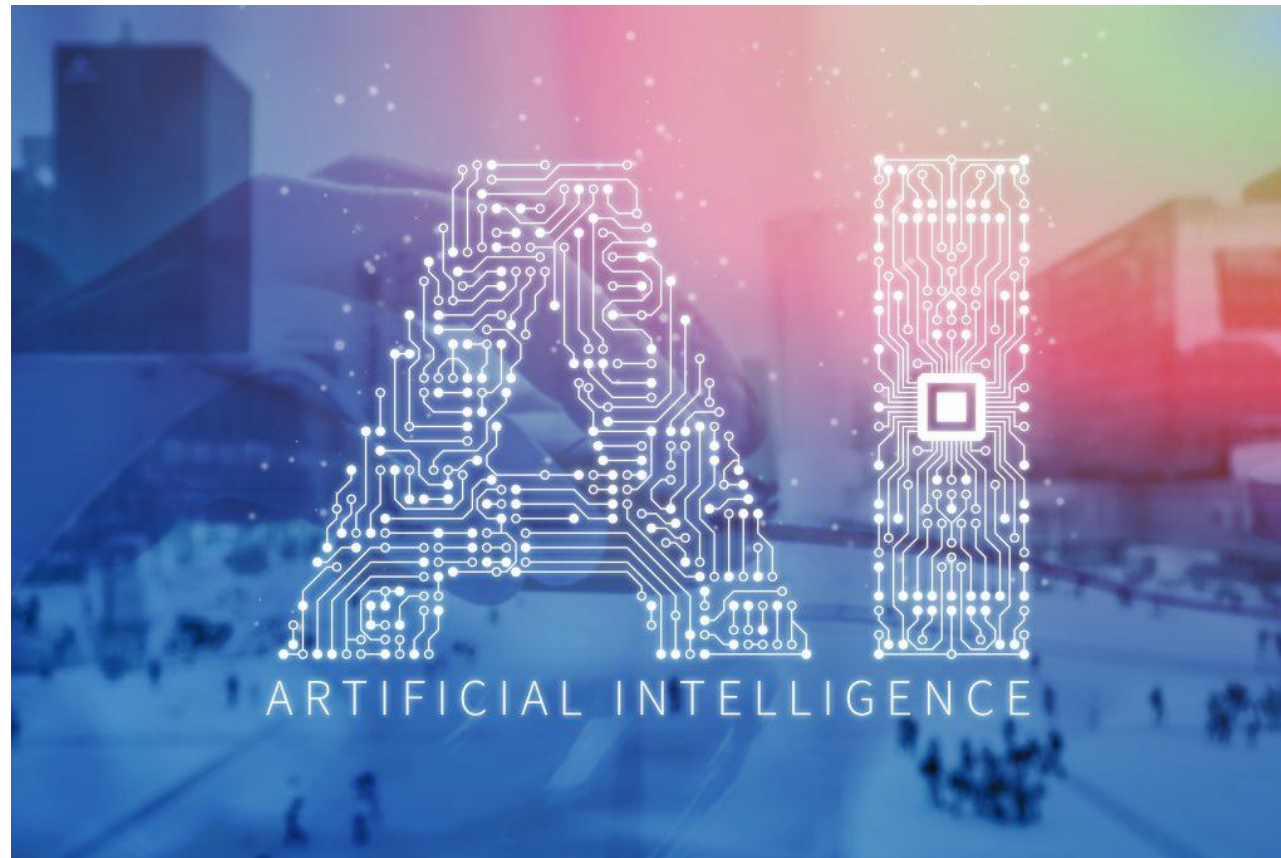
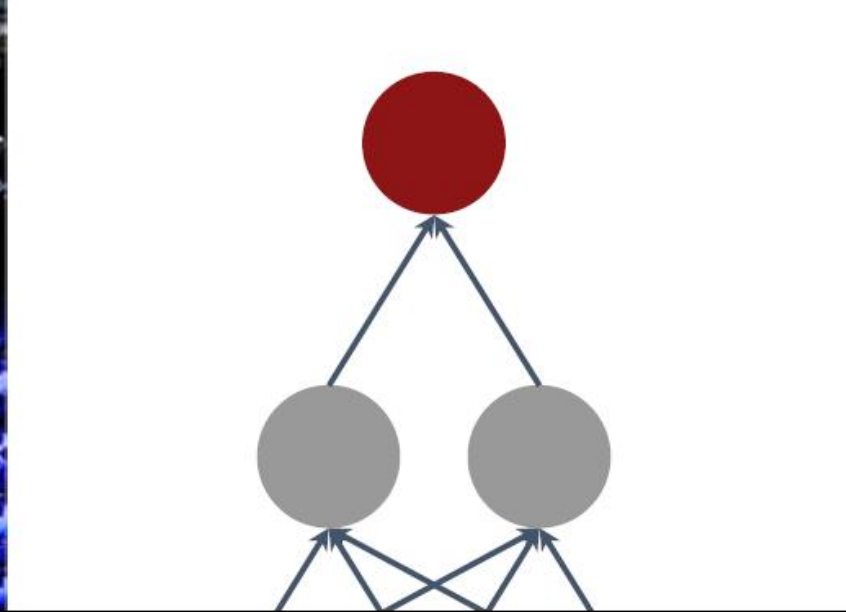


