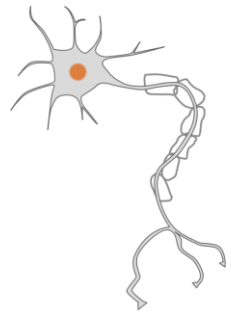
CNU 데이터 분석 교육



11th lecture

"Perceptron and neural network"

2022 - 11 - 21

수학과 최명수

지난시간?

1. Linear discriminant analysis(LDA)

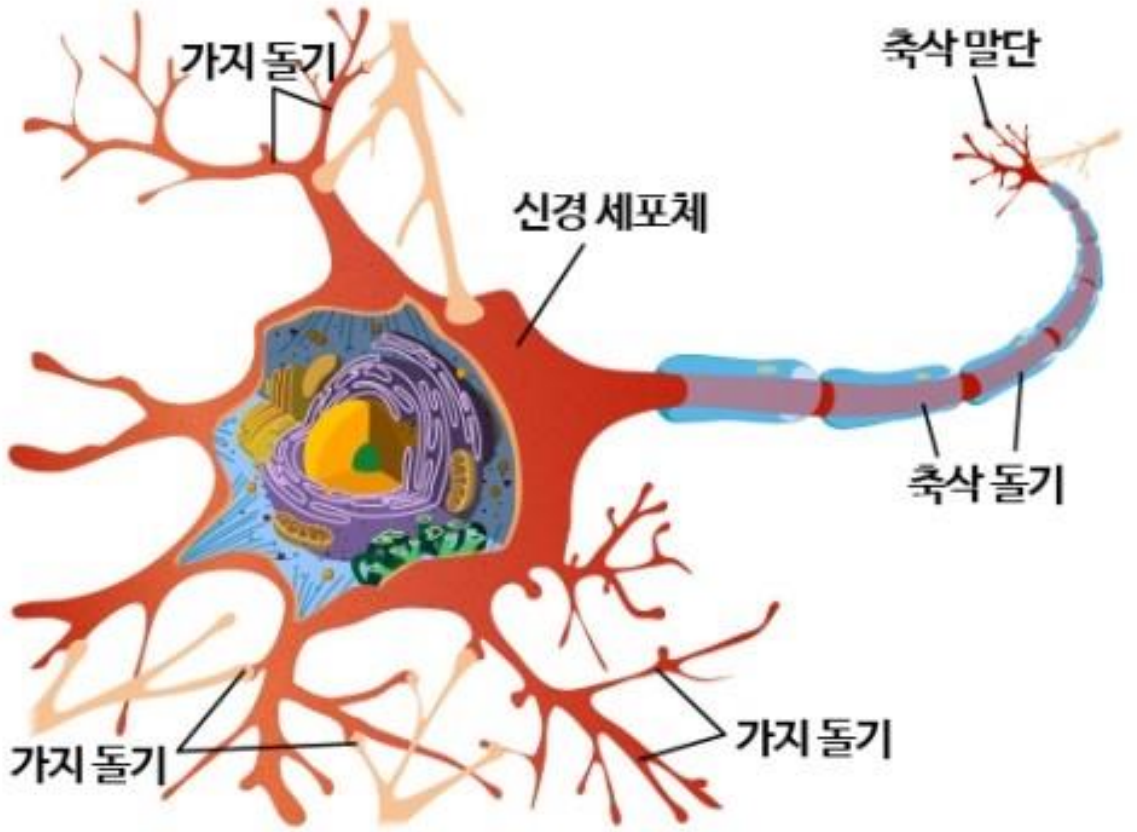
   - Dimension reduction
   - Find $\vec{w}$ that maximize $J(\vec{w}) = \frac{|\mu_1 - \mu_2|}{S_1^2 + S_2^2}$

오늘은 무엇을?

1. 인공신경망의 구조 살펴보기

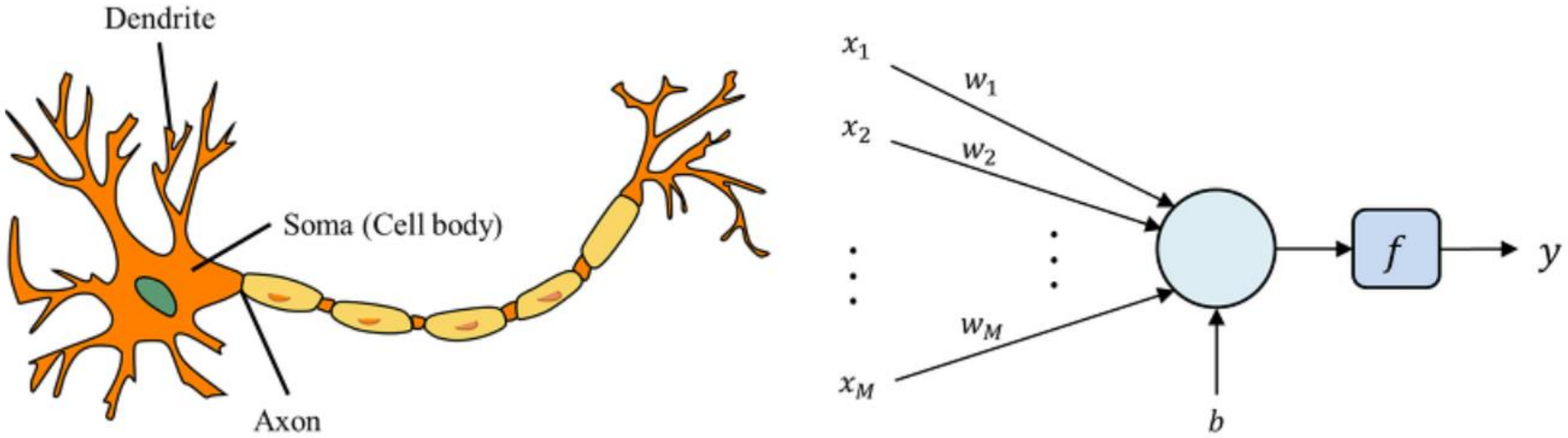
   - 단일 계층 퍼셉트론(Single layer perceptron)
   - 다중 계층 퍼셉트론(Multi layer perceptron)

