



딥러닝 홀로서기#6


주제 : Binary Classification / Multinomial Classification

링크 :

[#6.Lec] Binary / Multi-Label Classification - 딥러닝 홀로서기

발표자료 링크 : https://github.com/heartcored98/Standalone-DeepLearning/blob/master/Lec2/Lec2-A_before_class.pdf자료 저장소

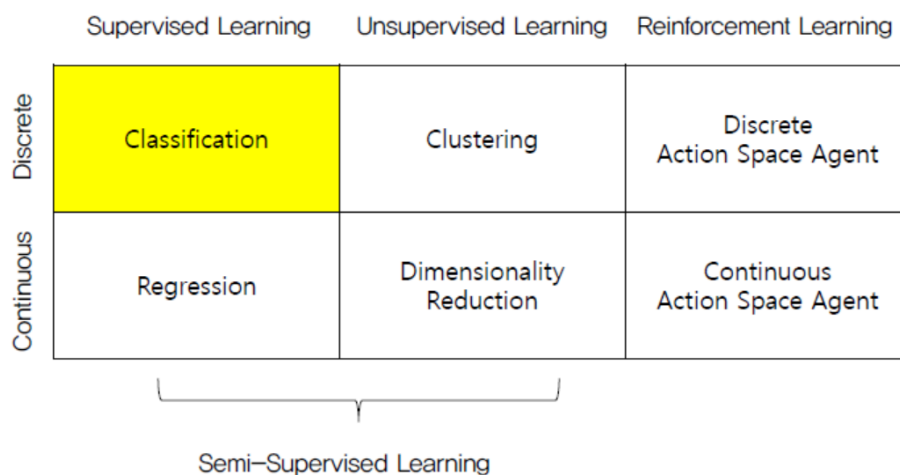
링크 : https://github.com/heartcored98/Standalone-DeepLearning/blob/master/Lec2/Lec2-A_before_class.pdf

 https://www.youtube.com/watch?v=is_Vw-aJMc4&list=PLSAJwo7mw8jn8iaXwT4MqLbZnS-LJwnBd&index=5&t=13s



1. Classification

- Classification : Supervised Learning의 한 종류로 Input X를 통해 Y를 분류하는 Machine Learning 기법
- Output Y가 Discrete한지 Continuous한지에 따라서 Classification / Regression의 방식으로 분류를 수행한다



2. Binary Classification

- 정의 : 분류를 해야하는 결과물이 2가지인 경우

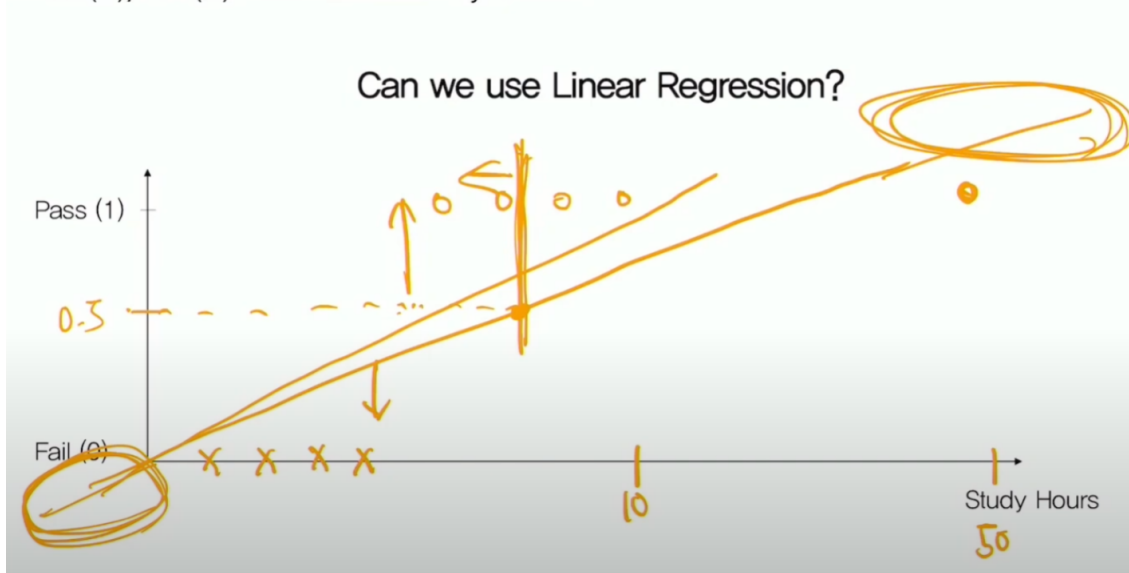
ex) 만약 공부한 시간에 따른 시험 Pass / Fail여부를 예측하는 Classification문제를 예를 들어보자

만약, Y값이 0.5 이상인 경우 Pass, 0.5이하인 경우 Fail이라 한다면, Linear Regression을 사용하여 이 문제를 풀 수도 있다

하지만 몇 가지 문제가 있다.

