Assignment-1: Neural Networks

Objective of the Work: The purpose of this assignment was to experiment with different neural network architectures using the IMDB movie reviews dataset. Starting from a simple baseline model, I tested how changing the number of hidden layers, units, activation functions, loss functions, and applying regularization (Dropout, L2) would affect the performance. The ultimate goal was to find which design gives the most balanced accuracy without overfitting or underfitting.

Dataset Used: The dataset is the IMDB Movie Reviews Dataset containing 50,000 reviews equally divided into training (25,000) and testing (25,000). Each review is labeled as either positive or negative, so this is a binary classification problem. Reviews were preprocessed into numeric format before training. For model evaluation, I also created a validation set from the training data.

Purpose of Experiments: The experiments in this assignment were designed to understand how different neural network configurations affect model performance on the IMDB dataset. Changing the number of hidden layers (1, 2, or 3) allowed us to study how model depth influences learning—whether a simpler or more complex architecture gives better accuracy. Adjusting the number of hidden units (16, 32, 64, etc.) tested how model capacity impacts results, since more neurons can capture complex patterns but may also cause overfitting. We also compared different loss functions—binary_crossentropy (commonly used for classification) versus mse (more typical for regression)—to see how optimization goals change outcomes. Similarly, trying different activation functions (ReLU and tanh) helped us analyze how non-linear transformations influence learning stability and convergence. Finally, we applied regularization techniques like Dropout and L2 to control overfitting, improve generalization, and make the models more reliable on unseen data. Overall, these experiments were necessary to identify which combination of design choices produces the most balanced and effective neural network.

Process: The neural network models were built using the standard IMDB codebase, and the following steps were followed in the model development process:

1. Import necessary libraries
2. Build the model
3. Compile the model
4. Prepare the validation set
5. Train the model
6. Retrain the model from scratch
7. Evaluate the model
8. Make predictions

Several neural network models were built using the standard IMDB codebase. First, the necessary libraries were imported, and the models were constructed and compiled. The IMDB dataset, containing movie reviews labeled as positive or negative, was prepared by converting the text reviews into numerical form so that the neural network could process them. The dataset was divided into three parts: a training set to teach the model, a validation set to monitor performance and adjust settings during training, and a test set to evaluate final accuracy on unseen data. Multiple models were designed by varying the number of hidden layers (1 to 3), the number of neurons per layer (16, 32, 64, or 128), activation functions like ReLU and Tanh, and loss functions such as Binary Cross-Entropy and MSE. Regularization techniques, including Dropout and L2, were applied to prevent overfitting. Each model was trained while tracking accuracy and loss for both training and validation sets, and graphs were plotted to visualize training loss versus validation loss and training accuracy versus validation accuracy. After training, the models were tested on the test set to measure performance on new data, providing insight into generalization and overfitting. Finally, results were compared using tables and graphs to assess how different configurations affected performance and to identify the most effective model.

Model Analysis: In this assignment, we followed a step-by-step approach to test how different neural network settings affect performance on the IMDB dataset. First, the text reviews were preprocessed and converted into numeric inputs so they could be fed into neural networks. The dataset was then divided into training, validation, and testing parts. Training data was used to teach the models, validation data was used to tune and check performance, and testing data was used for the final evaluation.

Next, we created multiple models by changing different settings:

- **Hidden layers:** 1, 2, or 3 layers.
- **Neurons (units):** 16, 32, 64, or 128.
- **Activation functions:** ReLU and Tanh.
- **Loss functions:** Binary Crossentropy and Mean Squared Error (MSE).
- **Regularization:** Dropout and L2 to reduce overfitting.

Each model was trained, validated, and tested. Accuracy and loss values were recorded for comparison. Finally, we analyzed the results through graphs and tables to identify which architecture worked best.

