

Neural Network 2:

ReLU and 초기값 정하기 (2006/2007
breakthrough)

까지 요약(모두의 딥러닝)

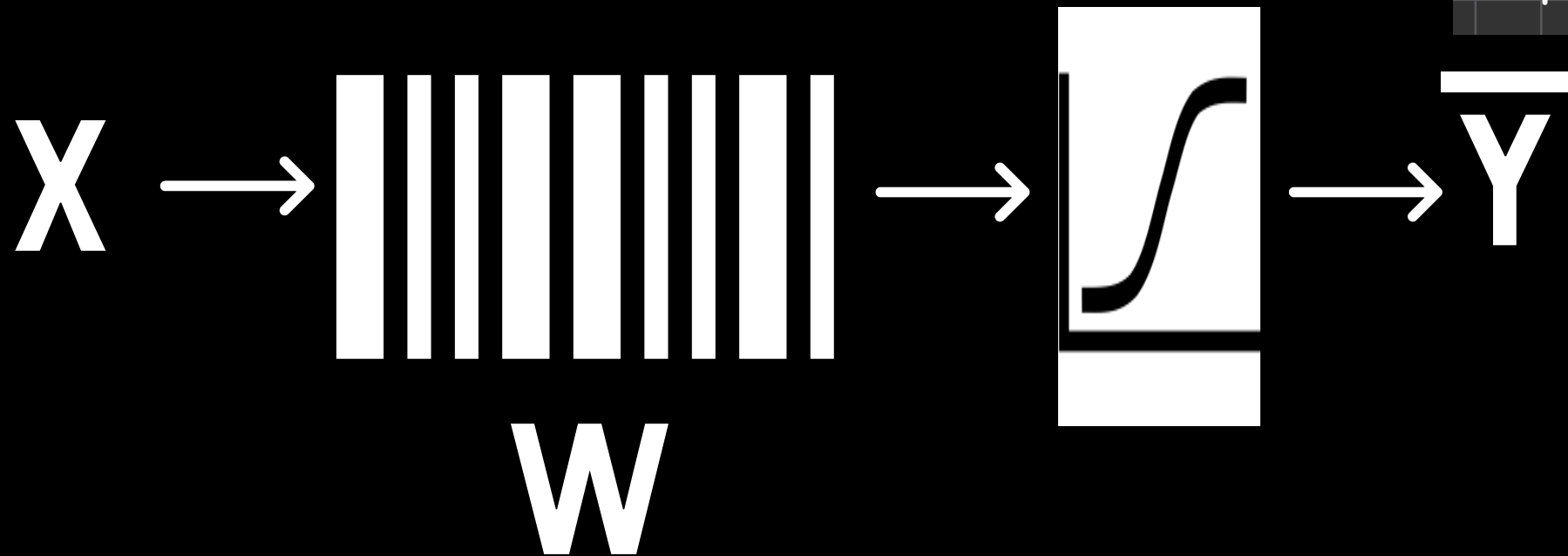
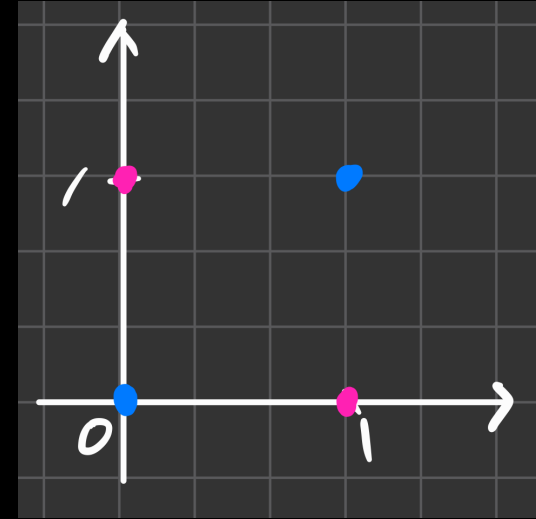
Contents

- **Solution of XOR Problem / NN(Neural Network)**
- **BackPropagation**
- **ReLU (Rectified Linear Unit)**
- **Initialize Weights**
- **Dropout and model ensemble**