- ① Study 시간이 50시간인 학생이 발생했다고 하자, 이 학생은 당연히 Pass할 것이다
다만, 이에 따라 Regression 모델이 수정이 되어야 하며, 모델 수정에 따라 기존에 Fail로 분류되었던 사람들이 Pass로 분류될 수도 있고 Pass로 분류되었던 사람들이 Fail로 분류될 수도 있다
- ② Continuous한 모델이었기 때문에 1보다 큰 값 또는 0보다 작은 값이 나올 수도 있다

Pass(1)/Fail(0) based on study hours



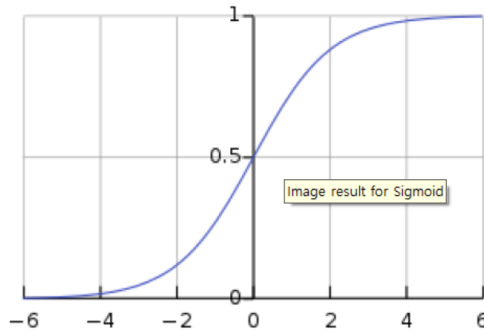
- Hypothesis

Binary Classification을 위한 Hypothesis는 기존 Regression에서 사용하던 식에 Sigmoid 함수를 적용하여 구현할 수 있다

$$H(x) = Wx + b$$

$$H(x) = G(Wx + b)$$

- Sigmoid (Logistic) Function



$$G(z) = \frac{1}{1 + e^{-z}}$$

이 식이 옳은지 증명을 해보자면,

- X가 양의 무한대로 간다면, $e^{-\infty}$ 는 0이 되기 때문에 $G(\infty) = \frac{1}{1+e^{-\infty}} = 1$ 로 수렴한다
- X가 음의 무한대로 간다면, e^{∞} 는 무한대가 되기 때문에 $G(-\infty) = \frac{1}{1+e^{\infty}} = 0$ 로 수렴한다

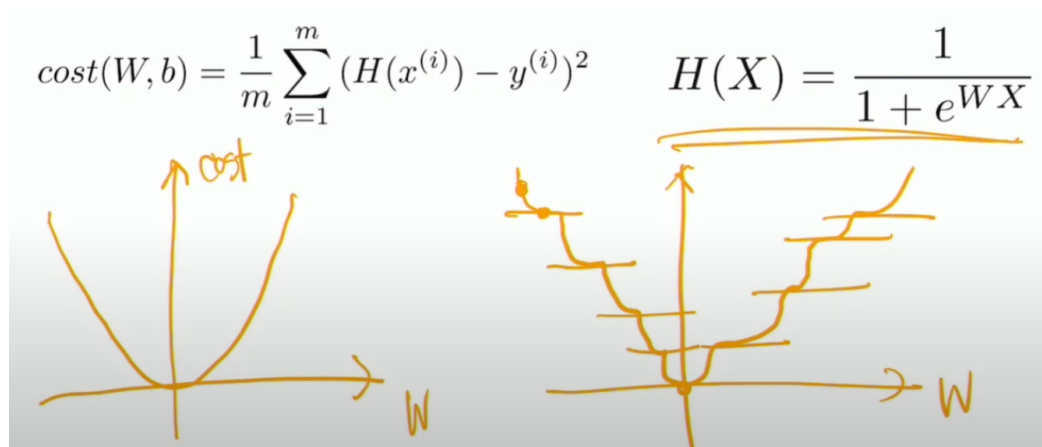
따라서 Hypothesis는 다음과 같이 정의될 수 있다

$$H(X) = \frac{1}{1+e^{-WX}}$$

• Cost Function

- Linear Regression에서 썼던 MSE Cost를 인용해서 사용할 수 있다

다만, W값에 따라서 Gradient가 0인 Saddle Point에(Local Minimum)에 갇혀 버릴 수도 있다



- Cross Entropy : 두 확률 분포 간의 차이를 나타내는 지표인데, 여기서 Saddle Point 문제를 해결하기 위한 새로운 Cost Function으로 도입하기로 한다
(화학에서 엔트로피 지수는 시스템이 얼마나 불안정한가를 나타내는 지표이다. 이 개념을 도입했다고 생각하면 쉽다)

$$H(P, Q) = - \sum P(x) \log(Q(x))$$

※ P(X) : 실제 확률, Q(X) : 예측 확률

예를 들어, 한국과 독일이 축구시합을 했다고 생각하자.

