# Lecture 14. ANNs : part II.

## 1. Machine learning components.

$$x \longrightarrow \boxed{f} \longrightarrow \hat{y} \longleftrightarrow y$$

1) Data : $\underline{(x, y)}$ → input data pair.

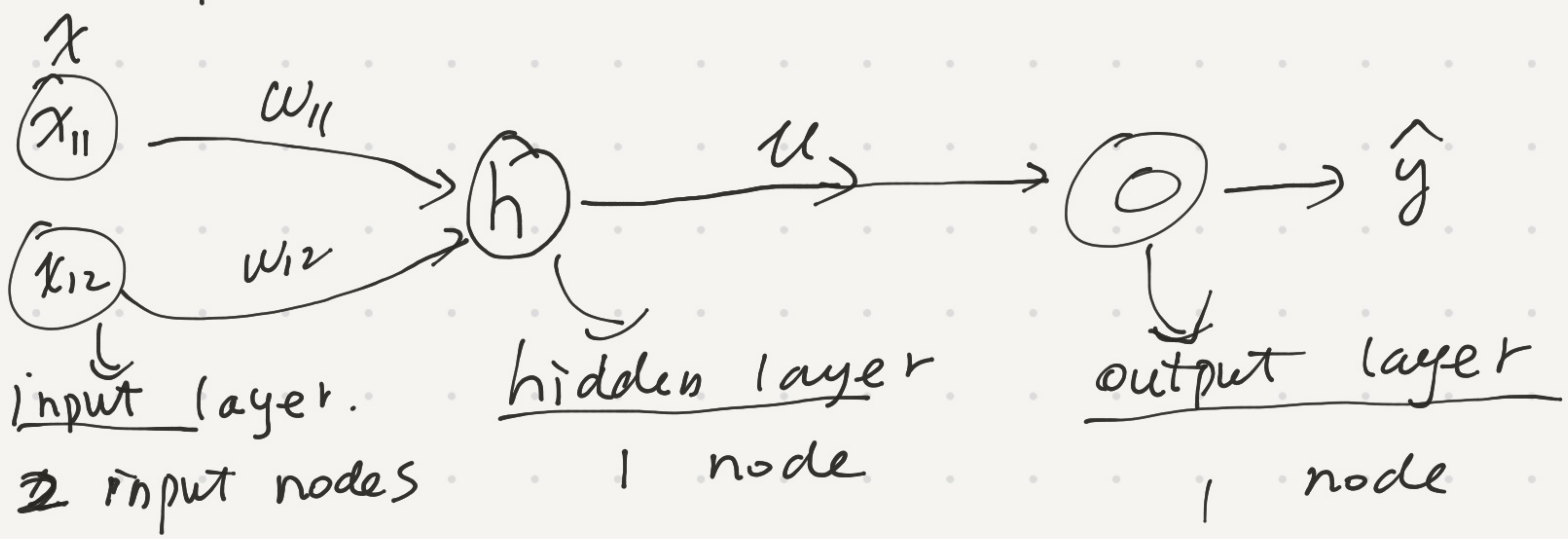
2) Model : $f$ : maps $x$ to $y$.
logistic regression: $f(x) = \frac{1}{1+e^{-(w^T x)}}$

$= \sigma(w^T \cdot x)$

3) Loss function: $\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$ MSE

4) Optimizer / optimization Algorithms: find a model $f$ than minimizes the loss function $\mathcal{L}$

## 2. A simple NN.




2.1. input nodes: No specific operations

the # of input nodes = the # of features

2.2 Hidden layer: extracts a set of features from the input data.


We could have multiple hidden layers in a NN; multiple hidden nodes in each layer.

