Deep Learning

Homework 1: Neural Networks

Announced: October 13

Deadline: November 11 at 23:59 **(eeclass)**

TA:

邵柏翔(b99872599@gmail.com)

Introduction:

There is a saying that goes "Great oaks from little acorns grow." In this assignment, you need to build a **shallow neural network** as well as the mini-batch SGD training process on Fashion-MNIST dataset from scratch using Python3. The performance of your model should surpass 87% accuracy on testing data.

Rules:

★ **Built-in machine learning libraries (Sklearn, PyTorch, TensorFlow, Keras ……) are not allowed, you need to use basic mathematical operations in NumPy to define the behavior of each layer.**

★ Do not use testing data to train the shallow neural network.

★ Please properly comment your code to let us understand your train of thought.

★ Discussions are encouraged, but plagiarism is strictly prohibited and punishable!

Submission:

You should compress your code (.py), model weights (.npy), dataset, report (hw1report_studentID.pdf), and readme.txt (explain how to run your code) into a ZIP file (hw1_studentID.zip), and submit it on **eeclass** before the due date. Zero credit for the submission after the deadline.

We will reproduce your results based on the codes, weights, and readme file you gave. Please make sure we can reproduce the results, otherwise, you would not get the scores.

Implementation [50%]

1. Fashion-MNIST reading: (2.5%)
   Use NumPy to read ubyte.
2. Dense neural layer: (2.5%)
   Every output neuron has full connection to the input neurons.
3. ReLU layer: (2.5%)
   We add nonlinear activation functions after the neural layer using ReLU.

4. Softmax output: (2.5%)
   The final layer is typically a Softmax function which outputs the probability of a sample being in different classes.
5. Cross-entropy loss calculation: (5%)
   Use the output of a batch of data and their labels to calculate the CE loss.
6. Backward propagation: (5%)
   Propagate the error backward and update each parameter.
7. Testing accuracy: (30%)
   Reach 87% to get 30 points, 87~86% -> 20 points, 86~85% -> 10points, zero otherwise.

Report [50%]

The format is not limited, but the following matters must be discussed in your report:

1. Show the model architecture, implementation detail, and testing accuracy. Please describe the methods to achieve the final result in detail, e.g., epochs, batch size, etc. (10%)
2. During testing, the model might overfit training data to achieve poor performance. How do you check overfitting and underfitting during training? Please describe the methods and experiments to prove that. (10%)
3. During training, you need to adjust the hyperparameters (epochs, batch size, …) to achieve higher accuracy. How do you choose these hyperparameters? Please conduct experiments to show that these hyperparameters are suitable to achieve higher accuracy for the final test results. (10%)
4. If we use a very deep NN with a large number of neurons, will the accuracy increase? Why or why not? Please conduct experiments to prove that. (10%)
5. Please do experiments to compare at least other two activation functions in the model. Analyze the activation functions you used and show if these activation functions help improve the performance during testing. (10%)