Advanced Classification Model

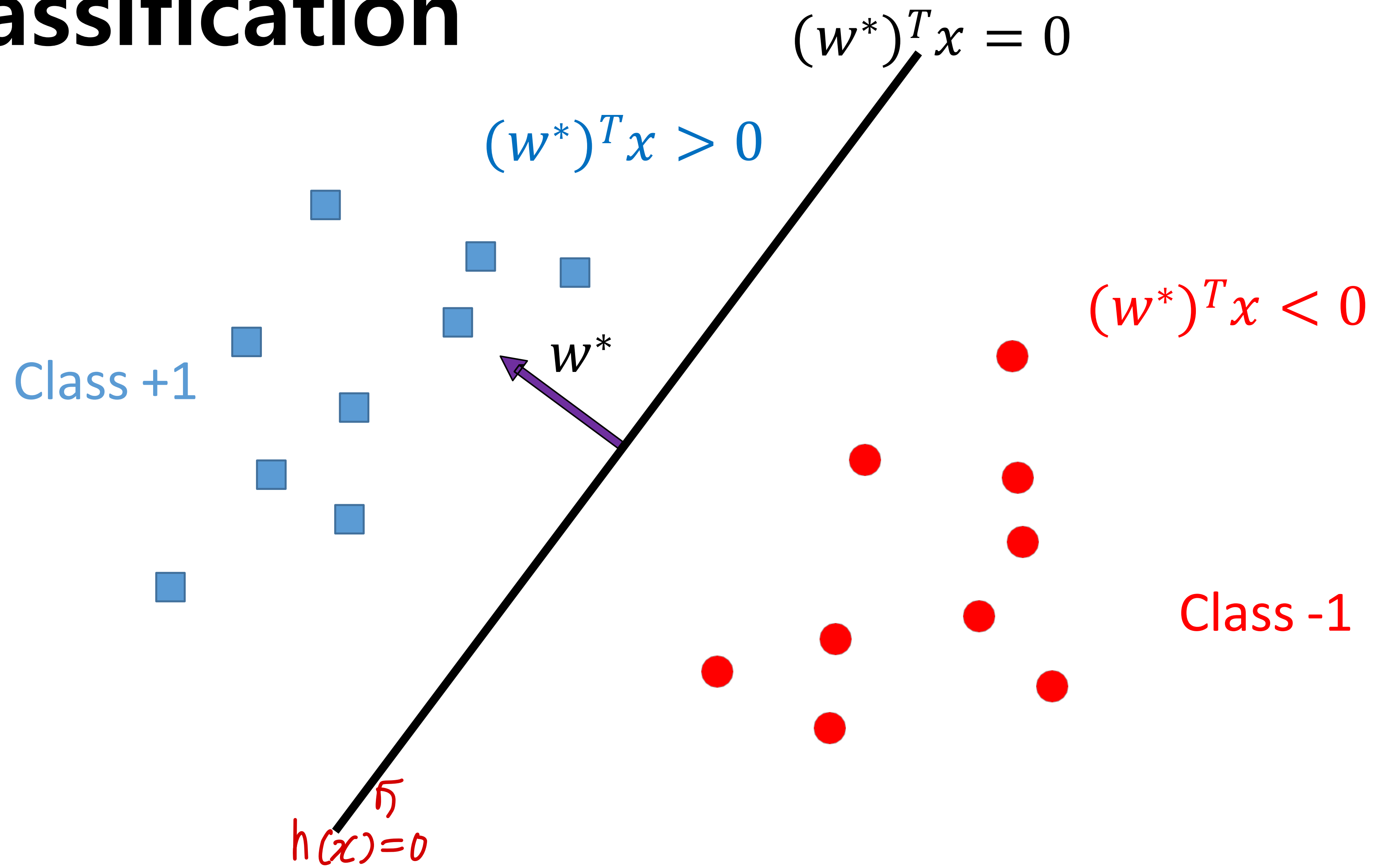
Linear and non-linear model...

Prof. Je-Won Kang
Electronic & Electrical Engineering
Ewha Womans University

Contents

- Support vector machine (SVM)
- Neural network (NN)

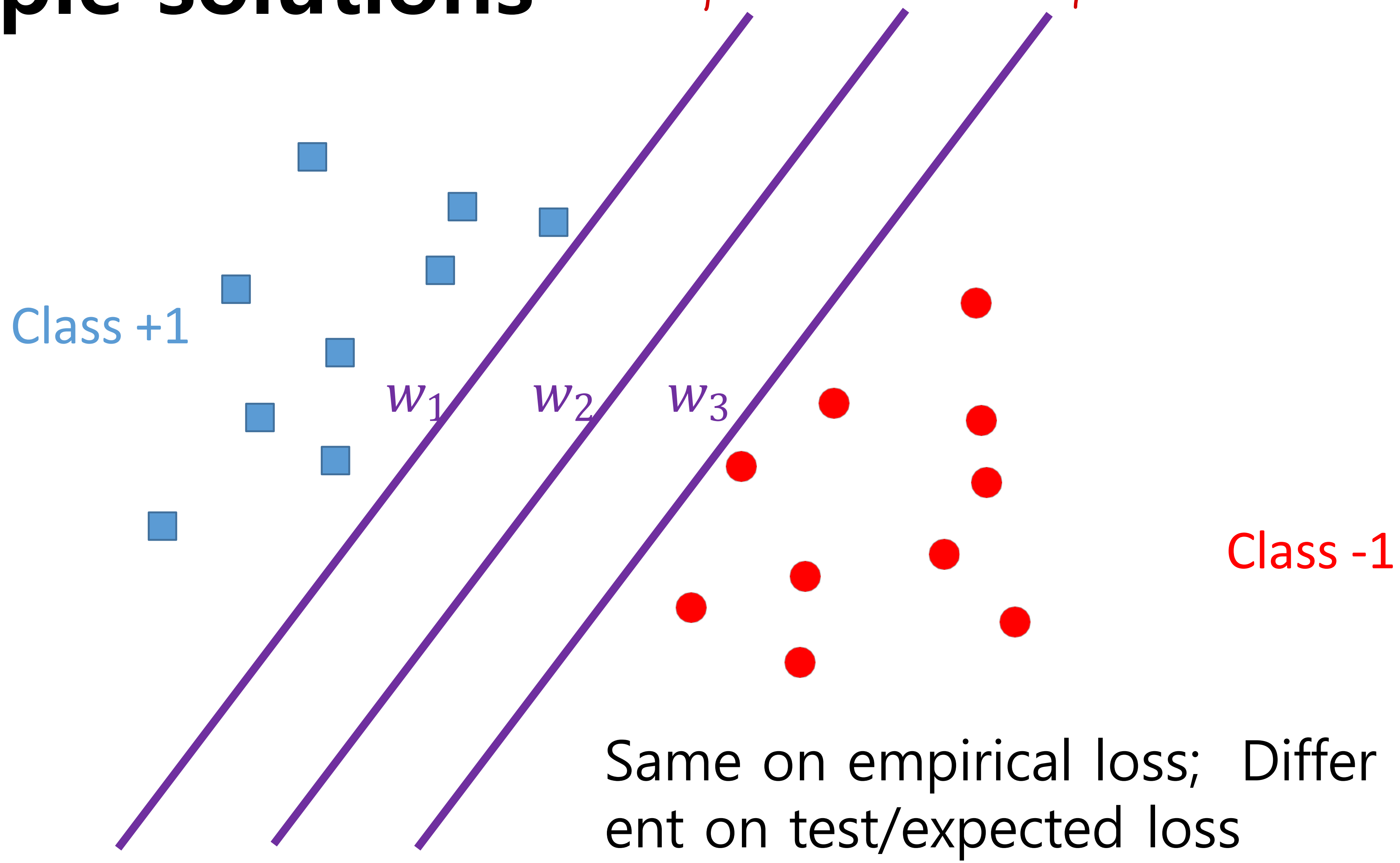
Linear classification



Assume perfect separation between the two classes using discriminant function w^*

