ASSIGNMENT-1 Neural Networks

BA-64061-001

Advanced Machine Learning

Name: Sai Shripad Achutuni

Student ID: 811363462

E-Mail ID: sachutun@kent.edu

Assignment-1 Summary Report

IMDB Dataset Tuning for Neural Networks

Goal:

This assignment's goal is to use the IMDB movie reviews dataset to examine how various neural network topologies affect sentiment categorization. Model factors including the number of hidden layers, neurons per layer, activation functions, loss functions, and regularization strategies are all methodically experimented with. The study intends to determine the impact of these variables on model correctness, generalization, and convergence behavior by altering them over controlled trials. Additionally, by highlighting trade-offs between model performance and complexity, the trials assist prevent overfitting while preserving predictive value. The assignment is to provide the best neural network architecture for text-based sentiment classification tasks by means of comparative analysis.

Dataset Description:

Dataset Used:

IMDB Movie Reviews Dataset.

Task:

Classifying movie reviews into two categories, positive or negative sentiment.

Size and Dimensions:

- 25,000 samples for training
- 25,000 samples for testing

Data Preparation:

1. Limited to the 10,000 most frequently occurring words in the vocabulary.
2. Every review was transformed into a binary bag-of-words vector that showed whether terms were present or not.

Dataset Split:

- Training subset: 15,000 reviews
- Validation subset: 10,000 reviews
- Test subset: 25,000 reviews

Methodology:

1. Data Acquisition and Loading:

- The IMDB Movie Reviews dataset was directly imported from tensorflow.keras.datasets.imdb.
- Reviews were pre-labeled for binary sentiment (positive = 1, negative = 0).

2. Data Preprocessing:

- Limited the vocabulary to the 10,000 most frequent terms.
- Converted reviews into binary bag-of-words vectors (presence/absence of terms).
- Applied padding and truncation to ensure all input vectors had a uniform length.
- Split the dataset into training (15k), validation (10k), and test (25k) subsets.

3. Model Architecture Setup:

- Built baseline neural network models using TensorFlow/Keras.
- Standard architecture: Input → Hidden Dense Layers → Dropout (optional) → Output Sigmoid Layer.

4. Experimental Design:

- Designed five experiments, each focusing on a specific hyperparameter or architectural choice:
    - Hidden layer depth
    - Neuron count per layer

    - Loss function
    - Activation function
    - Dropout regularization

5. Training and Evaluation:

- Models were trained on the training set with validation monitoring.
- Early stopping was used where necessary to prevent overfitting.
- Performance was measured primarily using accuracy and validation loss, with final evaluation conducted on the test set.

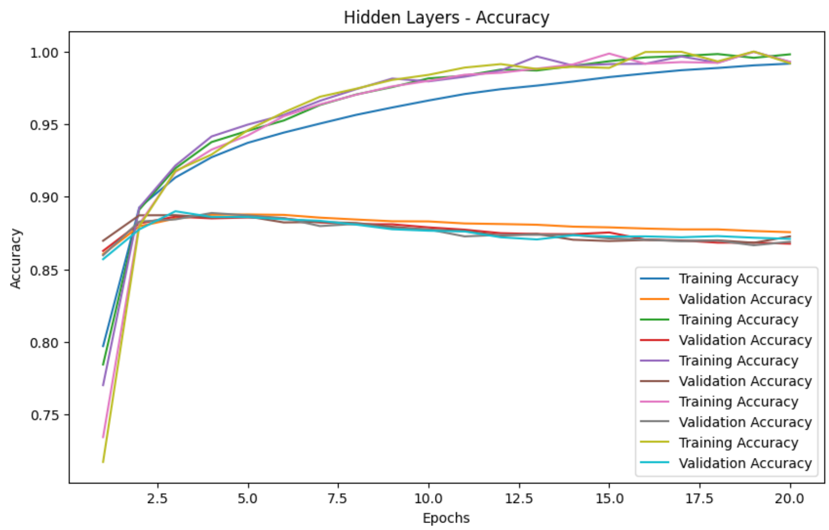
Experiments and Results:

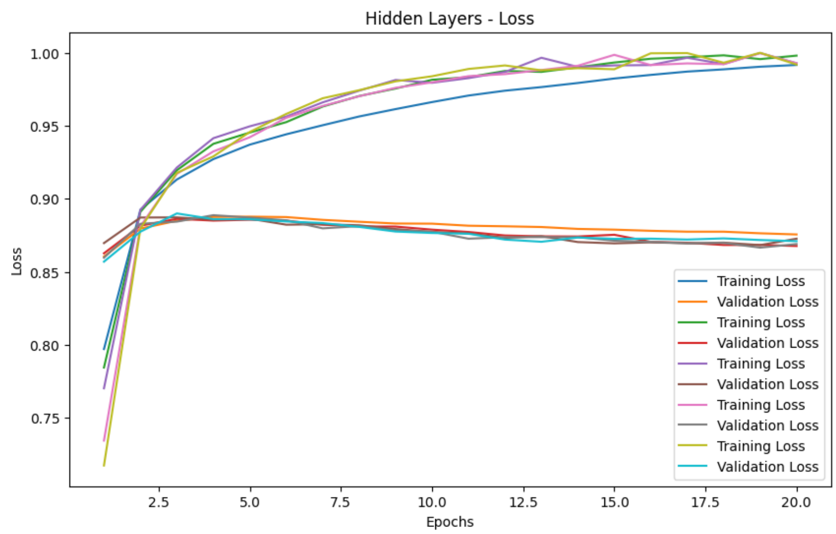
Experiment 1: Effect of Hidden Layers

Objective: To assess the impact of the number of hidden layers on the model's ability to train and perform well in classification.

Method: Built models with a set number of neurons in each of the 1–5 hidden layers. For a fair comparison, the same dataset split was used for training. tracked loss, validation accuracy, and training accuracy.

Results: Accuracy improved until 3 layers; deeper networks gave marginal gains and signs of overfitting.



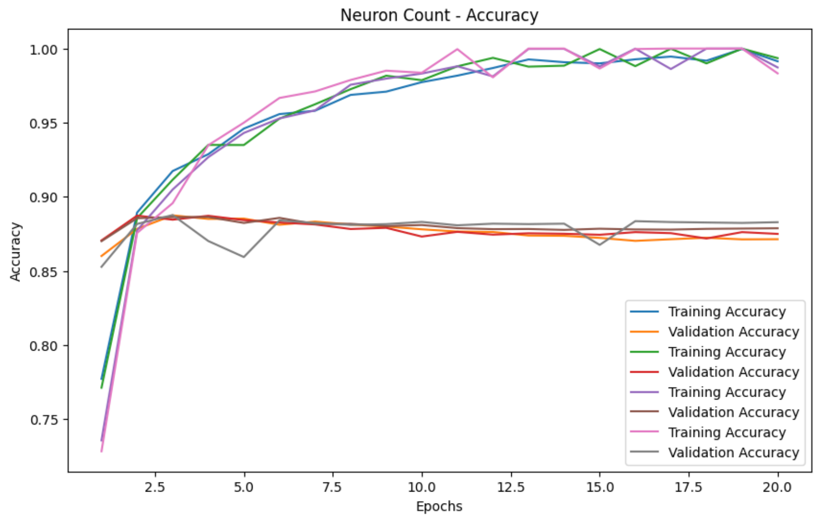


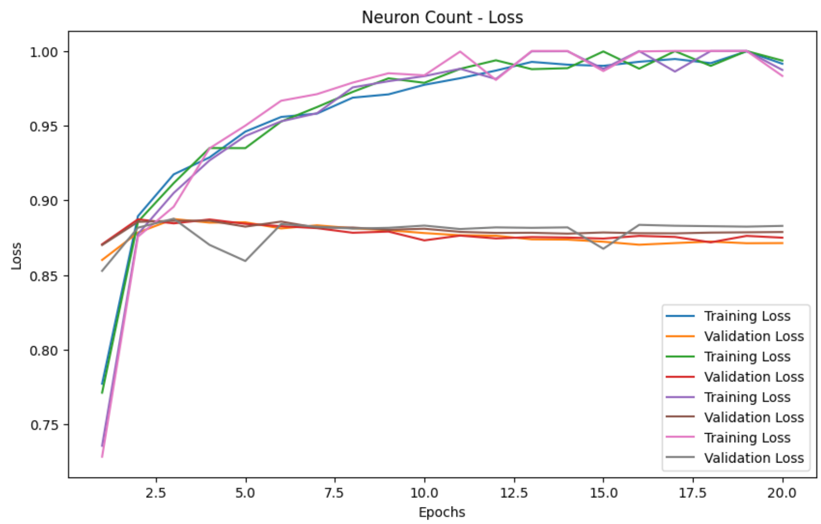
Experiment 2: Neuron Count per Layer

Objective: To ascertain how changing the hidden layer's neuron count affects the model's performance.

