Lab Course: distributed data analytics
Exercise Sheet 9

Nghia Duong-Trung, Mohsan Jameel
Information Systems and Machine Learning Lab
University of Hildesheim

Submission deadline: June 23^{rd}, Friday 23:59PM (on LearnWeb, course code: 3117)

Instructions

Please following these instructions for solving and submitting the exercise sheet.

1. You should submit a zip or a tar file containing two things a) python scripts and b) a pdf document.
2. In the pdf document you will explain your approach (i.e. how you solved a given problem), and present your results in form of graphs and tables.
3. The submission should be made before the deadline, only through learnweb.
4. If you are M.Sc. Data Analytics summer 2017 intake student, you should submit to “First term students” link on LearnWeb.
5. If you are not M.Sc. Data Analytics student, you can submit to anyone of the links above.

Setting up TensorFlow

Install TensorFlow in your system by following the installation instructions presented in the TensorFlow official homepage `https://www.tensorflow.org/install/`. All the solution’s code in this exercise sheet has been tested in several environment settings:

1. Local system
   (a) Windows 10, Anaconda3, Python 3.5
   (b) Nvidia Geforce 940MX. You can install the CPU-only version of TensorFlow if your system does not have a Nvidia graphics card. The solution’s code is also tested on the CPU-only version of TensorFlow.
   (c) Tensorflow 1.1.0
2. ISMLL’s Xeon-phi server
   (a) Ubuntu 16.04.2 LTS, Anaconda3, Python 3.5
   (b) Nvidia Tesla K80
   (c) Tensorflow 1.1.0

TensorFlow Tutorials

There are several good tutorials that you can start with:

1. Getting Started With TensorFlow
   `https://www.tensorflow.org/get_started/get_started`
2. Stanford CS 20SI: Tensorflow for Deep Learning Research `http://web.stanford.edu/class/cs20si/index.html`
3. Google Developers Channel: TensorFlow Dev Summit 2017
   `https://www.youtube.com/watch?v=mWl45NkFBOc&list=PLOU2XLYxmsIKGc_NBoIhTn2Qhraji53cv`

4. Hands-on TensorBoard
   `https://www.youtube.com/watch?v=eBbEDRsCmv4&index=5&list=PLOU2XLYxmsIKGc_NBoIhTn2Qhraji53cv`.
   Note: the slides and code of this video is provided in the video's comments.

5. TensorFlow Tutorial and Examples for beginners
   `https://github.com/aymericdamien/TensorFlow-Examples`

Exercise 1: Logistic Regression on the Iris dataset (10 points)

First of all, let's look at some logistic regression tutorials to understand how it works:

1. Logistic Regression Tutorial for Machine Learning
   `http://machinelearningmastery.com/logistic-regression-tutorial-for-machine-learning/`

2. Example of logistic regression in Python using scikit-learn
   `http://www.dataschool.io/logistic-regression-in-python-using-scikit-learn/`

You might also look at the TensorFlow Tutorial and Examples for beginners [5] in the tutorials section above.

The dataset used in this exercise is the Iris dataset. You can reuse it from exercise sheet 8 or load it using scikit-learn `http://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html`.

Along with your solution, you might have to follow the proposed procedure.

1. Load the Iris dataset, randomly split it into training set 90% and test set 10%.

2. Define a learning model using cross entropy cost function. Explain how you come up with the learning model. Cross entropy cost function is described in `http://neuralnetworksanddeeplearning.com/chap3.html`

3. Train the model on the training set and make prediction on the test set.

4. Report and plot accuracy on both training set and test set. A good solution might achieve 100% accuracy on the test set.

5. Create a Tensorboard that presents basic information such as scalars, graphs, distributions and histograms. You might check your Tensorboard at `localhost:6006`. The Tensorboard tutorial is provided in [4] in the tutorials section above.

Exercise 2: Logistic Regression on the Olivetti faces dataset (10 points)

In this exercise, let apply your logistic regression model on an image dataset, the Olivetti faces dataset `http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_olivetti_faces.html#sklearn.datasets.fetch_olivetti_faces`.

Along with your solution, you might have to follow the proposed procedure.

1. Load the Olivetti faces dataset, randomly split it into training set 90% and test set 10%.

2. Define a learning model using cross entropy cost function. Explain how you come up with the learning model.

3. Train the model on the training set and make prediction on the test set.

4. Report and plot accuracy on both training set and test set.

5. Create a Tensorboard that presents basic information such as scalars, graphs, distributions and histograms. You might check your Tensorboard at `localhost:6006`.