

Understanding Neural Network

Dr. Chandranath Adak

IIT Patna

Prerequisite

- What is supervised learning?
- Classification and regression
- Features, labels, classes
- Training, testing and validation datasets

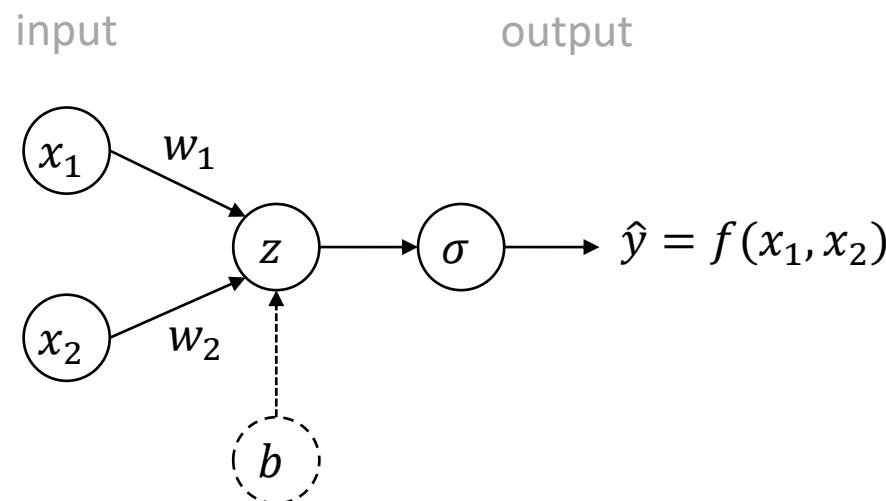
Neural Network

- A powerful tool used for supervised learning
- Imitates how human brain works
- Consists of
 - A set of artificial neurons
 - models the neurons in a biological brain
 - A set of connections between the artificial neurons
 - models synapses, able to transmit a signal to other neuron
- Is used to learn a function maps input parameters (often termed as a set of features) to outputs

Applications

Application	Input	Output	Type of Neural Network
Activity recognition	Sensor data	Activities: walking, standing, etc.	NN
Car price prediction	Features of the car	Price of the car	NN
OCR: Handwriting recognition	Handwritten character image	Identify the character	NN/ CNN
Medical Imaging	X-ray images	TB / non-TB	CNN
NLP: Machine translation	English language	French language	RNN
Speech recognition	Audio signal	Text transcript	RNN

General Architecture

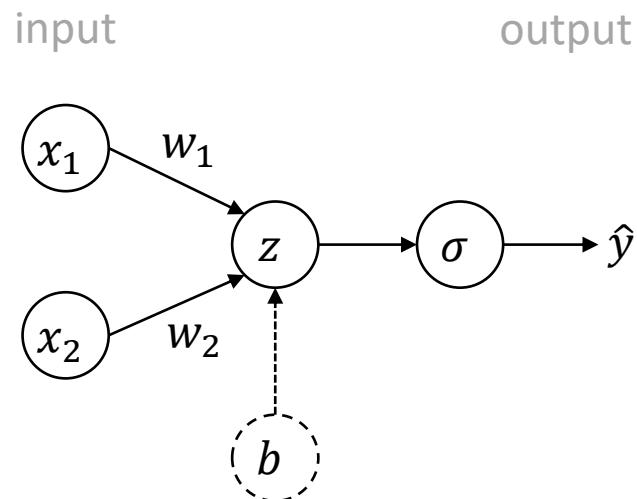


Objective: To learn a function that maps the inputs to the outputs, given training samples.

