

Neural Network Basic Assignment 1

이름: 이성범

1. Sigmoid Function을 z 에 대해 미분하세요.

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

1번 문제

1번 문제를 풀기전에 필수적으로 알아야 할 공식이 있다.
바로 몫의 미분법 공식이다.

* 몫의 미분법 공식

$$\frac{f(x)}{g(x)} = \frac{f'(x)g(x) - f(x)g'(x)}{g(x)^2}$$

= $\frac{\text{분자 미분} \cdot \text{분모} - \text{분자} \cdot \text{분모 미분}}{\text{분모 제곱}}$

$$\frac{d(1)}{dz} + \frac{d(e^{-z})}{dz} = 0 + -e^{-z} = -e^{-z}$$

* 몫의 미분법 공식을 가지고 시그모이드 함수를 미분

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

$$\frac{d}{dz} \sigma(z) = \frac{\frac{d(1)}{dz} \cdot (1 + e^{-z}) - 1 \cdot \frac{d(1 + e^{-z})}{dz}}{(1 + e^{-z})^2}$$

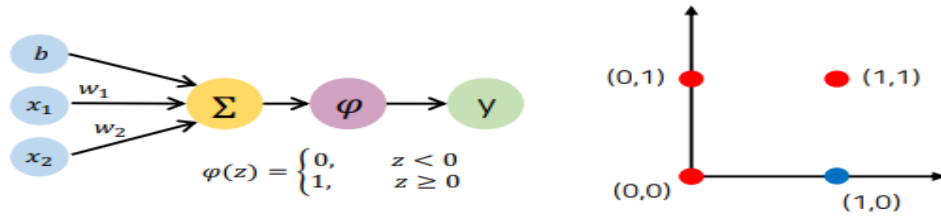
$$= \frac{0 - (-e^{-z})}{(1 + e^{-z})^2}$$

$$= \frac{e^{-z}}{(1 + e^{-z})^2}$$

미분원칙을
보기 좋게 정리하기 위해
분자에 1을 더한 배배준다.

$$\begin{aligned} &= \frac{1 + e^{-z} - 1}{(1 + e^{-z})^2} \\ &= \frac{1 + e^{-z}}{(1 + e^{-z})^2} - \frac{1}{(1 + e^{-z})^2} \\ &= \frac{1}{1 + e^{-z}} - \frac{1}{(1 + e^{-z})^2} \\ &= \left(\frac{1}{1 + e^{-z}}\right) \left(1 - \frac{1}{1 + e^{-z}}\right) \\ &= \sigma(z) \cdot (1 - \sigma(z)) \end{aligned}$$

2. 다음과 같은 구조의 Perceptron과 ● (=1), ● (=0)을 평면좌표상에 나타낸 그림이 있습니다.



2-1. ●, ●를 분류하는 임의의 b, w 를 선정하고 분류해보세요.

2-1번 문제

$$b=1, w_0=1, w_1=-1.5, w_2=1$$

$$\varphi(w_0 b + w_1 x_1 + w_2 x_2) = \gamma \quad \varphi(z) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0 \end{cases}$$

| x_1 | x_2 | z | \hat{y} | y |
|-------|-------|---|-----------|-----|
| 0 | 0 | $1.1 + (-1.5 \cdot 0) + (1 \cdot 0) = 1$ | 1 | 1 |
| 0 | 1 | $1.1 + (-1.5 \cdot 0) + (1 \cdot 1) = 2$ | 1 | 1 |
| 1 | 0 | $1.1 + (-1.5 \cdot 1) + (1 \cdot 0) = -0.5$ | 0 | 0 |
| 1 | 1 | $1.1 + (-1.5 \cdot 1) + (1 \cdot 1) = 0.5$ | 1 | 1 |

분류가 제대로 된 것을 알 수 있음

2-2. Perceptron 학습 규칙에 따라 임의의 학습률을 정하고 b, w 를 1회 업데이트 해주세요.