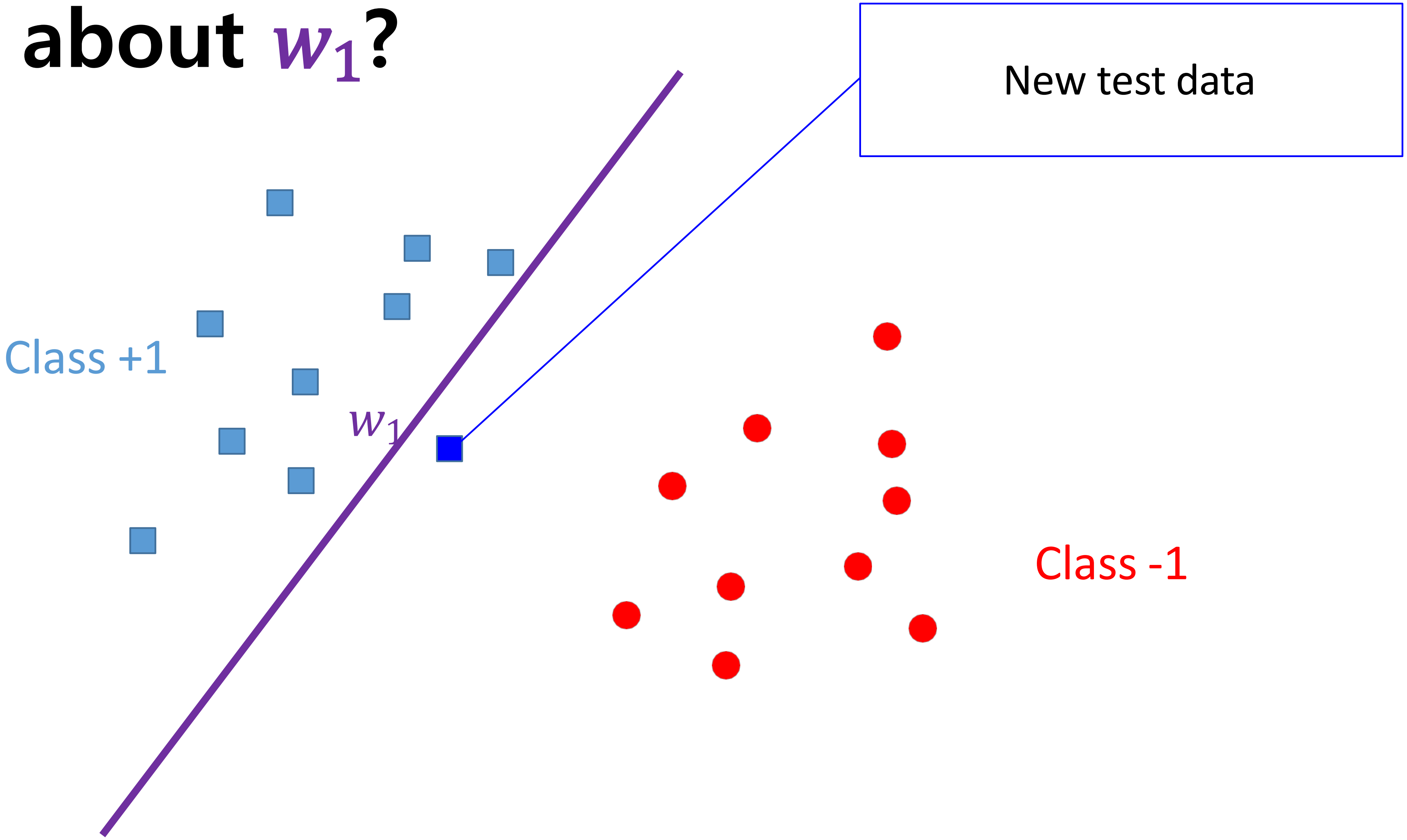
Multiple solutions

다양한 hyper plane 존재

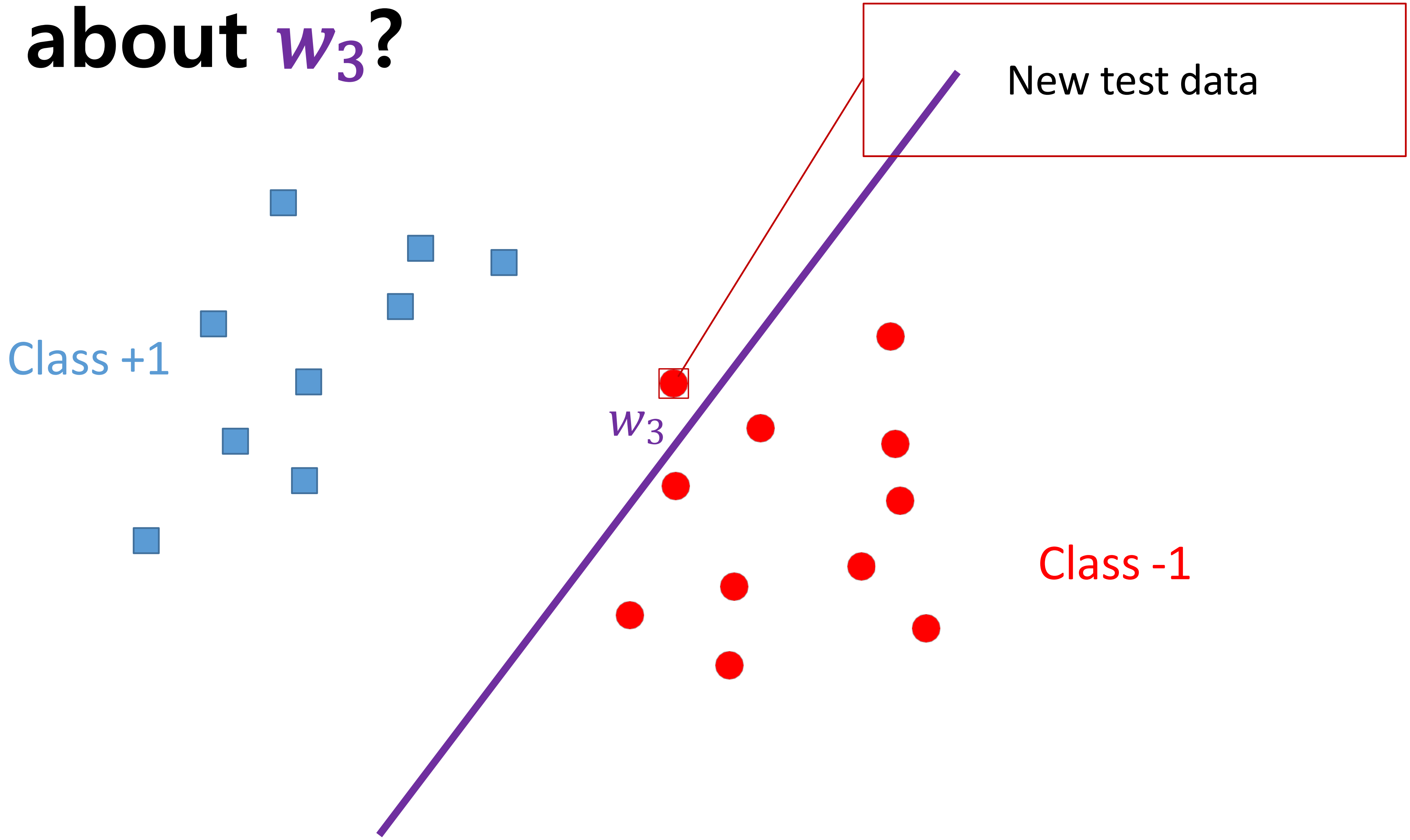


Same on empirical loss; Different on test/expected loss

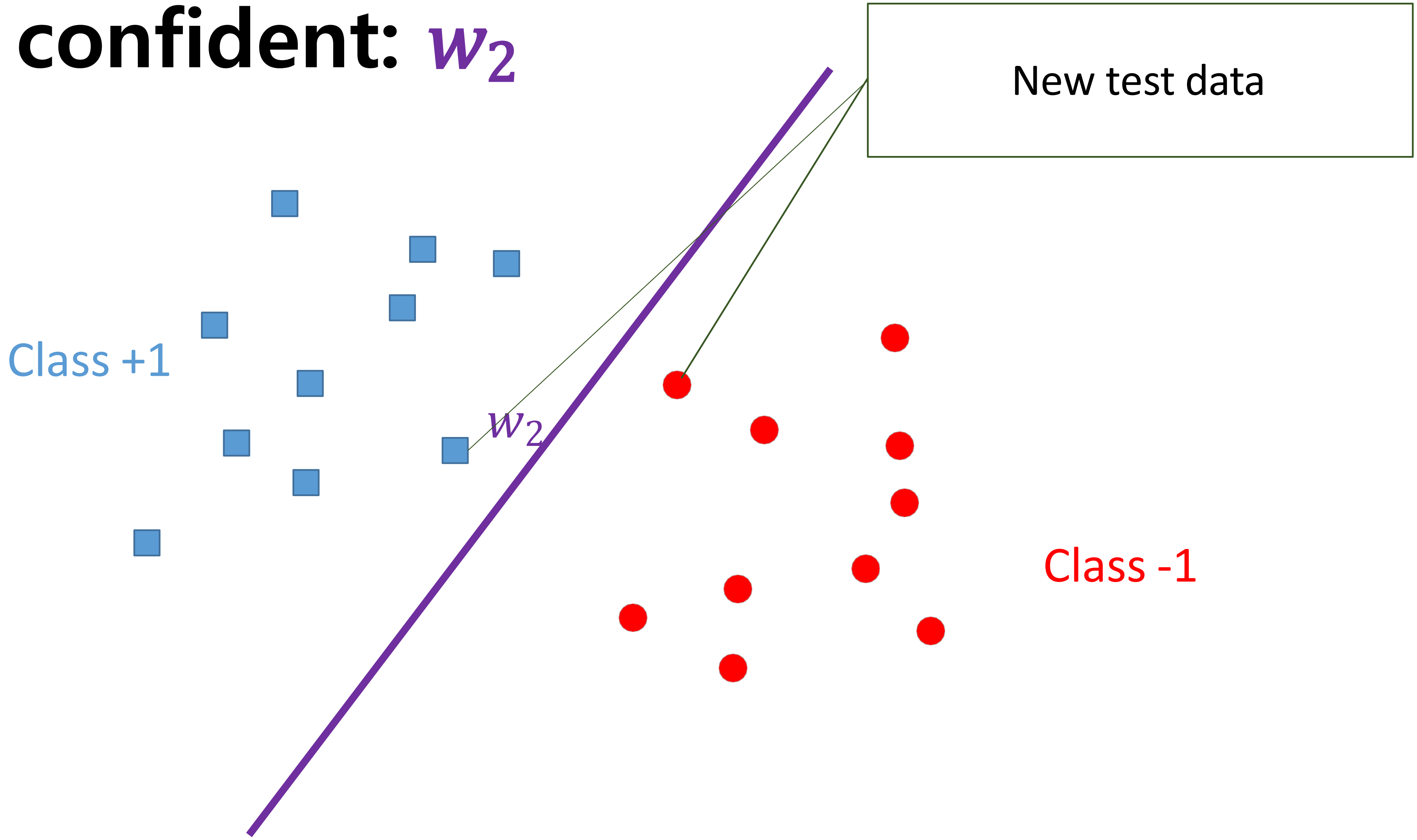
What about w_1?