Net input: $Net_H = x_{11} \cdot w_{11} + x_{12} \cdot w_{12} = w^T \cdot x$

Activation function: $g_H(Net_H) = \frac{1}{1+e^{-net_H}}$

non-linear function

logistic / sigmiod

## 2.2. Output layer: Converts hidden output to prediction $\hat{y}$

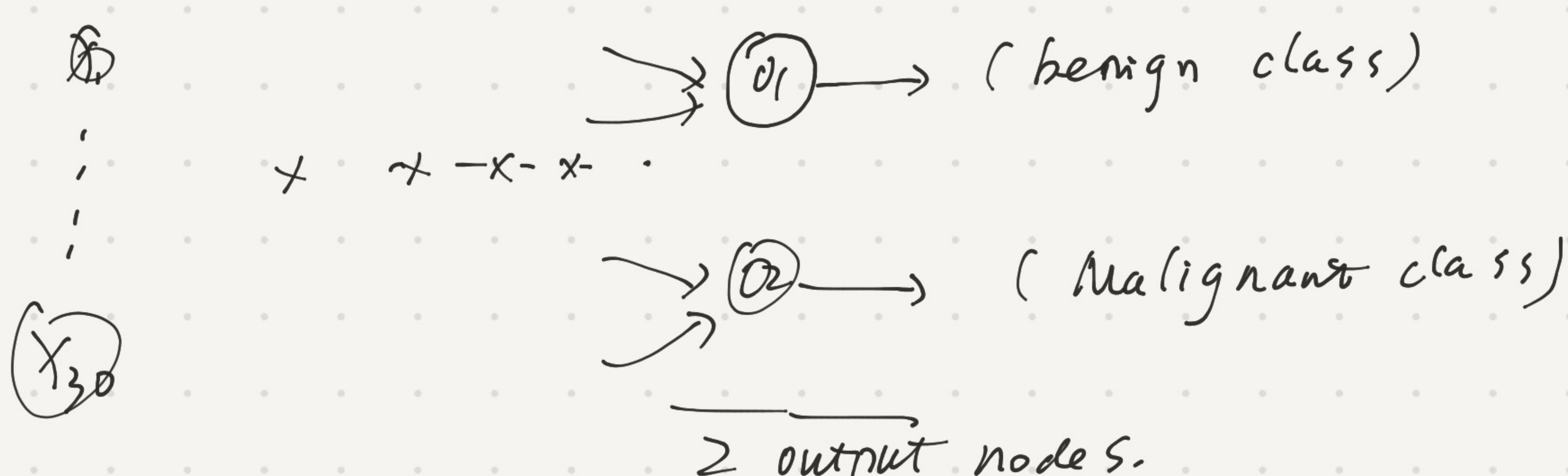
$h \xrightarrow{U} \left( Net_o \,|\, g_o \right) \xrightarrow{O}$ $\quad Net_o = h \cdot U. \quad$ $g_o(Net_o)$: activation function

We can have only one output layer in a NN, and we can have mutiple output nodes.

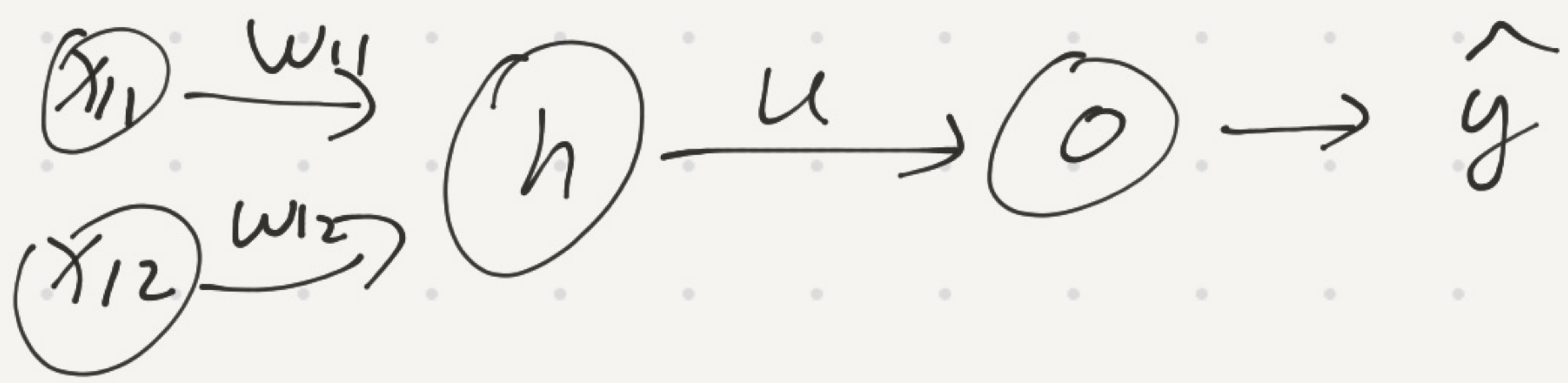
How many output nodes do we need in a NN?

