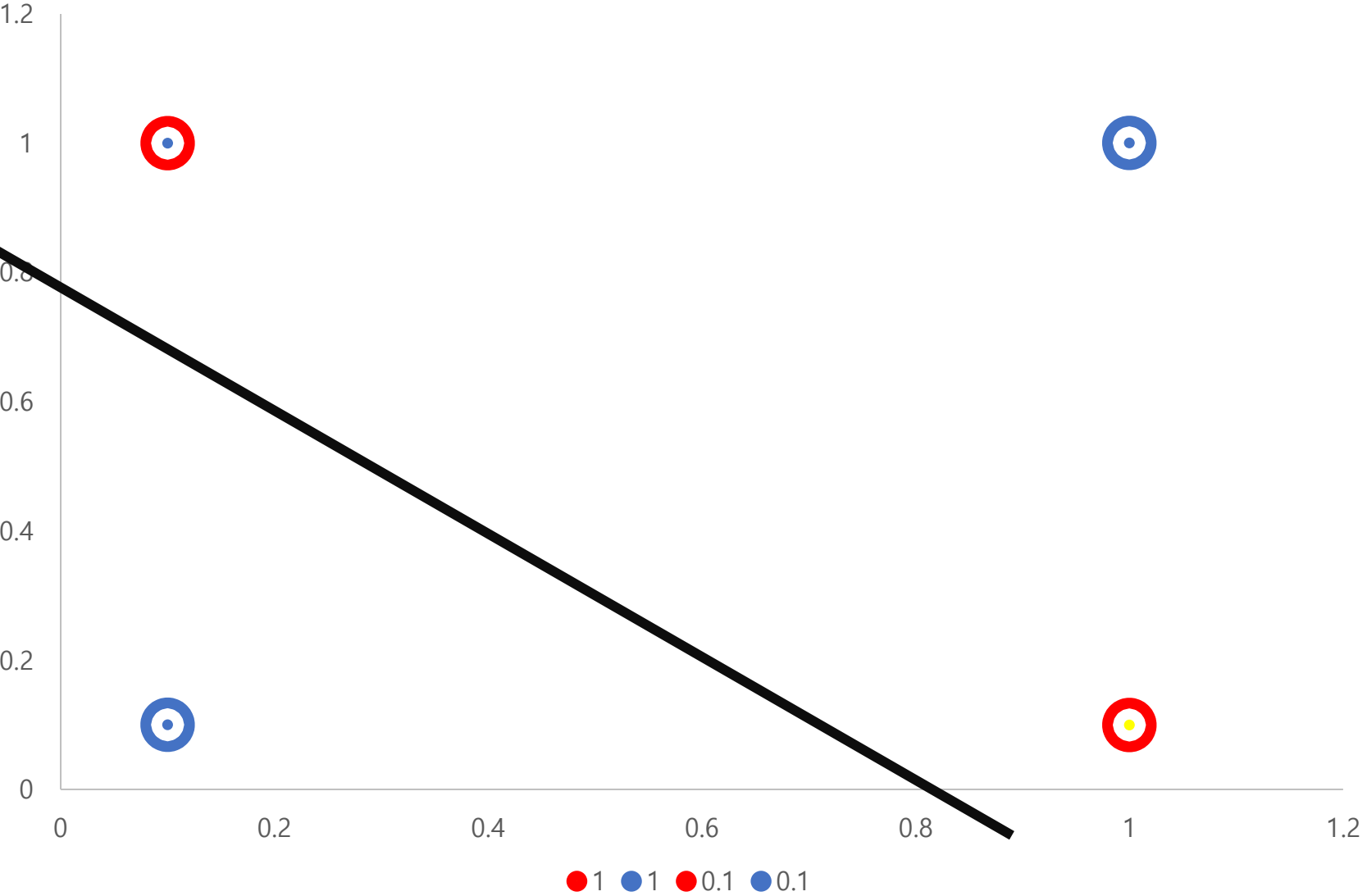


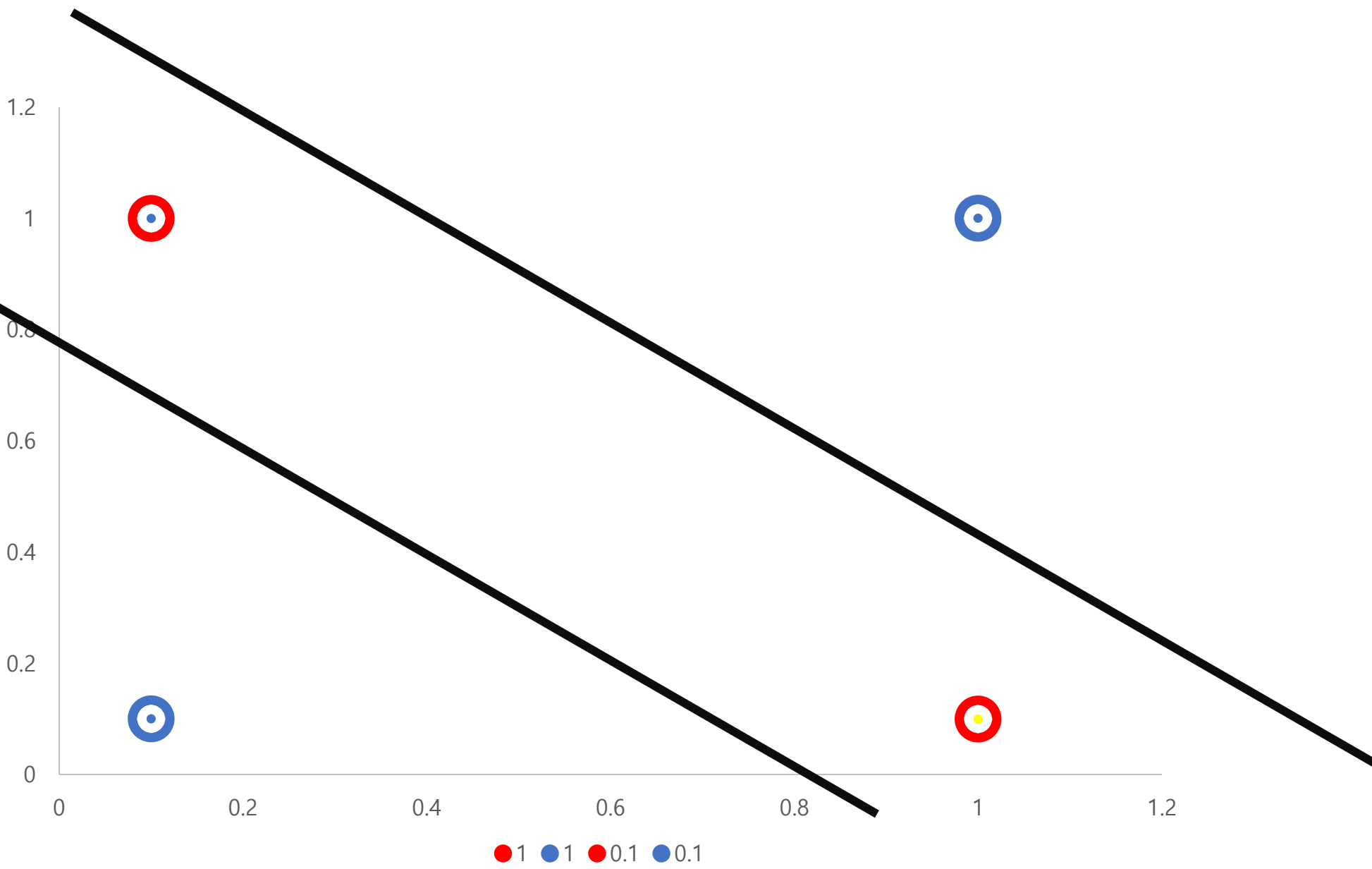
3주차 발표자료

이지환

XOR 문제

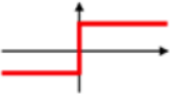
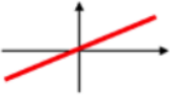
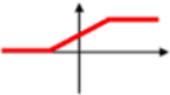
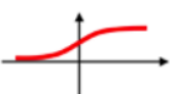




XOR 문제



퍼셉트론 (Perceptron)

- 퍼셉트론이란?: 인공 신경망의 구성 요소로서, 여러 입력 값을 하나의 출력 값으로 반환하는 역할을 한다.
- Perception (지각 능력) + neuron (뇌의 신경 세포)
- 퍼셉트론은 인간 신경계의 기본 단위인 뉴런을 본떠 만들었으며 통계적으로 모델링한 알고리즘이다.
- 뉴런과 퍼셉트론 비교
 - 뉴런: 저장한 신호 또는 자극의 크기가 임계값을 넘어갈 경우 축삭 돌기를 통해 외부로 전달
 - 퍼셉트론: 입력된 값과 가중치의 곱(+편차)을 활성화 함수의 임계값과 비교하여 출력 값을 결정
 - > ex. 즉, 활성화 함수는 퍼셉트론의 출력 값을 결정하는 역할을 하며 그 종류에는 sign, sigmoid, tanh, ReLU 등이 있다.

Activation function	Equation	Example	1D Graph
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = \max(0, z)$	Multi-layer Neural Networks	

다양한 종류의 활성화 함수(Sebastian Raschka, CC BY-SA 4.0)

<https://playground.tensorflow.org>

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.



