Business Analytics (IM561)

Module2. Underlying Mechanisms of Deep Learning

Intelligent Video Analytics with Deep Learning

Jongho Kim

December & Company Inc.

Spring, 2019

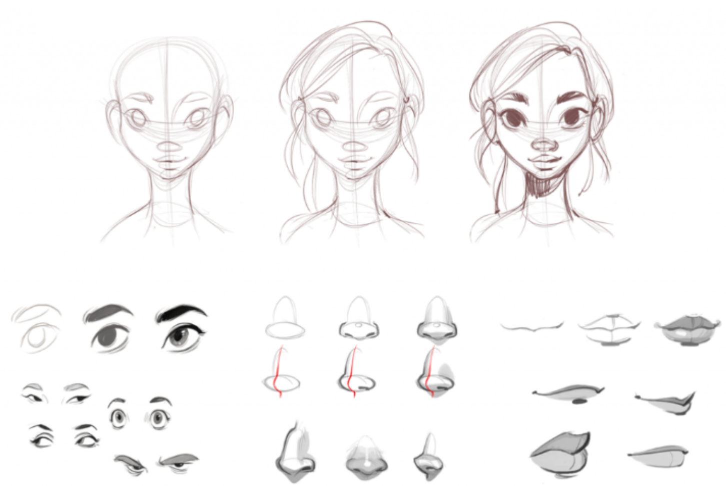
Note. This content mainly refers the summer session of KAIST organized by Jiyong Park(2018)

Module2. Underlying Mechanisms of Deep Learning

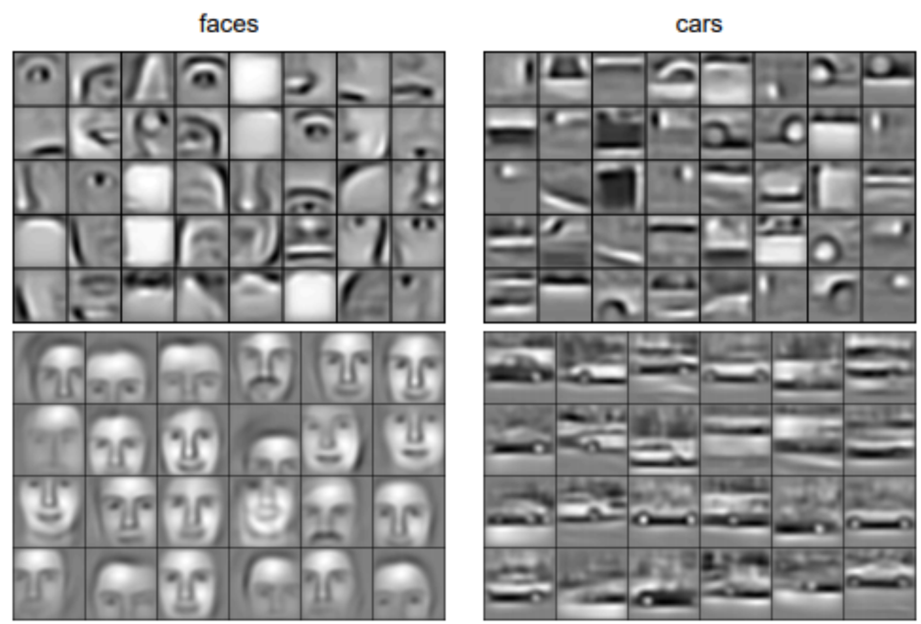
Intuitive Understanding of Deep Learning

- What is deep learning?
  - Deep learning is “**Representation Learning**” to learn and discover how to represent features of data. (Here, representation means a machine-understandable format)

How to draw faces?



How to represent faces?



Intuitive Understanding of Deep Learning

- Why important to well-represent the features of data?

- What needs to well-represent the features of data?

- Why is deep learning outstanding at representation learning?

Intuitive Understanding of Deep Learning

- Why important to well-represent the features of data? → Ch1. Representational Learning
- What needs to well-represent the features of data? → Ch2. Artificial Neural Networks
- Why is deep learning outstanding at representation learning? → Ch3. Deep Learning Algorithms

Ch1. Representational Learning

What is the meaning of Well-Representation of the Data Feature?

- First, let's look at an example of what a Representation is:

- How can you distinguish the objects?

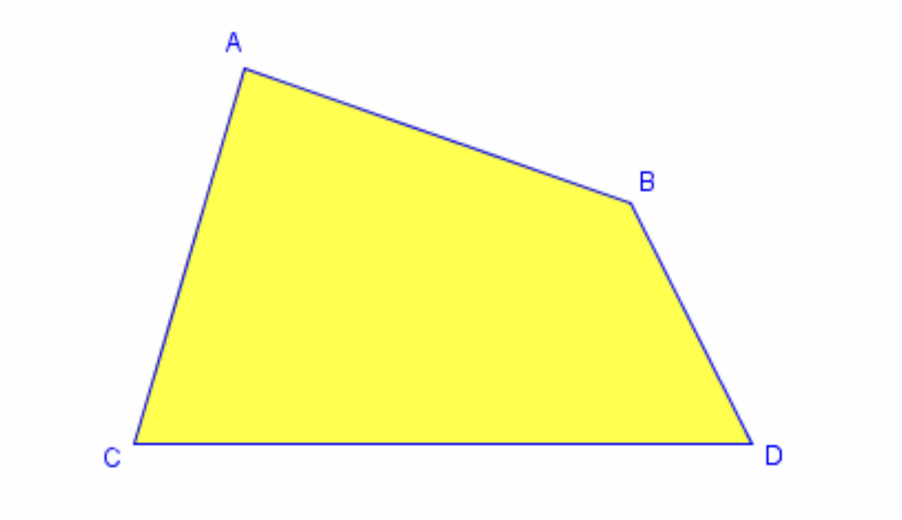
What is the meaning of Well-Representation of the Data Feature?

- First, let's look at an example of what a Representation is:

- If you try to solve the problem of Classify the following shape, the number of corners can be used as a Representation
  - Square: 4 corner
  - Triangle: 3 corner
  - Circle: 0 corner

What is the meaning of Well-Representation of the Data Feature?

- But how do we cope if this shape comes in?



