



Keras (1)

고려대학교 INI Lab

Contents

01 Deep Learning

02 Deep Learning Framework

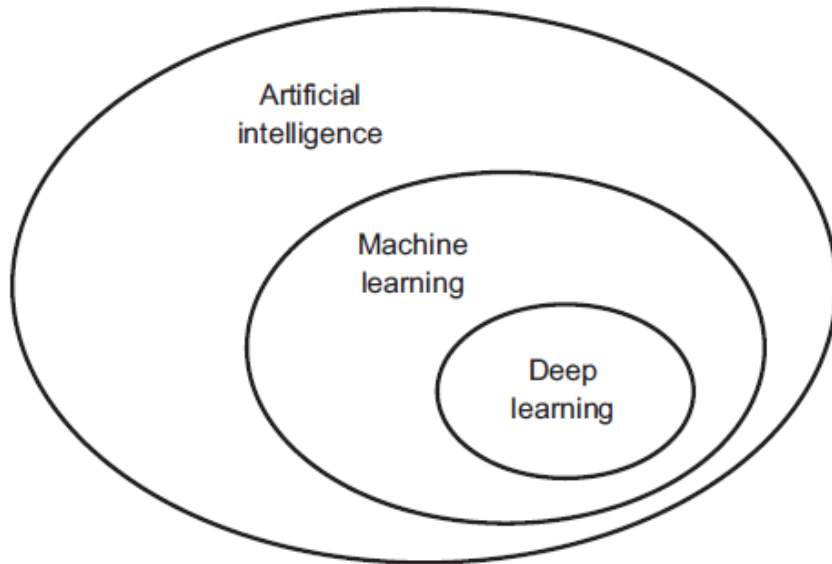
03 Feed Forward Neural Network

04 Improve Model



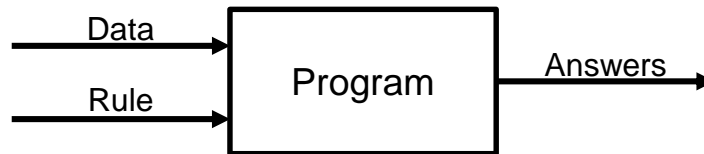
Deep Learning

What is Deep Learning?

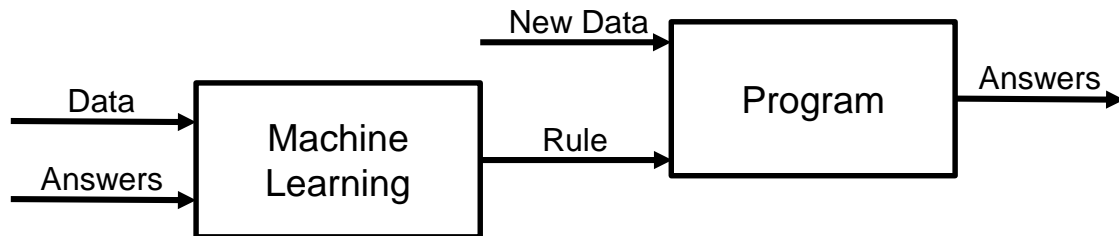


Machine Learning?

Machine Learning은 컴퓨터에게 사람이 직접 명시적으로 규칙을 지시하지 않아도 데이터를 통해 컴퓨터가 '학습'을 하고 그것을 사용해 컴퓨터가 자동으로 문제를 해결하도록 하는 것



<Classical Programming Architecture>



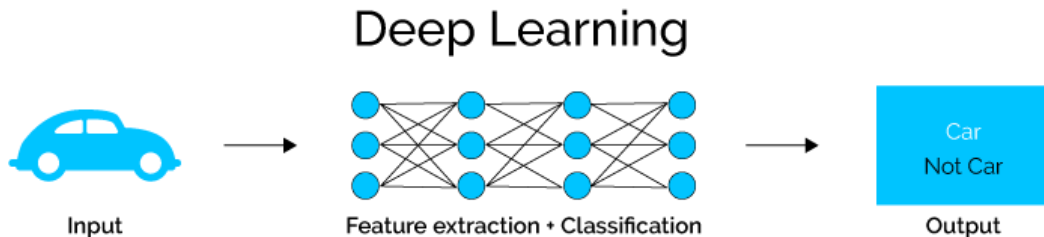
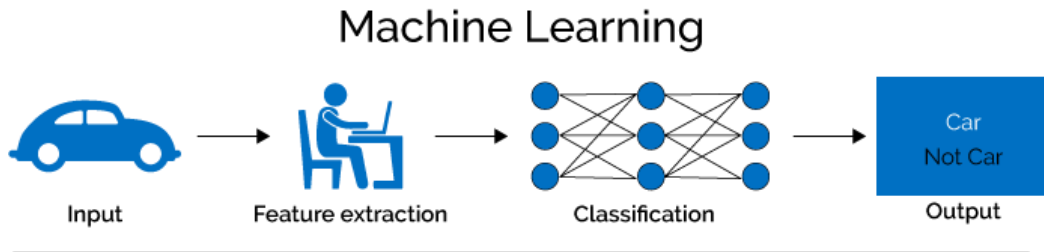
<Machine Learning Architecture>

Defect of traditional Machine Learning?

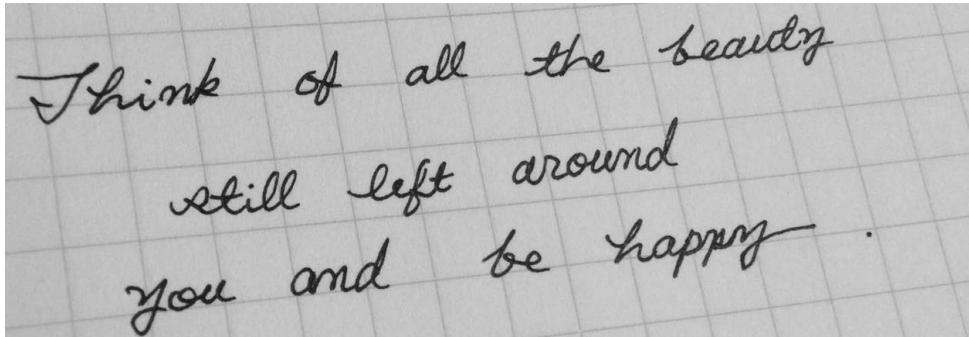
Low-level feature로 구성된 데이터에 대해서는 제대로 된 학습을 기대하기 어렵다.