Single layer perceptron



https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=moeblog&logNo=220446698924

Single layer perceptron

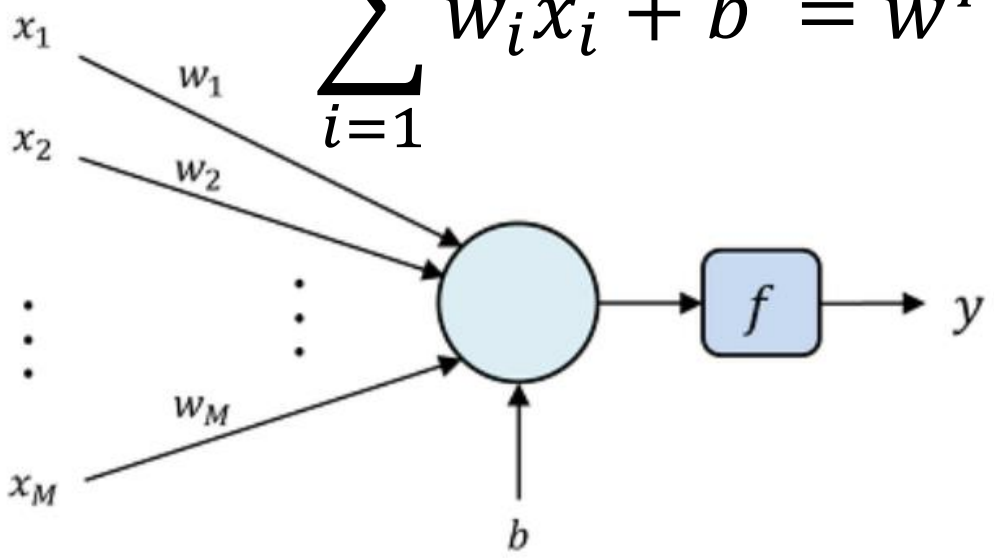


[그림 1] 생물체의 neuron (좌)과 artificial neuron (우)

https://smartstuartkim.wordpress.com/2019/01/27/history-of-neural-networks-1-perceptron/

Single layer perceptron

$$\sum_{i=1}^{M} w_i x_i + b = w^T x + b$$

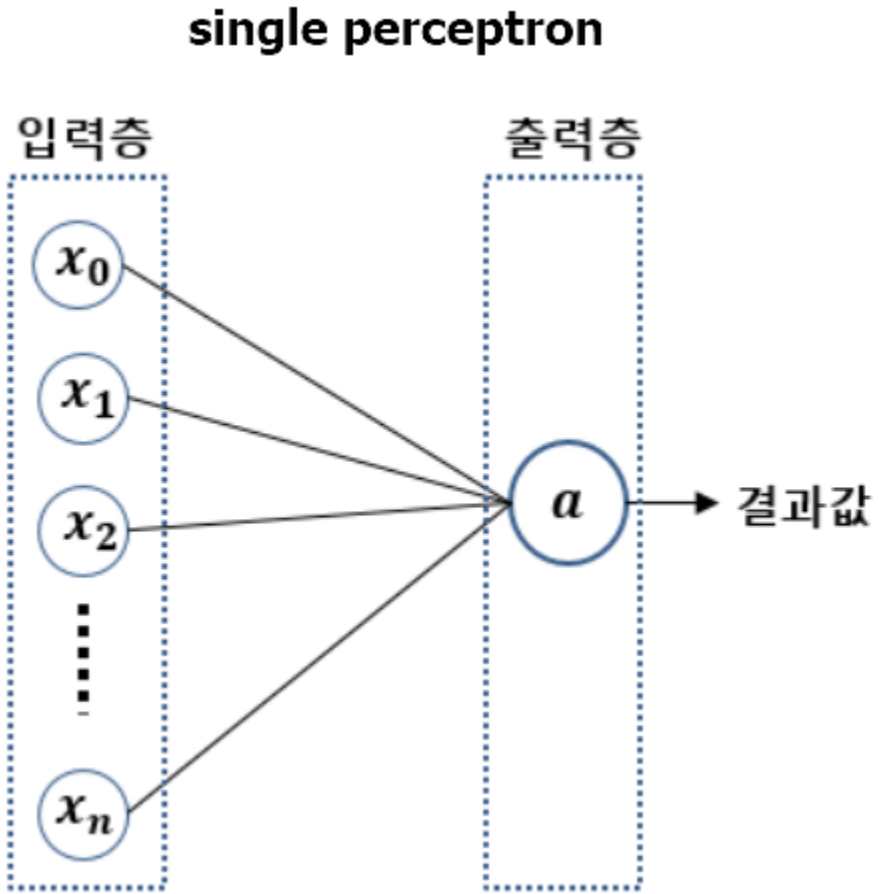


- x_i : data
- w_i : weight
- b : bias

- f : activation function

$$f(w^T x + b) = y$$

Single layer perceptron

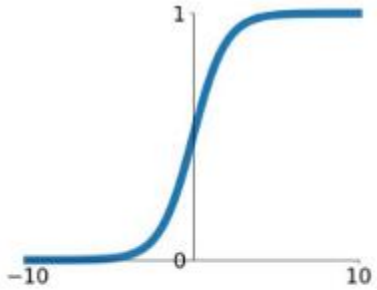


https://smartstuartkim.wordpress.com/2019/01/27/history-of-neural-networks-1-perceptron/

