Lec07: Artificial Neural Network (Part I)

충북대학교

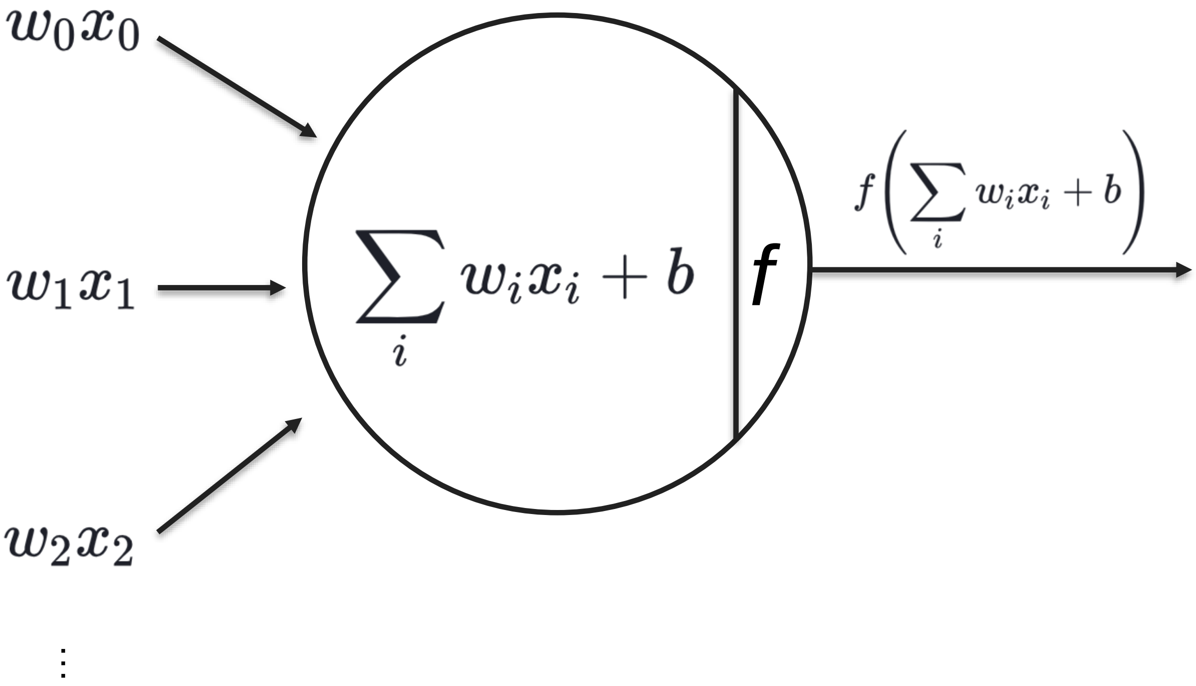
문성태 (지능로봇공학과)