사람이 데이터에 사전작업을 수행하여 High-level feature를 추출해야한다.

그런데 사람이 쉽게 이해할 수 있는 high-level feature를 설명하기는 어렵다.



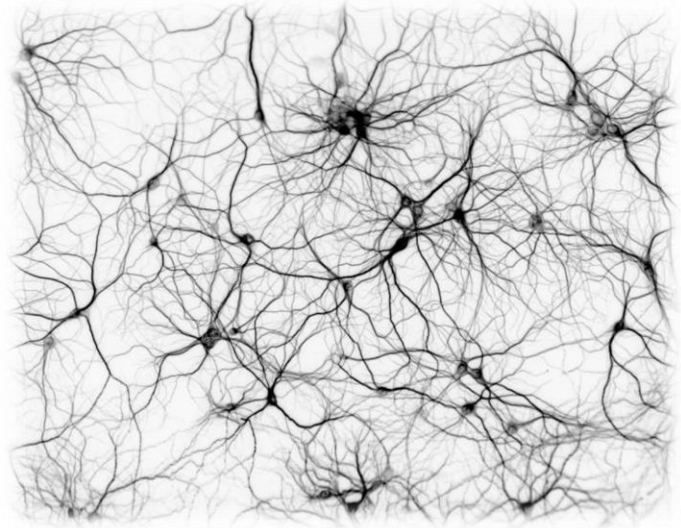
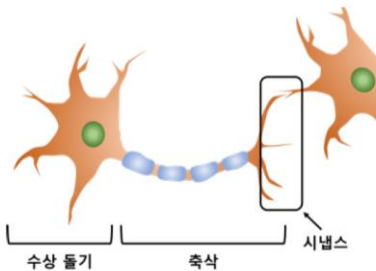
High-level and Low level



Think of all the beauty
still left around
you and be happy.

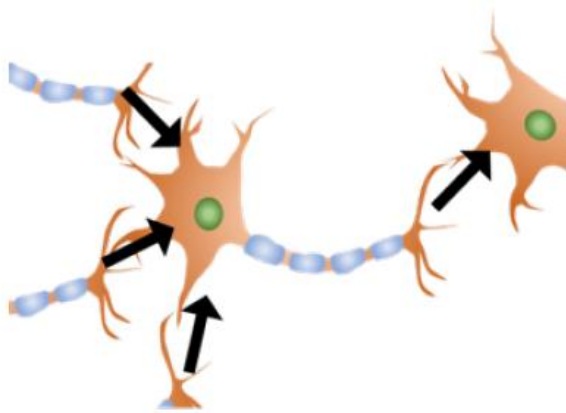
Neural Network

신경 세포와 뇌가 지식을 추론하고 학습하는 구조를 추상화한 모델

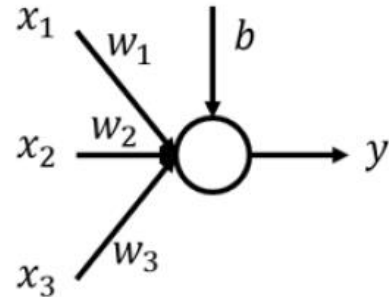


Neural Network

신경 세포와 뇌가 지식을 추론하고 학습하는 구조를 추상화한 모델

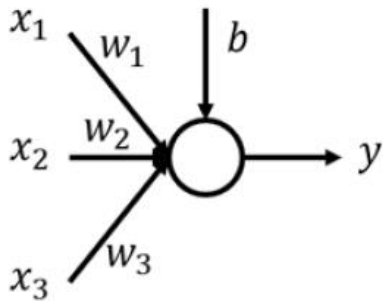


Neuron

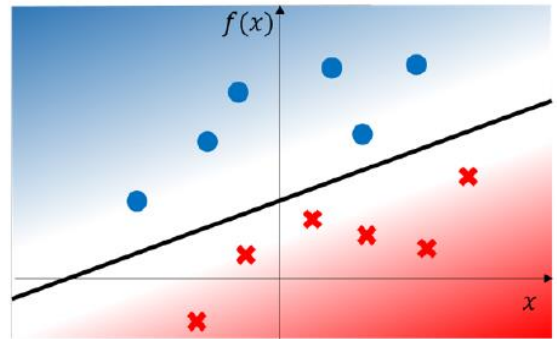


Perceptron

Perceptron

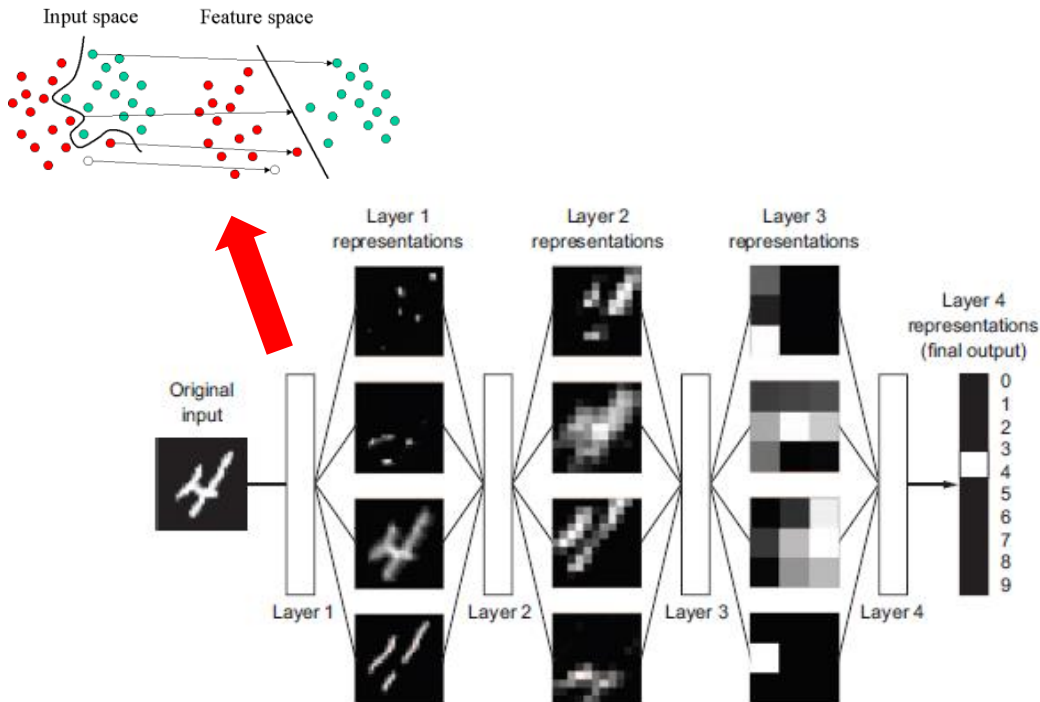


Perceptron



Linear Classification

Why Deep?



