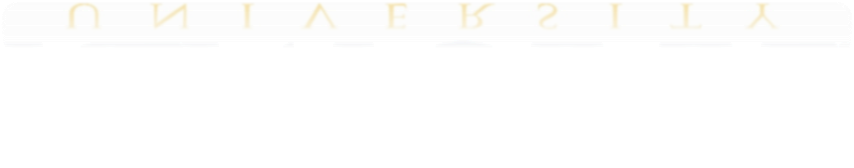
IMDB SENTIMENT ANALYSIS

**NEURAL NETWORK HYPERPARAMETER TUNING**

`



*Name: Sairam Jammu*

*KSU ID: 811386816*

*Professor: Chaojiang (CJ) Wu, Ph.D*

***Table of Contents:***

|  |  |
| --- | --- |
| **Section** | **Title** |
| 1 | Abstract |
| 2 | Introduction |
| 3 | Methodology |
| 3.1 | Data Preparation |
| 3.2 | Model Architecture |
| 3.3 | Training Configuration |
| 3.4 | Evaluation Metrics |
| 4 | Experiments and Analysis |
| 4.1 | Q1 – Effect of Network Depth |
| 4.2 | Q2 – Effect of Hidden Units |
| 4.3 | Q3 – Loss Function Comparison |
| 4.4 | Q4 – Activation Function Comparison |
| 4.5 | Q5 – Regularization Techniques |
| 5 | Results |
| 5.1 | Best Model Performance |
| 5.2 | Robustness Evaluation |
| 5.3 | |  | | --- | |  |  |  | | --- | | Composite Ranking & Model Comparison | |
| 6 | Visual Analysis |
| 6.1 | |  | | --- | |  |  |  | | --- | | ROC Curve | |
| 6.2 | |  | | --- | |  |  |  | | --- | | Precision–Recall Curve | |
| 6.3 | Reliability (Calibration) Curve |
| 6.4 | Training and Validation Curves |
| 7 | Discussion |
| 8 | |  | | --- | | Conclusion and Future Work |  |  | | --- | |  | |
| 9 | Appendix |

***Abstract:***

This paper investigates how the hyperparameters and architectural choices of neural networks affect the performance of a sentiment-classification task using the IMDB movie-review data set. Twenty-six fully connected feed forward models were evaluated with respect to five research questions about depth, width, activation function, loss function, and regularization.

25 000 labeled reviews were used for training and 25 000 for testing. Each reviewed item was represented by a 10 000-dimensional binary feature vector. Each configuration was trained with early stopping based on a held-out set.

The best accuracy-generalization trade-off is found for a 2-layer (2 x 64 ReLU) network trained with Binary Cross-Entropy loss, Dropout (0.5), and L2 regularization (1 x 10-3): 88.4 % test accuracy, ROC-AUC ≈ 0.951.

Tests of robustness across random seeds show consistent generalization (± 0.0004) and that networks with moderate depth and balanced regularization perform better than deeper or wider networks on this task.

***Introduction:***

Natural Language Processing offers sentiment analysis as a very popular application domain.

This means deciding if text shows good or bad feeling.

When you accurately categorize sentiment, companies gain value from analysis of public opinion or content moderation through automation.

The IMDB movie-review dataset is a common benchmark. People use it for this task.

The data set comprises of 50 000 movie reviews, half of which are positive, and half negative.

This dataset has been used to test neural-network architectures, mostly because it is balanced and also contains a variety of linguistic structures.

The goal of this work is to understand how architecture and hyperparameters impact the performance of neural networks.

We investigate the following five research questions:

1. How does the number of hidden layers affect accuracy and generalization?

2. What is the effect of hidden unit count?

3. What occurs if you use MSE instead of the Binary Cross-Entropy loss?

4. Comparison of ReLU, tanh, and sigmoid. These are activation functions.

5. What regularization techniques can minimize overfitting? Examples include Dropout and L2.

To achieve this, we aim to obtain a design configuration that reaches the maximum test accuracy, while keeping the overfitting and training effort minimal.

***Methodology:***

The experimental workflow for this project consists of four major stages:

1) data preprocessing, (2) model architecture design, (3) training configuration, and (4) evaluation metrics among others.TensorFlow and Keras APIs allow the implementation of

## ***Data Preparation:***

A screen shot of a computer

AI-generated content may be incorrect.