stmoon@cbnu.ac.kr

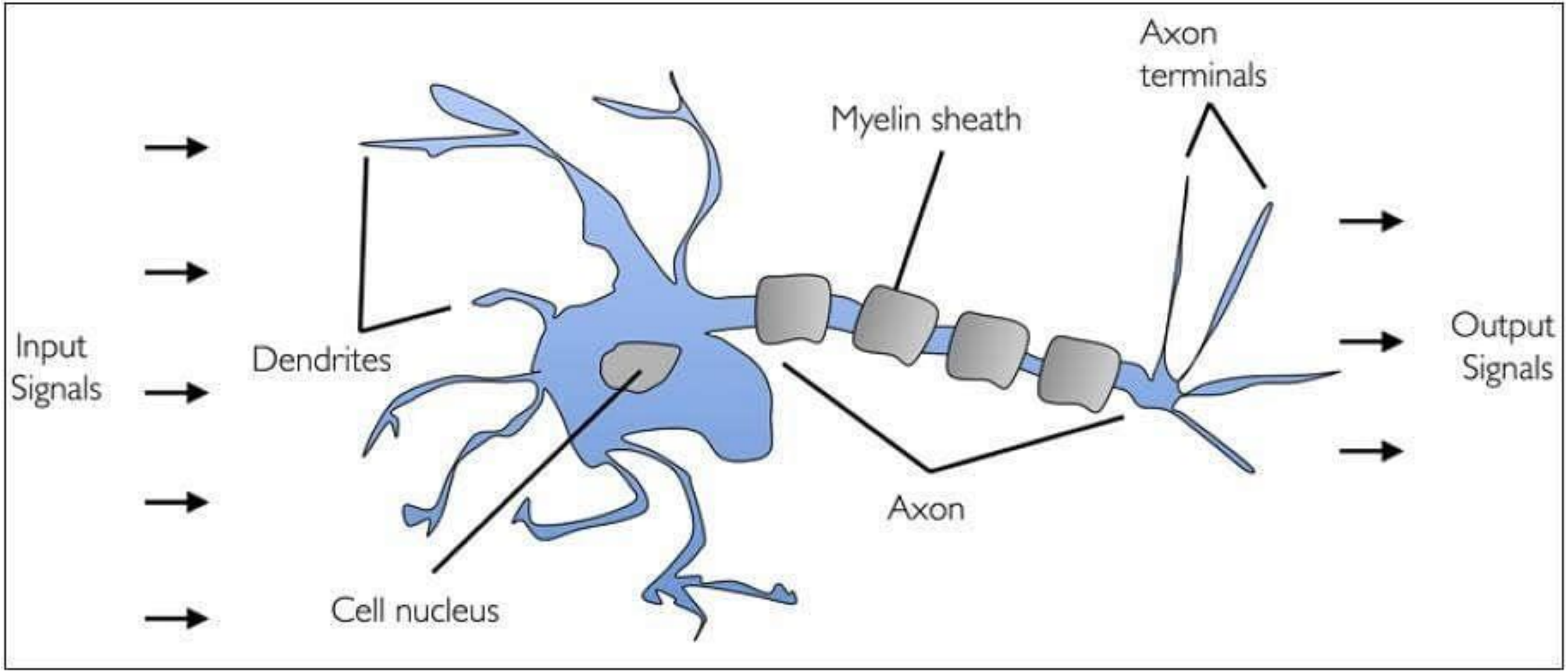
01

Perceptron

Recap: Logistic Regression

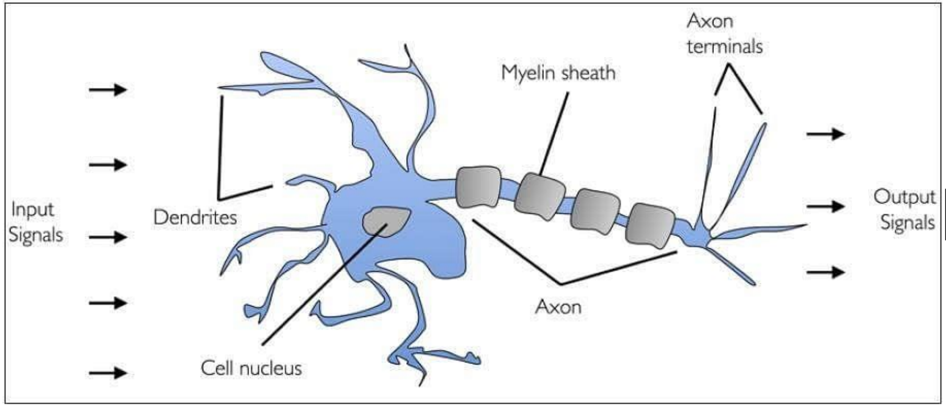


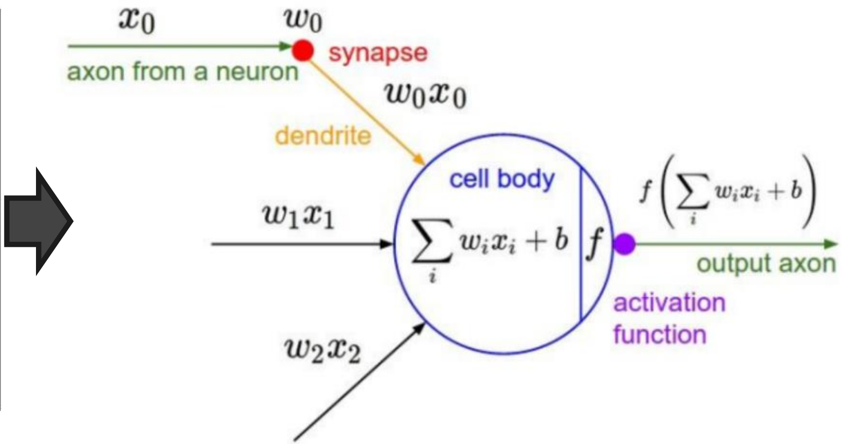
Biological Neuron



Perceptron

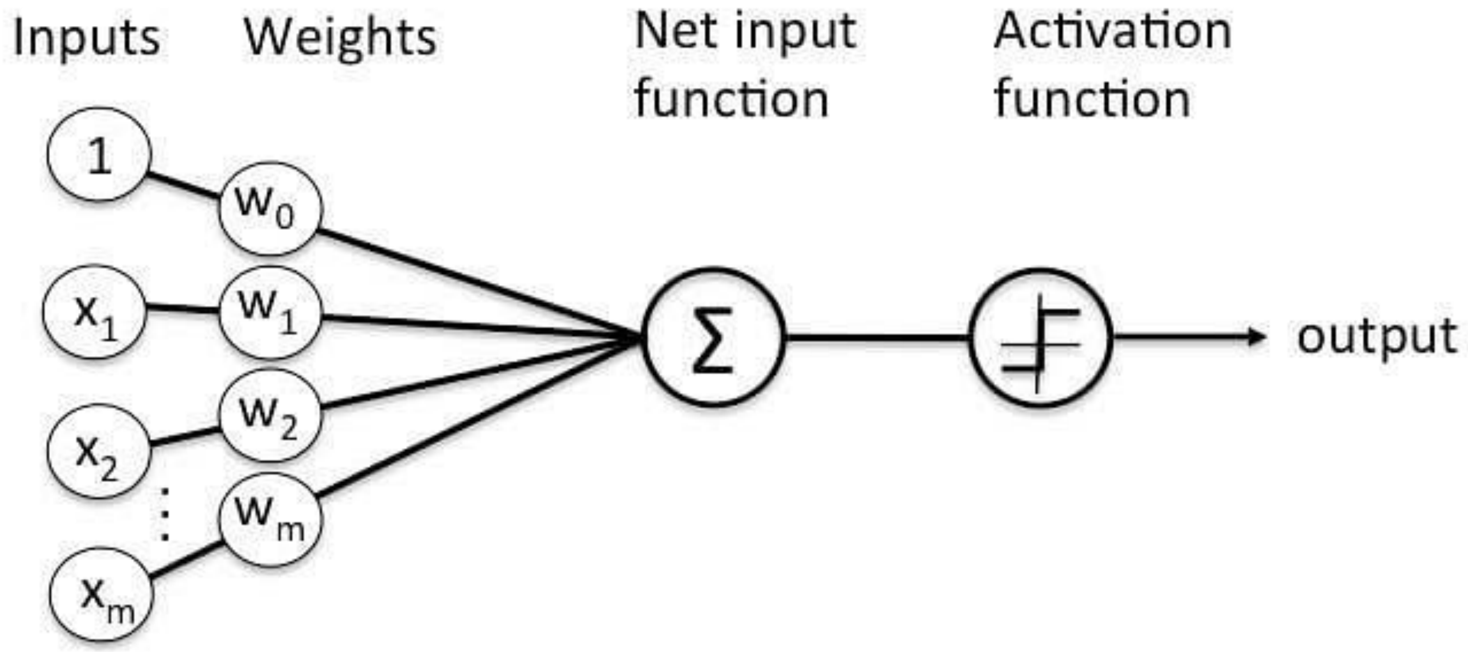
- 인공 신경망(Artificial Neural Network, ANN)의 구성 요소(unit)로서 다수의 값을 입력 받아 하나의 값으로 출력하는 알고리즘





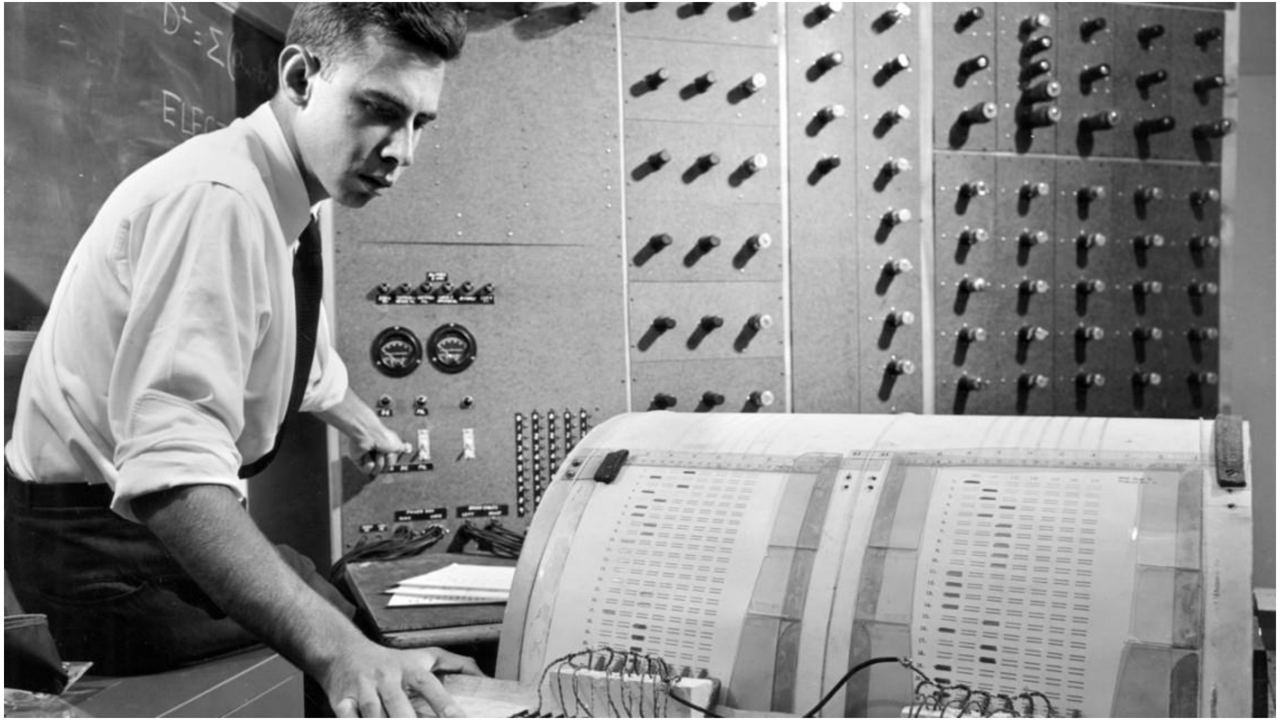
Perceptron

- Perceptron was introduced by Frank Rosenblatt in 1957



Perceptron

- Frank Rosenblatt



Perceptron

Vol. VI, No. 2, Summer 1958

research trends

CORNELL AERONAUTICAL LABORATORY, INC., BUFFALO 21, NEW YORK

The Design of an *Intelligent* AUTOMATON

by FRANK ROSENBLATT

Introducing the perceptron — A machine which senses, recognizes, remembers, and responds like the human mind.

STORIES about the creation of machines having human qualities have long been a fascinating province in the realm of science fiction. Yet we are now about to witness the birth of such a machine — a machine capable of perceiving, recognizing, and identifying its surroundings without any human training or control.

Development of that machine has stemmed from a search for an understanding of the physical mechanisms which underlie human experience and intelligence. The question of the nature of these processes is at least as ancient as any other question in western science and philosophy, and, indeed, ranks as one of the greatest scientific challenges of our time.

Our understanding of this problem has gone perhaps as far as had the development of physics before Newton. We have some excellent descriptions of the phenomena to be explained, a number of interesting hypotheses, and a little detailed knowledge about events in the nervous system. But we lack agreement on any integrated set of principles by which the functioning of the nervous system can be understood.

We believe now that this ancient problem is about to yield to our theoretical investigation for three reasons:

First, in recent years our knowledge of the functioning of individual cells in the central nervous system has vastly increased.

Second, large numbers of engineers and mathematicians are, for the first time, undertaking serious study of the mathematical basis for thinking, perception, and the handling of information by the central nervous system, thus providing the hope that these problems may be within our intellectual grasp.

Third, recent developments in probability theory and in the mathematics of random processes provide new tools for the study of events in the nervous system, where only the gross statistical organization is known and the precise cell-by-cell "wiring diagram" may never be obtained.