Single layer perceptron

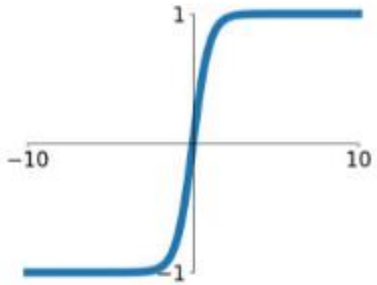
Sigmoid

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



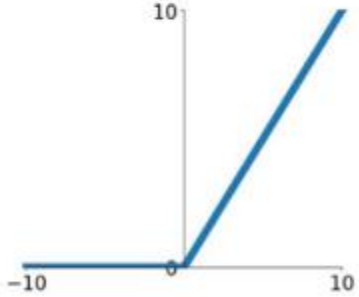
tanh

$$\tanh(x)$$



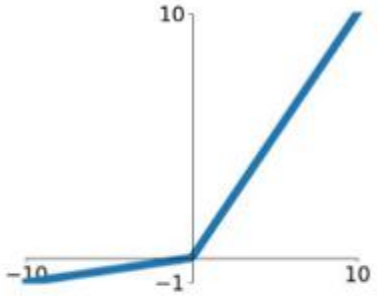
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

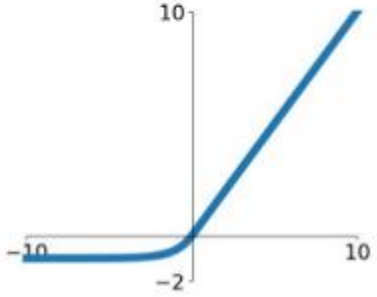


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

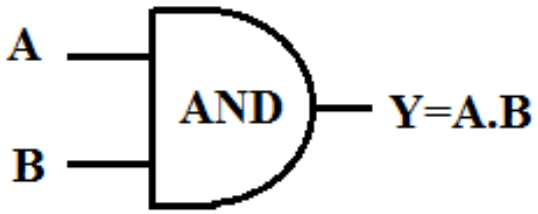
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



https://m.blog.naver.com/intelliz/221709291643

연습

AND gate



Inputs		Output
A	B	Y=A.B
0	0	0
0	1	0
1	0	0
1	1	1

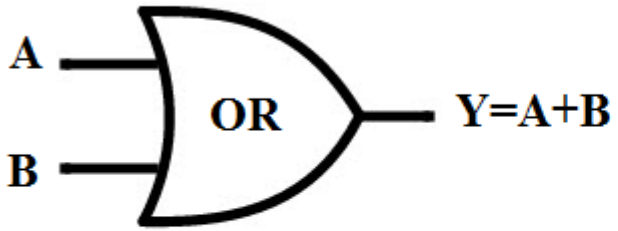
$$f(w^T x + b) = y$$

Find $w = (w_1, w_2)$ and b for step function f

https://electronics-club.com/logic-gates/

연습

OR gate



Inputs		Output
A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

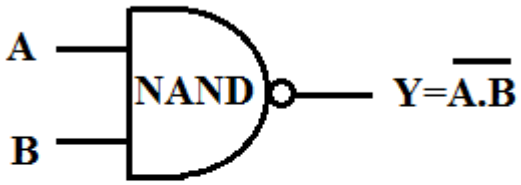
$$f(w^T x + b) = y$$

Find $w = (w_1, w_2)$ and b for step function f

https://electronics-club.com/logic-gates/

연습

NAND gate



Inputs		Output
A	B	$Y=\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

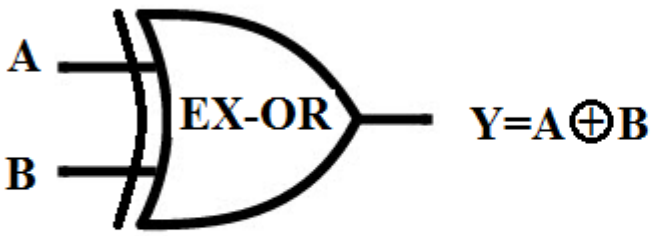
$$f(w^T x + b) = y$$

Find $w = (w_1, w_2)$ and b for step function f

https://electronics-club.com/logic-gates/

연습

XOR gate



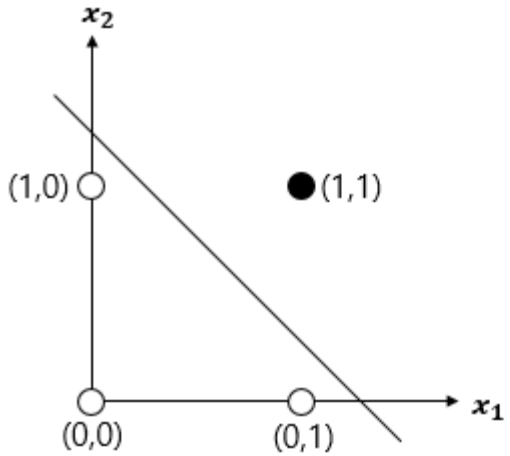
Inputs		Output
A	B	Y=A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

$$f(w^T x + b) = y$$

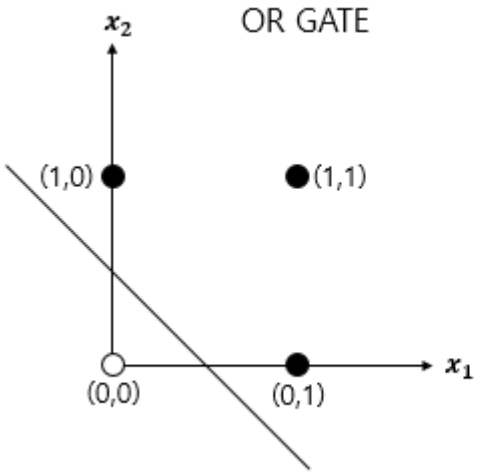
Find $w = (w_1, w_2)$ and b for step function f

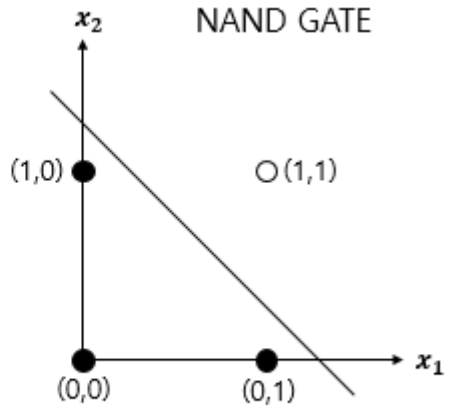
https://electronics-club.com/logic-gates/

