# OASIS ML Group TRAINING 01

## ◆ Linear Regression simple practice

In this practice, we are going to predict the house prices from the model which training by using the datasets called "The Boston Housing Dataset". This dataset used frequently in Machine Learning issues. Please remember that your goal is to use the attributes *(NOT contain PRICE)* to build a model for predicting house prices. Please follow hints below to finish it.

### ▫ Practice 01：

Please using "RM" as feature to train the model. You are asked to plot "**loss per iteration**" and "**linear regression line with input data**" *(Show below)*.

- ■ Recommend initial parameter：
  Iteration：50000, learning rate：0.01

- ▪ **Hint 01**：Import library you needed, but do not call API.
  *For eaxmple：numpy, pandas, sklearn are allowable.*

- ▪ **Hint 02**：Get the datasets from sklearn.

```
from sklearn.datasets import load_boston

boston = load_boston()
```

In this practice, you don't need to slice datasets into "training" and "testing".

- ▪ **Hint 03**：It's a simple linear regression model, so the prediction function is

$$\textbf{predict y} = w_0 + w_1 x$$

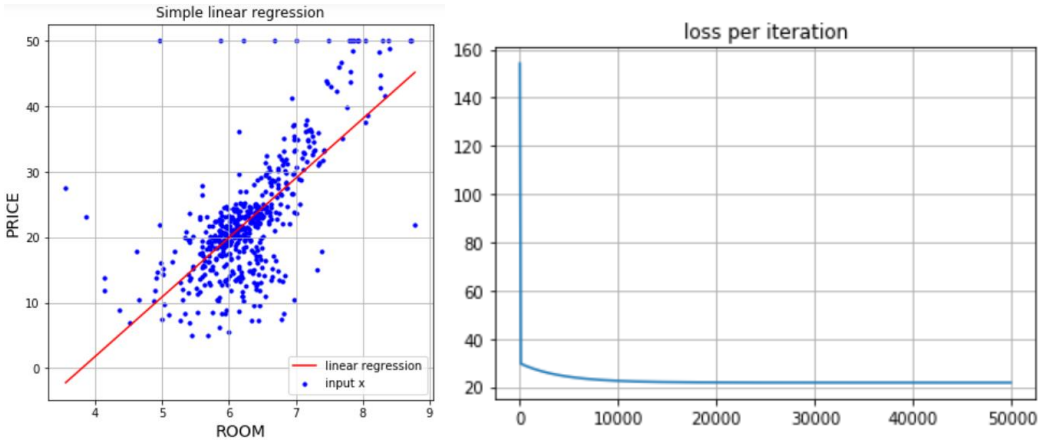
- ▪ **Hint 04**：Use MSE as loss function.

$$L = \frac{1}{2N}\sum_{m=0}^{N-1}\left(pred\_y^{(m)} - label\_y^{(m)}\right)^2, \textbf{ where N = total samples}$$

- ▪ **Hint 05**：Use gradient descent to update weights.

$$w = w - \alpha\frac{\partial Loss}{\partial w}$$

- ▪ **Hint 06**：After training, connect the predict y from x_min and x_max. Then it's the best line to describe all the input. *(Use "scatter" to plot the input points)*



## Practice 02：

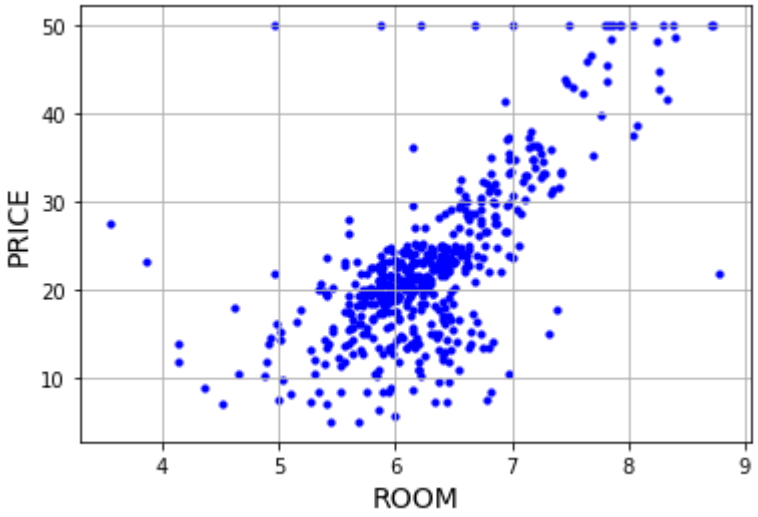
Add one more feature like “LSTAT”, then train again.
What happened?  Hot to fix the problem?
Please plot “loss per iteration” in figure.

## Practice 03：

Add one~three more features, then plot “**loss per iteration compare with different features**” in figure.

## Simple plot example：

```
plt.scatter(x[:,1], y_label, s=10, c='b')
plt.xlabel('ROOM', fontsize=14)
plt.ylabel('PRICE', fontsize=14)
plt.grid(True)
plt.show()
```



- **Review：**

**Gradient Descent**

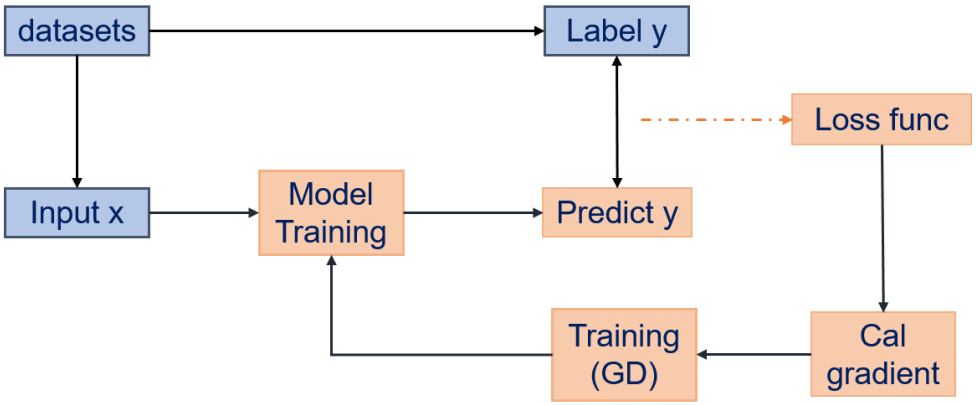



**Figure. Simple GD flow structure**

You can find that “weight update” plays an important role. Below are some useful ways to do. Try to figure out what those method means, why and how. You don’t need to implement those in this practice.

- Stochastic gradient descent, SGD
- Batch gradient descent, BGD
- Momentum
- Adagrad
- Adam