Low Level

High Level

Ideal Deep Learning

학습을 위한 지식 표현이 별도로 필요 없다

학습에 필요한 데이터 labeling 작업을 제외한 나머지 과정은 자동화가 가능하다

기계 학습 모델을 생성하기 위해서 고도의 수학적 지식이나 프로그래밍 능력을 요구하지 않는다

기계 학습을 통해 생성된 모델은 사람이 만든 모델과 유사한, 혹은 더 좋은 결과를 보여줄 수 있다

프로그램을 이용해 자신이 원하는 대로 사용할 수 있다

Real Deep Learning

학습을 위한 지식 표현이 별도로 필요 없다

- 해당 도메인에 대한 기본적인 지식이 없으면, 적합한 모델을 생성할 수 없다. 그리고 실제로 어떻게 학습이 일어나는지, 모델이 무엇을 학습했는지 사람이 파악하기 어렵다.

학습에 필요한 데이터 labeling 작업을 제외한 나머지 과정은 자동화가 가능하다

- 학습에 적합한 형태로 정제되지 않은 데이터를 사용할 경우 제대로 된 학습이 불가능한데, 이 과정 자체에 전문가를 요구하며 비용 또한 상당히 높다.

기계 학습 모델을 생성하기 위해서 고도의 수학적 지식이나 프로그래밍 능력을 요구하지 않는다

- 단순한 어플리케이션에서는 가능한 이야기지만, 차원이 방대하고 복잡한 데이터 도메인에 걸맞게 모델의 성능을 이끌어내기 위해서는 다양한 수학적 배경 및 프로그래밍 능력이 요구된다

기계 학습을 통해 생성된 모델은 사람이 만든 모델과 유사한, 혹은 더 좋은 결과를 보여줄 수 있다

- 사람보다 뛰어난 성능을 가진 모델을 생성하기 위해서는, 수많은 trial & error가 필요하다

프로그램을 이용해 자신이 원하는 대로 사용할 수 있다

- 다만 엄청나게 방대한 양의 데이터를 요구할 뿐

Example of Deep Learning

Machine translation

번역

즉석 번역 사용 안함



영어 한국어 독일어 언어 감지 ▼



한국어 영어 일본어 ▼

번역하기

구글 번역기는 이전에 사용되던 통계 기반에서 딥러닝 모델 기반의 번역으로 교체되었^x다. 새로운 번역기는 영어 전문가 수준의 질을 보여준다. 만약 제대로 된 번역이 되지 않는다면, 그 원인은 대부분 원문을 추상적으로 작성하는 사람의 오류일 가능성이 높다.

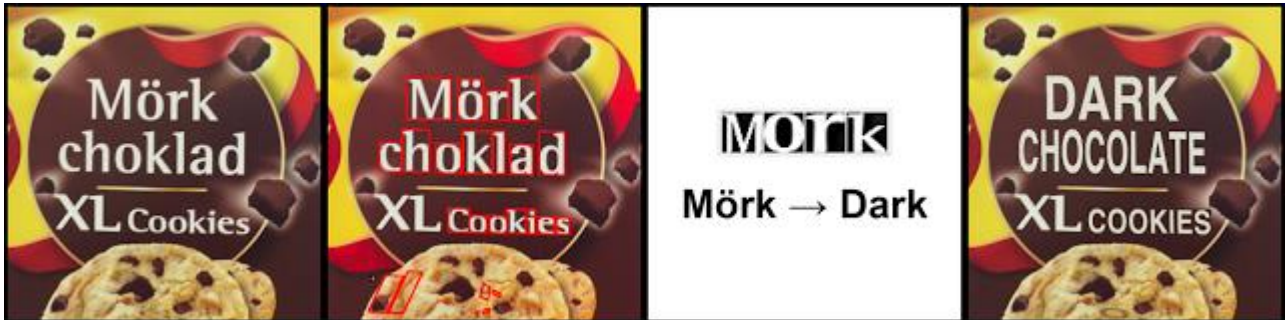


140/5000

Google Translator has been replaced by a translation based on the Deep Learning model from the previously used statistical basis. The new translator shows the quality of English professionalism. If it does not translate correctly, it is likely that the cause is mostly the error of the person writing the abstract.