Why Some Models Did Not Become the Best

Not every model could be considered the "best" even if the performance appeared good. Some models became too complex, such as Models 3, 4, and 14, where adding extra hidden layers or more units increased size and training time without improving accuracy, making them inefficient. Overfitting was another issue, as seen in Model 7, where high training accuracy was paired with a significant drop in validation accuracy, indicating the model was memorizing training data rather than learning useful patterns, making it unreliable on new data. Over-regularization also affected performance, as in Model 11, where strong L2 regularization suppressed learning and reduced accuracy. Finally, activation function choices influenced results: models using Tanh, like Models 5, 6, 9, and 15, performed reasonably well and offered some balance, but ReLU-based models, especially when combined with Dropout, proved more consistent and reliable.

Good but Not the Best:

Some models gave strong and stable results but still could not become the very best:

- **Model 2 (1 layer, ReLU):** Excellent accuracy and no overfitting. A top contender, but still slightly weaker than Model 13 because it lacked dropout regularization.
- **Model 5 (Tanh + MSE):** Balanced and surprisingly good. However, Tanh was not as consistently strong as ReLU, so it could not be considered the best.
- **Model 8 (Dropout):** Showed strong performance due to dropout, but Model 13 achieved a slightly better balance.
- **Model 12 (Dropout):** Another strong model with good generalization, but accuracy was still a little lower than Model 13.
- **Model 15 (Tanh + Dropout):** Reliable and accurate, but again, Tanh made it less consistent compared to the ReLU-based best model.

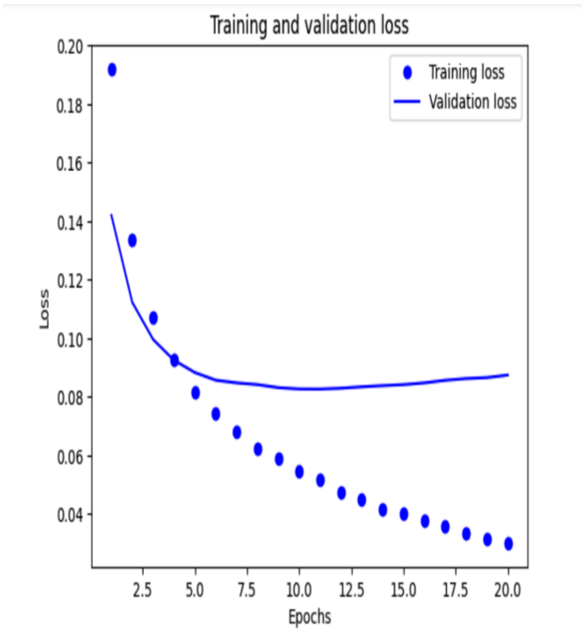
In summary, many models performed well and gave good accuracy, but the perfect balance of simplicity, regularization, and accuracy was only achieved by Model 13. That’s why it stands out as the best overall design.