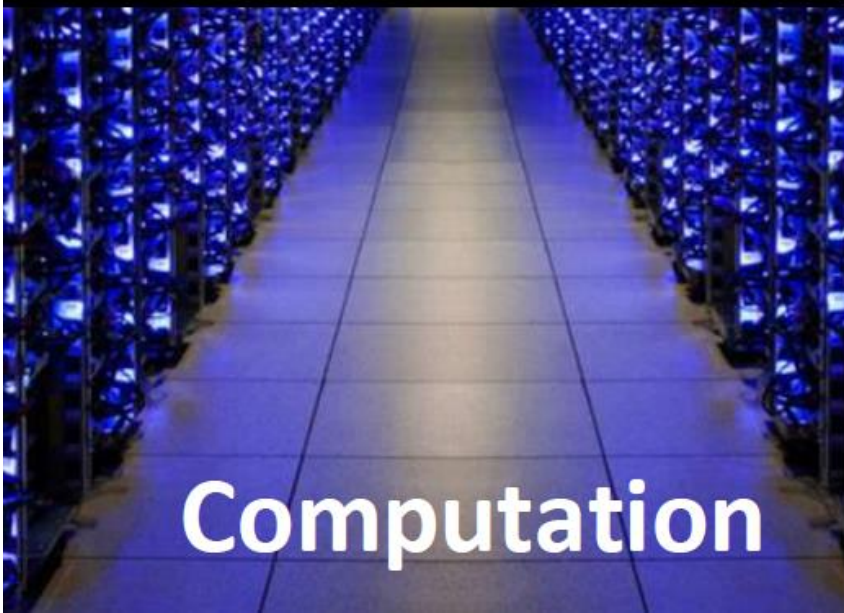
# 9. 딥러닝 모델 학습방법



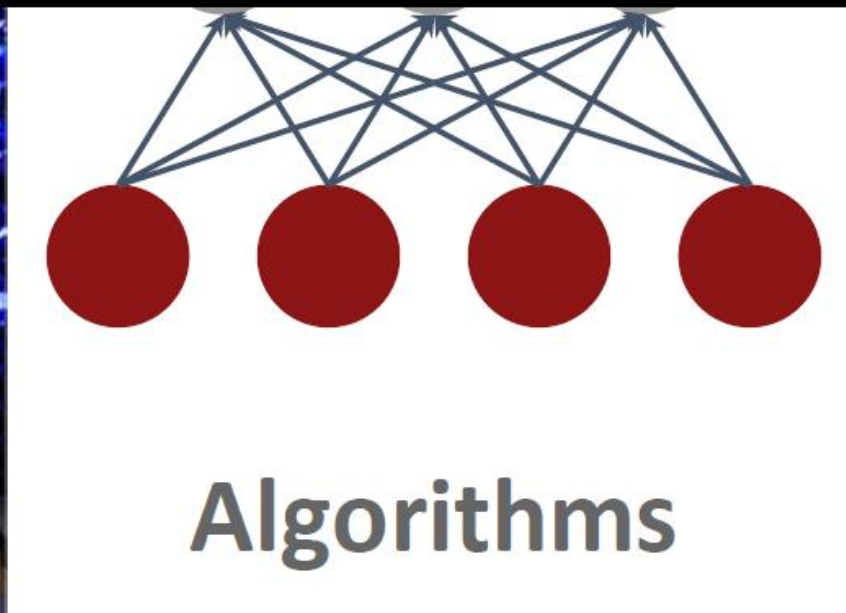




# The Deep Learning Revolution



Computation



Algorithms

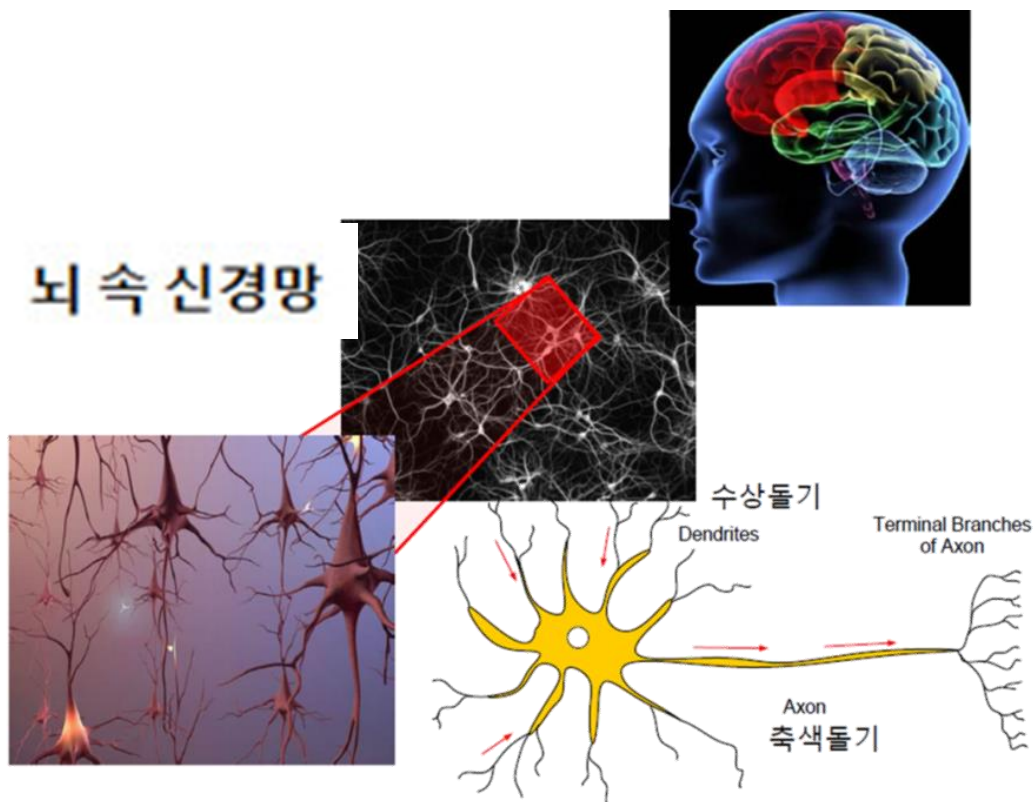


Data



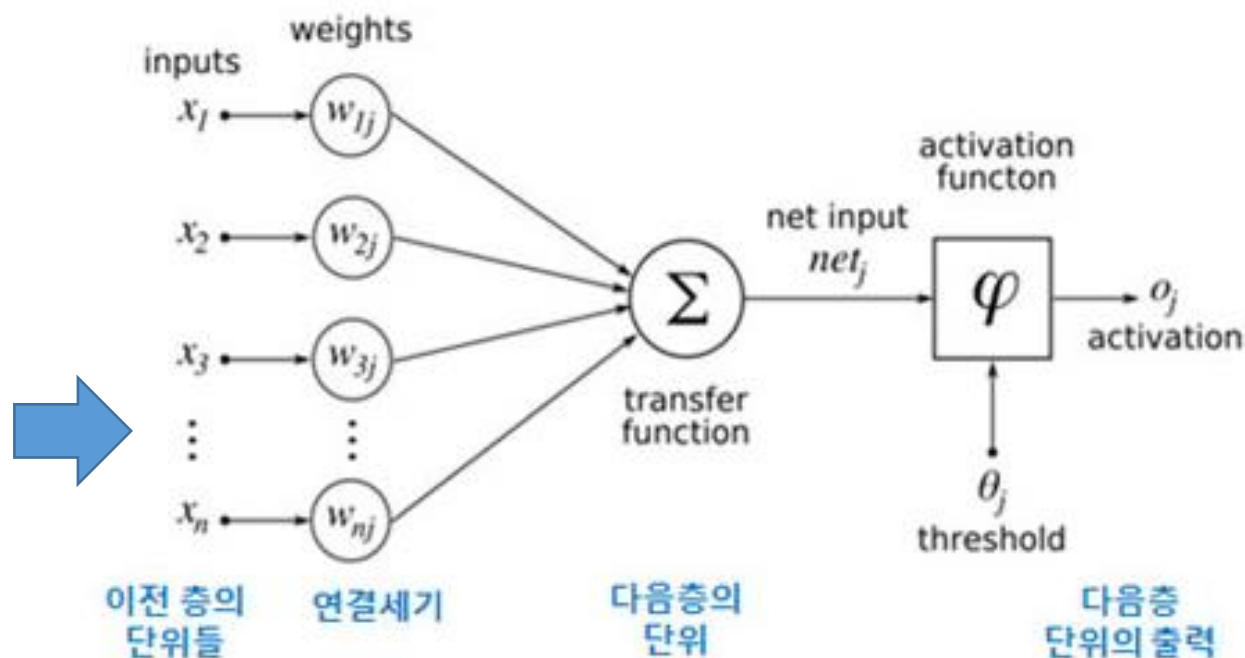
# 인공신경망(Artificial Neural Network, ANN)

- 뇌신경은 수많은 신경세포(뉴런, neuron)들이 연결되어 정보를 처리하고 전달합니다.
- 인공신경망은 뇌 신경계의 정보처리 구조를 모방하여 만든 계산 알고리즘입니다.
- 뇌 신경계와 같이 수많은 계산 함수를 연결하여 복잡한 정보를 처리하는 네트워크 구조입니다.

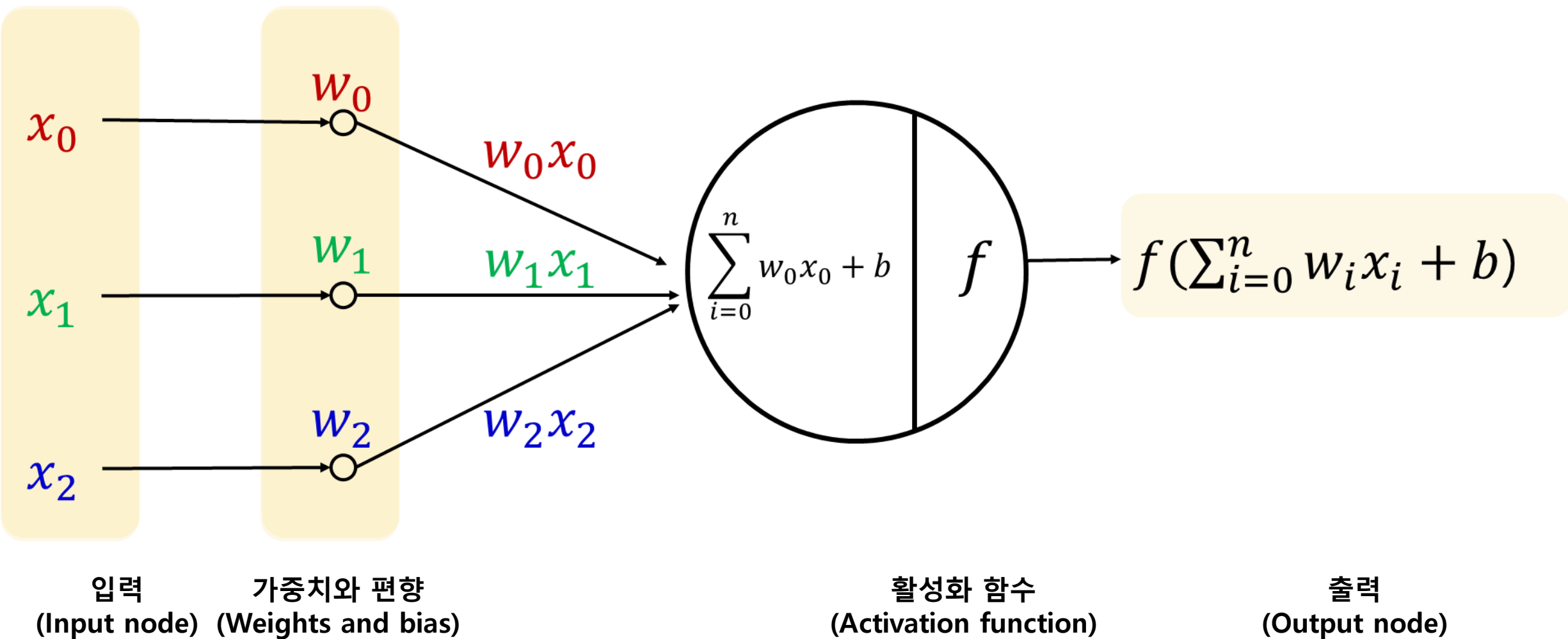


생물학적 뉴런(Neuron)