classification: the number of output nodes = the number of classes.

For example: Breast Cancer Classification



## 2.3 Final model of the simple NN




$$f(x) = g_o(Net_o) = g_o(u \cdot \underline{\underline{h}})$$

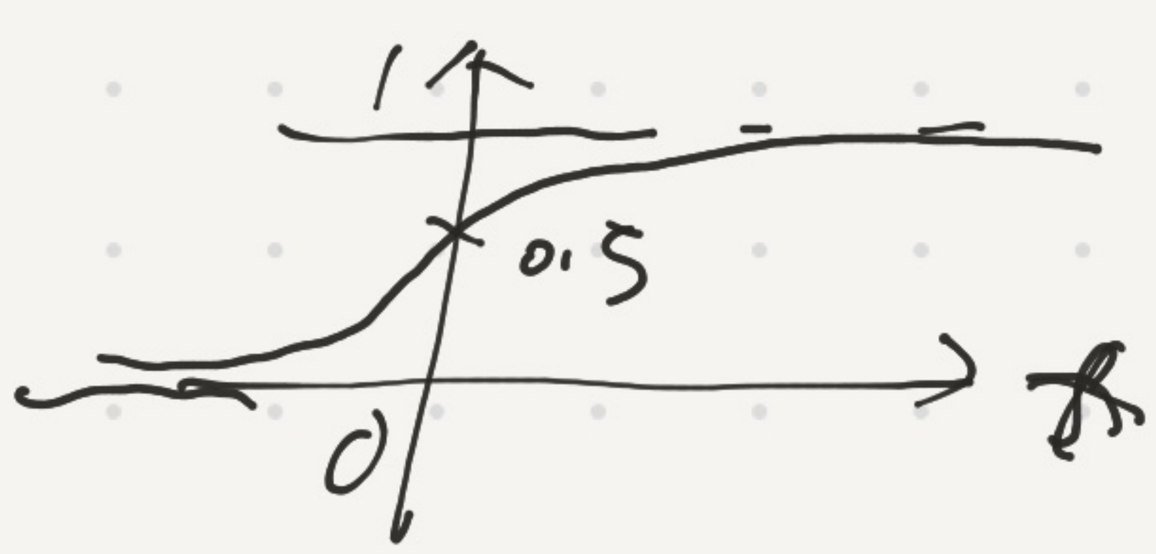
$$= g_o(u \cdot g_H(\underline{Net_H}))$$

$$= \underline{g_o(u \cdot g_H(x_{11} \cdot w_{11} + x_{12} \cdot w_{12}))} = \hat{y}$$