Receives Navy Support

In July, 1957, Project PARA (Perceiving and Recognizing Automaton), an internal research program which had been in progress for over a year at Cornell Aeronautical Laboratory, received the support of the Office of Naval Research. The program had been concerned primarily with the application of probability theory to

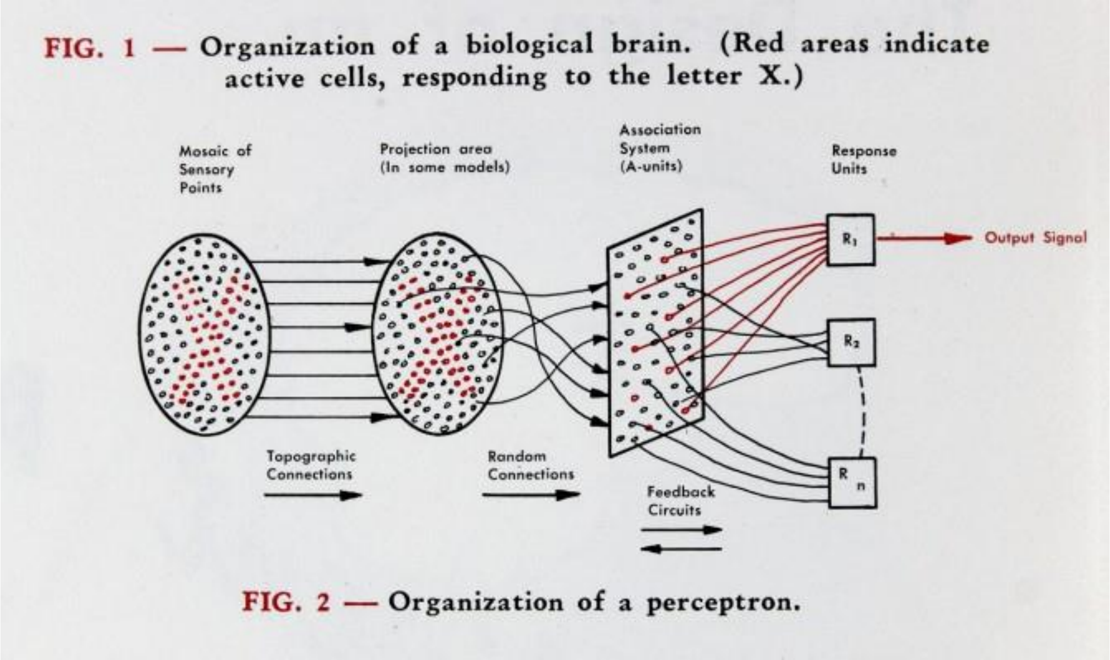
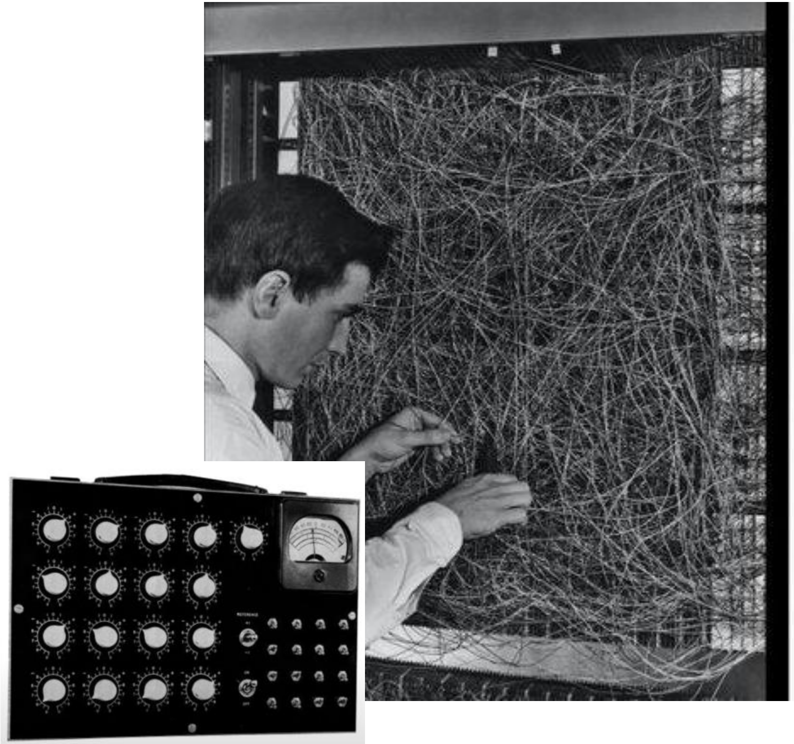


FIG. 1 — Organization of a biological brain. (Red areas indicate active cells, responding to the letter X.)

FIG. 2 — Organization of a perceptron.

Perceptron Hardware

- Frank Rosenblatt with his Mark I perceptron



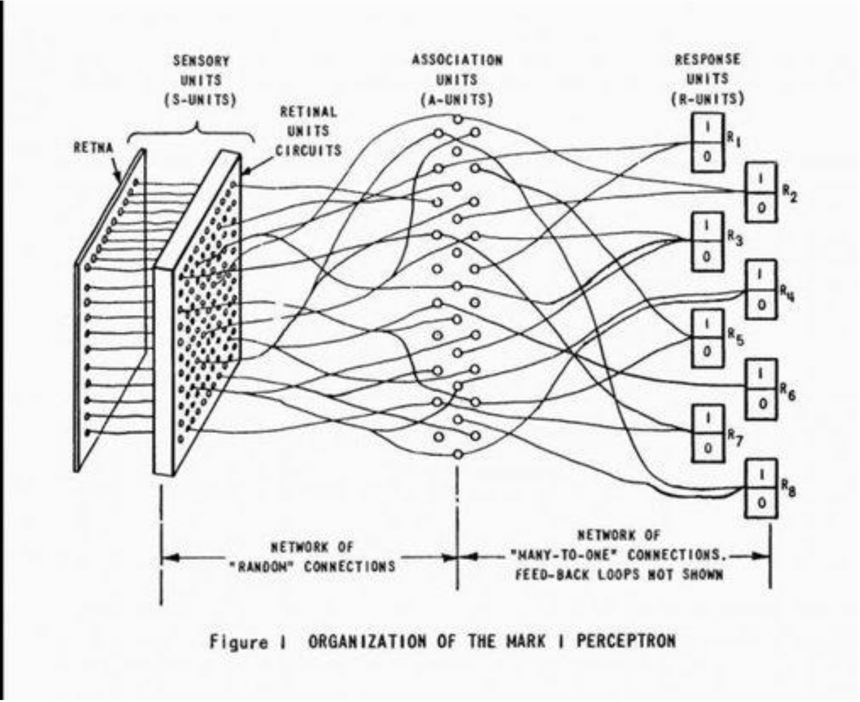


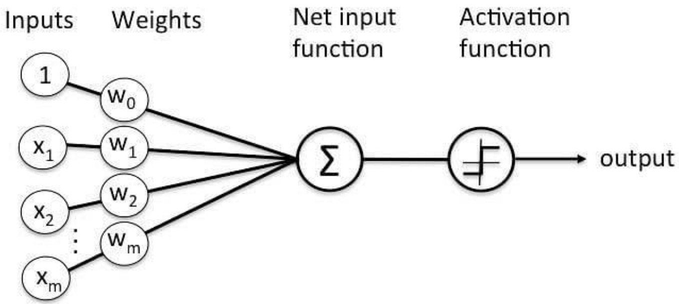
Figure I ORGANIZATION OF THE MARK I PERCEPTRON

False Promises

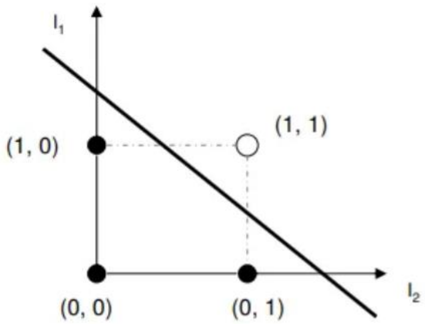
"The Navy revealed the embryo of an electronic computer today that *it expects will be able to walk, talk, see, write, reproduce itself* an be conscious of its existence … Dr. Frank Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers" The New York Times July 08, 1958

AND/OR Problem

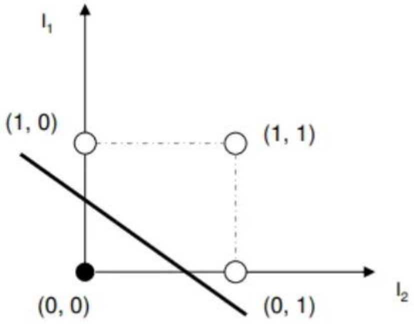
- perceptron can separate its input space with a hyperplane



AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



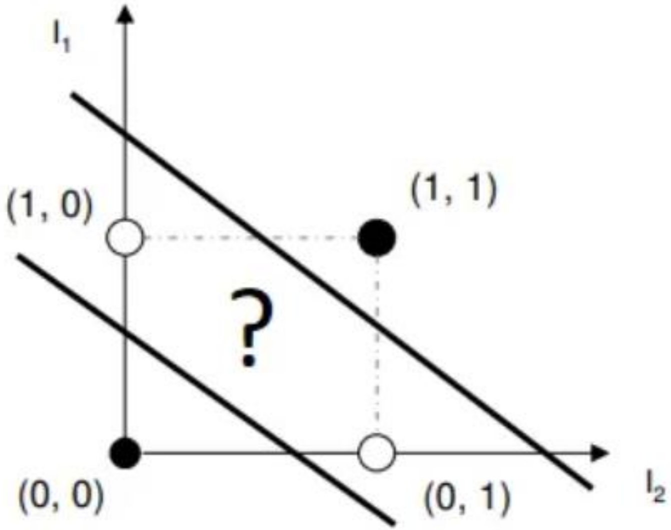
OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1



XOR Problem

- Linearly separable?