Solution of XOR Problem

XOR Problem

Just one Layer cannot solve the XOR problem



Solution :
Using multiple Layer

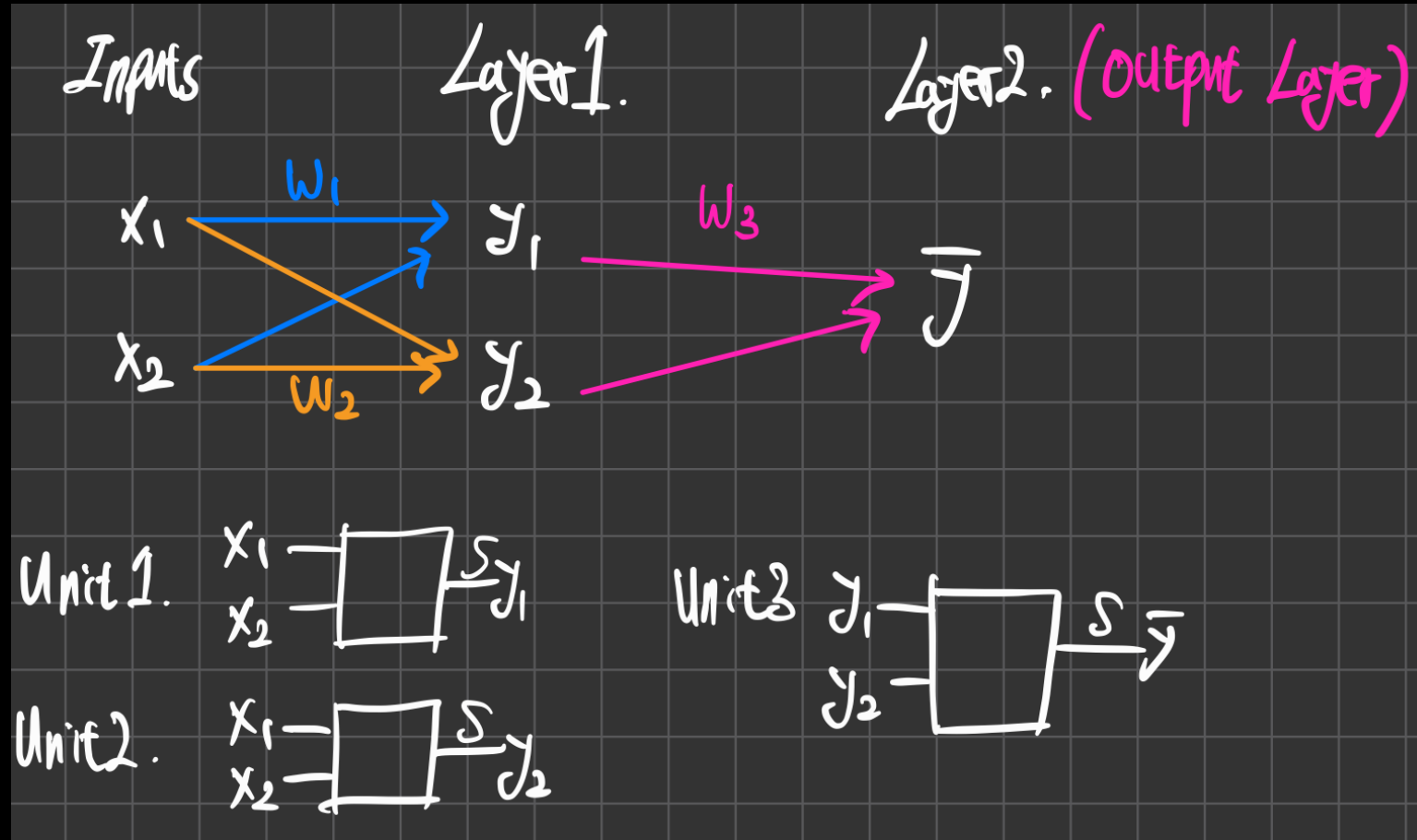
Solution :
Using multiple Layer

Neural Network(NN)

Neural Network(NN)

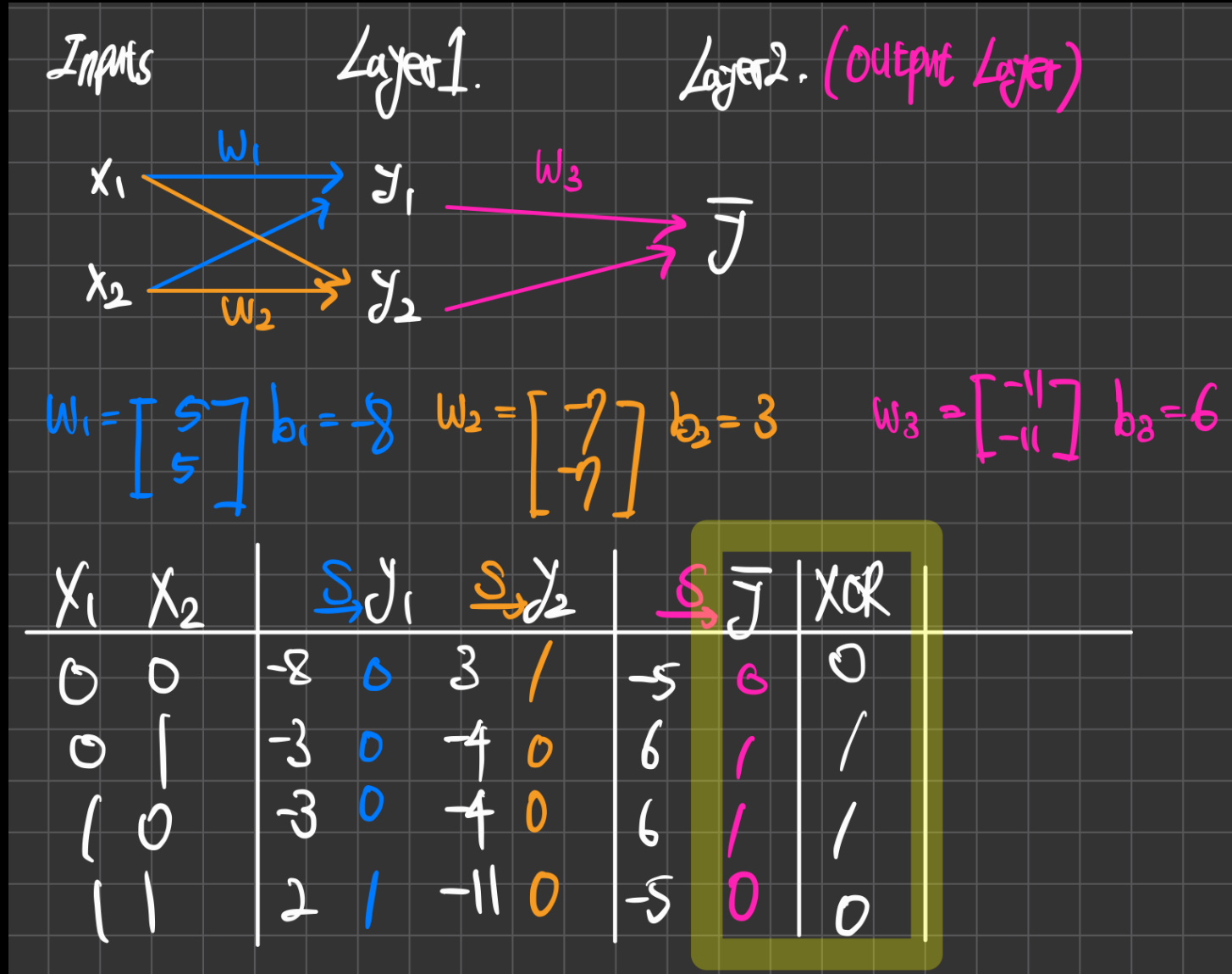
Example of XOR Problem

Example of XOR Problem



2Layers with 3Units

Example of XOR Problem

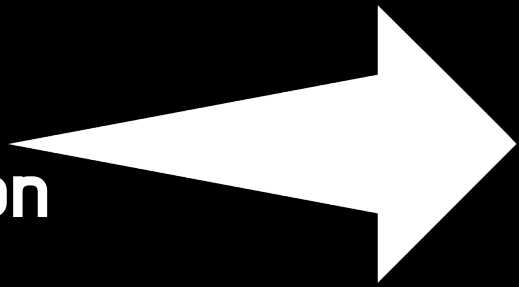


Example of XOR Problem

Using matrix for
Multinomial Classification

Example of XOR Problem

Using matrix for
Multinomial Classification



각 Layer의
Unit들의 연산을
Layer 당
하나의 행렬곱으로
표현가능!

Example of XOR Problem

Using matrix for
Multinomial Classification

$$W_1, W_2 \rightarrow W_1 = \begin{bmatrix} 5 & -7 \\ 5 & -7 \end{bmatrix}$$

$$W_3 \rightarrow W_2 = \begin{bmatrix} -11 \\ -11 \end{bmatrix}$$

$$X_1, X_2 \rightarrow X = \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

