#3 - Linear Regression

주제 : Linear Regression

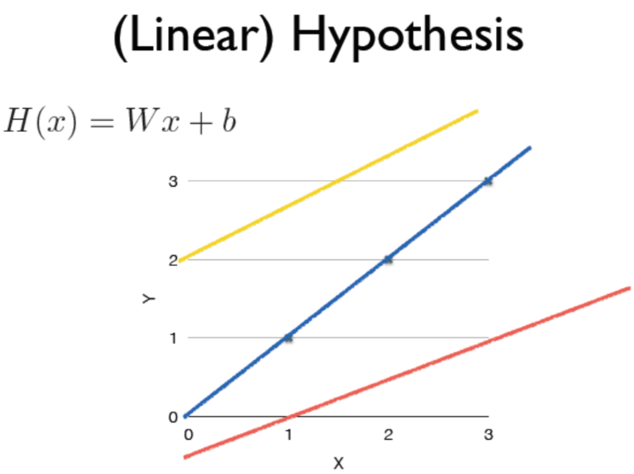
링크 :

[#3.Lec] Linear Regression - 딥러닝 홀로서기
발표자료 링크 슬라이드1: https://hunkim.github.io/ml/lec2.pdf 슬라이드2: https://hunkim.github.io/ml/lec3.pdf 슬라이드3: https://docs.google.com/presentation/d/1bHVx...
https://www.youtube.com/watch?v=DWdtr_IURkU&feature=youtu.be

1. Linear Regression

   예제) 공부한 시간(x)를 가지고 성적(y)을 예측하는 모델를 만들어보자

   F(x) = wx + b라고 가정하자



- Hypothesis (가설)

  다양한 F(x)가 만들 수 있는데, 이를 Hypothesis(가설)라 한다

- Cost Function (=Loss Function)

  이 가설이 좋은지 나쁜지를 가름하는 함수를 Cost Function이라 한다

    C1 = H(x) - y

    (*여기서 H(x) = Wx + b)

  → 각 관측점들(y)과 f(x)와의(관측점y에 대응되는 f(x)의 y값) 거리를 C1으로 정의

  → 음수와 양수를 더해서 0으로 수렴하면 좋은 모델이라고 판단할 수 있어서 보통은 절대값을 씌우거나 아래와 같이 제곱을 하여 평균을 취한다

$$\text{Cost(w, b)} = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$

Equals,

$$\text{Cost(w)} = \frac{1}{m}\sum_{i=1}^{m}(w_x^{(i)} - y^{(i)})^2$$

- Linear Regression은 이 Cost Function 수치를 Minimize하는 것을 목표로 한다

2. How to minimize Cost Function

- 기본적으로 Gradient Decent 방식을 활용한다

  ① Gradient Decent 방식

- Cost - W의 함수에서 임의의 지점의 w값이 4의 cost값을 구하고, 두번째로 4+e인 지점에서의 cost 값을 구해서 비교해본다
- 이때 4와 4+e지점에서의 cost 값의 기울기를 구해보면(수치해석) 양의 값이 나올 수 있다(여기서는 기울기가(= $\frac{\partial Cost}{\partial w}$) 5라 하자)
- 이는 즉, w값의 증가가 cost값의 증가로 이어진다는 결론이 나오고, Cost를 Minimize하기 위해서 w값을 감소시키기로 한다
- 이때 감소시키는 양을 Gradient라고(= $\frac{\partial Cost}{\partial w}$) 하고, 여기서는 4 - gradient가 될 것이다 (의문: 왜 함수의 계수에 해당하는 4에 기울기에 해당하는 Gradient값을 빼주는 것일까?)

