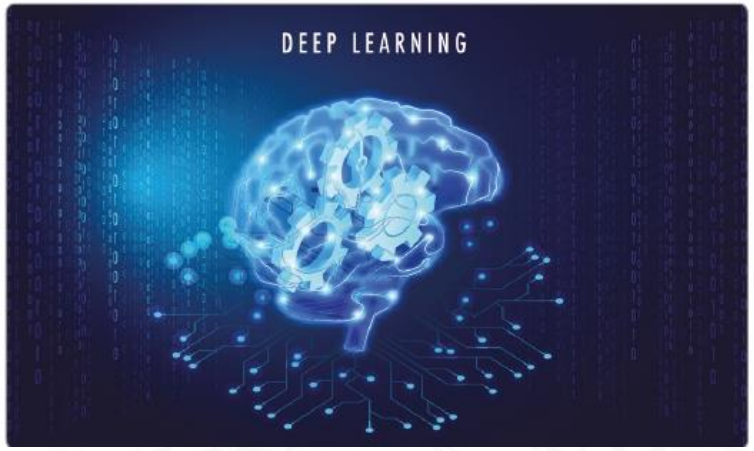
Neural Network

Introduction

❖ Appearance of Deep

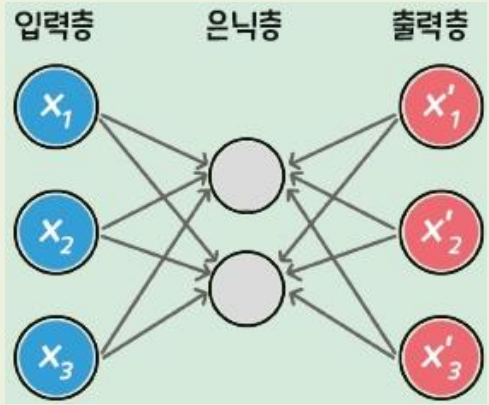
- In 2006, Jeffrey Hinton, who devised backpropagation, presented a study that showed that deep neural networks could learn if the initial value of weights was set properly
- In 2007, Benggio's team proposed a simpler pre-training method using Autoencoder
- This is when the term "Deep" was used instead of the artificial neural network



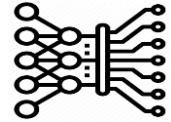
❖ Appearance of Deep

하나 더 알기 **오차 계산**

- **오토인코더(Autoencoder) :** 입력층과 출력층이 동일한 네트워크에 데이터를 입력하여 비지도학습을 하는 것
- 인코더를 통해 입력 데이터에 대한 특징을 추출하고, 디코더를 통해 원본 데이터를 재구성하는 학습
- 가중치의 좋은 초기값을 얻는 목적으로 이용됨



Introduction

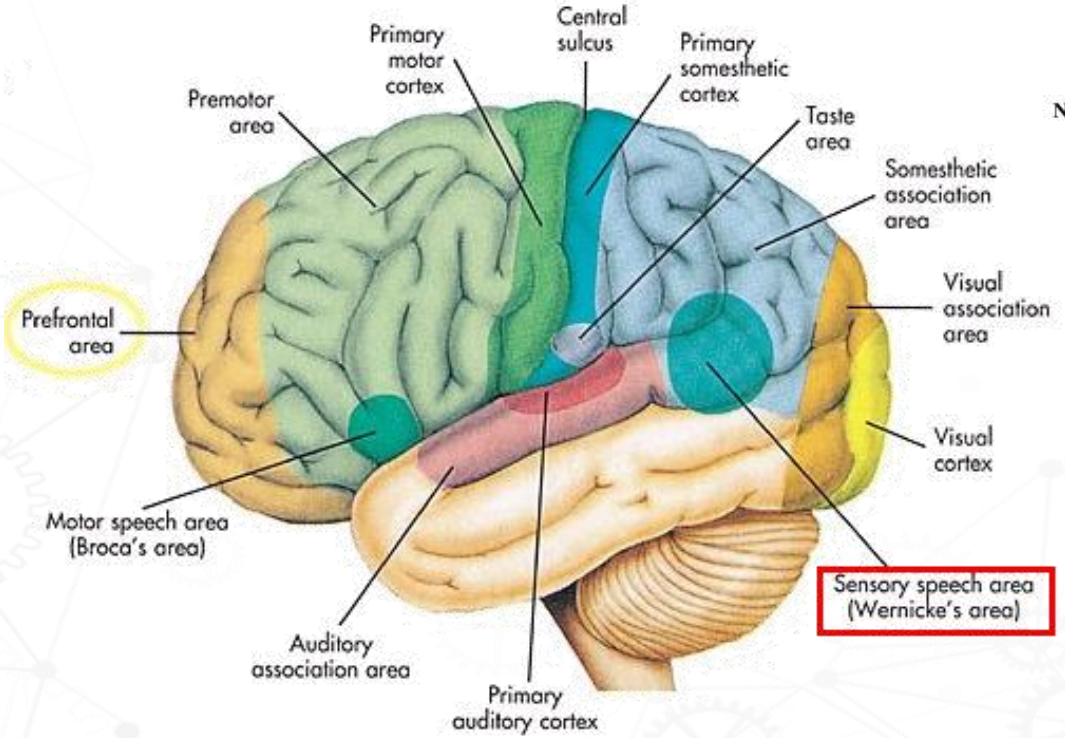


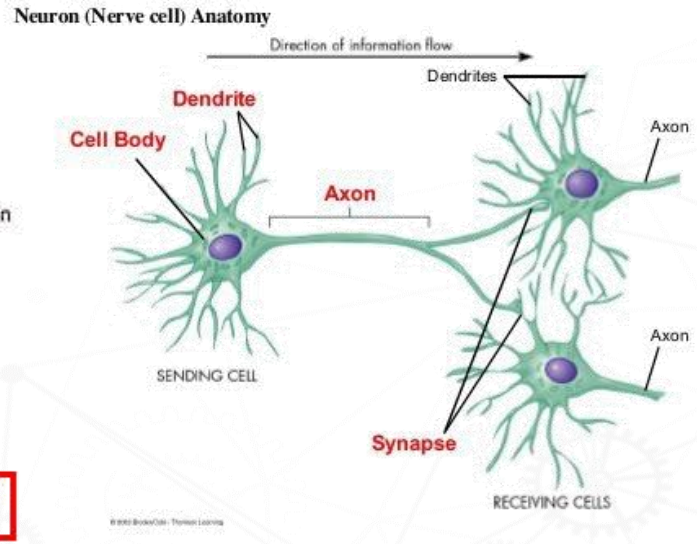
- Concepts of artificial neural network
  - The origin of deep learning is artificial neural networks
    - There are a lot of neurons in the human brain, and they're connected by synapses, which we call neural networks
  - Because artificial neural networks were created based on the human neural network structure, they artificially mimicked the connection of neurons, that is, neural networks

Neural Mechanism for Neural Network

❖ The structure of Brain

- Numerous neurons (neural cells) perform various functions as they form a network structure through synapses

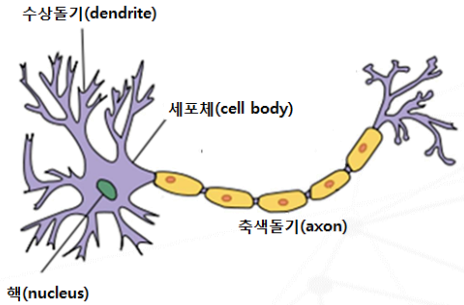




Neural Mechanism for Neural Network

❖ The structure of Neuron

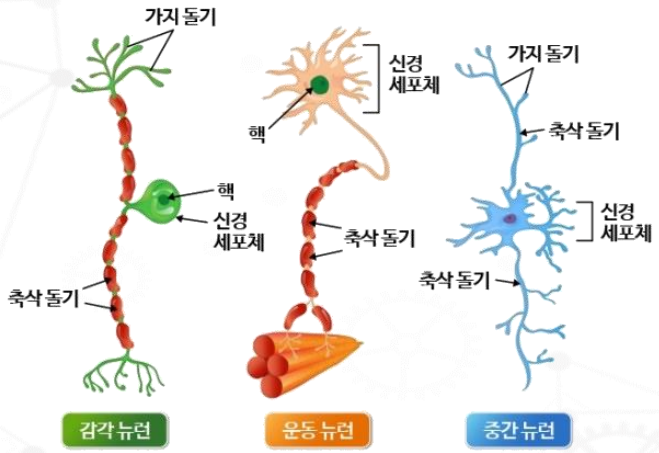
- Cell body (soma)
  - Located in the center of the nerve cell
  - Acceptance of information, computational processing, and transmission of output
- Dendrite
  - A number of thick, short bumps spread wide like branches
  - Link to other neurons to receive input signals

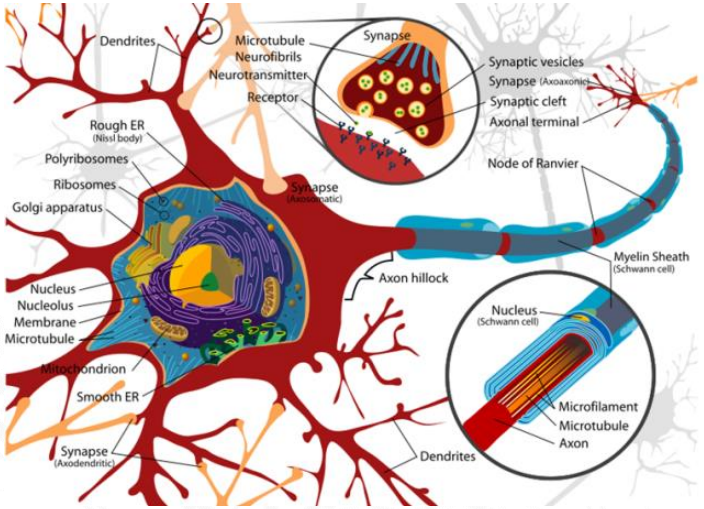


- Axon: a single thin fiber
  - It's attached to the cell body, it's electrically activated, it transmits pulses generated by neurons to other neurons
  - The ends of the axons are divided into thin branches
  - Connect via synapses with dendritic protrusions in other neurons

Neural Mechanism for Neural Network

- The exchange of information between neurons through synapses
- About 100 billion (10^{11}) neurons in the human cerebral cortex
- Each neuron has about 1,000 dendrites
- 100 trillion (10^{14}) synapses
- The main function of a neuron
  - Generating and transmitting electrical pulses of the action potential

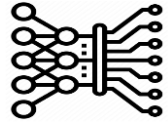




Comparison between Brain and Computer

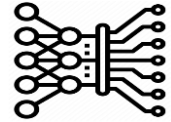
	Processing element	Element size	Energy use	Processing speed	Style of computation	Fault tolerant	Learns	Intelligent, conscious
	10^{14} synapses	10^{-6}m	30 W	100 Hz	parallel, distributed	yes	yes	usually
	10^{8} transistors	10^{-6}m	30 W (CPU)	10^{9} Hz	serial, centralized	no	a little	not (yet)

History of Neural Network



- The advent of perceptron
  - In 1969, Marvin Minsky and Seymour Peppert mathematically proved the hypothesis that XOR computation is absolutely impossible with current perceptrons
  - Multi-layer perceptron (MLP) stacked with multiple perceptrons can solve the problem of XOR computation, but it mentions the limitations of learning with perceptron, concluding that 'there is no way to learn each weight and bias'
  - Since then, artificial neural network research has faced a recession

History of Neural Network



- Progression (1943-1960)
  - First Mathematical model of neurons, Pitts & McCulloch (1943)
  - Beginning of artificial neural networks–**Perceptron**, Rosenblatt (1958)
- Degression (1960-1980)
  - Perceptron can't even learn the XOR function
  - We don't know how to train **MLP**
  - 1963 Backpropagation (Bryson et al.)
- Progression (1980-)
  - 1986 **Backpropagation** reinvented
- Degression (1993-)
  - SVM: Support Vector Machine is developed by Vapnik et al.[1995]
  - Graphical models are becoming more and more popular
  - Training deeper networks consistently yields poor results.
  - However, **Yann LeCun** (1998) developed deep **convolutional neural networks**
- Progression (2006-)
  - Deep Belief Networks (**DBN**) by **Hinton** et al. (2006)
  - Deep Autoencoder based networks by Greedy Layer-Wise Training of Deep Networks. **Bengio** et al.
  - Convolutional neural networks running on GPUs
  - **AlexNet** (2012). Krizhevsky et al.

출처: http://www.cs.cmu.edu/~10701/slides/Perceptron_Reading_Material.pdf

Structure of Perceptron

❖ Structure

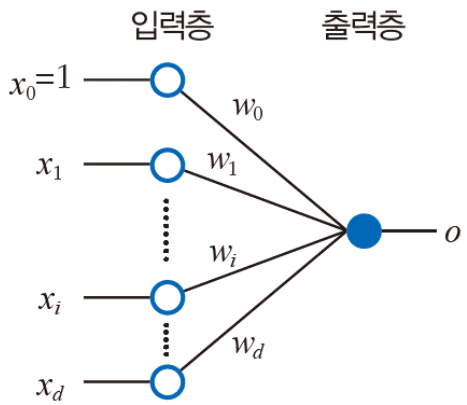
- Composed of input and output layers (output layer is one node)
- The input layer has d+1 nodes

  특징 벡터: $\mathbf{x}=(x_1, x_2, \cdots, x_d)$

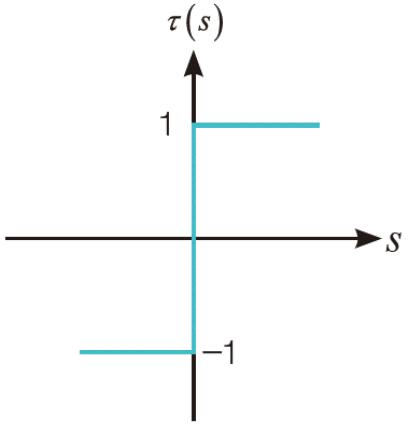
  - d is the dimension of the feature vector

    e.g., iris: d = 4, digit: d = 64

- The i^{th} input node and output node are connected by an edge with a weight *w*

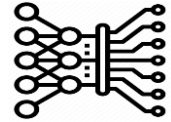


(a) 퍼셉트론의 구조



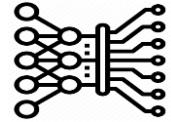
(b) 계단 함수를 활성 함수로 사용

Learning Mechanism



- Perceptron learning
  - Given a perceptron (weight) that recognizes the data along with the data
    - That is, given a perceptron that has completed learning with data
  - Since only data is given in real-world situations, the learning algorithm weights (*w0,w2,…),wd*) must be determined
    - For the OR data, it is a two-dimensional feature vector and has only four samples, so you can easily figure out the weights with a pencil
    - Sklearn's handwritten numeric data is a 64-D feature vector and has 1,797 samples, making it impossible without a learning algorithm

Learning Mechanism



❖ If you describe the process of learning swimming in an algorithmic format,

[알고리즘 4-1] 사람의 수영 학습

```
01. 적절한 동작을 취한다.
02. while (true)
03.     동작에 따라 수영을 하고 평가한다.
04.     if (만족스러움) break                    # break문으로 루프 탈출
05.     더 나은 방향으로 동작을 수정한다.
06. 동작을 기억한다.
```

[알고리즘 4-2] 사람의 수영 학습(수학적 기술)

```
01. 초기 동작 벡터 w=(팔 돌리는 속도, 팔꿈치 각도, 팔과 귀의 거리)를 초기화한다.
02. while (true)
03.     w에 따라 수영을 하고 동작 w의 점수 J(w)를 계산한다.
04.     if (J(w)가 만족스러움) break
05.     더 나은 방향 Δw를 계산한다.
06.     w=w+Δw
07. w를 기억한다.
```

Learning Mechanism

❖ The process of finding and improving a better direction little by little is the same as human learning

- Neural networks differ from humans in that they rely strictly on math
  - Learning of neural networks is an optimization problem
  - Optimization target is the neural network's weight (parameter)

[알고리즘 4-3] 신경망의 학습

입력: 훈련 데이터

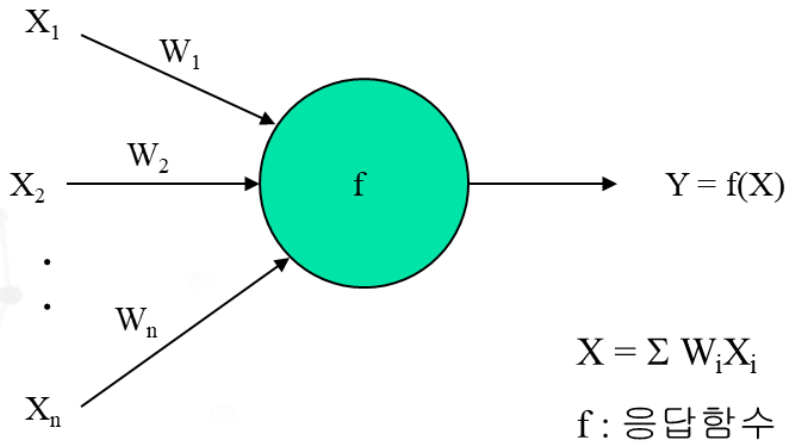
출력: 최적의 매개변수 값

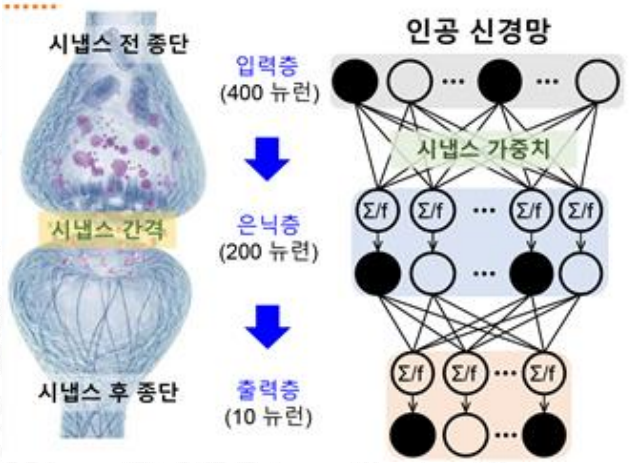
01. 난수로 매개변수 벡터 **w**를 초기화한다. # **w**는 신경망의 가중치([그림 4-2(a)]의 $(w_0, w_1, \cdots, w_d)$)
02. while (true)
03. **w**에 따라 데이터를 인식하고 손실 함수 $J(\mathbf{w})$를 계산한다.
04. if ($J(\mathbf{w})$가 만족스러움) break
05. 손실 함수 값을 낮추는 방향 $\Delta\mathbf{w}$를 계산한다.
06. $\mathbf{w}=\mathbf{w}+\Delta\mathbf{w}$
07. **w**를 저장한다.