인공 뉴런(Neuron)



# 인공 뉴런 모델의 수학적 표현

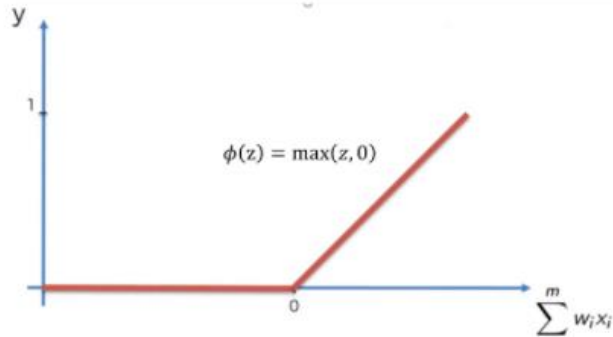


# 활성화 함수(Activation function)

입력값들의 수학적 선형결합을 다양한 형태의 비선형(또는 선형) 결합으로 변환하는 역할을 합니다.

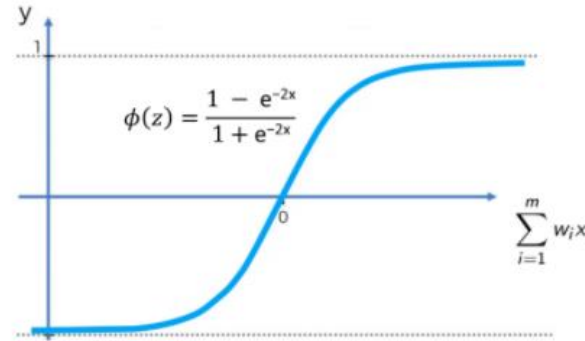
## 렐루 (ReLU)

입력이 양수일 때는  $x$ , 음수일 때는 0을 출력



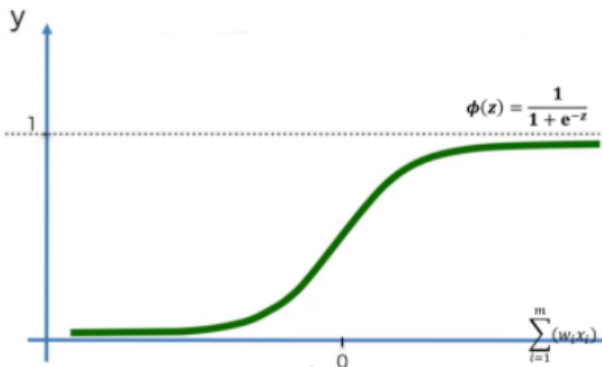
## 하이퍼볼릭 탄젠트 (Hyperbolic Tangent)

선형함수의 결과를 -1~1까지의 비선형 형태로 변경하는 함수



## 시그모이드 (Sigmoid)

0~1까지의 비선형 형태로 변경



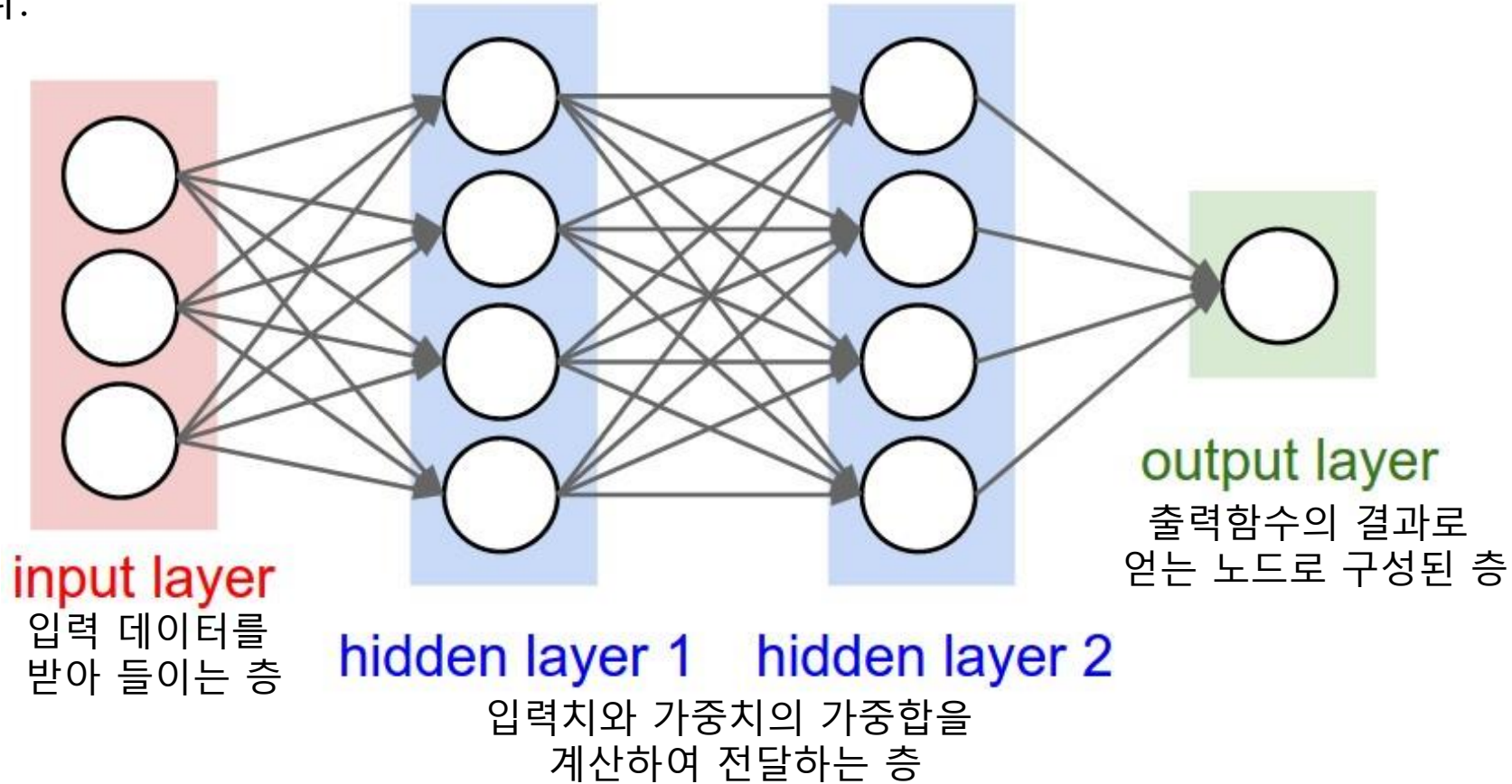
## 소프트맥스 (Softmax)

