Prof. Dr. Rudolf Mathar, Dr. Arash Behboodi, Emilio Balda

Exercise 2

Friday, October 27, 2017

Problem 1.

MNIST dataset (PyTorch version)

PyTorch is a python library developed by Facebook particularly for deep learning using GPU and CPU. In this tutorial, we load the MNIST dataset. We import two libraries *torch* and *torchvision*, the later for loading datasets.

```
In [1]: %matplotlib inline
        import torch
        from torchvision import datasets
        from matplotlib.pyplot import imshow
        import matplotlib.pyplot as plt
        import numpy as np
```

MNIST dataset is loaded using the following command. There is a download flag, `DNLD`, which is set to `True` if the dataset is to be downloaded from the Internet. If MNIST dataset for PyTorch has been downloaded, two files `training.pt` and `test.pt` are found in the folder `./processed`. In this case, put the flag to `False` so that it is not downloaded again. The flag `train` determines if you intend to load training or test set.

```
In [2]: DNLD=True
        trainingMNIST = datasets.MNIST('./MNIST_torch', train=True, download=DNLD)
        testMNIST = datasets.MNIST('./MNIST_torch', train=False, download=DNLD)
```

```
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz
Processing...
Done!
```

The size of training and test set is the same as before. Note that the validation set should be manually constructed from the training set.

```
In [3]: print("Size of:")
        print("- Training-set:\t\t{}".format(len(trainingMNIST)))
        print("- Test-set:\t\t{}".format(len(testMNIST)))
```

```
Size of:
- Training-set:         60000
- Test-set:             10000
```

Each entry of the dataset consists of `PIL image module` and a number which represent the label of the image:

```
In [4]: trainingMNIST[0]
```

```
Out[4]: (<PIL.Image.Image image mode=L size=28x28 at 0x7FC2C3198BA8>, 5)
```

The shape of the image and their labels follow standard description of MNIST dataset.

```
In [5]: print("shape of first entry:",trainingMNIST[0][0].size)
        print("shape of second entry:",trainingMNIST[1][0].size)
        print("shape of third entry:",trainingMNIST[2][0].size)
```

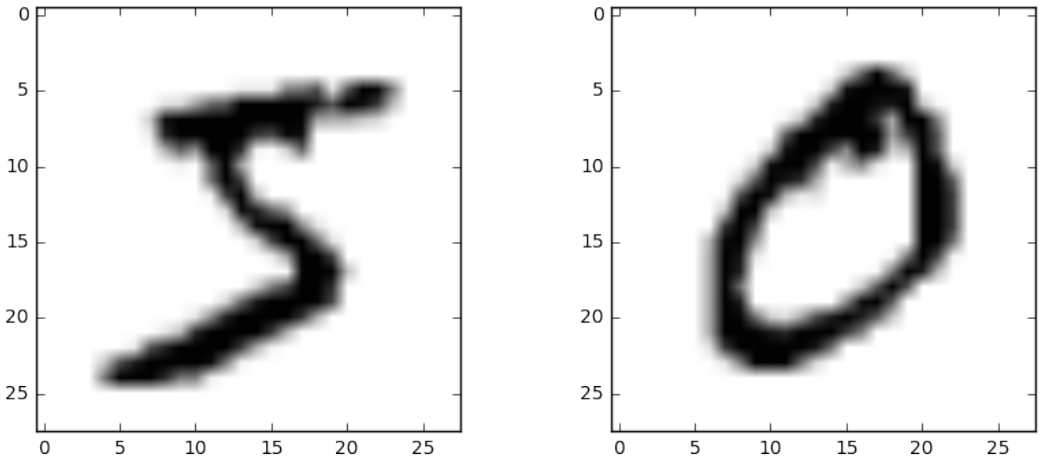
```
shape of first entry: (28, 28)
shape of second entry: (28, 28)
shape of third entry: (28, 28)
```

```
In [6]: print("label of first entry:",trainingMNIST[0][1])
        print("label of first entry:",trainingMNIST[1][1])
        print("label of first entry:",trainingMNIST[2][1])
```

```
label of first entry: 5
label of first entry: 0
label of first entry: 4
```

They can be converted to `numpy arrays` and manipulated using standard python.

```
In [7]: imshow(np.asarray(trainingMNIST[0][0]),cmap='binary')
        plt.show()
        imshow(np.asarray(trainingMNIST[1][0]),cmap='binary')
        plt.show()
```


0
5
10
15
20
25
0
5
10
15
20
25
0
5
10
15
20
25
0
5
10
15
20
25