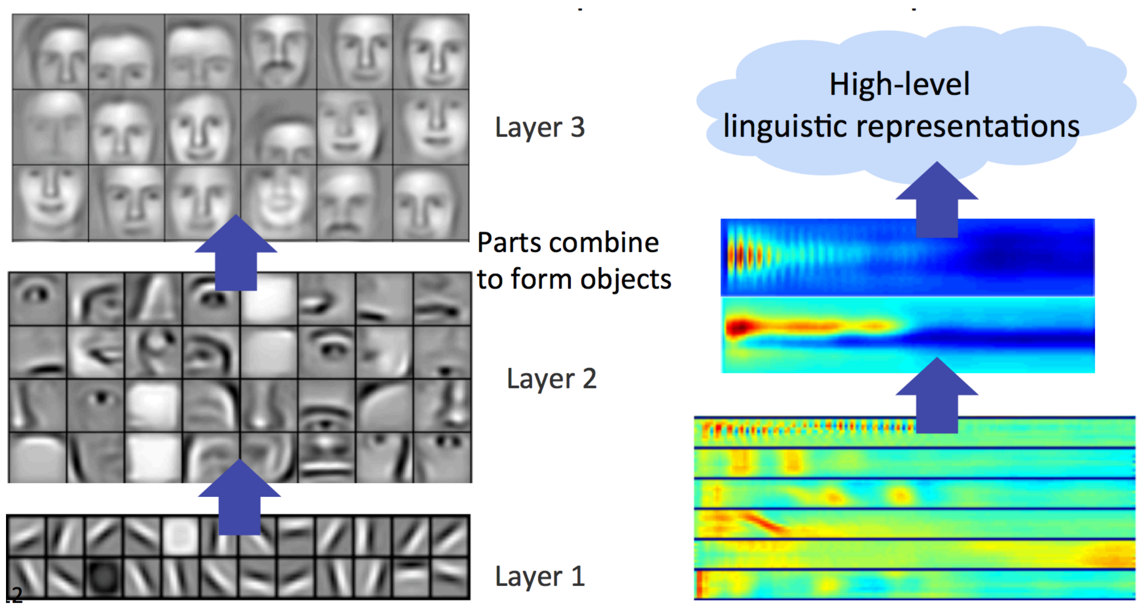
- You need great expertise in good representation, and Deep Learning automatically learns this representation

Why Should We Care about Learning Representation?

- The performance of machine learning methods is heavily dependent on the choice of data representation
- Conventional machine-learning techniques were limited in their ability to process natural data in their raw form.

Why Should We Care about Learning Representation?

- Deep-learning methods are representation-learning methods with multiple levels of representation from a concrete level to slightly more abstract level.

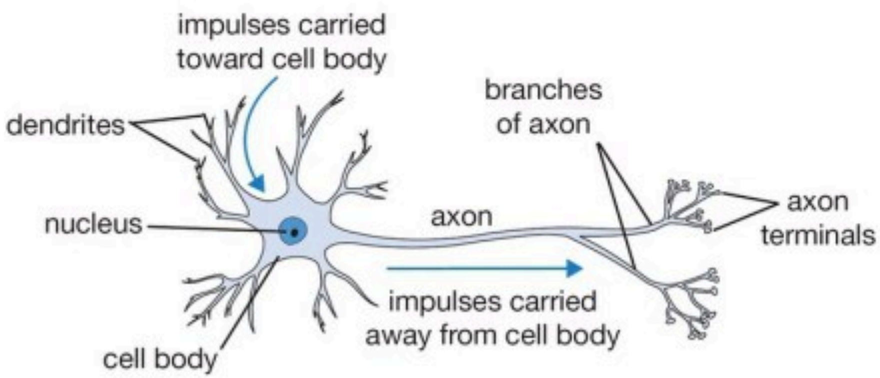


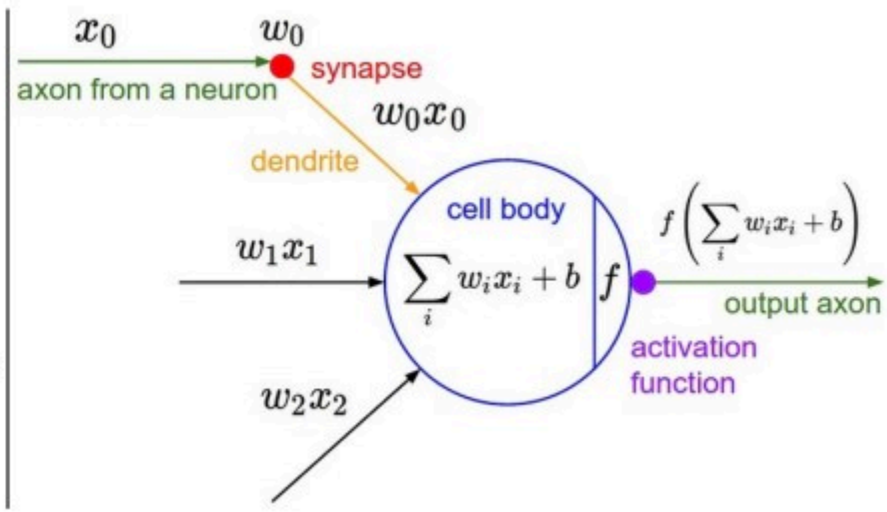
- Very complex functions can be learned by learning representation using Deep-learning

Ch2. Artificial Neural Networks

Neural Networks

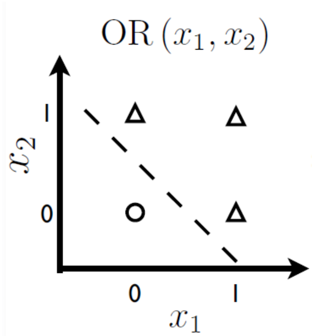
- Artificial neural network mimics the human brain
  - ➢ While neural network-based algorithms for classification or regression may be useful for the purpose of artificial intelligence (AI), neural network itself has nothing to do with.
  - ➢ Single neuron is activated (=1) or not (=0).