입력값을 0~1 사이 출력이 되도록 정규화, 출력값들의 총합은 항상 1

$$\phi(z) = \frac{e^i}{\sum_{j=0}^k e^j} \quad \text{where } i=0,1,\dots,k$$

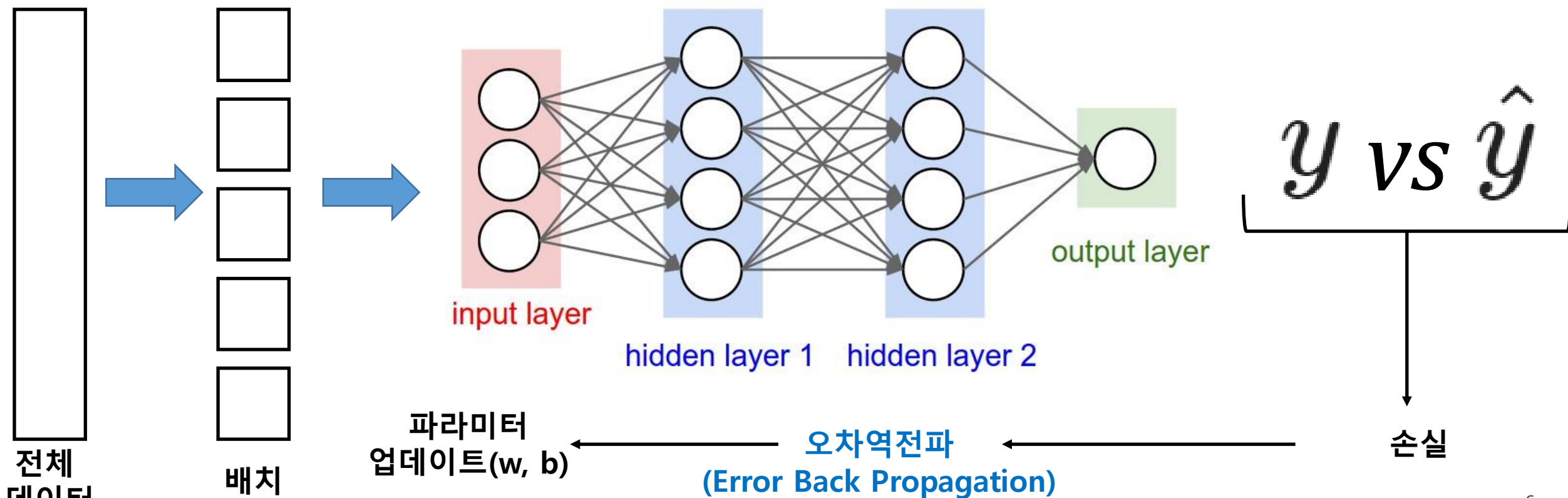
# 심층신경망(DNN , Deep Neural Network)

- 딥러닝은 여러 층(layer)을 가진 인공신경망(Artificial Neural Network)을 사용하여 학습을 수행하는 것입니다.
- 심층신경망은 입력층과 출력층사이에 다수의 은닉층(hidden layer)을 포함하는 인공신경망입니다.
- 머신러닝에서는 비선형 분류를 위해 여러 trick을 사용하지만, DNN은 다수의 은닉층으로 비선형 분류가 가능해집니다.



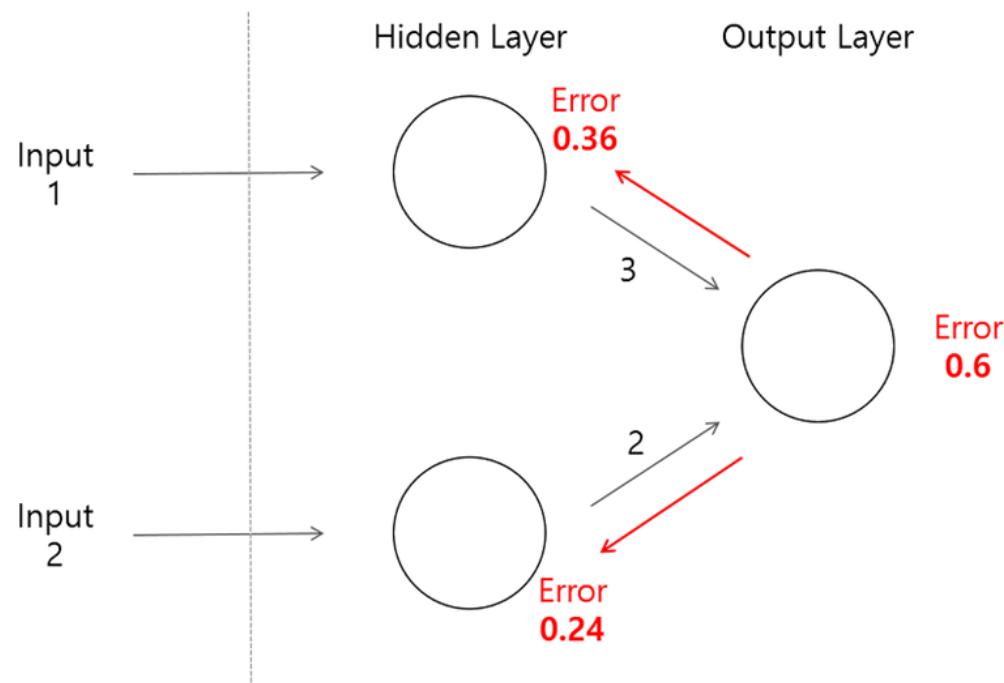
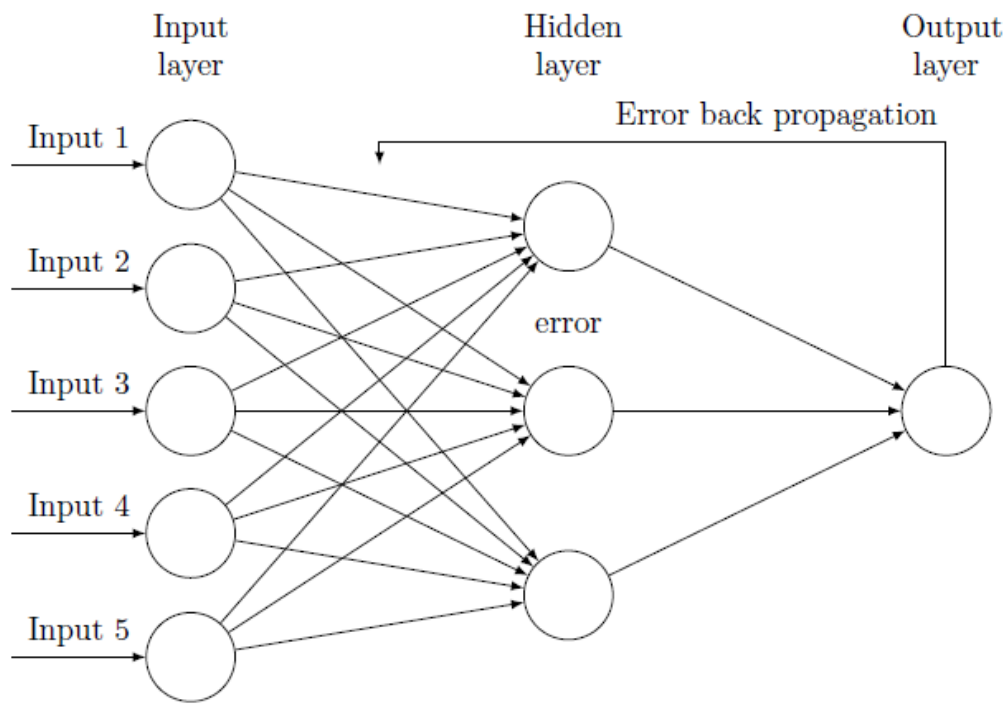
# 딥러닝 모델 학습과정

1. 미니배치  $x$ 를 샘플링합니다.
2.  $x$ 를 신경망에 통과시켜  $y$ 에 대한 예측  $\hat{y}$ 을 만듭니다.
3.  $y$ 와  $\hat{y}$ 을 비교하여 손실을 계산합니다.
4. 손실(Loss)을 기반으로 경사하강법을 적용해  $x$ 가  $y$ 를 더 잘 예측할 수 있도록  $w$ 와  $b$ 를 조정합니다.



# 딥러닝 모델 학습과정

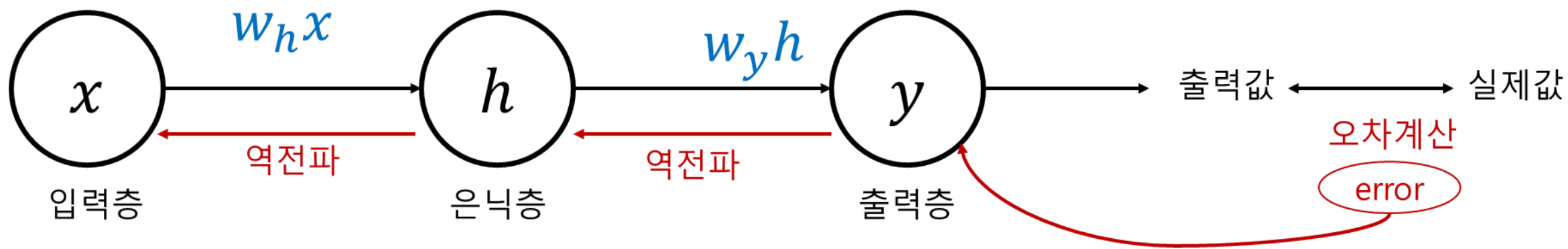
- 딥러닝 모델 학습은 모델에 입력값을 넣었을 때의 출력값이 최대한 정답과 일치하게 하는 것입니다.
- 딥러닝 모델 학습은 손실(Error/Loss)를 최소화 하는 인공신경망의 가중치(weight)와 편향(bias)을 찾는 과정입니다.
- 딥러닝 모델 매개변수(weight, bias)를 무작위로 부여한 후, 반복학습(순전파-오차역전파)을 통해 모델의 출력값을 정답과 가깝게 되도록 매개변수(weight, bias)를 조금씩 조정합니다.
- 모델 학습은 순전파(Forward Propagation)와 **오차역전파(Error Back Propagation)**의 반복으로 진행 됩니다.