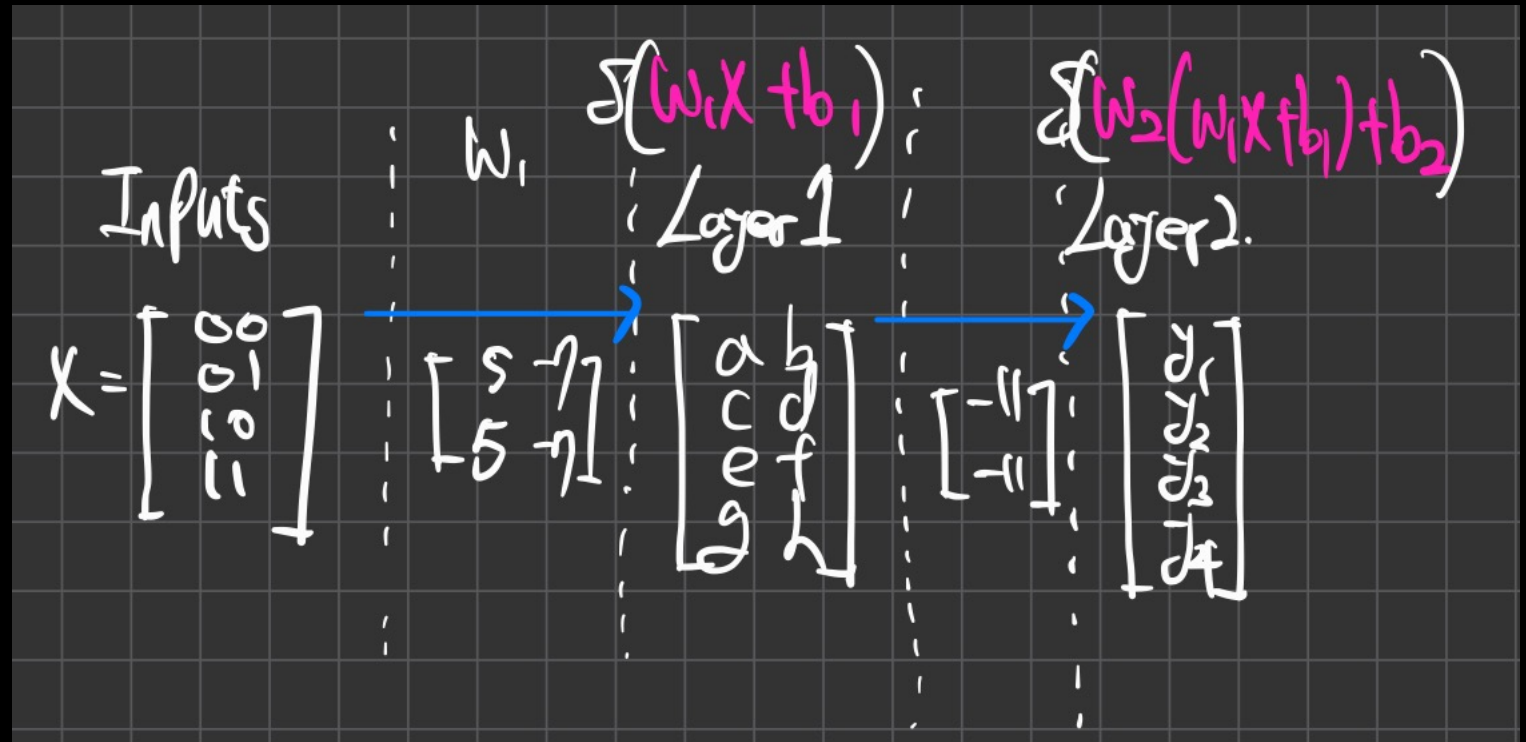
$$X = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} -8 \\ -8 \\ -2 \\ -8 \end{bmatrix} \quad b_2 = \begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}$$

Example of XOR Problem

Using matrix for Multinomial Classification

$$\begin{aligned} w_1, w_2 &\rightarrow W_1 = \begin{bmatrix} 5 & -7 \\ 5 & -7 \end{bmatrix} & b_1 = \begin{bmatrix} -8 \\ -8 \end{bmatrix} & b_2 = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} \\ w_3 &\rightarrow W_2 = \begin{bmatrix} -11 \\ -11 \end{bmatrix} \\ x_1, x_2 &\rightarrow X = \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \\ & \quad \quad \quad \gamma = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \end{aligned}$$

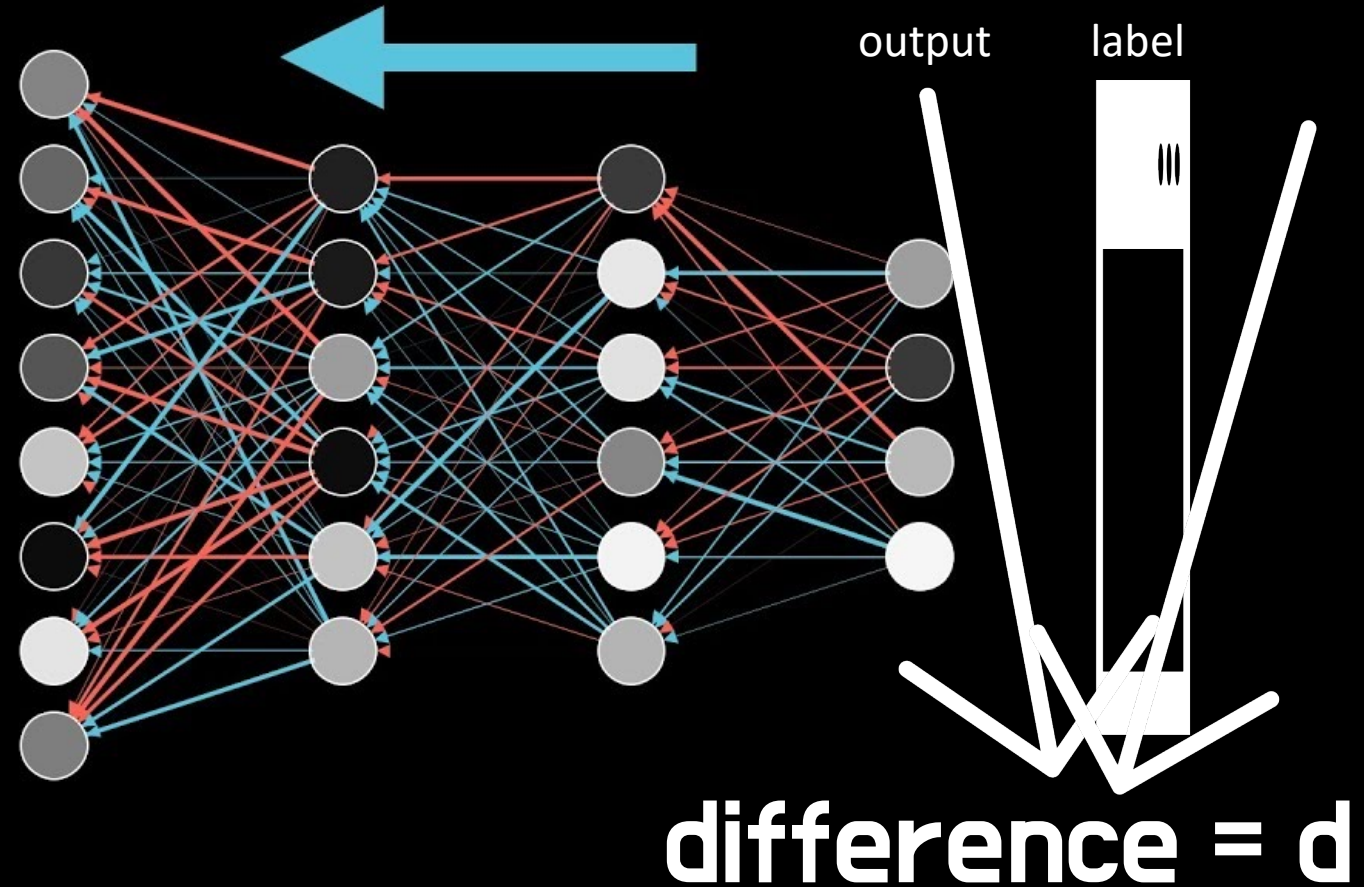


How to training W ?