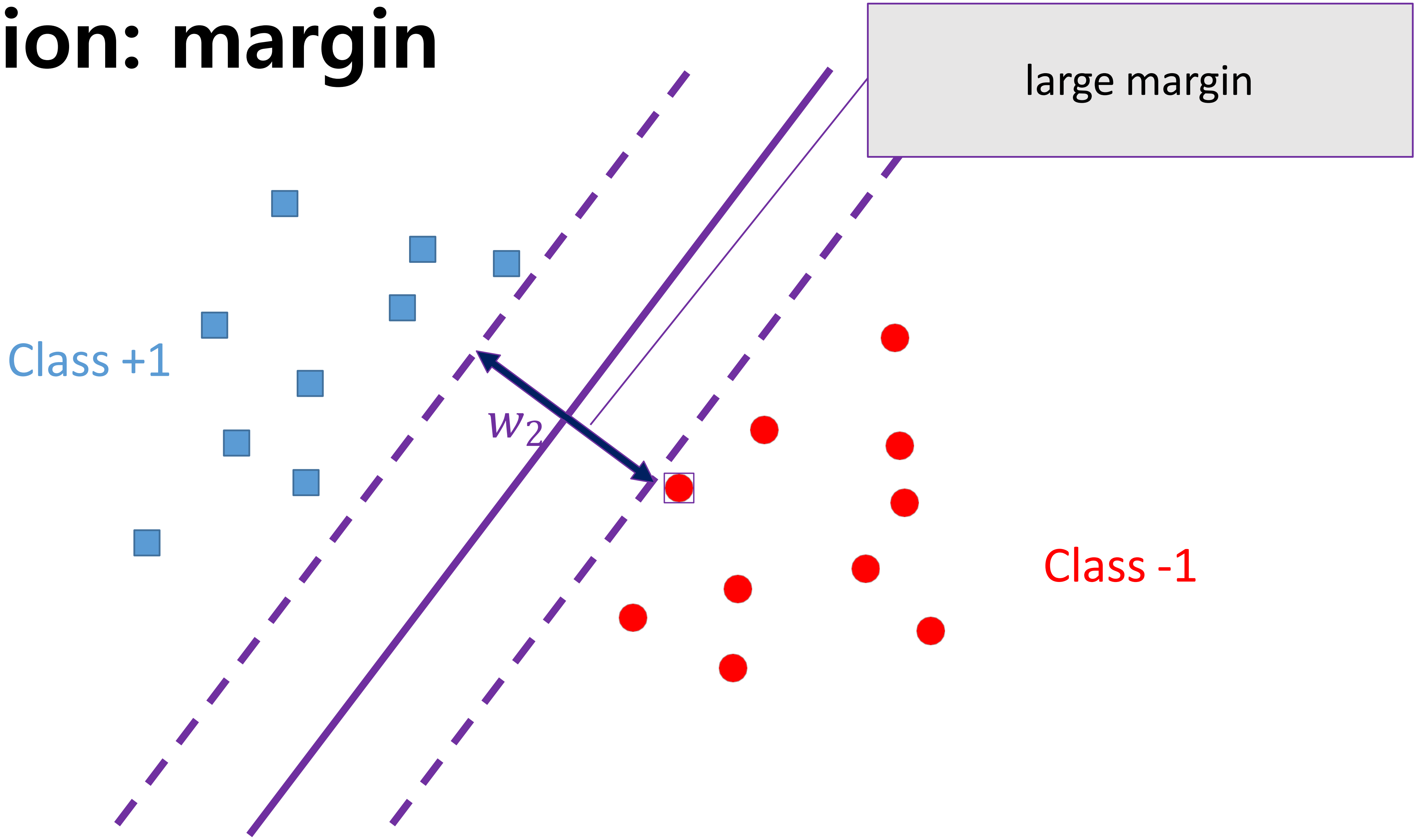
What about w_3?



Most confident: w_2



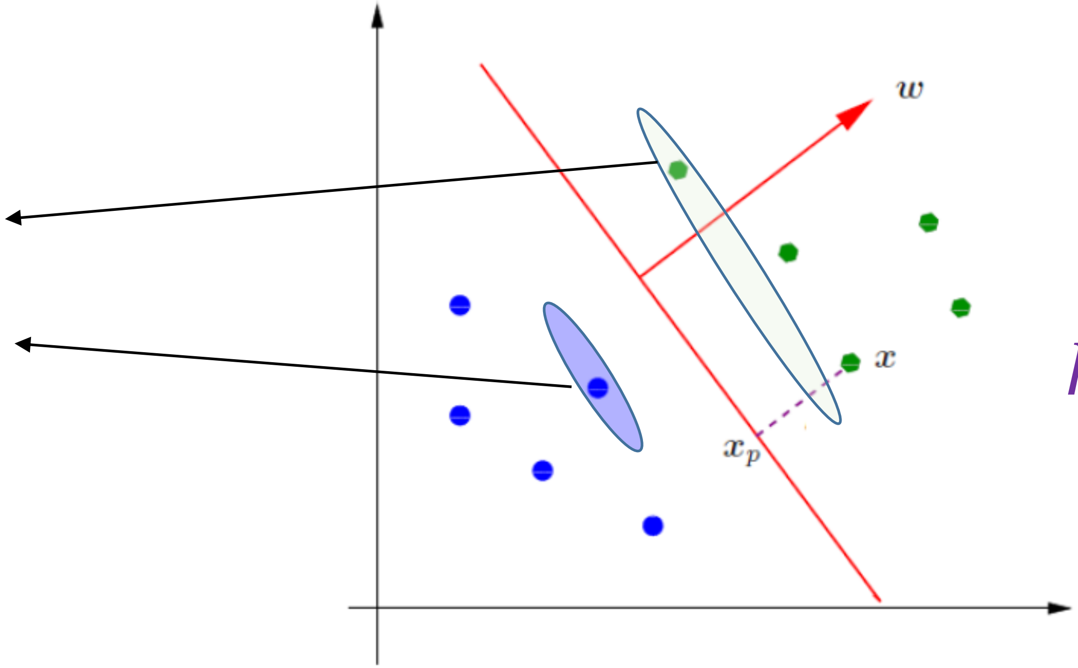
Intuition: margin



Support Vector Machine

- Choose the linear separator (hyperplane) with the largest **margin** on either side
  - Maximum margin hyperplane with **support vectors**
  - Robust to outliers

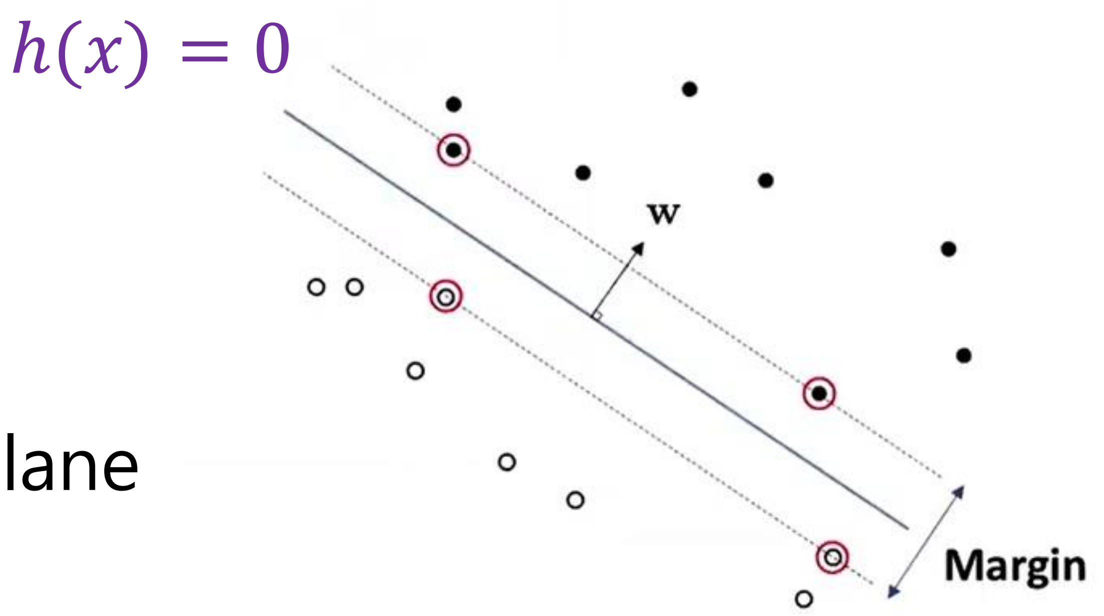
Support vector : an instance with the minimum margin, which will be the most sensible data points to affect the performance



$$h(x) = w^T x + b$$

Margin

- Twice the distance from the hyperplane to the nearest instance on either side



- w is orthogonal to the hyperplane

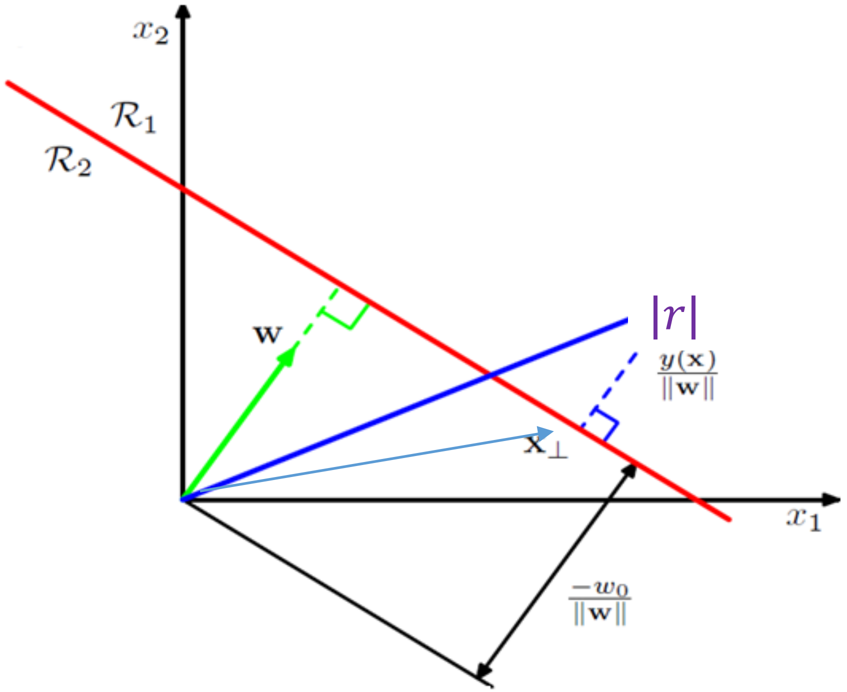
- Lemma : x has distance $\frac{|h_{w\,b}(x)|}{||w||}$

  to the hyperplane $h_{w\,b}(x) = w^T x + b = 0$

Margin distance

Proof:

- Let $x = x_\perp + r\frac{w}{||w||}$, then $|r|$ is the distance
- Multiply both sides by w^T and add b
- Left hand side: $w^T x + b = \ h_{w,b}(x)$
- Right hand side: $w^T x_\perp + r\frac{w^T w}{||w||} + b = 0 + r||w||$