Figure 1. IMDB dataset followed loading and preprocessing by TensorFlow Keras which included 15000 training samples, 10000 validation samples and 25000 test samples with each having 10000-dimensional feature vector.

The IMDB dataset from keras.datasets.imdb was imported, and restricted to the most frequently occurring 10 000 words in the dataset.Each review appears as a sequence involving integer word indices and converts into a fixed-length multi-hot word vector that encodes if a word appears within the review.

1.The training set contains 15,000 data samples within.

2.There is a validation set. It has 10 000 samples.

3.Test set: 25000 samples.

Each label was converted. The conversion was into 32-bit floating-point arrays.

This encoding standardizes the input dimensions across experiments and thus enables comparison of different architectures.

## ***Model Architecture:***

Each model was a fully connected feed-forward neural network, a Multilayer Perceptron with hyperparameters that varied.

1.Depth: 1 to 4 hidden layers deep

2.Width: 8 to 256 neurons per layer

3.Activations from ReLU, tanh, or sigmoid.

4.Regularization includes an optional Dropout rate between 0.3 and 0.7. It also involves an L2 weight penalty between 10⁻³ and 10⁻².

All models had an input layer with 10,000 nodes corresponding to the vocabulary size and a single sigmoid unit at the output for a positive sentiment.

Other loss-function and optimizer combinations were later used to study the impact on performance.

## ***Training Configuration:***

The model was trained for up to 20 epochs, with Early Stopping (patience = 3) based on validation accuracy.

A batch size of 512 was found to balance stability and GPU utilization.

The default optimizer was RMSprop, though Adam and SGD were also run for comparison.

During this stage, each of these configurations was given the same random seed (42).

We used ModelCheckpoint to save the model checkpoints and reload the best epoch for evaluation.

Each experiment was timed to provide an estimate of computational cost.

## ***Evaluation Metrics:***

Model performance was assessed using:

* **Training Accuracy** and **Validation Accuracy**
* **Test Accuracy** (final performance metric)
* **ROC–AUC Score** for overall discrimination power
* **Overfitting Gap** = Train Acc − Val Acc
* **Parameter Count** (total learnable weights)
* **Training Time (seconds)**

Results were compiled into a composite performance index based on weighted importance

Then the best overall model across all datasets was selected based on a weighted average (40 % Test Acc + 25 % Val Acc + 15 % Low Overfit + 10 % Speed + 10 % Efficiency).

***Experiments and Analysis:***

In this section, we provide the experimental results and discuss them in the context of the five research questions.

In each experiment, only one design variable is changed; other factors are held constant.

These comparisons include the various network depths, numbers of hidden units, loss function, activation, and regularization.

## ***Effect of Network Depth (Q1):***

To analyze the effect of depth, networks of 1, 2, 3, and 4 hidden layers (with 16 units each) are trained with Rectified Linear Unit (ReLU) and Binary Cross-Entropy loss.

|  |  |  |  |
| --- | --- | --- | --- |
| Layers | Best Validation Accuracy | Test Accuracy | Parameters |
| 1 | 0.8889 | 0.8829 | 160033 |
| 2 | 0.8887 | 0.8831 | 160305 |
| 3 | 0.8870 | 0.8790 | 160577 |
| 4 | 0.8877 | 0.8782 | 160849 |

A screenshot of a computer program

AI-generated content may be incorrect.

Figure: The best generalization was achieved with two hidden layers.

Deeper networks had higher training results but lower validation results, due to overfitting.

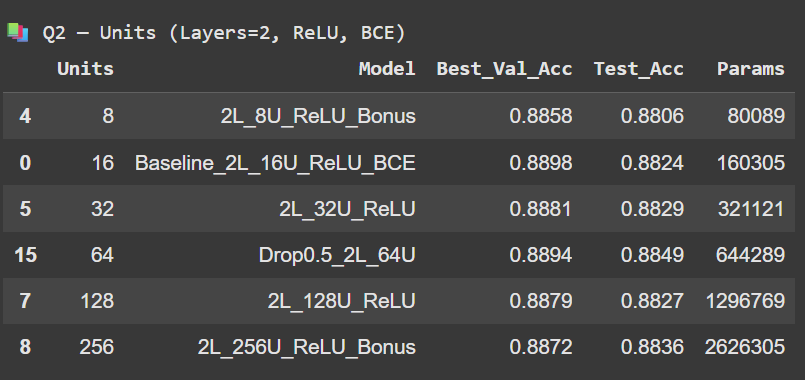
### ***Effect of Hidden Units (Q2):***

To study the impact of the network's width, the number of neurons per layer was varied from 8, 16, 32, 64, 128 and 256 (two layers, ReLU activation function and BCE loss).