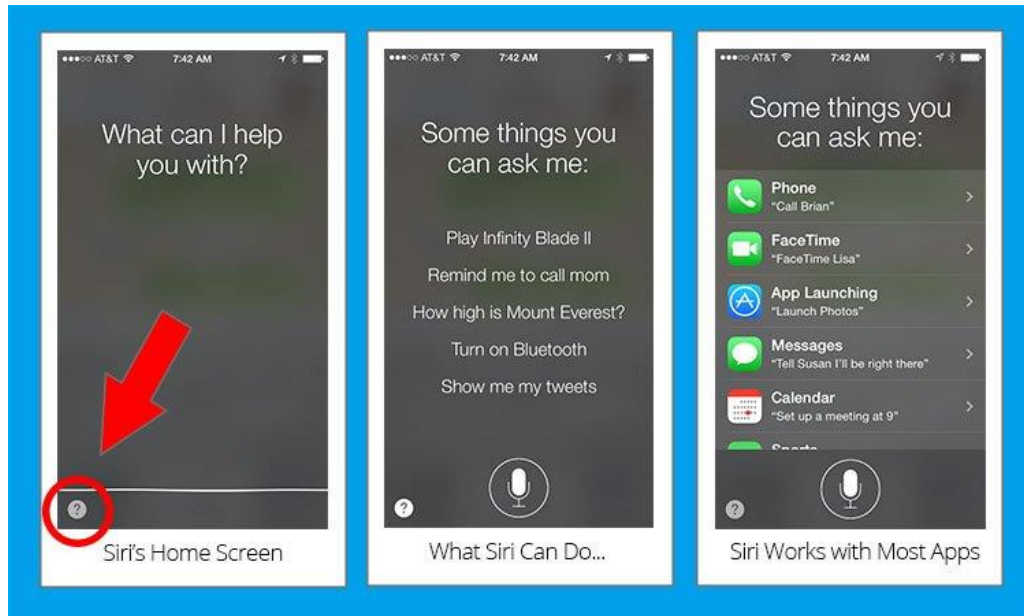


수정 제한하기



Example of Deep Learning

Speech Recognition



Example of Deep Learning

Self-Drive Car



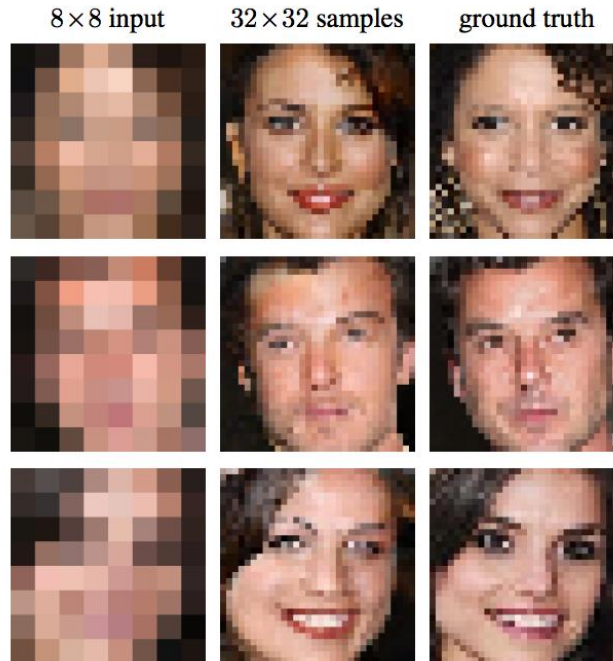
Example of Deep Learning

Object Recognition



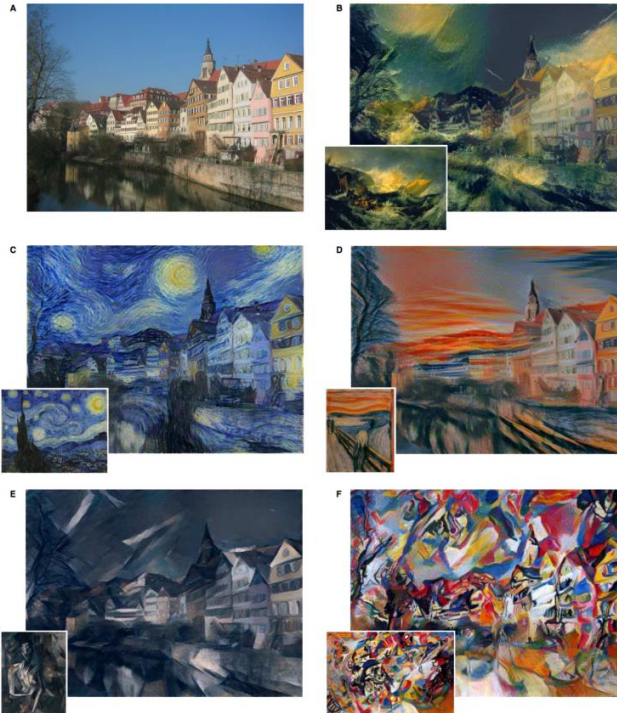
Example of Deep Learning

Image restoring



Example of Deep Learning

Image Generation



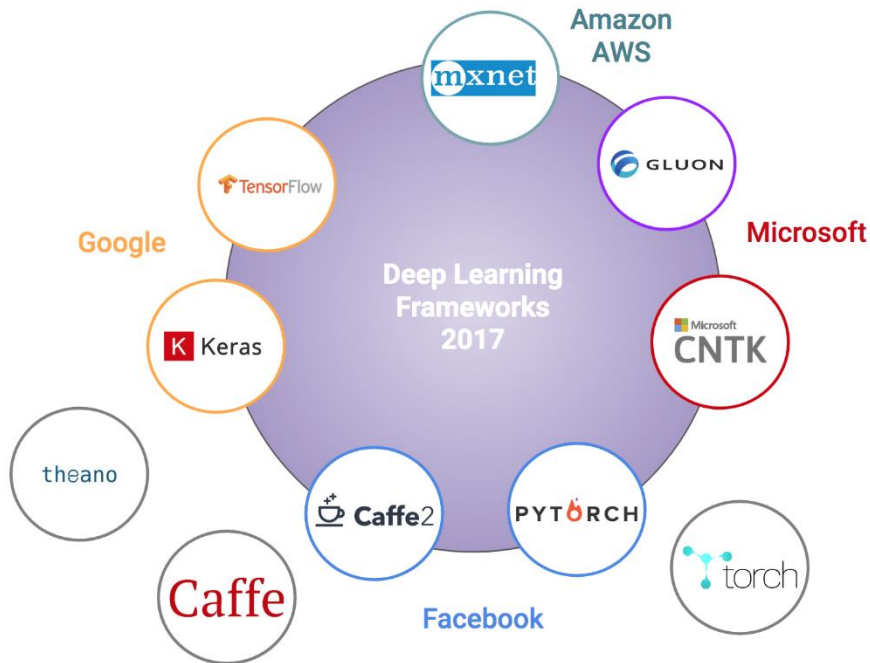
Example of Deep Learning

Video Generation

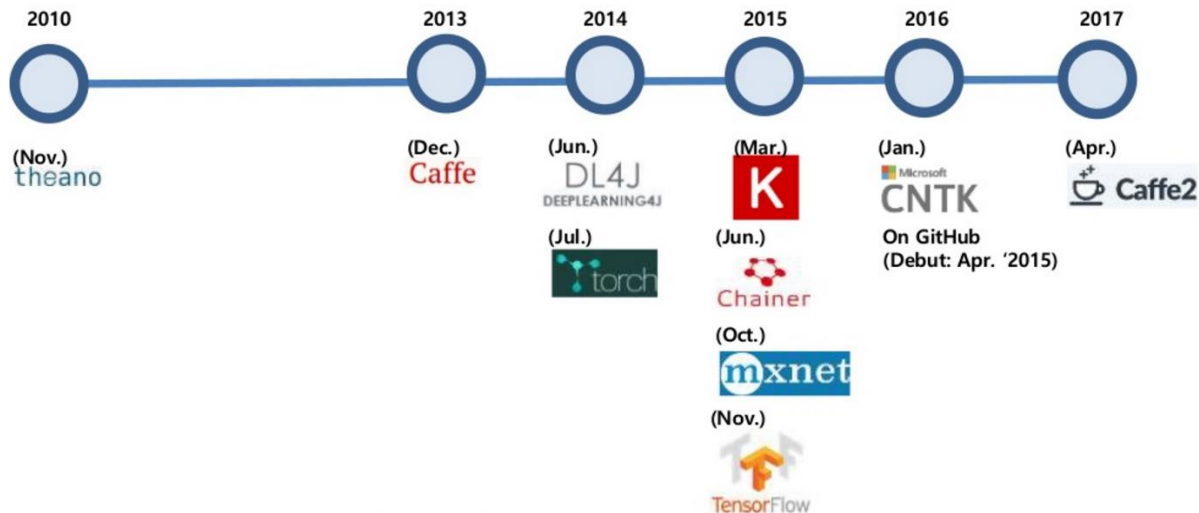


Deep Learning Framework

Deep Learning Framework



History of Deep Learning Framework



출처 : <https://www.popit.kr/딥러닝-프레임워크-조사와-몇가지-홍보>

Deep Learning Framework Comparison

F/W	주체	플랫폼	모바일	언어	인터페이스	OpenMP	CUDA	OpenCL	멀티GPU	분산
Caffe	BAIR	Linux, Mac	-	C++	Python, MATAB	Y	Y	-	Y	
Chainer	Preferred Networks	Linux	-	Python	Python	-	Y	-	Y	Y
CNTK	Microsoft	Linux, Windows	-	C++	Python, C++	Y	Y	-	Y	Y
DL4J	SkyMind	Cross-platform (JVM)	Android	Java	Java, Scala, Python	Y	Y	-	Y	Y (Spark)
Keras	François Chollet	Linux, Mac, Windows	-	Python	Python	Y(Theano) N(TF)	Y	-	Y	
MXNet	DMLC	Linux, Mac, Windows, Javascript	Android, iOS	C++	C++, Python, Julia, MATLAB, JavaScript, Go, R, Scala, Perl	Y	Y	-	Y	Y