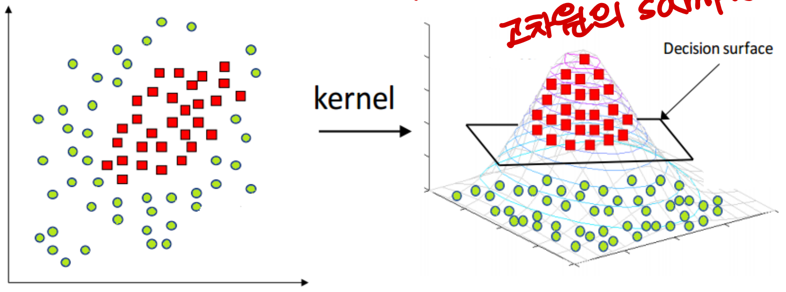


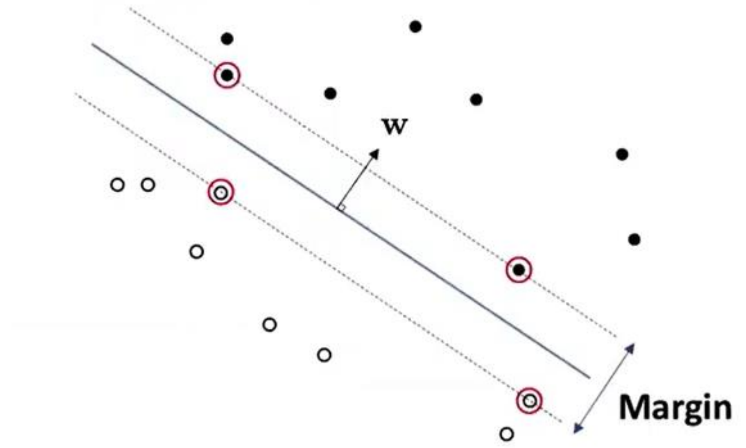
Optimization

- Optimal weight w and bias b
  - Classifies points correctly as well as achieves the largest possible margin
  - **Hard margin SVM** – assumes linear separability
  - Soft margin SVM – extends to non-separable cases
  - Nonlinear transform & kernel trick

어느정도 에러 허용

kernel을 이용해서 고차원의 sample로 매핑 ㄱㄴ





Optimization

constraints: linearly separable; hard-margin linear SVM

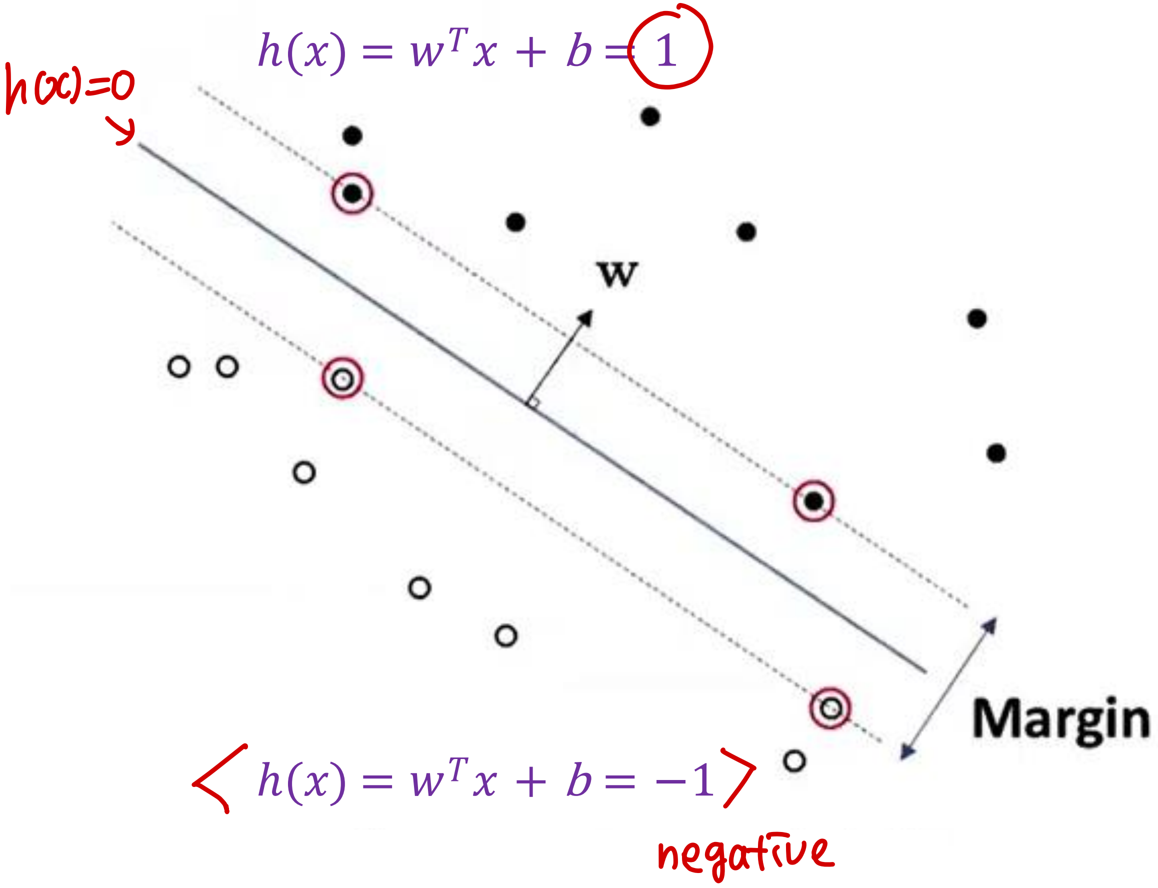
$$h(x) = w^T x + b \geq 1 \text{ for } y = 1$$

$$h(x) = w^T x + b \leq -1 \text{ for } y = \text{-}1$$

$$y(w^T x + b) \geq 1 \text{ for all samples}$$

⇓

SVM에서의 sample constraints



Optimization

objective function: linearly separable; hard-margin linear SVM

Distance from a support vector to the hyper plane

$$\frac{w^T x + b}{\|w\|} = \frac{\pm 1}{\|w\|} \longrightarrow \frac{2}{\|w\|}$$

Maximize the margin

SVM Primal problem

Equivalently, we want to minimize $\|w\|^2$ 계산의 편의상

Subject to $y_i(w^T x_i + b) \geq 1$ for all samples

Constrained optimization problem, solved by convex quadratic programming

Can be converted into an unconstrained optimization problem and dual form

Support Vector Machine

linearly separable; soft-margin linear SVM

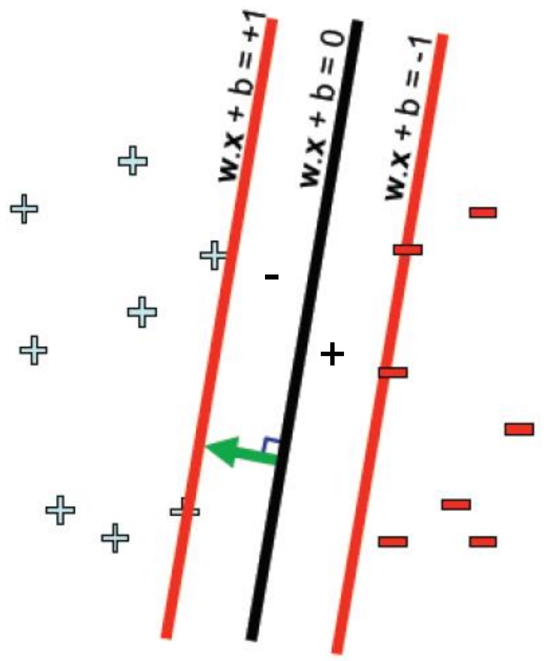
* kernel 함수

↳ linearly sepable 하지 않은 데이터 샘플들이 있다고 할 때, 그 차수를 높여 linearly sepable 하게 만드는 과정

- What if the data is not linearly separable? e.g., there are outliers or noisy measurements, or the data is slightly non-linear.

- Assign a slack variable to each instance $\xi i \geq 0$, which can be thought of distance from the separating hyperplane if an instance is misclassified and 0 otherwise.
- Minimize $\frac{1}{2}\|w\|^2 + C\sum \xi_i$

  subject to $y_i(w^T x_i + b) \geq 1\text{-}\xi_i$



a=1 for scale invariance

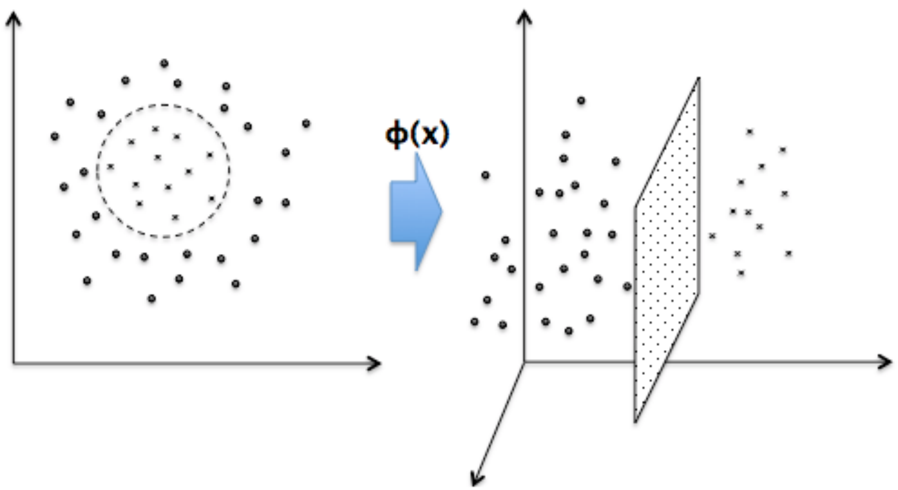
Problem of SVM

- What if the data samples are not linearly separable?
- Linearity in w

$$\sum w^T x \longrightarrow \sum w^T \varphi(x)$$

Both linear to x and w

linear to w which allows non linear version of x while maintaining the analytic capability



