Evaluation and Analysis of Neural Network Models

Comparing a from-scratch NumPy implementation against a PyTorch implementation on the No-show Appointments dataset.

1 Convergence Time

Model	Total Training Time	Epochs to Converge
Scratch NN	518 seconds	5000 epochs
PyTorch NN	107 seconds	50 epochs

Table 1: Training time and convergence speed for both implementations.

2 Performance

Model	Accuracy	F1 Score	PR-AUC
Scratch NN	0.593	0.441	0.350
PyTorch NN	0.614	0.441	0.361

Table 2: Validation performance metrics.

3 Confusion Matrix

$$\begin{bmatrix} 9547 & 8095 \\ 911 & 3553 \end{bmatrix}$$

(a) Scratch NN

$$\begin{bmatrix} 10221 & 7421 \\ 1102 & 3361 \end{bmatrix}$$

(b) PyTorch NN

Table 3: Confusion matrices on the validation set. Rows represent true classes; columns represent predicted classes.

4 Analysis and Discussion

- **Convergence Speed:** The PyTorch model converged faster due to optimized C++ backend and GPU support. The scratch model uses Python loops and NumPy operations, which are less efficient.
- **Performance Differences:** Both models achieved similar accuracy, but the PyTorch implementation showed slightly higher F1 and PR-AUC, likely benefiting from stable numerical operations and built-in regularization utilities.
- **Framework Optimizations:** PyTorch reuses highly-tuned BLAS/LAPACK routines and fuses operations for better throughput. Manual implementations lack these low-level enhancements.
- **Numerical Stability:** Library functions in PyTorch often implement numerically stable activation function and loss, reducing the problem of gradient exploding or vanishing.