashed orange line) was also lower and less stable than your baseline models' performance. It shows that **relu is a superior choice of activation function** for your network.



2.4 Regularization with Dropout

Dropout is a regularization technique used in neural networks to prevent **overfitting**.



The plot shows overfitting with the baseline model. While its training accuracy shot up, its validation peaked early and then started to drop.

In contrast, **the model with dropout of 0.5 was much more stable**. Its validation accuracy climbed to about 88% and then stayed very **consistent** for the rest of the training, without the major drop-off.

3. Results and Recommendations

The performance of each experimental model was tracked, with the final validation accuracy serving as the primary metric for comparison.

3.1 Summary Table

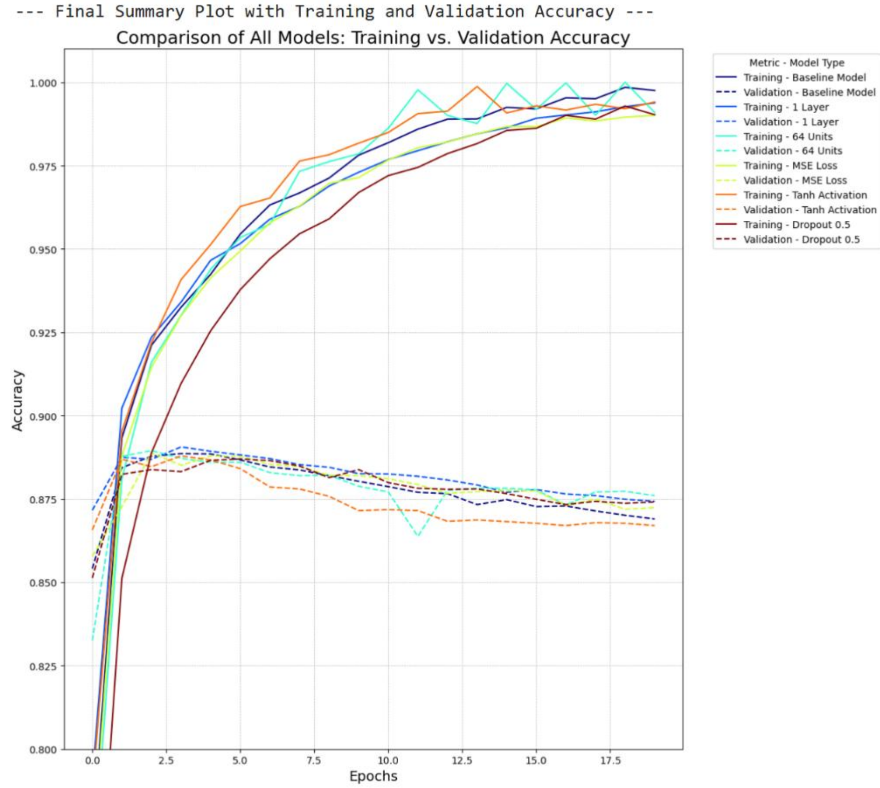
--- Summary of Experimental Results ---

	Experiment	Final Validation Accuracy
3	64 Units	0.8760
0	1 Layers	0.8743
7	Dropout 0.5	0.8741
2	32 Units	0.8731
4	MSE Loss	0.8724
6	Dropout 0.0	0.8690
5	Tanh Activation	0.8670
1	3 Layers	0.8638

64-unit model achieving the highest final validation accuracy of 87.60%

3.2 Key Findings

--- Final Summary Plot with Training and Validation Accuracy ---



- **Highest Performance**:
  The model with **64 hidden units** demonstrated one of the highest peak performances on the validation data in addition to having the best accuracy score.
- **Most Stable Performance**:
  The most reliable model towards overfitting was the one with a **50% dropout rate**. After peaking, the validation accuracies of other models decreased, but the dropout model's performance stayed steady and reliable.
- **Architecture**:
  A more straightforward **one-layer network** outperformed a better than three-layer network in terms of effectiveness. It was confirmed that more layers resulted in worse overfitting for this issue because the 3-layer model had the lowest final accuracy.
- **Baseline** :
  Because the MSE and Tanh models performed worse, the trials verified that the baseline selections of **binary_crossentropy loss and relu** activation were ideal.

3.3 Conclusion and Recommendation

- My conclusion is that the best model is a model that not only is high in performance but also stable. The table shows the 64-unit model performed highest, but the plot shows the dropout model was most consistent.

- Therefore, my final suggestion is a 64 hidden unit neural network with a 50% dropout. This weighs learning power and resistance to overfitting the best and would be the most efficient and trustworthy option for this task.

CITATIONS:

- Canvas Advanced Machine Learning Modules
- https://keras.io/api/datasets/imdb/
- https://www.geeksforgeeks.org/deep-learning/relu-activation-function-in-deep-learning/
- https://www.geeksforgeeks.org/deep-learning/binary-cross-entropy-log-loss-for-binary-classification/
- https://www.geeksforgeeks.org/deep-learning/tanh-activation-in-neural-network/
- https://www.geeksforgeeks.org/deep-learning/dropout-regularization-in-deep-learning/