Linear regression and stochastic gradient descent

Goal

The goal of this experiment it:

1. Further understand of linear regression， closed-form solution and stochastic gradient descent.

2. Conduct some experiments under small scale dataset.

3. Realize the process of optimization and adjusting parameters.

Dataset

For this experiment we used Housing scale version in LIBSVM Data, which including 506 samples and each sample has 13 features.the we will divide it into training set and validation set.

Environment for experiment:

To realize our experiement we will need to septup the following software and necessary library

python3, at least including following python package: sklearn，numpy，jupyter，matplotlib

It is recommended to install anaconda3 directly, which has built-in python package above.

Experiment step

The experiment will follow two step which are:

Closed-form solution of linear regression and linear regression and stochastic Gradient descent

- Closed form solution of linear regression

1- Load the experiment data. You can use load_svmlight_file function in sklearn library.

2- Divide dataset. You should divide dataset into training set and validation set using train_test_split function. Test set is not required in this experiment.

3- Initialize linear model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.

4- Select a Loss function and calculate the value of the Loss function of the training set,denoted as Loss.

5- Get the formula of the closed-form solution, the process details the courseware ppt.

6- Get the value of parameter W by the closed-form solution, and update the parameter.

7- Get the Loss， loss_train under the training set and loss_val by validating under validation set.

8- Output the value of Loss, loss_train and loss_val.

-Linear regression and stochastic gradient descent

1- Load the experiment data. You can use load_svmlight_file function in sklearn library.

2- Divide dataset. You should divide dataset into training set and validation set using train_test_split function. Test set is not required in this experiment.

3- Initialize linear model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.

4- Choose loss function and derivation: Find more detail in PPT.

5- Calculate G toward loss function from each sample.

6- Denote the opposite direction of gradient G as D .

7- Update model:Wt = Wt-1 + nD.n is learning rate, a hyper-parameter that we can adjust.

8- Get the loss loss_train under the training set and loss_val by validating under validation set.

9- Repeat step 5 to 8 for several times, and output the value of loss_train as well as loss_val.