How to find W for the model?

Backpropagation

Backpropagation



Backpropagation

difference = d

output layer에서 부터
미분 값을 전달하면
i번째 Layer의 W에 대한
편미분 값을 구할 수 있다!

By Using Chain Rule

$$W_adj = \partial d / \partial W_i$$

ReLU (Rectified Linear Unit)

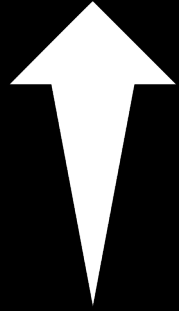
ReLU

Sigmoid 함수를
사용하게 되면

ReLU

Sigmoid 함수를
사용하게 되면

Layer 수

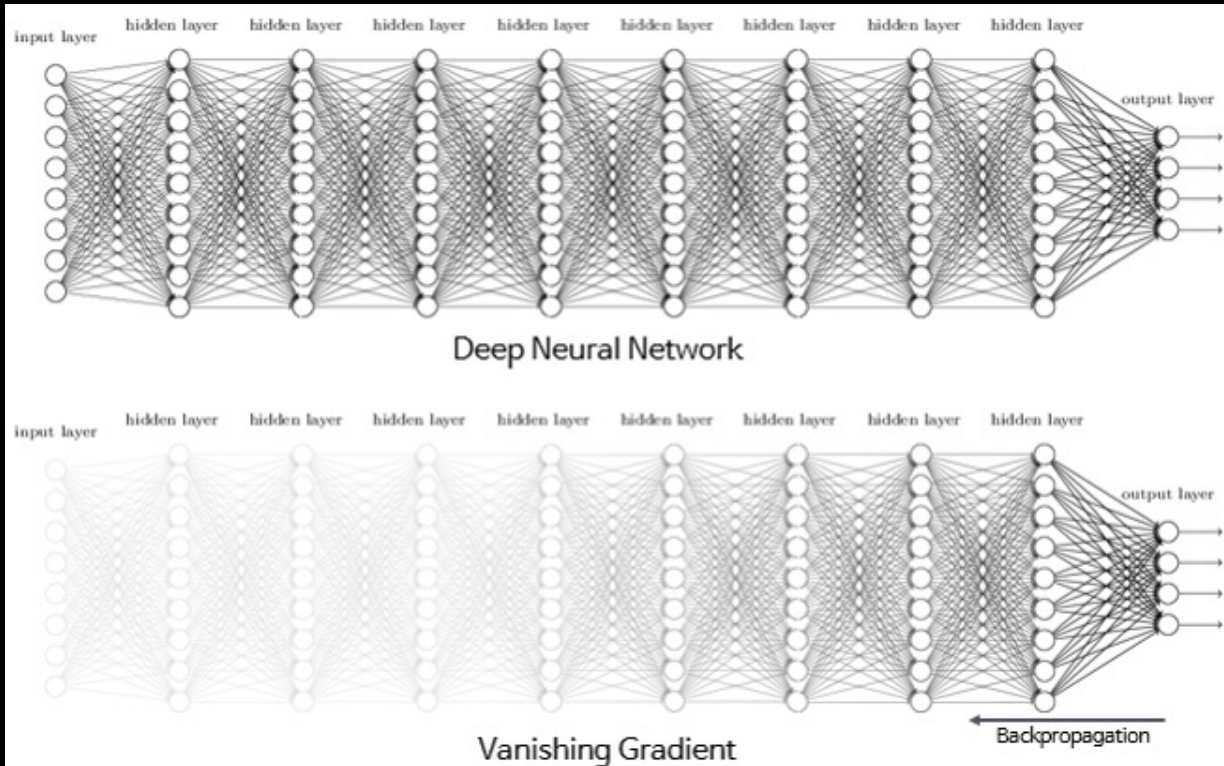


Vanishing Gradient



학습이 잘 되지
않는다!

ReLU



Layer 수



Vanishing Gradient



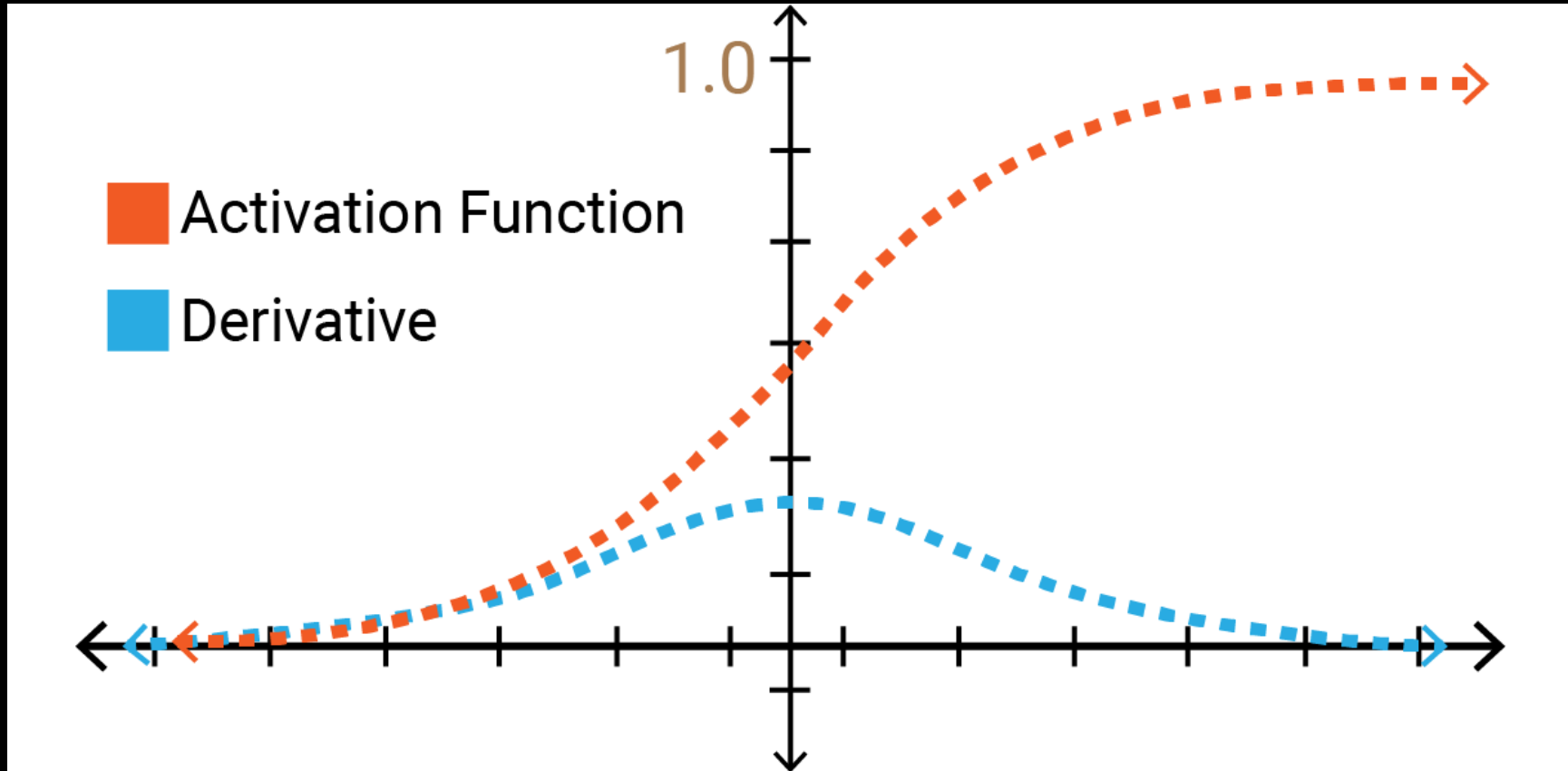
학습이 잘 되지
않는다!

