2. Programming Tasks

Prerequisites

1. **Python:** 2.7+
2. **One of the following Deep Learning frameworks:**
    - PyTorch: http://pytorch.org/
    - Tensorflow: https://www.tensorflow.org/install/

2.1 Build a Convolutional Neural Network

This task is meant to get you acquainted with one of the following Deep Learning frameworks

- PyTorch (the Python version of the Torch Framework, originally for the Lua language);
- Tensorflow

The main goal of this task is to build a simple Convolutional Neural Network and train it on the MNIST dataset.

Task Description

Convolutional Neural Networks (Conv Nets, a.k.a. CNNs) have led to a paradigm shift in the field of Computer Vision. While popular Conv Net architectures (e.g., Alex Net, LeNet, ResNet, etc.) are generally used in vision problems directly, we still need to come up with our own architectures to deal with very specific problems.

Part 1

For this part of the task you need to develop the following Conv Net architecture using the Deep Learning framework of your choice (the layers need to be implemented in the order specified):

Layers 1. Conv Layer: `num_filters=64`, `kernel_size=(3, 3)`, `activation=relu`, `padding=valid`, `strides=(2, 2)`

2. Conv Layer: `num_filters=32`, `kernel_size=(2, 2)`, `activation=relu`, `padding=same`, `strides=(1, 1)`

3. **Max Pool Layer:** `pool_size=(2, 2)`, `strides=(1, 1)`
4. **Dropout:** `rate=0.35` (Fraction of the input units to drop)
5. **Flatten**
6. **Dense:** `num_units=256`, `activation=tanh`
7. **Dropout:** `rate=0.5` (Fraction of the input units to drop)
8. **Dense:** `num_units=10`, `activation=softmax`

Part 2

Train the above architecture on the MNIST dataset with the following optimizers for 10 epochs each and plot their corresponding loss curves.

Use **categorical Cross Entropy** as the loss function.

Optimizers

1. Adam
2. ADADELTA
3. Stochastic Gradient Descent

Resources

1. PyTorch CNN MNIST Example

- https://github.com/pytorch/examples/tree/master/mnist

2. PyTorch Optimizers

- http://pytorch.org/docs/master/optim.html

3. Tensorflow CNN MNIST Example

- Beginners: https://www.tensorflow.org/get_started/mnist/beginners
- Advanced: https://www.tensorflow.org/get_started/mnist/pros

4. Tensorflow Optimizers

- https://www.tensorflow.org/api_guides/python/train