Assignment 4

April 17, 2025

Q1. AdaBoost

Implement the AdaBoost algorithm from scratch using decision stumps (i.e., decision trees of depth 1). Your implementation should include: [3]

- A training loop that updates sample weights based on misclassification
- Calculation of error using weighted 0-1 loss, and corresponding classifier weights β.
- A final prediction rule based on boosted classifier.

Evaluate your implementation on MNIST. Use classes 0 and 1. Use only 1000 samples per class for training. Test for full test set of both classes and report the accuracy. Apply PCA to reduce the dimension to 5. Provide the following plots:

1. Train/val/test loss as a function of boosting rounds.
2. Training error as a function of the number of boosting rounds

You can go upto 200 rounds.

Note: While obtaining a decision stump, you need to evaluate all possible cuts along each of the 5 dimensions. You can choose 3 cuts spaced uniformly between min and max of each dimension.

Q2. Gradient Boosting for Regression

Implement the gradient boosting algorithm for regression from scratch using decision stumps. You will experiment with two loss functions: squared loss and absolute loss. [3]

Dataset

Generate a synthetic dataset as follows:

- Let $x \sim \mathcal{U}[0, 1]$, sampled uniformly with 100 points.
- Let $y = \sin(2\pi x) + \cos(2\pi x) + \epsilon$, where ϵ is Gaussian noise with $\mathcal{N}(0, 0.01)$.
- Split the data into 80% training and 20% testing.

Your gradient boosting implementation should include:

- Computation of negative gradients for each loss function.
- Fitting decision stump on the negative gradients.
- A learning rate of $\eta = 0.01$.
- Final prediction formed by the sum of weak learners.

Provide the following:

1. Plot predictions and ground truth vs. iterations for both squared and absolute loss functions. This should be done for both train and test.
2. Plot training loss over iterations.

Note: While obtaining a decision stump, you need to evaluate all possible cuts along the given dimension. You can choose 20 cuts spaced uniformly between 0 and 1.

Q3. Neural Network for Binary Classification

Implement a simple feedforward neural network for binary classification with the following architecture: [3]

- **Input layer**: 2 input features
- **Hidden layer**: 1 neuron with sigmoid activation
- **Output layer**: 1 neuron no activation.

Use synthetic dataset: Label 0 is sampled as $X \sim \mathcal{N}(\begin{bmatrix} -1 \\ -1 \end{bmatrix}, I)$ and Label 1 is sampled as $X \sim \mathcal{N}(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, I)$. Assume 10 samples are obtained from each label. 50% is used for train and 50% for test.

1. Initialize all parameters randomly.

2. Train the network using gradient descent and a squared error loss. You need to show gradient descent step in the code.

3. Report the MSE over test samples.