A training sample (x_1, x_2) , y
input features target output

Predicted output: \hat{y}
Learning parameters: w_1, w_2, b

Working Principle

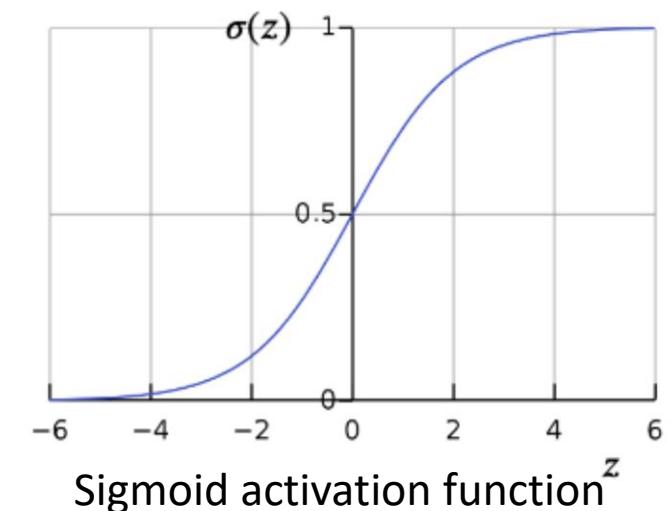


Feed forward:

- Linear operation: $z = w_1x_1 + w_2x_2 + b$
(Linearity is not always sufficient)
- Non-linearity is introduced through Activation function
 - Sigmoid activation function: $\sigma(z) = \frac{1}{1+e^{-z}}$
- The output is predicated through a series of linear and non-linear functions
- Finally, loss is calculated as a function of the predicted and actual output
 - half-squared loss: $L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$

Backpropagation:

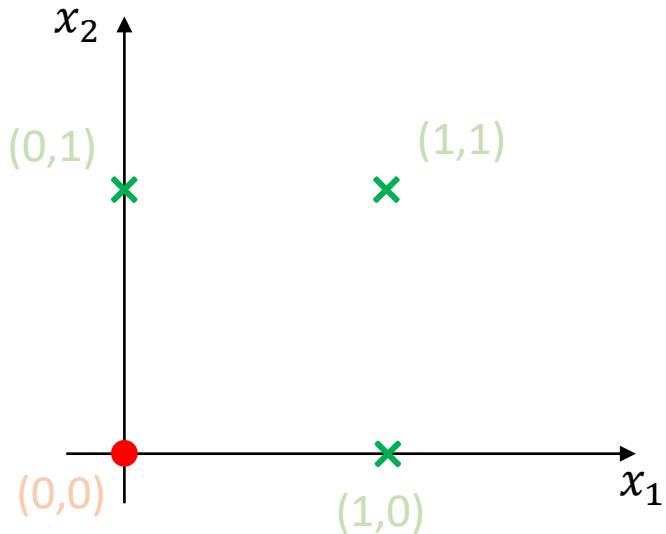
- The error is propagated back to adjust the learning parameters
- Objective: Updating learning parameters to minimize the loss
- Popular optimization technique: Gradient Descent



An example for illustration

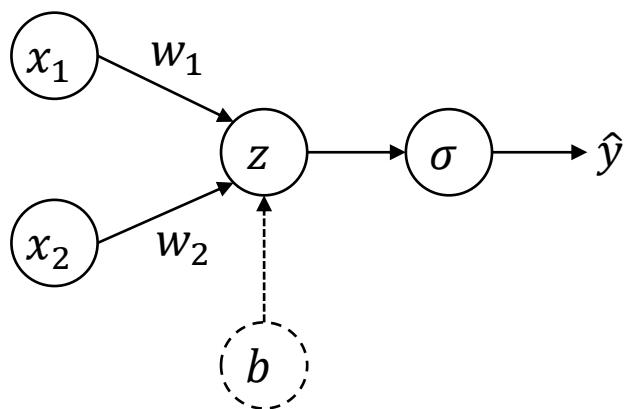
feature		label
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Boolean OR function



Objective: To train a neural network so that it can learn the Boolean OR functionality

