

Introduction to ML

Exercise 7+8

Due: May 31th, 2018 (no extensions)

1 Submission Instructions:

1. Practical Part - Ex_8

- (a) You are allowed to use numpy package and PyTorch framework
- (b) Use Python 2.7
- (c) In order to submit your solution - please submit the relevant files to the corresponding assignment on the Submit system.

Your files should include:

- i. **details.txt** - A text file with your full name (in the first line) and ID (in the second line).
- ii. **ex_8_code.py** - Your code file.
- iii. **test.pred** - Your prediction for the test file.

2. Theoretical part - Ex_7

- (a) In order to submit your solution - please submit the relevant files to the corresponding assignment on the Submit system.

Your files should include:

- i. **ex_7_report.pdf** - A pdf file in which you describe your model and parameters, and explain your work.(2 pages at most)

Make sure you receive a confirmation mail from the Submit system.

Good Luck!

Exercise 4:

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.

Implementation:

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
 - b. Dropout – add dropout layers to model in B. You should place the dropout on the output of the hidden layers
 - c. Batch Normalization - add Batch Normalization layers to model in B. You should place the Batch Normalization before the activation functions
 - d. Optional – implement a convolutional neural network with any dimensions you feel like

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

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 Tirgul 8 or in the PyTorch examples repository on GitHub (link provided on Piazza).

You should split the training set to train and validation (80:20), you may find the code in the following link useful: <https://am207.github.io/2018spring/wiki/ValidationSplits.html>

Note: you do not need to load FashionMNIST manually, you can use PyTorch built-in data loaders.

Evaluation:

You should report the following results for each model in the PDF file (`ex_7_report.pdf`):

1. Training set accuracy
2. Validation set accuracy
3. Test set accuracy
4. Average training set loss
5. Average validation set loss
6. Average test set loss
7. Plot the average loss per epoch for the validation and training set