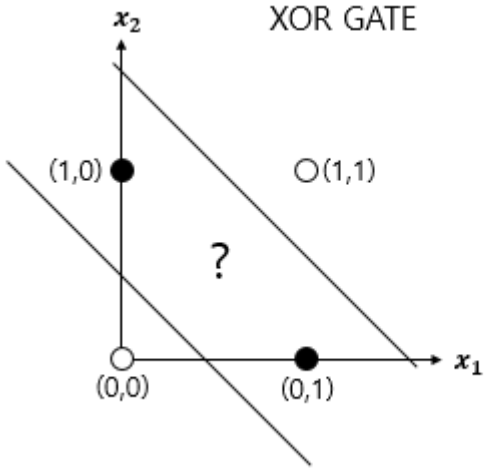
연습



x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1





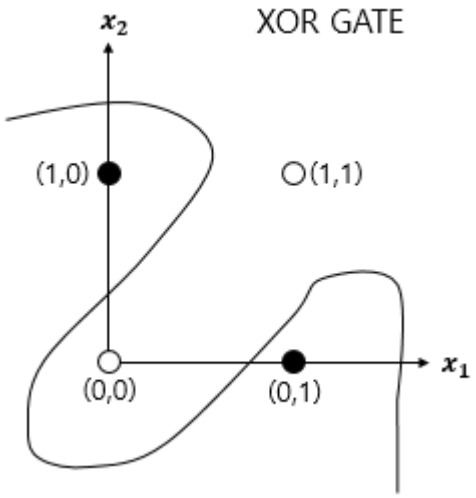


x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

https://wikidocs.net/24958

연습

Single layer perceptron 으로는 선형 영역에 대해서만 분리가 가능하다.
고로 비선형 영역으로 분리해야 한다.

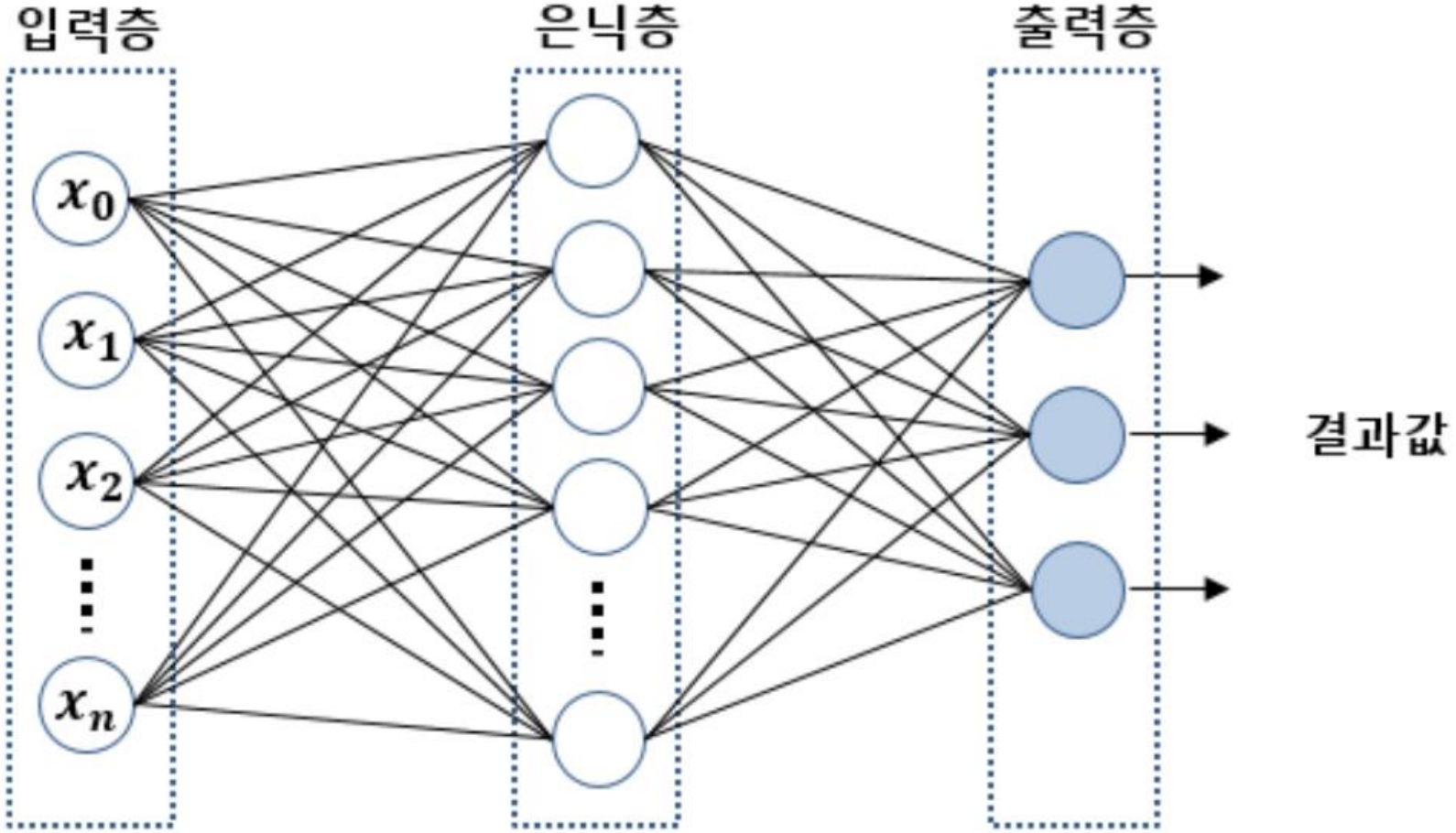


그럼 layer를 늘려볼까?