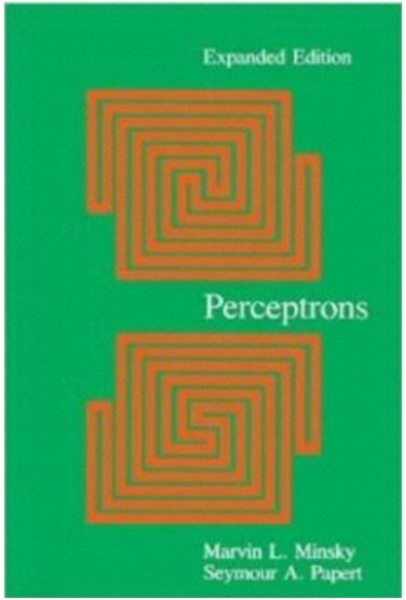
XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0



Perceptrons (1969)

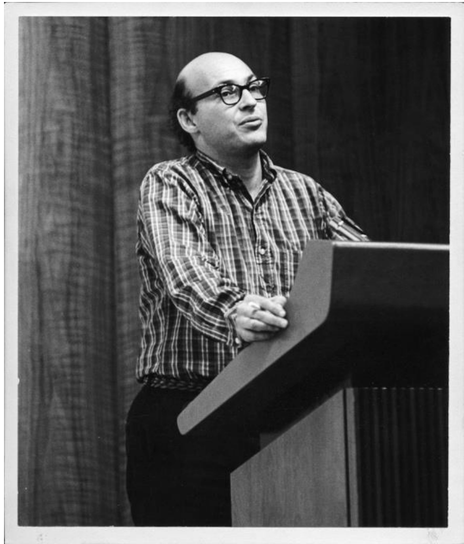
- Perceptrons (1969) by Marvin Minsky, founder of the MIT AI Lab

“No one on earth had found a viable way to train”



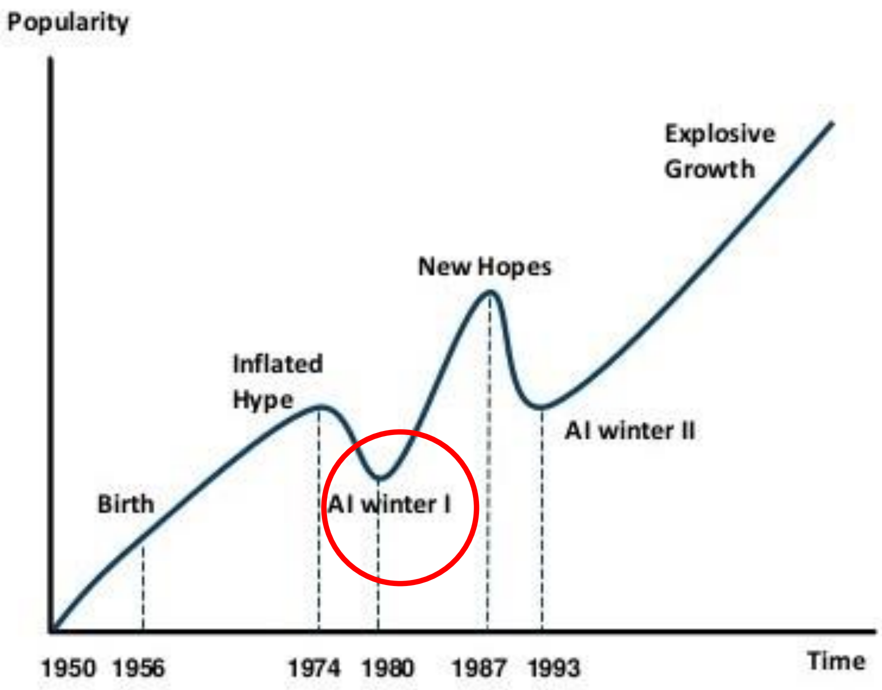
We need to use MLP, multilayer perceptrons (multilayer neural nets)

No one on earth had found a viable way to train MLPs good enough to learn such simple functions



AI Winter I

AI HAS A LONG HISTORY OF BEING "THE NEXT BIG THING"…



Timeline of AI Development

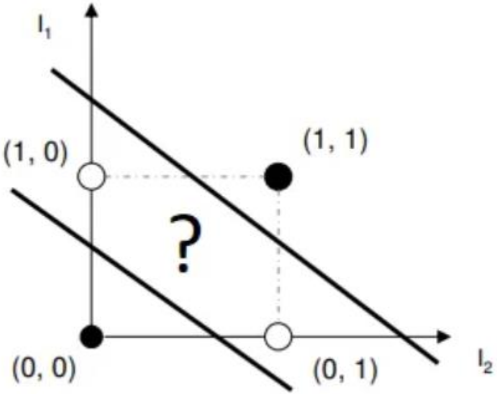
- **1950s-1960s**: First AI boom - the age of reasoning, prototype AI developed
- **1970s**: AI winter I
- **1980s-1990s**: Second AI boom: the age of Knowledge representation (appearance of expert systems capable of reproducing human decision-making)
- **1990s**: AI winter II
- **1997**: Deep Blue beats Gary Kasparov
- **2006**: University of Toronto develops Deep Learning
- **2011**: IBM's Watson won Jeopardy
- **2016**: Go software based on Deep Learning beats world's champions

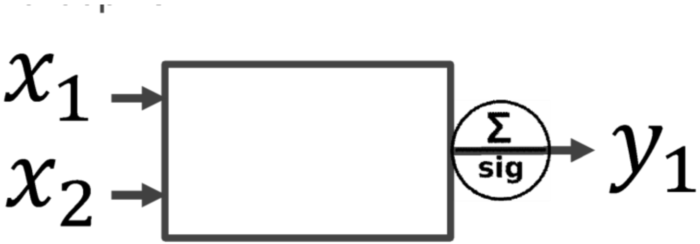
02

Multi-Layer Perceptron

XOR Problem

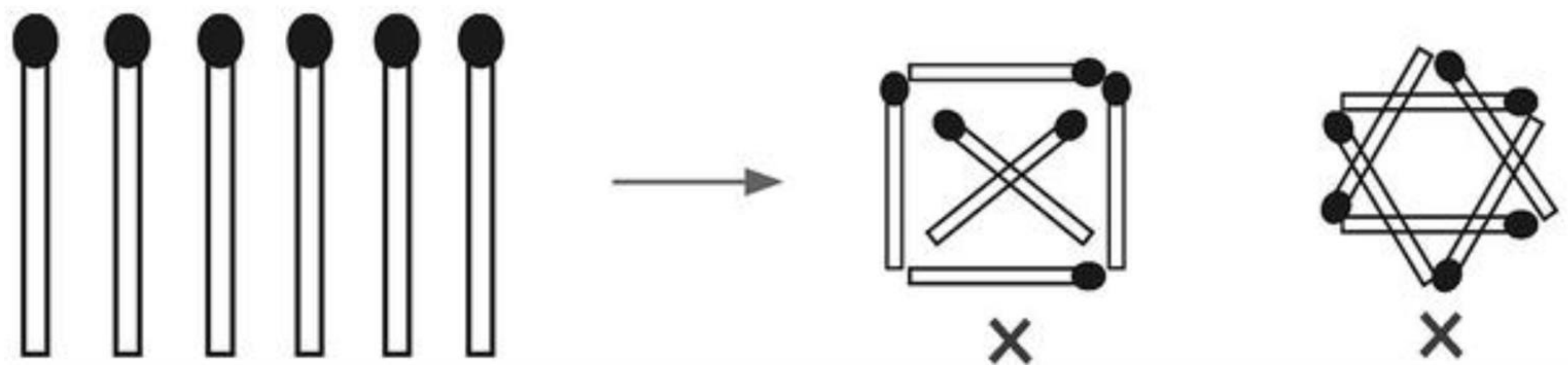
XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0