Learning Mechanism

❖ Neuronal model (multi-input, one-output) (MaCulloch & Pitts) (1943)

- Formal neuron processing element (PE)
  - Unit, cell, node





Learning Mechanism

- Y = f(X): Y is the state of PE at that point

  Y = 1 neuron excited, Y = 0 neuron suppressed

  Wi: The strength of synaptic connections of the i^{th} input (weight, strength)

  - Positive value for excitatory coupling (exciting synapse)
  - Negative value for inhibitory binding (inhibitive synapse)

  *Value changes due to learning

- Example of an active function

  - Step function

$$f(X) = \begin{cases} 1 & X \geq 0 \\ 0 & X < 0 \end{cases}$$

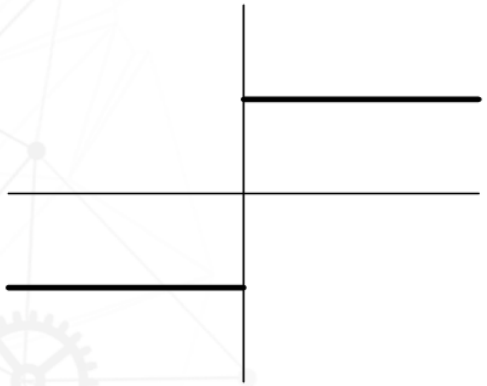
  - Sigmoid function

$$f(X) = 1/(1+\exp(-X))$$

Learning Mechanism

❖ Kinds of activation function

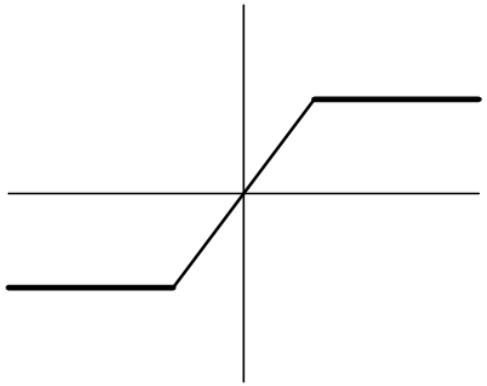
(1) 계단함수



$$\begin{cases} y = -1 & (x \le 0) \\ y = +1 & (x > 0) \end{cases}$$

불연속

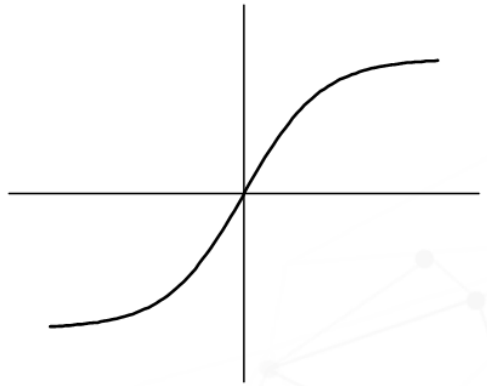
(2) 임계함수



$$\begin{cases} y = -1 & (x \le -1) \\ y = x & (-1 < x \le 1) \\ y = +1 & (x > 1) \end{cases}$$

연속이나 한 개의 함수값으로 표현 불가능

(3)시그모이드함수



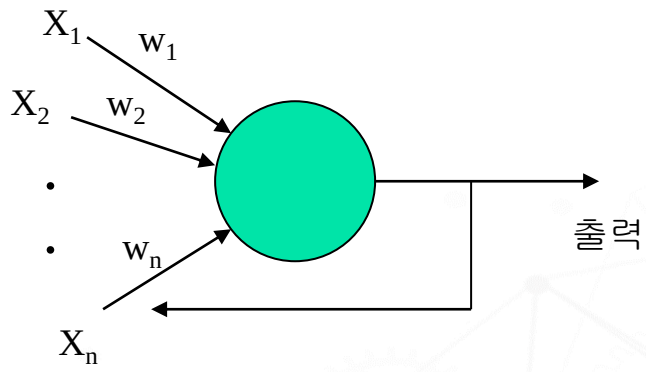
$$y = \frac{2}{1 + e^{-x}} - 1 \qquad (-\infty < x < \infty)$$

연속이며 모든 점에서 미분가능

Learning Mechanism

❖ Hebb's Learning Rules

- Psychologist Hebb (1949): Proposing a Learning Model of the Brain
- Physiological learning rule techniques for adjusting the connection strength of synapses known as "Hebb's synapses"

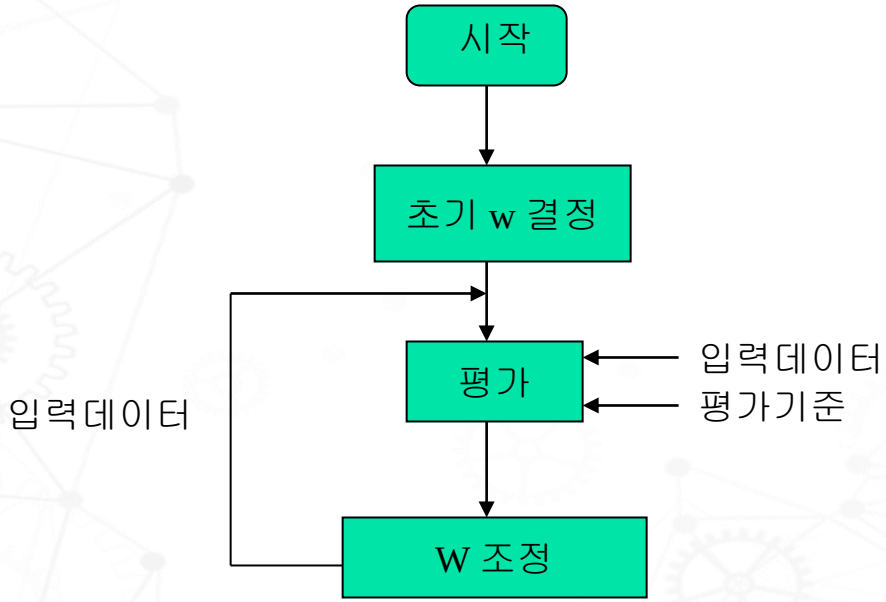


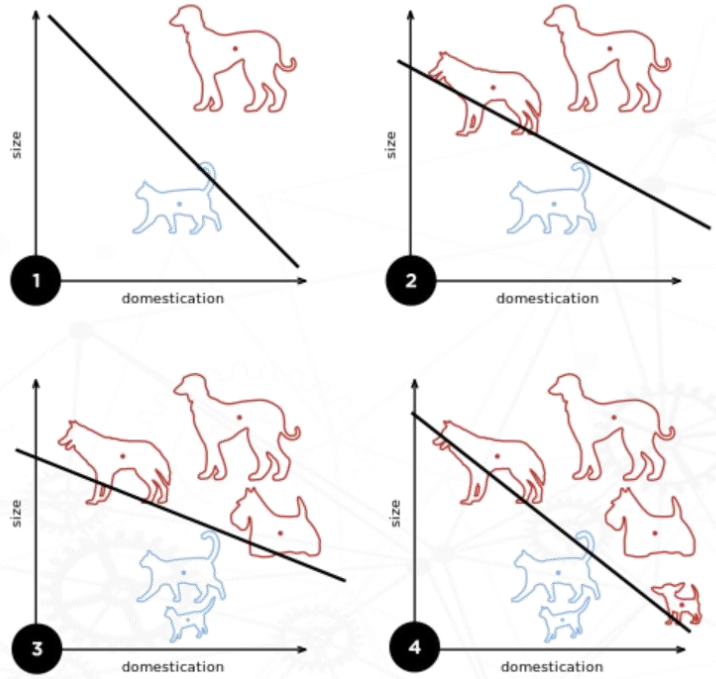
Hebb’s learning model

Learning Mechanism

❖ Learning procedure

- Make a random decision without initial value
- Enter the learning data to evaluate using the given evaluation criteria
- Adjust the weight according to the evaluation results
- Repeat iteration, approach to optimum

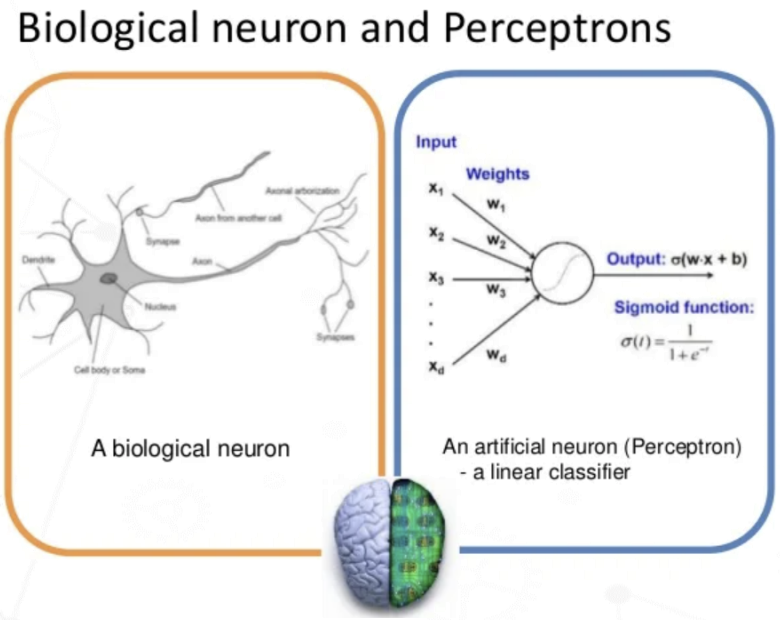


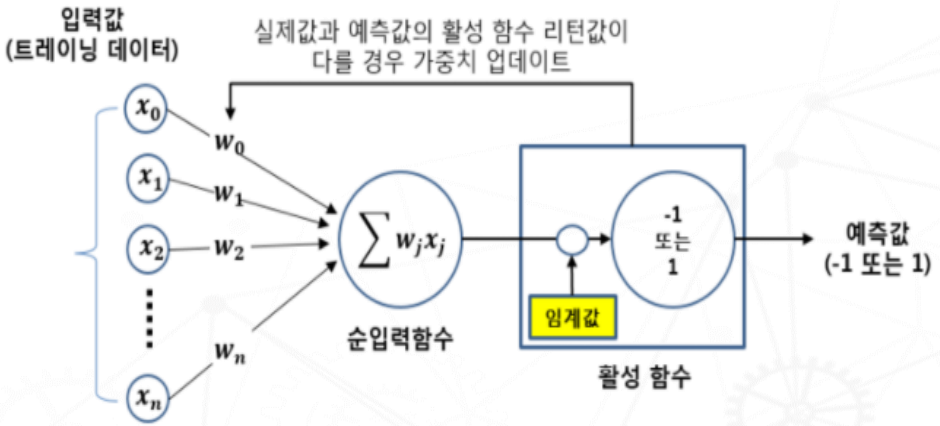


Learning Mechanism

❖ Single Perceptron

- A neural network model proposed by psychologist Rosenblatt (1957) to classify patterns
- Model that can improve recognition by learning when given two signals from the outside





Learning Mechanism

❖ The process of finding and improving a better direction little by little is the same as human learning

- Neural networks differ from humans in that they rely strictly on math
  - Learning of neural networks is an optimization problem
  - Optimization target is the neural network's weight (parameter)

[알고리즘 4-3] 신경망의 학습

입력: 훈련 데이터

출력: 최적의 매개변수 값

01. 난수로 매개변수 벡터 **w**를 초기화한다. # **w**는 신경망의 가중치([그림 4-2(a)]의 $(w_0, w_1, \cdots, w_d)$)
02. while (true)
03. **w**에 따라 데이터를 인식하고 손실 함수 $J(\mathbf{w})$를 계산한다.
04. if ($J(\mathbf{w})$가 만족스러움) break
05. 손실 함수 값을 낮추는 방향 $\Delta\mathbf{w}$를 계산한다.
06. $\mathbf{w}=\mathbf{w}+\Delta\mathbf{w}$
07. **w**를 저장한다.

Learning Mechanism

❖ The process of finding and improving a better direction little by little is the same as human learning

- Neural networks differ from humans in that they rely strictly on math
  - Learning of neural networks is an optimization problem
  - Optimization target is the neural network's weight (parameter)

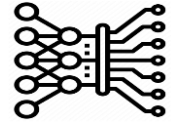
[알고리즘 4-3] 신경망의 학습

입력: 훈련 데이터

출력: 최적의 매개변수 값

01. 난수로 매개변수 벡터 **w**를 초기화한다. # **w**는 신경망의 가중치([그림 4-2(a)]의 $(w_0, w_1, \cdots, w_d)$)
02. while (true)
03. **w**에 따라 데이터를 인식하고 손실 함수 $J(\mathbf{w})$를 계산한다.
04. if ($J(\mathbf{w})$가 만족스러움) break
05. 손실 함수 값을 낮추는 방향 $\Delta\mathbf{w}$를 계산한다.
06. $\mathbf{w}=\mathbf{w}+\Delta\mathbf{w}$
07. **w**를 저장한다.

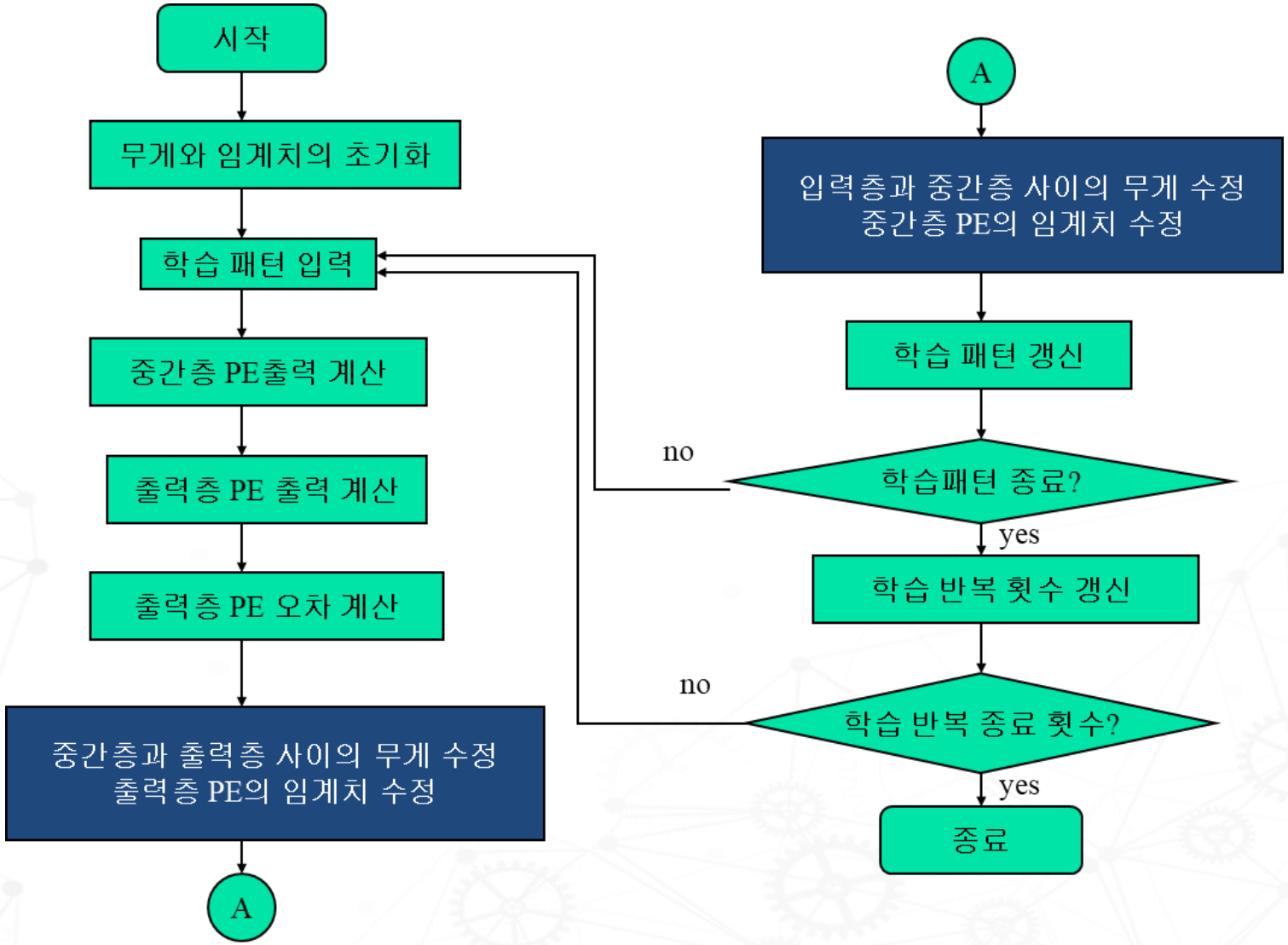
Learning Mechanism



- A neural network with one or more hidden layers between the input layer and the output layer
- Generalized delta rule, learning by backpropagation rule
- Backpropagation learning algorithms follow the slope