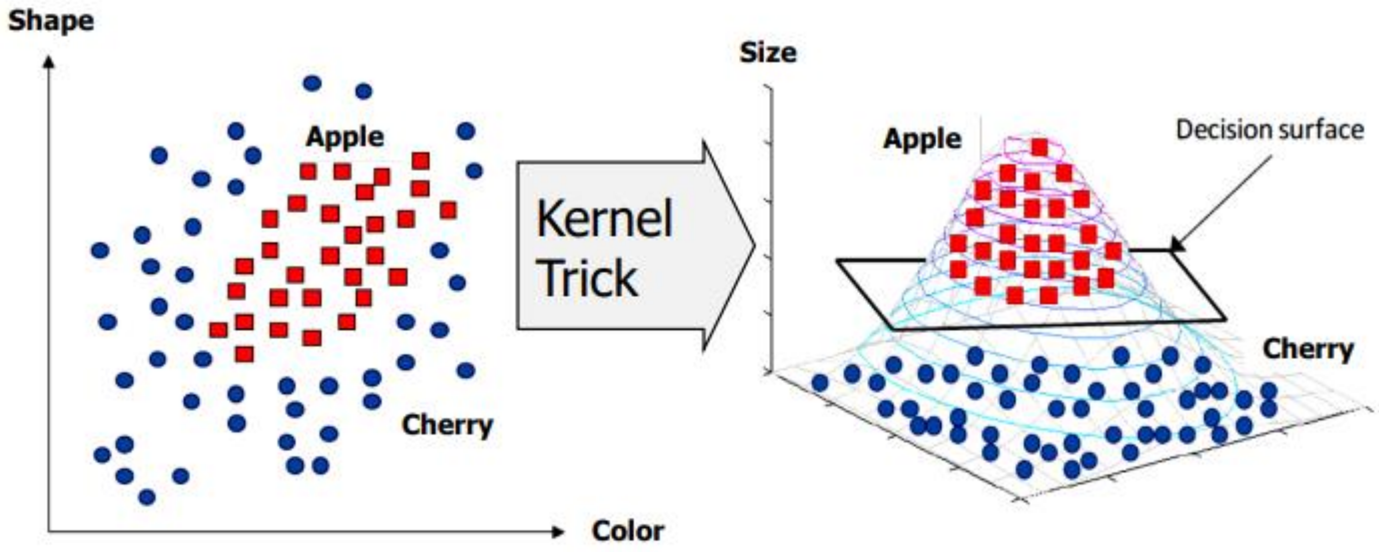
Nonlinear transforms can be applied to the input to extend the hard-margin SVM to non-linear problem

Transform input into new space and use linear model (kernel transformation)

Support Vector Machine
not linearly separable; Kernel Trick

- Data is not linearly separable in the input space

- Data is linearly separable in the feature space obtained by a kernel



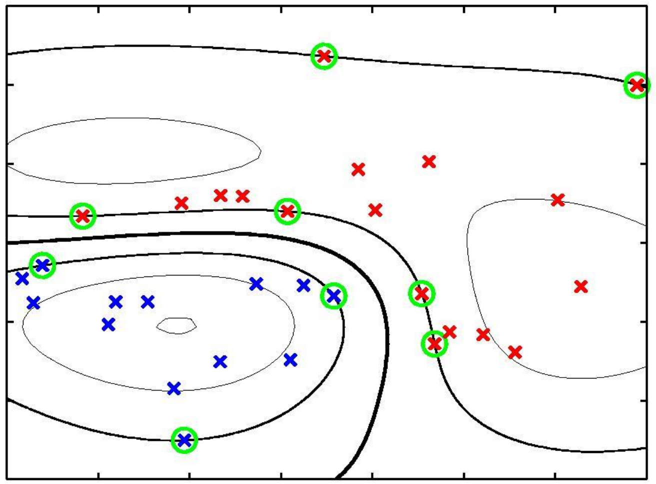
Some commonly used kernels

① Polynomial: $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}.\mathbf{y}+1)^p$

② Gaussian radial basis function (RBF) $K(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$

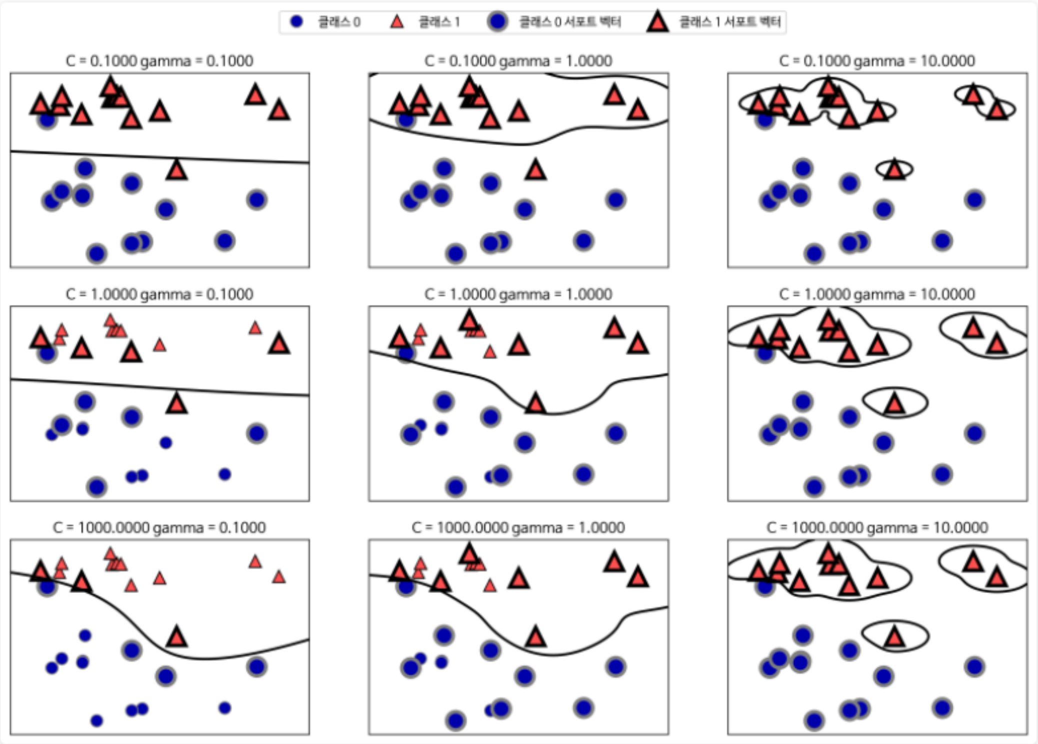
③ Hyperbolic tangent (multilayer perceptron kernel) $K(\mathbf{x},\mathbf{y}) = \tanh(k\,\mathbf{x}.\mathbf{y} - \delta)$

Parameters that the user must choose



Radial-basis function (RBF) kernel

- Radial-basis function kernel



$$\boxed{K(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma ||\mathbf{x} - \mathbf{x}'||^2\right)}$$

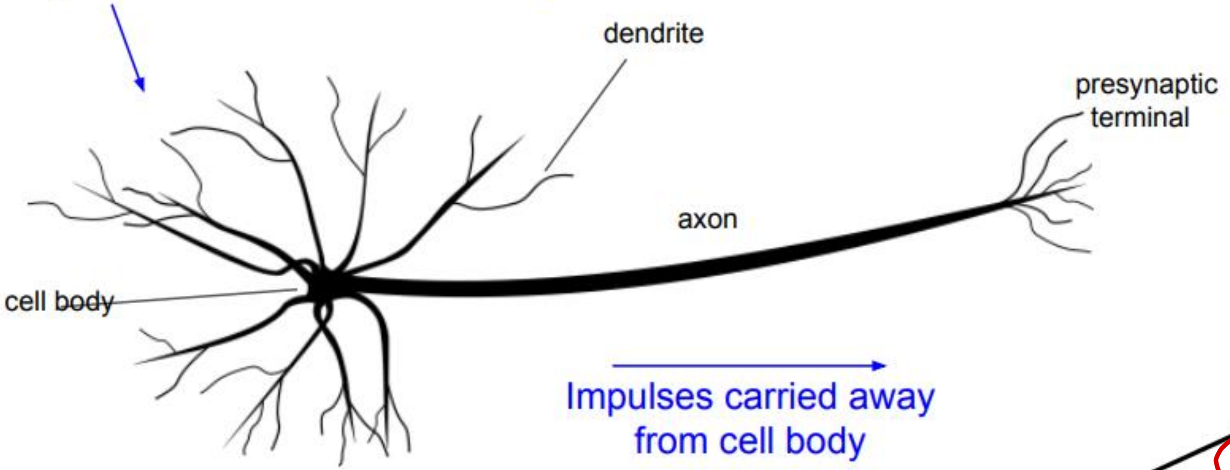
Large γ : sharp Gaussian -> overfit
Small γ : not enough linearity

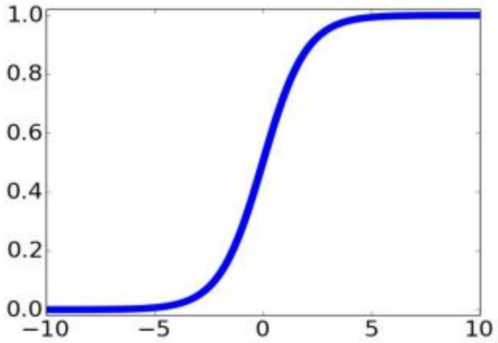
Artificial neural network (ANN)