# 역전파 알고리즘(Error Back Propagation)

- 신경망 모델은 파라미터( $w$ ,  $b$ ) 값을 랜덤하게 초기화 합니다.
- 신경망 모델에  $x$ 값을 입력하면 신경망은 예측  $y_{\text{hat}}$ 을 출력합니다.
- 신경망에서 손실 최소화를 위해 가중치를 업데이트 해야 합니다.
- 역전파를 사용해 신경망에 있는 모든 가중치에 대한 비용함수의 Gradient를 계산합니다.
- 가중치에 대한 비용함수의 Gradient에 비례하여 가중치를 조정함으로써 역전파는 비용을 감소하는 방향으로 가중치를 변경할 수 있습니다.



# 손실함수 (Loss Function)

인공신경망 학습의 목적함수로 출력값(예측값)과 정답(실제값)의 차이를 계산합니다.

## ■ 회귀(Regression)

평균제곱오차 : Mean Squared Error

평균절대오차 : Mean Absolute Error

## ■ 이진분류(Binary Classification)

Binary Cross Entropy

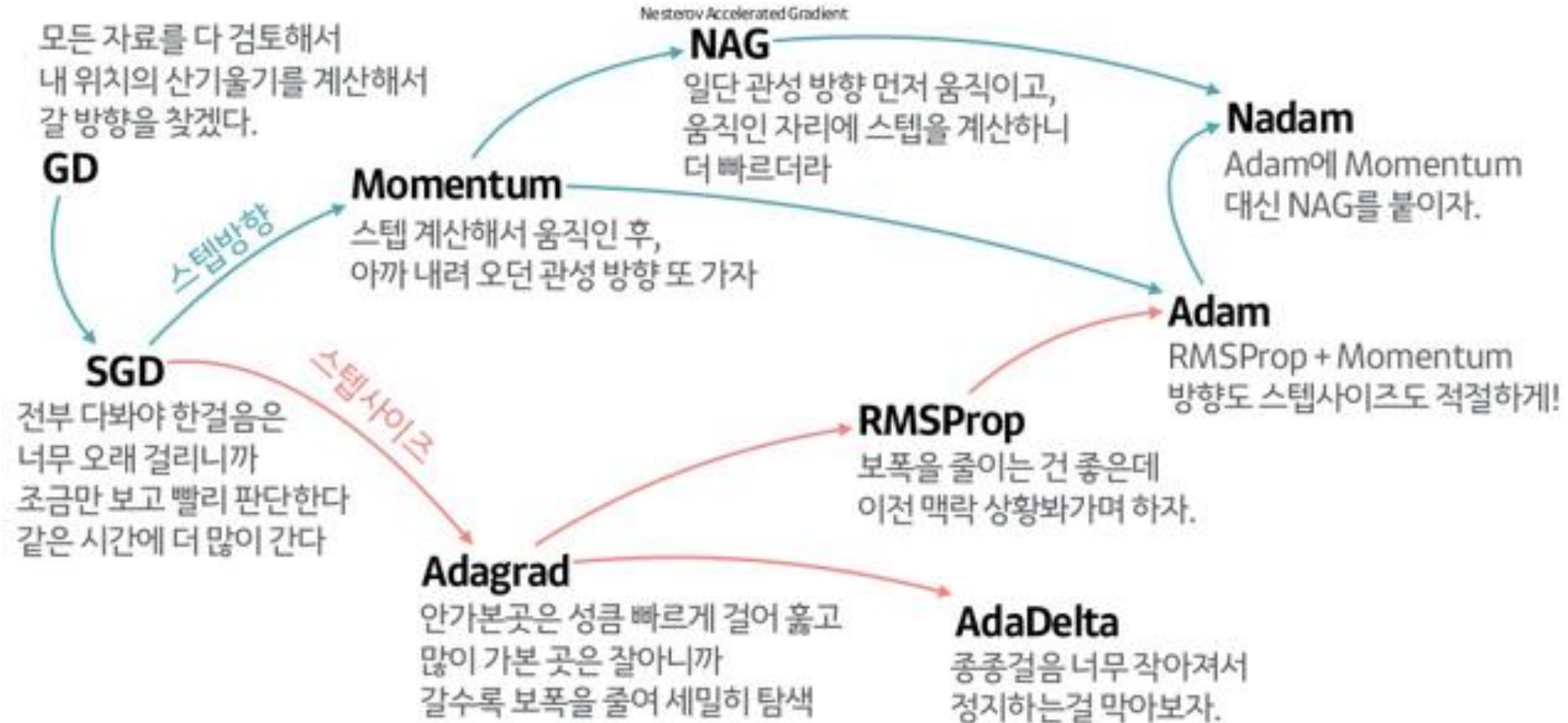
## ■ 다중분류(Multi-class Classification)

Categorical Cross Entropy : label(target, 출력값)이 **원핫 벡터(One-Hot Vector)**

Sparse Categorical Cross Entropy : label(target , 출력값)이 정수(0, 1, 2, 3 또는 n)

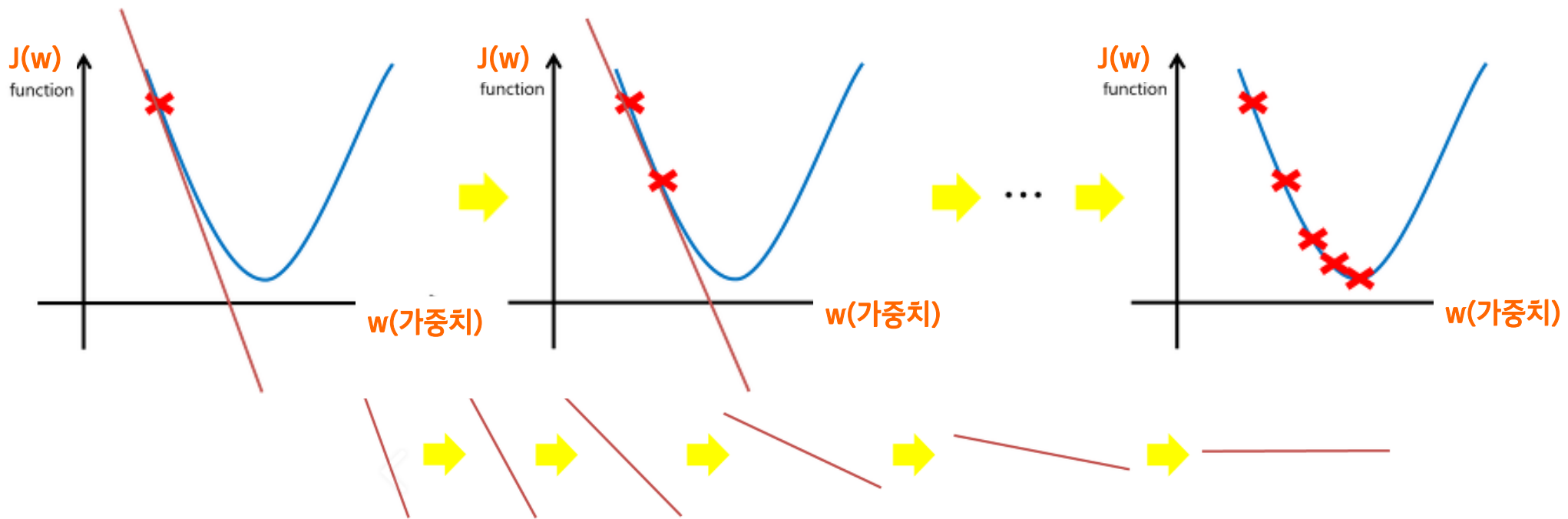
# 옵티마이저 (Optimization Algorithm)

딥러닝 모델의 매개변수(weight, bias)를 조절해서 손실함수의 값을 최저로 만드는 과정으로 경사하강법(Gradient Descent)이 대표적입니다.