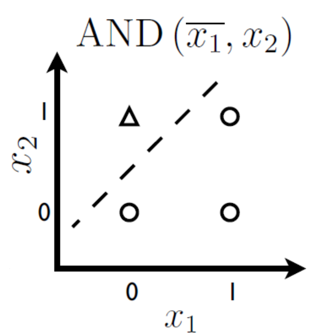


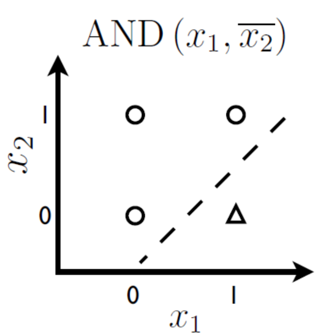


Capacity of Neural Networks

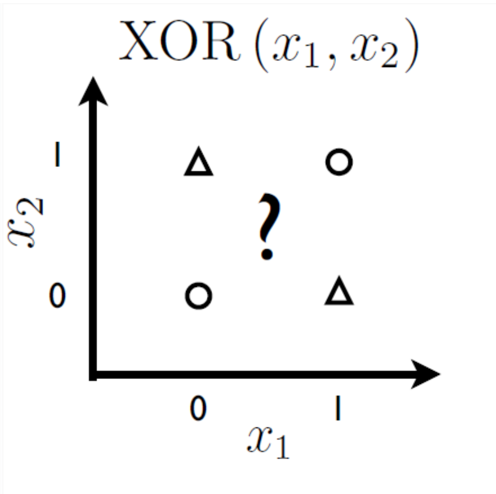
- Linearly separable problems





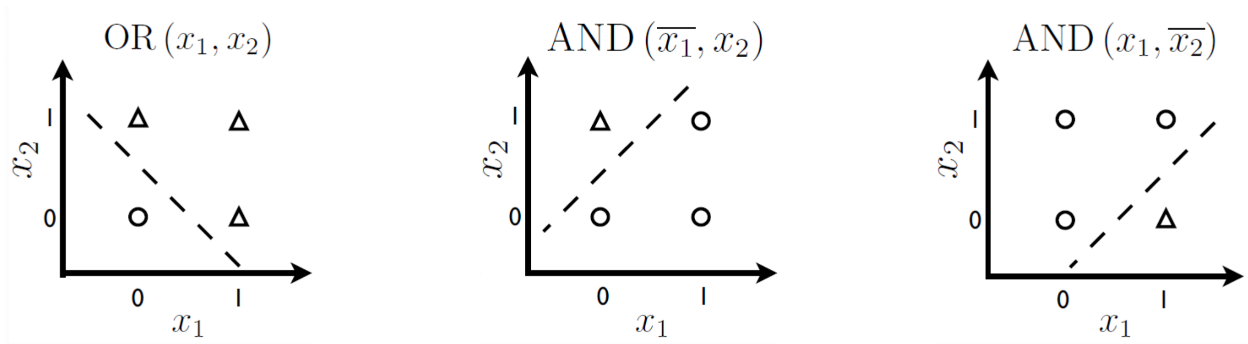


- Non-linearly separable problems

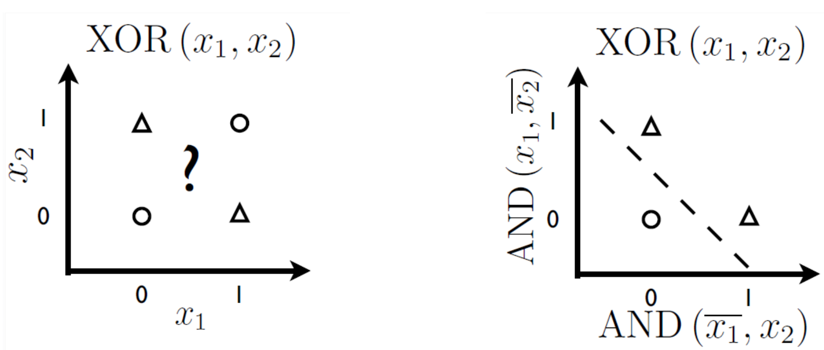


Capacity of Neural Networks

- Artificial neural networks can solve non-linearly separable problems.
  - ➢ That is, neural networks can represent more complex features!
- Linearly separable problems



- Non-linearly separable problems

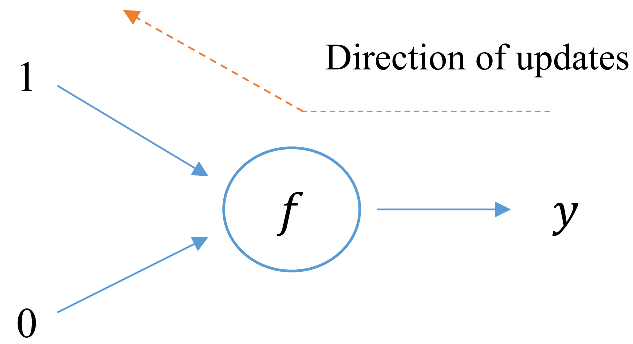


Multi-layer neural networks can solve it by transforming the input in a better representation.

Learning Algorithm of Neural Networks

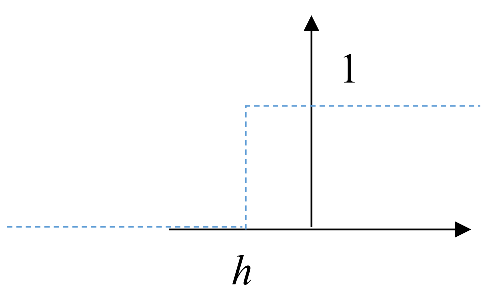
- Back-propagation

  = Learning parameters to minimize the error from the true value in the reverse way



Activation function

$$y = f(w_1 * 1 + w_0 * 0)$$



(1) If the model outputs 1 and the true value is 0, how should you modify the parameters (w_1, w_0, h)?

(2) If the model outputs 0 and the true value is 1, how should you modify the parameters (w_1, w_0, h)?