## 2.4 popular activation functions

1) sigmoid function

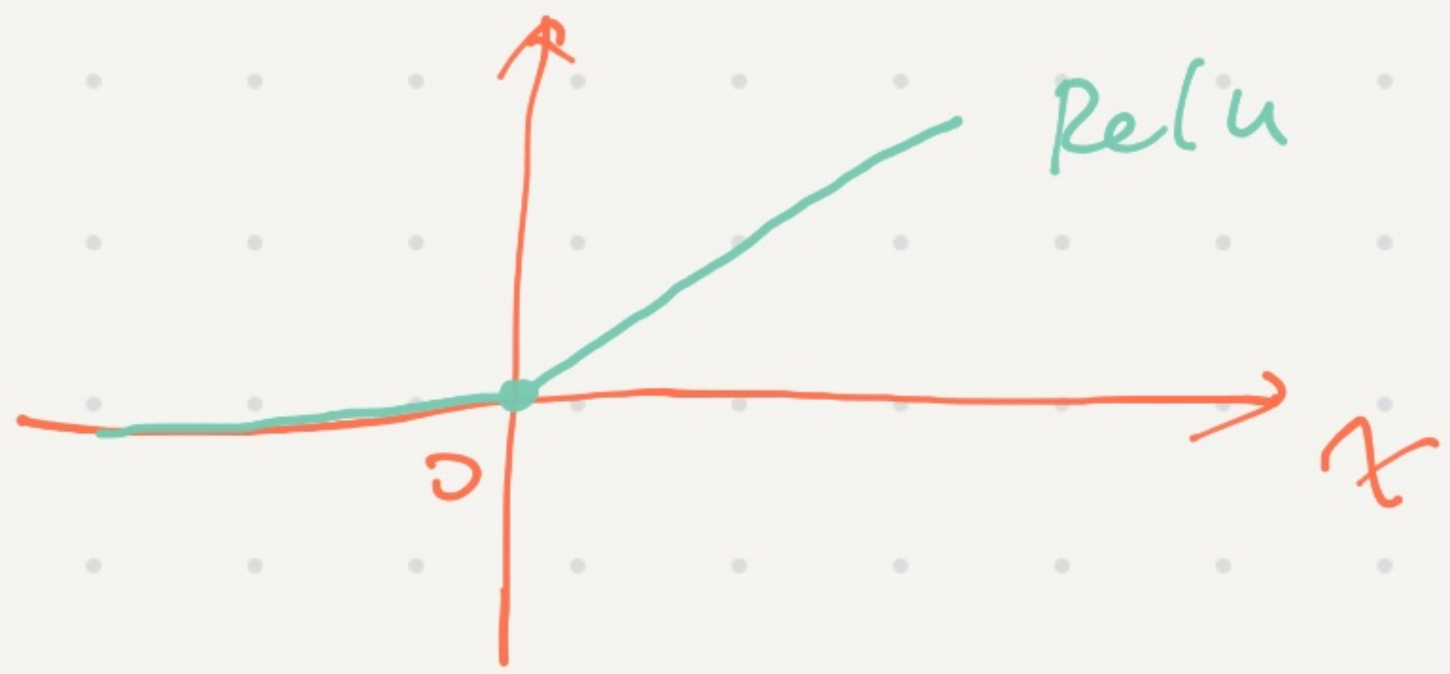
$$g(x) = \frac{1}{1+e^{-x}}$$




we can use it for both hidden and output nodes

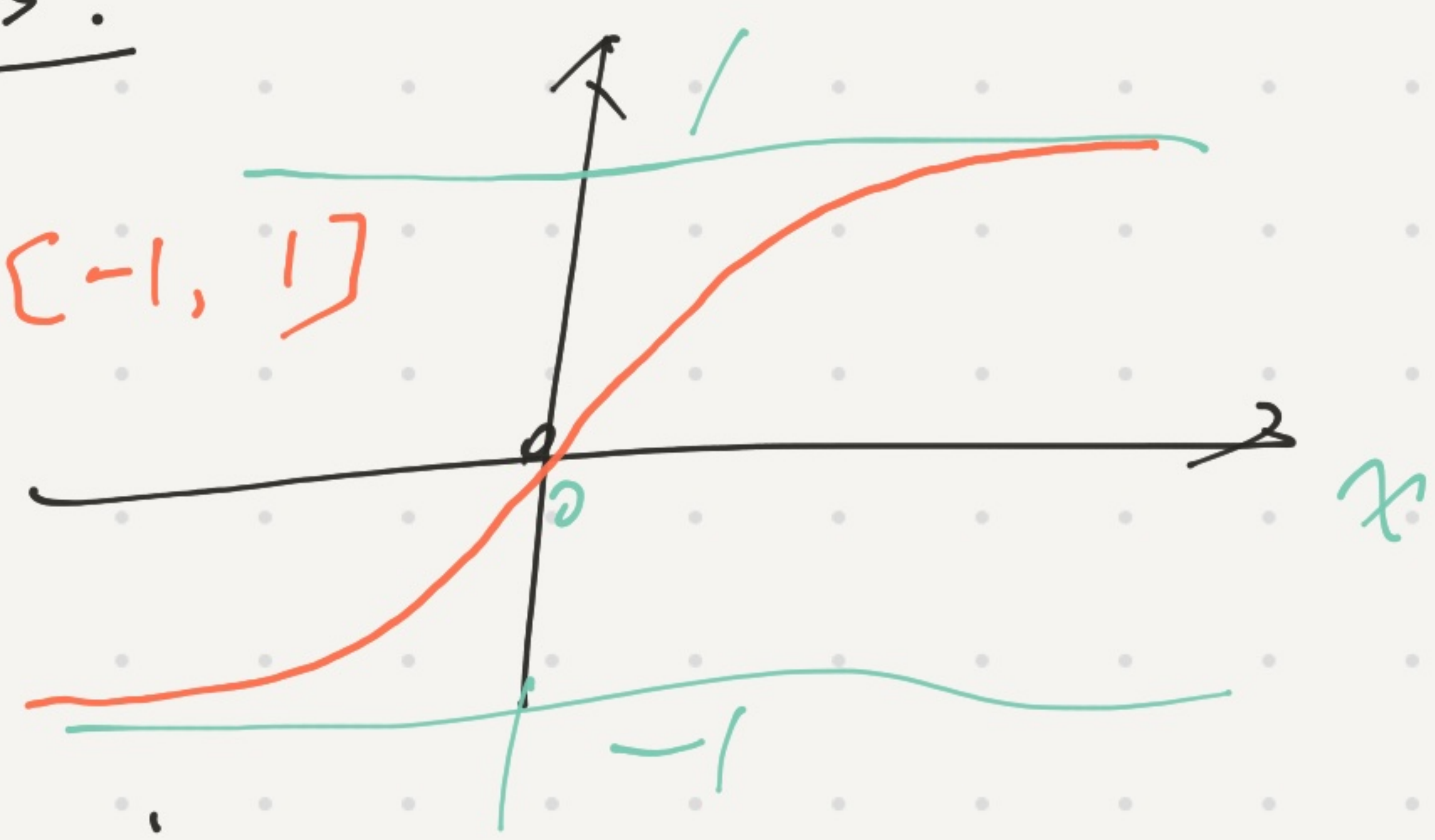
2) Rectified linear units (Relu)

$$g(x) = \max\{0, x\} = \begin{cases} 0, & \text{if } x < 0 \\ x & \text{otherwise} \end{cases}$$




was proposed for deep NNs. and used for hidden layers.

3) $\tanh(x) \in [-1, 1]$




both hidden and output layer.