대부분의 사람들은 한국이 이길 확률을 굉장히 낮게 예측할 것이고 (여기서는 1%라 하자), 만약 실제로는 한국이 독일을 이겼다면 (실제 확률 100%) 여기서 Cross Entropy는 0.01에 Log를 취하여 엄청 큰 값을 리턴할 것이다 (Log 함수에서 X값이 작으면 Y값은 엄청 크므로)

다시말해, 이 결과를 두고 Cross Entropy에서는 이 시스템은(예측을 잘 못한) 불안정하다고 판단한다는 것이다

만약, 반대의 경우(한국 vs 중국)이었다면, Log함수를 통해 작은 수가 리턴되고 이 시스템은 안정성이 있다고 판단할 것이다 (Cost Function의 역할은 잘 예측할 수록 적은 값을 리턴해 주는 역할이기 때문에 Cross Entropy으로 Cost Function을 사용하기에 적합하다)

▼ Cross Entropy Cost Function

다음과 같이 정의할 수 있다

(여기서 Y = 1 은 (Predict : True)를 나타내며, Y = 0 은 (Predict : False)를 나타낸다)

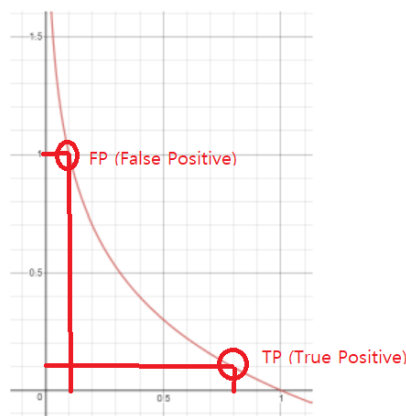
$$cost(W) = \frac{1}{m} \sum_{i=1}^n c(H(x^{(i)}), y^{(i)})$$

$$c(H(x), y) = \begin{cases} -\log(H(x)) & : y = 1 \\ -\log(1 - H(x)) & : y = 0 \end{cases}$$

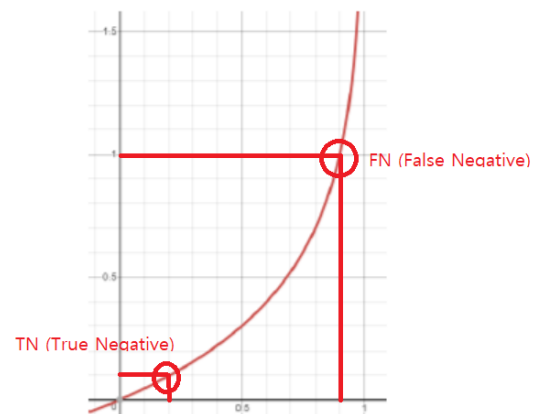
하나의 식으로 나타내면 다음과 같다

$$c(H(x), y) = -y \log(H(x)) - (1 - y) \log(1 - H(x))$$

실제로 $-\log(H(x))$ 와 $-\log(1 - H(x))$ 그래프를 그려보면 다음과 같다



$-\log(H(x))$ 그래프



$-\log(1 - H(x))$ 그래프

Y값이 1일때와 0일때 각각의 그래프에서 나타내는 결과치는 정확히 예측했을 때 낮은 값을 리턴한다

3. Multinomial Classification

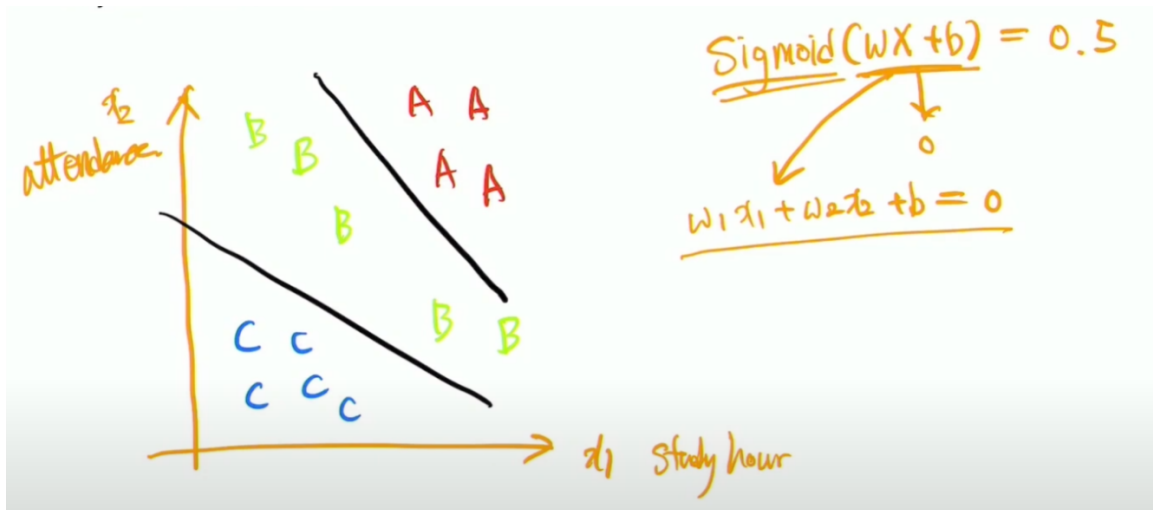
- 정의 : 분류 해야하는 Label이 두 개이상인 분류 문제를 말함

예제) Study Hour와 Attendance를 통해서 성적을 분류하는 문제가 있다. 성적의 Class는 A, B, C이렇게 세가지로 분류가 된다