- 문제점
  - 지역 최소값(local minima)에 빠질 염려
  - 전방향(feedforward) 방식
  - 학습이 수렴할 때까지 많은 시간 걸림
  - 추가학습시 전체적인 재학습 필요
  - 학습의 완료시점을 예측할 수 없음

Learning Mechanism

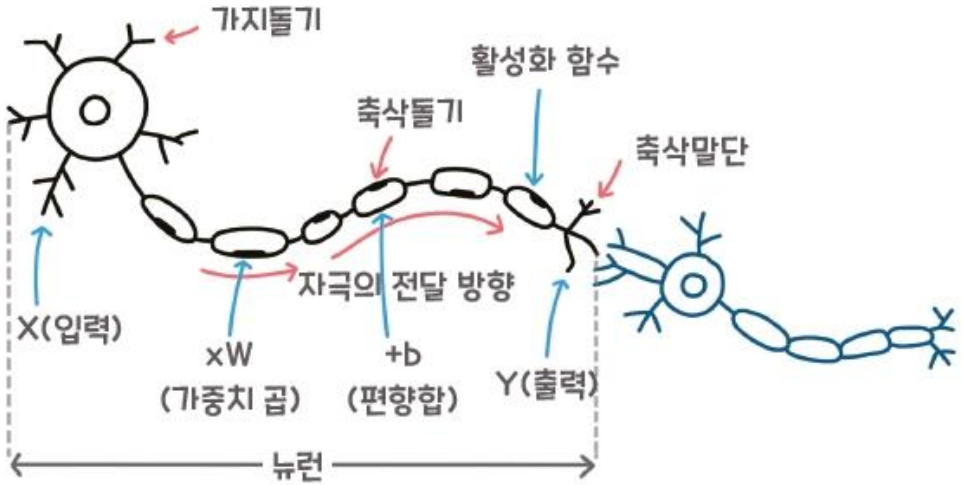


Comparison between Neural Networks

신경망모델	주응용분야	장점/비고	단점
Hopfield/Kohonen	최적화문제	대규모구현	학습기능 없음, w고정
Perceptron	인쇄체문자인식	최초의 신경망	변화민감, XOR해결 못함
Multilayer Perceptron	패턴인식	Feed -forward형	복잡한 패턴 인식 불가
Input driven MLP	문자, 음성인식	학습용이, MLP개선	방대한 학습 데이터 사용
Boltzmann Machine	패턴인식	최소 에너지 상태 도달	학습시간이 김
Undirected FT model	문자인식에서 전처리	병렬형 전자회로 설계	학습기능 없음
Selt-Organizing Map	기하학정 영역 mapping	성능이 좋음	광범위한 학습
Neocognitron	문자인식	복잡한 패턴인식	뉴런과의 연결이 많음

❖ Concept

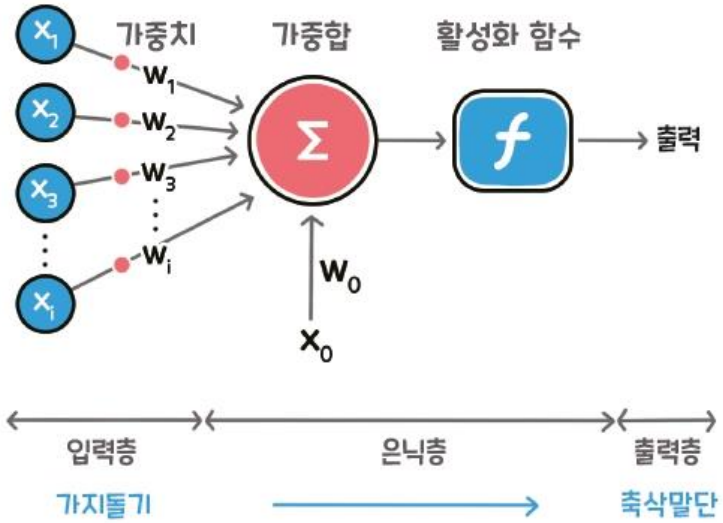
- The basic mechanism of a human neural network



① 가지돌기에서 신호를 받아들임
② 신호가 축삭돌기를 지나 축삭말단으로 전달됨
③ 축삭돌기를 지나는 동안 신호가 약해져서 축삭말단까지 전달되지 않거나 강하게 전달되기도 함
④ 축삭말단까지 전달된 신호는 다음 뉴런의 가지돌기로 전달됨
⑤ 수억 개의 뉴런 조합을 통해 손가락을 움직이거나 물체를 판별하는 등 다양한 조작과 판단 수행 가능

❖ Concept

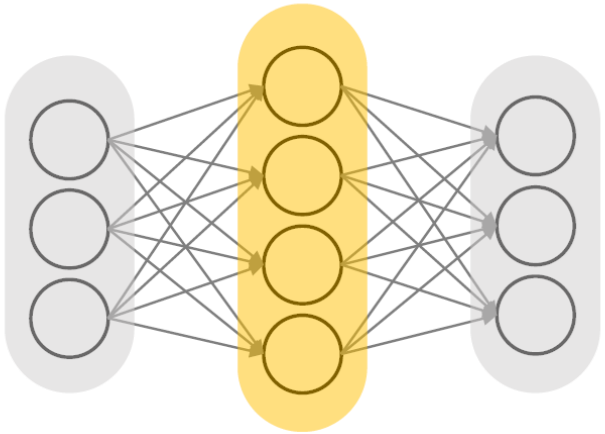
- ANN, Artificial Neural Network
  - Dendrite : Input layer
  - Axon : Output layer
  - Scale of action potential : Weight