4) softmax$(X)$ → output layer.

input

$$X = \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix}$$

$$\text{softmax}(X) = \begin{pmatrix} S(x_1) \\ S(x_2) \\ \vdots \\ \vdots \\ S(x_n) \end{pmatrix}$$

$\forall i$ $\quad S(x_i) = \dfrac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \in (0, 1]$

$S(x_1) + S(x_2) + \cdots S(x_n) = 1$

3. Loss function     Binary Cross-entropy

Classification: $\mathcal{L} = -\sum_{i=1}^{N} \left[ \underbrace{y_i \log \hat{y}_i}_{1} + \underbrace{(1-y_i) \log (1-\hat{y}_i)}_{2} \right]$

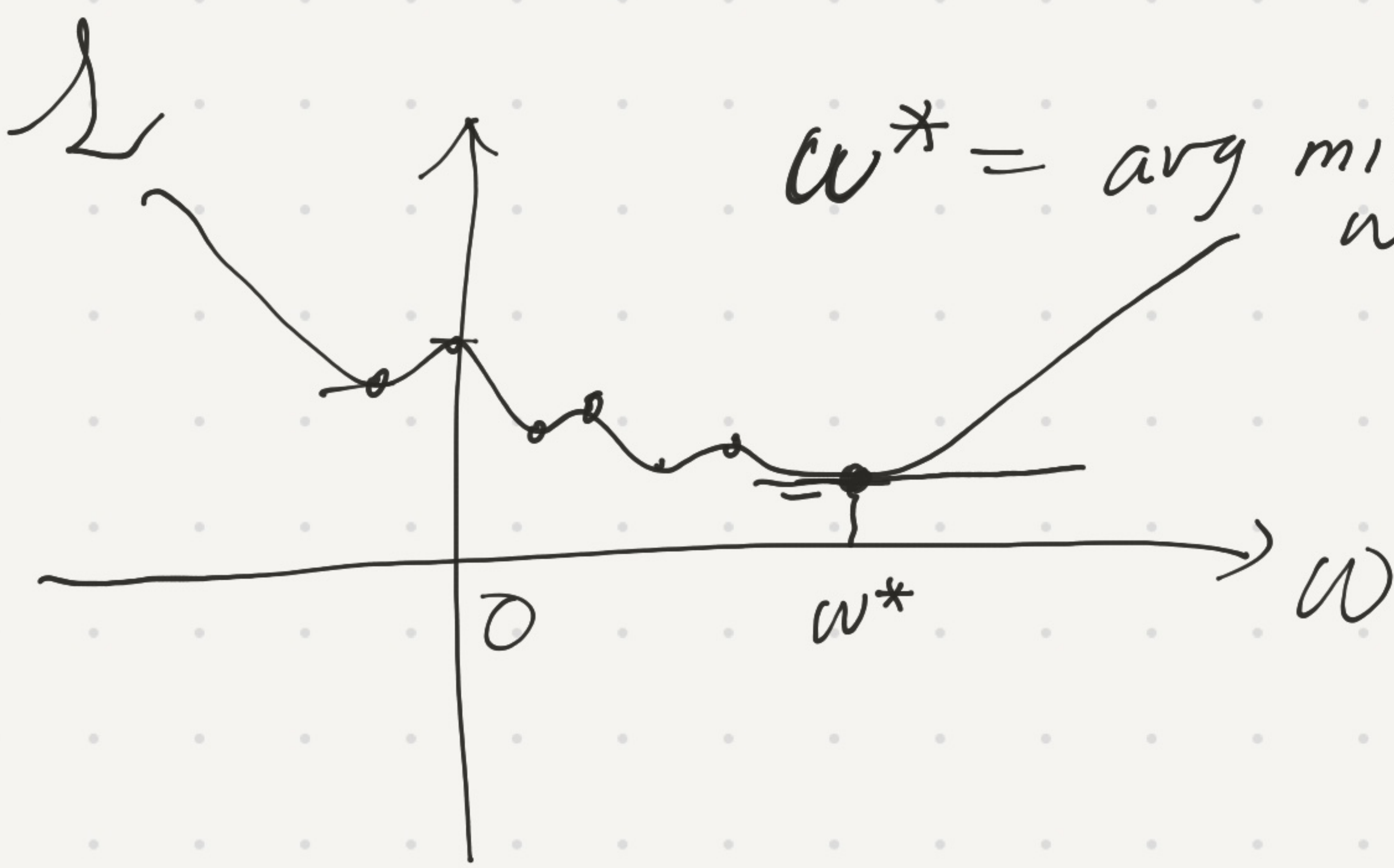
Regression:     MSE

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

4. Optimizers: learn model paramers than minimize $\mathcal{L}$

$f(x) \longrightarrow (w_{11}, w_{12} \text{ and } u)^T = \mathbf{w}$

model     parameters:




$$w^* = \arg\min_{w} \mathcal{L} = \arg\min_{w} \underbrace{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$$

Find $w^*$.