Epoch
000,000

Learning rate
0.03

Activation
Tanh

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

$X_1 X_2$

$\sin(X_1)$

+ - 2 HIDDEN LAYERS

+ -

4 neurons

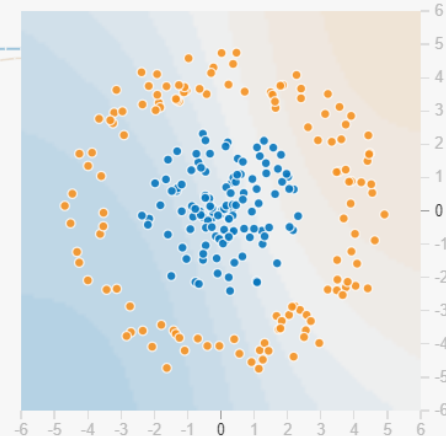
+ -

2 neurons

OUTPUT

Test loss 0.503

Training loss 0.502



The outputs are mixed with varying **weights**, shown by the thickness of the lines.

This is the output from one **neuron**. Hover to see it larger.



Epoch
000,000

Learning rate
0.01

Activation
ReLU

Regularization
None

Regularization rate
0

Problem type
Classification

활성화 함수, 출력 값을 결정

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

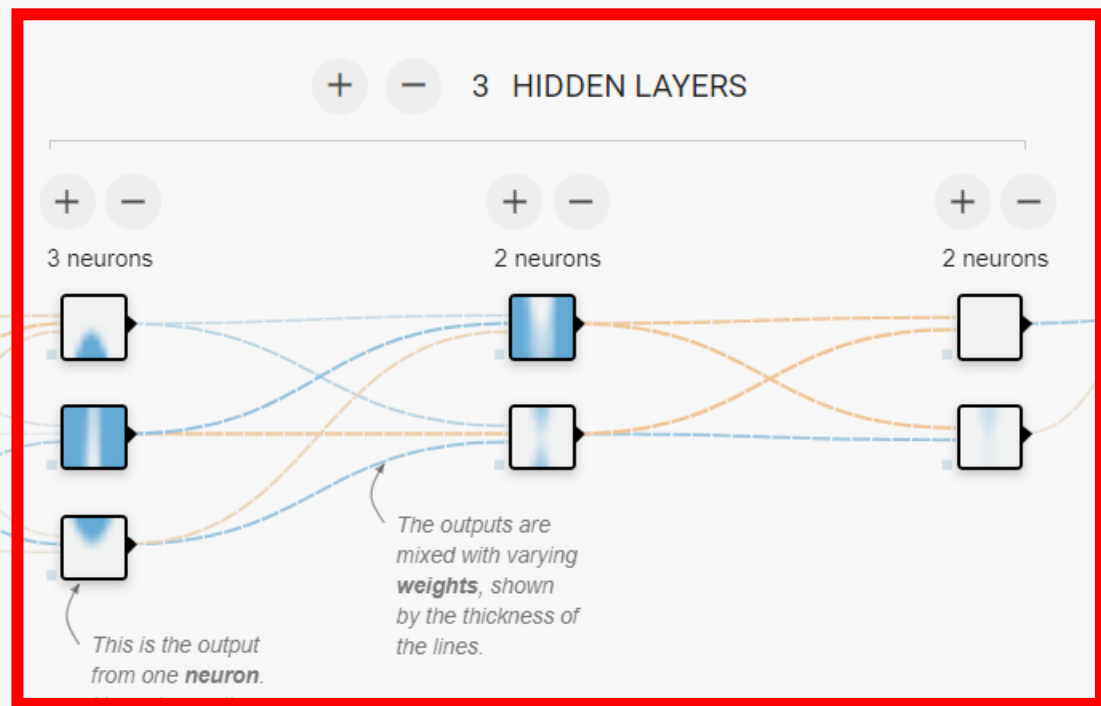
X_1^2

X_2^2

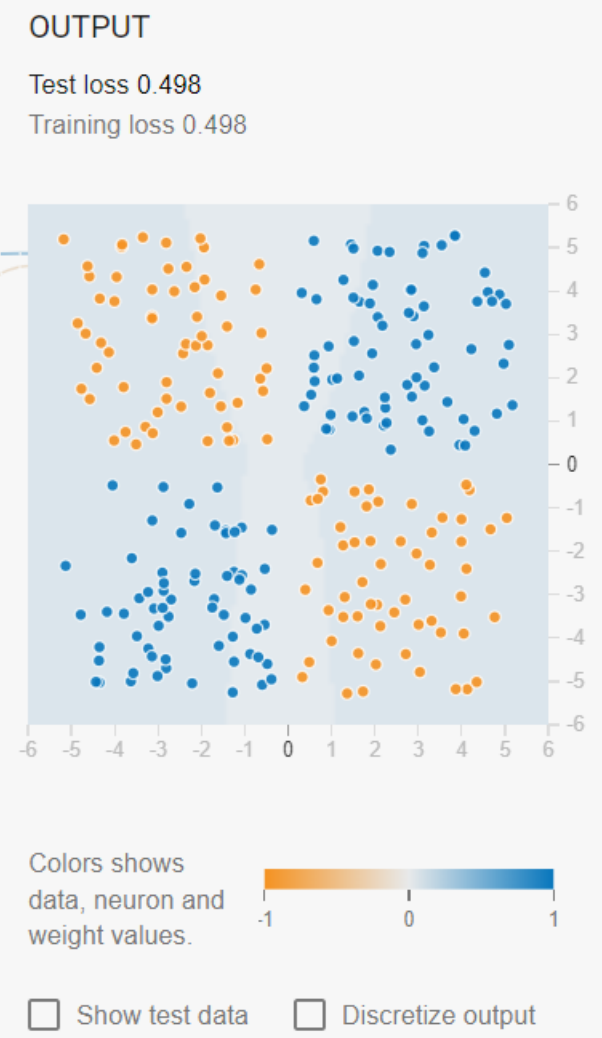
X_1X_2

$\sin(X_1)$

$\sin(X_2)$



은닉층, 퍼셉트론들의 집합



입력층의 모양을 결정

$w_1x_1 + w_2x_2 + w_3x^2 + \dots$

순전파 vs 역전파

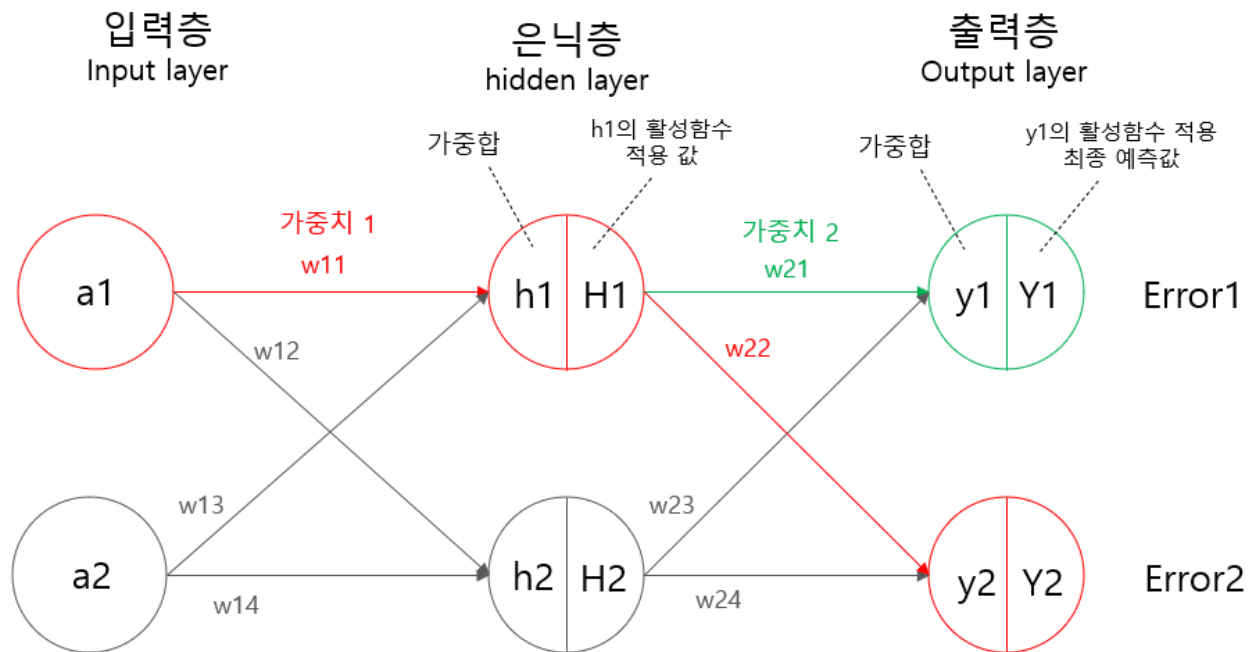
- 순전파: 입력 데이터에서부터 은닉층, 출력층까지 변수를 계산하고 결과를 도출
- 역전파: 순전파 결과와 input & output을 가지고 출력층에서부터 입력층으로 돌아가며 가중치를 업데이트한다. 이때, **학습률 (learning rate)를 설정**하여 가중치의 업데이트 정도를 결정한다.
- 역전파의 목표는 '**최적의 가중치를 찾는 것**'이다.
- 역전파가 최적의 가중치를 찾는 기준은
가중치 변화에 대한 총 에러의 변화를 최소화하는 가중치를 찾는 것이다.

-> $\frac{\partial E_{total}}{\partial W}$, E: Error, W: Weight

연쇄 법칙 (Chain rule)

- 그러나 Error를 구하는 식에서는 가중치에 대해서 바로 편미분할 수 없다.