이를 Binary Classification으로 설명을 한다고 하면, Sigmoid($wx + b$)으로 표현할 수 있으며, Sigmoid ($wx + b$)는 0과 1사이의 값이기 때문에 0.5보다 클 경우 1이라 생각하고 0.5보다 작으면 0이라 생각할 수 있다

이를 다시 풀어서 써보자면 Input이 2차원의 평면의 문제이기 때문에 $w_1x_1 + w_2x_2 + b = 0$ 이라 표현할 수 있다

그리고 각 Output을 결정짓는 Decision Boundary Line을 평면에 그을 수 있다



하지만 여기서의 문제는 Binary Classification의 경우에는 평면안에 선을 하나밖에 긋지 못하는데에 있다

A와 B 사이에 선을 긋는다면, B와 C를 분류할 수 없는 문제가 생기며, B와 C 사이에 선을 그으면 B와 A를 분류할 수 없는 문제가 생긴다

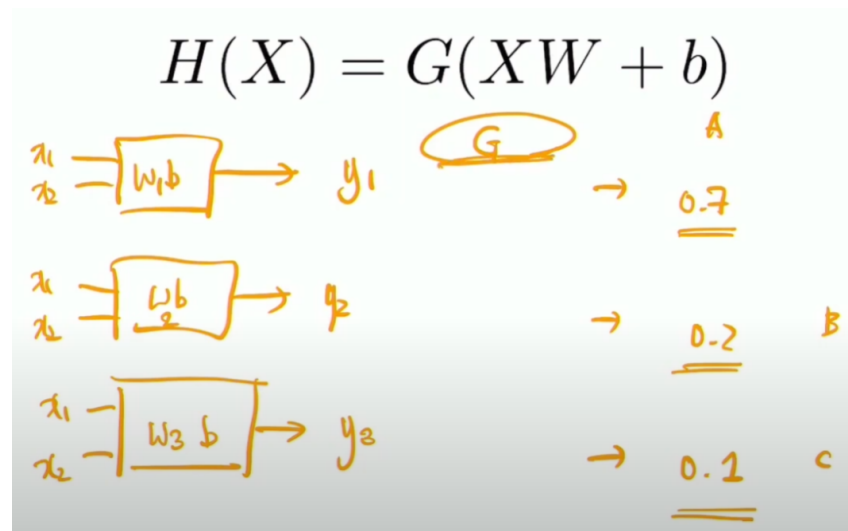
따라서, 이런 Multinomial Classification의 Hypothesis를 다음과 같이 재 정의하기로 하자

- Hypothesis

Binary Classification Hypothesis인 $H(X) = G(XW + b)$ 는 행렬 식으로 다음과 같이 표현될 수 있는데, 이를 여러번에 걸쳐서 Class마다 Continuous 값을 뽑아보자

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} (w_1 \ w_2) + b = x_1 w_1 + x_2 w_2 + b$$

그럼 다음과 같이 LinearRegression Model Y_1, Y_2, Y_3 가 나오게되고, 이는 $-\infty \sim +\infty$ 까지 Continuous 값을 표현하며, 이를 어떤 특정 함수 G를 통하면 특정 클래스와 대응될 확률로 매칭될 수 있다



이런 특정 함수 G를 정의한 것이 바로 Softmax 함수이며, Softmax함수는 Multinomial Classification을 위한 Hypothesis로 사용할 수 있다

$$\text{Softmax}(\hat{y}_i) = \frac{e^{\hat{y}_i}}{\sum_j e^{\hat{y}_j}}$$

$$= \frac{e^{y_1}}{e^{y_1} + e^{y_2} + e^{y_3}} = A \text{의 Prob.}$$

- Cost Function

예측한 값과 실제 값을 비교한 값이기 때문에, Softmax를 거쳐 나온 확률 값들을 실제 값과 비교한다고 했을시, Cross Entropy의 개념을 가져와 식을 정리하면 다음과 같다

$S(y)$	y
0.7	1
0.2	0
0.1	0

$$Cost(W, b) = \sum_i y_i \log(S(WX + b))$$

$$\hat{y} = WX + b$$

- Summary

	Binary	Multinomial
Hypothesis	$Sigmoid(WX)$	$Softmax(WX)$

$$Cost = -y \log(H(x)) - (1 - y) \log(1 - H(x)) \quad \sum -y \log(H(X))$$