Assignment 5: MLP for Adult Income Classification

Overview

This assignment builds upon Assignment #4 to design and train a Multi-Layer Perceptron (MLP) network for binary classification of the Adult Income dataset. The focus is on network architecture optimization and comprehensive performance evaluation.

Score: 90/100 ⭐

Objectives

- Design an MLP network for income classification
- Optimize network architecture and hyperparameters
- Evaluate model performance using multiple metrics
- Visualize training progress with learning rate history

Dataset

Adult Income Dataset: Based on preprocessed data from Assignment #4.

Task Requirements

Data Preparation

- **Skip Step 6** (train_test_split) from Assignment #4
- **Use Step 7** (StratifiedKFold) from Assignment #4
- Use the **first split** of training and test data for model training and validation

Feature Engineering

Determine and justify:

- Which features to use in the network
- Transformation methods for each feature:
  - Label encoding
  - One-hot encoding
  - Standardization

Model Training Configuration

Training Parameters

- **Epochs**: 500 epochs
- **Optimizer**: SGD (mandatory - do not use other optimizers)

Network Architecture Optimization

Find the best configuration by experimenting with:

- Number of neurons per layer
- Number of hidden layers
- Activation functions (ReLU, LeakyReLU, Tanh, etc.)
- Learning rate schedulers

Performance Evaluation

After training, display the following metrics on **test data**:

1. **Confusion Matrix**
2. **Accuracy**
3. **Precision**
4. **Sensitivity (Recall)**
5. **Specificity**
6. **F1 Score**
7. **AUC (Area Under Curve)**

Visualization Requirements

Plot accuracy and loss curves with learning rate history overlay:

- Training accuracy/loss over epochs
- Validation/test accuracy/loss over epochs
- Learning rate changes over epochs (overlaid)

Reference: See slide 23 from "07_PyTorch_Custom Dataset" lecture material.

Code Quality Standards

Required

- ✅ Clean and concise code
- ✅ Remove unnecessary code
- ✅ Show data shapes throughout
- ✅ Use provided template and fill all fields

Not Required

- ❌ Do not show intermediate results
- ❌ Keep only final outputs

Deliverables

- Jupyter notebook containing:
  - Data preprocessing and feature selection
  - MLP model architecture definition
  - Training loop for 500 epochs
  - All evaluation metrics
  - Accuracy/loss plots with learning rate history
  - Clear documentation of best configuration

Important:

- ✅ Include all results in the notebook
- ❌ Do NOT compress the notebook file
- ⚠️ **Plagiarism Warning**: Both copier and copied will receive 0 score with no second chances

Requirements

```python
torch
pandas
numpy
scikit-learn
matplotlib
seaborn  # for confusion matrix visualization
```

Installation

```bash
pip install torch pandas numpy scikit-learn matplotlib seaborn
```

Key Components

Model Architecture

Design considerations:

- Input layer size (based on selected features)
- Hidden layers (experiment with depth)
- Output layer (binary classification)
- Activation functions

Optimizer

```python
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)
```

Evaluation Metrics

```python
from sklearn.metrics import (
    confusion_matrix,
    accuracy_score,
    precision_score,
    recall_score,
    f1_score,
    roc_auc_score
)
```

For specificity, calculate manually:

```
Specificity = TN / (TN + FP)
```

Example Network Configurations to Try

1. **Baseline**: 2 hidden layers (128, 64) + ReLU + Constant LR
2. **Deep**: 3 hidden layers (256, 128, 64) + ReLU + StepLR
3. **Wide**: 2 hidden layers (512, 256) + LeakyReLU + ExponentialLR
4. **Custom**: Experiment with your own configuration

Results

Successfully designed and trained an MLP classifier achieving strong performance on the Adult Income dataset. Comprehensive evaluation metrics demonstrate model effectiveness in binary income classification.

Assignment completed as part of Deep Learning coursework