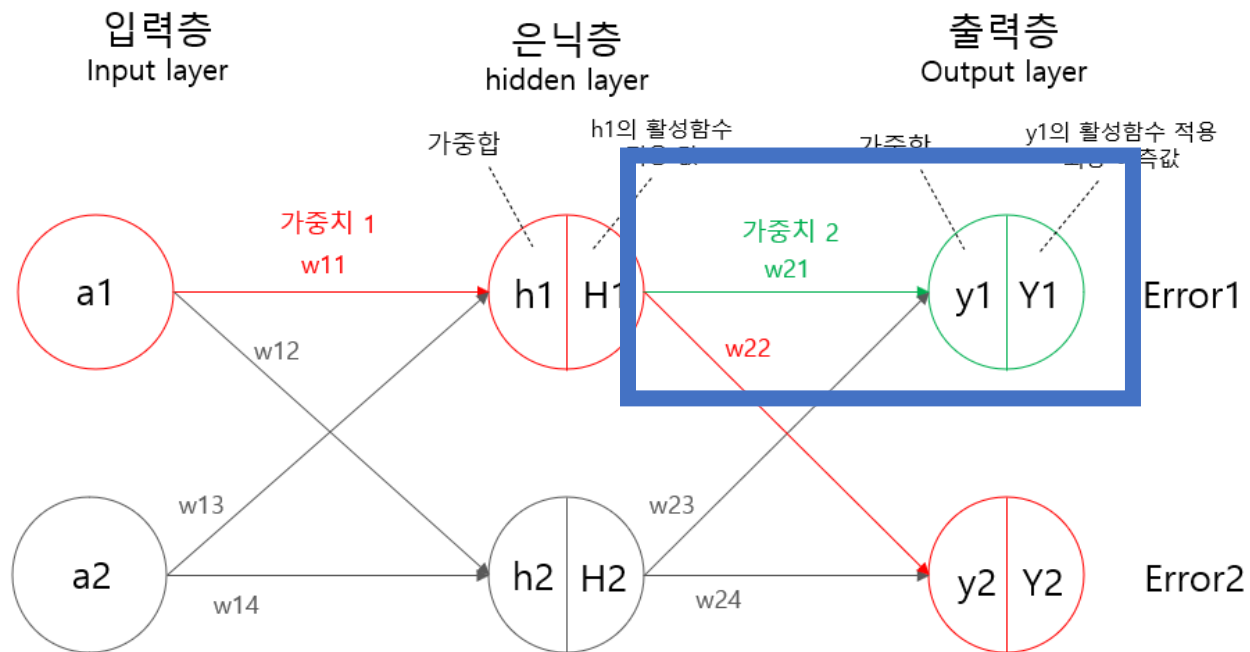
$$E_{total} = Error_1 + Error_2 = (Y_{1,real} - Y_{1,predict}) + (Y_{2,real} - Y_{2,predict})$$
$$\frac{\partial E_{total}}{\partial w_{21}} = ???$$



연쇄 법칙 (Chain rule)

- 따라서, 미분의 연쇄 법칙을 활용하여 다음과 같이 $\frac{\partial E_{total}}{\partial w_{21}}$ 를 계산 가능하도록 만든다.

$$\frac{\partial E_{total}}{\partial w_{21}} = \frac{\partial E_{total}}{\partial Y_1} \frac{\partial Y_1}{\partial y_1} \frac{\partial y_1}{\partial w_{21}}$$

