Introduction to ML
Exercise 4

Due Date: January 4th 22:00, 2021

Yosi Shrem Yael Segal and Joseph Keshet

December 21, 2020

Guidelines

1. You are allowed to work in pairs.
2. You are allowed to use numpy and scipy packages and PyTorch framework.
3. Technical questions about this exercise should be asked at the course' piazza or during the TIRGUL.
4. Personal issues regarding the deadline should be directed to **Yael Segal**.
5. In order to submit your solution please submit the following files:
   (a) `details.txt` - A text file with your full name (in the first line) and ID (in the second line).
   (b) `ex_4.py` - A python 3.6+ file that contains your main function (attach ANY additional files needed for your code to run).
   (c) `ex_4_report.pdf` - A pdf file in which you describe your model and parameters.
   (d) `test_y` - your model's predictions on the given test set (see instructions below).

Follow the instructions and submit all files needed for your code to run.

Good Luck!

Ex4

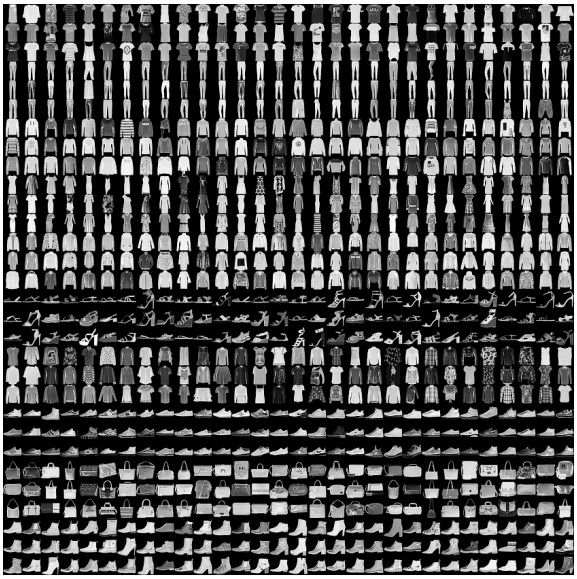
In this exercise you will implement, train and evaluate your neural network using PyTorch package.

Installation First, you will need to install PyTorch package. Installation instructions were uploaded to the Piazza. Please follow them and discuss issues there.

Data - FashionMNIST The same one from exercise 3. 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. This pixel-value is an integer between 0 and 255.

Labels. The possible labels are:

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



Instructions

In this exercise you will implement fully connected neural networks via PyTorch. You will need to implement several settings and report the effect of each setting in terms of loss and accuracy. You should explore the following:

1. Model A - Neural Network with two hidden layers, the first layer should have a size of 100 and the second layer should have a size of 50, both should be followed by ReLU activation function. Train this model with SGD optimizer.

2. Model B - Neural Network with two hidden layers, the first layer should have a size of 100 and the second layer should have a size of 50, both should be followed by ReLU activation functionm, train this model with ADAM optimizer.

3. Model C - Dropout – add dropout layers to model A. You should place the dropout on the output of the hidden layers.

4. Model D - Batch Normalization - add Batch Normalization layers to model A. You should place the Batch Normalization before the activation functions

5. Model E - Neural Network with five hidden layers:[128,64,10,10,10] using **ReLU** .

6. Model F - Neural Network with five hidden layers:[128,64,10,10,10] using **Sigmoid**.

In all these experiments you should use log_softmax as the output of the network and nll_loss function (see code example in recitation 8 slides).

Training

You should train your models using FashionMNIST dataset (the same one from ex. 3). You should train your models for 10 epochs each. You can use the code example we provide you in recitation 8 or in the PyTorch examples repository on GitHub.

You should split the training set to train and validation (80:20). Note: you do not need to load FashionMNIST manually, you can use PyTorch built-in data loaders.

Finally, you should use your best model to generate predictions for the examples in `test_x` and write them into a file named `test_y`, similarly to the previous exercise. Your predictions file should contain 5000 rows exactly.

Evaluation - Report

Your report file, `ex_4_report.pdf`, should include the following for EACH model:

1. Plot the average **loss** per epoch for the validation and training set in a single image.

2. Plot the average **accuracy** per epoch for the validation and training set in a single image.

3. Test set accuracy (original FashionMNIST test set).

4. Hyper parameters.

Good Luck!