Feed Forward



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

Input for first sample: $x_1 = 0, x_2 = 0$

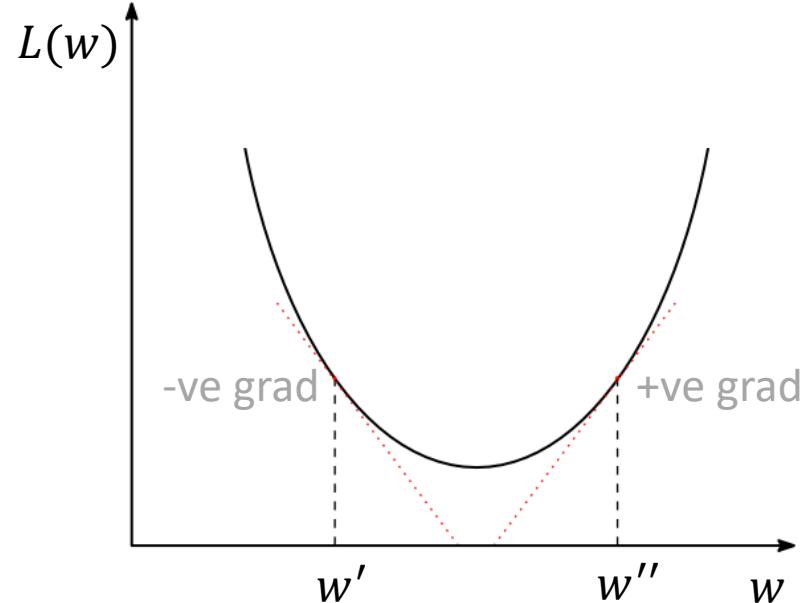
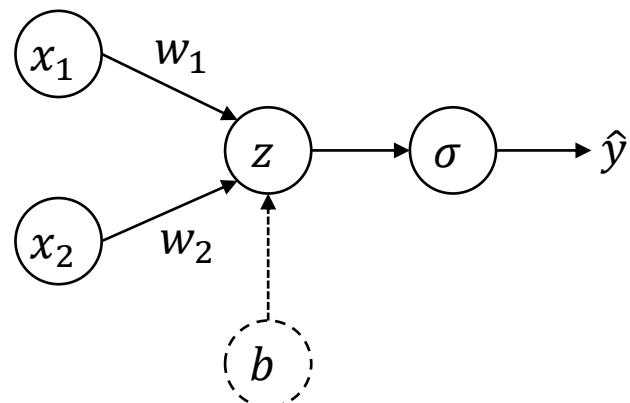
Initialize weight randomly: $w_1 = 0.2, w_2 = 0.5, b = 0$

$$z = w_1 x_1 + w_2 x_2 + b = 0.2 * 0 + 0.5 * 0 + 0 = 0$$

$$\hat{y} = \sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^0} = \frac{1}{2} = 0.5$$

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(0 - 0.5)^2 = 0.125$$

Backpropagation: Intuition



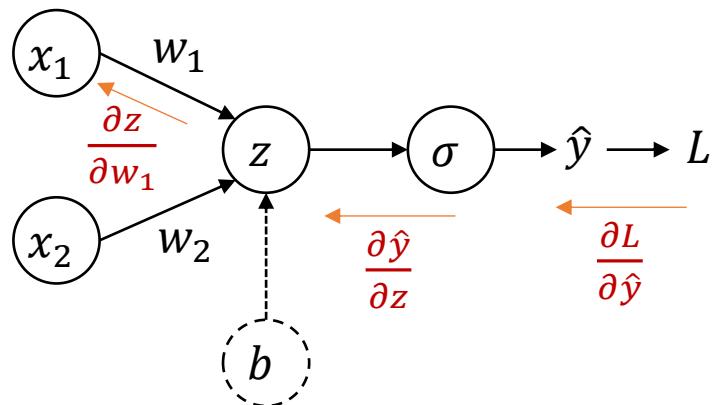
$$L(y, \hat{y})_{\begin{cases} w_1=0.2 \\ w_2=0.5 \\ b=0 \end{cases}} = \frac{1}{2}(y - \hat{y})^2 = 0.125$$

Aim: To obtain values of the learning parameters (i.e., w_1, w_2, b in this example) to minimize $L(y, \hat{y})$
 $\underset{w_1, w_2, b}{\text{minimize}} L(y, \hat{y})$

How do we revise the values of the learning parameters?

- Loss will be propagated in backward direction to revise values of the learning parameters.