Method: 32, 64, 128 and 256 neurons per layer were used in the trained models. Other hyperparameters and the architecture remained unchanged. Monitored the correctness of validation and the rate of convergence.

Results: The best-performing models have 128–256 neurons, demonstrating a high capacity for learning. While extremely big networks (256 neurons) shown minor overfitting, very tiny networks (32 neurons) underfitted. The best-balanced setting was 128 neurons.



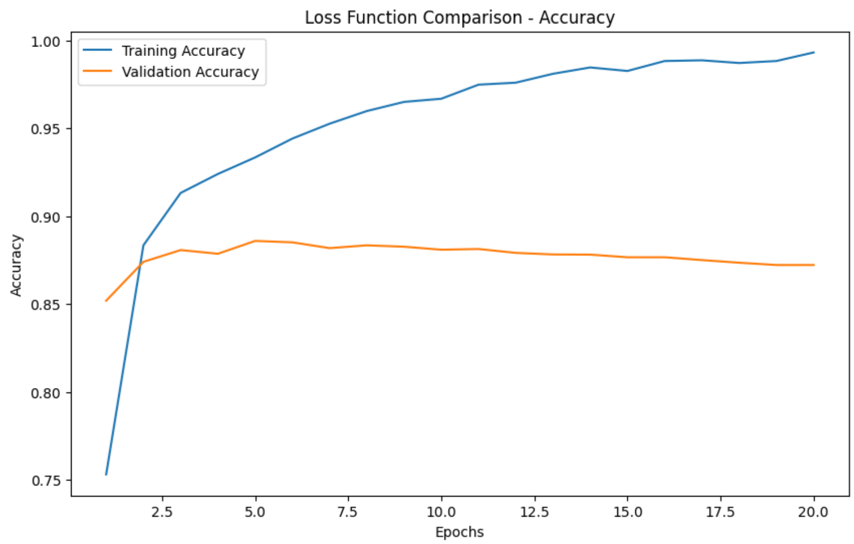


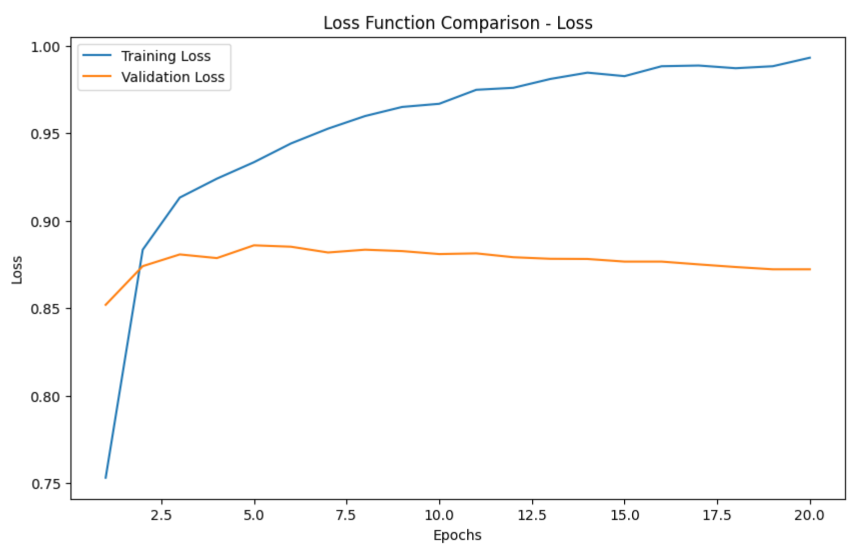
Experiment 3: Comparison of Loss Function

Objective: To evaluate the effects of various loss functions on classification accuracy and training dynamics.

Method: Binary Cross-Entropy (BCE) and Mean Squared Error (MSE) were used to implement the models. Both employ the same optimizer and network design. compared loss and accuracy throughout epochs.

Results: With greater accuracy and quicker convergence, Binary Cross-Entropy fared better than MSE. MSE resulted in less precise categorization and slower training. The best option for binary classification was determined to be BCE.