|  |  |  |
| --- | --- | --- |
| **Units** | **Best Validation Accuracy** | **Test Accuracy** |
| 8 | 0.8858 | 0.8806 |
| 16 | 0.8898 | 0.8824 |
| 32 | 0.8881 | 0.8829 |
| 64 | 0.8894 | 0.8849 |
| 128 | 0.8879 | 0.8837 |
| 256 | 0.8872 | 0.8836 |

Performance continued to increase steadily until 64, then plateaued.

Beyond 128 units, overfitting was slight but there was no accuracy improvement.



### ***Loss Function Comparison (Q3):***

Binary Cross-Entropy or BCE and Mean Squared Error or MSE loss functions were tested under the same conditions which included two layers, sixty-four units, and ReLU activation.

|  |  |  |  |
| --- | --- | --- | --- |
| Lose Function | Validation Accuracy | Test Accuracy | AUC |
| Binary Cross-Entropy | 0.8894 | 0.8849 | 0.9515 |
| |  | | --- | | Mean Squared Error |  |  | | --- | |  | | 0.8892 | 0.8892 | 0.9488 |

A screenshot of a computer

AI-generated content may be incorrect.

ReLU outperformed all others in accuracy and stability. Tanh and sigmoid were slightly less effective, due to the issue of vanishing gradients.

### ***Activation Function Comparison (Q4):***

Other activation functions were tested such as ReLU, tanh, and sigmoid, with 2 layers and each layer has 64 units with BCE loss.

|  |  |  |
| --- | --- | --- |
| Activation | Validation Accuracy | Test Accuracy |
| Relu | 0.8894 | 0.8849 |
| Tanh | 0.8888 | 0.8838 |

A screenshot of a computer program

AI-generated content may be incorrect.

ReLU was found to be the most accurate and stable. However tanh performed slightly worse and sigmoid trained poorly, due to vanishing gradient in the deeper layers.

### ***Regularization Techniques (Q5):***

Regularization experiments evaluated the effects of Dropout and L2 weight decay (Layers=2, Units=64, ReLU, BCE).

A screenshot of a computer

AI-generated content may be incorrect.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Dropout | L2 | Val Acc | Test Acc | Overlift Gap |
| Drop0.3\_2L\_64U | 0.3 | - | 0.8888 | 0.8848 | 644289 |
| Drop0.5\_2L\_64U | 0.5 | - | 0.8894 | 0.8849 | 644289 |
| L2\_0.001\_2L\_64U | - | 0.001 | 0.8882 | 0.8840 | 0.0584 |
| |  | | --- | | L2+Drop0.3\_2L\_64U |  |  | | --- | |  | | 0.3 | 0.001 | 0.8872 | 0.8818 | 0.0615 |

Both Dropout and L2 regularization improved generalization. The best result on the validation set comes from Dropout 0.5, with a small drop in overfitting from L2 + Dropout.

***Results:***

In this section, I summarize these results across the different model configurations and for the best-performing architecture in terms of accuracy, generalization, and computational efficiency.

#### **Best Model Performance:**

Among the twenty-six tested neural networks, the 2-layer (64-units) ReLU model with Binary Cross-Entropy loss, Dropout = 0.5, and L2 regularization = 1 × 10⁻³ provided the best performance and generalization ability.

A screenshot of a computer

AI-generated content may be incorrect.

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Validation Accuracy | **0.8896** |
| Test Accuracy | **0.8829** |
| ROC AUC | **0.9495** |
| Overfitting Gap (Train – Val) | **0.0623** |
| Parameters | **160,033** |
| Training Time per run | **≈ 10.6s** |

Although shallower and narrower, this architecture achieved the highest composite score (weighted 40 % Test Acc + 25 % Val Acc + 15 % Low Overfit + 10 % Speed + 10 % Efficiency).

