Brandeis University | COSI-101A | Artificial Intelligence
Assignment 4 (100 points)
Due Date: Dec 10, Sunday, 11:59 PM (EST)

Instruction of homework submission

1. Submit your solutions in one pdf file named cosi101b-assignment-4-[yourname].
2. Submit your code files in one zip file named cosi101b-assignment-4-code-[yourname]. The code should include the necessary comments.
3. Late policy: (late hours / 24) * 10%, that is one day late leads to 10% loss. If late hours are > 7 * 24 (one week), it will be a 100% loss.
4. Similar problems may be available on the internet while you should finish on your own.
5. Please obey academic integrity, no plagiarism will be allowed. Plagiarize will lead to serious penalties, e.g., failure on the assignment or failure in the course.
6. For any questions, please get in touch with the TA (jiazhengli@brandeis.edu, wenxiaoxiao@brandeis.edu) or the instructor (chuxuzhang@brandeis.edu).

Problem A – Convolutional Neural Network (50 pts)

In this problem, you will build a convolutional neural network model for image classification (binary classification) using Pytorch (https://pytorch.org/). You will need to finish the code in cnn_main.py and cnn_utils.py, and answer some questions.

Four datasets (pkl format) in the datasets folder:
train_data_x.pkl: training data
train_data_y.pkl: label of training data
test_data_x.pkl: test data
test_data_y.pkl: label of test data

Part I: Model (30 pts)

(10 pts): Finish Net class for constructing CNN model in cnn_utils.py.
CNN model detail: 5 layers in total. The first 2 layers are convolution layers with kernel size = 5*5, stride = 1, and output channel numbers = 6, 12. The filter size of max pooling layer after each convolution layer is 2*2. The last 3 layers are fully connected layers with output hidden numbers = 120, 64, and 2, respectively. Use relu activation in each hidden layer and sigmoid activation in the last layer. Use default parameter initialization in Pytorch.

(10 pts): Finish model_train function for CNN model training (using train data) with Adam optimization in cnn_main.py. Set batch size = 5.

(10 pts): Finish model_test function for model testing (using test data) in cnn_main.py.

Part II: Performance (20 pts)

Run your code and report accuracy over test data (Hint: accuracy is over 80%, the result is usually not stable), and show a screenshot of your result.

Problem B – Graph Neural Network (50 pts)

In this problem, you will build a graph convolutional network for paper classification (7 classes) using Pytorch. You will need to finish the code in gnn_main.py and gnn_utils.py, and answer some questions.

Dataset in the datasets folder:
cora.content – paper information (paper id, paper word vector, paper label)
cora.cites – citation relationships of different papers (id of cited paper, id of citing paper)

Part I: Model (30 pts)
(10 pts): Finish GCN class and GraphConvolution class for constructing GCN model in gnn_utils.py. GCN model detail: 2 layers in total. The hidden size = 32. Use relu activation in the hidden layer, and set dropout ratio = 0.5.

(10 pts): Finish model_train function for GCN model training (using train data) with Adam optimization in gnn_main.py.

(10 pts): Finish model_test function for model testing (using test data) in gnn_main.py.

Part II: Performance (20 pts)
Run your code and report accuracy over test data (Hint: accuracy is over 80%), and show a screenshot of your result.