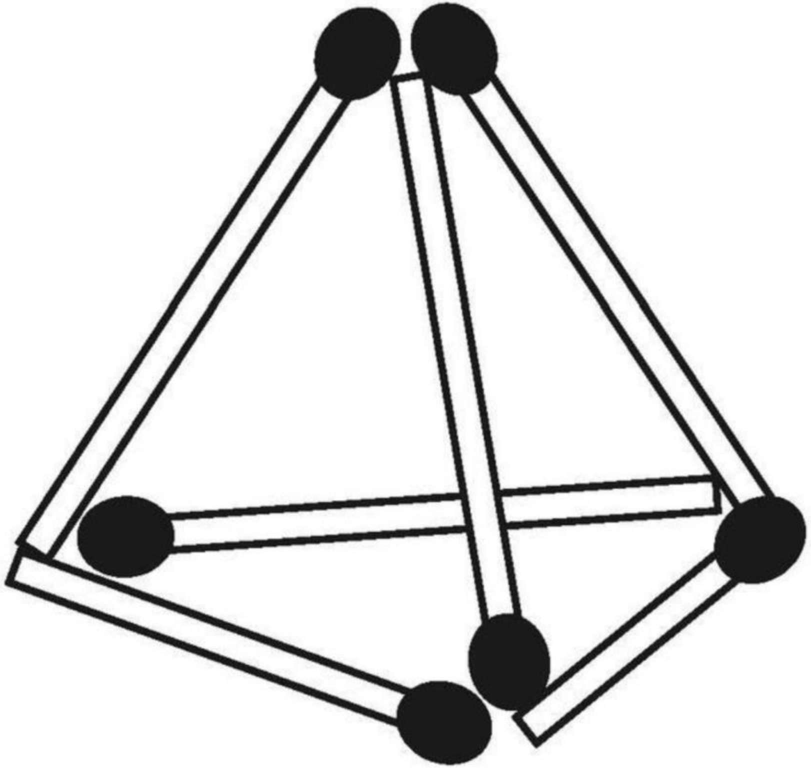


Quiz

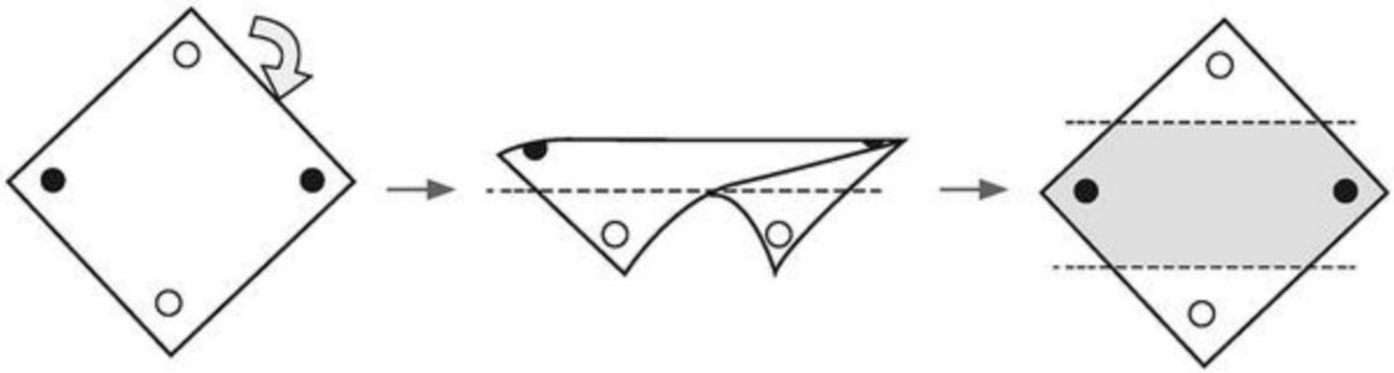
- 성냥개비 6개로 정삼각형 4개만 만드세요



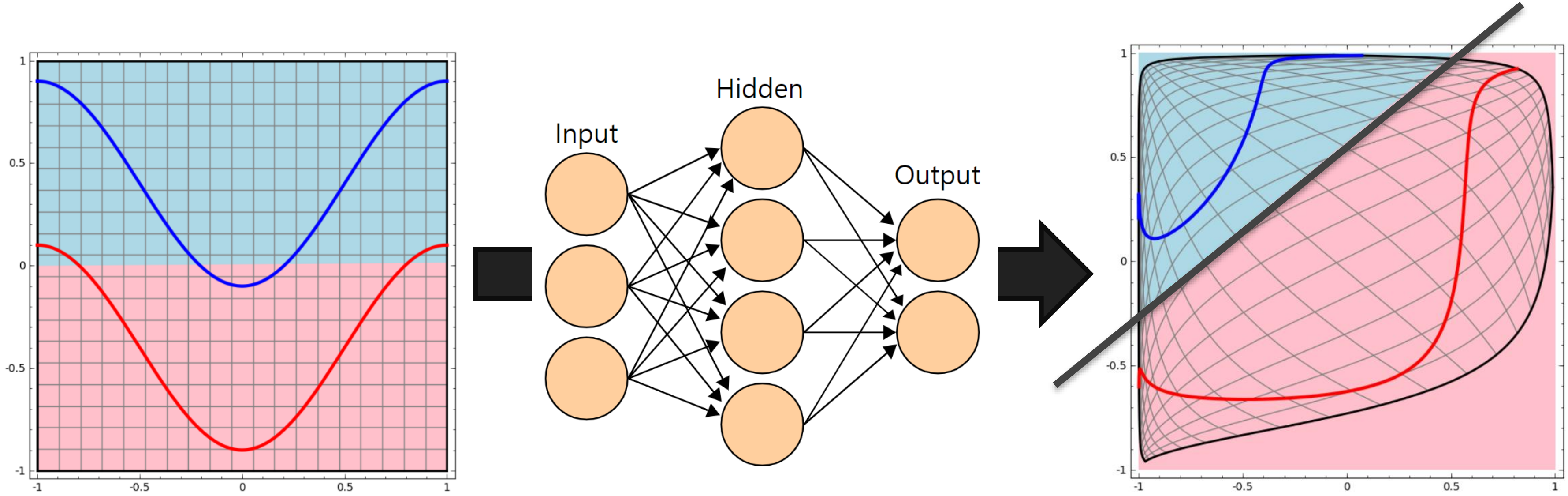
Quiz Solution



XOR Solution

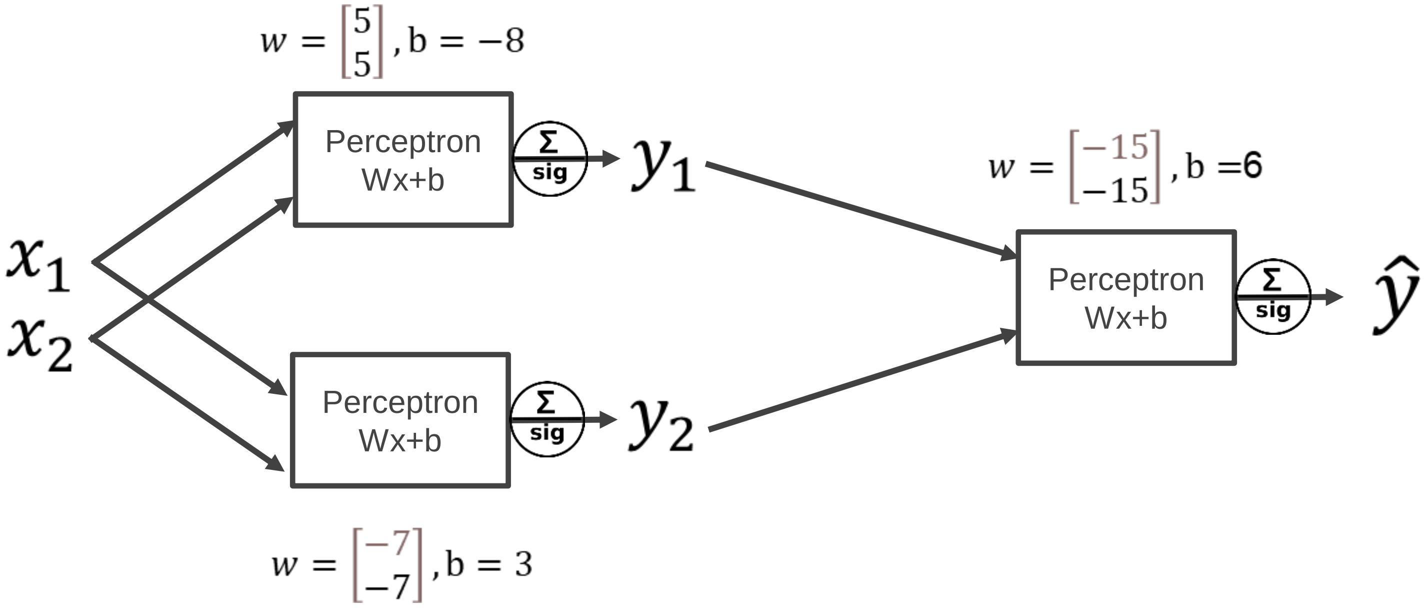


XOR Problem Solution



https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

XOR Problem Solution



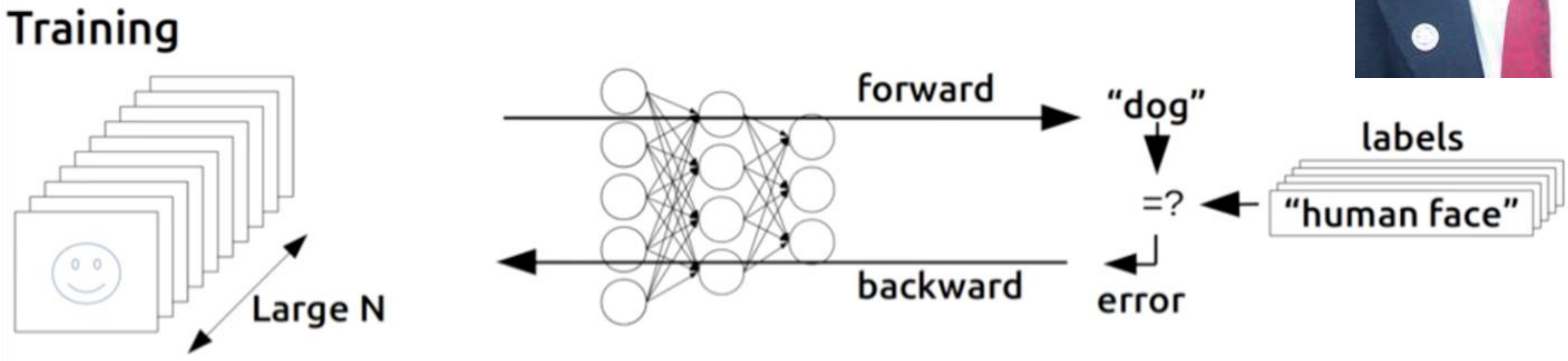
How can we learn W and b from training data?

03

Backpropagation

Backpropagation

- 1974, 1982 by Paul Werbos, 1986 by Hinton
  - Paul Werbos, based on his 1974 Ph.D. thesis, publicly proposes the use of Backpropagation for propagating errors during the training of Neural Networks



Before learning backpropagation…

Basic derivative

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- $f(x) = 3$ $\qquad \emptyset$
- $f(x) = 2x$ $\qquad \frac{d + (x)}{dx} = 2$
- $f(x) = x + 3$ $\qquad = 1$

Basic derivative (Chain Rule)

$$F(x) = f(g(x))$$
$$F'(x) = f'(g(x)) \cdot g'(x)$$

핵심

Basic derivative (Sigmoid)

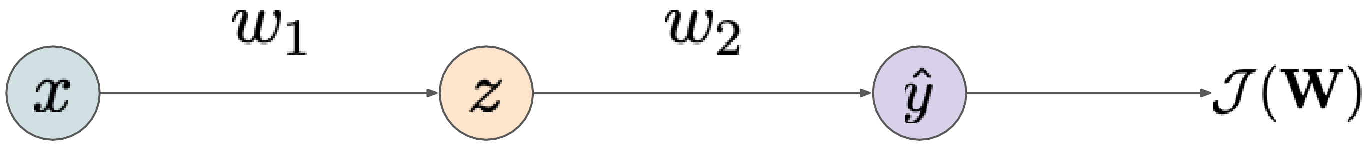
$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Computing gradients of weights in neural network

- Simplest example: two-layer neural network with one hidden node

$$\hat{y} = f(x; \mathbf{W})$$

$$\boxed{\{w_1, w_2\}}$$



Computing gradients of weights in neural network

- Simplest example: two-layer neural network with one hidden node

$$\hat{y} = f(x; \mathbf{W})$$