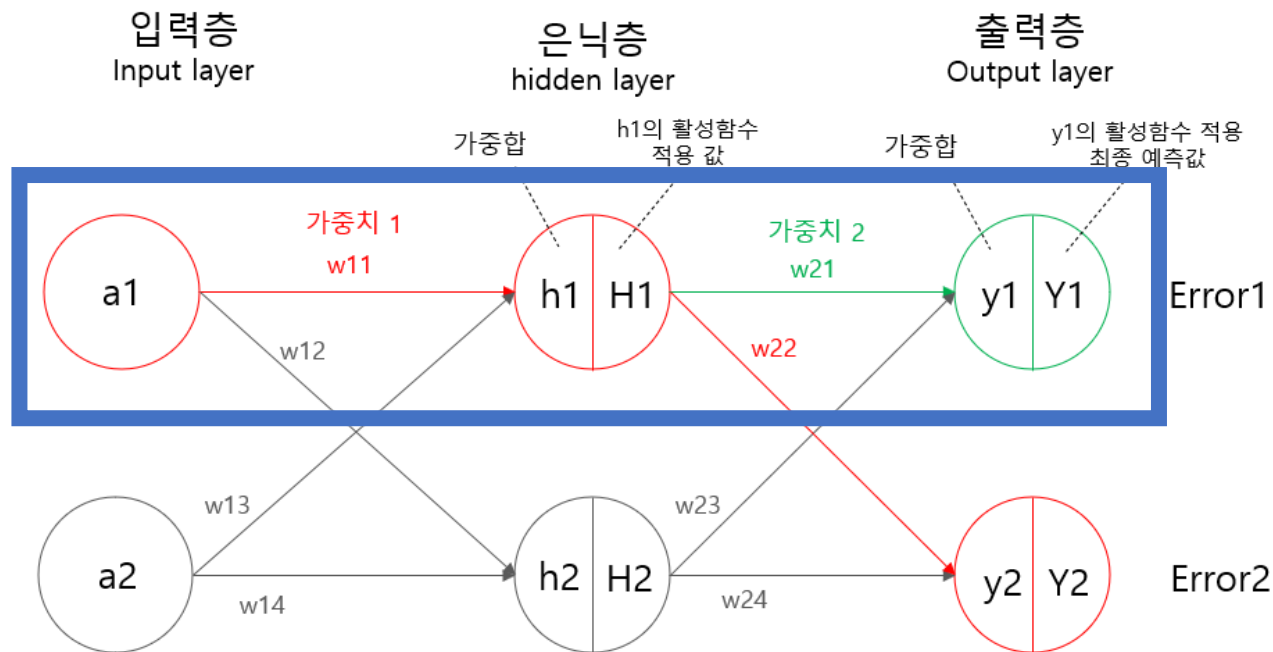


기울기 소실

(af: active function)

- 그런데 이 과정에서 $\frac{\partial H_1}{\partial h_1} = \frac{\partial af(h_1)}{\partial h_1}$ 와 같이 활성화 함수를 미분하게 되는데, 활성화 함수가 **시그모이드 함수처럼 미분 값이 작을 경우 연쇄 법칙을 적용 시 미분값(기울기)이 0으로 수렴하게 된다**. 이를 **기울기 소실**이라고 한다. (반대로 기울기가 발산하는 현상을 기울기 폭주라고 부른다)

$$\frac{\partial E_{total}}{\partial w_{11}} = \frac{\partial E_{total}}{\partial H_1} \frac{\partial H_1}{\partial h_1} \frac{\partial h_1}{\partial w_{11}} = \left(\frac{\partial E_1}{\partial H_1} + \frac{\partial E_2}{\partial H_1} \right) \frac{\partial H_1}{\partial h_1} \frac{\partial h_1}{\partial w_{11}} = \left(\frac{\partial E_1}{\partial y_1} \frac{\partial y_1}{\partial H_1} + \frac{\partial E_2}{\partial y_2} \frac{\partial y_2}{\partial H_1} \right) \frac{\partial H_1}{\partial h_1} \frac{\partial h_1}{\partial w_{11}}$$

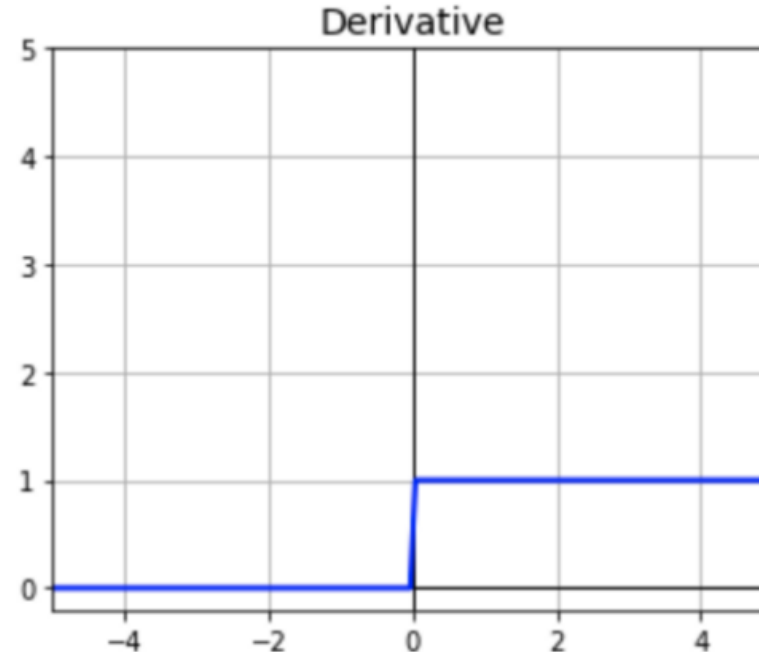
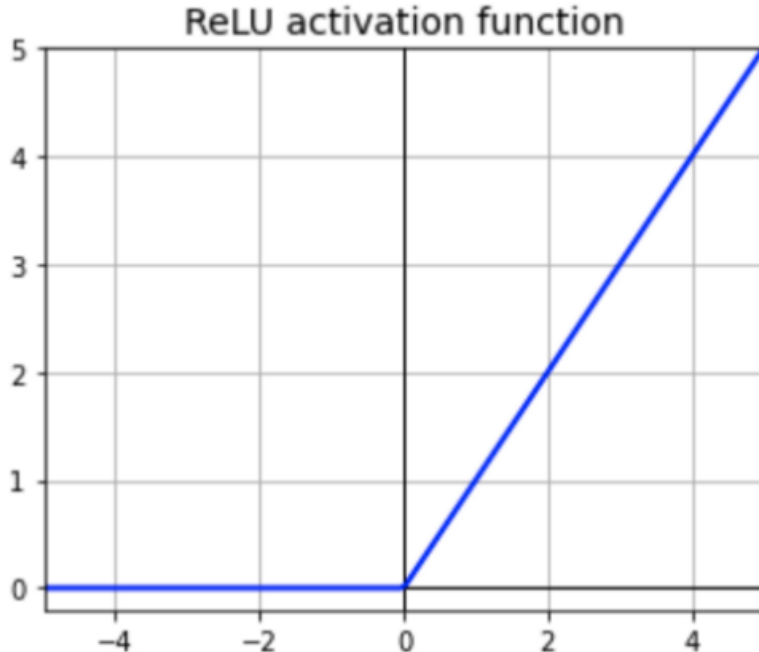