https://wikidocs.net/24958

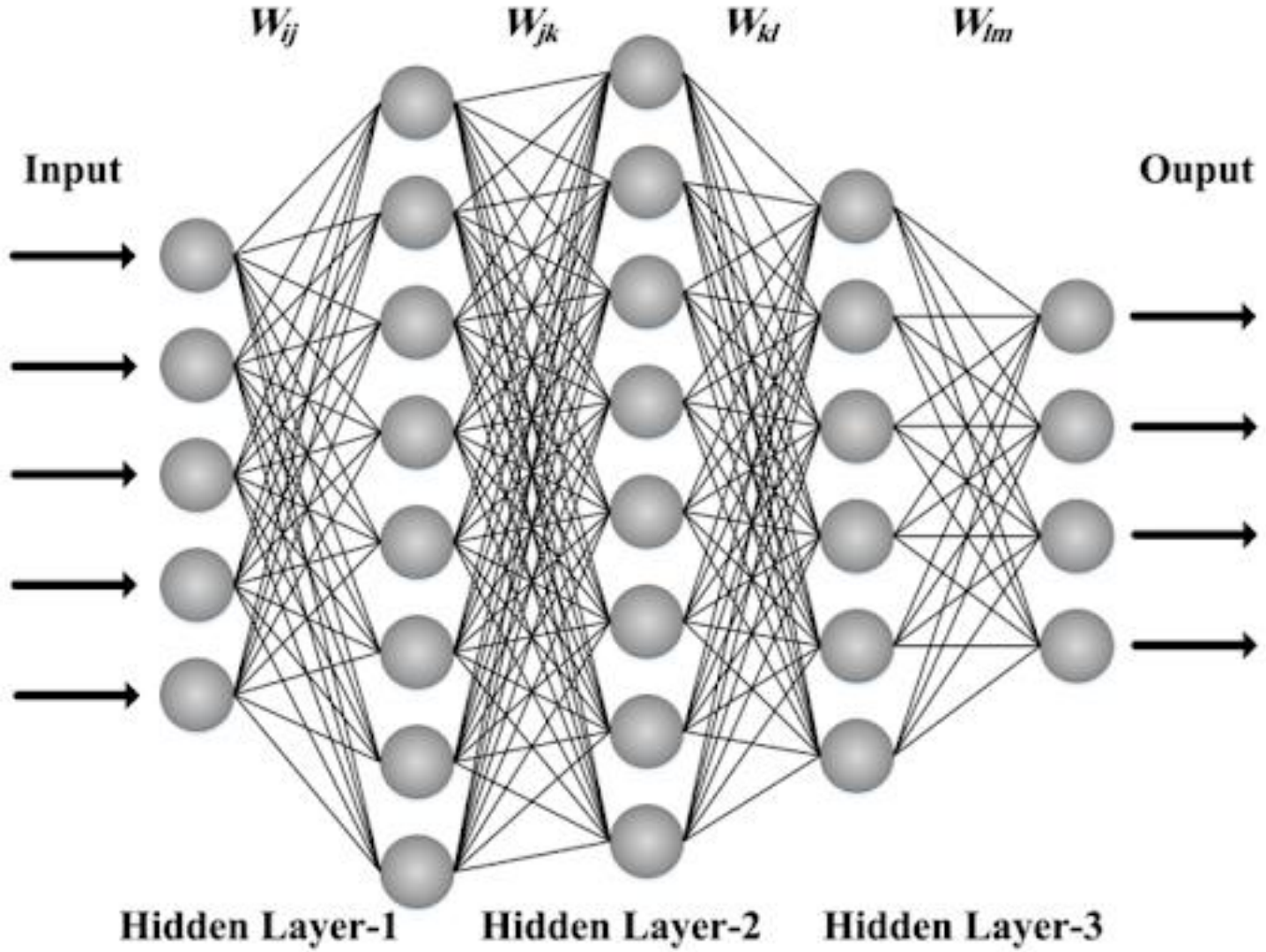
Multi layer perceptron

Deep Neural Network 라고도 하며,
Perceptron을 여러 개 쌓아 만들 모델을 의미한다.



https://smartstuartkim.wordpress.com/2019/01/27/history-of-neural-networks-1-perceptron/

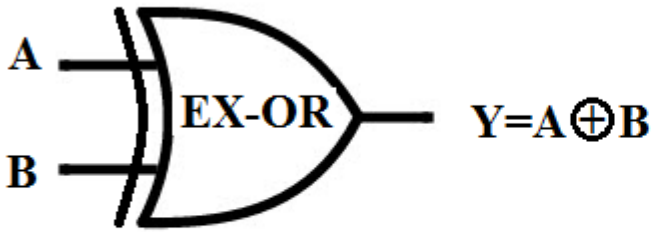
Multi layer perceptron



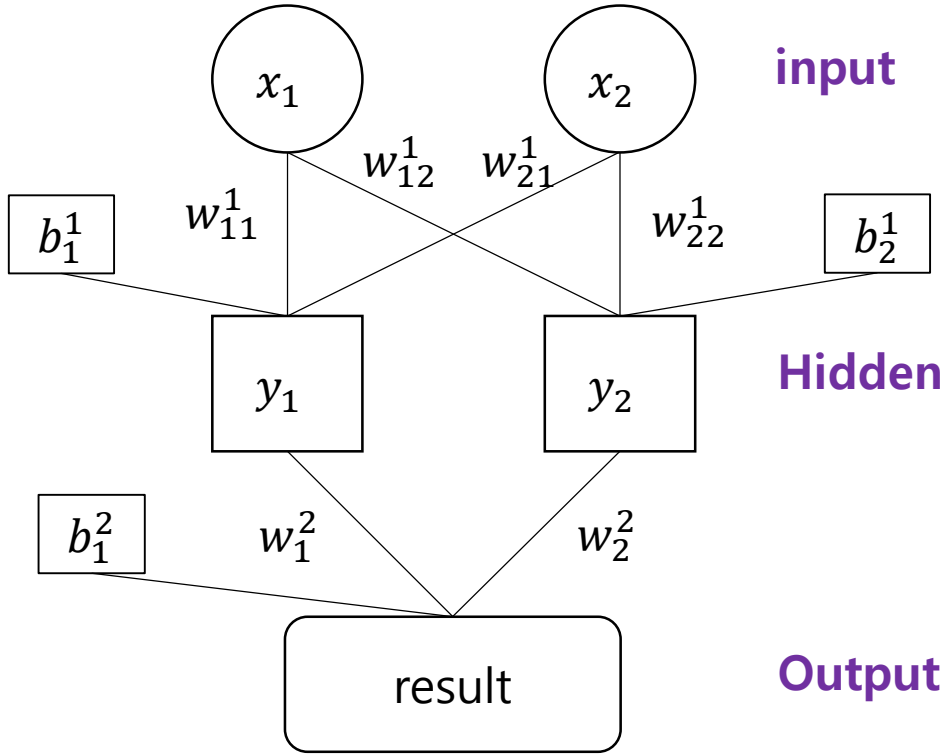
https://yceffort.kr/2019/01/27/pytorch-2-multi-perceptron(1)

