AML Assignment – 2

IMDB Sentiment Classification — Hyperparameter Tuning Report

Executive Summary

This project creates small feed-forward neural networks on the IMDB dataset to output a positive or negative label for a review. The reviews are encoded as 10,000-dimensional multi-hot vectors. Many different architectures and training settings were tested in this process. The best validation result was from a 2-layer × 64-unit (ReLU, BCE) model (Val 0.8911, Test 0.8834). The best test result came from a 2-layer × 16-unit model with Dropout 0.5 (Val 0.8876, Test 0.8836), demonstrating that a smaller network had strong generalization performance. Going deeper than 2 layers did not provide improvement, while some added width and dropout generally provided the most consistent improvement.

Data and Representation

- **Dataset:** IMDB movie reviews (binary sentiment).
- **Input features:** Multi-hot bag-of-words over the top 10,000 tokens.
- **Shapes printed in notebook:** train-part (15,000 × 10,000), validation (10,000 × 10,000), test (25,000 × 10,000).
- **Rationale:** Multi-hot vectors are quick to compute and provide a strong baseline, but ignore order.

2) Training Setup

- Family of models: Small dense neural networks, ReLU in hidden layers, sigmoid in output layers.

- Loss / optimizer: Binary Cross-Entropy (BCE) with Adam (unless otherwise stated).

- Regularization: Also compared Dropout and L2.

- Early stopping: Monitored validation loss and the best weights were loaded.

- Environment (printed): TensorFlow 2.19.0, NumPy 2.0.2.

3) Model Experiments and Observations

Baseline (2×16, ReLU, BCE) : A two-layer compact network forms the baseline. Val 0.8890, Test 0.8810. Training stability with only a small amount of overfitting.

Shallower (1×16, ReLU, BCE) : The depth is reduced to test the same capacity. Val 0.8894, Test 0.8813. Nearly identical to the baseline in performance; one hidden layer is sufficient in representing this problem.

Deeper (3×16, ReLU, BCE) : Increased the depth to test for more non-linearity. Val 0.8882, Test 0.8782. A little overfitting on the test with a lower test accuracy on the multi-hot inputs.

Wider (2×32, ReLU, BCE) : Increase the width and kept the depth constant. Val 0.8898, Test 0.8825. A small constant improvement over the baseline.

Wider+ (2×64, ReLU, BCE) : Increased width again and took the validation score and test score. Val 0.8911, Test 0.8834. Highest validation score with the slightly higher test performance indicating a boost in capacity is a positive thing.

Narrow (2×8, ReLU, BCE) : Reduced capacity and gathered scores. Val 0.8898, Test 0.8804. Slight underfitting relative to other larger models.

Different loss (2×16, ReLU, MSE) : Changed the loss function from BCE to MSE. Val 0.8890, Test 0.8810. No advantage when changing to MSE; BCE is the appropriate loss function for probabilistic binary outputs.

Different activation (2×16, tanh, BCE) : Replaced RELU with Tanh for activation. Val 0.8875, Test 0.8814. A little worse performance with tanh the model likely slowed down in learning due to saturation in the tanh activation.

Added L2 regularization (λ=0.001) on 2×16, ReLU, BCE) : Added weight decay; supervised a stronger capacity decline. Val 0.8874, Test 0.8804. Lighter L2 decay may have preferred but the strength cut out useful capacity.

Dropout 0.5 on 2×16, ReLU, BCE : Apply dropout after hidden layers with pt 0.5. Val 0.8876, Test 0.8836. A float lower in deployment but the best test accuracy overall a drop out of 0.5 presumably increases the general capacity of the model the drop out demonstrates a small model has generalization variables.

4) Results Summary

Configuration	**Validation**	**Test**
2×16, ReLU, **BCE** (Baseline)	0.8890	0.8810
1×16, ReLU, BCE	0.8894	0.8813
3×16, ReLU, BCE	0.8882	0.8782
2×32, ReLU, BCE	0.8898	0.8825

Configuration	Validation	Test
2×64, ReLU, **BCE**	**0.8911**	0.8834
2×8, ReLU, BCE	0.8898	0.8804
2×16, ReLU, **MSE**	0.8890	0.8810
2×16, **tanh**, BCE	0.8875	0.8814
2×16, ReLU, BCE, **L2=0.001**	0.8874	0.8804
2×16, ReLU, **BCE + Dropout 0.5**	0.8876	**0.8836**

Differences of ~0.001–0.003 are small; close results should be treated as roughly tied without repeated runs.

5) Synthesis and Interpretation

• **Depth vs. width:** More depth did not help - three layers slightly overfit, while a modest increase in width yielded the most consistent benefit in multi-hot features**.**

• **Regularization:** A dropout of 0.5 produced the best test accuracy and didn't increase parameters, suggesting greater generalization with a compact model. The tested L2 weight decay was a bit too strong for this baseline.

• **Loss/activation:** BCE + ReLU remained the strongest default setting - neither MSE nor tanh improved the performance.

6) Limitations and Next Steps (concise, human-style)

- **Stability.** Single runs can mislead. Repeat 3–5 seeds and report **mean ± std**; note any rank flips.
- **Input representation.** Bag-of-words ignores word order. Try **Embedding + 1D-CNN** or a small **GRU/LSTM** to capture phrases and negation.
- **Hyperparameters.** Sweep **learning rate (3e-4…3e-3)**, **batch size (128…1024)**, **dropout (0.2…0.6)**. Keep early stopping; select by validation.
- **Regularization balance.** Pair **lighter L2 (1e-5…1e-4)** with dropout; stop increasing once validation ceases to improve.
- **Deployment concerns.** Prefer the small **dropout model** for latency/memory. Consider **pruning** or **quantization** if needed.

- **Evaluation hygiene.** Keep the test set untouched until the end. Use the same tokenizer across splits and stratify labels when splitting.
- **Calibration (if probabilities matter).** Check reliability; apply **temperature scaling** if predictions are miscalibrated.
- **Error review.** Inspect misclassified reviews (e.g., negation, sarcasm) to decide whether sequence models are warranted.

7) Reproducibility Notes

- **Environment:** TensorFlow **2.19.0**, NumPy **2.0.2** (as printed).
- **Data pipeline:** keras.datasets.imdb with num_words=10_000; custom multi-hot vectorization; identical preprocessing across splits.
- **Training control:** EarlyStopping on validation loss with best-weight restore.
- **Artifacts:** Console logs include per-model **best validation** and **test** metrics for comparison.

Conclusion

In the case of this multi-hot IMDB baseline, the results show that wider (but shallow) networks perform better than deeper networks. The 2×64 (ReLU, BCE) model obtained the highest validation score (0.8911) and achieved solid test performance (0.8834), so this would be the most straightforward choice when opting based on validation performance. The 2×16 model with Dropout 0.5 had the highest test accuracy (0.8836) with a reasonable size and performance, making it a phenomenal choice when the priority lies on generalization performance, as well as resources used. Overall, modest capacity plus dropout achieves the most balance for this feature representation.