Phase 2 Report: Overfitting and Model Comparison

Cstokes-ai

May 2025

1 Objective

The primary goal of Phase 2 was to intentionally overfit the dataset to observe the model's behavior and performance. Overfitting was achieved by training models on the entire dataset without a validation split, using small batch sizes, and increasing the model's complexity. The objective was to maximize accuracy and analyze the impact of model architecture and hyperparameters on overfitting. Despite these efforts, the maximum accuracy achieved was 92%, and further attempts to reach 100% accuracy were unsuccessful.

To better understand the results, a comparison was made between two models:

- A larger, high-capacity model (Cell 3): This model had more layers and neurons, designed to overfit the data.
- A smaller, simpler model (Last Cell): This model had fewer layers and neurons, providing a baseline for comparison.

2 Methodology

2.1 Dataset Preparation

The dataset was loaded from `loan_data.csv`. Features (`X_raw`) and the target variable (`y`) were separated. Numerical features were scaled using `StandardScaler`, and categorical features were one-hot encoded using `OneHotEncoder`. The preprocessed data (`X`) was used for training without splitting into training and validation sets to encourage overfitting.

2.2 Model Architectures

Larger Model (Cell 3):

- Architecture:

  - Input Layer: 128 neurons, ReLU activation
  - Hidden Layer 1: 64 neurons, ReLU activation
  - Hidden Layer 2: 32 neurons, ReLU activation
  - Output Layer: 1 neuron, Sigmoid activation
- Training:
  - Epochs: 10
  - Batch Size: 1 (to encourage overfitting)

Smaller Model (Last Cell):

- Architecture:
  - Input Layer: 16 neurons, ReLU activation
  - Hidden Layer 1: 8 neurons, ReLU activation
  - Output Layer: 1 neuron, Sigmoid activation
- Training:
  - Epochs: 15
  - Batch Size: 64 (to speed up training)

3 Results

3.1 Larger Model (Cell 3)

- Final Loss: 0.20
- Final Accuracy: 92%

Observations:

- Despite the high capacity of the model, it failed to achieve 100% accuracy.
- The model overfit the data to some extent, as evidenced by the high accuracy on the training set.
- Increasing the number of layers and neurons did not significantly improve accuracy beyond 92%.

3.2 Smaller Model (Last Cell)

- Final Validation Loss: 0.25
- Final Validation Accuracy: 91%

Observations:

- The smaller model achieved comparable accuracy to the larger model, despite having fewer layers and neurons.
- The simpler architecture was more efficient and trained faster due to the larger batch size and fewer parameters.
- The smaller model's performance suggests that the dataset's complexity may not require a high-capacity model.

4 Comparison and Insights

Metric	Larger Model (Cell 3)	Smaller Model (Last Cell)
Architecture	128-64-32-1	16-8-1
Epochs	10	15
Batch Size	1	64
Final Loss	0.20	0.25
Final Accuracy	92%	91%
Training Time	Longer	Shorter

Key Insights:

- **Model Complexity:** The larger model did not significantly outperform the smaller model, indicating that the dataset's complexity does not necessitate a high-capacity model. The smaller model's comparable performance suggests that simpler architectures may suffice for this problem.
- **Overfitting:** Both models overfit the data to some extent, as expected. However, neither model achieved 100% accuracy, suggesting that the dataset contains inherent noise or features that limit perfect classification.
- **Efficiency:** The smaller model trained faster and required fewer resources, making it a more practical choice for deployment without sacrificing much accuracy.

5 Conclusion

In Phase 2, overfitting was intentionally induced to analyze model performance. Despite efforts to maximize accuracy, the highest achieved was 92%. The comparison between the larger and smaller models revealed that a simpler architecture can achieve nearly identical performance while being more efficient. This

suggests that further improvements may require better feature engineering or addressing potential noise in the dataset rather than increasing model complexity.