TensorFlow	Google	Linux, Mac, Windows	Android, iOS	C++, Python	Python, C/C++, Java, Go	N	Y	-	Y	Y
Theano	Université de Montréal	Linux, Mac, Windows	-	Python	Python	Y	Y	-	Y	
Torch	Ronan, Clément, Koray, Soumith	Linux, Mac, Windows	Android, iOS	C, Lua	Lua	Y	Y	Y	Y	Not official y

출처 : <https://www.popit.kr/딥러닝-프레임워크-조사와-몇가지-홍보>

Deep Learning Framework Comparison

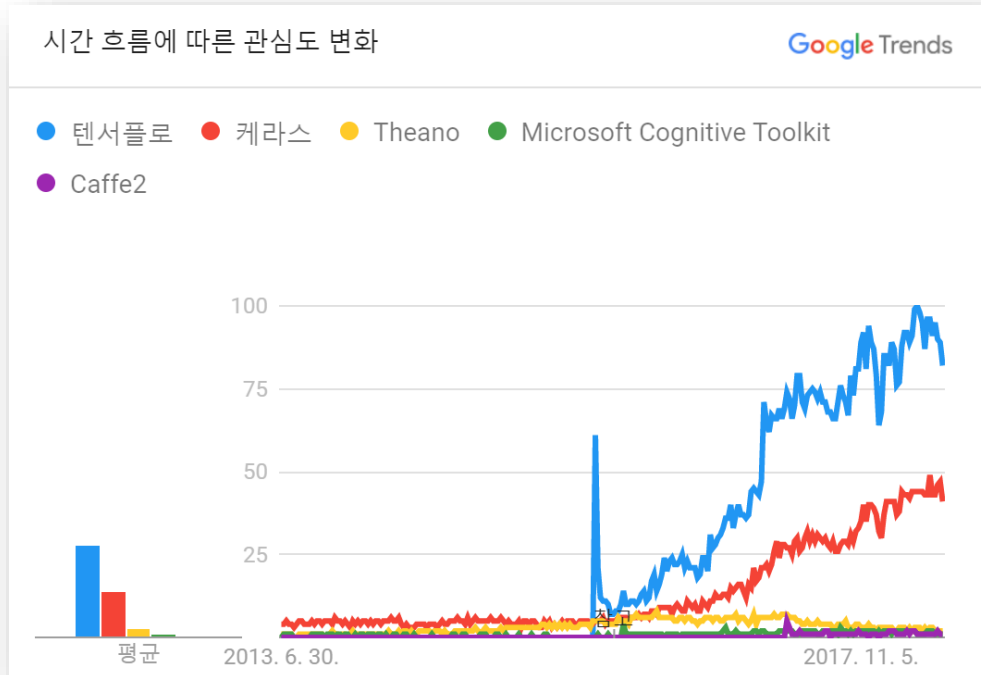
Design Choice	Torch.nn	Theano	Caffe	Chainer	MXNet	Tensor-Flow	PyTorch
NN definition	Script (Lua)	Script* (Python)	Data (protobuf)	Script (Python)	Script (many)	Script (Python)	Script (Python)
Backprop	Through graph	Extended graph	Through graph	Through graph	Extended graph	Extended graph	Through graph
Parameters	Hidden in operators	Separate nodes	Hidden in operators	Separate nodes	Separate nodes	Separate nodes	Separate nodes
Update formula	Outside of graphs	Part of graphs	Outside of graphs	Outside of graphs	Outside of graphs	Part of graphs	Outside of graphs
Graph construction	Static	Static	Static	Dynamic	Static	Static	Dynamic
Graph Optimization	-	Supported	-	-	-	Supported	-
Parallel computation	Multi GPU*	Multi GPU	Multi GPU*	Multi GPU**	Multi node Multi GPU	Multi node Multi GPU	Multi GPU**