인공신경망	인간의 뇌
Σ	뉴런
→	전기·화학적 신호가 흐르는 방향
가중치	신호의 크기

Neural Network

❖ Structure

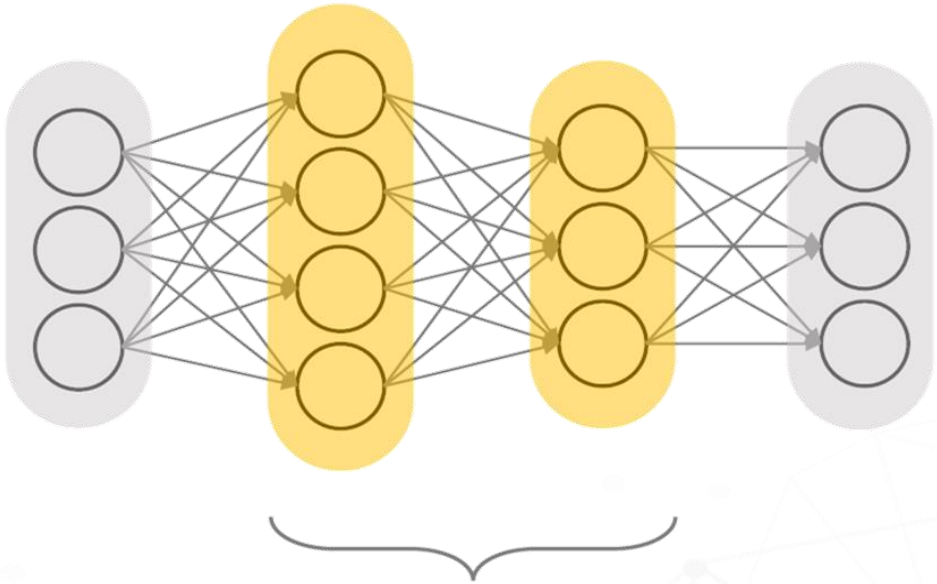


Single-layer perceptron

Deep Neural Network

❖ Structure

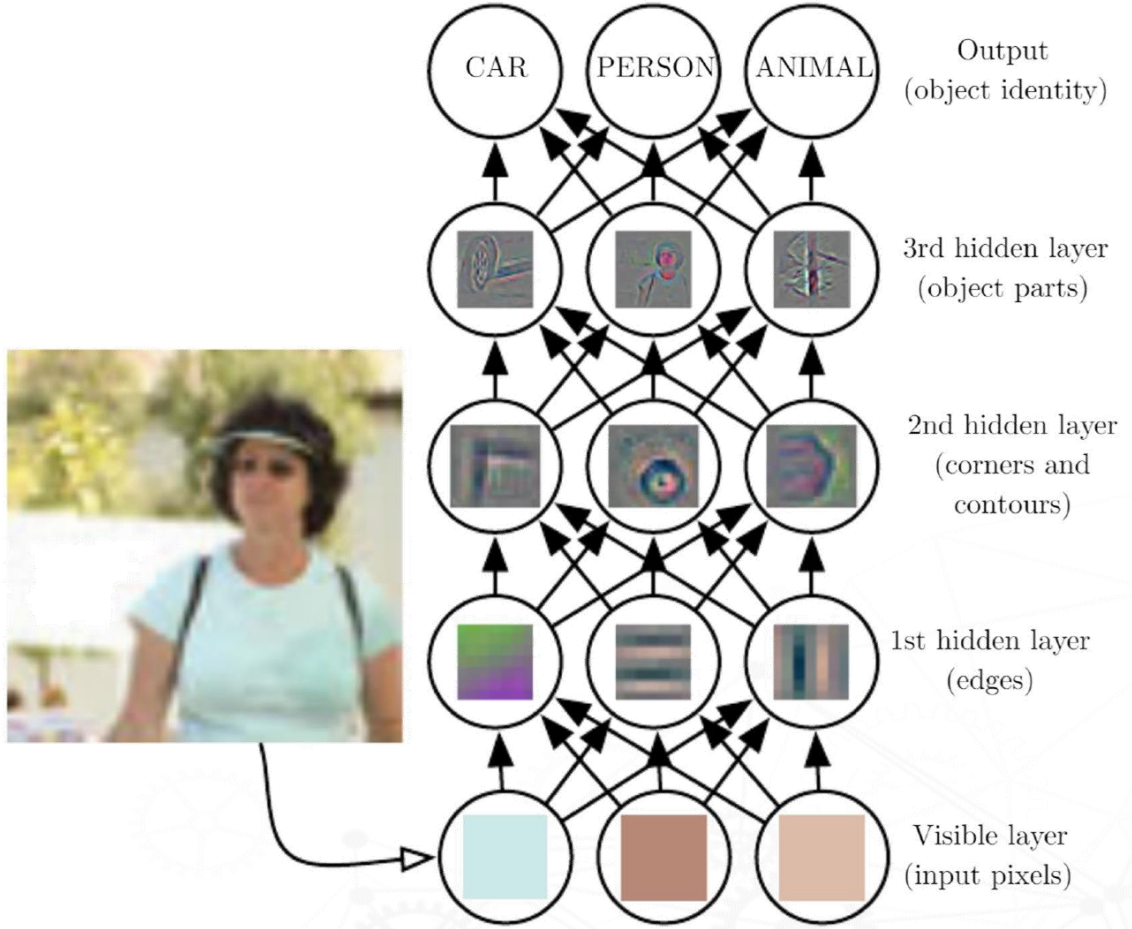
- Two or more hidden layer



Multi-layer perceptron

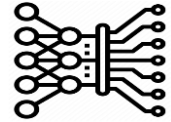
Principle of Deep Neural Network

❖ Analyzing hierarchical high-level features



출처: http://www.deeplearningbook.org, MIT Press, 2016

Kinds of Deep Learning



Neural Networks
신경망 자체

Multi-Layer Perceptron

Deep Neural Networks
- DBN RBM AE
- **CNN**
- RNN

2-Layer Perceptron ~ Regression
- Linear
- Logistic
- Softmax

Reinforcement Learning

GAN

Supervised Learning
Unsupervised Learning

Discriminative Model
Generative Model

Investigators Related to Deep Learning

마이클 조던

제프리 힌톤

요슈아 벤지오

블라디미르 뱁닉

앤드류 응

쥬빈

얀 르쿤

이안 굿펠로우

리차드 서튼

IMAGENET (2009)
http://www.image-net.org

페이페이리

Research Fields

Artificial Intelligence

Data Science

Pattern Recognition

Neural Networks

Machine Learning

Image Processing

Cloud System

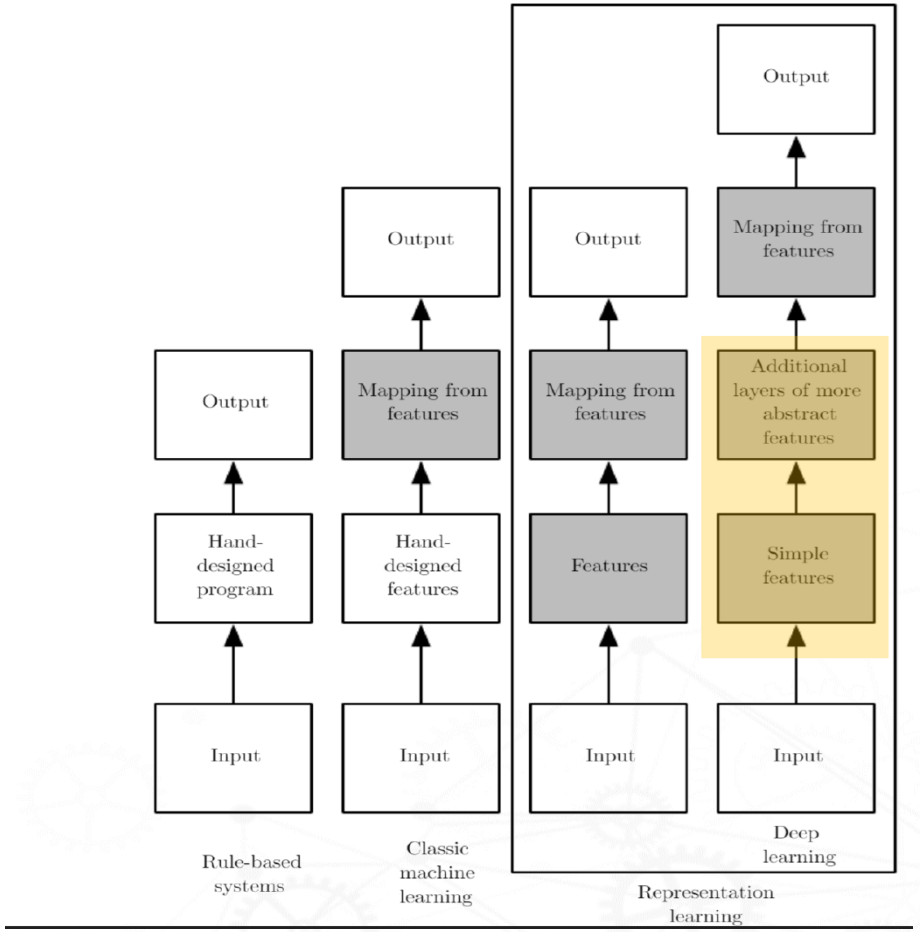
Big Data

Computer Vision

Neural Engineering

Principle of Deep Learning

❖ Neural networks automatically extract features through learning

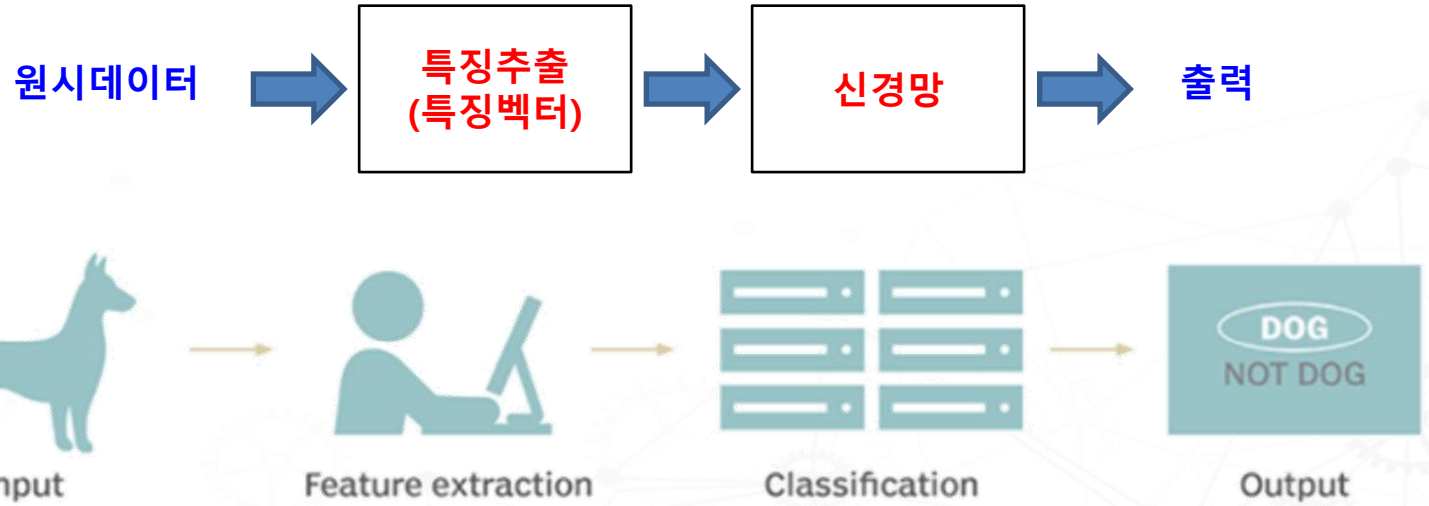


출처: http://www.deeplearningbook.org, MIT Press, 2016

Deep Learning Framework

❖ Comparison between basic NN and deep learning

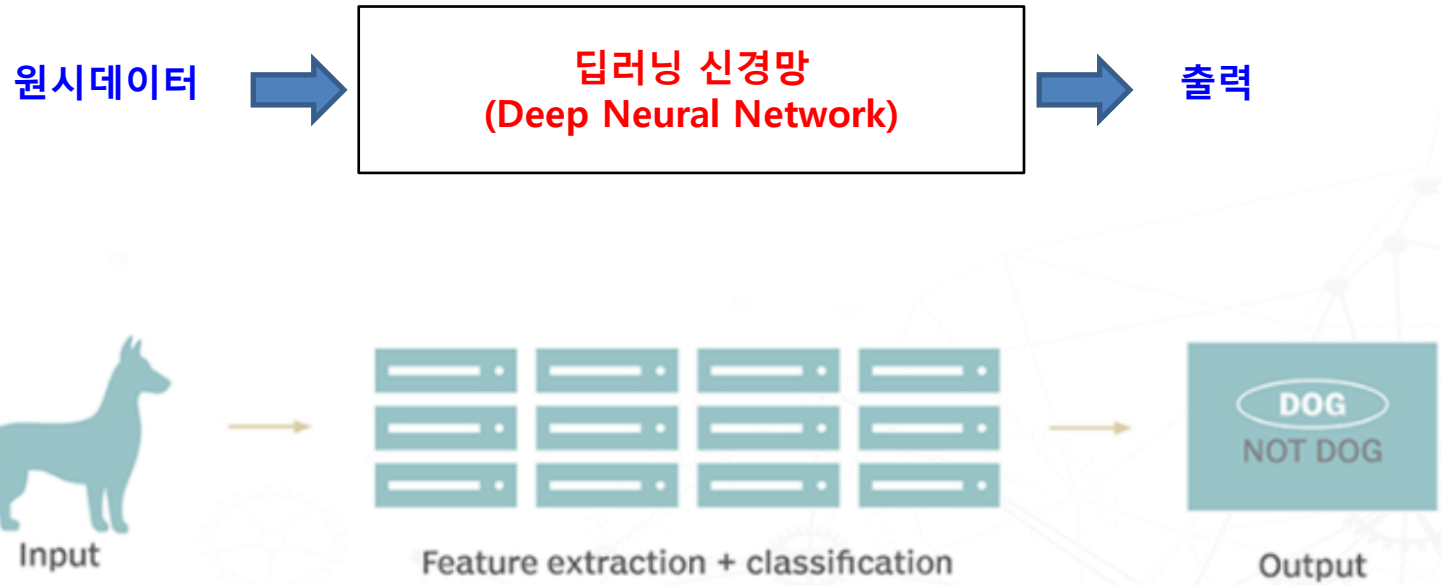
- Basic neural network
  - Use feature vectors created by extracting handcrafted features directly from original data as input
  - Influence the quality of feature vectors



Deep Learning Framework

❖ Comparison between basic NN and deep learning

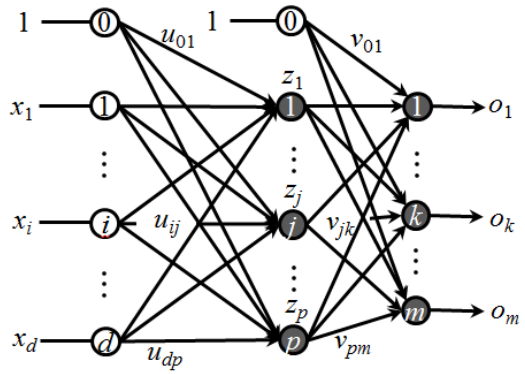
- Deep Learning
  - Feature extraction and trainin simultaneously
  - Extract high-level features from data during training ➔ Superior performance



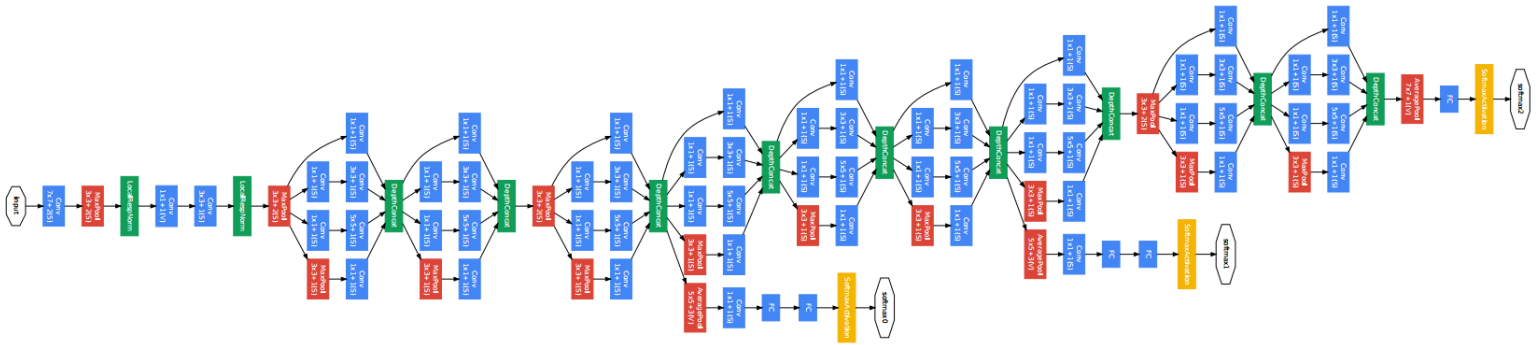
Deep Learning Framework

❖ Comparison between basic NN and deep learning

- Includes a small number of hidden layers



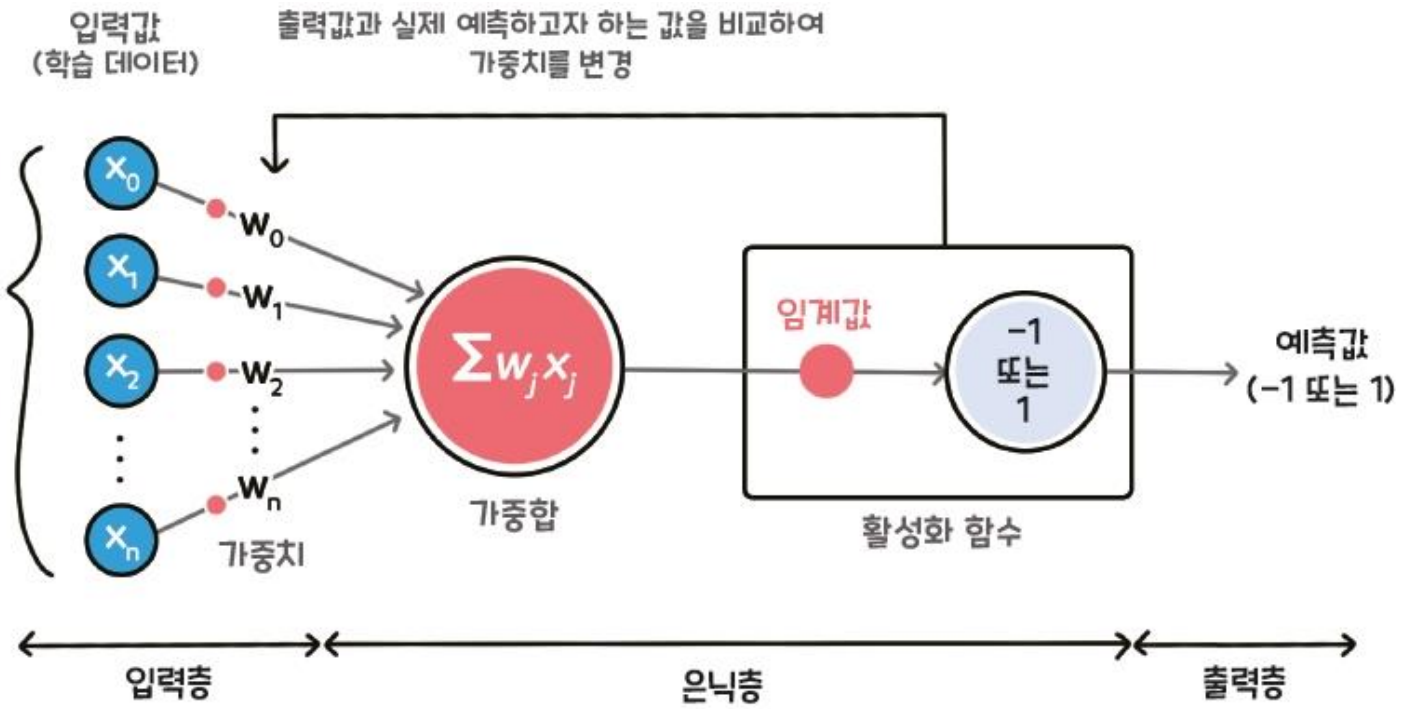
- Includes a large number of hidden layers



출전: googLeNet (google)

Deep Learning Framework

❖ Concepts



Deep Learning Framework

❖ Definition

구분	설명
입력층	학습하고자 하는 데이터를 입력받음
은닉층	모든 입력 노드로부터 입력값을 받아 가중합을 계산하고, 이 값을 활성화 함수에 적용하여 출력층에 전달
출력층	최종 결과 출력
가중치	입력 신호가 출력에 미치는 영향을 조절하는 매개변수로, 입력값의 중요도를 결정
편향	가중합에 더하는 상수로, 하나의 뉴런에서 활성화 함수를 거쳐 최종적으로 출력되는 값을 조절

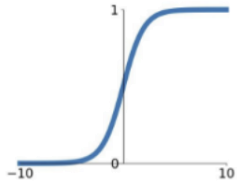
Deep Learning Framework

❖ Activation function

- Activation helps determine whether neurons need to be activated or not
- Function: Mechanisms by which neurons process and transmit information through neural networks

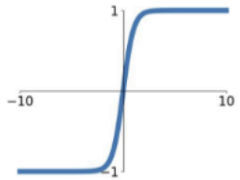
Sigmoid

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



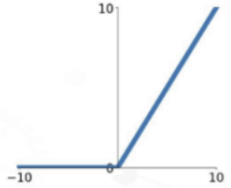
tanh

$\tanh(x)$



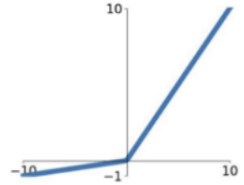
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

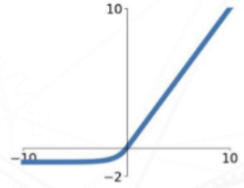


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Deep Learning Framework

❖ Why activation function?

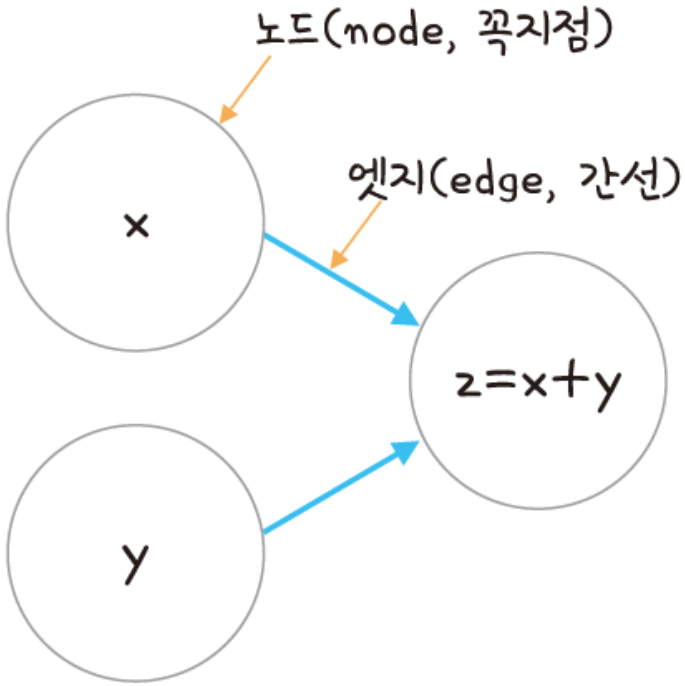
- If the value of the neuron you are trying to convey is available from -∞ to ∞ we cannot determine whether a signal needs to be transmitted to the neuron
- Turn data into non-linear, helping to deepen the network
- Linear systems can be implemented as a single hidden layer even as the network deepens

$$z = \sum(\omega x) + b$$

❖ Backpropagation

Computational graph

- A computational graph is a graph of the calculation process and is expressed in nodes and edges
- The node defines the operation, and the edge indicates the direction in which the data flows

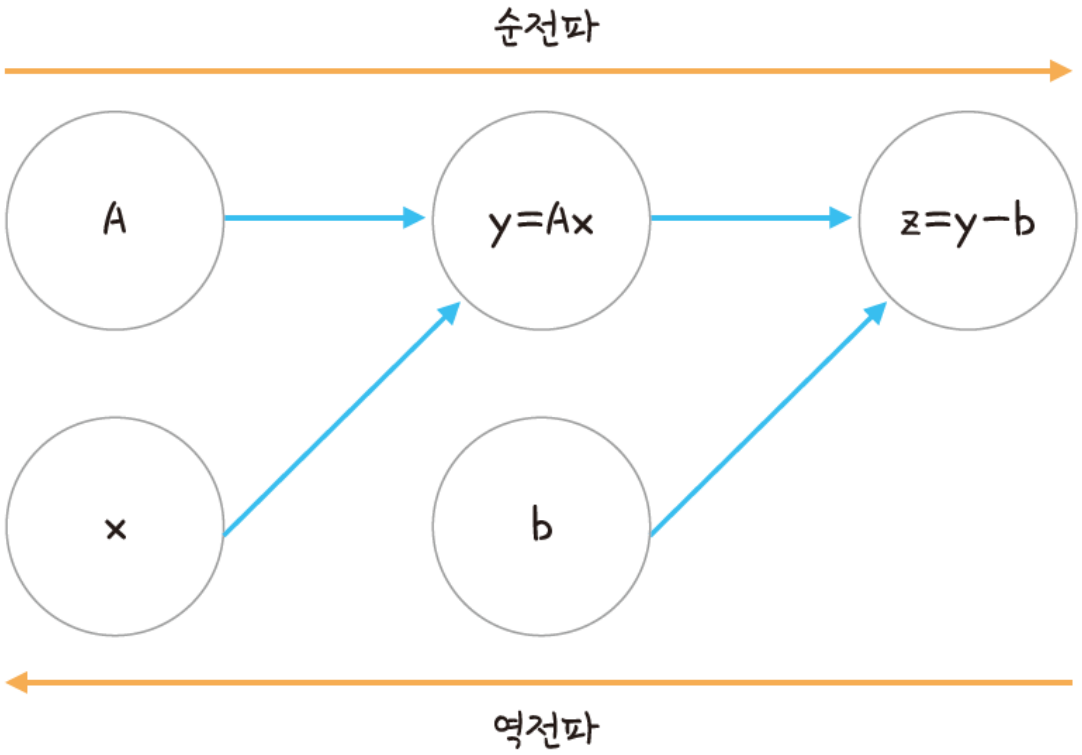


❖ Backpropagation

- The use of computational graphs in the computational process is due to the advantages
  - Local computation can simplify the problem by focusing on the computation of each node
    - Local computation: means calculating only within the scope of a calculation graph that is directly related to you
  - Can efficiently calculate differentials with backpropagation

❖ Backpropagation

- There are two methods for solving computational graphs: forward propagation and backward propagation
- If the calculation graph progresses from left to right, it is called a net wave
- On the other hand, if you go from right to left, it's called backpropagation



❖ Backpropagation

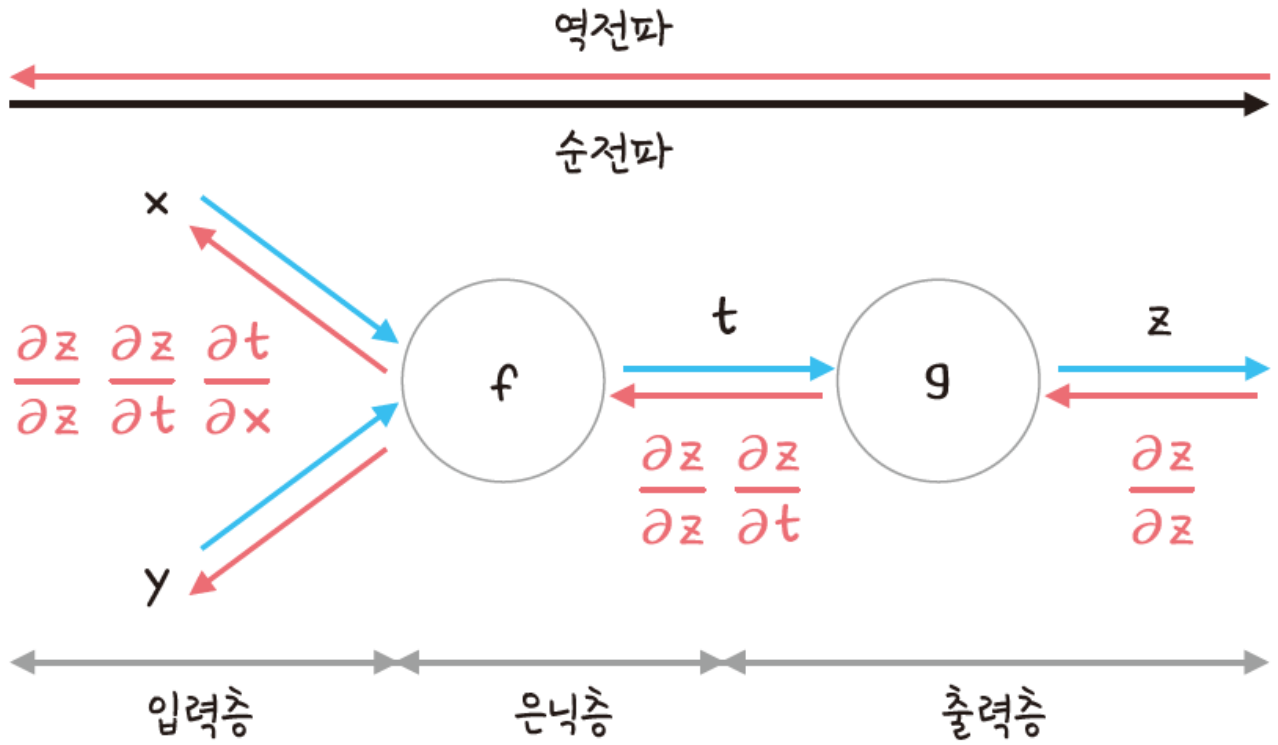
- Backpropagation corrects the **weight and bias of node values involved in this error by obtaining an error between the calculation result and the correct answer**
- The backpropagation is repeatedly corrected **in the process of decreasing the error**
- As the process repeated, the accuracy increases, but there is a disadvantage that it takes a long time
- Fewer times reduce accuracy, but reduce time
- This number of cycles is called an **epoch**
- **Increasing the epoch, updating (learning) weights and biases to reduce errors**