# 경사하강법 (Gradient Descent)

- 손실함수  $J(w)$ 는 가중치( $w$ )의 함수로, 볼록함수 형태라면 미분으로 손실이 가장 작은 가중치를 찾을 수 있습니다.
- 하지만, 딥러닝에서는 손실함수가 복잡하고 계산량이 매우 크고, 미분이 0이 되는 값이 여러 개 존재하므로 미분만으로 최소값을 찾기 어려워 경사하강법(Gradient Descent)을 사용합니다.
- 경사하강법은 손실함수의 현 가중치에서 기울기를 구해서 손실(Loss)을 줄이는 방향으로 업데이트 해 나갑니다.



경사가 점차 감소되는 현상을 이용하므로 경사감소법!

참고 : [https://angeloyeo.github.io/2020/08/16/gradient\\_descent.html](https://angeloyeo.github.io/2020/08/16/gradient_descent.html)  
[https://angeloyeo.github.io/2020/08/16/gradient\\_descent.html](https://angeloyeo.github.io/2020/08/16/gradient_descent.html)



# 확률적 경사하강법 (SGD, Stochastic Gradient Descent)

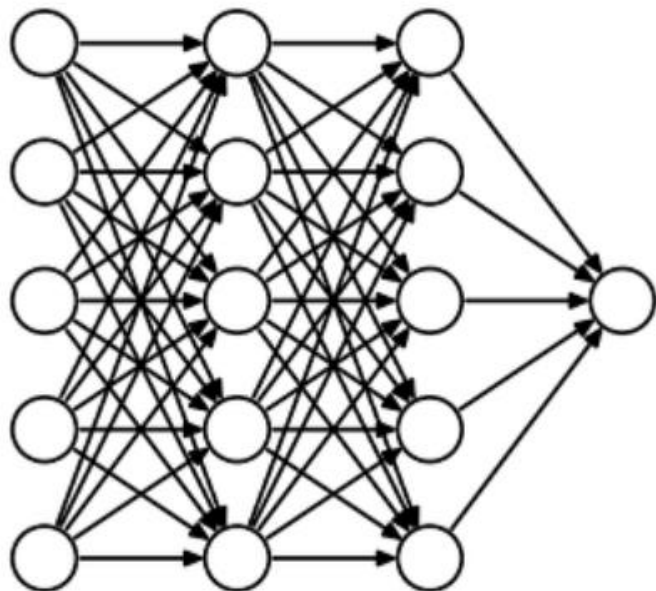
- 대규모 데이터셋은 전체 데이터를 메모리에 적재하기 어렵고, 수백만개의 파라미터를 가진 신경망 훈련의 계산복잡도로 기본 경사하강법은 비효율적입니다.
- 메모리와 계산의 제약에 대한 해결책으로, 훈련 데이터를 미니배치로 나누어 경사하강법을 수행하는 확률적 경사하강법이 있습니다.



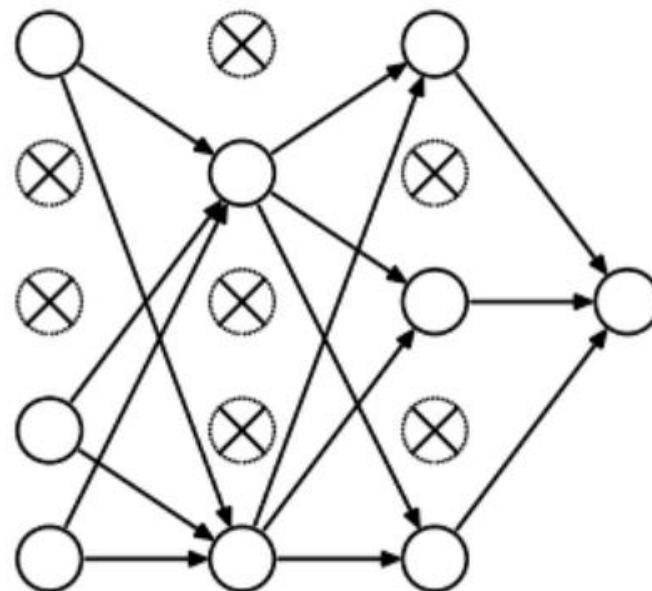
$$\text{미니배치 개수} = \left\lceil \frac{\text{훈련 데이터 크기}}{\text{배치 크기}} \right\rceil$$

# 드롭아웃(Dropout)

Hidden Layer의 일부 유닛이 동작하지 않게 하여 overfitting(과적합)을 막는 방법입니다.



일반 심층신경망



Dropout이 적용된 심층신경망

## ■ 과적합(overfitting)

생성된 모델이 학습 데이터와 지나치게 일치하여 새 데이터를 올바르게 예측하지 못하는 경우입니다.

## ■ 일반화(generalization)

모델학습에 사용된 데이터가 아닌 이전에 접하지 못한 새로운 데이터에 대해 올바른 예측을 수행하는 능력을 의미합니다.

# 텐서 (Tensor)

Scalar  
(0 Dimension)



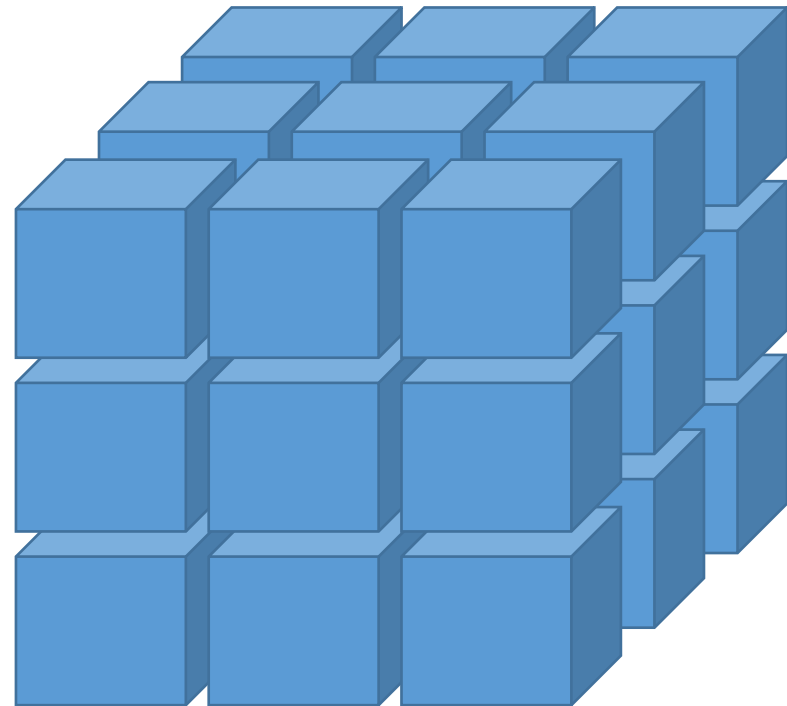
Vector  
(1 Dimension)



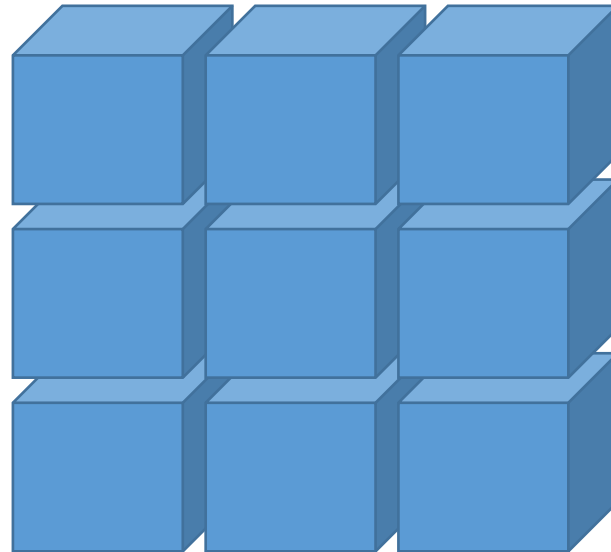
Matrix  
(2 Dimension)



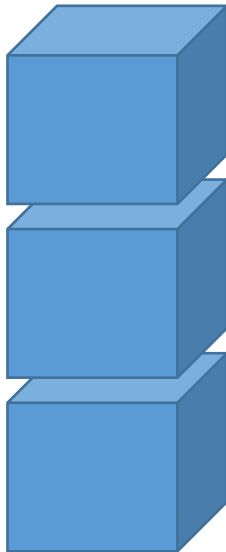
Tensor  
(6 Dimension)



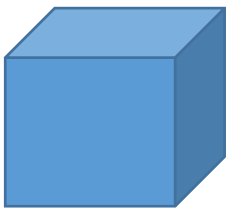
Tensor  
(5 Dimension)



Tensor  
(4 Dimension)