연습

XOR gate

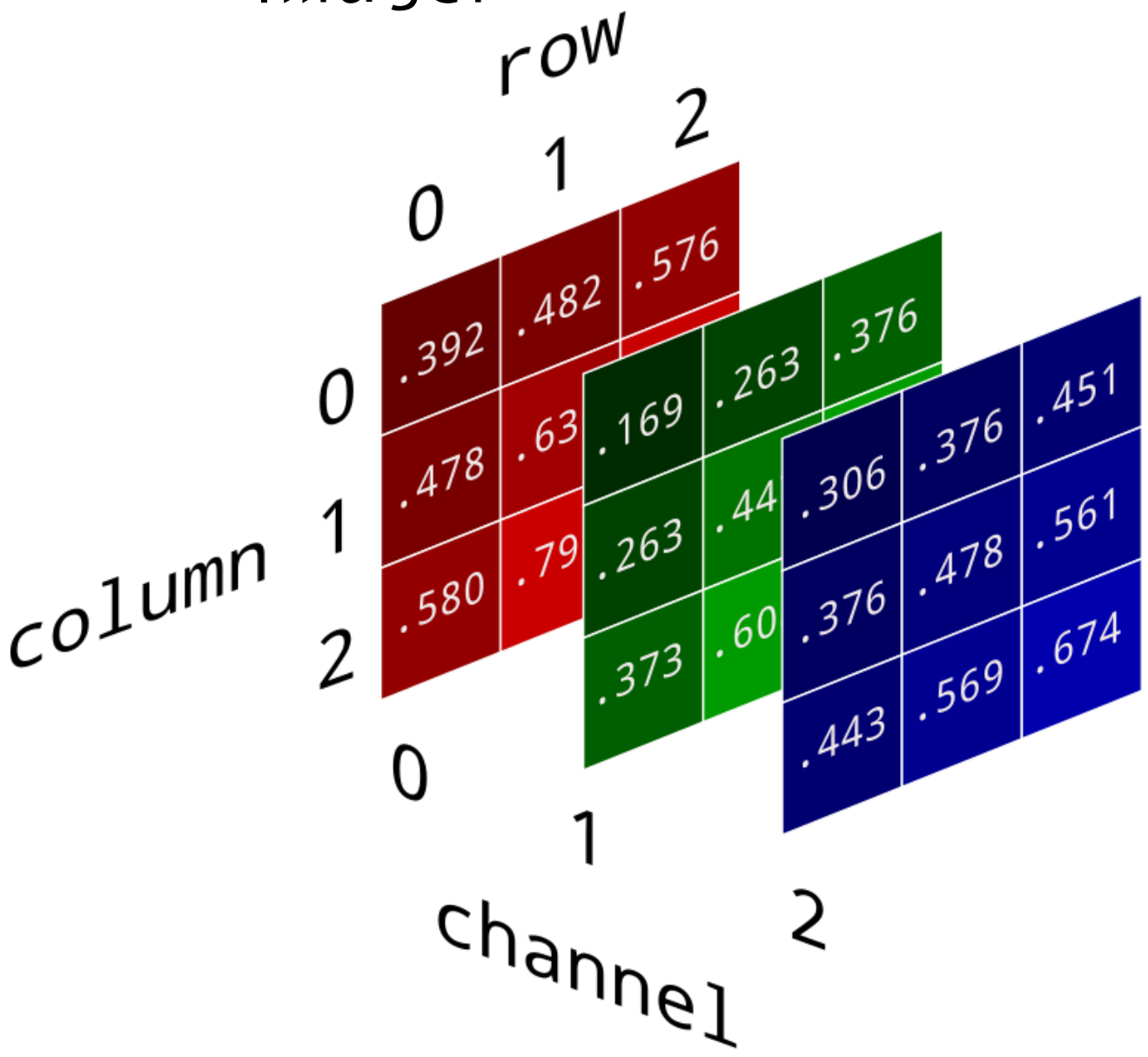


Inputs		Output
A	B	Y=A⊕B
0	0	0
0	1	1
1	0	1
1	1	0



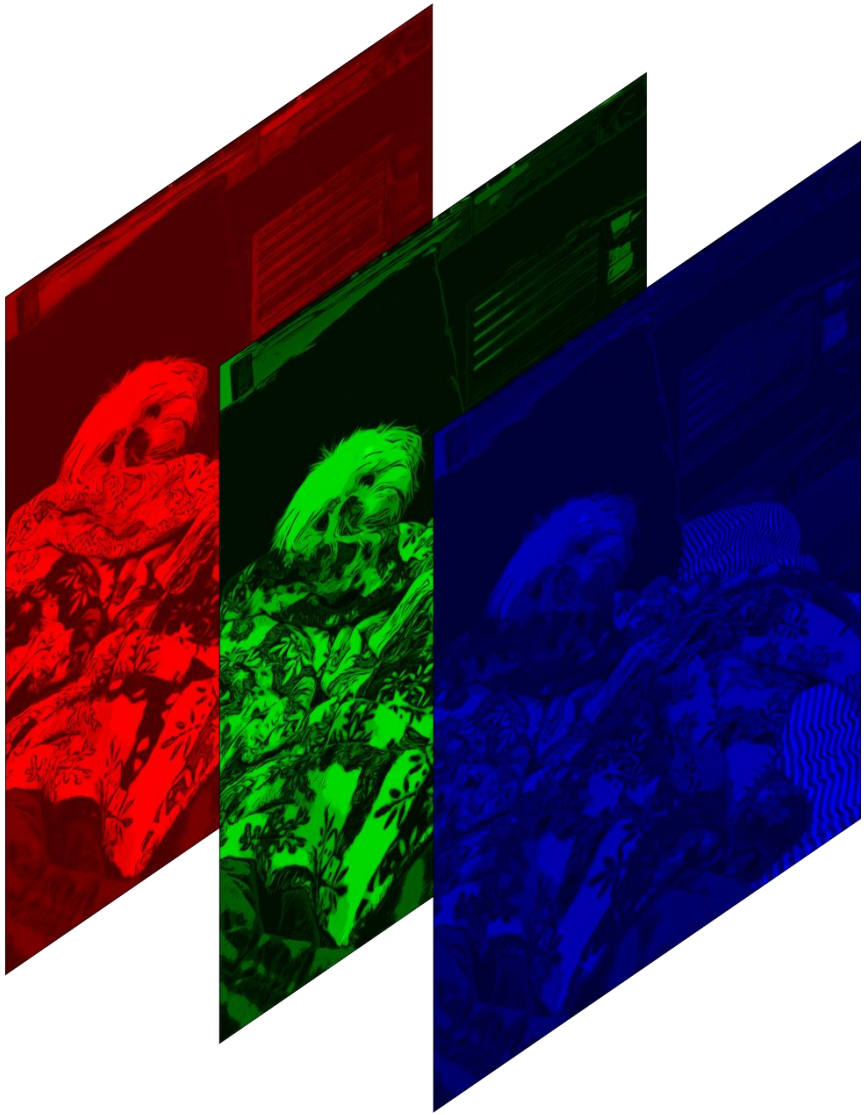
https://electronics-club.com/logic-gates/

Image?



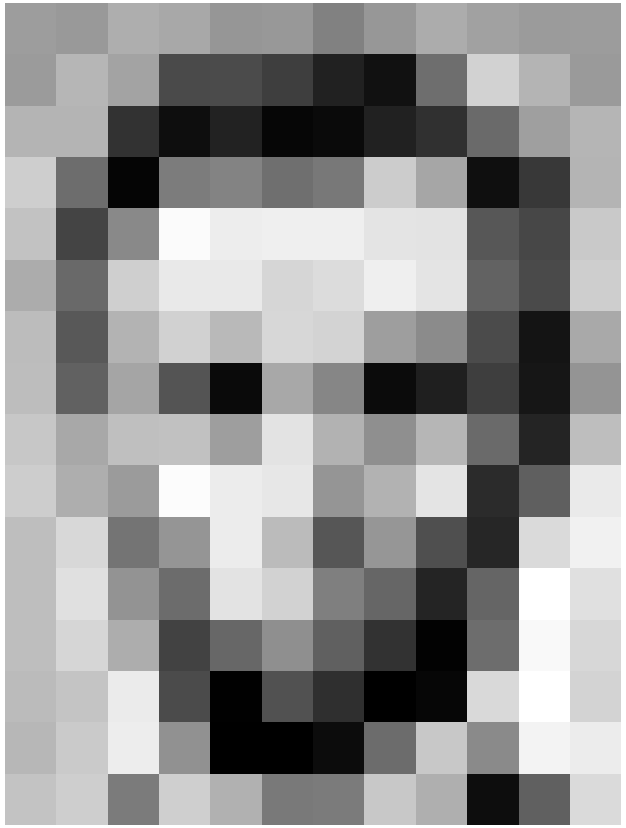
https://www.kdnuggets.com/2019/12/convert-rgb-image-grayscale.html

Image?



https://www.kdnuggets.com/2019/12/convert-rgb-image-grayscale.html

Image?



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