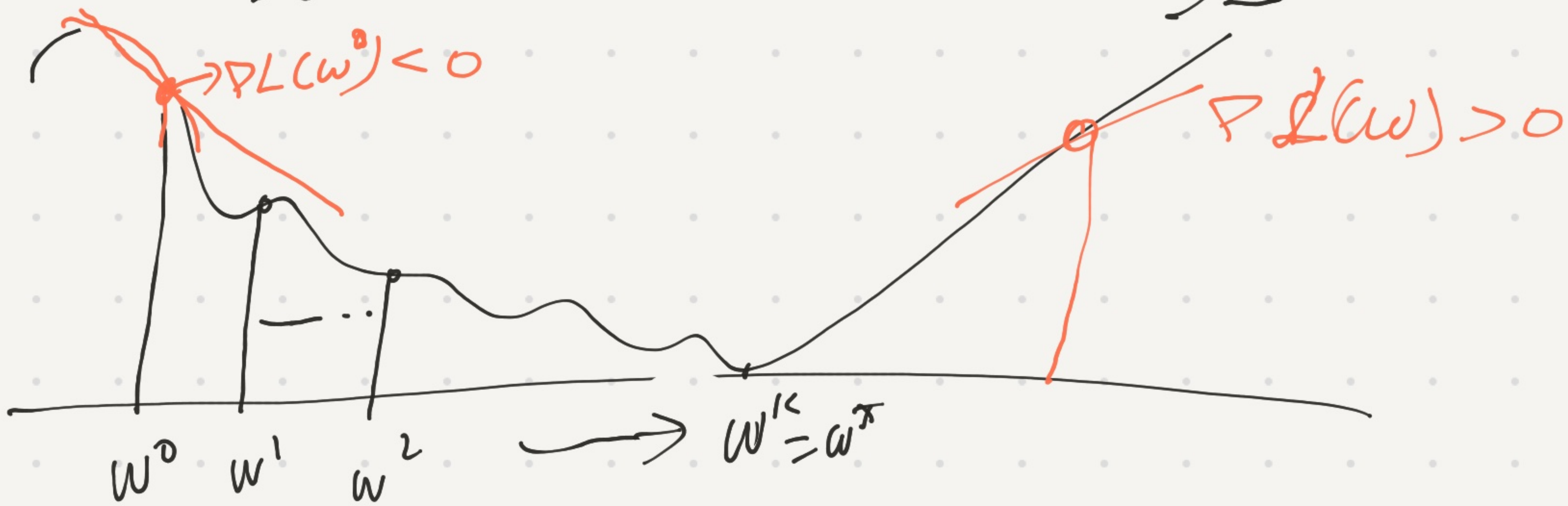
1). $\underline{\nabla \mathcal{L}(w^*) = 0}$

2) Find $w^*$ iteratively.

Find a sequence $\longrightarrow$ initial parameter.

$w^0, w^1, w^2, w^3, \cdots\cdots, w^k = w^*$

$L(w^0) > L(w^1) > L(w^2) > L(w^3) \cdots > L(w^*)$




produce the sequence:

1) moving direction

1) step size

$\longrightarrow$ Gradient descent: (GD)

$\nabla L(w^0) > 0 \longrightarrow$ move to negative direct.

$\nabla L(w^0) < 0, \longrightarrow$ move to the positive direction