CSA 250: Deep Learning
(Spring 2020)

Sargur N. Srihari
Indian Institute of Science,
Bengaluru, Karnataka
email : srihari@buffalo.edu

January 28, 2020

1 Task

This project is to implement neural network and convolutional neural network for the task of classification. The classification task will be that of recognizing an image and identify it as one of ten classes. You are required to train the classifiers using Fashion-MNIST clothing images. Following are the two tasks to be performed:

1. Build multi-layer Neural Network with open-source neural-network library (pytorch/tensorflow) on Fashion-MNIST dataset.
2. Build Convolutional Neural Network with open-source neural-network library (pytorch/tensorflow) on Fashion-MNIST dataset.

2 Dataset

For training and testing of our classifiers, we will use the Fashion-MNIST dataset. The Fashion-MNIST is a dataset of Zalando's article images, consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes.

Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255. The training and test data sets have 785 columns. The first column consists of the class labels (see above), and represents the article of clothing. The rest of the columns contain the pixel-values of the associated image.

Each training and test example is assigned to one of the labels as shown in table 1.

Figure 1: Example of how the data looks like.

1	T-shirt/top
2	Trouser
3	Pullover
4	Dress
5	Coat
6	Sandal
7	Shirt
8	Sneaker
9	Bag
10	Ankle Boot

Table 1: Labels for Fashion-MNIST dataset

You can simply load the Fashion MNIST dataset using pytorch/tensorflow API. The details can be found in `https://github.com/zalandoresearch/fashion-mnist`. The pytorch API to download the dataset is at `https://pytorch.org/docs/master/torchvision/datasets.html#fashion-mnist` and the tensorflow API can be found at `https://www.tensorflow.org/api_docs/python/tf/keras/datasets/fashion_mnist`.

3 Plan of Work

1. **Train using Multi-Layer Neural Network** with high level Neural Network library.
2. **Train using Convolution Neural Network** with high level Neural Network library
3. **Tune hyper-parameters:** For steps 1 and 2: Validate the classification performance of your model on the validation set. Change your hyper-parameters and repeat the step. Try to find what values those hyperparameters should take so as to give better performance on the testing set.
4. **Test your machine learning scheme on the testing set:** For steps 1 and 2: After tuning the hyper-parameters, fix your hyper-parameters and model parameter and test your model's per-

formance on the testing set. This shows the ultimate effectiveness of your model's generalization power gained by learning.

4 Evaluation

1. Plot graph of training loss vs number of epochs while training on each classifier.
2. For each classifier evaluate solution on the test set using classification accuracy:

$$Accuracy = \frac{N_{correct}}{N} \tag{1}$$

   Where $N_{correct}$ is the number of corrected classified data samples, and N is the total number of samples of the validation set.

3. Construct a confusion matrix for each classifier and observe the relative strengths and weaknesses.

5 Deliverables

The following documents/codes must be submitted:

1. Prepare a report and name it **Deep Learning Report 2.pdf**. In your report, you need to briefly describe what you have done, present the results (in a form you think is good) and provide a brief discussion of the results you obtained. Please feel free to play with the model architecture and hyper-parameter settings. Please make it concise and to-the-point.
2. You need to save the trained model in a specific format and put it in model directory. During test time, your code should use this saved model and not retrain it from scratch.
3. The python code which is used for training.
4. On running the file **main.py**, your model should generate the classes for all the images in the test data (included with the dataset API) and output them to files titled **multi-layer-net.txt** and **convolution-neural-net.txt**. Sample output files will be updated through piazza.

The submission instructions will be provided in a separate document at a later stage. Please check piazza periodically for updates.

The deadline for the submission is **February 19, 2020**.