❖ Backpropagation

- The calculation method of error backpropagation is as follows
  - (1) The input value multiplied by the weight and the bias are combined, and if the value exceeds the threshold of 0, 1 is outputted, and otherwise 0 is outputted
  - (2) Obtain the error that is the difference between the output value and the correct answer
    - Correction to a weight value that reduces the error in the reverse direction (how the differentiation method expressed in red obtains the error)
  - (3) Modifies the weight value of the output layer
  - (4) Modify the weight value of the hidden layer

❖ Backpropagation

- Repeat steps (2) through (4) until the error is no longer reduced



❖ Backpropagation

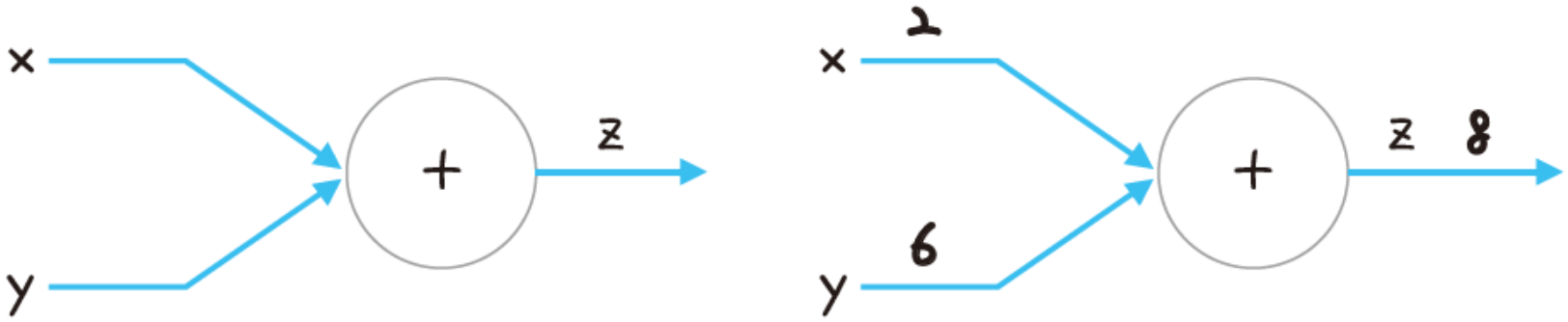
- The backpropagation is said to correct the weight immediately before it in the direction of reducing the error
- When correcting the weight, the partial differential value of y = g(f(x) calculated from the net propagation is multiplied by the error and transmitted to the downstream node (the hidden layer)
- The reason for using **partial differential** at this time is that it does not need to consider all the weight values given to numerous nodes, only the connected weights need to be considered
- This is because using the chain law, even if there are many hidden layers between the output layer and the input layer, the slope can be calculated with a simple differential

❖ Backpropagation

Backpropagation calculation

Node backpropagation of addition

- The calculation graph for the expression z = x + y is as follows
- The forward propagation calculation for z = x + y when x = 2, y = 6 is as follows



❖ Backpropagation

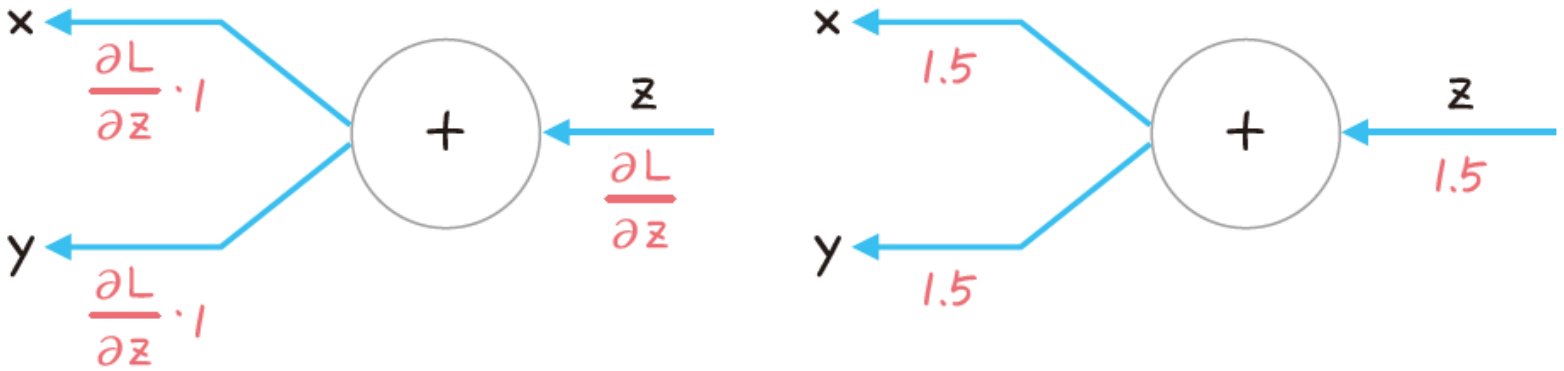
- It's not difficult because you can calculate the forward propagation sequentially
- Backpropagation calculations can be difficult to differentiate
- The derivative of the backpropagation of the addition is

$$z = x + y$$

$$\frac{\partial z}{\partial x} = 1, \frac{\partial z}{\partial y} = 1$$

❖ Backpropagation

- Differentiable (L is considered the final output value)

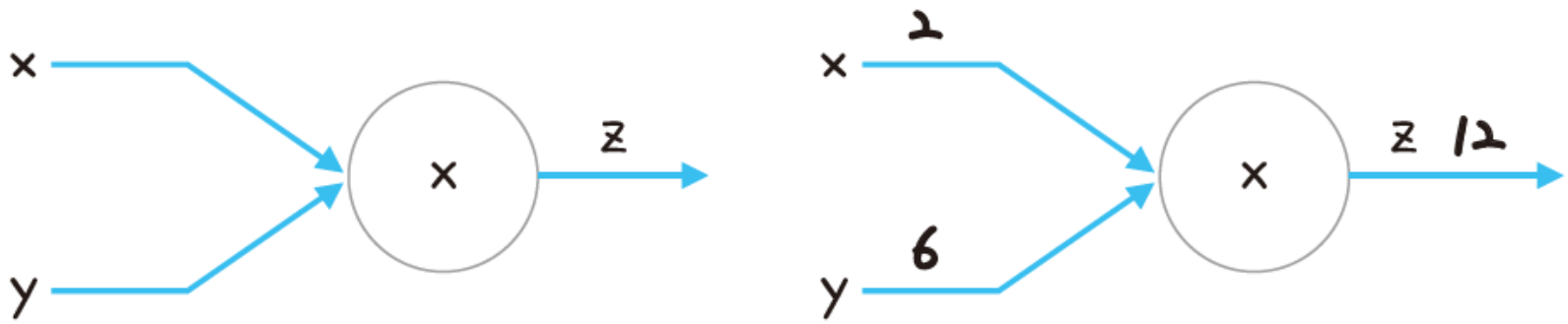


- Assuming that a value of 1.5 is input at upstream (output), the backpropagation of the addition node propagates the input value to the next node

❖ Backpropagation

Node backpropagation of multiplication

- The calculation graph for z = xy expression is shown on the left
- The forward propagation calculation for z = xy when x = 2, y = 6 is equal to the right



❖ Backpropagation

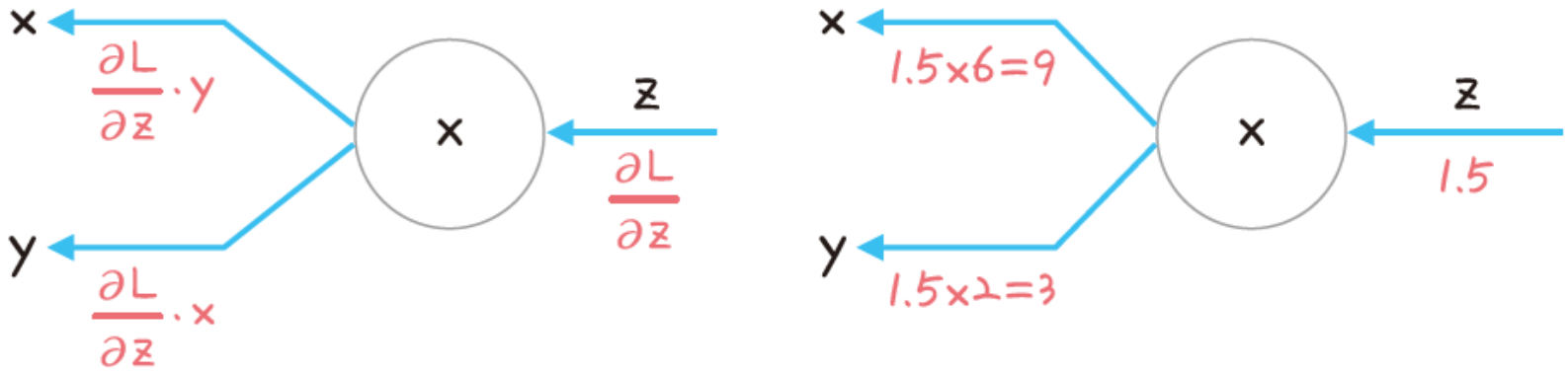
- The forward propagation of multiplication is also not difficult to calculate sequentially, but backpropagation requires knowing how to differentiate
- The derivative of the backpropagation of multiplication is

$$z = xy$$

$$\frac{\partial z}{\partial x} = y, \quad \frac{\partial z}{\partial y} = x$$

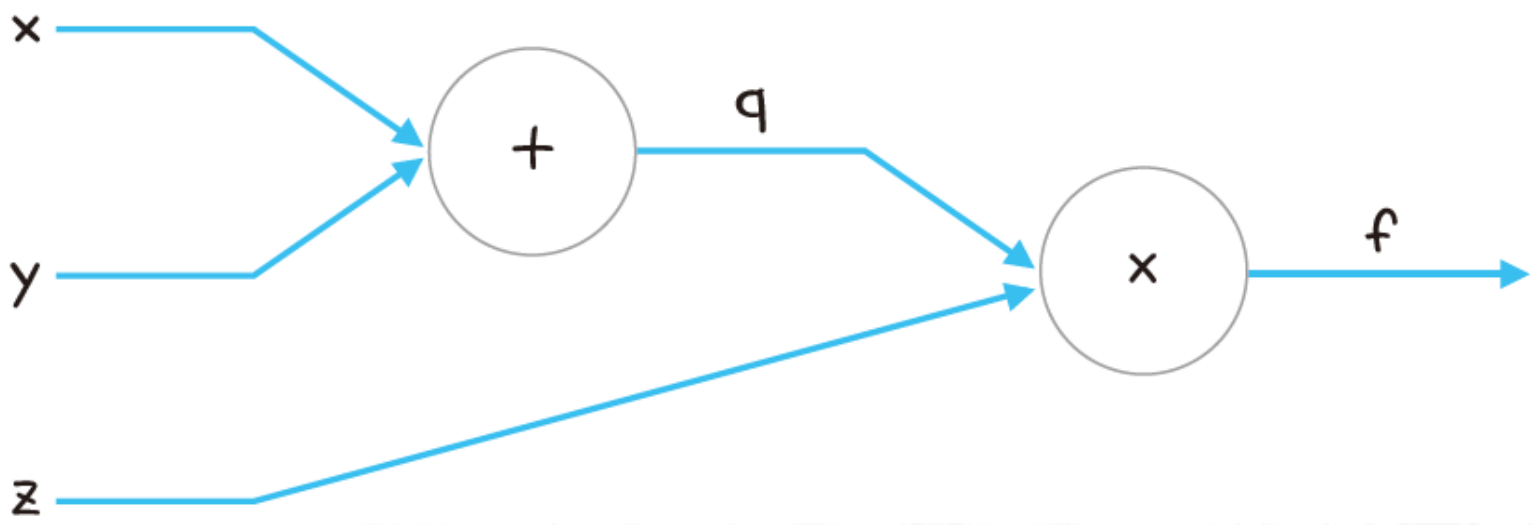
❖ Backpropagation

- Differential is possible as shown on the left
- Assuming that a value of 1.5 is input from the upstream, the backpropagation of the multiplication node can be sent downstream (the hidden layer) by multiplying the upstream (output) value by the input signal of the forward propagation by the 'switched value'



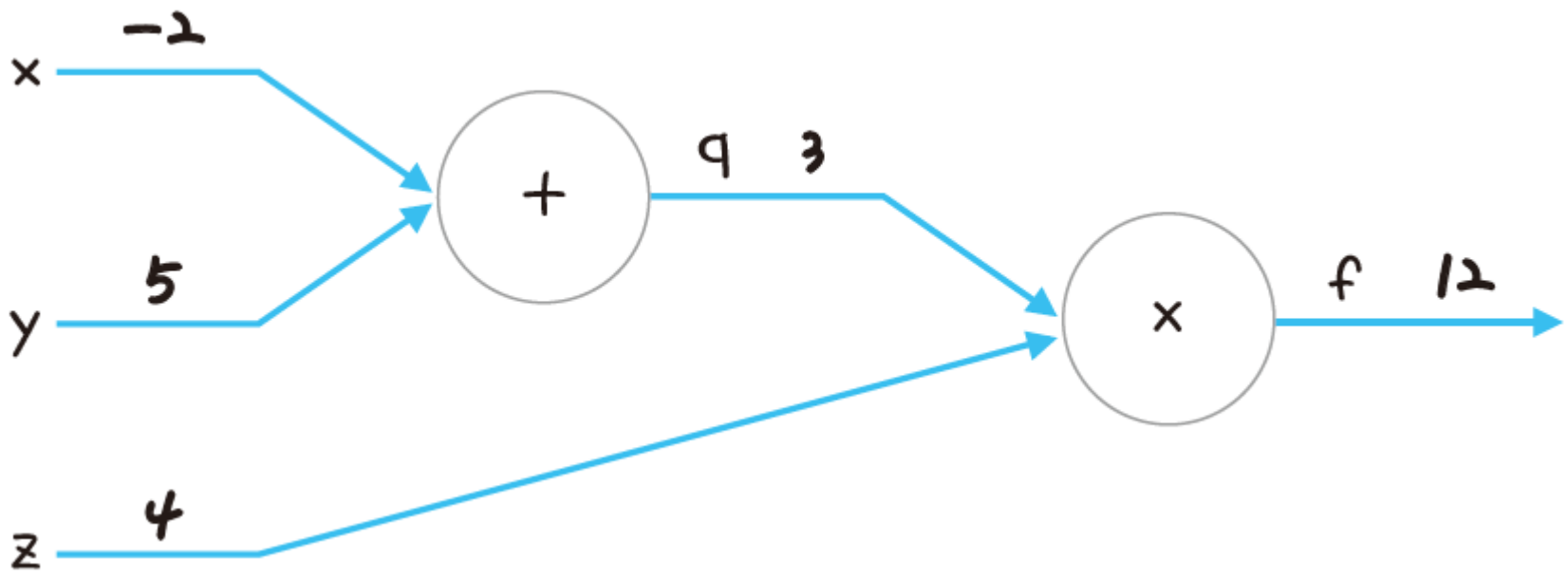
❖ Backpropagation

- Let's combine the backpropagation of addition and multiplication and look at it as a concrete example
- For example, if you have two functions, q(x) = x + y, f(x) = q(x) × z, and you graph them with computational graphs



