

Artificial Intelligence (EI06024001)

Assignment 4: Linear Regression (Gradient)

Reminder: Gradient Descent

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmin}} \underbrace{\frac{1}{n} \sum_{i=1}^n (y_i - \theta^T x_i)^2}_{\mathcal{L}(\theta)}$$

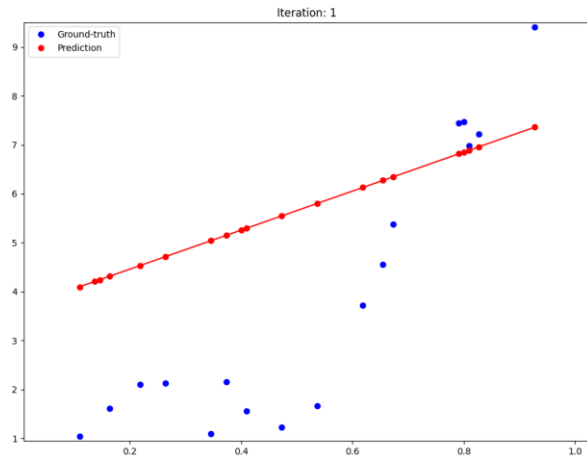
$$\theta^{new} = \theta^{old} - \alpha \cdot \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$$

$$= \theta^{old} - \alpha \cdot \begin{bmatrix} \frac{\partial \mathcal{L}(\theta_1)}{\partial \theta_1} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta_p)}{\partial \theta_p} \end{bmatrix}$$

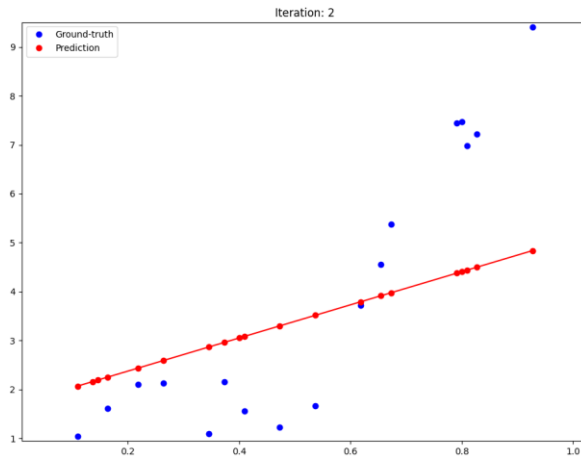
main_linear_reg_grad.py (1)

```
# STEP 3: DO GRADIENT DESCENT -----  
def get_gradient(theta, X, Y):  
    # Write code here!  
    mse = 100.0 # 이 부분을 지우고 작성하시오!  
    gradient = np.ones((2, 1)) # 이 부분을 지우고 작성하시오!  
    return gradient, mse  
  
theta_hat = np.random.randn(2, 1) # theta_hat 초기값을 random하게 정한다.  
alpha = 0.5  
tolerance = 1e-5
```

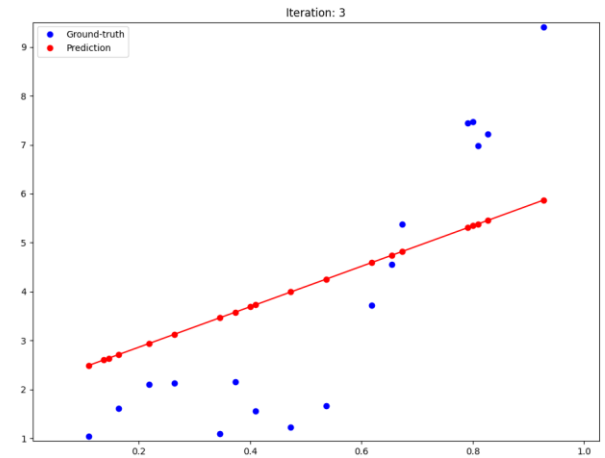
Expected Result



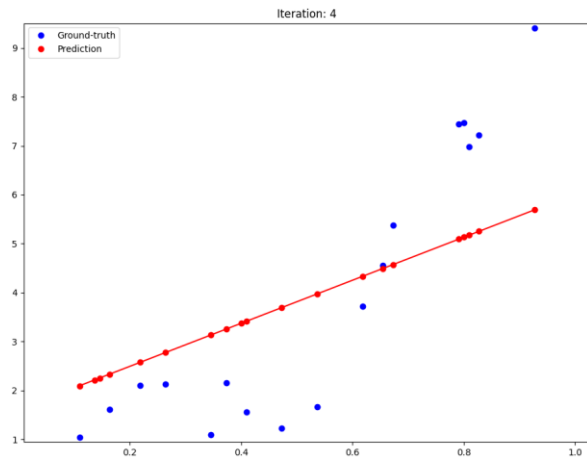
Iter: 1



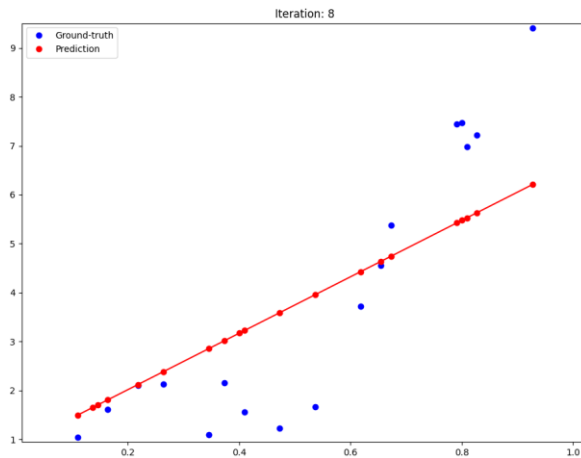
Iter: 2



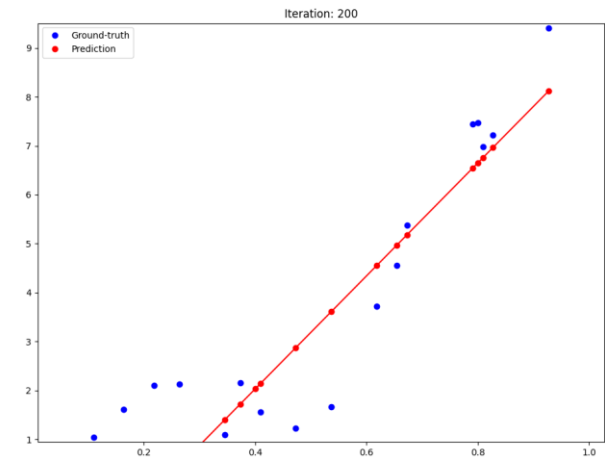
Iter: 3



Iter: 4



Iter: 8



Iter: 200