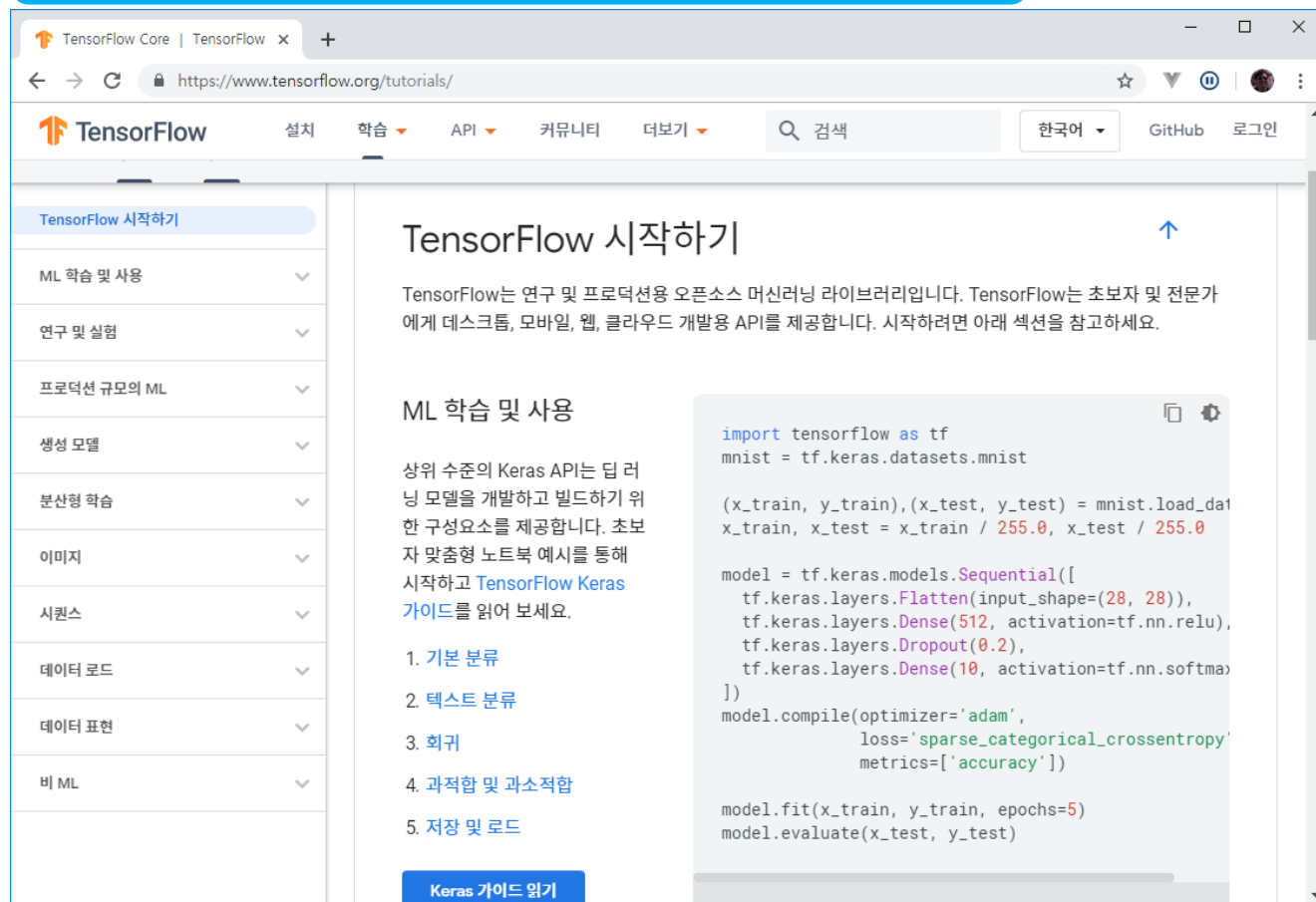
Tensor  
(3 Dimension)



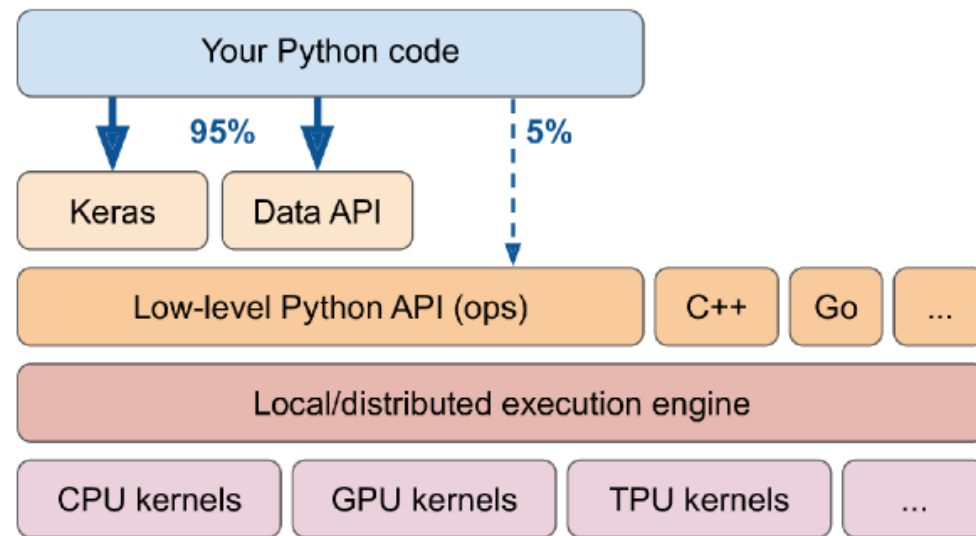
# TensorFlow

딥러닝 프로그램을 쉽게 구현할 수 있도록 다양한 기능을 제공하는 프레임워크입니다.

<https://www.tensorflow.org/tutorials/>



## 텐서플로 구조

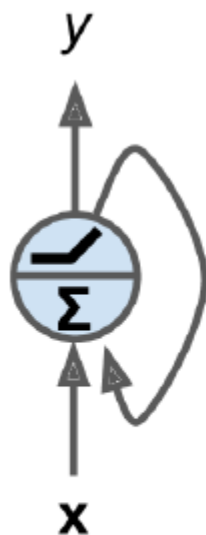




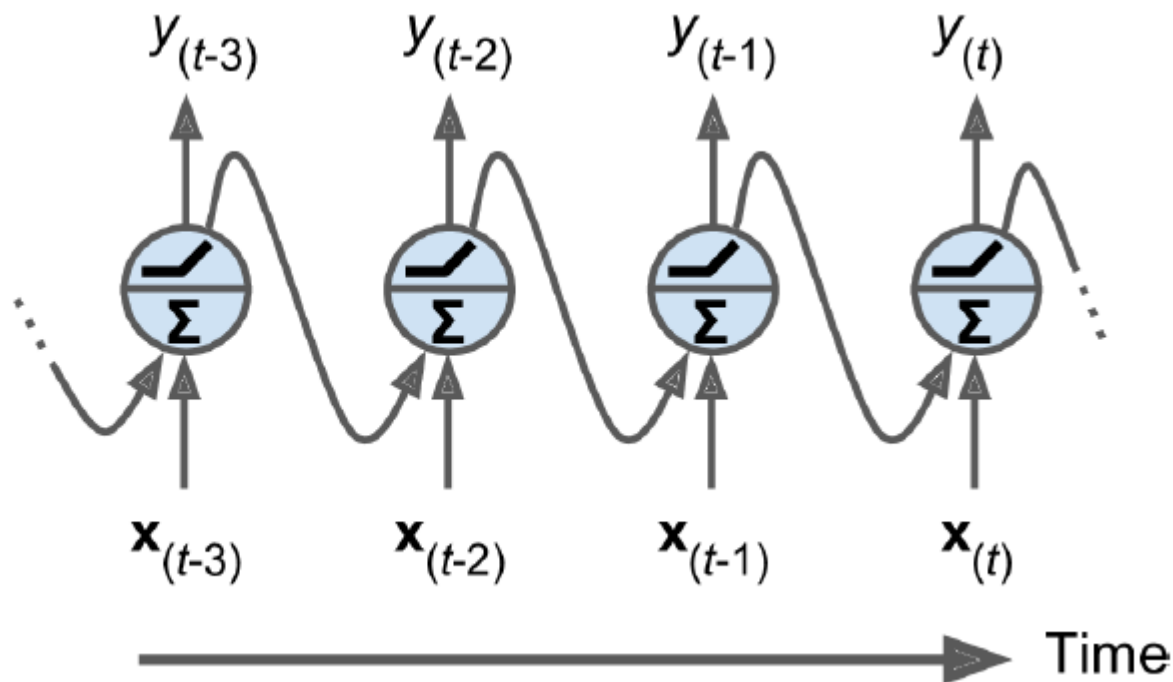
# 순환 신경망(RNN, Recurrent Neural Network)

순환신경망은 고정 길이 입력이 아닌 임의의 길이를 가진 순차데이터(Sequence Data)를 다룰 수 있습니다.  
시계열 데이터로부터 미래값을 예측하고, 문장/오디오를 입력으로 받아 자동번역, 자연어처리를 하는 작업에 유용합니다.