ReLU

WHY?

Vanishing Gradient

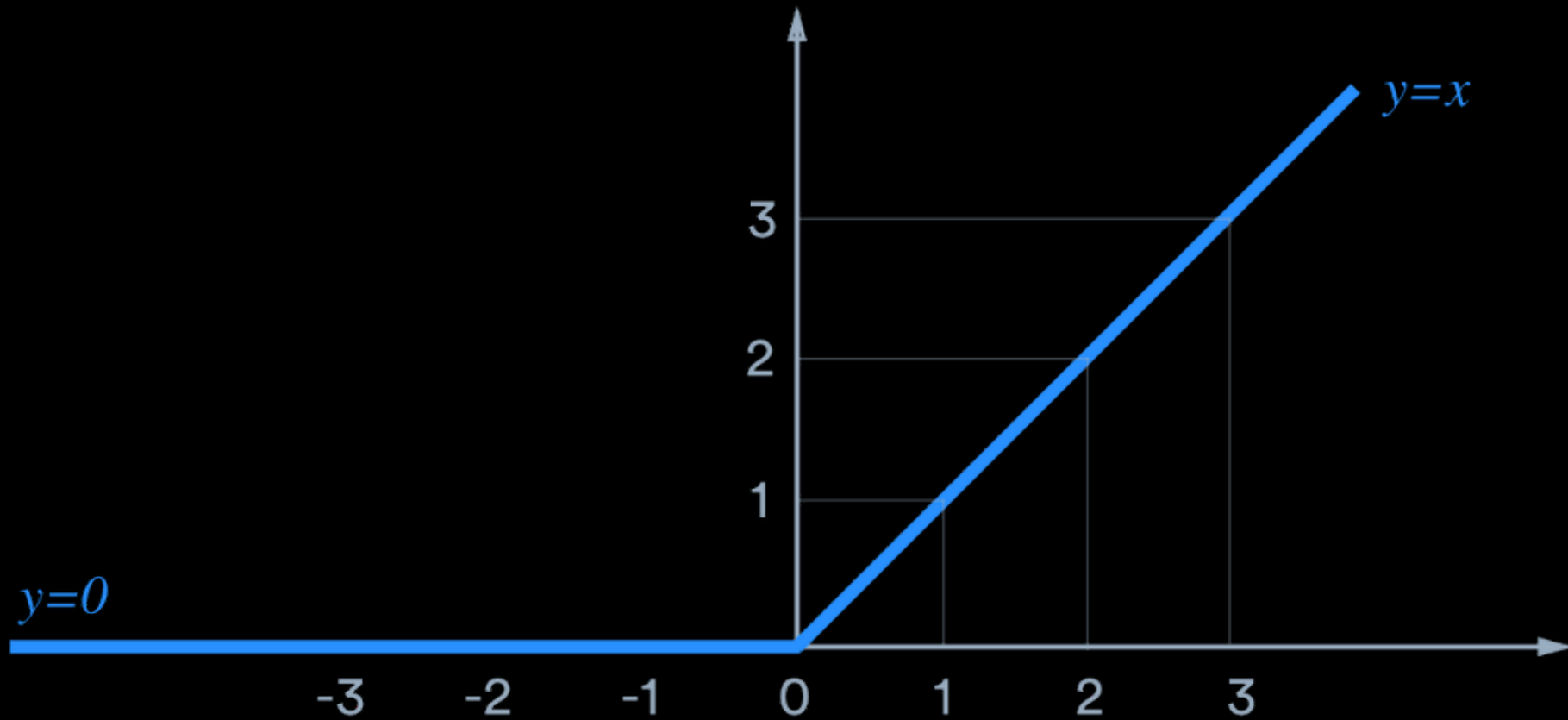
ReLU



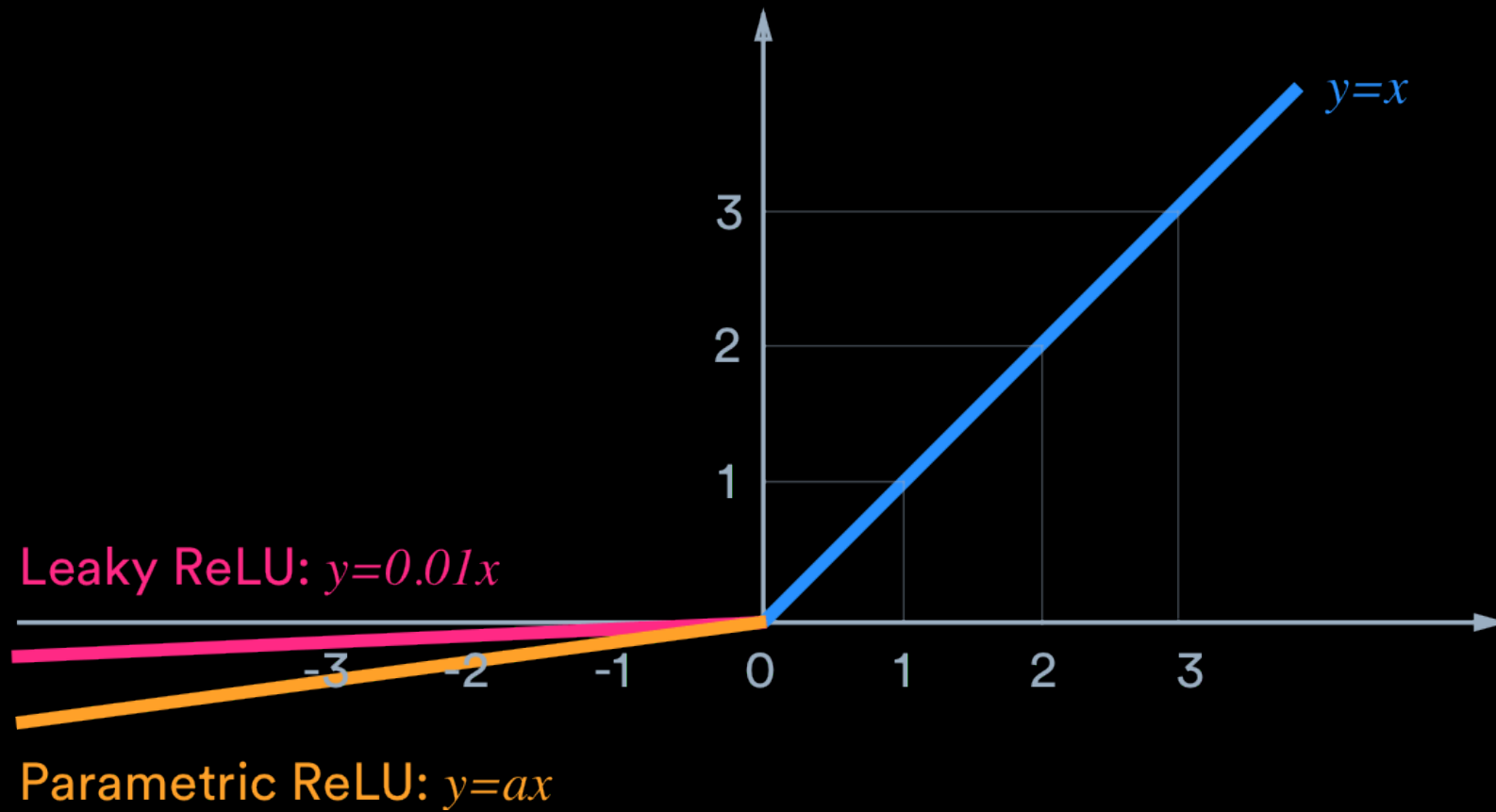
Sigmoid함수의 미분계수가 0에 아주 가까움 ->
backpropagation에서 chain rule을 적용할 때 input layer와
가까워질 수록 0과 가까운 값을 많이 곱하게 됨 ->
