INFO284 - Second Obligatory Assignment Convolutional Neural Networks for Image Classification on Fashion MNIST dataset

than.soe@uib.no

16 March 2020 - Spring 2020

1 Overview

This project requires you to build a Convolutional Neural Network in PyTorch for the classification task on Fashion MNIST Dataset `https://github.com/zalandoresearch/fashion-mnist`. The goal of the project is for you to learn how to build a CNN for image classification as well as learn the basics of building and training a neural network. You can work in a group of two or individually. If you work in a group of two only one submission is required but write both names at the *beginning* of the submission. The deadline for the project is *6th April 2020 at 11:59 a.m (noon)!.*

2 Description of the data

The dataset description from its creators is as follows.

> 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. We intend Fashion-MNIST to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits.

You don't have to create the prepare the data for this dataset as it is it is included in the PyTorch's torchvision library `https://pytorch.org/docs/stable/torchvision/datasets.html#fashion-mnist`

3 Assignment

You should attempts the following tasks to complete the assignment.

1. Define your neural network with convolutional layers and polling layers. You are free to experiment with the number of layers, sizes and filters and strides. The requirement is that the network should have at least three convolutional layers.

2. Train your neural network. Document the training process and monitor the accuracy after each training epoch. Copy a screenshot of your training in the network and include it in the submission. Make a comment of your training based on the learning parameters (learning rate, batch size etc) that worked and that did not work.

3. Save your neural network model and upload it to any online storage that is accessible to you and include the link to your network in the submission. We should be able to access the trained model without requiring login.

4. Compute the accuracy and confusion matrix for your model.

The submission should contain all steps above documented as well as working code. Follow the style guide PEP8 mentioned here `https://www.python.org/dev/peps/pep-0008/`.

4 Resources

- Lab 6 and Lab 7 as Fashion MNIST works as a a drop in replacement for MNIST.

- Fashion MNIST `https://github.com/zalandoresearch/fashion-mnist`

- PyTorch NN `https://pytorch.org/docs/stable/nn.html`. Don't be scared of the word Tensor. Esentially, it is the same as a Numpy array that you have used multiple times before.

- Optional! If you want to monitor your training performance it is a good tool `https://pytorch.org/docs/stable/tensorboard.html`

- If you are curious about how your model fares in comparison to others `http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/`