non-linear classification model

nonlinear 제공, DNN의 기본

Impulses carried toward cell body





sigmoid activation function

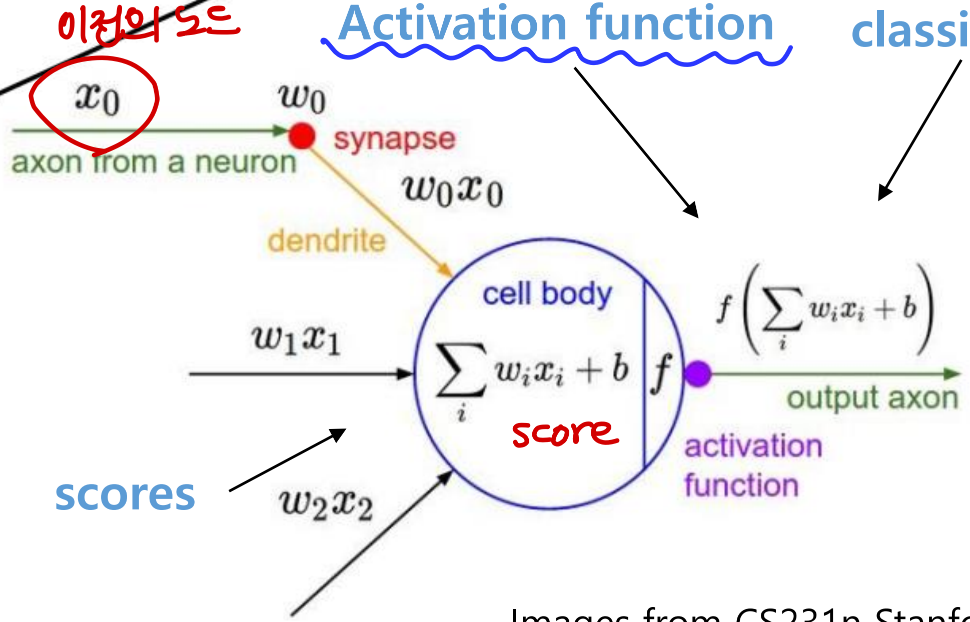
$$\frac{1}{1+e^{-x}}$$

nonlinear로 매핑

Activation function

Binary classifier

scores



Images from CS231n Stanford

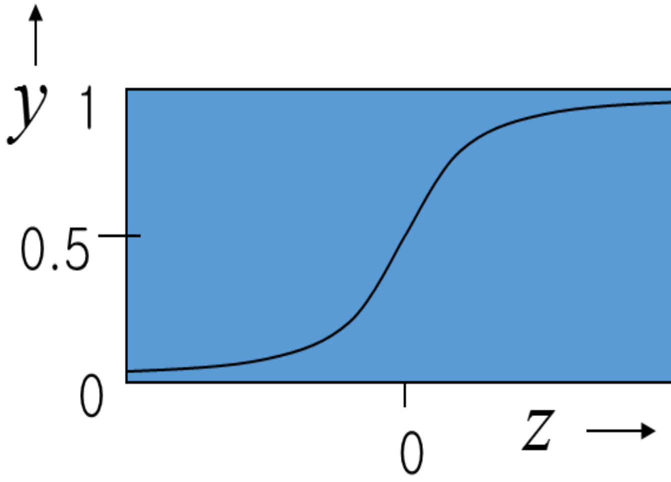
Artificial neural network (ANN)
activation functions

- **Sigmoid neurons** give a real-valued output that is a smooth and bounded function of their total input.
- **Non-linearity due to the activation functions**

Sigmoid 효과적이지 x

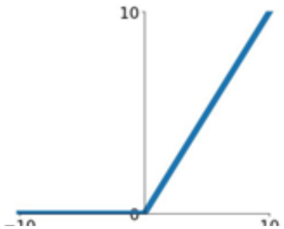
$$z = b + \sum_{i} x_i w_i$$

$$y = \frac{1}{1 + e^{-z}}$$



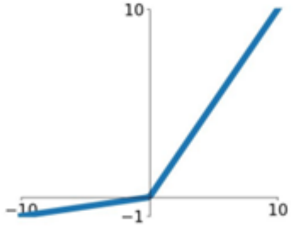
Activation functions

1 **ReLU**
$\max(0, x)$



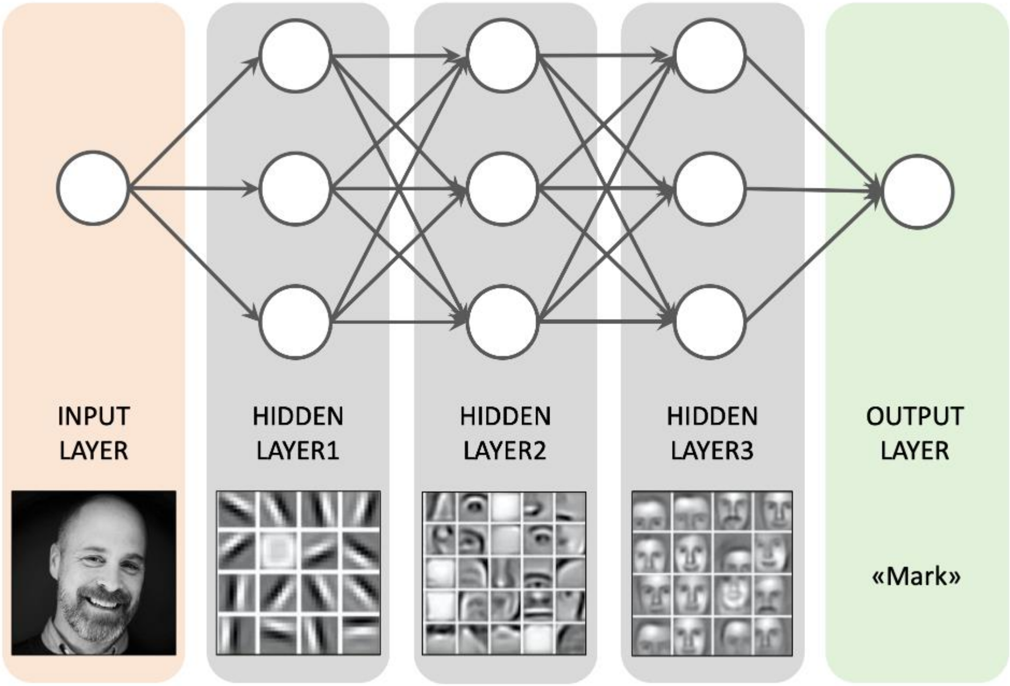
미분해도 gradient = 1 유지.

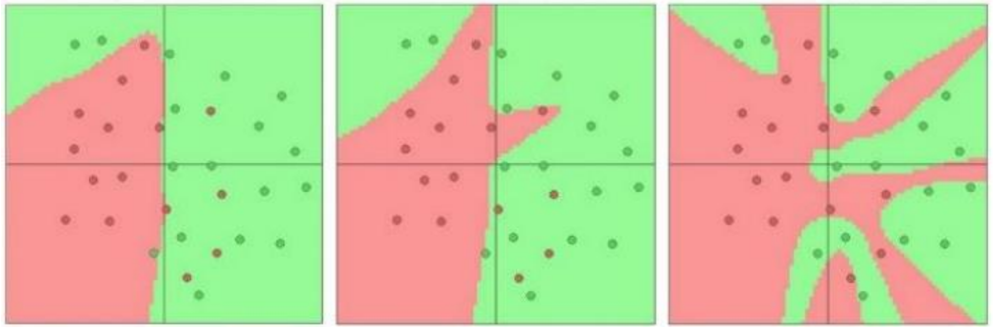
2 **Leaky ReLU**
$\max(0.1x, x)$



Images from CS231n Stanford

Artificial neural network (ANN)
deep neural network



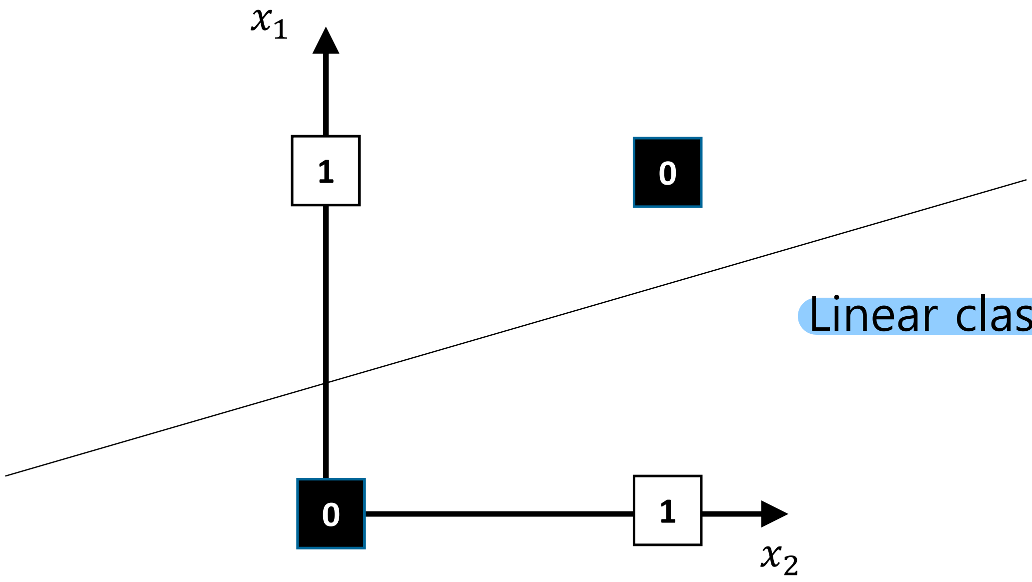


- **Can represent more complex (non-linear) boundaries with increasing neurons**

Artificial neural network (ANN)

multilayer perceptron

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

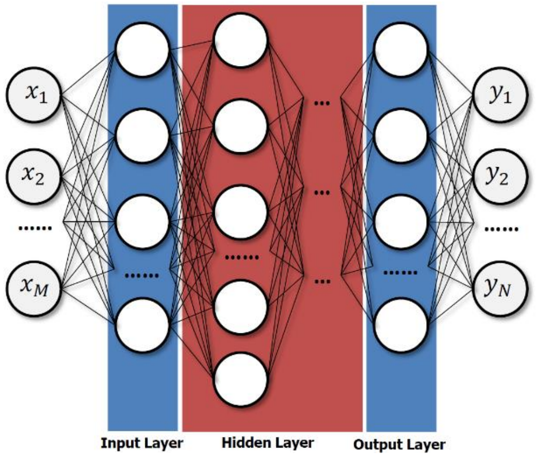


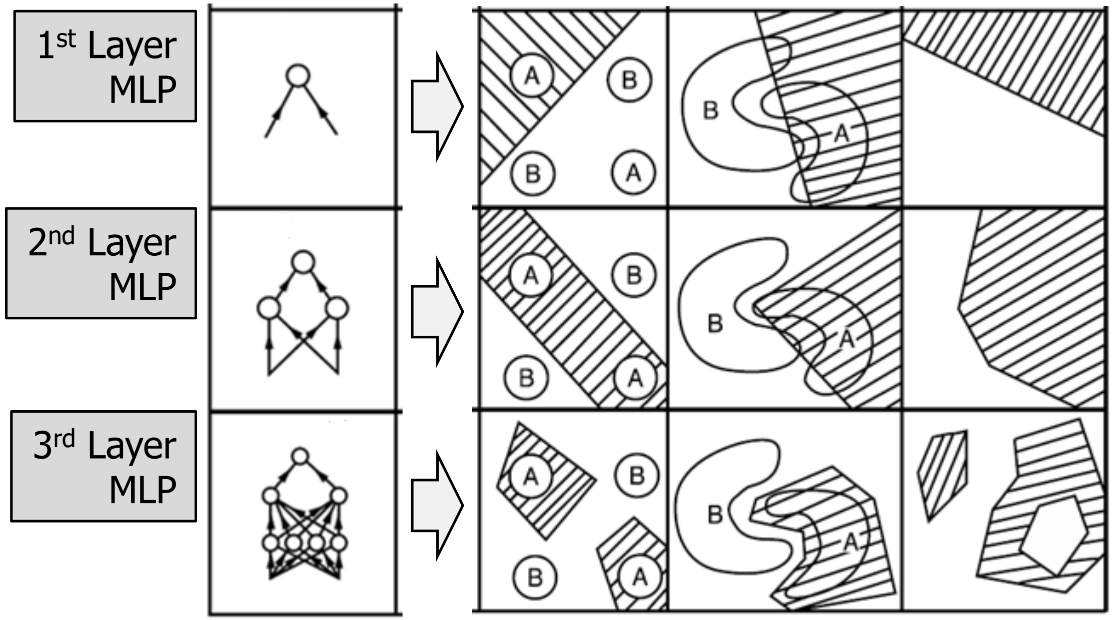
Linear classifier cannot resolve XOR!