https://towardsdatascience.com/computer-vision-101-working-with-color-images-in-python-7b57381a8a54

Flattening

2차원 데이터를 1차원 데이터로 변환

3x3

9x1

https://towardsdatascience.com/computer-vision-101-working-with-color-images-in-python-7b57381a8a54

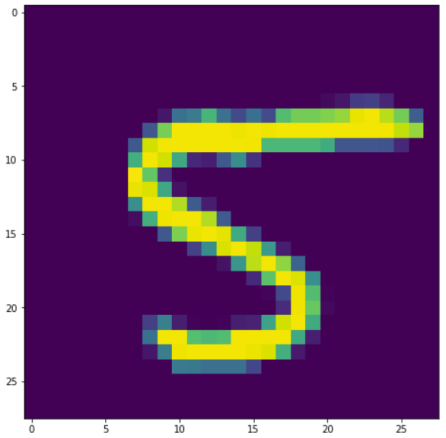
Softmax

$$\text{Softmax}\,\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

1. 모든 결과값의 합이 1
   즉 각각의 값을 확률로 해석 가능!

Softmax

Input :



$\sigma(Image)$

0	1	2	3	4	5	6	7	8	9
0.01	0.01	0.01	0.01	0.01	**0.91**	0.01	0.01	0.01	0.01

Input의 결과가 5일 확률이 가장 크다!