Backpropagation



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

$x_1=0, x_2=0$
 $w_1 = 0.2, w_2 = 0.5, b = 0$
 $z = 0$
 $\hat{y} = \sigma(z) = 0.5$
 $L(y, \hat{y}) = 0.125$

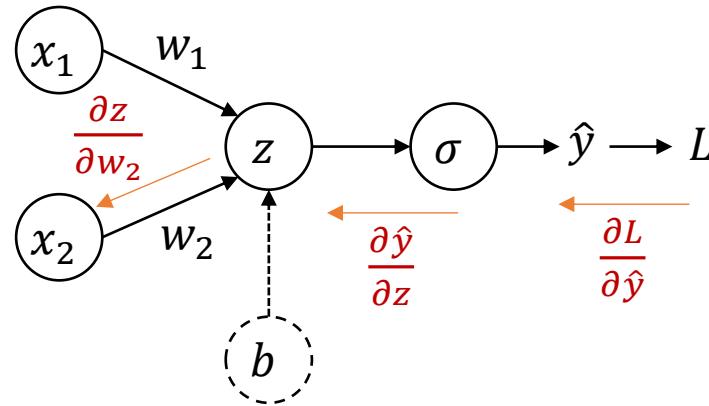
$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_1}$$

$$\frac{\partial L}{\partial \hat{y}} = \frac{\partial [\frac{1}{2}(y - \hat{y})^2]}{\partial \hat{y}} = -(y - \hat{y}) = -(0 - 0.5) = 0.5$$

$$\frac{\partial \hat{y}}{\partial z} = \frac{\partial [\sigma(z)]}{\partial z} = \frac{\partial [\frac{1}{1 + e^{-z}}]}{\partial z} = \frac{1 + e^{-z} - 1}{(1 + e^{-z})^2} = \frac{1}{1 + e^{-z}} - \frac{1}{(1 + e^{-z})^2} = \sigma(z)[1 - \sigma(z)] = 0.5(1 - 0.5) = 0.25$$

$$\frac{\partial z}{\partial w_1} = \frac{\partial [w_1 x_1 + w_2 x_2 + b]}{\partial w_1} = x_1 = 0$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_1} = 0.5 * 0.25 * 0 = 0$$



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

$x_1 = 0, x_2 = 0$
 $w_1 = 0.2, w_2 = 0.5, b = 0$
 $z = 0$
 $\hat{y} = \sigma(z) = 0.5$
 $L(y, \hat{y}) = 0.125$

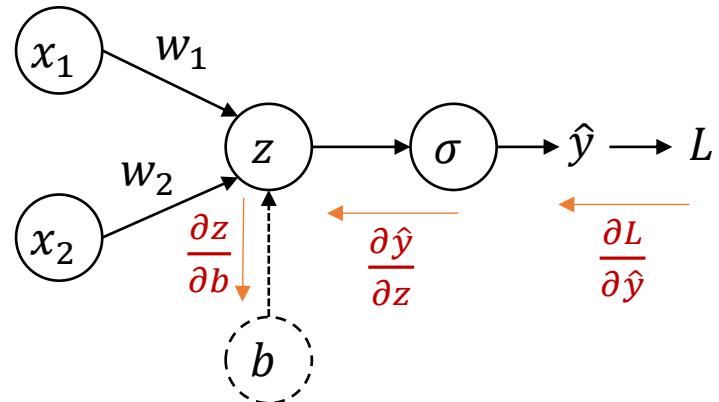
$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_2}$$

$$\frac{\partial L}{\partial \hat{y}} = \frac{\partial [\frac{1}{2}(y - \hat{y})^2]}{\partial \hat{y}} = -(y - \hat{y}) = -(0 - 0.5) = 0.5$$

$$\frac{\partial \hat{y}}{\partial z} = \frac{\partial [\sigma(z)]}{\partial z} = \frac{\partial [\frac{1}{1 + e^{-z}}]}{\partial z} = \sigma(z)[1 - \sigma(z)] = 0.5(1 - 0.5) = 0.25$$

$$\frac{\partial z}{\partial w_2} = \frac{\partial [w_1 x_1 + w_2 x_2 + b]}{\partial w_2} = x_2 = 0$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_2} = 0.5 * 0.25 * 0 = 0$$



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

$x_1 = 0, x_2 = 0$
 $w_1 = 0.2, w_2 = 0.5, b = 0$
 $z = 0$
 $\hat{y} = \sigma(z) = 0.5$
 $L(y, \hat{y}) = 0.125$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial b}$$

$$\frac{\partial L}{\partial \hat{y}} = \frac{\partial [\frac{1}{2}(y - \hat{y})^2]}{\partial \hat{y}} = -(y - \hat{y}) = -(0 - 0.5) = 0.5$$

$$\frac{\partial \hat{y}}{\partial z} = \frac{\partial [\sigma(z)]}{\partial z} = \frac{\partial [\frac{1}{1 + e^{-z}}]}{\partial z} = \sigma(z)[1 - \sigma(z)] = 0.5(1 - 0.5) = 0.25$$

$$\frac{\partial z}{\partial b} = \frac{\partial [w_1 x_1 + w_2 x_2 + b]}{\partial b} = 1$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial b} = 0.5 * 0.25 * 1 = 0.125$$

