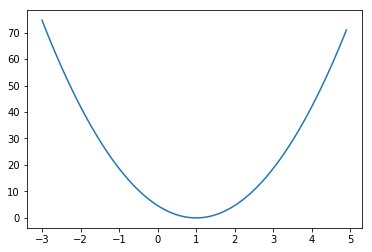
머신러닝 4주차

Linear\_Regression.py

* 변수를 이용한 방법
* PlaceHolder를 이용한 방법
* Gradient Descent
* 어떤 data가 주어졌을 때, 그 값을 가장 갈 나타내는 직선을 찾는다.
* H(X) = W\*X + b  
  H(X) : 가장 잘 나타냈다고 생각되는 직선(가설:Hypothesis)  
  X : 학습 시킬 data  
  b : Bias W : Importance Weight
* W,b의 값을 찾는것이 관건

**minimize\_cost**

[](https://github.com/DKU-Ratatouille/eddie94/blob/master/%EC%BD%94%EB%93%9C/Deep%20Learning/Linear_Regression/img/minimize_cost.png)

Gradient Descent를 통해 최소값을 찾는다  
그림에서 W=1일때 최소가 됨을 알 수 있다.

**Multi Variable**

* H(x1,x2,x3) = w1x1+w2x2+w3x3+b 의 형태로 가설이 바뀜  
  H(**X**) = **XW** 의 행렬 형태로 표현할 수도 있음

**data file의 개수가 많은 경우**

많은 양의 data를 numpy를 통해 읽어 오는 경우  
메모리 부족 현상이 나타날 수 있다  
이런 경우에는 Tensorflow의 Queue Runner를 이용해  
data를 한번에 읽어오지 않고 나눠서 읽어 오는 것이 가능

Logistic classification

* H(x)의 값이 0과1사이의 값이 되어야 함  
  기존의 가설로는 0보다 작거나, 1보다 큰값이 나오므로 조정 필요
* 따라서 g(z)를 이용해 가설을 0과1 사이의 값으로 만들어줌  
  (sigmoid function)
* z = **WX**  
  H(**x**)=g(z)
* 기존의 Gradient descent로는 sigmoid함수의 최소값을 결정할 수 없다
* 따라서 새로운 cost function이 필요함  
  c(H(x),y)= -log(H(x)) : y=1  
  -log(1-H(x)) : y=0

Multinomial classification

* Logistic classification과 달리 클래스가 여러개인 경우
* Softmax  
  softmax는 갖고있는 data를 확률로 normalize시켜준다
* cost function은 cross-entropy를 통해 구함