$$x \xrightarrow{w_1} z \xrightarrow{w_2} \hat{y} \longrightarrow \mathcal{J}(\mathbf{W})$$

$$\frac{\partial \mathcal{J}(\mathbf{W})}{\partial w_1} =$$

?

Computing gradients of weights in neural network

- Simplest example: two-layer neural network with one hidden node

$$\hat{y} = f(x; \mathbf{W})$$

$$x \xrightarrow{w_1} z \xrightarrow{w_2} \hat{y} \longrightarrow \mathcal{J}(\mathbf{W})$$

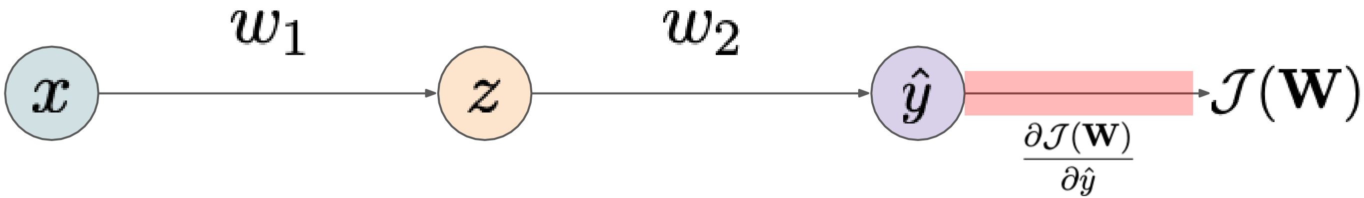
Chain rule: propagating the gradient across the layers

$$\frac{\partial \mathcal{J}(\mathbf{W})}{\partial w_1} = \frac{\partial \mathcal{J}(\mathbf{W})}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_1}$$

Computing gradients of weights in neural network

- Simplest example: two-layer neural network with one hidden node

$$\hat{y} = f(x; \mathbf{W})$$

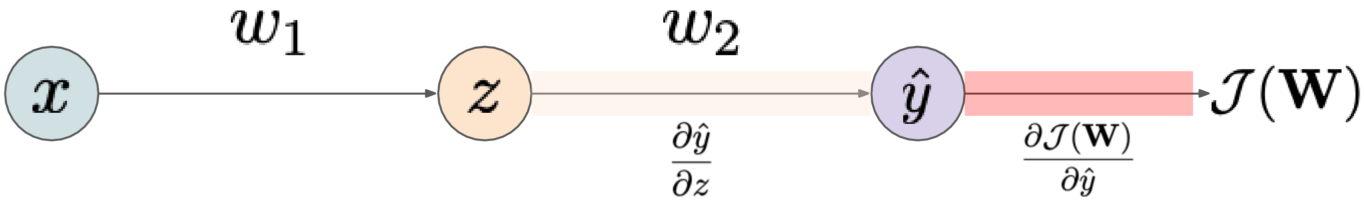


Chain rule: propagating the gradient across the layers $\frac{\partial \mathcal{J}(\mathbf{W})}{\partial w_1} = \frac{\partial \mathcal{J}(\mathbf{W})}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_1}$

Computing gradients of weights in neural network

- Simplest example: two-layer neural network with one hidden node

$$\hat{y} = f(x; \mathbf{W})$$



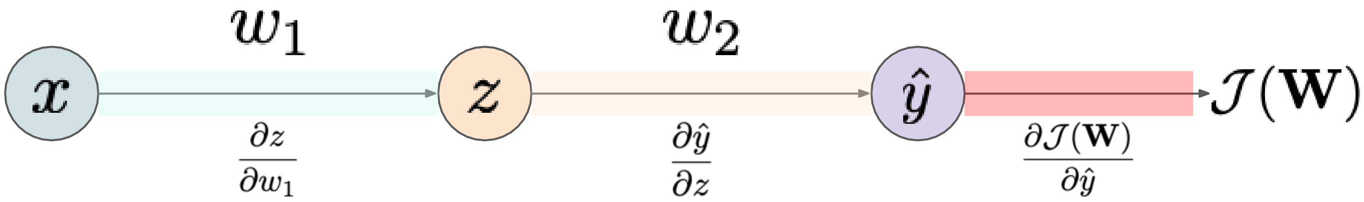
Chain rule: propagating the gradient across the layers

$$\frac{\partial \mathcal{J}(\mathbf{W})}{\partial w_1} = \frac{\partial \mathcal{J}(\mathbf{W})}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_1}$$