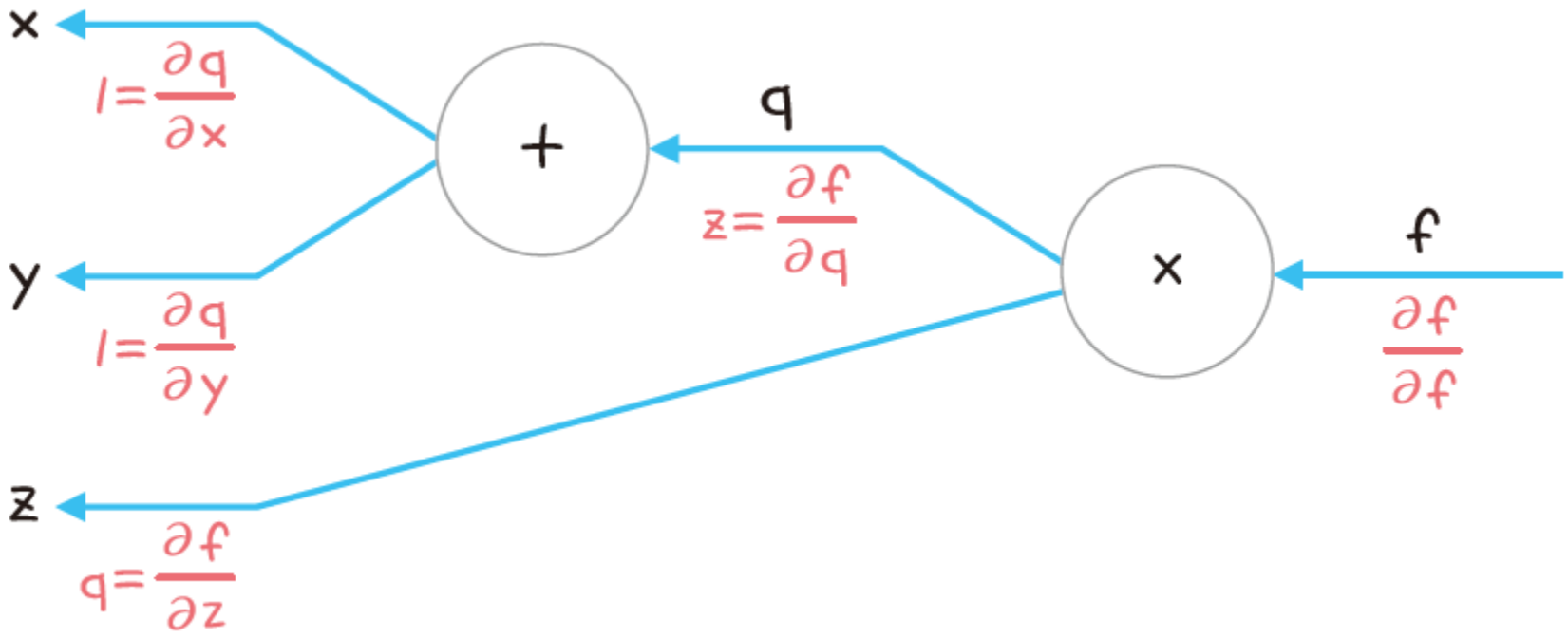
❖ Backpropagation

- If the forward calculation is performed when x = -2, y = 5, z = 4, the result is 12 as follows



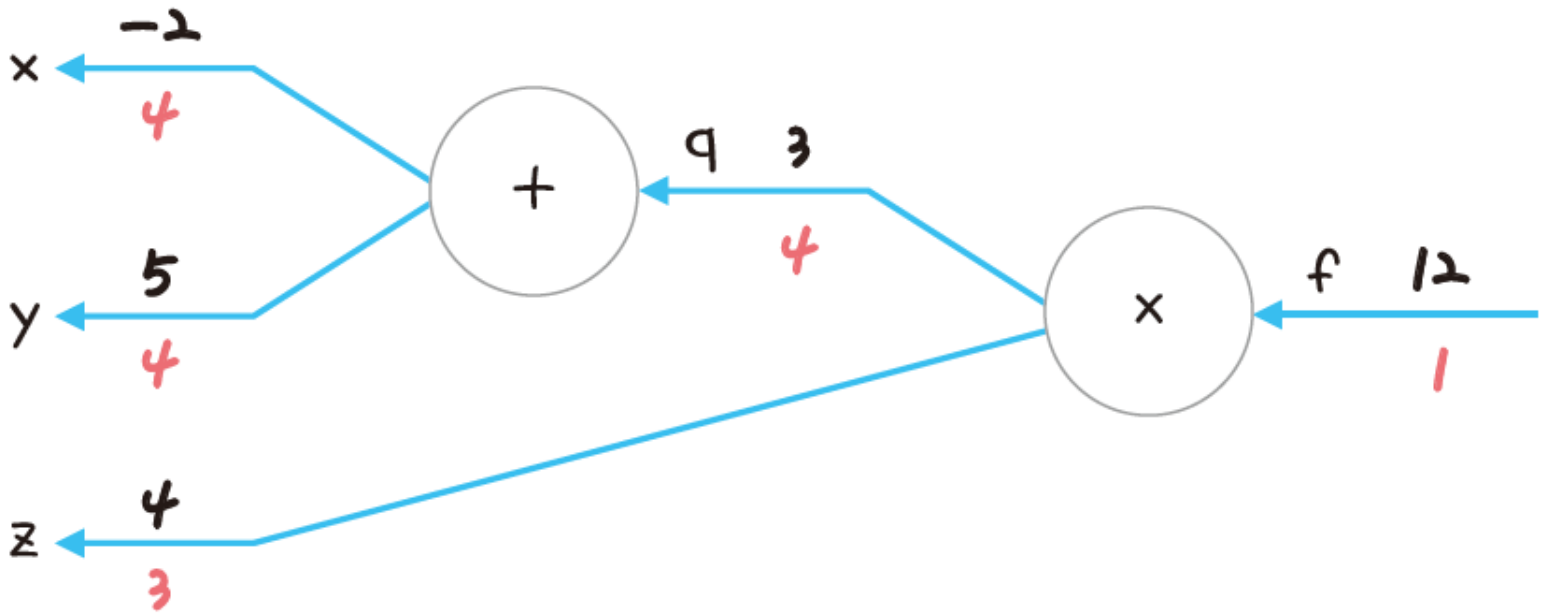
❖ Backpropagation

- If differential is applied to calculate the backpropagation, it can be differentiated as follows



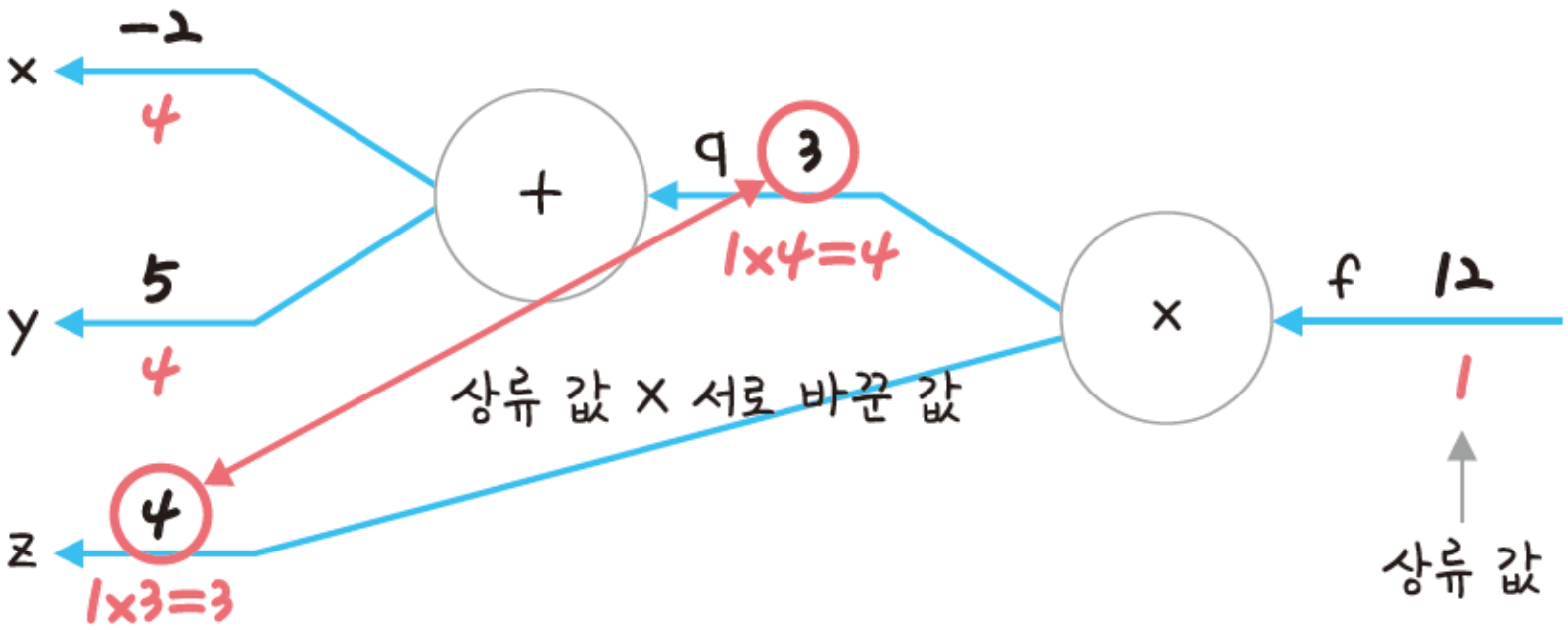
❖ Backpropagation

- In other words, for backpropagation of multiplication, the upstream (output) value (starting from 1) is multiplied by the input signal of the net propagation by the 'switched value' and sent downstream (the hidden layer)
- If you send the input value downstream (input) for backpropagation of the addition, the following results are obtained



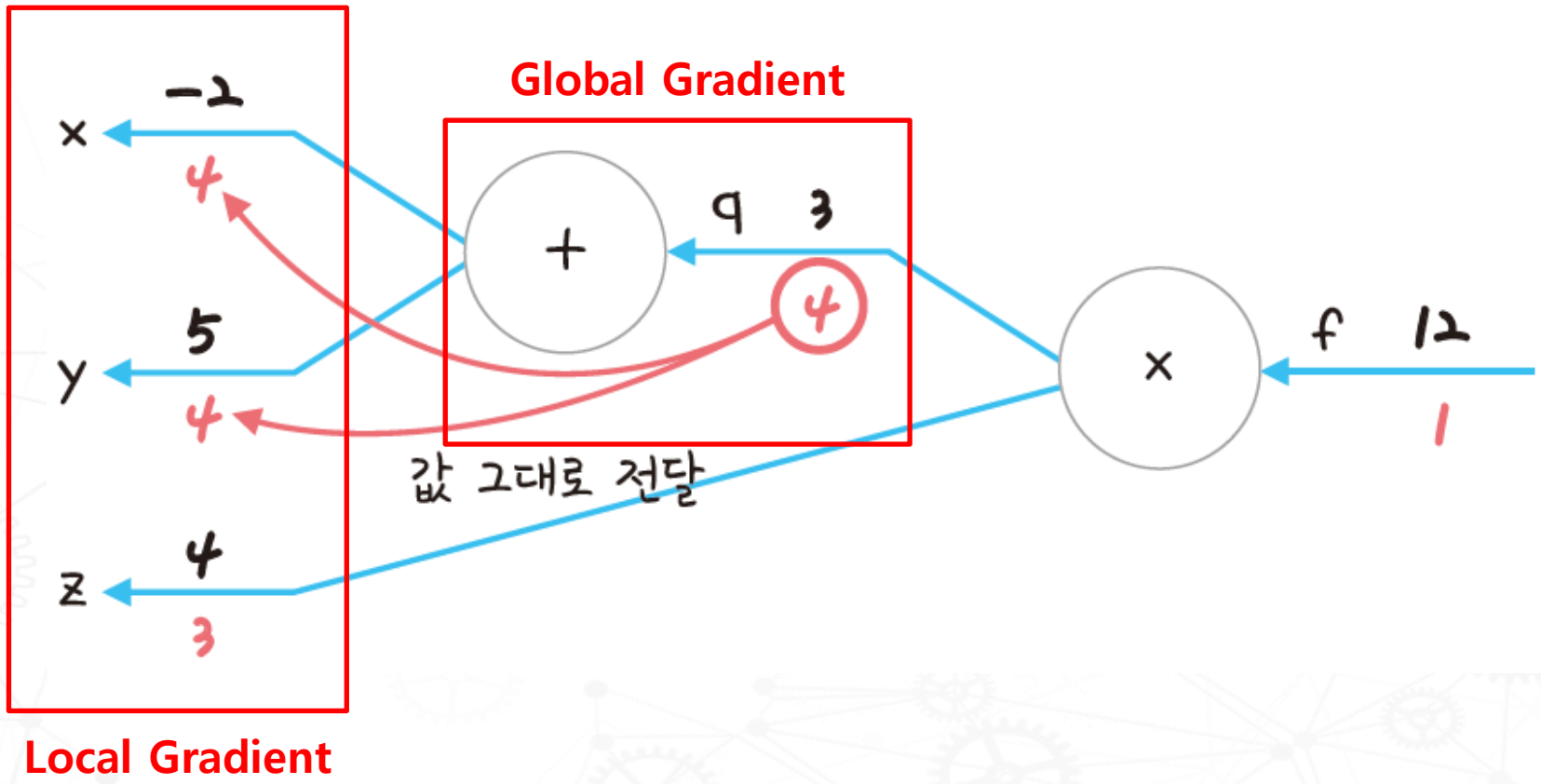
❖ Backpropagation

- If you look at the previous results in detail, first, the backpropagation for multiplication is as follows
- You can replace the q-value with the z-value and multiply it with the upstream value



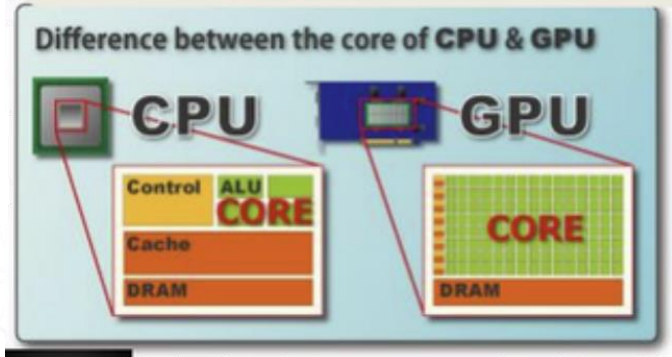
❖ Backpropagation

- The backpropagation for addition is



Reasons for Deep Learning Advances

❖ GPU enables faster parallel computation than CPU



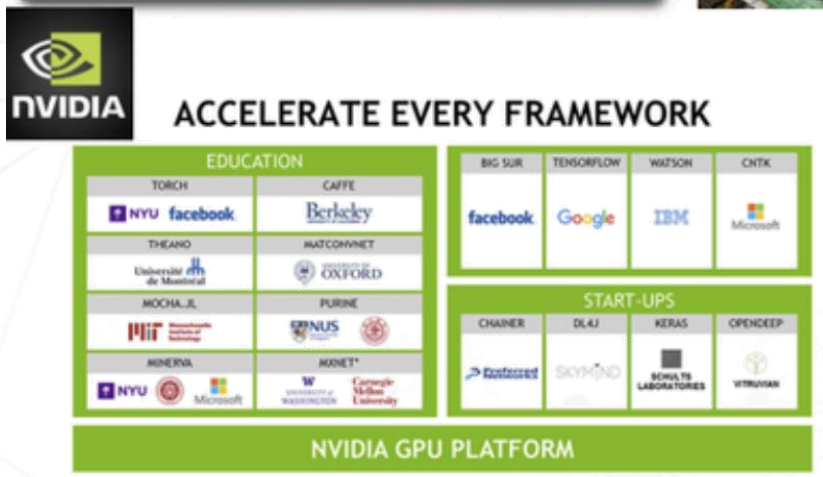
Large-scale Deep Unsupervised Learning using Graphics Processors

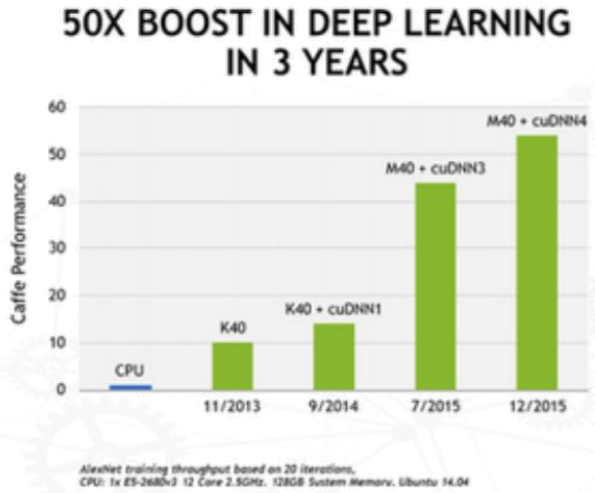
ICML, 2009

Rajat Raina
Anand Madhavan
Andrew Y. Ng
Computer Science Department, Stanford University, Stanford CA 94305 USA

Andrew Ng.