ReLU

- 따라서, 이러한 문제를 해결하기 위한 활성화 함수로 ReLU가 제안되었으며, 식은 다음과 같다.

$$f(x) = \max(0, x)$$

- ReLU 함수는 0보다 작은 값은 0으로 반환하고 0보다 큰 값은 값 그대로 반환하는 함수이며, 입력 값에 상관없이 입력 값이 양수이면 항상 미분값은 1이기 때문에 역전파 과정에서 기울기가 소실되는 문제를 해결할 수 있다.

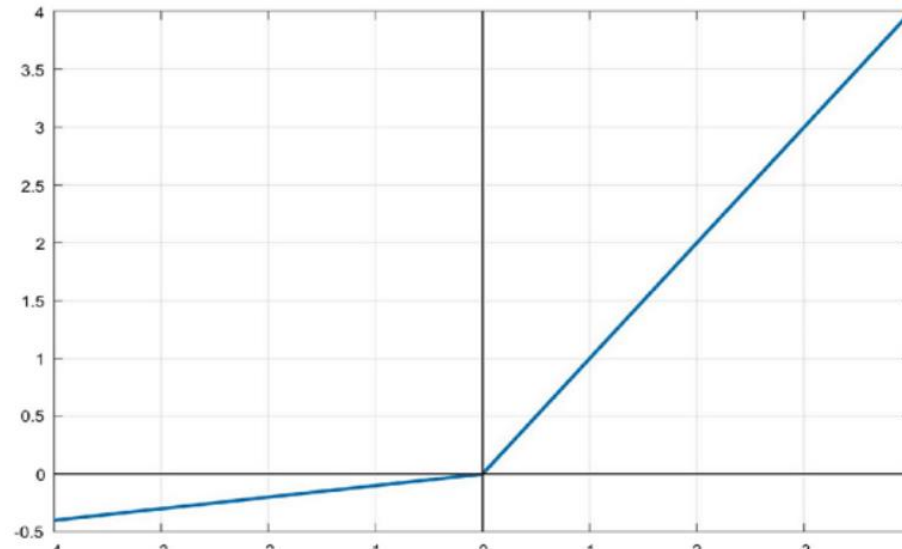


+ Leaky ReLU

- 하지만 입력값이 음수일 땐 미분값이 항상 0이기 때문에 입력값이 음수인 경우에는 이후에 학습이 이루어지지 않는 한계가 있다. 이를 죽어가는 ReLU(Dying ReLU)라고 한다.
- Dying ReLU 현상을 보완하기 위해 변형된 ReLU인 Leaky ReLU을 활성화 함수로 사용하며, 식은 다음과 같다.

$$f(x) = \max(-0.01x, x)$$

- Leaky ReLU 함수는 입력값이 음수일 때 출력값을 0이 아닌 a 가 곱해진 값을 출력하도록 하는 함수로 입력값이 음수여도 기울기가 0이 되지 않기 때문에 Dying ReLU 현상을 방지할 수 있다.