Computing gradients of weights in neural network

- Simplest example: two-layer neural network with one hidden node

$$\hat{y} = f(x; \mathbf{W})$$



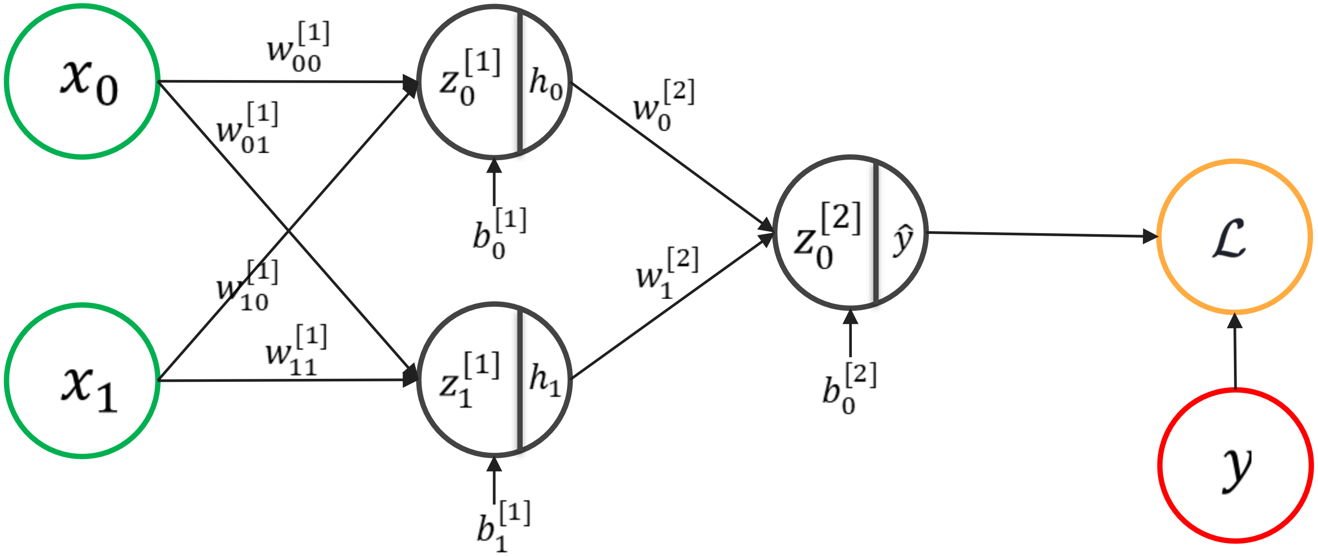
Chain rule: propagating the gradient across the layers

$$\frac{\partial \mathcal{J}(\mathbf{W})}{\partial w_1} = \frac{\partial \mathcal{J}(\mathbf{W})}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_1}$$

03

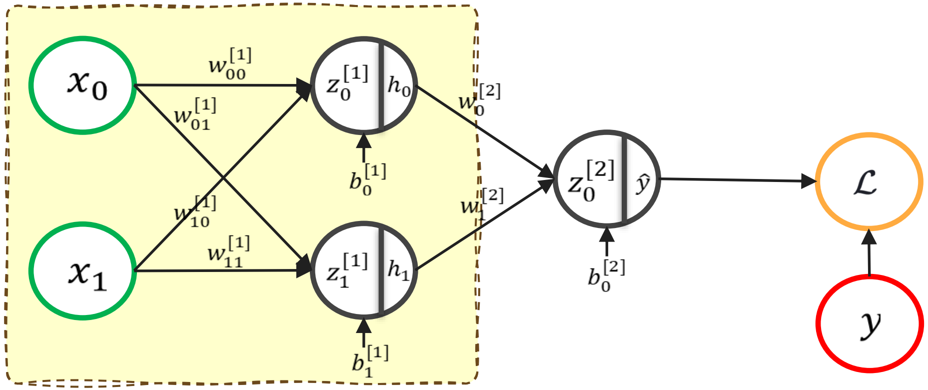
Backpropagation for XOR

XOR neural network



$$W^{[1]} = \begin{bmatrix} w_{00}^{[1]} & w_{01}^{[1]} \\ w_{10}^{[1]} & w_{11}^{[1]} \end{bmatrix} \quad B^{[1]} = \begin{bmatrix} b_0^{[1]} \\ b_1^{[1]} \end{bmatrix} \quad W^{[2]} = \begin{bmatrix} w_{00}^{[2]} \\ w_{10}^{[2]} \end{bmatrix} \quad B^{[2]} = \begin{bmatrix} b_0^{[2]} \end{bmatrix}$$

Forward propagation



$$z_0^{[1]} = w_{00}^{[1]} x_o + w_{10}^{[1]} x_1 + b_o^{[1])}$$
$$z_1^{[1]} = w_{01}^{[1]} x_o + w_{11}^{[1]} x_1 + b_1^{[1]}$$

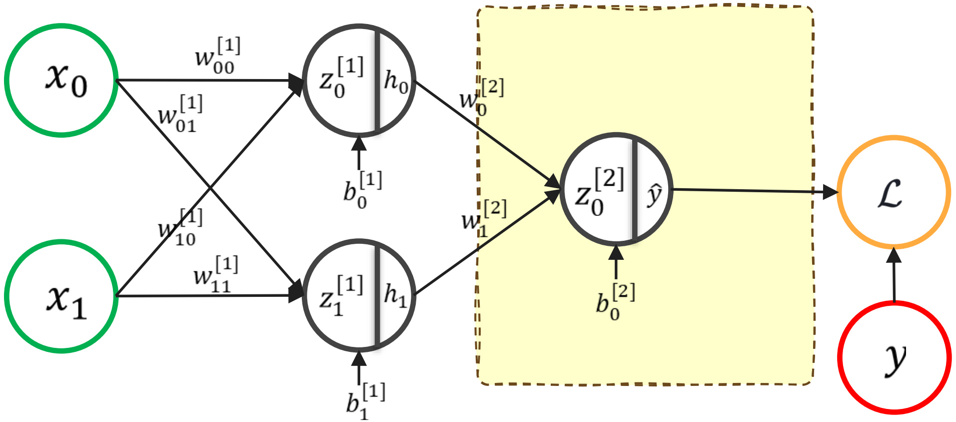
➡ $Z^{[1]} = W^{[1]^T} X + B^{[1]}$

$$W^{(1)} = \begin{bmatrix} w_{00}^{(1)} & w_{01}^{(1)} \\ w_{10}^{(1)} & w_{11}^{(1)} \end{bmatrix} \quad B^{(1)} = \begin{bmatrix} b_0^{(1)} \\ b_1^{(1)} \end{bmatrix}$$

$$h_0 = \sigma(z_0^{(1)})$$
$$h_1 = \sigma(z_1^{(1)})$$

➡ $H^{(1)} = \sigma(Z^{[1]})$

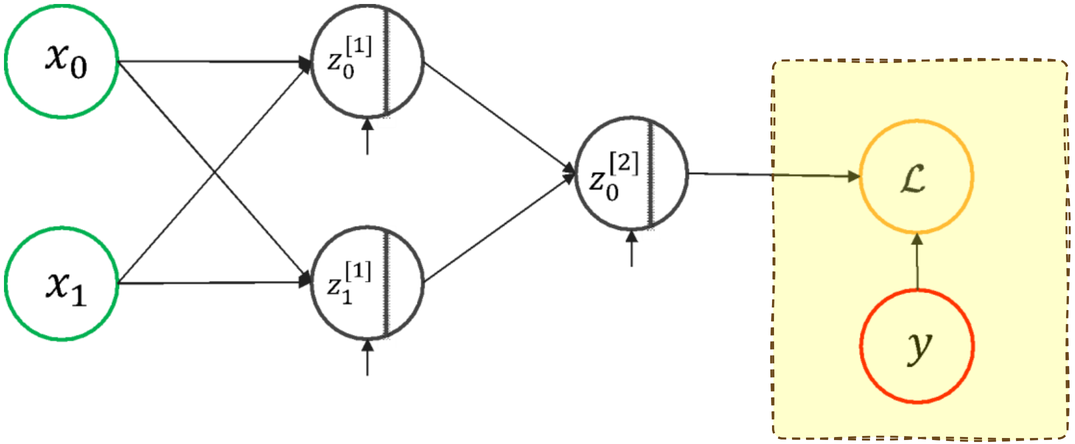
Forward propagation



$$z_0^{[2]} = w_0^{[2]} h_o + w_1^{[2]} h_1 + b_o^{[2]} \Rightarrow z^{[2]} = W^{[2]^T} H + b_0^{[2]} \qquad W^{(2)} = \begin{bmatrix} w_{00}^{(2)} \\ w_{10}^{(2)} \end{bmatrix}$$

