Exercise Session
CIVIL-459
Spring 2018
Lab 2

EPFL
School of Civil Engineering
Prof. Alexandre Alahi
TA: George Adaimi

Setting Up - [If jupyter and other packages are not working]

First download and install:

Oracle's VirtualBox

then download the file:

DAIT VM.ova

and open it in VirtualBox with File ⇒ Import Appliance.

You should now see an entry in the list of VMs. The first time it starts, it provides a menu to choose the keyboard layout you want to use.

The VM automatically starts a JupyterLab on port 8888 and exports that port to the host. This means that you can access this JupyterLab with a web browser on the machine running VirtualBox at:

`http://localhost:8888/`

Yo can use python notebooks, view files, start terminals, and edit source files. Typing !bye in a notebook or bye in a terminal will shutdown the VM.

Files saved in the VM are erased when the VM is re-installed so make sure to copy your files from the VM to a local storage

Getting Started

Familiarize yourself with pytorch using the *Tutorial 1* and *Tutorial 2* notebooks as well as the tutorials found on the Pytorch official website. You also need to follow the instruction on the official Pytorch website (here) to install the Pytorch package. If you are installing it on a machine with no GPU, choose 'None' for the CUDA setting.

If you are using the VM provided, upload the whole exercise session to Jupyter using the upload button while maintaining the file hierarchy provided in the zip file.

Problem Set 2

The problem set contains 2 notebooks. The first notebook helps you implement softmax using just python. The second notebook explains how to use Pytorch to build a simple fully connected neural network. Follow the problem statement of each exercise.

Note: Do not forget to submit, with your solution, the trained model of both notebooks (Softmax and Pytorch) as well as filling the "run.py" file. This file will be used to test your model on our dataset and check your accuracy. The methods to save and load your models are mentioned in the notebook

What to do

To implement Softmax, there are 3 files that need to be filled: **linear_classifier.py, softmax.ipynb, softmax.py**

- linear_classifier.py
  - train(): Implement Stochastic Gradient Descent to train your model
  - predict(): Use the weights learned using the train() method to predict the labels
- softmax.py
  - softmax_loss_vectorized(): Returns the softmax loss function and its derivative (Vectorized method)
- softmax.ipynb
  - In the 4th cell, use the methods you created to train the model and tune your learning rate. Use the validation data to try many different learning rates and get the best model. Don't forget to save the best model for submission.

To implement the neural network using Pytorch, there is 1 file that needs to be filled: **Linear Classification pytorch.ipynb**

1. Define the different layers of the neural network. The neural network should have at least 2 layers but can be as large you want.
2. Set up the forward pass by connecting the different layers of the neural network. **Note:** Do not forget to use activation layers.
3. Create the network by specifying the input and output size. Then choose the optimizer you want to use (e.g., SGD or Adam). You should tune the parameters of the optimizer.
4. Implement the training process and save your best model

Finally, for us to test your code and get an accuracy, you should implement inside the **run.py** both methods: **predict_usingSoftmax(), predict_usingPytorch()**. In both methods, you should load the saved model, input to it the data and get the prediction labels. We will be using our own test data.

Helpful References

- Pytorch Documentation: `http://pytorch.org/docs/0.3.1/`
- Pytorch Tutorial: Official link, Collection