PA1 – Perceptron

Due: February 10, 2026 at 11:59 PM

Overview

In this assignment you will implement binary and multiclass perceptrons, apply simple feature engineering, and evaluate performance on the handwritten digits (8x8) dataset. All graded tasks are marked with TODO comments in `PA1_starter.ipynb`.

Learning goals

- Implement the perceptron update rule for binary and multiclass settings.
- Practice basic data preprocessing and evaluation metrics.
- See how feature mapping can make XOR linearly separable.
- Interpret training behavior using a learning curve.

Rules and allowed libraries

- Use only Python standard libraries for training logic.
- It is OK to use sklearn to load the dataset.

Deliverables

- Submit `PA1_starter.ipynb` with all TODOs completed.
- Do not rename the notebook (the autograder expects this filename).

Point values (100 total)

Question 0: Setup + Utils (10 points)

- 0.1 Load the dataset (`load_digits_data`) – 5 pts
- 0.2 Normalize features (`normalize_features`) – 5 pts

Question 1: Binary Perceptron (35 points)

- 1.1 Initialize parameters (`BinaryPerceptron.create`) – 5 pts
- 1.2 Score an example (`BinaryPerceptron.score`) – 5 pts
- 1.3 Predict a label (`BinaryPerceptron.predict`) – 5 pts
- 1.4 Update on a mistake (`BinaryPerceptron.update`) – 10 pts
- 1.5 Train over epochs (`BinaryPerceptron.fit`) – 10 pts

Question 2: Binary Classification Task (20 points)

- 2.1 Filter to two digits (`make_binary_dataset`) – 8 pts
- 2.2 Binary accuracy (`binary_accuracy`) – 6 pts
- 2.3a XOR dataset (`xor_dataset`) – 3 pts
- 2.3b XOR feature map (`map_xor_features`) – 3 pts

Question 3: Multiclass Perceptron (25 points)

- 3.1 Initialize parameters (`MulticlassPerceptron.create`) – 5 pts
- 3.2 Compute scores (`MulticlassPerceptron.scores`) – 5 pts
- 3.3 Predict class (`MulticlassPerceptron.predict`) – 5 pts
- 3.4 Update on a mistake (`MulticlassPerceptron.update`) – 5 pts
- 3.5 Train over epochs (`MulticlassPerceptron.fit`) – 5 pts

Question 4: Evaluation + Learning Curve (5 points)

- 4.1 Multiclass accuracy (`multiclass_accuracy`) – 2 pts
- 4.2 Mistakes per epoch (`train_binary_with_mistakes`) – 3 pts

Question 5: Main Experiment (5 points)

- Integration check (`main`) – 5 pts

Notes

- Train/test split (provided utility) and Part 4.3 (plotting) are provided.
- Question 5 is graded as a single integration check; parts 5.1–5.6 are not graded individually.

Submission checklist

- All TODOs are implemented.
- Notebook runs end-to-end without errors.
- You did not change function signatures.

Tips

- Start with the small utility functions (Question 0), then build the binary perceptron (Question 1) before moving to multiclass.
- For shuffling, keep X and y aligned by shuffling indices.
- Use small digit pairs (e.g., 0 vs 1) to sanity-check your binary model.