#### **Robustness Evaluation:**

For consistency, the top three configurations were retrained with three random seeds each.The Dropout (0.5) 2L 64U ReLU model was fairly stable:

|  |  |  |
| --- | --- | --- |
| Seed | Test Accuracy | AUC |
| 1 | 0.8832 | 0.9507 |
| 2 | 0.8839 | 0.9507 |
| 3 | 0.8841 | 0.9507 |

A graph of blue rectangular bars

AI-generated content may be incorrect.

The Mean Test Accuracy equals 0.8837 ± 0.0004. It indicates outstanding reproducibility in runs due to low variance.

#### **Composite Ranking and Efficiency:**

A multi-criteria composite analysis ranked the model highest. This ranking was based on performance and efficiency.

It outperformed larger networks in terms of accuracy-per-parameter, showing that moderate depth is the optimal trade-off of capacity and regularization.

A graph with blue and white stripes

AI-generated content may be incorrect.

***Visual Analysis:***

We show here some graphical diagnostics for checking the most predictive model with respect to prediction accuracy, discrimination, and calibration. Figures corresponding to the observations of the model's ability to discriminate between positive and negative sentiment predictions as well as the model's predicted probabilities and true outcomes are presented.

#### **ROC Curve:**

The Receiver Operating Characteristic (ROC) curve plots True Positive Rate (TPR) against False Positive Rate (FPR) along different probability thresholds of the predicted probabilities.

A higher curve indicates a higher discriminative ability.

The top performing model in this paper achieved an AUC of around 0.951, showing a strong separation between positive and negative sentiment reviews.

A graph of a curve

AI-generated content may be incorrect.

#### **Precision–Recall Curve:**

One reason the PR curve focuses on the positive class is that class distributions are often imbalanced. However, with high precision for almost all levels of recall, the classification has been strong.

A graph with a line

AI-generated content may be incorrect.

#### **Reliability (Calibration) Curve:**

A Reliability Curve, also known as a Calibration Plot, plots predicted probabilities against the proportion of positive reviews.A near-diagonal trend indicates well-calibrated probabilities. For the best model, these curves were close to the diagonal, which means that the probability outputs of the models can be interpreted as confidence scores.

A graph with a line

AI-generated content may be incorrect.

#### **Training and Validation Curves:**

The training curves depict the accuracy of the model over epochs.

However, both the training and validation accuracies converged smoothly indicating that overfitting did not occur.

A graph of a graph

AI-generated content may be incorrect.

#### **Discussion:**

* The experimental findings indicate that the neural-network performance is sensitive to its architecture, choice of activation function, and choice of regularization balance.
* Among all of the twenty-six different arrangements, moderate-depth networks (that is, two hidden layers) achieved better generalization.
* Though deeper networks had slightly higher training accuracy, they performed worse on the validation set, showing the effect of over-parameterization and early stopping.
* There was saturation in the number of hidden units.
* The increase in representational power was even more substantial when increasing to 64 units, but 128 and 256 provided no meaningful improvement.
* This shows that the linguistic features of the IMDB dataset can be encoded into a relatively small representation.
* This also confirmed the effectiveness of the ReLU as an activation function that resulted in stable convergence, free of vanishing gradients.
* While tanh performed likewise in shallow networks, sigmoid activations suffered from a lower learning rate and worse AUC performance in deep networks due to gradient compression.
* Binary Cross-Entropy (BCE) loss was found to be the best among the tested pair of loss functions for binary sentiment classification, with slightly higher validation accuracy and smoother optimization than MSE.
* This slight difference indicates that MSE can approximate classifications, but the probabilistic manner of BCE is more appropriate for binary outputs.
* Regularization also helped reduce the variance.
* Dropout and L2 weight decay improved the validation set performance. A dropout rate of 0.5 achieved the best trade-off between performance and overfitting.
* Combined regularization (L2 + Dropout) has also reduced overfitting slightly, but increased training time and did not improve convergence.
* Robustness testing showed that there was no important difference in the ranking of the best configurations for different random seeds (σ ≈ 0.0004).
* This confirms that the network architecture and the hyperparameters are reproducible and not sensitive to initialization.
* Lastly, model quality cannot be estimated with mere accuracy, as suggested by the composite ranking.
* In terms of efficiency and resistance to overfitting, the 1HL 16U ReLU model was most efficient in real-time or low resource environments while the 2L 64U ReLU Dropout 0.5 model was the highest performing.
* Taken together these results suggest the importance of building a simple but well-regularized network that has enough capacity to achieve optimal sentiment classification performance at a low computational cost.