미분계수가 아주 작아짐 -> w_adj값이 아주 작아짐

ReLU

Hidden Layer에서 사용

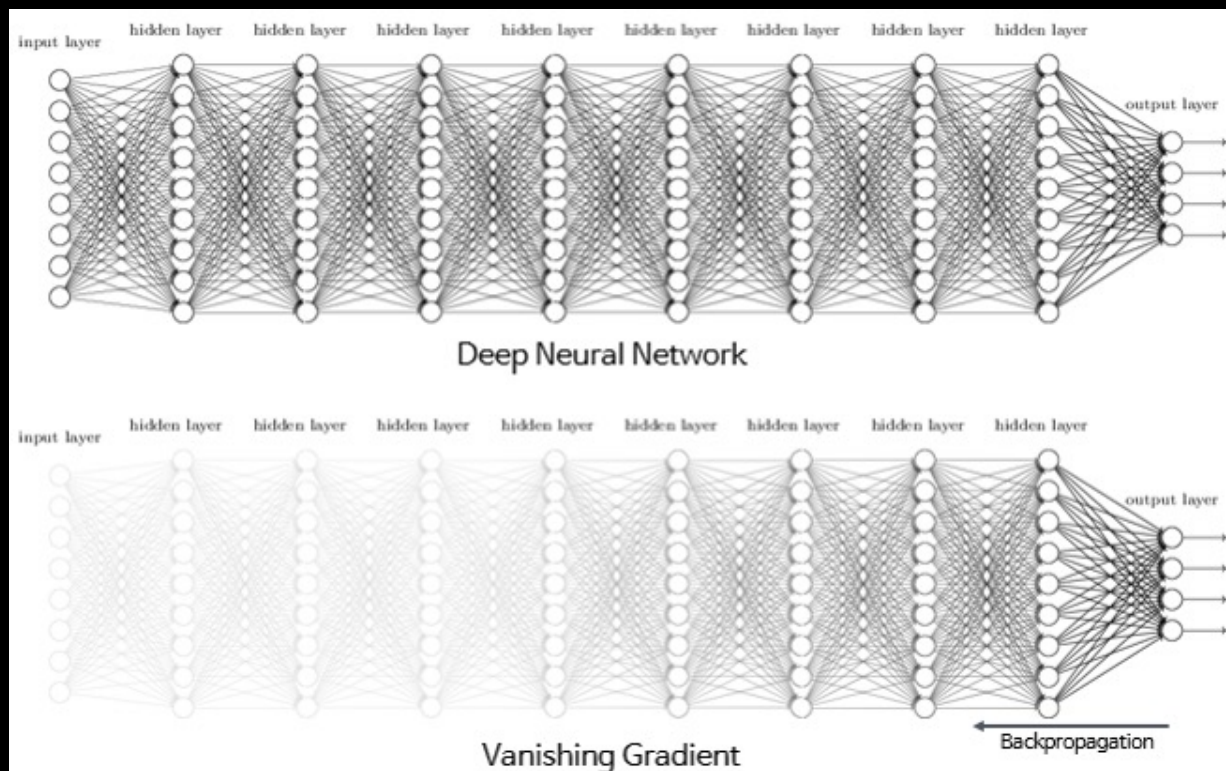


ReLU



Initialize Weights

Initialize Weights



마찬가지로 W 가 너무 0에
가까우면 학습이 제대로 안됨

Initialize Weights

Solution :
RBM(Restricted Boltzmann Machine)