2-2번 문제

$$b=1, w_0=1, w_1=-1.5, w_2=1, \eta=0.1$$

$$w_i = w_i + \eta (y - \hat{y}) x_i$$

$$t=1, x_1=0, x_2=0, y=1 \text{ 일 때}$$

$$w_0 \leftarrow 1 + 0.1 (1 - 1) \cdot 1 = 1$$

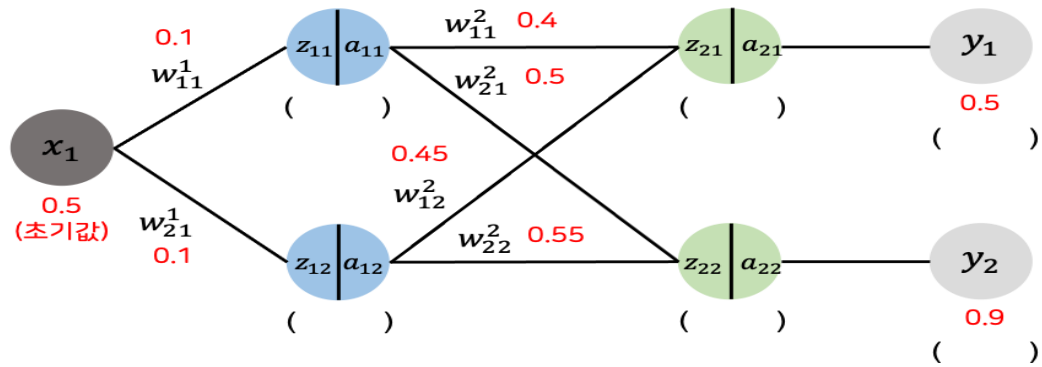
$$w_1 \leftarrow -1.5 + 0.1 (1 - 1) \cdot 0 = -1.5$$

$$w_2 \leftarrow 1 + 0.1 (1 - 1) \cdot 0 = 1$$

↳ 분류가 제대로 되었기 때문에 가중치 업데이트는 일어나지 않았다.

이제 위의 같은 방식으로 (0,1), (1,0), (1,1) 순으로 가중치를 업데이트 해가면 된다.

3. 다음과 같은 구조와 초기값을 가진 Multilayer Perceptron이 있습니다.



3-1. Forward Propagation이 일어날 때, 각 노드는 어떤 값을 갖게 되는지 빈 칸을 채워주세요.
(Sigmoid Function 사용)

```

1 # 3-1 번 문제
2 import numpy as np
3
4 # 시그모이드 함수 구현
5 def sigmoid(x):
6     return 1 / (1 + np.exp(-x))
7
8 x_1 = 0.5
9
10 w_1_11 = 0.1
11 w_1_21 = 0.1
12
13 z_1_1 = x_1 * w_1_11
14 z_1_2 = x_1 * w_1_21
15
16 a_1_1 = sigmoid(z_1_1)
17 a_1_2 = sigmoid(z_1_2)
18
19 w_2_11 = 0.4
20 w_2_21 = 0.5
21 w_2_12 = 0.45
22 w_2_22 = 0.55
23
24 z_2_1 = (a_1_1 * w_2_11) + (a_1_2 * w_2_12)
25 z_2_2 = (a_1_1 * w_2_21) + (a_1_2 * w_2_22)
26
27 y1 = sigmoid(z_2_1)
28 y2 = sigmoid(z_2_2)
29
30 y1_true = 0.5
31 y2_true = 0.9
32
33 print(f'a_1_1 : {a_1_1}')
34 print(f'a_1_2 : {a_1_2}\n')
35 print(f'z_2_1 : {z_2_1}')
36 print(f'z_2_2 : {z_2_2}\n')
37 print(f'y1 : {y1}')
38 print(f'y2 : {y2}')
39
40 두번째 레이어의 경우
41 활성화 함수 이후의 출력 값을
42 적으면 y의 값과 같아지기 때문에
43 활성화 함수 이전의
44 출력 값을 출력했습니다.
45
a_1_1 : 0.5124973964842103
a_1_2 : 0.5124973964842103
z_2_1 : 0.4356227870115788
z_2_2 : 0.5381222663084209
y1 : 0.6072155356954312
y2 : 0.6313755010102448

```

3-2. output layer에 있는 노드들의 Mean Squared Error를 구해주세요.

```

1 # 3-2번 문제
2 def MSE(y_true, y_pred):
3     mse = ( 1/len(y_pred) ) * ( 1/2*np.sum((y_true - y_pred)**2) )
4     return mse
5
6 y_true = np.array([y1_true, y2_true])
7 y_pred = np.array([y1, y2])
8 print(f"MSE : {MSE(y_true, y_pred)}")

```

MSE : 0.020913573137988827

3-3. 3-2에서 구한 답을 토대로, Back Propagation이 일어날 때 가중치 w_{11}^1 과 w_{11}^2 의 조정된 값을 구해주세요. (학습률 $\eta = 0.5$)

```
1 # 3-3번 문제
2
3 lr = 0.5
4
5 dE = -(y1_true - y1)
6 ds = y1 * (1 - y1)
7 dz = a_1_1
8
9 dw = dE * ds * dz
10
11 w_2_11_update = w_2_11 - lr*dw
12
13 print(f'업데이트된 w_2_11: {w_2_11_update}')
14
15 dE1 = -(y1_true - y1) * y1 * (1 - y1) * w_2_11
16 dE2 = -(y2_true - y2) * y2 * (1 - y2) * w_2_21
17
18 dE = dE1 + dE2
19 ds = a_1_1 * (1 - a_1_1)
20 dz = X_1
21
22 dw = dE * ds * dz
23
24 w_1_11_update = w_1_11 - lr*dw
25
26 print(f'업데이트된 w_1_11: {w_1_11_update}')
```

```
업데이트된 w_2_11: 0.39344735614480647
업데이트된 w_1_11: 0.10131363680057552
```

참고자료

1. <https://wikidocs.net/37406>