< cost function : Logistic Regression vs Softmax Regression >

[Logistic Regression]

$z = \vec{w} \cdot \vec{x} + b$

$a_1 = g(z) = \frac{1}{1+e^{-z}} = P(y=1|\vec{x})$

$a_2 = 1 - a_1 = P(y=0|\vec{x})$

$loss = -y \log a_1 - (1-y) \log(1-a_1)$

Cost function : $J(\vec{w}, b)$ = average loss

[Softmax Regression]

$$a_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + \cdots + e^{z_N}} = P(y=1|\vec{x})$$

$$\vdots$$

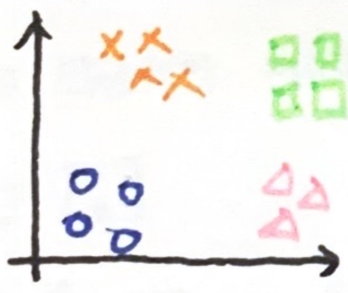
$$a_N = \frac{e^{z_N}}{e^{z_1} + e^{z_2} + \cdots + e^{z_N}} = P(y=N|\vec{x})$$

$$loss(a_1, \cdots, a_N, y) = \begin{cases} -\log a_1 & \text{if } y=1 \\ -\log a_2 & \text{if } y=2 \\ \vdots & \\ -\log a_N & \text{if } y=N \end{cases}$$

* One-hot Encoding

: 선택해야하는 class의 개수만큼의 차원을 가지면서 각 인덱스에 해당하는 원소에는 1, 나머지는 0값을 가지도록 하는 표현 방법

ex) Training set



→ -선택해야할 class 수 = (4) (예측) ※ class 수 = 차원 벡터

- 임의로 인덱스 부여 ⇒ X = 1, ○ = 2, □ = 3, △ = 4

"one-hot encoding" ⇒ [1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]

→ 각 클래스별로 부여된 인덱스에만 1, 나머지는 0

features				target	⇓
x_1	x_2	x_3	x_4	y	one-hot encoding
1	2	3	4	X	[1,0,0,0]
5	6	7	8	○	[0,1,0,0]
9	10	11	12	□	[0,0,1,0]
13	14	15	16	△	[0,0,0,1]
⋮	⋮	⋮	⋮	⋮	

ex) training example 1