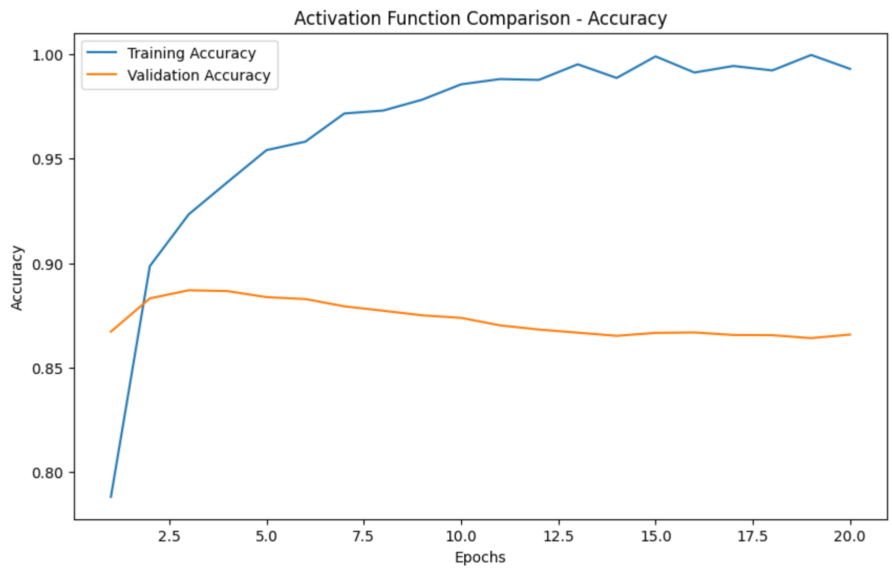


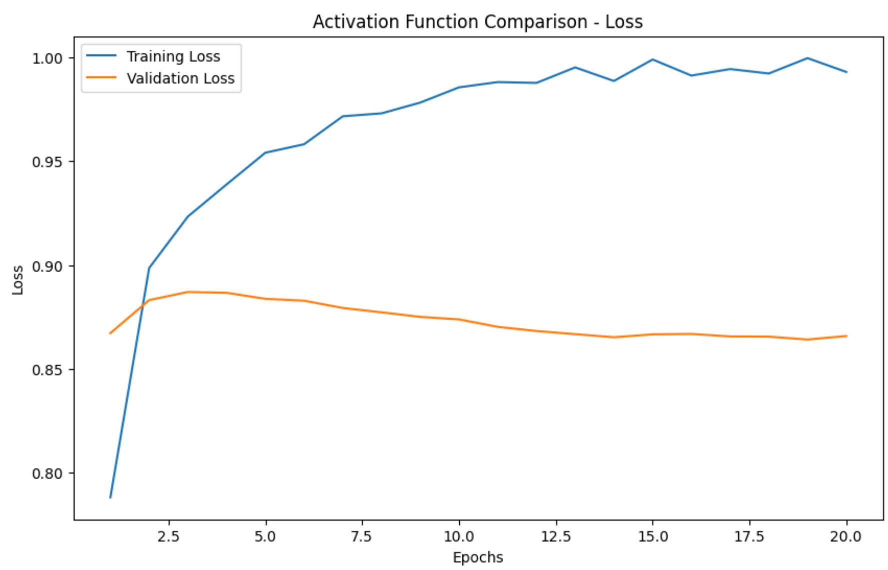
Experiment-4: Comparing the Activation Functions

Objective: Examine how various activation functions affect the performance of the hidden layer.

Methodology: ReLU and Tanh activations in hidden layers were used to build the models. To separate the activation influence, other factors were kept constant. compared final accuracy and learning speed.

Results: ReLU offered improved accuracy, stable gradients, and quicker training. Tanh did a respectable job, but the results were poorer and the convergence was delayed. The preferred activation function was determined to be ReLU.





Experiment 5: Regularization of Dropouts

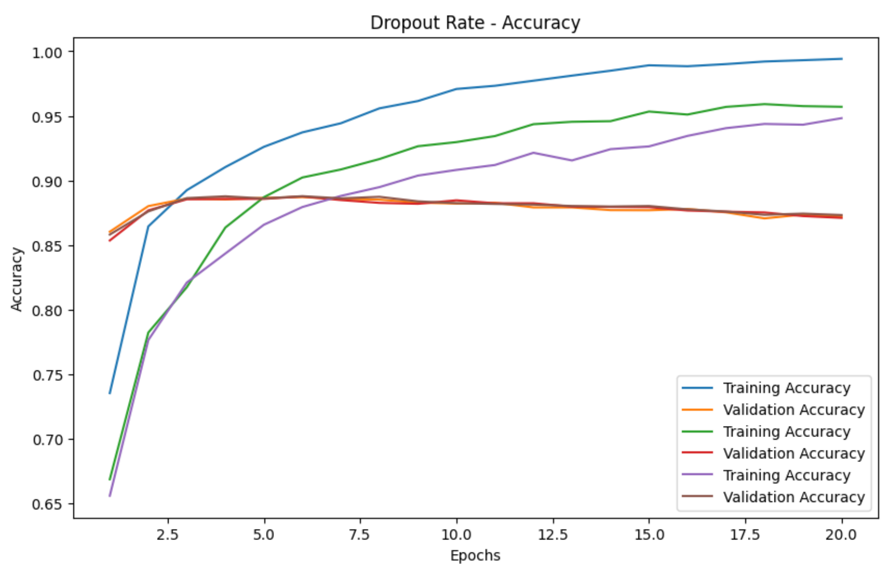
Objective: To evaluate how much dropout improves generalization and lessens overfitting.