(3) Until when? Minimizing *the error (loss) function*

Jongho Kim (quantic.jh@gmail.com)

KAIST COLLEGE OF BUSINESS

Ch3. Deep Learning Algorithms

Representation Learning + Deep Architecture

- "Deep Learning can be considered as special case of representation learning algorithms which learn representations of the data in a Deep Architecture with multiple levels of representations." (Najafabadi et al. 2015, p. 5)

- Recent deep learning algorithms are mostly based on neural networks.
  - ➢ Neural network + Deep architecture

    = Feed-forward networks with many hidden layers

    = Multi-layer perceptron (MLP)

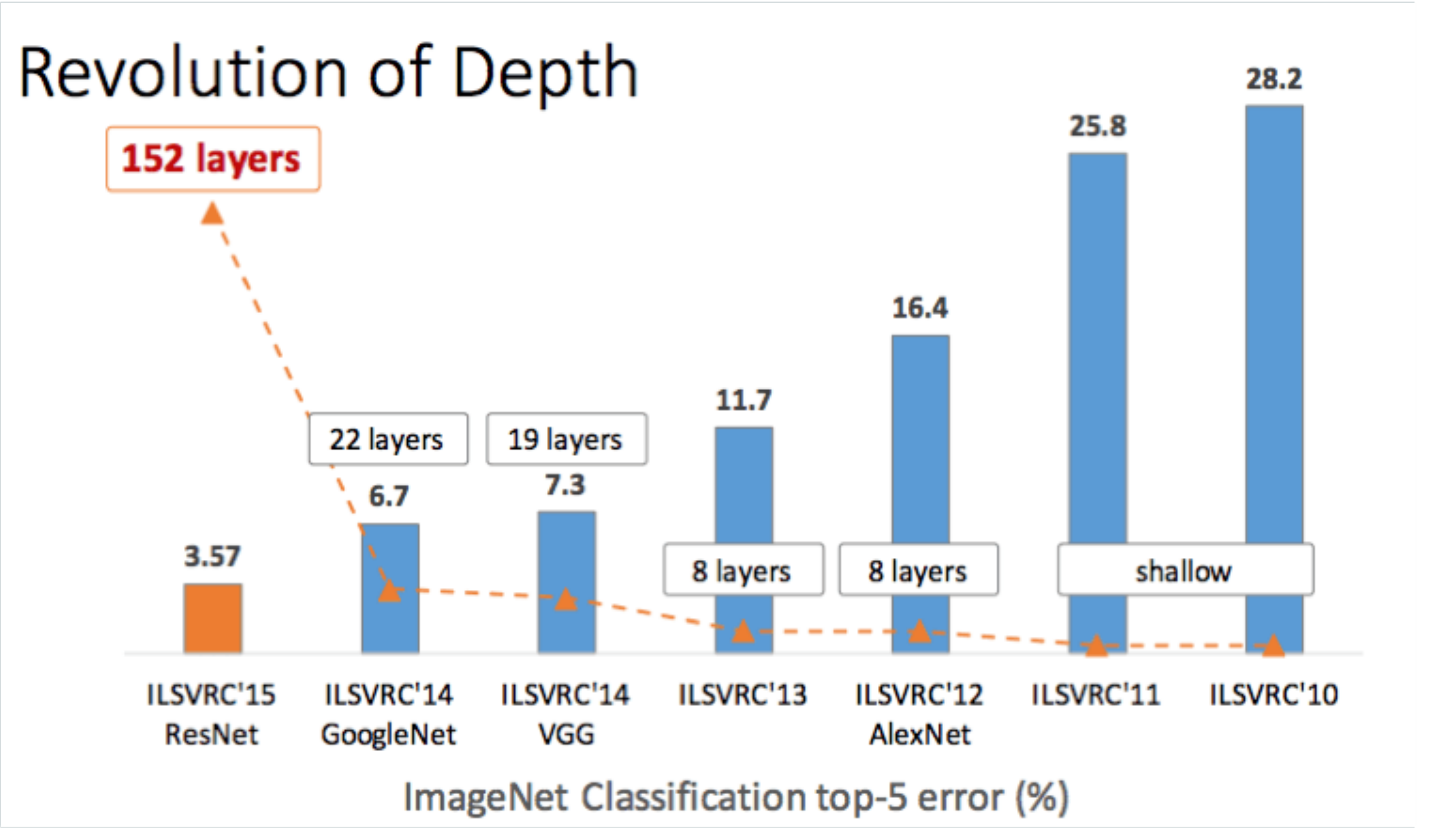
    = Deep neural network (DNN)

  - ➢ More specialized algorithms: CNN, RNN, LSTM, etc.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep Learning Applications and Challenges in Big Data Analytics. *Journal of Big Data*, *2*(1), 1.

Representation Learning + Deep Architecture

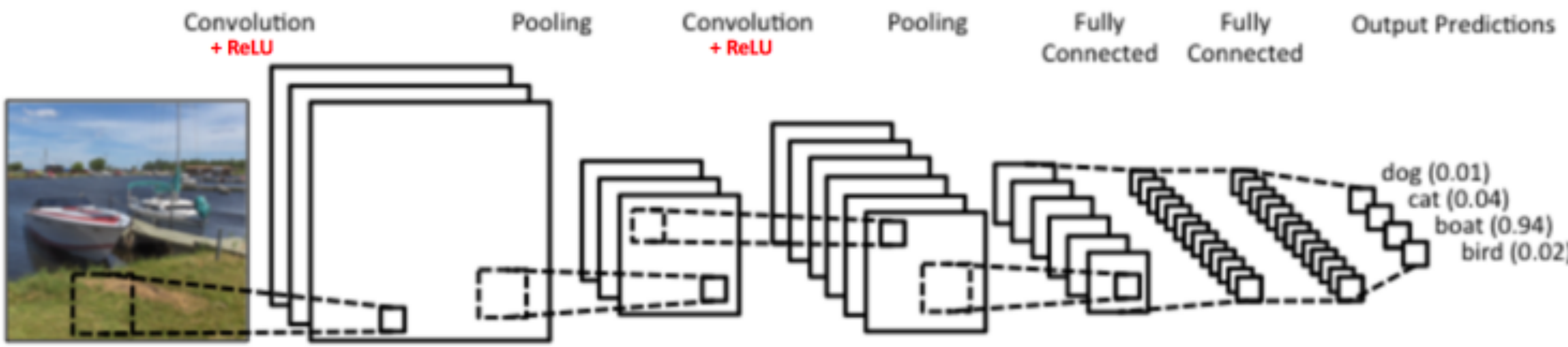
- Then, how deep?