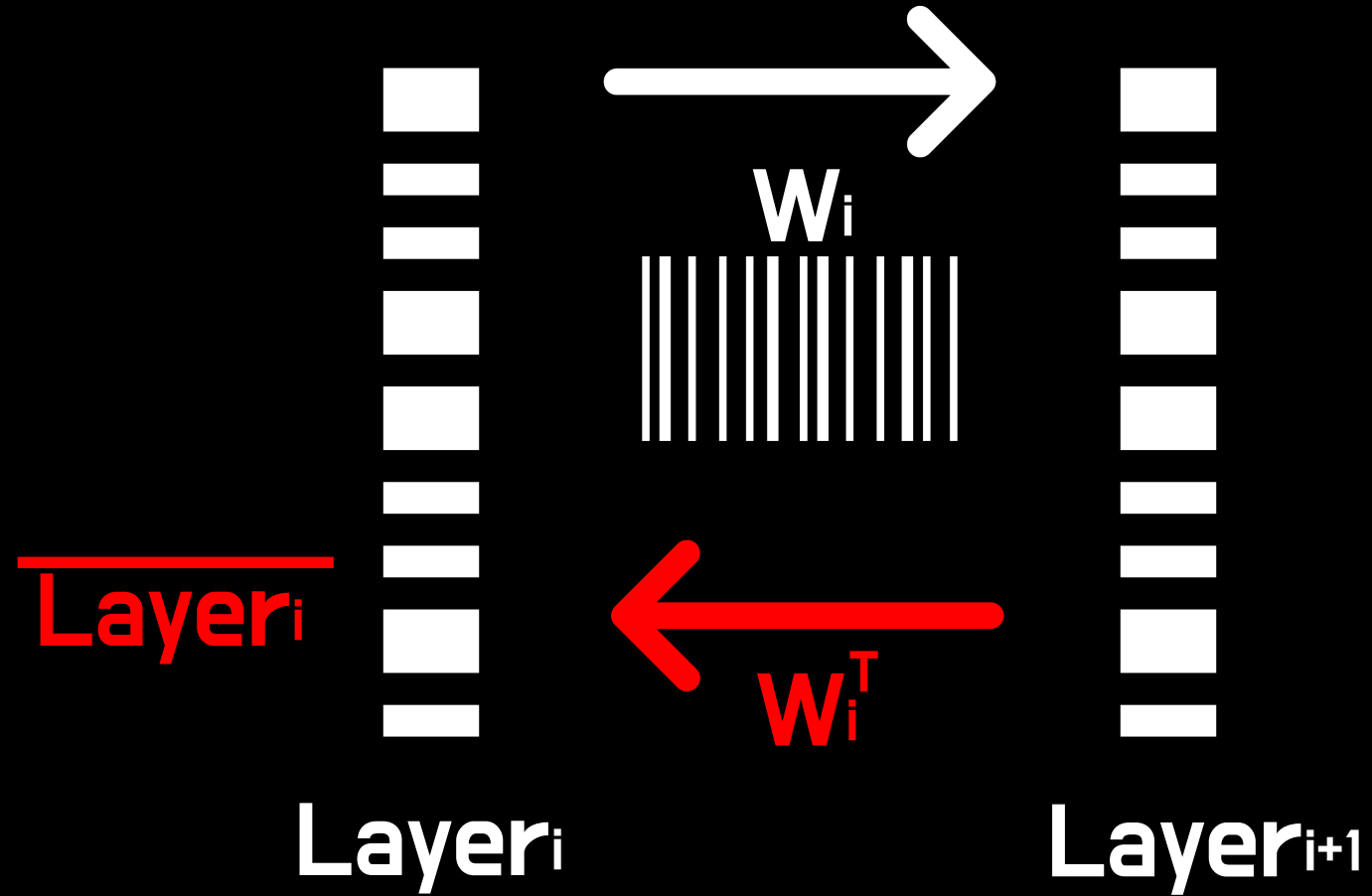
Initialize Weights

prelearning!

Initialize Weights

How?

Initialize Weights



Initialize Weights

Layer_i Layer_i 차이가 최소가 되도록

W_i prelearning!

Initialize Weights

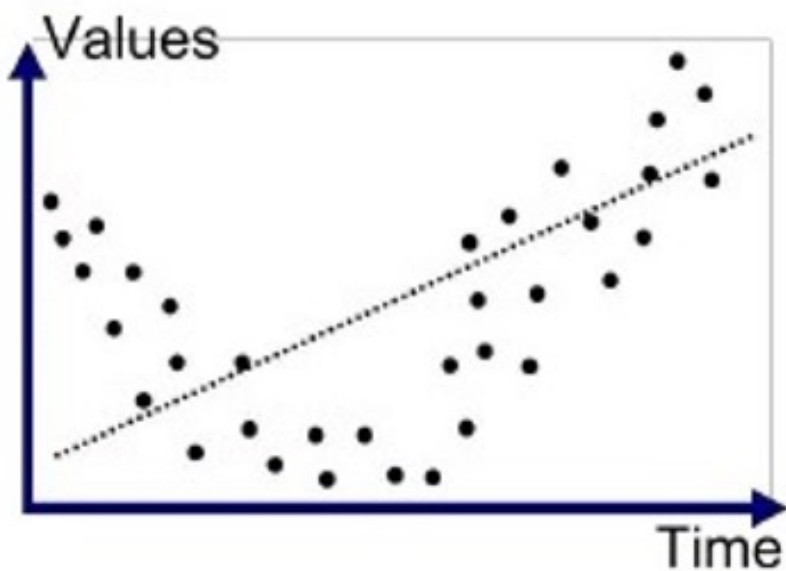
More simple Way?

He's initialization
Xavier initialization...

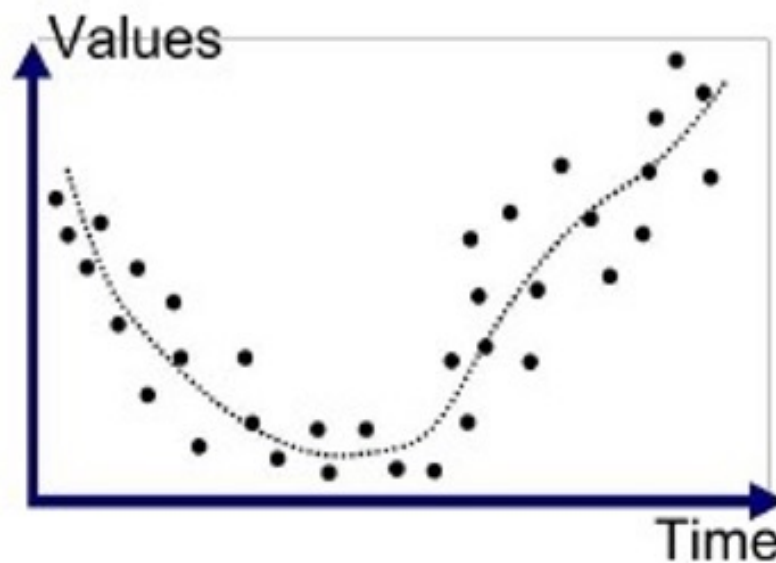
Dropout and model ensemble

Dropout and model ensemble

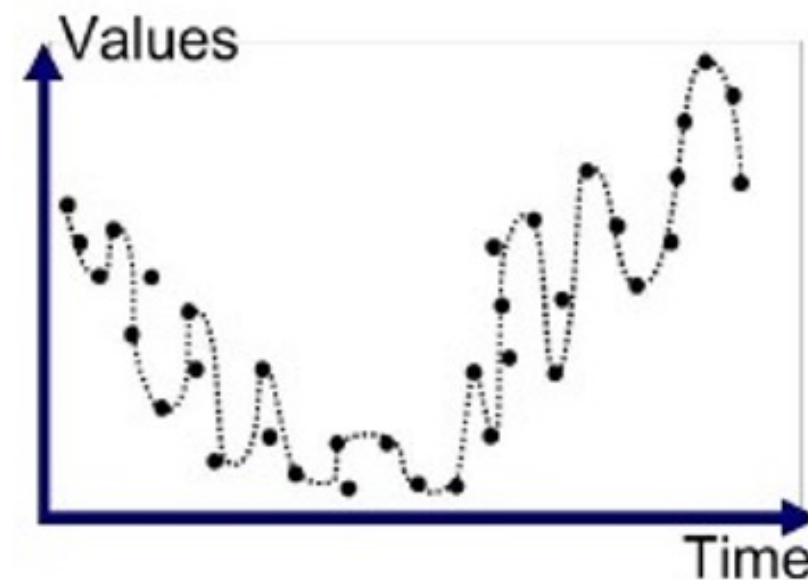
Overfitting



Underfitted



Good Fit/Robust



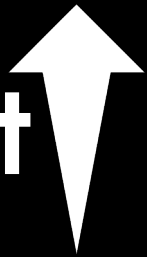
Overfitted

Dropout and model ensemble

Soltution?