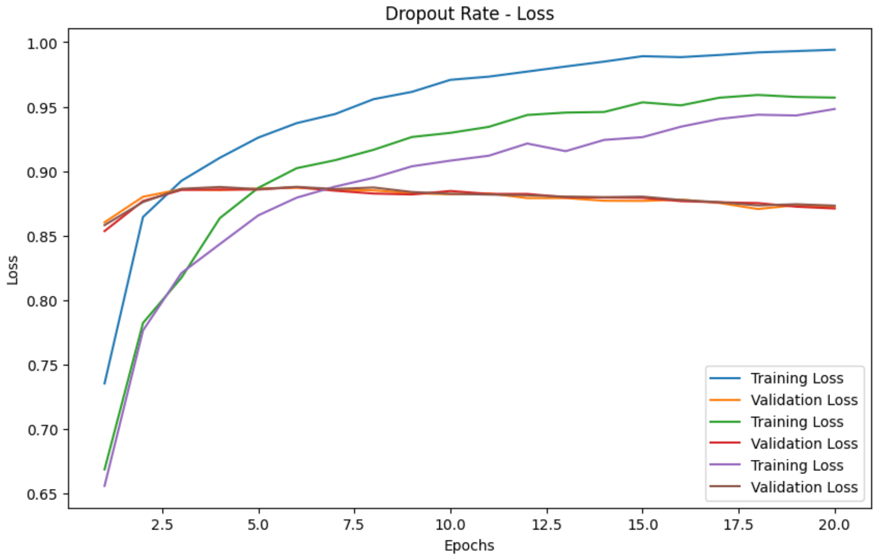
Method: Trained models with 0.3, 0.5, and 0.7 dropout rates. Overfitting behavior and validation accuracy was compared across settings.

Results:

The ideal balance was struck with dropout 0.5, which decreased overfitting without sacrificing precision. While 0.7 hampered learning ability, dropout 0.3 was insufficiently strong.

The suggested dropout rate for this job is 0.5.





Summary Table:

Experiment	Best Setting	Key Result
Hidden Layers	3 layers	Improved accuracy; deeper models began overfitting.
Neuron Count	128–256 neurons	Strong accuracy; 32 underfit, 256 slightly overfit.
Loss Function	Binary Cross-Entropy	Faster convergence, higher accuracy than MSE.
Activation Function	ReLU	Trained faster, achieved better accuracy than Tanh.
Dropout	0.5	Best balance; reduced overfitting without hurting learning.
Final Test	Dropout=0.5 model	**Test Loss: 0.5992, Test Accuracy: 0.8630**

Viewpoints:

1) Up to a certain point, adding hidden layers enhances learning; however, adding too

many layers adds complexity with no improvement in accuracy.

2) Representational capacity is directly affected by the number of neurons; too few neurons result in underfitting, whereas too many neurons might lead to overfitting.

3) Selecting the right loss function is essential:

Compared to MSE, binary cross-entropy is more compatible with binary sentiment categorization.

4) Tanh was constantly surpassed by ReLU activation, which provided more stable gradients and quicker convergence.

5) Generalization requires the use of dropout; a dropout rate of 0.5 was found to be ideal for striking a balance between regularization and performance.

6) All things considered, the optimal setup included 128 neurons, 3 hidden layers, ReLU, Binary Cross-Entropy, and dropout = 0.5.

Conclusion:

The study demonstrates that architectural and hyperparameter decisions have a significant impact on neural network performance in sentiment analysis. Accuracy was first increased by adding additional hidden layers and neurons, but overfitting resulted from excessive complexity. The best parameters were found to be Binary Cross-Entropy and ReLU activation, which allowed for quicker learning and superior classification outcomes. Dropout regularization effectively decreased overfitting while preserving high accuracy, especially at 0.5. Overall, three hidden layers, 128 neurons per layer, ReLU activation, Binary Cross-Entropy loss, and dropout of 0.5 were the best configuration for this challenge, which produced dependable and broadly applicable results on the IMDB dataset.