Mathematically proven by
Prof. Marvin Minsky at MIT (1969)

Artificial neural network (ANN)
multilayer perceptron

- Multilayer Perceptron (MLP)
  - Proposed by Prof. Marvin Minsky at MIT (1969)
  - Can solve XOR Problem





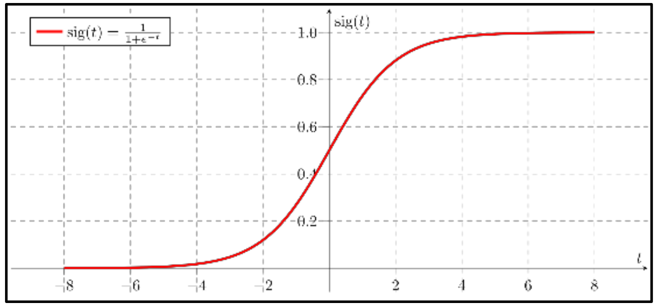
점점복잡

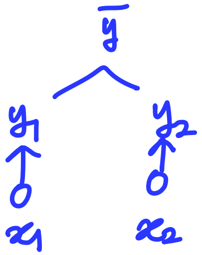
$W = \begin{pmatrix} 5 \\ 5 \end{pmatrix}, b = -8$

$W = \begin{pmatrix} -7 \\ -7 \end{pmatrix}, b = 3$

$W = \begin{pmatrix} -11 \\ -11 \end{pmatrix}, b = 6$

- $(x_1\ x_2) = (0\ 0)$
  - $(0\ 0)\begin{pmatrix} 5 \\ 5 \end{pmatrix} + (-8) = -8$, i.e., $y_1 = Sigmoid(-8) \cong 0$
  - $(0\ 0)\begin{pmatrix} -7 \\ -7 \end{pmatrix} + (3) = 3$, i.e., $y_2 = Sigmoid(3) \cong 1$
  - $(y_1\ y_2)\begin{pmatrix} -11 \\ -11 \end{pmatrix} + (6) = -11 + 6 = -5$, i.e., $\bar{y} = Sigmoid(-5) \cong 0$





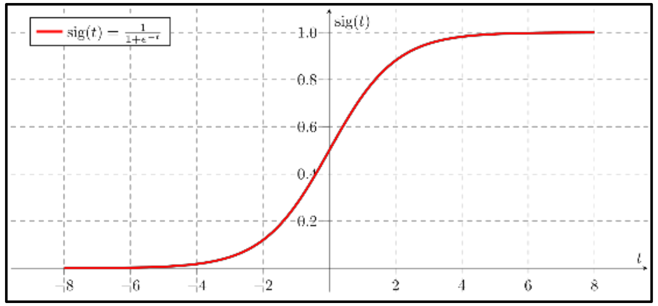
x_1	x_2	y_1	y_2	$\bar{y}$	XOR
0	0	**0**	**1**	**0**	**0**
0	1				**1**
1	0				**1**
1	1				**0**

$W = \begin{pmatrix} 5 \\ 5 \end{pmatrix}, b = -8$

$W = \begin{pmatrix} -7 \\ -7 \end{pmatrix}, b = 3$

$W = \begin{pmatrix} -11 \\ -11 \end{pmatrix}, b = 6$

- $(x_1\ x_2) = (0\ 1)$
  - $(0\ 1)\begin{pmatrix} 5 \\ 5 \end{pmatrix} + (-8) = -3$, i.e., $y_1 = Sigmoid(-3) \cong 0$
  - $(0\ 1)\begin{pmatrix} -7 \\ -7 \end{pmatrix} + (3) = -4$, i.e., $y_2 = Sigmoid(-4) \cong 0$
  - $(y_1\ y_2)\begin{pmatrix} -11 \\ -11 \end{pmatrix} + (6) = 6$, i.e., $\bar{y} = Sigmoid(6) \cong 1$



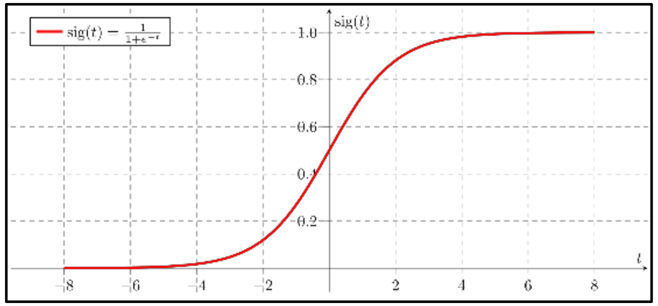
x_1	x_2	y_1	y_2	$\bar{y}$	XOR
0	0	**0**	**1**	**0**	**0**
0	1	**0**	**0**	**1**	**1**
1	0				**1**
1	1				**0**

$W = \begin{pmatrix} 5 \\ 5 \end{pmatrix}, b = -8$

$W = \begin{pmatrix} -7 \\ -7 \end{pmatrix}, b = 3$

$W = \begin{pmatrix} -11 \\ -11 \end{pmatrix}, b = 6$

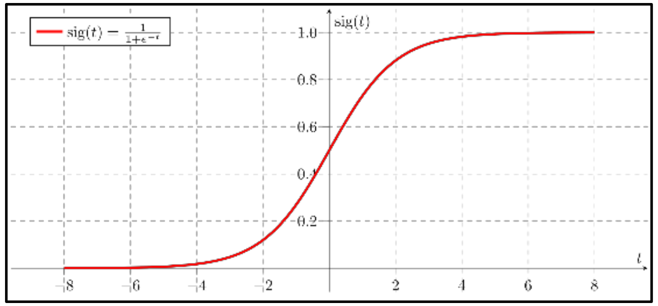
- $(x_1\ x_2) = (1\ 0)$
  - $(1\ 0)\begin{pmatrix} 5 \\ 5 \end{pmatrix} + (-8) = -3$, i.e., $y_1 = Sigmoid(-3) \cong 0$
  - $(1\ 0)\begin{pmatrix} -7 \\ -7 \end{pmatrix} + (3) = -4$, i.e., $y_2 = Sigmoid(-4) \cong 0$
  - $(y_1\ y_2)\begin{pmatrix} -11 \\ -11 \end{pmatrix} + (6) = 6$, i.e., $\bar{y} = Sigmoid(6) \cong 1$



x_1	x_2	y_1	y_2	$\bar{y}$	XOR
0	0	**0**	**1**	**0**	**0**
0	1	**0**	**0**	**1**	**1**
1	0	**0**	**0**	**1**	**1**
1	1				**0**

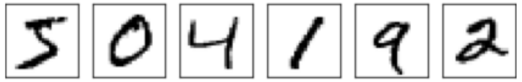
$W = \begin{pmatrix} 5 \\ 5 \end{pmatrix}, b = -8$ $\quad W = \begin{pmatrix} -7 \\ -7 \end{pmatrix}, b = 3$ $\quad W = \begin{pmatrix} -11 \\ -11 \end{pmatrix}, b = 6$

- $(x_1\ x_2) = (1\ 1)$
  - $(1\ 1)\begin{pmatrix} 5 \\ 5 \end{pmatrix} + (-8) = 2$, i.e., $y_1 = Sigmoid(2) \cong 1$
  - $(1\ 1)\begin{pmatrix} -7 \\ -7 \end{pmatrix} + (3) = -11$, i.e., $y_2 = Sigmoid(-11) \cong 0$
  - $(y_1\ y_2)\begin{pmatrix} -11 \\ -11 \end{pmatrix} + (6) = -5$, i.e., $\bar{y} = Sigmoid(-5) \cong 0$

