Report

1. Introduction

This task involves building a feedforward neural network from scratch using PyTorch to predict whether an individual's income exceeds $50,000 per year. Used the UCI Adult Income dataset, which includes demographic and employment features.

2. Data Preprocessing

2.1 Encoding & Normalization

- **Encoding**: One-hot encoding was used for all categorical variables.
- **Scaling Methods Compared**:
    - **StandardScaler**: Validation Accuracy = 84.54%
    - **MinMaxScaler**: **Best** with Validation Accuracy = **84.63%**
    - **RobustScaler**: Validation Accuracy = 81.16%

Conclusion: MinMaxScaler provided the best results, likely because it scaled features to a consistent range, benefiting the sigmoid output layer.

2.2 Dataset Split

- Train: 70%
- Validation: 15%
- Test: 15%

3. Model Architecture Ablation Study

Architectures Tried:

- [64] ReLU (Val Acc: 84.47%)
- [128] ReLU (Val Acc: **84.73%**, Best)
- [64, 32] ReLU (Val Acc: 84.73%)
- [128, 64] ReLU/Tanh/LeakyReLU (Val Acc: ~84.1-84.2%)
- [256, 128, 64] ReLU (Val Acc: 84.36%)

Conclusion: [128] with ReLU gave the best performance and lowest parameter count among top performers (4,481 params).

4. Training Configuration Ablation

4.1 Optimizer & Learning Rate

- **Adam (lr=0.001)**: 84.53%
- **RMSprop (lr=0.001)**: **84.58% (Best)**
- **SGD (lr=0.01)**: 82.69%
- **Adam (lr=0.0001/0.01)**: ~84.07%-84.44%

4.2 Loss Functions

- **BCELoss** with Sigmoid performed better than **BCEWithLogitsLoss**.

Conclusion: RMSprop + BCE loss + ReLU gave best convergence and generalization.

5. Regularization Study

Settings Tested:

Dropout	Weight Decay	BatchNorm	Val Acc
0.2	0.0	**True**	**84.78%**
0.2	0.0	False	84.73%
0.5	0.0	False	84.44%
0.7	0.0	False	84.22%

Conclusion: Light dropout (0.2) with BatchNorm performed best, improving generalization and mitigating overfitting.

6. Final Model Configuration

- **Preprocessing**: MinMaxScaler
- **Architecture**: [128], ReLU
- **Loss**: Binary Cross Entropy
- **Optimizer**: RMSprop, lr=0.001
- **Regularization**: Dropout=0.2, BatchNorm=True

Training Summary: - Early stopped at epoch 47 - Final Test Accuracy: **85.74%** - Precision: 0.7550 - Recall: 0.5978 - F1 Score: 0.6673 - Total Parameters: 4,737

7. Model Evaluation

Classification Report:

Class	Precision	Recall	F1-score	Support
<=50K	0.88	0.94	0.91	5574
>50K	0.76	0.60	0.67	1753
Accuracy			**0.86**	7327

Observations:

- High accuracy and good F1-score on majority class.
- Lower recall on minority class (>50K) indicates room for improvement in class balance handling.

8. Feature Importance Analysis

Using permutation importance, top features:

1. `education-num`
2. `capital-gain`
3. `age`
4. `marital-status_Married-civ-spouse`
5. `workclass_Self-emp-not-inc`

Conclusion: Educational attainment and capital-related features were most predictive of income.

9. Summary

Component	Best Option	Validation Accuracy
Preprocessing	MinMaxScaler	84.63%
Architecture	[128], ReLU	84.73%
Optimizer+Loss	RMSprop + BCE	84.58%
Regularization	Dropout=0.2 + BatchNorm	**84.78%**

Final Test Accuracy: **85.74%**

Key Insights:

- Small architectures (1 layer of 128 neurons) generalize well.
- MinMaxScaler helps due to Sigmoid output layer.
- Regularization is crucial to prevent overfitting.

- There’s a performance gap for the >50K class that warrants further fairness analysis.

10. Future Work

- Try SMOTE or weighted loss to address class imbalance
- Explore SHAP for local explanations
- Perform error analysis on misclassified >50K samples