Convolutional Neural Network

What is Convolutional Neural Network (CNN)?

- A typical example of CNN



➢ Inputs (2-dimension × channels) → [Convolution -> ReLu -> Pooling] → …
→ [Convolution -> ReLu -> Pooling] → Fully connected layer
→ Output prediction (Multi-class classification)

Example: 1D Convolution

- How can we apply convolution on 1D time series data?
- 1D Convolution of discrete time signals
- Example
  - Kernel:

-1	1

  - Time series data:

0	0	0	1	0	1	1	1

Example: 1D Convolution

- Example
  - N = 0: -1*0 + 1*0 = 0

-1	1

0	0	0	1	0	1	1	1

  - N = 1: -1*0 + 1*0 = 0

-1	1

0	0	0	1	0	1	1	1

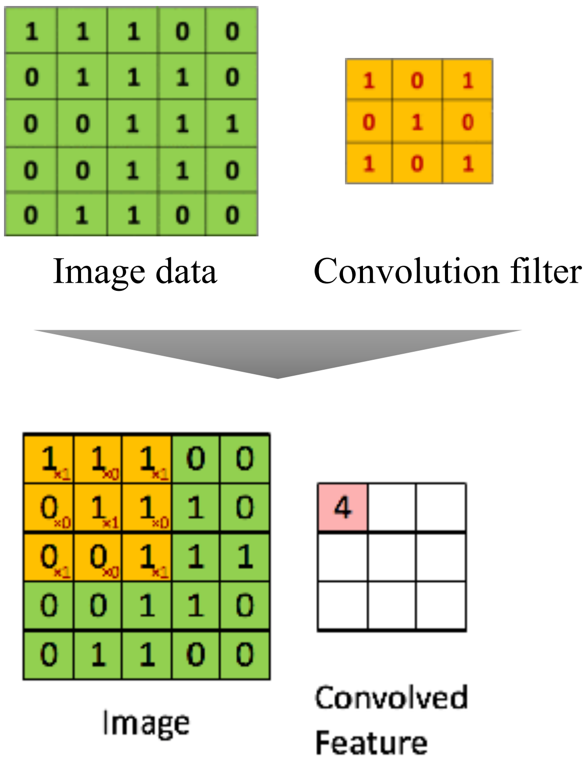
  - N = 2: -1*0 + 1*1 = 1

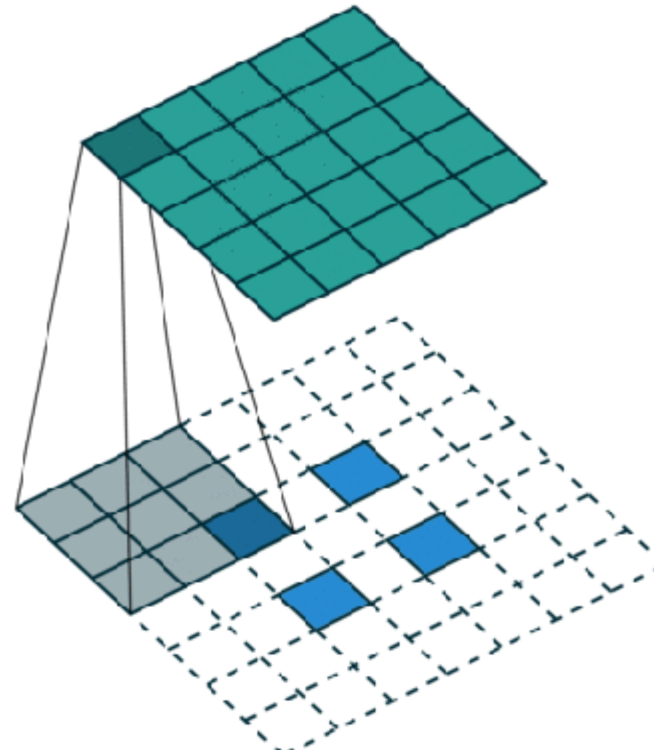
-1	1

0	0	0	1	0	1	1	1

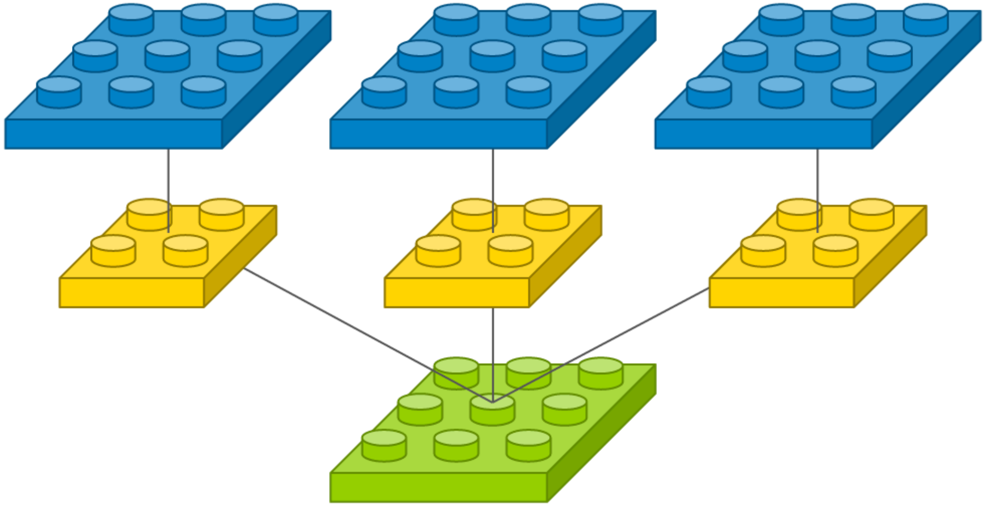
Example: 2D Convolution

- Convolution



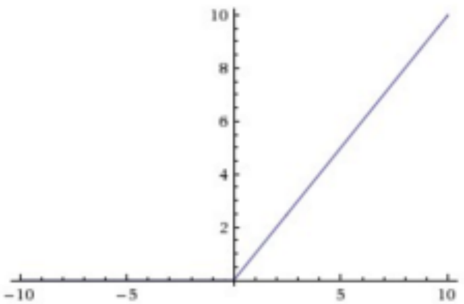


Applying 2D Multiple Kernel

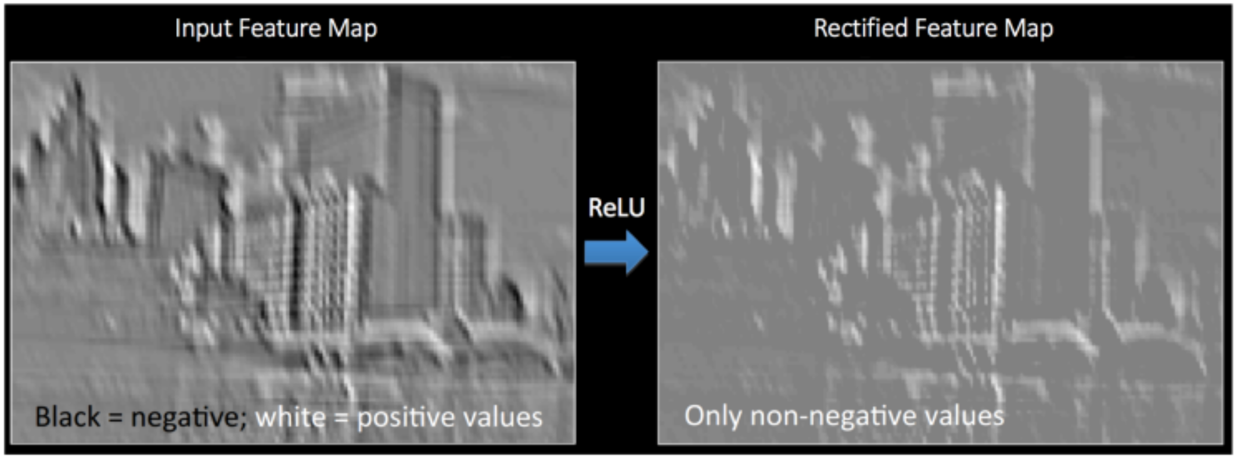


ReLU

- Introducing non-linearity (ReLU)
  - ➢ The purpose of ReLU is to introduce non-linearity in CNN.