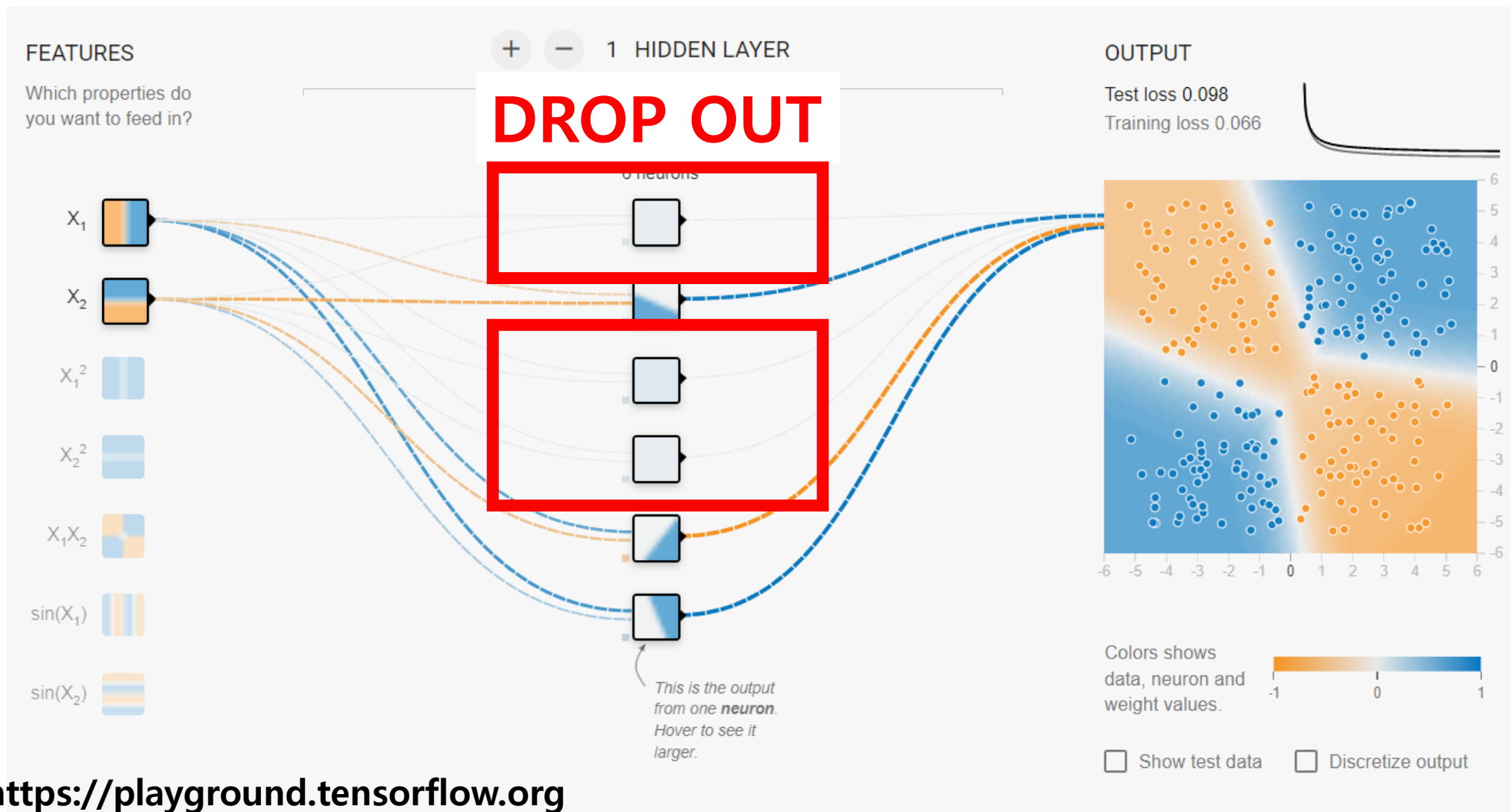


Batch Regularization (배치 정규화)

- 학습 과정에서 각 layer(미니 배치 학습법의 경우 batch도 포함)의 입력을 정규화하여 학습의 안정성과 속도를 향상시키는 방법이다.
- 이는 심층 신경망에서 자주 발생하는 내부 공변량 변화 (Internal Covariate Shift) 문제를 완화하는데 사용된다.
- 이때 내부 공변량 변화란, 데이터가 신경망의 각 layer를 지날 때마다 분포가 변하는 현상을 의미하며, 이는 학습의 안정성과 속도를 낮출 수 있는 요인 중 하나이다.
- 또한, 활성화 함수 설정만으로 해결할 수 없는 기울기 소실 또는 폭주를 예방하기도 한다.
- 배치 정규화 과정: 학습 과정에서 각 Batch 또는 layer마다 평균과 분산을 활용하여 분포를 정규화하여 Scale을 통일시킨다.

Dropout

- 학습 시 은닉층의 일부 퍼셉트론(노드)들을 중지, 즉 weight를 0으로 만듦으로써 Overfitting을 방지한다.



패션 분류기 만들기(Fashion MNIST Classifier)

개요	평가기준표	제출	리뷰 결과
----	-------	----	-------

리뷰어 지정 전

리뷰 #1 | -

 프로젝트 평가	 코드 리뷰	 수강생 메모	 리뷰 목록
---	---	--	---

리뷰	결과	리뷰일	리뷰어
리뷰 #1 (현재 리뷰)	리뷰어 지정 전	2023.09.27	-