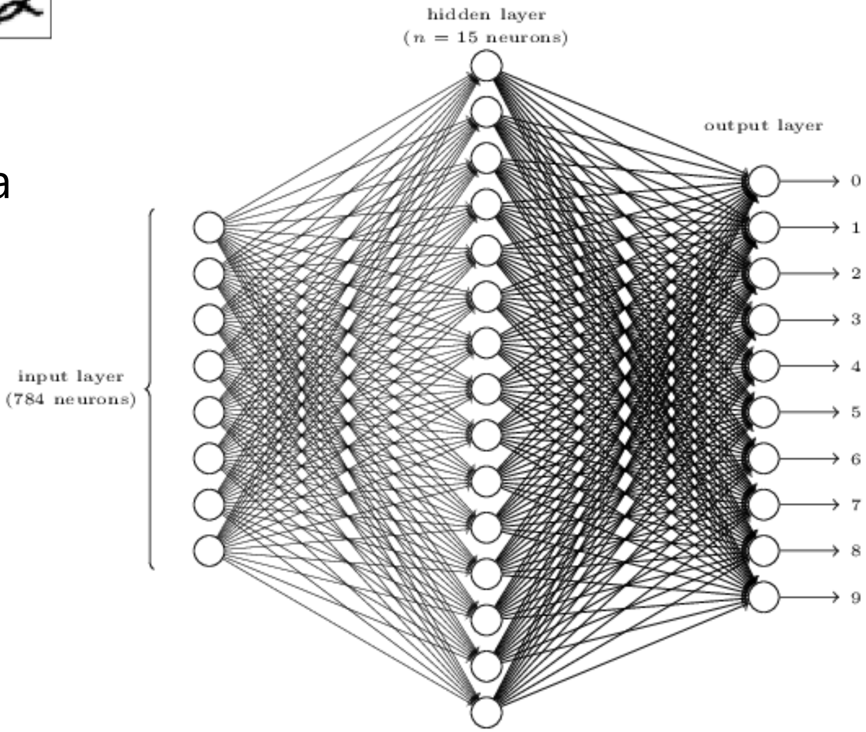


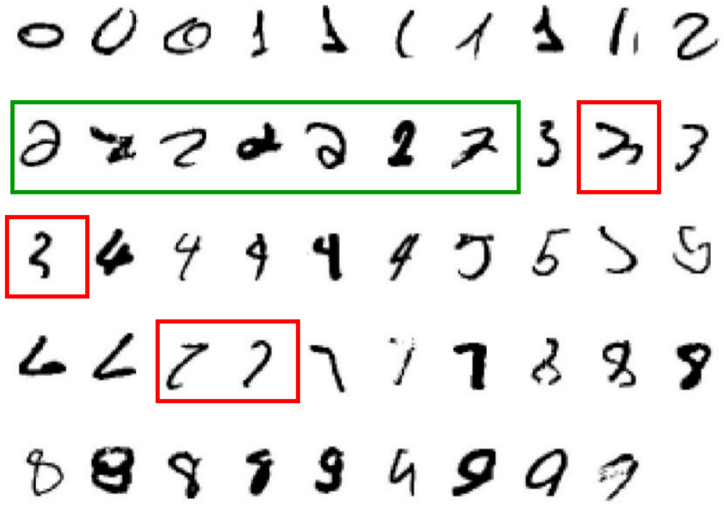
x_1	x_2	y_1	y_2	$\bar{y}$	XOR
0	0	0	1	0	0
0	1	0	0	1	1
1	0	0	0	1	1
1	1	1	0	0	0

Example: MNIST data recognition



28x28 input image
->1x28x28 input data vector



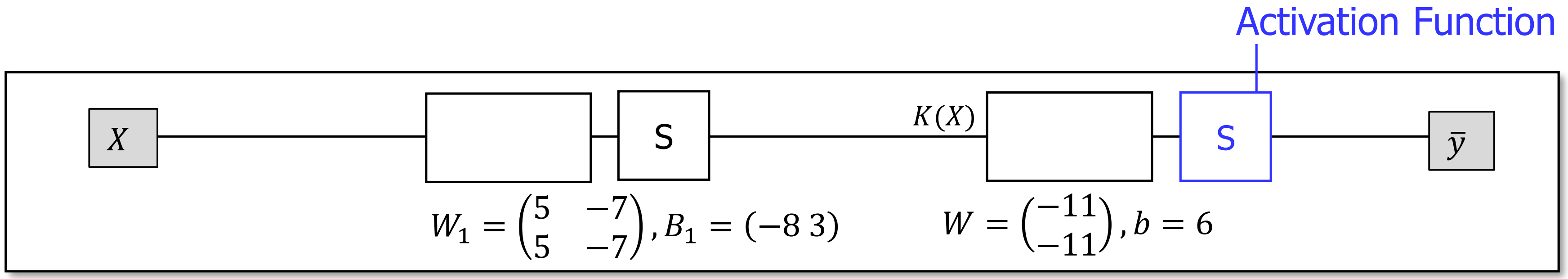


Class of 1~10 Digits

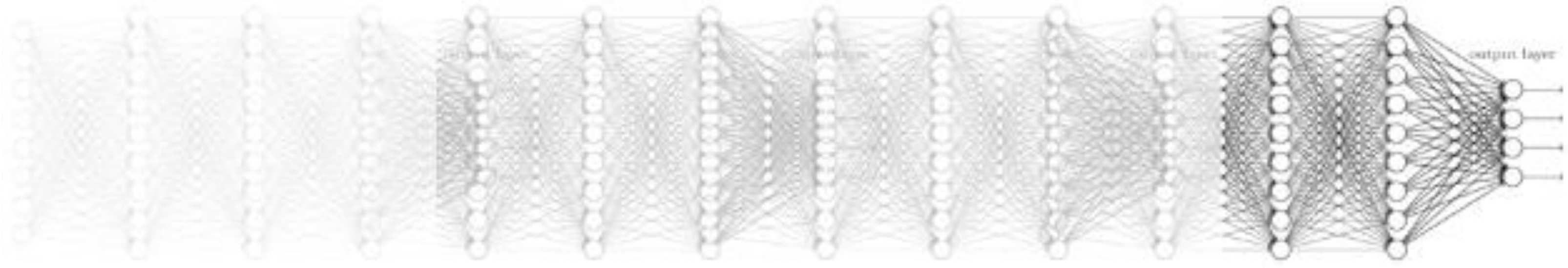
MNIST data used for Optical Character Recognition (OCR)

Artificial neural network (ANN)

ANN for non-linear problem



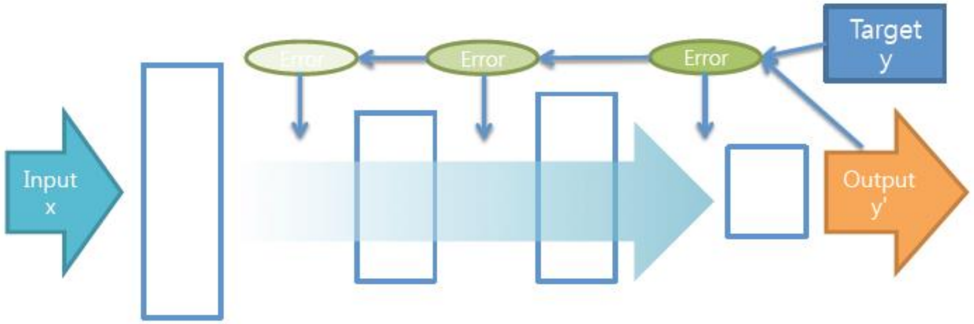
- Observation
  - There exists cases when the accuracy is low even if the # layers is high. Why?
  - Answer
    - The result of one ANN is the result of sigmoid function (**between 0 and 1**).
    - The numerous multiplication of this result converges to near zero.
      → **Gradient Vanishing Problem**!!



$$\frac{\partial error}{\partial w1} = \frac{\partial error}{\partial output} * \frac{\partial output}{\partial hidden2} * \frac{\partial hidden2}{\partial hidden1} * \frac{\partial hidden1}{\partial w1}$$

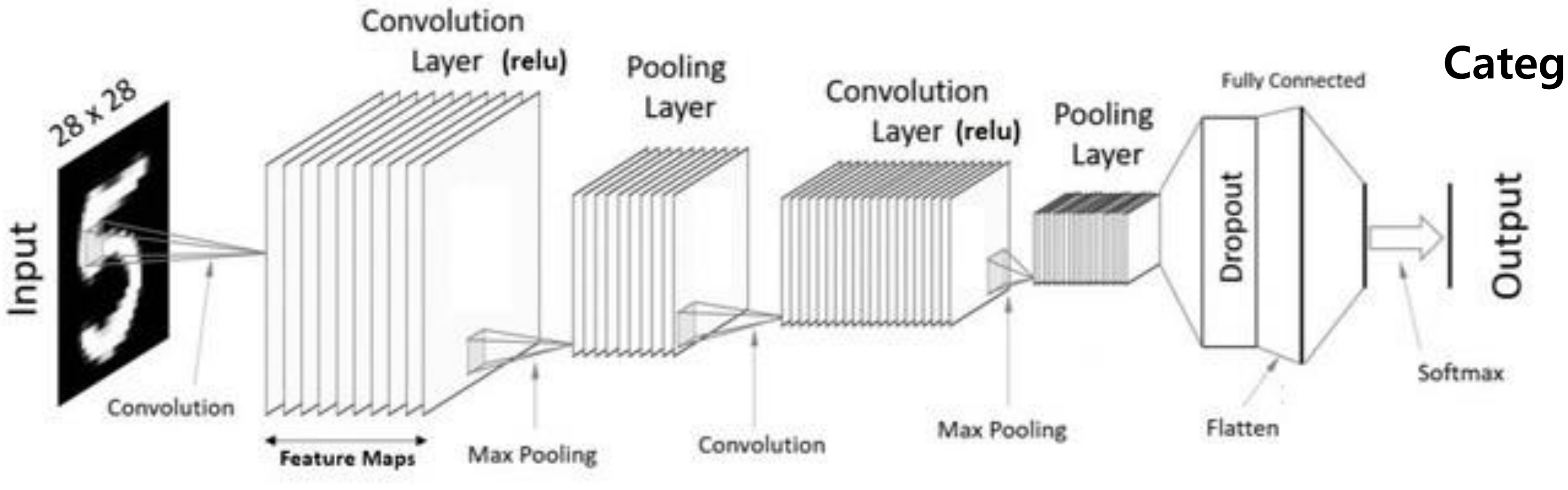
Breakthrough in Back Propagation

- Backpropagation (BP) barely changes lower-layer parameters (vanishing gradient)
- Breakthrough
    - Pre-training+ fine tuning
    - Convolutional neural networks (CNN) for reducing redundant param eters.
    - Rectified linear unit (constant gradient propagation)
    - Dropout

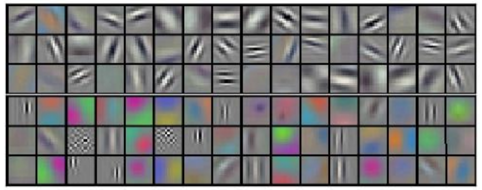


Convolutional neural network

State-of-the-art classification model for high-dimensional data (image, video, etc.)



Categorical cross-entropy loss



Feature extraction

Classification
(fully connected layers)

Reference

- Book: Pattern Recognition and Machine Learning (by Christopher M. Bishop)
- Book: Machine Learning: a Probabilistic Perspective (by Kevin P. Murphy)
- https://www.andrewng.org/courses/