  - ➢ Other non-linear functions such as tanh or sigmoid can also be used, but ReLU has been found to perform better in most situations.



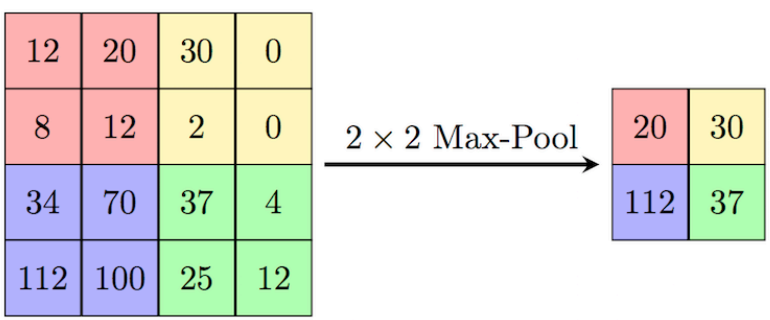
Output = Max(zero, Input)

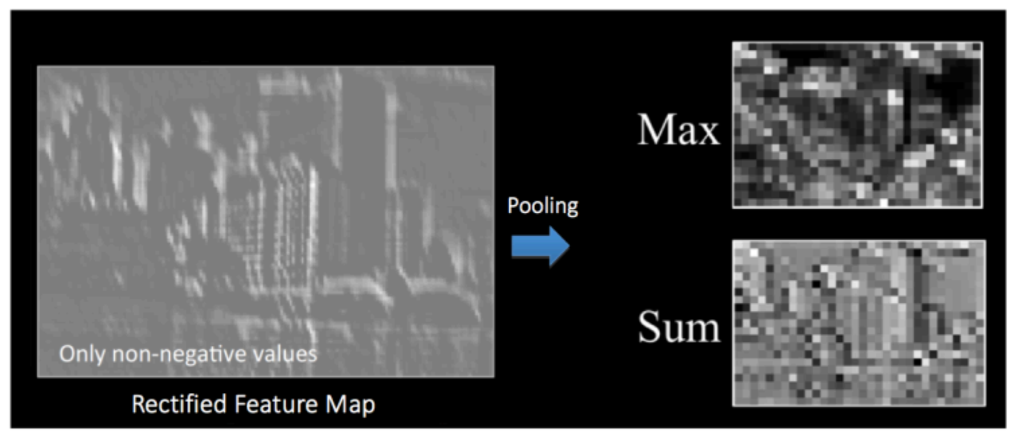


Source: An Intuitive Explanation of Convolutional Neural Networks (Data Science Blog)
https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Max Pooling

- Reducing the dimensionality of each feature map (Pooling)
  - ➢ The purpose of pooling is to reduce the number of parameters and computations in the network, therefore, controlling overfitting.
  - ➢ Another purpose is to make the network invariant to small transformations, distortions and translations in the input image.
  - ➢ Pooling can be of different types: Max, Average, Sum, etc.

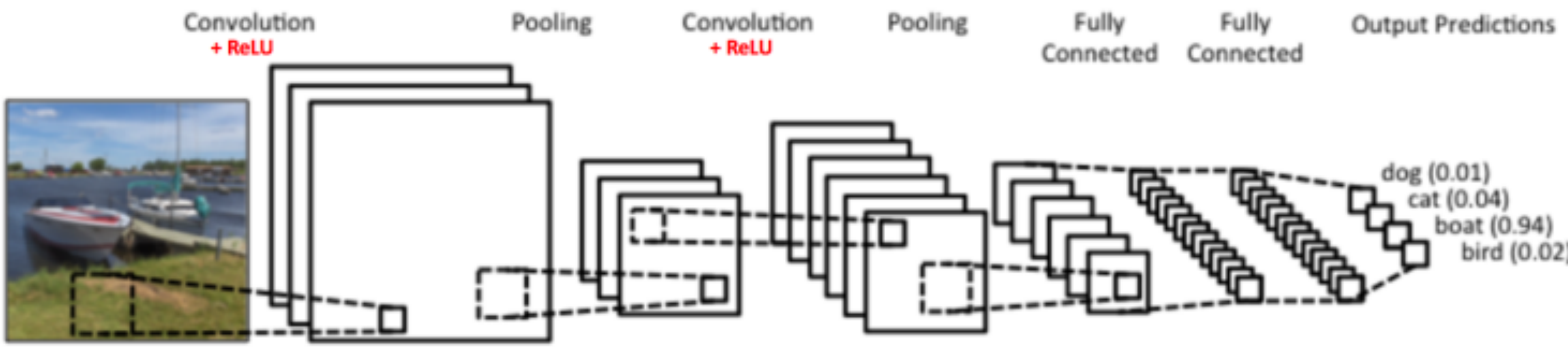




Source: An Intuitive Explanation of Convolutional Neural Networks (Data Science Blog)
https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

What is Convolutional Neural Network (CNN)?