- 하지만, 기존 Gradient(=기울기)값은 너무도 큰 수이기 때문에 Gradient값에 multiple하여 작은 수로 바꾸어 준다 (이때 Multiple = Learning Rate라고 부른다)
- 따라서 4 - Learning Rate x Gradient하여 가장 작은 Cost 값을 찾는 것이 Gradient Decent의 방식이다
- 하지만, 이렇게 수치해석적으로 분석하면 소수점이 날아가는 등의 문제가 있을 수 있으므로, 편미분을 통해서 공식을 한번에 변형시키면 w값만 대입하면, 쉽게 구할 수 있다

$$cost(W) = \frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

$$\Downarrow$$

$$cost(W) = \frac{1}{2m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

$$cost(W) = \frac{1}{2m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha\frac{\partial}{\partial W}cost(W)$$

$$W := W - \alpha\frac{\partial}{\partial W}\frac{1}{2m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

$$W := W - \alpha\frac{1}{2m}\sum_{i=1}^{m}2(Wx^{(i)} - y^{(i)})x^{(i)}$$

$$W := W - \alpha\frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})x^{(i)}$$

미분 과정을 웹사이트에서 계산해주는 곳도 있음

(Derivative Calculator • With Steps! (derivative-calculator.net))

- Gradient Decent의 또 다른 단점은 기울기가 0인 지점에 도달했을 때 Local Minimum에 갇혀버리는 문제가 발생할 수 있다 (중간에 멈춰버려서 최종 Minimum Cost지점을 찾지 못한다는 것이다) 따라서 이에 대한 Optimizator를 사람들이 따로 만들어놓은 것도 있다
- Gradient Decent의 또 다른 특징은 시작지점에 따라 종착지가 달라질 수 있다는 점이 있다 따라서, 보통은 여러 번 시도를 해서 평균값을 선택하는 등의 방식을 사용한다

3. Multivariable Linear Regression

- 여러 개의 변수들을 활용하여 Y를 예측하는 Regression

예제) 앞의 예제에서 Quiz H(x) = Wx + b라 할때, 만약, 변수가 여러 개 (X_1, X_2, X_3)로 Y값을 예측한다고 한다면, Hypothesis와 Cost Function은 다음과 같이 정의할 수 있다

- $H(X_1, X_2, X_3) = w_1X_1 + w_2X_2 + w_3X_3 + b$

$$cost(W, b) = \frac{1}{m}\sum_{I=1}^{m}(H(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}) - y^{(i)})^2$$

이때, $H(X_1, X_2, X_3 ... X_n)$을 n차원 벡터라 할 수 있고, 이를 선형대수로 표현하면 다음과 같이 표현할 수 있다

$$\begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} x_1w_1 + x_2w_2 + x_3w_3 \end{pmatrix}$$

$$H(X) = XW$$

- (여기서 XW로 해주는 까닭은 행렬 계산시, WX는 전치행렬로 바꾸어주어야 하는 번거로움이 있기 때문이다)
- 이를 활용하여 다양한 레코드들의 Matrix값을 구할 수 있다

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} x_{11}w_1 + x_{12}w_2 + x_{13}w_3 \\ x_{21}w_1 + x_{22}w_2 + x_{23}w_3 \\ x_{31}w_1 + x_{32}w_2 + x_{33}w_3 \\ x_{41}w_1 + x_{42}w_2 + x_{43}w_3 \\ x_{51}w_1 + x_{52}w_2 + x_{53}w_3 \end{pmatrix}$$

[5, 3] **[3, 1]** [5, 1]

그리고 Y값이 한개가 아니라 여러개일 경우, n차원의 값들을 예측해야 하는데, 이때는 w 값을 n차원으로 변경시키면 된다

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \end{pmatrix} \cdot \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix} = \begin{pmatrix} x_{11}w_{11} + x_{12}w_{21} + x_{13}w_{31} & x_{11}w_{12} + x_{12}w_{22} + x_{13}w_{32} \\ x_{21}w_{11} + x_{22}w_{21} + x_{23}w_{31} & x_{21}w_{12} + x_{22}w_{22} + x_{23}w_{32} \\ x_{31}w_{11} + x_{32}w_{21} + x_{33}w_{31} & x_{31}w_{12} + x_{32}w_{22} + x_{33}w_{32} \\ x_{41}w_{11} + x_{42}w_{21} + x_{43}w_{31} & x_{41}w_{12} + x_{42}w_{22} + x_{43}w_{32} \\ x_{51}w_{11} + x_{52}w_{21} + x_{53}w_{31} & x_{51}w_{12} + x_{52}w_{22} + x_{53}w_{32} \end{pmatrix}$$

[n, 3] **[3, 2]** [n, 2]