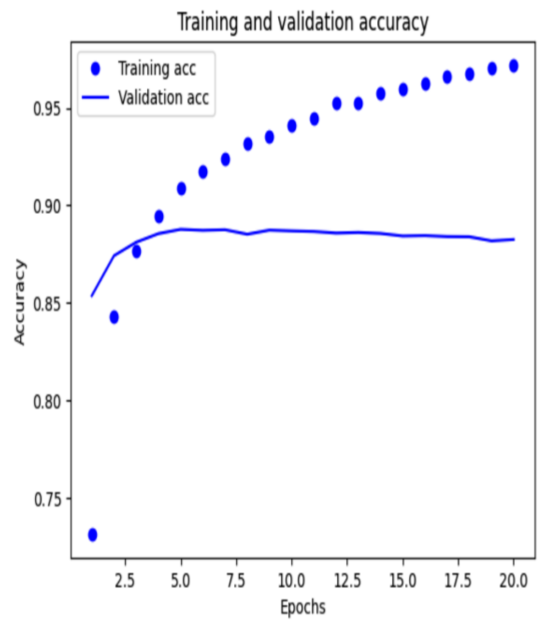
Final Results

Model No.	Layers	Units /Layers	Activation	Loss Function	Regularization	Validation Accuracy	Validation Loss	Testing Accuracy	Testing Loss
1	2	16	relu	Binary_Crossentropy	None	87.38%	0.3449	88.46%	0.2873
2	1	16	relu	Binary_Crossentropy	None	88.35%	0.2924	88.76%	0.2803
3	3	16	relu	Binary_Crossentropy	None	87.31%	0.3658	88.36%	0.2927
4	2	32	relu	Binary_Crossentropy	None	87.28%	0.3592	88.27%	0.2964
5	1	32	tanh	MSE	None	88.01%	0.0883	88.72%	0.0838
6	1	64	relu	Binary_Crossentropy	None	87.80%	0.3189	88.11%	0.2983
7	2	64	relu	Binary_Crossentropy	None	85.87%	0.4294	88.26%	0.2984
8	1	32	tanh	MSE	Dropout(0.5)	88.26%	0.0864	88.85%	0.0837
9	2	16	tanh	MSE	None	87.15%	0.1013	88.22%	0.0867
10	2	16	relu	MSE	None	87.81%	0.0906	88.51%	0.0852
11	1	16	relu	MSE	L2(0.01)	85.76%	0.1447	85.70%	0.1506
12	1	16	tanh	Binary_Crossentropy	Dropout(0.5)	87.50%	0.0975	88.56%	0.0851
13	1	16	relu	MSE	Dropout(0.5)	88.76%	0.0826	88.77%	0.0865
14	2	64	relu	Binary_Crossentropy	None	88.10%	0.0897	88.50%	0.0848
15	1	64	tanh	MSE	Dropout(0.5)	88.16%	0.0872	88.66%	0.0835

Best Model Performance Analysis - Model 13

Training and Validation Loss & Training and Validation Accuracy





Accuracy Analysis: The model's training accuracy steadily increased with each epoch. Validation accuracy also improved alongside and stabilized around 88.76%. The testing accuracy is almost the same (88.77%). Since the gap between training and validation is very small, it shows that the model is not overfitting and is generalizing well.

Loss Analysis: The training loss continuously decreased, and the validation loss also dropped and stabilized at 0.0826. The testing loss is very close (0.0865). This indicates that the model is stable, not memorizing the data, but instead learning useful patterns.

Model Design:

Layers: 1 hidden layer with 16 units

Activation: ReLU

Loss Function: MSE (Mean Squared Error)

Regularization: Dropout (0.5) applied → to avoid overfitting

Optimizer: RMSprop

This design is simple but effective. Dropout ensures that neurons do not become too dependent on each other, which improves generalization.

Interpretation:

- Both training and validation accuracy increased together and stabilized → the model is consistent.
- Dropout effectively controlled overfitting.
- Validation and testing accuracy are almost equal (**88.76% vs 88.77%**) → which proves the model performs equally well on unseen data.

Perspective:

High Accuracy (88%+): Can correctly classify customer reviews (positive/negative).

Generalization: Reliable performance on new, real-world data.

Efficiency: The architecture is simple and lightweight (only 16 units), making the model fast and scalable for large datasets.

Best Performing Model Analysis: Model 13 achieved the best results with a simple yet powerful design: one hidden layer of 16 units using ReLU activation, combined with Dropout (0.5) and MSE loss. This architecture struck the perfect balance, avoiding both overfitting and underfitting. The testing accuracy (88.77%) and validation accuracy (88.76%) were almost identical, proving that the model generalized extremely well. Dropout ensured that neurons did not become overly dependent on each other, while the lightweight structure made the model efficient and scalable.

Conclusion: The experiments with different neural network architectures on the

IMDB dataset showed that model performance depends heavily on design choices such as the number of layers, hidden units, activation functions, loss functions, and regularization methods. While many models performed well, not all were equally effective due to issues like overfitting, over-regularization, or unnecessary complexity. Among all tested models, Model 13 stood out as the best-performing design, achieving nearly identical validation and testing accuracies (88.76% and 88.77%). This close match proves that the model generalized very well to unseen data and did not suffer from overfitting. The simple architecture of one hidden layer with 16 units, ReLU activation, MSE loss, and Dropout regularization ensured both efficiency and reliability. Overall, the predictions confirmed that a balanced and lightweight model can achieve strong accuracy while remaining practical for real-world applications such as sentiment analysis of customer reviews.