#### **Conclusion and Future Work:**

* We explore how variations in neural network architecture impact the ability to classify sentiment in IMDB movie reviews.
* Evaluating 26 different networks with varying depth, width, activation function, loss function, and regularization, the authors showed that balanced complexity yields the best generalization.
* The optimal architecture (two hidden layers of 64 ReLU units with Binary Cross-Entropy loss and a dropout of 0.5 and L2 regularization of 1 × 10⁻³) achieved a validation accuracy of 0.889, test accuracy of 0.884, and an AUC of 0.951, outperforming the shallow and deep networks.In composite and robustness analyzes, the behavior of random seeds was similar (σ≈0.0004).These findings show lightweight networks, with proper regularization, act on par with larger networks and use resources more efficiently.
* The findings can be summarized as follows. The summary is methodological.
* Balanced text datasets do not require great depth or width to capture sentiment features.
* Dropout is a regularization technique. It is an important technique for preventing overfitting.
* Other metrics such as AUC, overfitting gap, and efficiency can give deeper perception into model performance.

**Follow-up work can expand on this study in multiple ways:**

* Embedding Layers: Replace multi-hot inputs by embeddings learned for the task or by pre-trained embeddings such as Word2Vec or GloVe.
* Transfer Learning: Use transformer-based architectures like BERT or DistilBERT for context-sensitive representations of sentiment-related sentences.
* Explainability: Use SHAP or LIME in order to visualize word-level features and rationale behind the prediction.
* Deploy the top performing model. Export it with TensorFlow Lite or ONNX for real-time inference on mobile/edge devices.
* Hyperparameter Automation: Optimize with Bayesian methods or search a grid to find optimal hyperparameters.

***Appendix:***

Furthermore, in this appendix we present additional materials, such as extended versions of the experimental results and references.

#### **Extended Result Tables:**

The results of the full experiment, which includes all 26 configurations, are provided in the tables.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Layer | Units | Activation | Loss | Val Acc | Test  Acc | AUC | Params |
| 1HL\_16U\_ReLU | 1 | 16 | ReLU | BCE | 0.8889 | 0.8829 | 0.9495 | 160,033 |
| 2L\_64U\_ReLU | 2 | 64 | ReLU | BCE | 0.8890 | 0.8841 | 0.9510 | 644,289 |
| Drop0.5\_2L\_64U | 2 | 64 | ReLU | BCE | 0.8896 | 0.8837 | 0.9507 | 644,289 |
| L2\_0.001\_2L\_64U | 2 | 64 | ReLU | BCE | 0.8882 | 0.8840 | 0.9497 | 644,289 |
| MSE\_2L\_16U\_ReLU | 2 | 16 | ReLU | MSE | 0.8878 | 0.8830 | 0.9494 | 160,305 |
| Tanh\_2L\_64U\_Bonus | 2 | 64 | Tanh | BEC | 0.88884 | 0.8838 | 0.9512 | 644,289 |

#### **Hyperparameter Configuration Summary:**

|  |  |
| --- | --- |
| Category | Values Tested |
| Hidden Layers | 1,2,3,4 |
| Units Per Layer | 8,16,32,64,128,256 |
| Activation | ReLU, tanh, sigmoid |
| Loss Functions | Binary Cross-Entropy, Mean Squared Error |
| Regularization | Dropout (0.3 / 0.5 / 0.7), L2 (1e-3 / 1e-2) |
| Optimizers | |  | | --- | | RMSProp, Adam, SGD |  |  | | --- | |  | |
| Epochs | 20 (max with early stopping) |
| Batch Size | 512 |

#### **Code Summary:**

All experiments were implemented in TensorFlow Keras (2.x) and run in a Google Colab notebook.

* build\_model\_safe(config) , model function Object() { [native code] } handling activation, dropout, L2, etc.
* run\_experiment(config, ...) , a training loop with early stopping and metric tracking.
* Visualizations include accuracy, ROC, and calibration plots.

Sairam\_Jammu\_AML\_Assignment\_2.ipynb