■ 순환뉴런

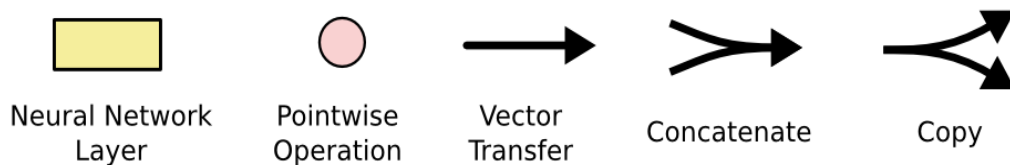
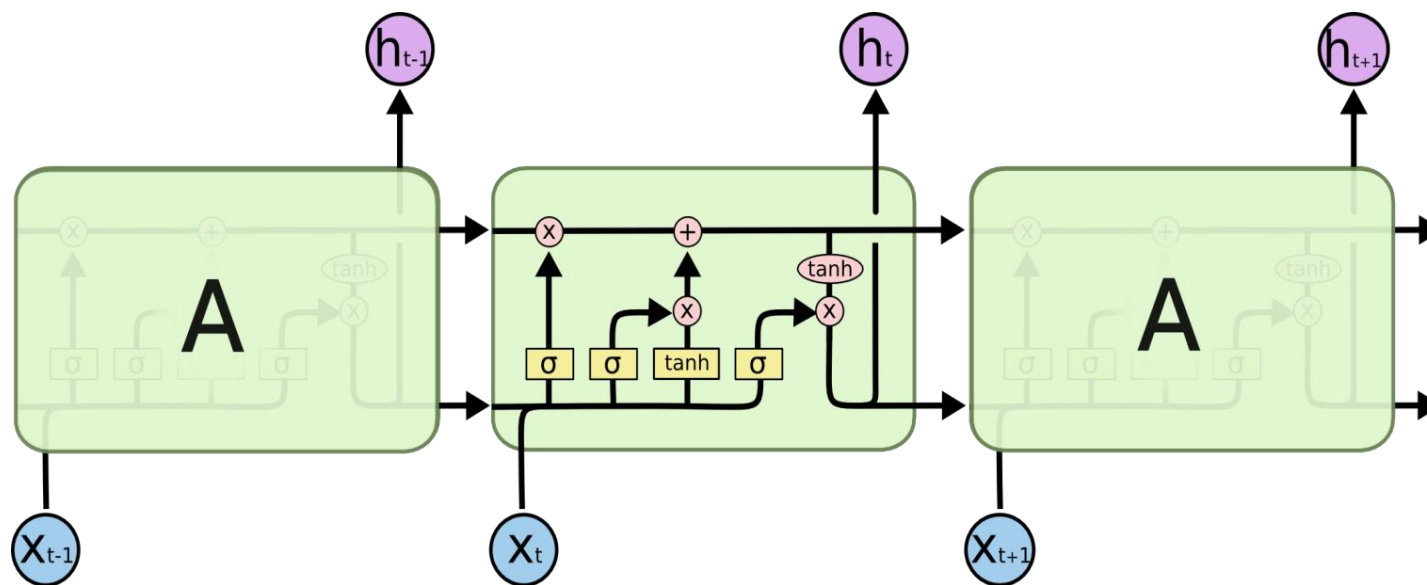


■ 순환뉴런을 타임 스텝으로 펼친 모습



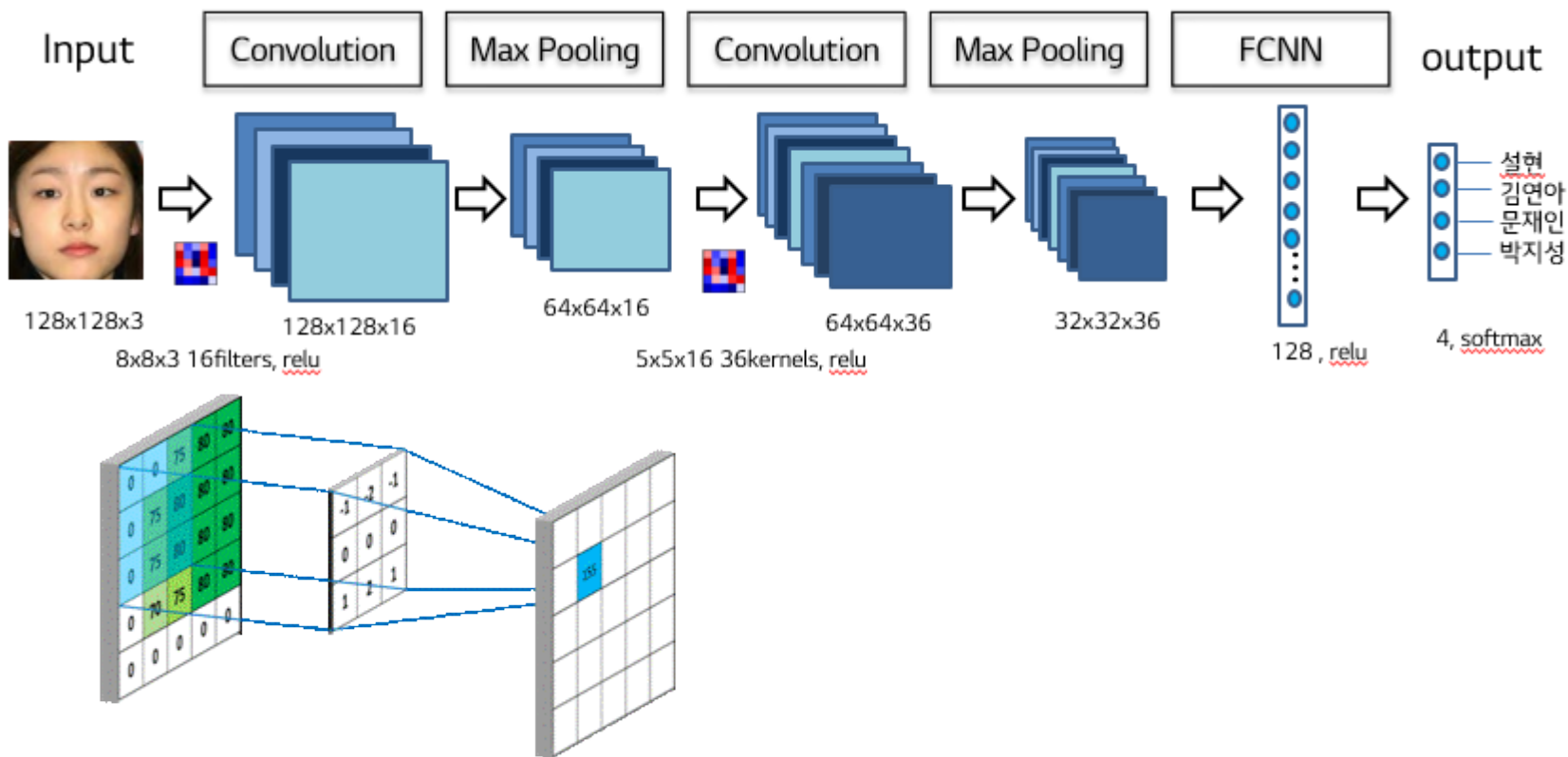
# 순환 신경망 - LSTM(Long-Short Term Memory)

LSTM 네트워크는 장기적인 종속성을 학습할 수 있는 특수한 종류의 RNN입니다.  
LSTM은 Forget gate, Input gate, Output gate를 통한 정보전이 및 전파를 제어합니다.



# 합성곱 신경망(CNN, Convolutional Neural Network)

사람의 시각 피질 메커니즘에 영감을 받아 설계된 이미지, 영상등을 인식하는 신경망 모델  
Convolution층에서는 각 filter가 입력 이미지의 픽셀 전체를 차례로 훑고 지나가며  
linear combination을 진행하고 Feature Map을 구성합니다.



# Thank you