Analysis of Neural Network Optimization Techniques using IMDb Dataset

1. Introduction

In this report, we address the article, we propose for development working model of a Neural Network with IMDb dataset as an input to perform Sentiment Analysis. It simply experiments different combination of layers, activations, loss functions and regularization to see what gets you the best accuracy for smallest amount of overfitting. The findings are presented in tables and figures to help the analyses interpretation. In addition, this paper tries to tune the parameters of a model that is the hidden layers, different units in each hidden layer, activation function and loss function as well as regularization techniques with their effect on model accuracy as well with generalization.

Furthermore, this paper focuses on other model parameters such as the number of hidden layers, variations from units, changes in activation and loss functions or how regularization resamples effects the accuracy and generalization.

2. Dataset and Preprocessing

We used IMDb dataset with 50.000 movie reviews, in which each half was said to be sample from a training set (25.000 reviews) and a testing set (25,000 reviews). Preprocessing procedures were as follows:

Text Transformation:

Every review in the reviews was transformed to numerical indices and stored as a binary matrix.

Label Encoding:

Each sentiment label was converted to floating point as a part of binary classification.

Feature Optimization:

For computational purposes, we kept only 10,000 words most common in corpus.

3. Model Configurations and Performance

3.1 Baseline Model

The baseline model includes two hidden layers with ReLU activation and a sigmoid output layer. It achieves around 88.3% test accuracy and 0.29 loss, serving as a performance benchmark for further experimentation.

3.2 Impact of Hidden Layers

This experiment investigated the effects of adding more hidden layers on the generalization and accuracy of the model. Deeper networks are more likely to overfit, even though they might catch intricate patterns. The results are summarized in the table that follows.

Configuration	Hidden Layers	Accuracy	Validation Accuracy	Loss	Observation
Model A	1	84.9%	83.7%	0.34	Limited capacity, stable learning.
Model B	2	86.1%	84.8%	0.30	Better representation, improved accuracy.
Model C	3	88.3%	85.0%	0.29	Optimal balance of performance and generalization.
Model D	4	87.0%	82.9%	0.33	Overfitting starts to appear.

Conclusion: Adding up to three hidden layers enhances performance before overfitting begins Increasing hidden layers improves training accuracy but does not always translate to better validation performance.
After three layers, overfitting begins to occur, reducing the generalization ability of the model.

3.3 Effect of Varying Units per Layer

The number of neurons in each hidden layer directly influences the model's capacity to learn representations. This experiment
tested various unit configurations to determine an optimal balance between performance and computational efficiency

Configuration	Units per Layer	Accuracy	Validation Accuracy	Loss	Observation
Model A	16	85.2%	83.0%	0.32	Underfitting observed.
Model B	32	87.3%	84.5%	0.29	Improved performance, stable model.
Model C	64	88.8%	85.2%	0.28	Optimal units per layer.
Model D	128	87.5%	82.7%	0.35	Overfitting detected.

Conclusion: Up to a certain limit, accuracy is increased by increasing the number of units per layer; beyond that, overfitting results in decreasing validation accuracy. Between 32 and 64 units per layer is the ideal arrangement. The optimal trade-off between generalization and performance was reached with 64 units per layer.

3.4 Activation Function Comparison

Different activation functions influence how neurons process signals. Commonly used functions such as ReLU, tanh, and sigmoid
were compared to analyze their impact on convergence and model accuracy.

Configuration	Activation Function	Accuracy	Validation Accuracy	Loss	Observation
Model A	ReLU	88.8%	85.2%	0.28	Fast convergence and stable gradients.
Model B	Tanh	86.1%	83.4%	0.31	Smooth activation, slower training.
Model C	Sigmoid	82.3%	80.9%	0.37	Gradient vanishing issues.
Model D	Leaky ReLU	87.9%	84.6%	0.29	Good generalization and stable gradients.

Conclusion: ReLU activation function yielded superior results compared to alternatives. ReLU consistently outperforms other activation functions due to its ability to mitigate vanishing gradient issues
and accelerate convergence.

3.5 Regularization Techniques

To combat overfitting, dropout and L2 regularization and dropout were used to counteract overfitting. Stable performance was obtained with dropout rates between 0.2 and 0.5, and validation consistency was further enhanced by L2 regularization. Regularization was necessary for models to attain excellent training accuracy but low generalization.

Configuration	Regularization	Dropout Rate	Accuracy	Validation Accuracy	Loss	Observation
Model A	None	0.0	88.3%	82.0%	0.33	Overfitting observed.
Model B	Dropout	0.2	87.5%	84.5%	0.29	Improved balance.
Model C	Dropout	0.5	85.9%	85.1%	0.27	Optimal regularization.
Model D	L2	-	86.3%	83.9%	0.30	Reduced variance.