딥러닝 모델 학습에 대규모로 GPU를 적용하여
획기적 속도 향상됨을 보임
(DBN 알고리즘의 경우 70배)



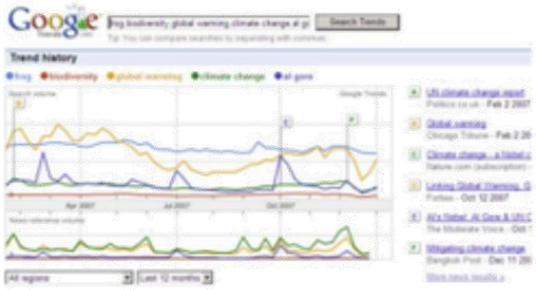


Reasons for Deep Learning Advances

- Big data analytics and processing speed
  - Create a smarter experience for the entire digital space

대용량 DB
대용량 저장 장치

클라우드 컴퓨팅



이십여 년 이상 누적된
전세계 검색 기록

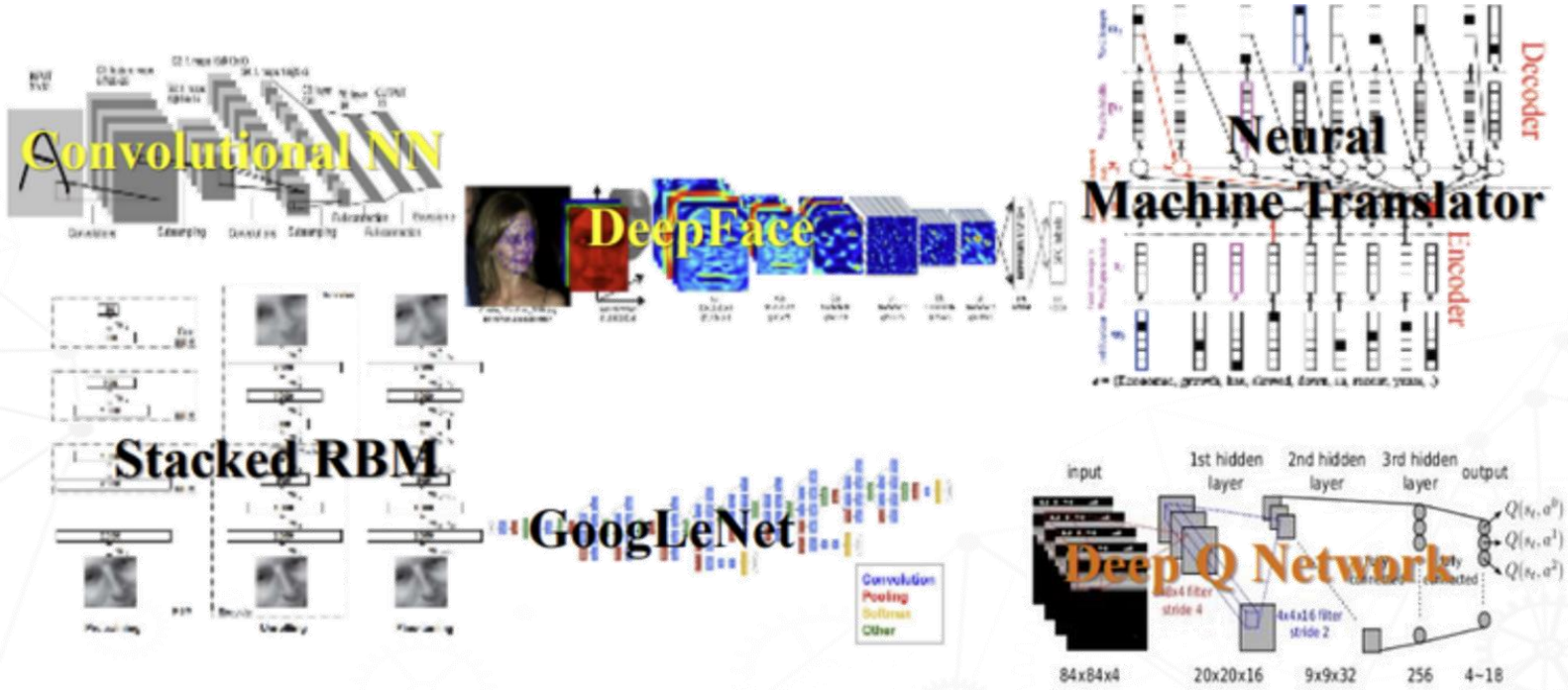
인터넷 사용 기록
(web cookies, online footprints, …)

사물인터넷

Wikipedia 등의 무료 정보원

Reasons for Deep Learning Advances

- Evolution of machine learning algorithms



Reasons for Deep Learning Advances

- Advancing of development environment
  - CLOUD SERVICE + MACHINE LEARNING/DEEP LEARNING

GitHub

Microsoft CNTK https://github.com/Microsoft/CNTK

TensorFlow https://github.com/tensorflow/tensorflow

theano https://github.com/Theano/Theano

torch https://github.com/torch/torch7

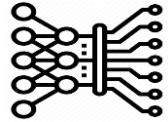
Caffe https://github.com/BVLC/caffe

Reasons for Deep Learning Advances

- An academic case of innovation
  - Convolutional neural networks open the door to deep learning
    - Extract superior features using small-sized convolutional masks
    - In the 1990s, LeKun dramatically improved the performance of handwritten numbers (automatic check recognition system)
  - AlexNet shows that natural image recognition is possible with a convolutional neural network
    - Won the 2012 ILSVRC competition with a phenomenal 15.3% error rate at the time
    - Since then, computer vision research has shifted from classical machine learning to deep learning

  - Transforming in Speech Recognition
    - Professor Hinton applied deep learning to reduce the error rate by 20% at once
    - "I accomplished what took 10 years at once. … Ten technological innovations took place all at once," he said

Reasons for Deep Learning Advances

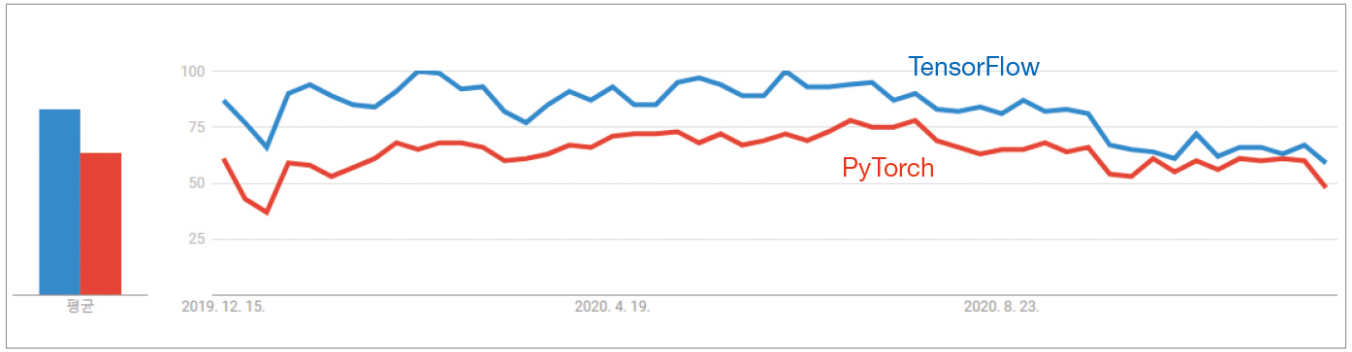


NOTE ImageNet과 ILSVRC 대회

자연 영상을 분류하는 문제는 ImageNet이라는 데이터베이스가 만들어진 이후에 다시 주목받기 시작했다. ImageNet은 WordNet의 계층적 단어 분류 체계에 따라 부류를 정하고 약 2만 부류에 대해 각각 500~1,000장의 영상을 인터넷에서 수집해 구축하였다[Deng2009]. ILSVRC는 ImageNet에서 1,000개의 부류를 뽑아 분류 문제를 푸는 대회이다[Russakovsky2015]. 총 120만 장의 훈련 집합, 5만 장의 검증 집합, 15만 장의 테스트 집합이 주어진다. [그림 5-2]는 1,000개의 부류 중 'cat' 부류의 예제 영상인데, 같은 부류 안에서 변화가 아주 심하다는 것을 확인할 수 있다. 만일 신경망이 cat 부류에 속하는 영상을 보고 'cat(0.9), dog(0.1), …'이라고 출력하면 1순위로 맞힌 것으로 간주한다. 괄호 속 숫자는 해당 부류에 속할 확률이다. 그리고 만약 'dog(0.5), bear(0.3), swing(0.1), cat(0.09), abacus(0.02), Great white shark(0.01), …'이라고 출력하면 정답 부류가 5순위 안에 있으므로 5순위로 맞힌 것으로 간주한다. ILSVRC는 1순위 오류율과 5순위 오류율 두 가지로 성능을 측정한다. 2012년에 AlexNet은 5순위 오류율 15.3%를 달성했다. 그리고 2015년에는 마이크로소프트 팀에서 지름길 연결이라는 아이디어를 구현한 ResNet이 3.5%의 5순위 오류율로 우승했다.

Deep Learning Software

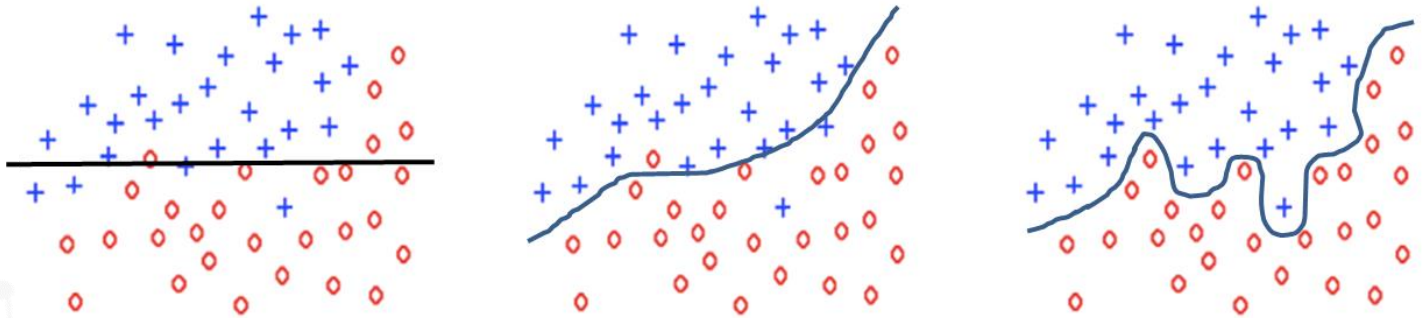
이름	개발 그룹	최초 공개일	작성 언어	인터페이스 언어	전이학습 지원	철저한 관리
씨아노(Theano)	몬트리올 대학교	2007년	파이썬	파이썬	O	X
카페(Caffe)	UC버클리	2013년	C++	파이썬, 매트랩, C++	O	X
텐서플로(TensorFlow)	구글 브레인	2015년	C++, 파이썬, CUDA	파이썬, C++, 자바, 자바스크립트, R, Julia, Swift, Go	O	O
케라스(Keras)	프랑소와 숄레(François Chollet)	2015년	파이썬	파이썬, R	O	O
파이토치(PyTorch)	페이스북	2016년	C++, 파이썬, CUDA	파이썬, C++	O	O



Limitations of Deep Learning

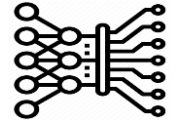
❖ Overfitting

- Model over-aligned to learning data
- Data contains noise or errors
  - Overfitting models degrade on unlearned data



- Solving method
  - Regularization
  - Batch normalization
  - Dropout

Limitations of Deep Learning



❖ Regularization

- Define an error function as an error term and a model complexity terms
- Add model complexity as penalty term because complexity can overfit the model

Cost function = (cost) +α (model complexcity)

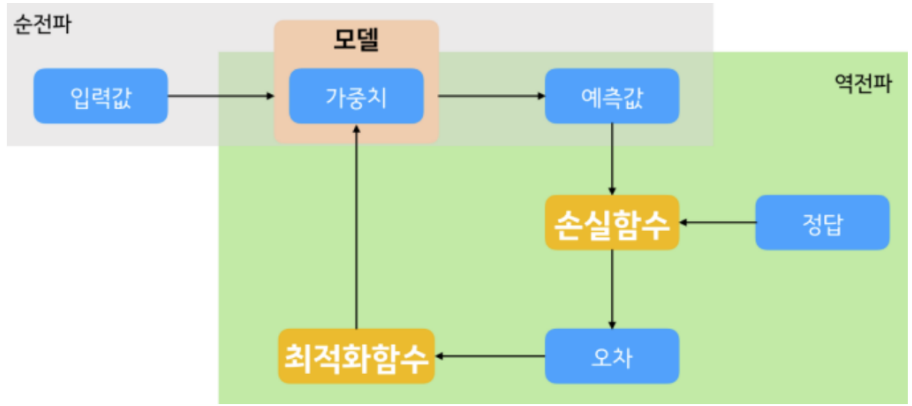
α : parameter ratio

Limitations of Deep Learning

❖ Minibatch

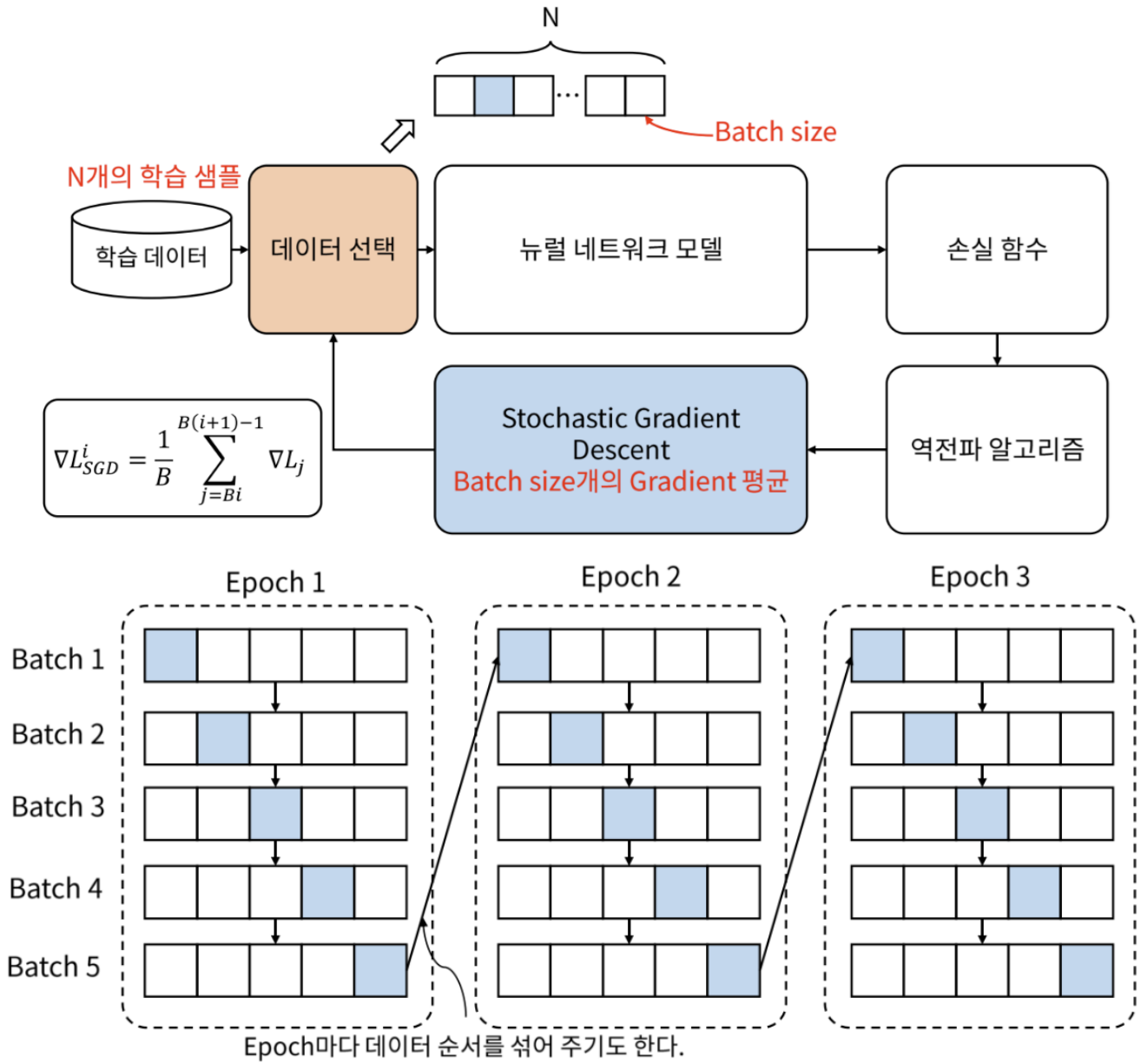
- Divided the entire learning data into a certain size (N-division)
- Learn in mini-batch units when learning data is large
  - Learning large-sized data at once takes a long time
- Gradient of minibatch when applying gradient-descent method
  - Average use of gradients for each data in a mini-batch

$$\nabla g = \frac{1}{10}\sum_{i=1}^{10}\nabla g_i$$



- Mini-batch enables to learn insensitively about the errors contained in the data

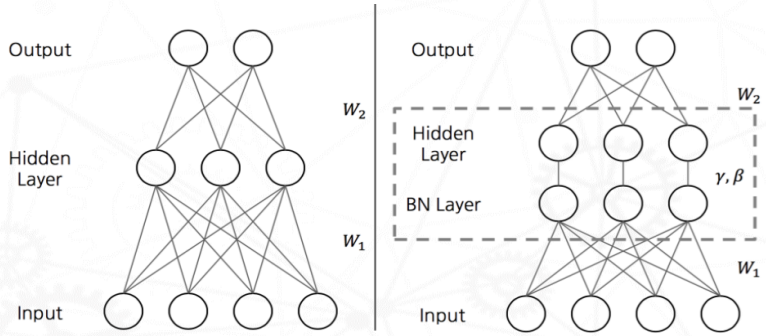
Limitations of Deep Learning



Limitations of Deep Learning

❖ Batch normalization

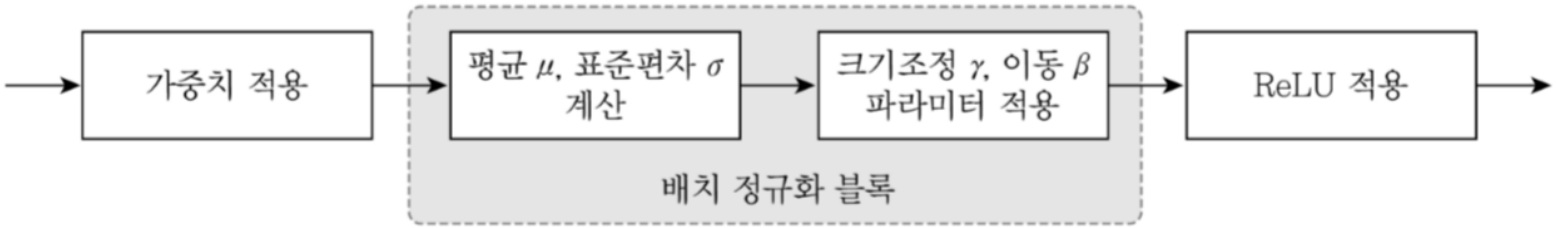
- Normalizing the distribution of data batches at each layer of the neural network
  - Deep neural networks, even with the same input value, can yield completely different output values with a slight difference in weight
- Advantages
  - Less dependent on weight initial value selection
  - Normalizes output values each time you learn
  - Reduce the risk of overfitting
  - Troubleshooting slope extinction
  - Improve learning speed



