Assignment#2: Linear regression in TensorFlow

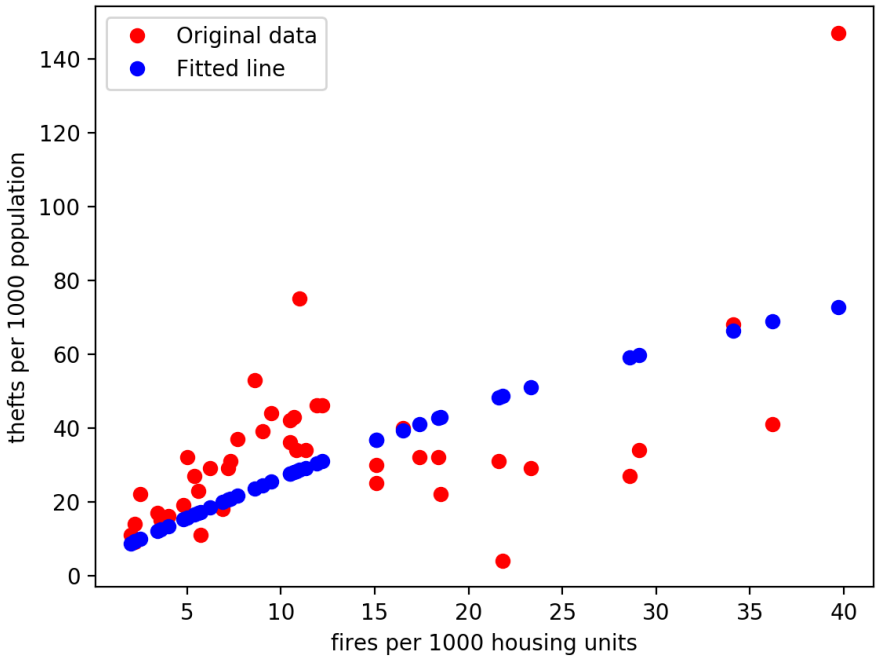
prepared by Teeradaj Racharak (r.teeradaj@gmail.com)

Problem 2.1. (Polynomial regression) This is continuous from Problem 1 of Lab 2.

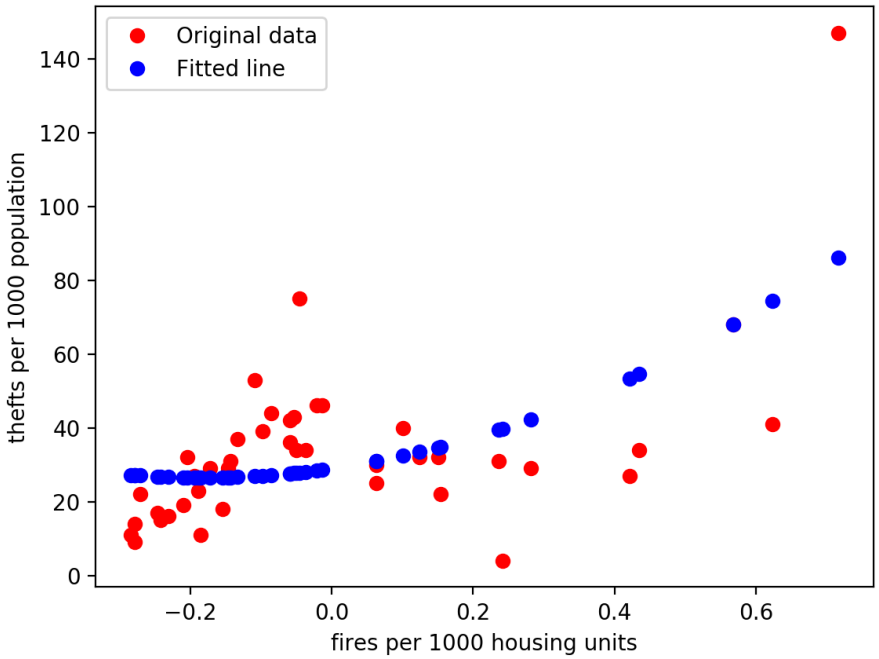
After training for 30,000 epochs, you found that mean squared error was around 180.41008, which is still large. Now, you want to improve the error by fitting the line with a polynomial equation as follows:

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

Write a program based on batch gradient descent in TensorFlow and see if the new hypothesis function can improve mean squared error or not (Hint: try setting $\alpha = 0.00001$ and training epochs to 100,000). The following figure shows its training result.



Problem 2.2. (Feature scaling) Why the number of training epochs is more (*i.e.* mean squared error ≈ 207.20158, $\theta_0 \approx 3.9169812$, $\theta_1 \approx 2.4351342$, and $\theta_2 \approx -0.017747309$)? Again, let's try to improve it with feature scaling. Using mean normalization, you will see that training epochs are reduced to 30,000 and the learning rate can be also changed to 0.3.



Problem 2.3. (Multivariate Linear Regression) This example is based on a 1993 dataset of a study of different prices among some suburbs of Boston. It originally contains 13 variables and the mean price of the properties there. However, we will ignore to consider a certain column (named 'B'). The remaining variables are used to model a linear function.

Pandas Library

This example uses this library, so I will explain it shortly here.

From the Pandas site (*i.e.* pandas.pydata.org):

> "Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for Python."

Pandas's main features are as follows:

- It has read/write file capabilities from CSV and text files, MS Excel, SQL databases, and even the scientifically oriented HDF5 format;
- The CSV file-loading routines automatically recognize column headings and support a more direct addressing of columns;
- The data structures are automatically translated into NumPy multidimensional arrays.

Dataset Description

The dataset is represented in a CSV file and we'll open it using Pandas library.

The dataset includes the following variables:

- CRIM: Per capita crime rate by town;
- ZN: Proportion of residential land zoned for lots over 25,000 feet^2;
- INDUS: Proportion of non-retail business acres per town;
- CHAS: Charles River dummy variable (=1 if tract bound rivers; 0 otherwise);
- NOX: Nitric oxides concentration (parts per 10 million);
- RM: Average number of rooms per dwelling;
- AGE: Proportion of owner-occupied units built prior to 1940;
- DIS: Weighted distances to 5 Boston employment centers;
- RAD: Index of accessibility to radial highways;
- TAX: Full-value property-tax rate per $10,000;
- PTRATIO: Pupil-teacher ratio by town;
- LSTAT: %lower status of the population;
- MEDV: Median value of owner-occupied homes in $1,000's.

Goal

Write a program that models multivariate linear regression for predicting MEDV value.

Further Reading

You may further check this blog post (https://medium.com/all-of-us-are-belong-to-machines/gentlest-intro-to-tensorflow-part-3-matrices-multi-feature-linear-regression-30a81ebaaa6c) to see an idea about how we can use matrices to simplify an implementation of multivariate linear regression model !