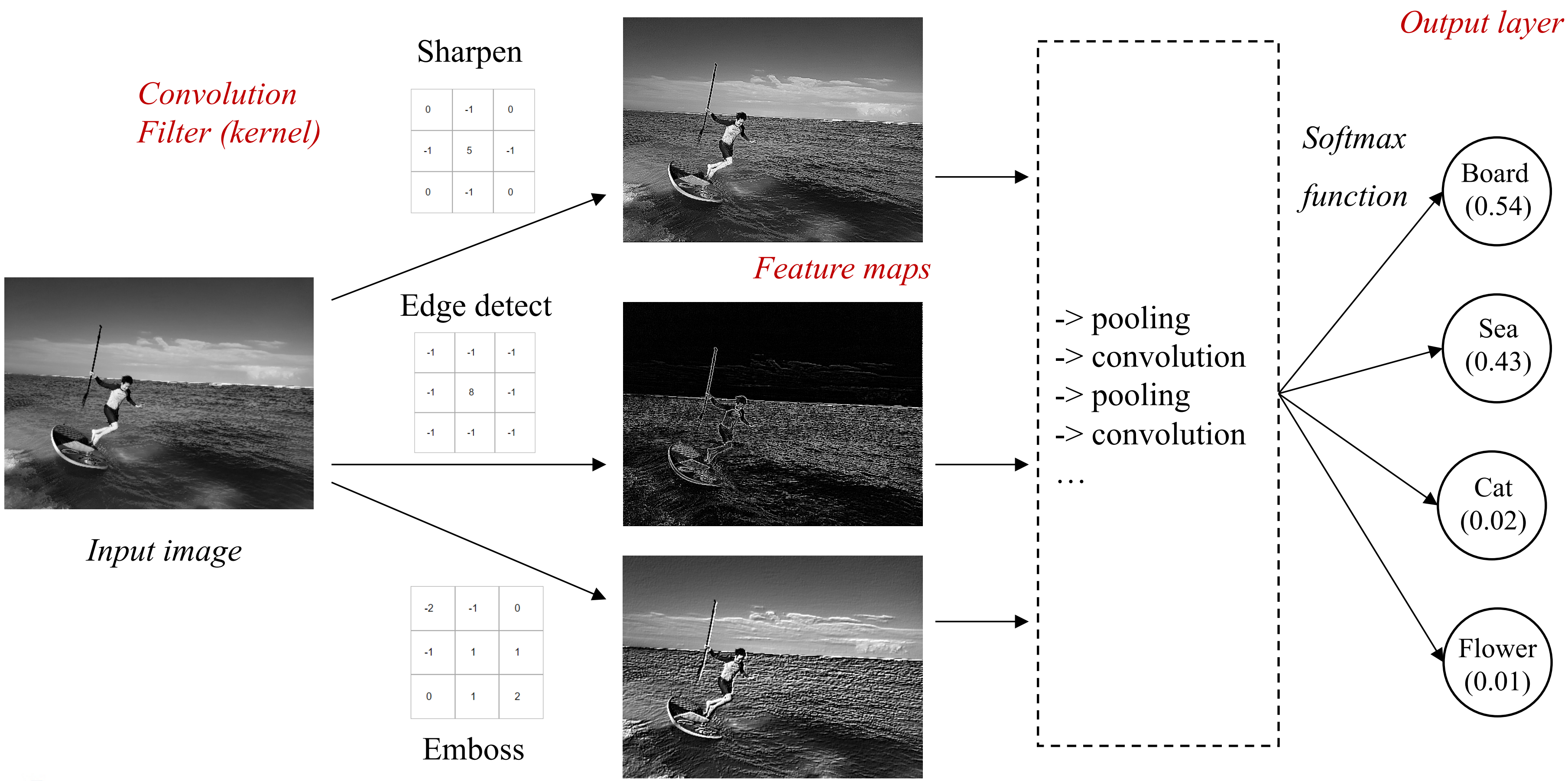
- A typical example of CNN



➢ Inputs (2-dimension × channels) → [Convolution -> ReLu -> Pooling] → …
→ [Convolution -> ReLu -> Pooling] → Fully connected layer
→ Output prediction (Multi-class classification)

Convolutional Neural Network (CNN)

- CNN automatically learns the best filters to extract the feature maps for image classifications through back propagation.



Source: Image Kernels demo, http://setosa.io/ev/image-kernels/

Convolutional Neural Network (CNN)

- CNN has been considered as a basic deep learning algorithm. Why is CNN superior at feature representations?
  - Effectively reducing computational complexity (Convolution filter)
  - Hierarchical feature representation (multiple convolution layers)
  - Non-linearity and less overfitting (Rectified Linear Unit (ReLU) and Pooling)