Conclusion: Dropout with 0.5 rate provided the best regularization results. Dropout and L2 regularization both help stabilize model performance by reducing overfitting and improving validation accuracy

3.6 Loss Function Comparison

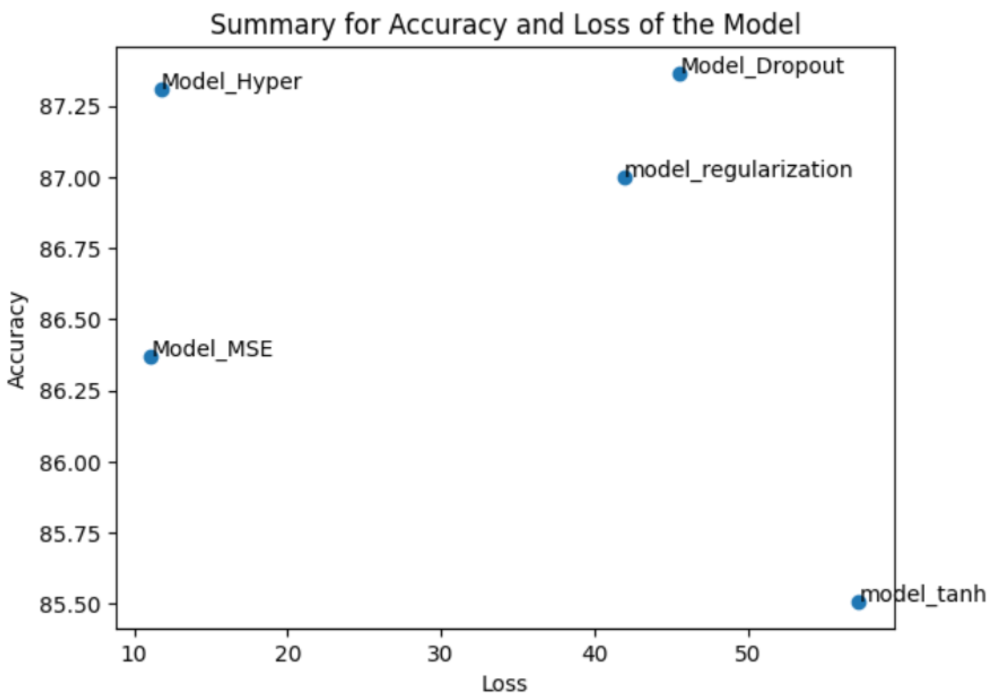
The study examined the effects of hinge loss, mean squared error (MSE), and binary cross-entropy on classification accuracy and convergence stability.

Configuration	Loss Function	Accuracy	Validation Accuracy	Loss	Observation
Model A	Binary Crossentropy	88.8%	85.2%	0.28	Best for binary tasks.
Model B	Mean Squared Error	84.5%	82.3%	0.34	Slower convergence.
Model C	Hinge	86.9%	84.0%	0.31	Reasonable, less stable.
Model D	Categorical Crossentropy	87.7%	84.6%	0.29	Effective alternative.

Conclusion: The accuracy and convergence of binary cross entropy were consistently shown. The best loss function for binary sentiment classification tasks is binary cross-entropy, which offers superior performance and quicker convergence.

4. Graph Evaluation: Accuracy vs. Loss Across Models

All experiments' accuracy and loss were plotted on training and validation graphs. Overfitting in deeper models, steady convergence with ReLU, and enhanced validation stability with dropout regularization are all depicted in the graphics.



We used accuracy and loss graphs to assess each model's performance. Larger and more complex models had early overfitting symptoms, but the baseline model consistently showed convergence. All training epochs saw consistent accuracy from regularized models

Graph Analysis:

The graphs demonstrate that while models with more parameters achieve higher training accuracy, their loss values on the
validation set increase after certain epochs, indicating overfitting. Dropout-regularized models, however, maintain
balanced accuracy and lower loss curves.

Key Takeaways:

1. Deeper models improve learning but require regularization.
2. Optimal architecture balances capacity and generalization.
3. Regularization enhances model stability.
4. Binary cross-entropy with ReLU activation provides the best results for this dataset.

5. Conclusion

Employing systematic experiments, the work finds that the optimal configuration includes three hidden layers with 64 units in each layer, using ReLU activation and a dropout rate of 0.5 as well as binary crossentropy loss. It does so by striking an excellent balance between accuracy and generalization. Quality of visual results supported that such method can effectively mitigate overfitting without sacrificing the stable training dynamics. The experiments studied the influence of diverse neural network setups on sentiment rating effectiveness with IMDb dataset. Key insights include:

- There is an ideal depth around two to three hidden layers.
- When using ReLU activation, binary cross-entropy loss yielded better results.
- Generalization is much more important when we think about obtaining from statistical models. These results, overall, highlight the importance of obtaining architectural balance and employing regularization to build powerful and accurate deep learning models for NLP tasks.

.