Gradient Descent (GD)

Start with random values of learning parameters

In each iteration, update the values learning parameters to reduce the value of L

Terminate the algorithm once termination criteria is satisfied

```
Repeat until terminated {  
     $w_i := w_i - \alpha \frac{\partial L}{\partial w_i}; \quad \forall i$   
}
```

α is a hyper-parameter
represents *learning rate*

One epoch of GD:

$$w_1 := w_1 - \alpha \frac{\partial L}{\partial w_1} = 0.2 - 1 * 0 = 0.2 ; \text{ (let, } \alpha=1)$$

$$w_2 := w_2 - \alpha \frac{\partial L}{\partial w_2} = 0.5 - 1 * 0 = 0.5$$

$$b := b - \alpha \frac{\partial L}{\partial b} = 0 - 1 * 0.125 = -0.125$$

Feed-forward in 2nd iteration:

$$z = w_1 x_1 + w_2 x_2 + b = 0.2 * 0 + 0.5 * 0 + (-0.125) = -0.125$$

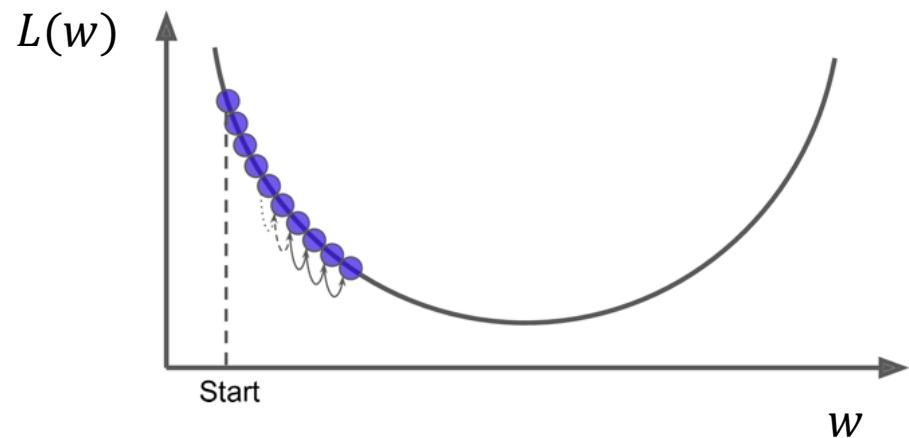
$$\hat{y} = \sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{0.125}} = 0.469$$

$$L(y, \hat{y}) = \frac{1}{2} (y - \hat{y})^2 = \frac{1}{2} (0 - 0.469)^2 = 0.109$$

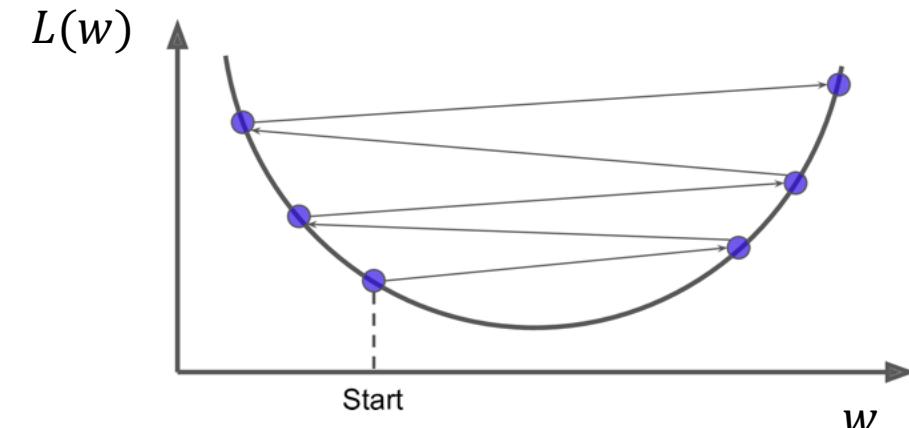
Previously, $L(y, \hat{y}) = 0.125$

Learning Rate (α)