출처 : DLIF: Differences of deep learning frameworks

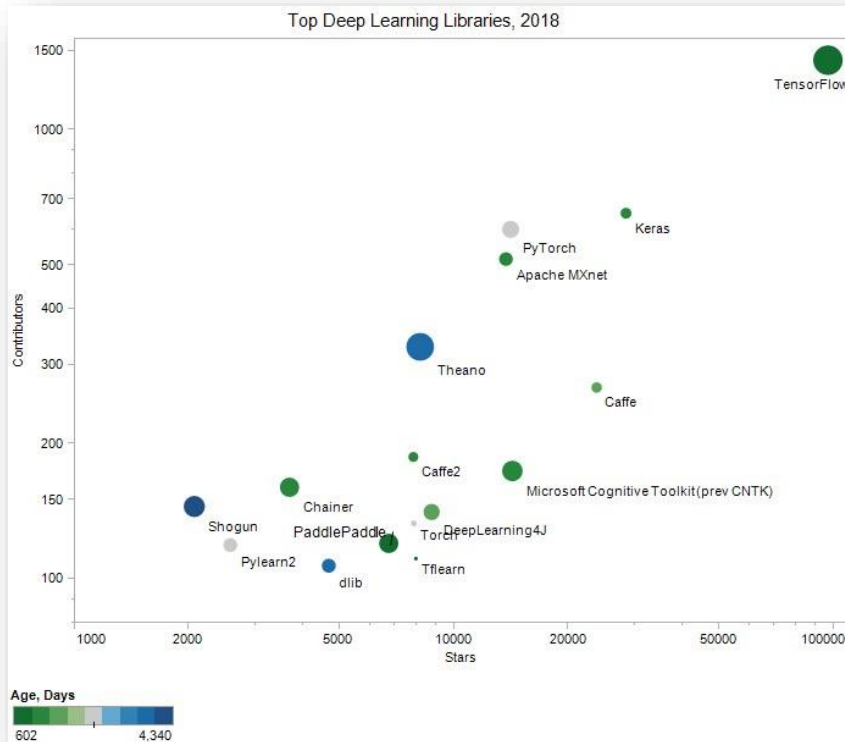
Which Framework to Choose?

1. You are a PhD student on DL itself: **TensorFlow, Theano, Torch**
2. You want to use DL only to get features: **Keras, Caffe**
3. You work in industry: **TensorFlow, Caffe**
4. You started your 2 month internship: **Keras, Caffe**
5. You want to give practice works to your students: **Keras, Caffe**
6. You are curious about deep learning: **Caffe**
7. You don't even know python: **Keras, Torch**