분류 모델의 평가 방법

오분류표, 혼돈행렬 (Confusion matrix)

Predicted Values \ Actual Values	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

https://wikidocs.net/123608

분류 모델의 평가 방법

- Accuracy (정확도)

모델이 제대로 예측한 비율: $\frac{TP+TN}{TP+FP+FN+TN}$

		실제	
		Y (1)	N (0)
예측	Y (1)	True Positive (TP)	False Positive (FP)
	N (0)	False Negative (FN)	True Negative (TN)

정확도

암 환자와 아닌 환자를 정확하게 판단하는 정도

https://yogyui.tistory.com/entry/Confusion-Matrix-%ED%98%BC%EB%8F%99%ED%96%89%EB%A0%AC

분류 모델의 평가 방법

- **Precision (정밀도)**

모델이 1로 예측한 결과 중 실제 데이터도 1인 경우의 비율: $\frac{TP}{TP+FP}$

'모델'의 관점

		실제	
		Y (1)	N (0)
예측	Y (1)	True Positive (TP)	False Positive (FP)
	N (0)	False Negative (FN)	True Negative (TN)

정밀도

의사가 암 환자라고 했는데, 그것이 실제 암 환자일 가능성

https://yogyui.tistory.com/entry/Confusion-Matrix-%ED%98%BC%EB%8F%99%ED%96%89%EB%A0%AC

분류 모델의 평가 방법

- **Recall (재현율)**

실제 값이 1인 데이터 중 모델이 1로 예측한 비율: $\frac{TP}{TP+FN}$

'데이터'의 관점

= **Sensitivity** (민감도) = **TPR** (True Positive Rate) = Hit Rate

		실제	
		Y (1)	N (0)
예측	Y (1)	True Positive (TP)	False Positive (FP)
	N (0)	False Negative (FN)	True Negative (TN)

재현율, 민감도

실제 암 환자중에서 의사가 암 환자를 판단할 수 있는 정도

https://yogyui.tistory.com/entry/Confusion-Matrix-%ED%98%BC%EB%8F%99%ED%96%89%EB%A0%AC

분류 모델의 평가 방법

- **Specificity (특이도)**

실제 값이 0인 데이터 중 모델이 0으로 예측한 비율: $\frac{TN}{FP+TN}$

'데이터'의 관점

= **TNR** (True Negative Rate)

		실제	
		Y (1)	N (0)
예측	Y (1)	True Positive (TP)	False Positive (FP)
	N (0)	False Negative (FN)	True Negative (TN)

정상 환자중에서 의사가 정상 환자를 판단할 수 있는 정도 특이도

https://yogyui.tistory.com/entry/Confusion-Matrix-%ED%98%BC%EB%8F%99%ED%96%89%EB%A0%AC

분류 모델의 평가 방법

- **F1 Score (F-Measure)**

정밀도와 재현율의 조화평균: $\frac{2}{\frac{1}{precision}+\frac{1}{recall}} = \frac{2\times TP}{2\times TP+FP+FN}$

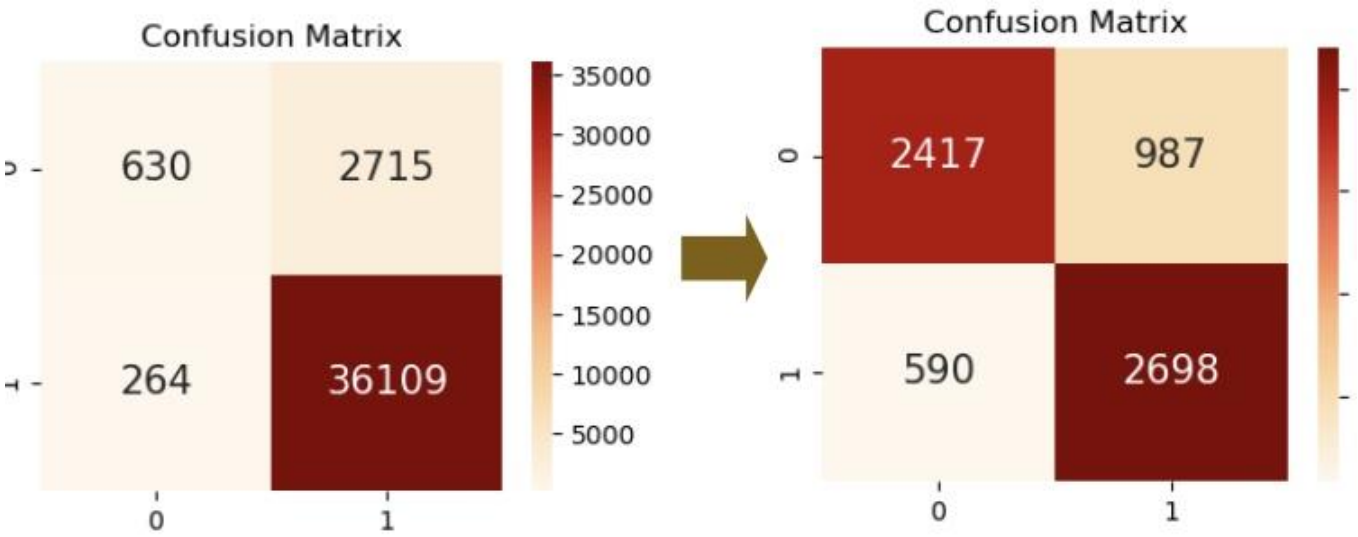
모델의 예측 성능을 수치화, 값의 범위는 0 ~ 1

정밀도 혹은 재현율 둘 중 하나만 높은 경우보다 정밀도와 재현율 둘 다 값이 클 경우 F1 점수는 1에 가까워진다 = 모델의 예측 성능이 좋다

※ F1-score까지는 이해를 하던 암기를 하던 필기시험 칠때는 무조건 머리에 집어넣자

https://yogyui.tistory.com/entry/Confusion-Matrix-%ED%98%BC%EB%8F%99%ED%96%89%EB%A0%AC

분류 모델의 평가 방법



조정 전	
Accuracy (정확도)	0.92
Precision (정밀도)	0.93
Recall (재현율)	0.99
F1	0.96

조정 후	
Accuracy (정확도)	0.76
Precision (정밀도)	0.73
Recall (재현율)	0.82
F1	0.77

True와 False의 비율을 5:5로 해줘야 한다.

정리 요약

1. 신경망은 Perceptron 으로부터 출발한다.

   - 생물 신경 구조를 본 따 만든 모델.
   - 단일로는 선형적인 모델밖에 안됨.

2. Perceptron을 여러 개 나열한것이 Neural network 이다.

   - 이것이 바로 딥 러닝(Deep learning) 모델이다.
   - 비선형적인 모델제작 가능

공지

1. 신경망은 어떻게 학습을 할까?
   -Backpropagation

끝

담에봅시당