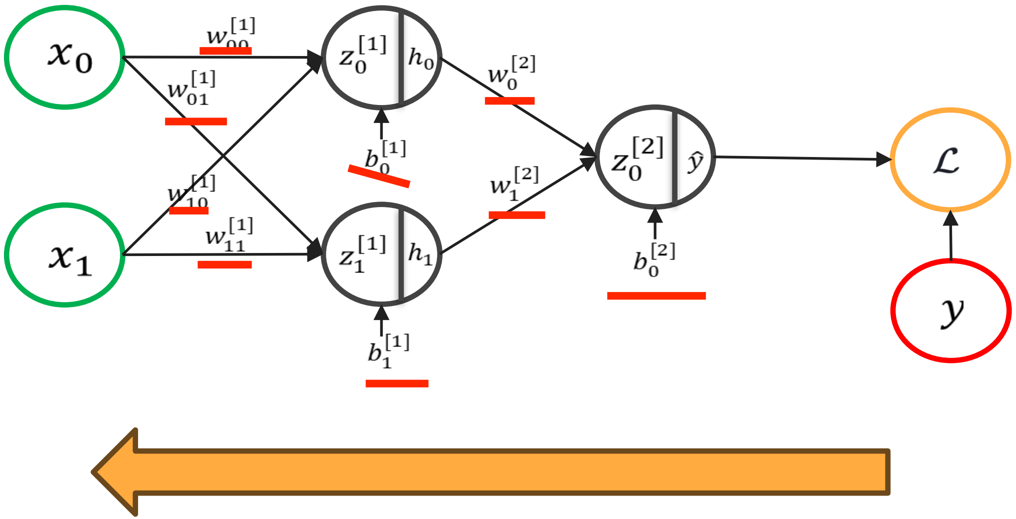
$$\hat{y} = \sigma(z^{[2]})$$

Forward propagation



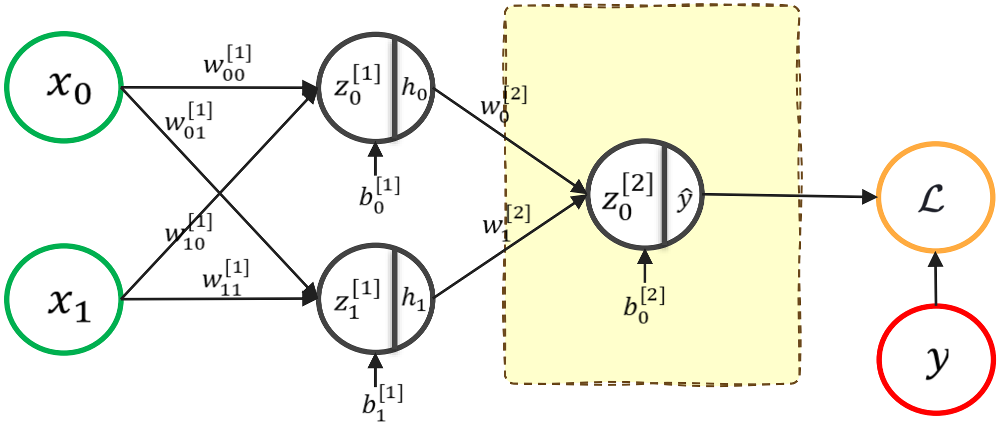
$$\mathcal{L} = -\frac{1}{m}\sum_i^m\{y_i \log \hat{y_i} + (1 - y_i)\log(1 - \hat{y_i})\}$$

Backward Propagation



$$\frac{\partial \mathcal{L}}{\partial W^{(1)}}, \frac{\partial \mathcal{L}}{\partial B^{(1)}}, \frac{\partial \mathcal{L}}{\partial W^{(2)}}, \frac{\partial \mathcal{L}}{\partial B^{(2)}}$$

Backward Propagation



$$\frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial w^{[2]}_{00}}} = \frac{\partial \mathcal{L}}{\partial z^{[2]}_0}\frac{\partial z^{[2]}_0}{\partial w^{[2]}_{00}} = \frac{\partial \mathcal{L}}{\partial z^{[2]}_0} h_0$$

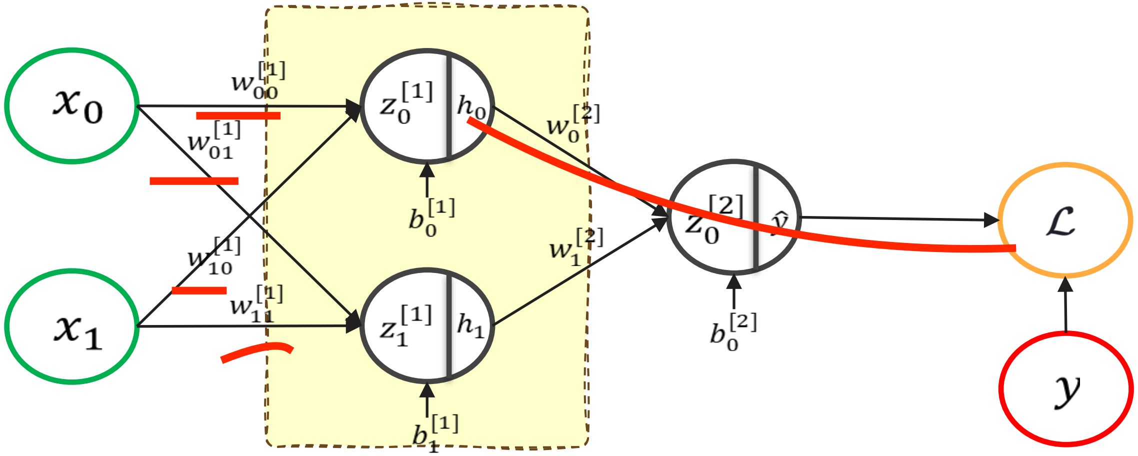
$$\frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial w^{[2]}_{10}}} = \frac{\partial \mathcal{L}}{\partial z^{[2]}_0}\frac{\partial z^{[2]}_0}{\partial w^{[2]}_{10}} = \frac{\partial \mathcal{L}}{\partial z^{[2]}_0} h_1$$

$$\frac{\boldsymbol{\partial \mathcal{L}}}{\partial b^{[2]}} = \frac{\partial \mathcal{L}}{\partial z^{[2]}_0}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial z^{[2]}_0} &= \frac{\partial \mathcal{L}}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial z^{[2]}_0} \\ &= \frac{\partial \mathcal{L}}{\partial \hat{y}}\hat{y}(1-\hat{y}) \\ &= \left(-\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}\right)\hat{y}(1-\hat{y}) \\ &= \hat{y} - y\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{y}} &= \frac{1}{\partial \hat{y}}(-y\log\hat{y} + (1-y)\log(1-\hat{y})) \\ &= -\frac{\mathrm{y}}{\hat{y}} + \frac{1-y}{1-\hat{y}}\end{aligned}$$

Backward Propagation



$$\frac{\partial \mathcal{L}}{\partial W^{[1]}} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial w_{00}^{[1]}} & \frac{\partial \mathcal{L}}{\partial w_{01}^{[1]}} \\ \frac{\partial \mathcal{L}}{\partial w_{10}^{[1]}} & \frac{\partial \mathcal{L}}{\partial w_{11}^{[1]}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial z_0^{[1]}} x_0 & \frac{\partial \mathcal{L}}{\partial z_1^{[1]}} x_0 \\ \frac{\partial \mathcal{L}}{\partial z_0^{[1]}} x_1 & \frac{\partial \mathcal{L}}{\partial z_1^{[1]}} x_1 \end{bmatrix}$$

$$\frac{\partial \mathcal{L}}{\partial B^{[1]}} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial b_0^{[1]}} \\ \frac{\partial \mathcal{L}}{\partial b_1^{[1]}} \end{bmatrix}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}_0^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_0^{[1]}} \frac{\partial z_0^{[1]}}{\partial b_0^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_0^{[1]}}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}_1^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_1^{[1]}} \frac{\partial z_1^{[1]}}{\partial b_1^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_1^{[1]}}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_{00}^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_0^{[1]}} \frac{\partial z_0^{[1]}}{\partial w_{00}^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_0^{[1]}} x_0$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_{01}^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_0^{[1]}} \frac{\partial z_0^{[1]}}{\partial w_{01}^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_1^{[1]}} x_0$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_{10}^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_0^{[1]}} \frac{\partial z_0^{[1]}}{\partial w_{10}^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_0^{[1]}} x_1$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_{11}^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_1^{[1]}} \frac{\partial z_1^{[1]}}{\partial w_{11}^{[1]}} = \frac{\partial \mathcal{L}}{\partial z_1^{[1]}} x_1$$

$$\frac{\partial \mathcal{L}}{\partial z_0^{[1]}} = \frac{\partial \mathcal{L}}{\partial h_0} \frac{\partial h_0}{\partial z_0^{[1]}} = \frac{\partial \mathcal{L}}{\partial h_0} h_0(1 - h_0)$$

$$\frac{\partial \mathcal{L}}{\partial z_1^{[1]}} = \frac{\partial \mathcal{L}}{\partial h_1} \frac{\partial h_1}{\partial z_1^{[1]}} = \frac{\partial \mathcal{L}}{\partial h_1} h_1(1 - h_1)$$

$$\frac{\partial \mathcal{L}}{\partial h_0} = \frac{\partial \mathcal{L}}{\partial z_0^{[2]}} \frac{\partial z_0^{[2]}}{\partial h_0} = \frac{\partial \mathcal{L}}{\partial z_0^{[2]}} w_{00}^{[2]}$$

$$\frac{\partial \mathcal{L}}{\partial h_1} = \frac{\partial \mathcal{L}}{\partial z_0^{[2]}} \frac{\partial z_0^{[2]}}{\partial h_1} = \frac{\partial \mathcal{L}}{\partial z_0^{[2]}} w_{10}^{[2]}$$