Lab Course: Distributed Data Analytics
Exercise Sheet 5

Mofassir ul Islam Arif
Information Systems and Machine Learning Lab
University of Hildesheim

Submission deadline: Monday June 03, 23:59PM (on LearnWeb, course code: 3116)

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 the form of graphs and tables.
3. The submission should be made before the deadline, only through learnweb.

Complex Data: Time Series Forecasting

A time series data consists of data points in a sequence observed at a particular time. The time series data usually has some internal structure i.e. trends, seasonal variation etc. In the first part of this lab you will handle the time series data and apply some simple preprocessing techniques and ARIMA model for forecasting.
Note: In Exercise 1A and 1B you are not required to do a performance analysis.

TensorFlow Tutorials

Install TensorFlow in your system by following the installation instructions presented in the TensorFlow official homepage `https://www.tensorflow.org/install/` 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: Linear Regression (10 points)

Exercise 1A: Univariate Linear Regression (5 Points)

1. Generated Toy Data Using:
   - $y = 0.5 \times x + 2 + \lambda$

- x has size 1000 is generated from a Uniform distribution
- λ is gaussian noise centered around $\mu = 0.0$ and $\sigma = 50$
- split the data into 90-10 train/test splits

2. Using tensorflow implementation (not keras, nor numpy), write a program to fit the parameters of linear regression.

$$y = Wx + b \tag{1}$$

The aim of linear regression is to fit a W and b to the data, (W and b are scalars for univariate linear regression. You need to initial W and b to zeros and then iteratively update them until the model converges (Hint: You could try initializing these to random and see how that impacts training).

3. Your code should run for N number of epochs over the entire data. You can set N yourself but it should be high enough for the model to learn the parameters.

4. You need to write the loss operation yourself.

5. You can use the tensorflow optimizers to minimize the loss operation.

6. After tranining is complete, you need to plot the ground truth values and your model prediction in the same plot (use scatter plot).

Exercise 1B: Multivariate Linear Regression (5 points)

1. Update Exercise 1A to handle Multivariate linear regression.

2. Use `https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data` as the data. Here the MPG is the target.

3. You need to ensure that the data has no NA values

4. You need to ensure that you change the categorical values to one-hot encoding and normalize the data.

   - Multivariate linear regression has more than one feature therefore W and b are no longer scalars but rather vectors.
   - Adapt your implementation accordingly.
   - For this question you need to implement Mean Absolute Error, Mean Square Error and Root Mean Squared Error (separately).
   - You need to present the loss graphs for all three. Comment on the behavior you see. (Hint: Vary the learning rates and observe convergence)

5. After tranining is complete, you need to plot the ground truth values and your model prediction in the same plot (use scatter plot).

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. Report and plot loss on training set.
6. Use 3 different optimizers, comment on the behavior of each (Hint: Vary the learning rate for each optimizer and see how the optimizer can handle different learning rates.)

Annex

1. Time Series Analysis `https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm`
2. ARIMA model `https://www.digitalocean.com/community/tutorials/a-guide-to-time-series-forecasting`
3. sklearn SGD `http://scikit-learn.org/stable/modules/sgd.html`