Deep Learning Framework Trend



Deep Learning Framework Trend

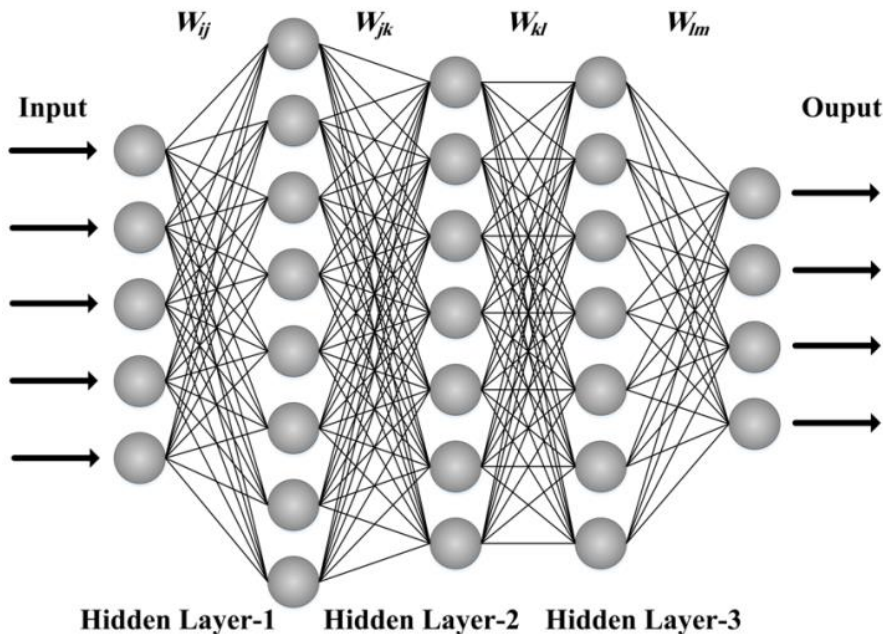


Feed Forward Neural Network

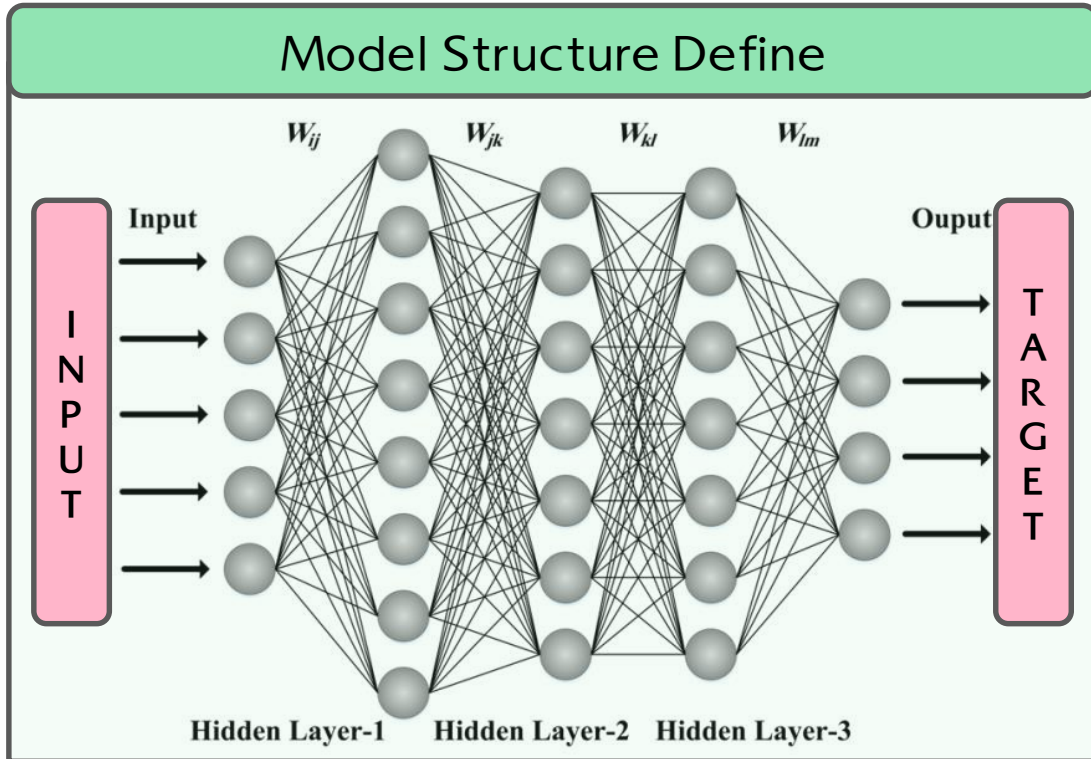
Feed Forward Neural Network

여러 hidden layer를 dense하게 쌓아서 학습시키는 모델

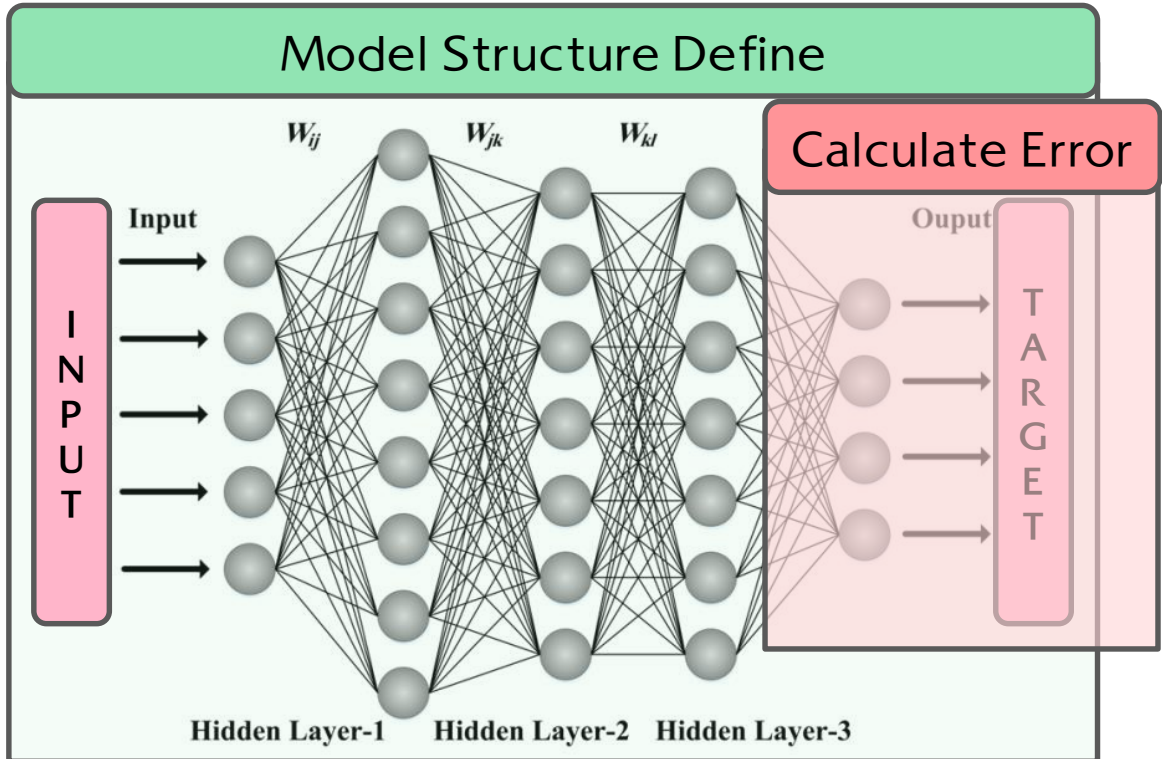
Multi-Layer Perceptron이라고도 부름



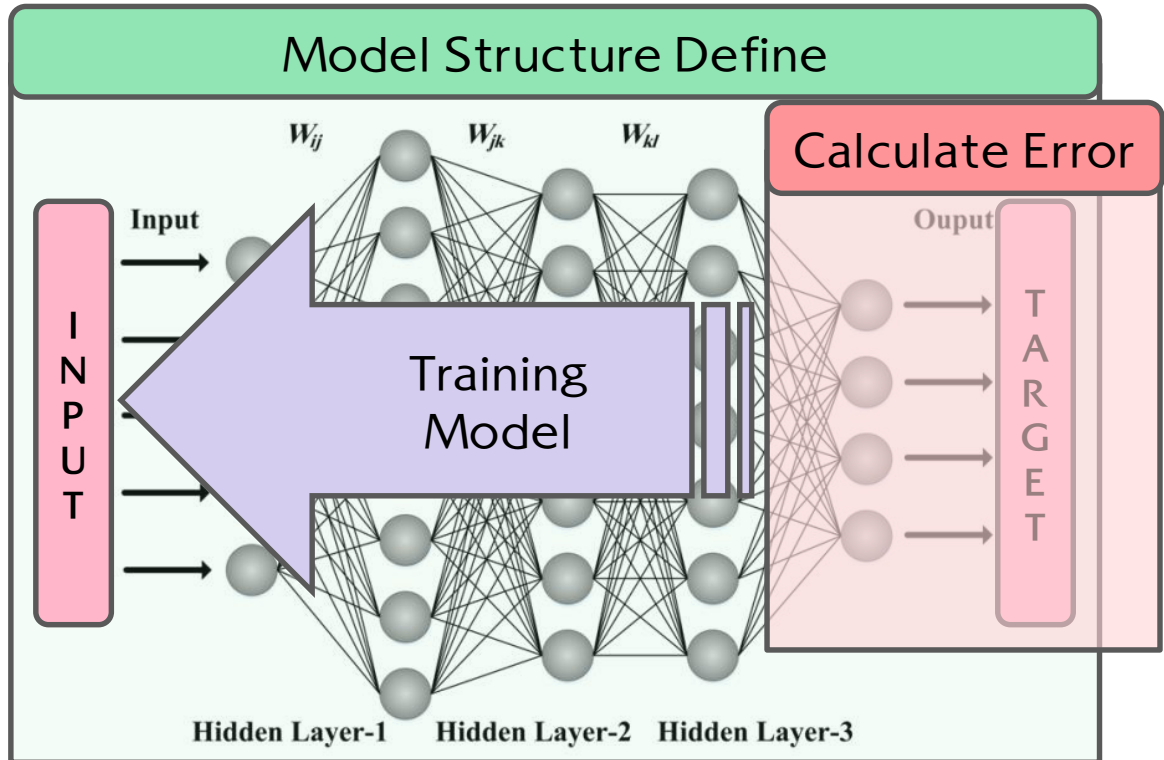
Build FFNN Model



Build FFNN Model



Build FFNN Model