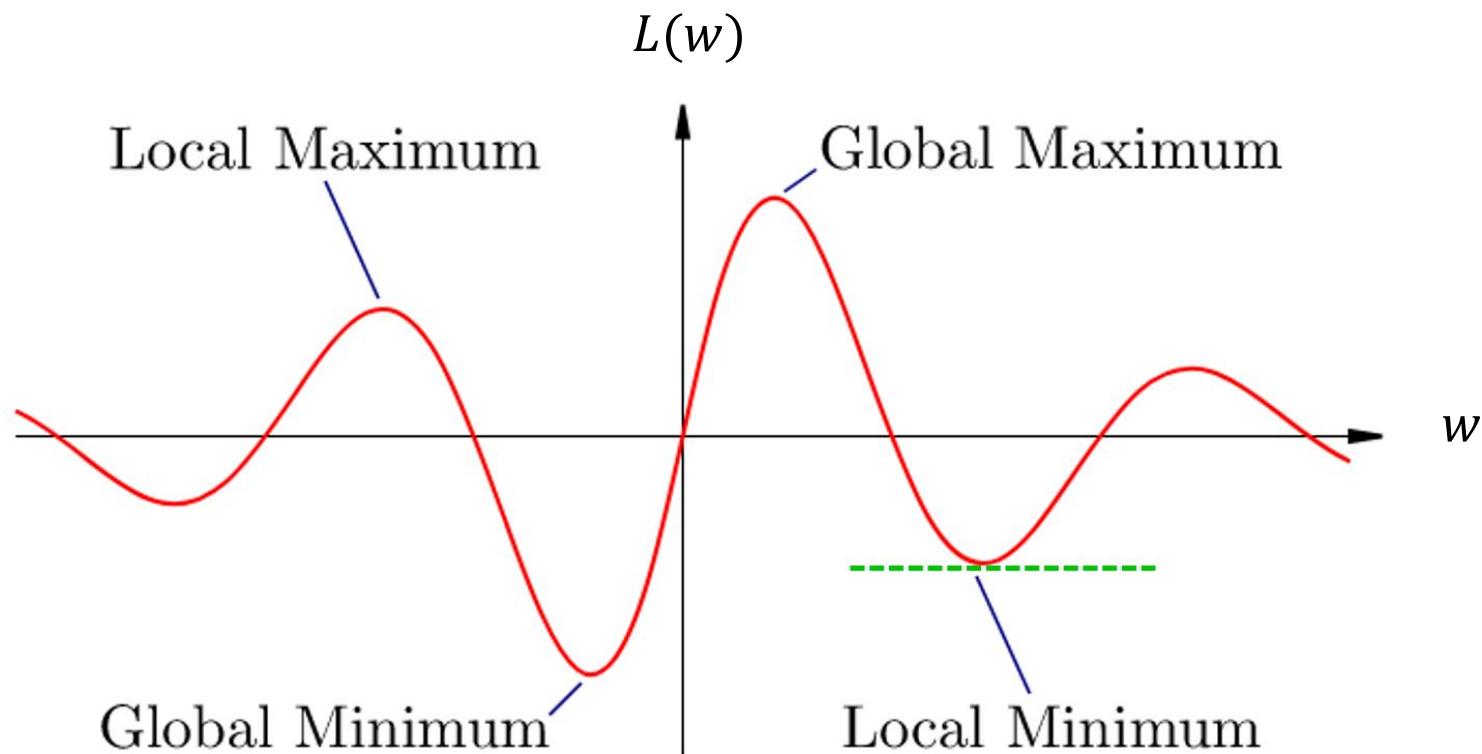
- What happens if the value of α is very small?
 - Learning will be done slowly



- What happens if the value of α is very big?
 - May overshoot the minimum value
 - May fail to converge, even diverge



Problem of GD



Some remaining topics

- Different loss functions
- Cost function
- Activation functions: Pros & Cons
- Optimization functions
- Regularization
- Multilayer
- ...



Thank You!

adak32@gmail.com