Due: December 8, 2017

Homework 6

CPSC 475/BENG 475

Goal

This assignment will help you get familiar with Python and the Tensorflow™ framework. Tensorflow™ is an open source software for numerical computation using data flow graphs. For this assignment we will use the MNIST dataset, which contains handwritten digits already labeled. You can expect to learn the following:

- Tensorflow™ built-in operations, such as `tf.matmul`, `tf.reduce_mean` and the `tf.nn` module.
- Use `tf.Session()` to initialize the variables in the computation graph and to train the model.
- Use tensorboard for visualization.
- Build a softmax classifier.
- Build a convolutional neural network and learn how to determine the dimension for each layer.
- Verify the robustness issue of neural networks by implementing an adversarial attack for a neural network.
- Get familiar with the handy Jupyter Notebook.

Notice that some of the materials are not covered in the class. Several references are included in the assignment to help you if you'd like to learn a bit more about them. You are strongly encouraged to read those materials, and naturally we will be available to help you understand them. Here are some of the relevant topics: softmax classifier, convolution neural nets, max pooling, rectified linear unit, dropout, cross entropy loss, gradient-based optimization (such as SGD, Adagrad, Nesterov accelerated gradient, etc.).

Setting Up the Environment

You need to use python (2.7+ or 3.5+) to work on this assignment. The following python packages are required: numpy, pillow, and Tensorflow™. Python and all the necessary packages are pre-installed in the Zoo. We strongly recommend that you work on zoo machines unless you are already familiar with installing python modules. If you'd prefer to set up an environment on you personal computer, check below.

Anaconda (optional)

Anaconda is a popular Python data science platform. Check the following link for instructions.

https://www.anaconda.com/distribution/

Virtual Environment (optional)

Virtualenv and Virtualenvwrapper are python packages used to create isolated python environments (if you install Anaconda, it already comes with its own environment managar). You can maintain many python environments with different packages installed on them. And those environments can be removed easily if they are no longer needed.

http://docs.python-guide.org/en/latest/dev/virtualenvs/

https://virtualenv.pypa.io/en/stable/

http://www.pythonforbeginners.com/basics/how-to-use-python-virtualenv

Installing Python Packages

Numpy, pillow, jupyter notebook and Tensorflow™ are required for this assignment. To install them, run:

```
pip install numpy pillow jupyter
```

```
pip install --upgrade Tensorflow
```

You may need `sudo` if you try to install the packages on the python used by the operational system and when you encounter the "permission denied" error.

Working on the assignment

To work on the assignment, you need to start the `jupyter notebook` first.

- open a terminal and go to the root directory of the assignment.
- execute `jupyter notebook`
- A webpage should pop up in your browser. If not, open: http://localhost:8888/tree

Tensorflow 101 (optional)

If you find the assignment challenging, you could go through the following tutorials, as well as the default tutorial on the Tensorflow™ webpage.

`http://web.stanford.edu/class/cs20si/lectures/slides_01.pdf`

`https://cs224d.stanford.edu/lectures/CS224d-Lecture7.pdf`

Q1: Softmax Classifier

Softmax classifier is a generalization of binary logistic regression. For details about the algorithm and concepts, check this link. In this question, you will train a softmax classifier for the MNIST dataset.

Q2: Convolutional Neural Network

In Q2, we will build a convolution neural neural net with two convolution layers. For the tutorial on the convolution neural networks, please check the link.

Q3: Adversarial Attack

In this question, we will implement fast gradient-sign attack (FGSM) on a convolution neural network for MNIST dataset. This reveals the lack of robustness issue from which deep networks suffer.

Happy coding! And remember to come talk to us if you are having trouble with python or the material.