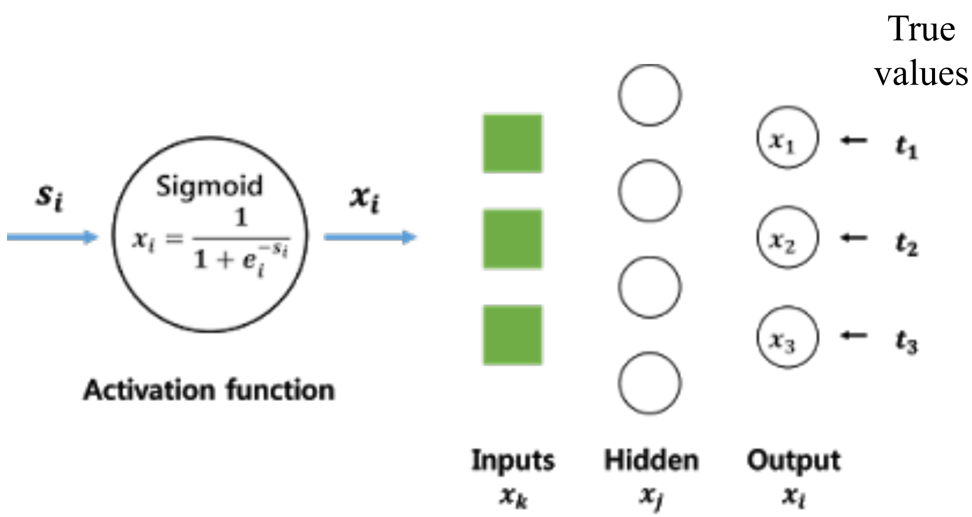
Thank you ☺

Contact Info: quantic.jh@gmail.com

(Appendix) Learning Algorithm of Neural Networks

- Gradient descent method (Back-propagation)

  = Learning parameters to minimize the error from the true value in the systematic, reverse way even for multiple hidden layers



The goal is to minimize the following error (loss) function,

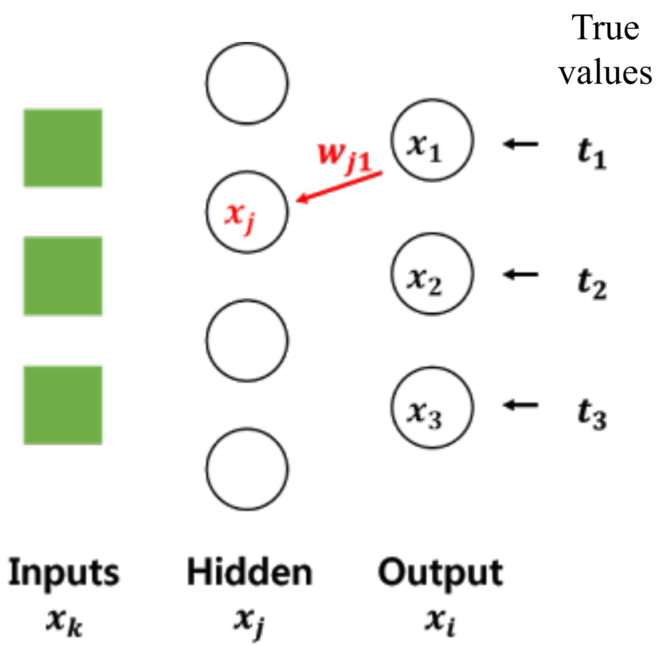
$$E = -\sum_{i=1}^{3}[t_i \log(x_i) + (1 - t_i)\log(1 - x_i))]$$

Source: Backpropagation 설명 예제와 함께 완전히 이해하기 (Jaejun Yoo's Playground)
http://jaejunyoo.blogspot.com/2017/01/backpropagation.html

(Appendix) Learning Algorithm of Neural Networks

- Gradient descent method (Back-propagation)

  = Learning parameters to minimize the error from the true value in the systematic, reverse way even for multiple hidden layers



Firstly, update the parameters between output and hidden layers.

$$\frac{\partial E}{\partial w_{j1}} = \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}} \qquad \text{(Chain rule)}$$

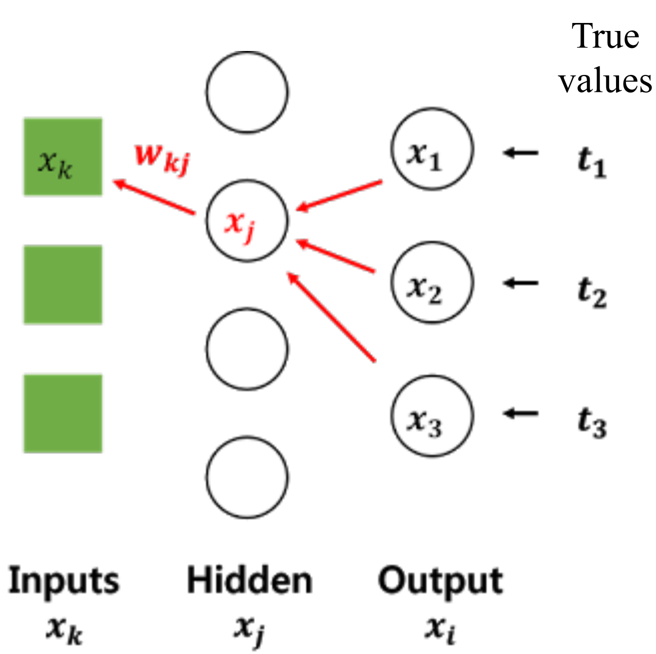
$$\therefore \frac{\partial E}{\partial w_{j1}} = (x_i - t_i)x_j$$

Source: Backpropagation 설명 예제와 함께 완전히 이해하기 (Jaejun Yoo's Playground)
http://jaejunyoo.blogspot.com/2017/01/backpropagation.html

(Appendix) Learning Algorithm of Neural Networks

- Gradient descent method (Back-propagation)

  = Learning parameters to minimize the error from the true value in the systematic, reverse way even for multiple hidden layers



Secondly, update the parameters between hidden and input layers.

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial s_j}\frac{\partial s_j}{\partial w_{ki}}$$

$$\frac{\partial E}{\partial w_{kj}} = \sum_{i=1}^{3} \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial x_j}\frac{\partial x_j}{\partial s_j} \times \frac{\partial s_j}{\partial w_{ki}}$$

$$\therefore \frac{\partial E}{\partial w_{kj}} = \sum_{i=1}^{3} (x_i - t_i) w_{ji} (x_j(1 - x_j)) \times x_k$$

Source: Backpropagation 설명 예제와 함께 완전히 이해하기 (Jaejun Yoo's Playground)
http://jaejunyoo.blogspot.com/2017/01/backpropagation.html