

# ELEC 4511/5511

## Lab 3

### Build your own simple linear regression class

In this lab, you will use python and NumPy to create a class called "LinearModel" to solve the regression problem of  $y=wx+b$ . For details of the specific steps, please refer to the relevant content in the lecture.

The template of class and the code of some methods are given below, please complete the rest. ("loss", "gradient\_descent", and "fit")

```
class LinearModel(object):
    def __init__(self):
        self.w = np.random.randn(1)[0]
        self.b = np.random.randn(1)[0]

    def loss(self,x,y): # Create a cost function. Calculate the gradient of w and b

    def gradient_descent(self, gradient_w, gradient_b, learning_rate = 0.1):
        #update w, b

    def fit(self, X, y): # this is the main method. You may call "loss" and "gradient_descent" in this method

    def result(self):
        return self.w, self.b
```

You will use the data in "Salary\_Data.csv" to test the class you designed. Please refer to the related content in the video lecture for the way of reading data. The test method is as follows. Where X refers to the data in the "YearsExperience" column, and y is the "Salary" column.

```
lm = LinearModel()
lm.fit(X,y)
lm.result()

plt.scatter(X,y,c='red')

plt.plot(x, lm.result()[0]*x+lm.result()[1], color = 'green')
```

**Canvas submission:** Please submit your report in “**.pdf**” file format and compress your codes into a “**.zip**” file.

What needs to be included in your report:

1. Screenshots of the results you get after running the program.
2. Copy and paste your code. And write comments for each function/method.
3. Please write a short analysis of each problem, mainly explaining how you think about the design of the function, what troubles you encountered during the design process, and how you finally solved them, etc.

What needs to be included in your “**.zip**” file:

Your Python code (.py file) for each problem.