Dropout and model ensemble

Soltution?

- trainig set 
- Regularization

Dropout and model ensemble

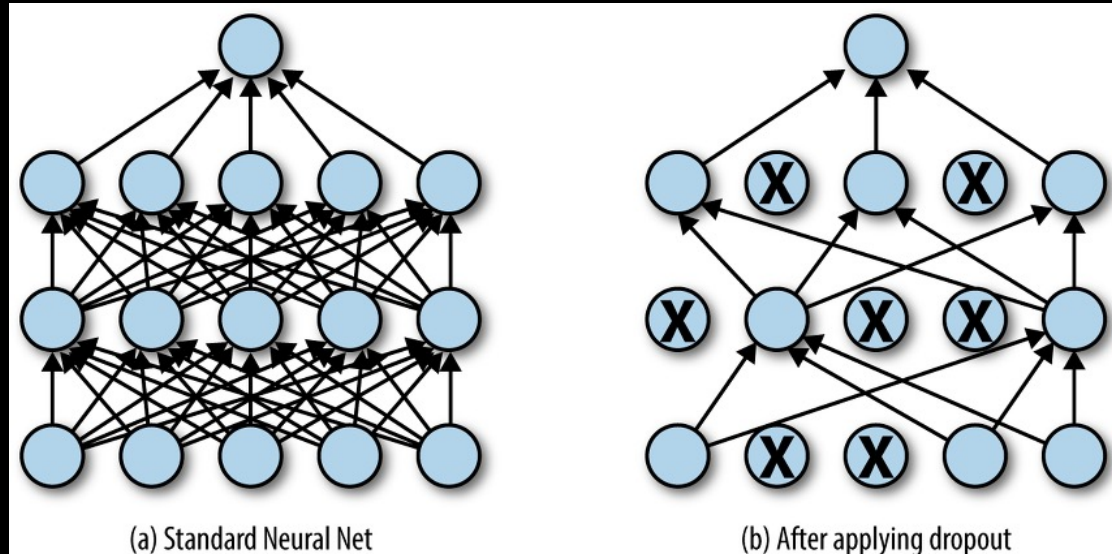
Soltution?

- trainig set 
- Regularization
- Dropout!

Dropout and model ensemble

Dropout

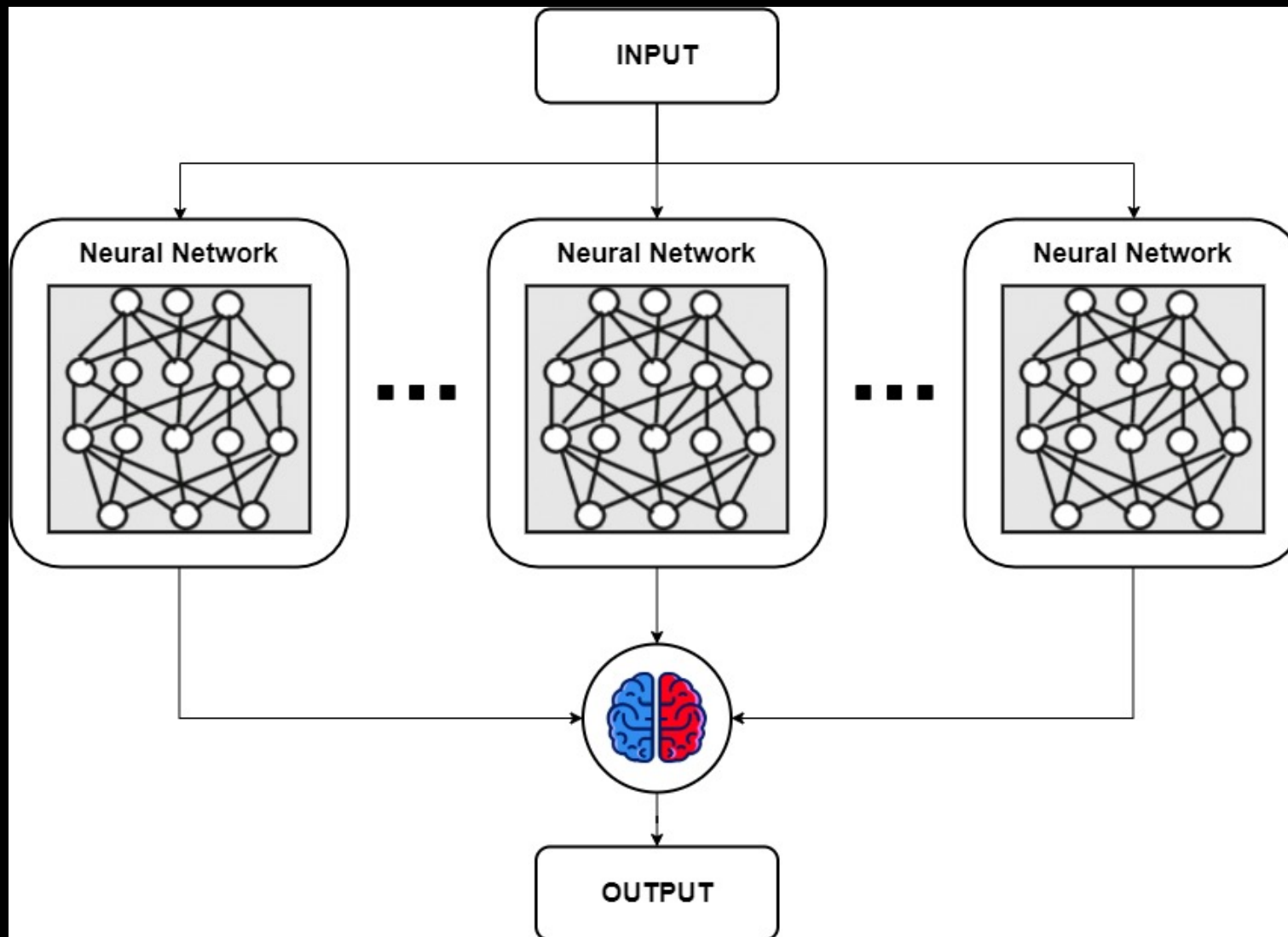
학습하는 과정에서
특정 유닛을 꺼버리는 방식



Dropout and model ensemble

ensemble

Dropout and model ensemble



Dropout and model ensemble