Limitations of Deep Learning

❖ Batch normalization

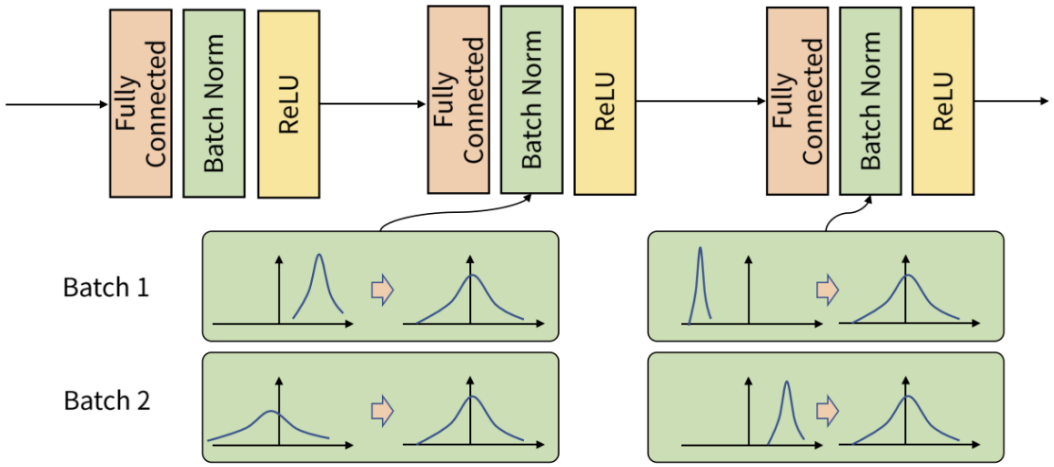
- Normalizing the distribution of x_i at each layer of the neural network, which is the result of applying weight operations to each data in the mini-batch B
  1. x_i의 평균 μ_B가 0이 되고 표준편차 σ_B는 1가 되도록 변환
  2. 크기조정(scaling) 파라미터 γ와 이동(shift) 파라미터 β 적용
  3. 변환된 데이터 y_i 생성



Limitations of Deep Learning

❖ Batch normalization

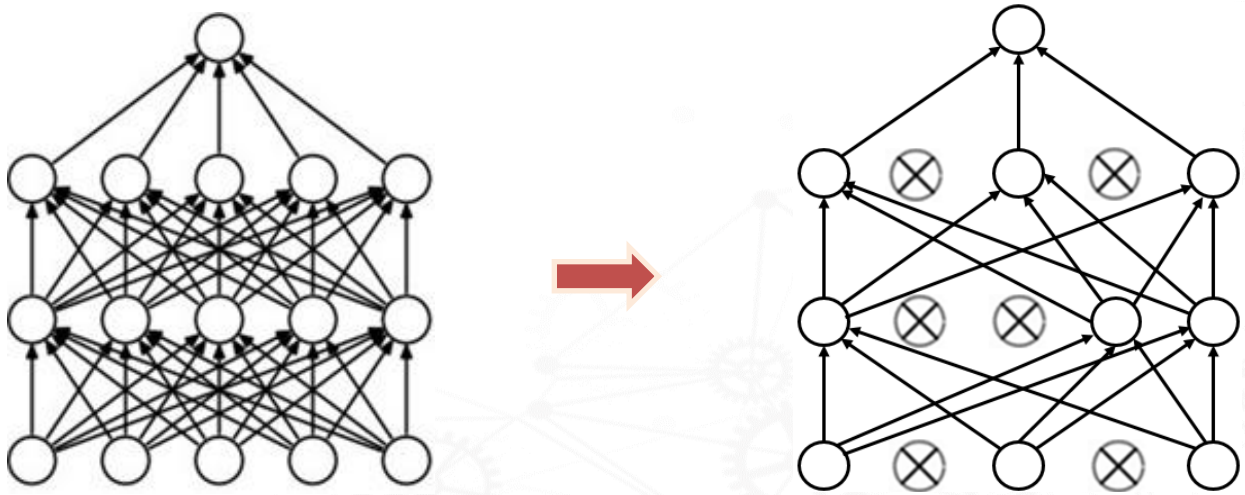
- Normalizing the distribution of x_i at each layer of the neural network, which is the result of applying weight operations to each data in the mini-batch B
  1. x_i의 평균 μ_B가 0이 되고 표준편차 σ_B는 1가 되도록 변환
  2. 크기조정(scaling) 파라미터 γ와 이동(shift) 파라미터 β 적용
  3. 변환된 데이터 y_i 생성



Limitations of Deep Learning

❖ Dropout

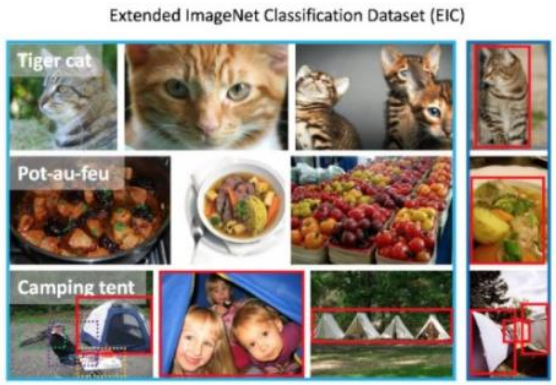
- Select nodes at random with a certain probability, and learn by considering that there are no weight connections connected back and forth of the selected node
- Drop out every mini-batch or learning cycle (epoch),
  - i.e. select and learn new nodes to consider as absent



Deep Learning Advances

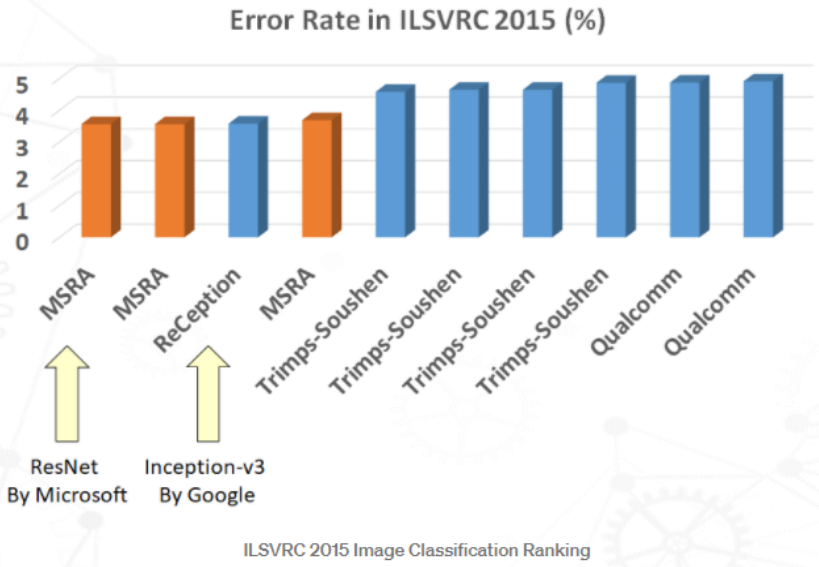
- ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is a competition that evaluates object recognition and image classification algorithms on a large scale
  - More than 14 million images are manually labeled (supervised) to indicate which objects are displayed
  - Approximately 1 million+ images also provide rectangular areas

  Evaluate performance with top-5 errors



Deep Learning Advances

- ResNet, developed by MS Labs for the first time in 2015, announced a higher image recognition rate than humans
  - Classic CNNs performed poorly as the depth of the network exceeded certain limits
  - Millions of parameters can be stored in hidden layers

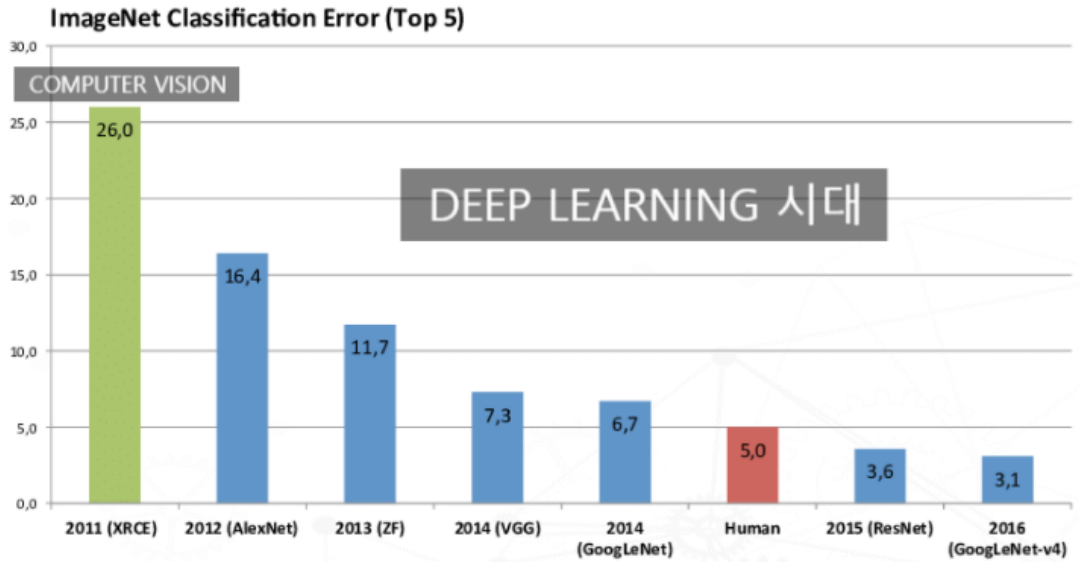


method	top-5 err. (**test**)
VGG [40] (ILSVRC'14)	7.32
GoogLeNet [43] (ILSVRC'14)	6.66
VGG [40] (v5)	6.8
PReLU-net [12]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Deep Learning Advances

❖ ResNet, developed by MS Labs for the first time in the world in 2015, announced a higher image recognition rate than humans

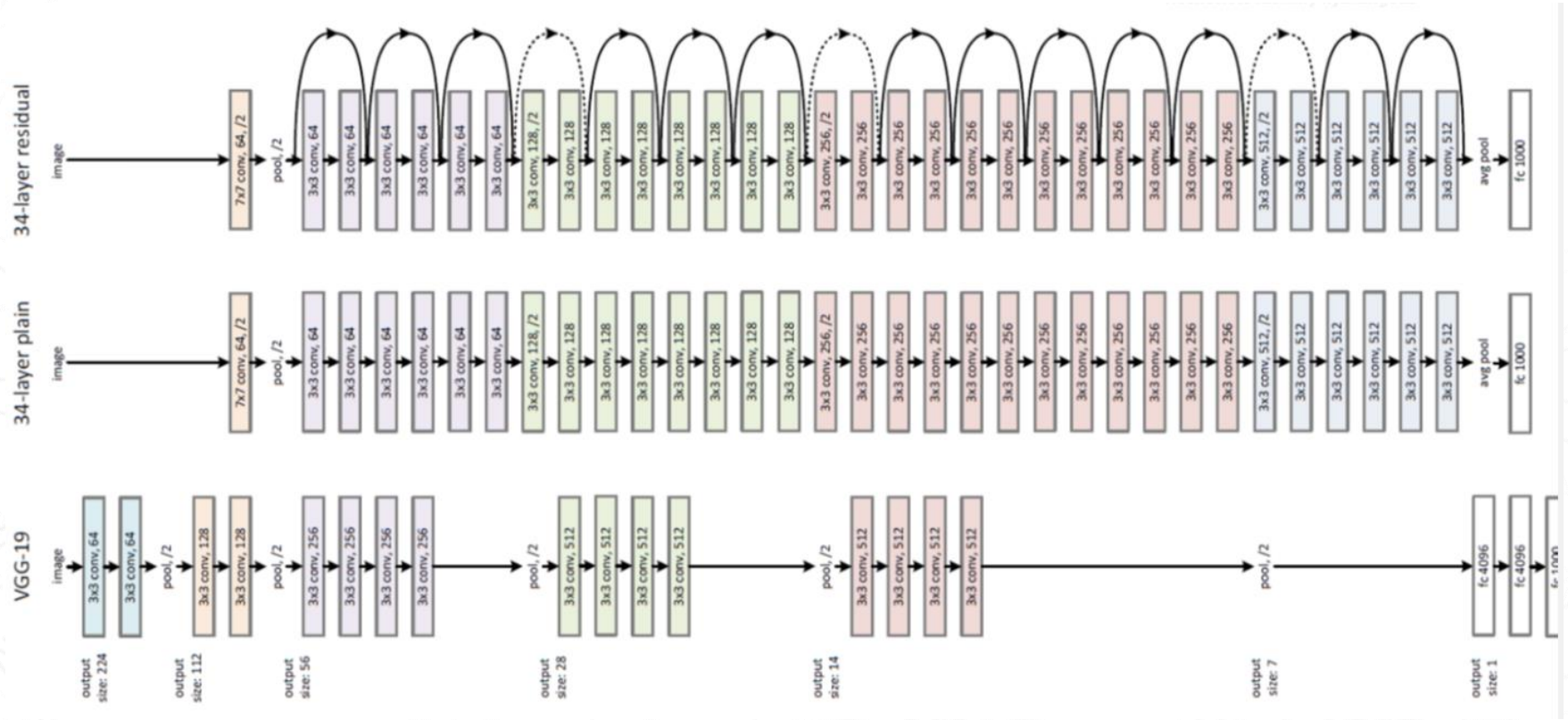
- Basic CNNs performed poorly as the depth of the network exceeded certain limits
- Millions of parameters can be stored in hidden layers



이미지넷 경진대회 역대 우승 아키텍처 모델

Deep Learning Advances

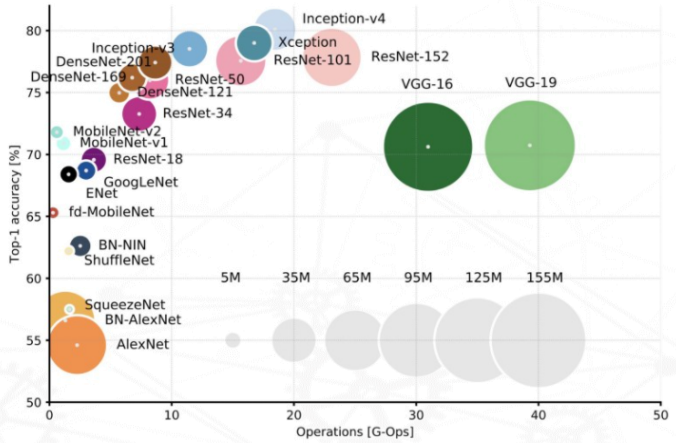
❖ **ResNet**



Deep Learning Architecture

❖ Focus on optimization, accuracy and efficiency

- Accuracy
  - The number of Hyperparameters embedded in the hidden layers of deep learning is important and needs to be designed to learn and evaluate them efficiently
  - The most important thing in deep learning design is to efficiently distribute and design more hidden layers, resulting in more time to evaluate and learn

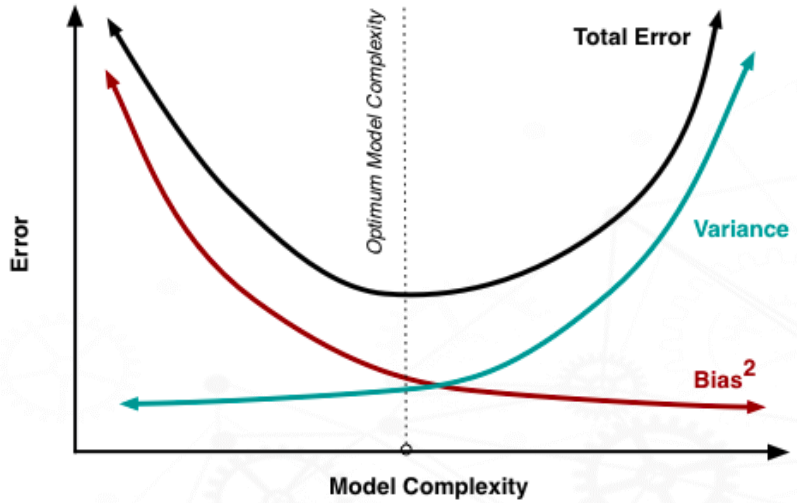


딥러닝 파라매터수와 인식율의 관계 그래프

Deep Learning Architecture

❖ Focus on optimization, accuracy and efficiency

- Computation power
  - Deep learning has significant memory and computing requirements during training
  - Even if you want to deploy models to run locally on your mobile, you need to consider the size of the final learned model



딥러닝 파라매터수와 에러율의 및 성능 지표

Deep Learning Architecture

- Deep feedforward network
- Recurrent neural network
- Long short-term memory
- Convolutional neural network
- Generative adversarial network
- Transformer network
- Attention network