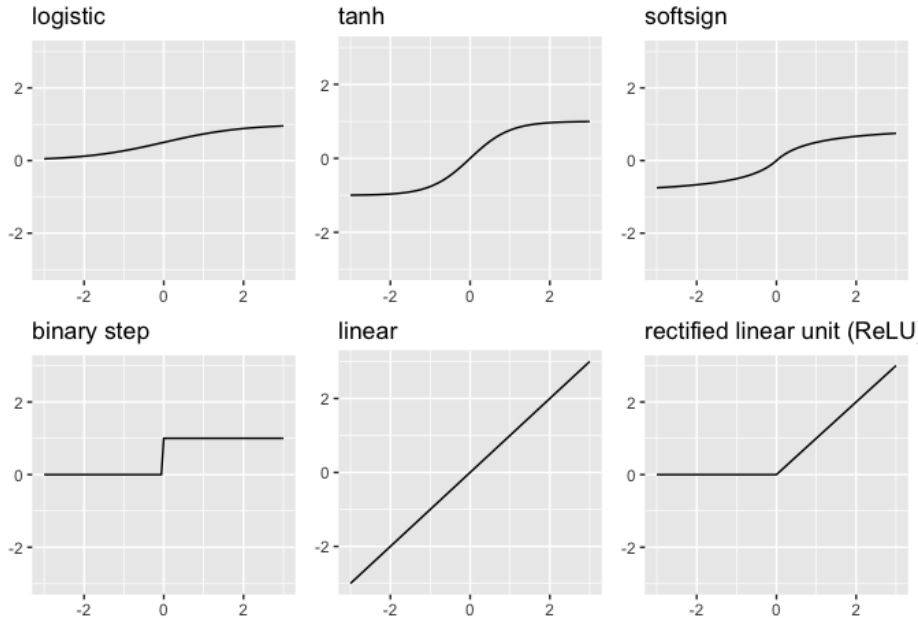


Improve Model

Activation Function

Layer에서 학습하는 결과를 Non-linear하게 나타내기 위해 사용

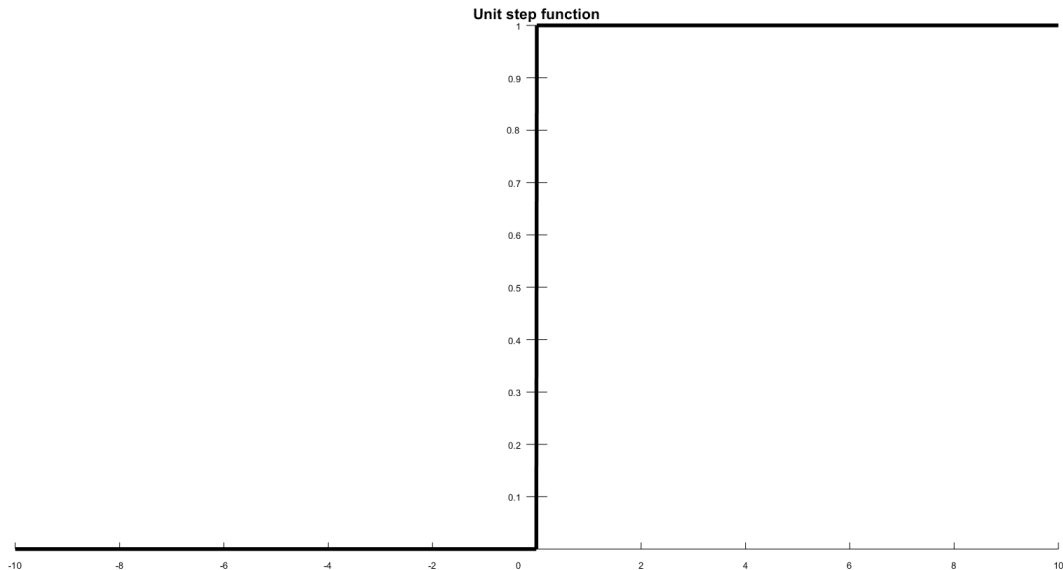
Common ANN Activation Functions



Binary(Step) Activation

Neuron의 원래 의미에 가장 가까운 Activation Function

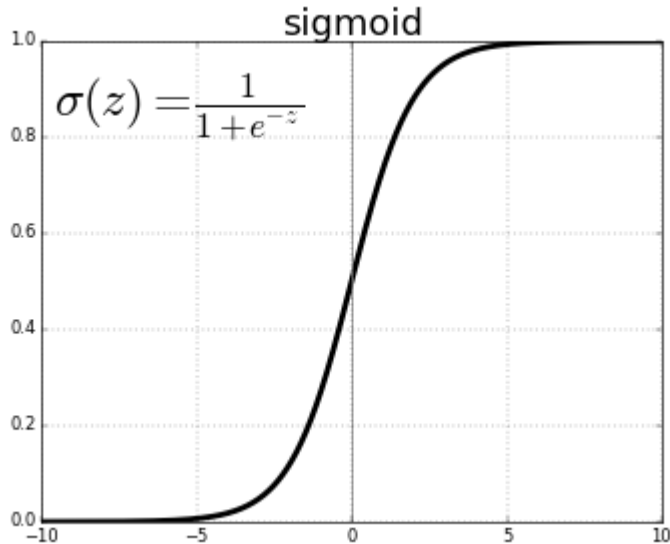
- ▶ 미분이 항상 0이기 때문에, 미분에 기반한 값 수정이 아닌 고정된 값을 이용한 update를 진행해야 함.



Sigmoid Activation

(0, 1) 사이의 실수 값으로 출력 값을 반환함.

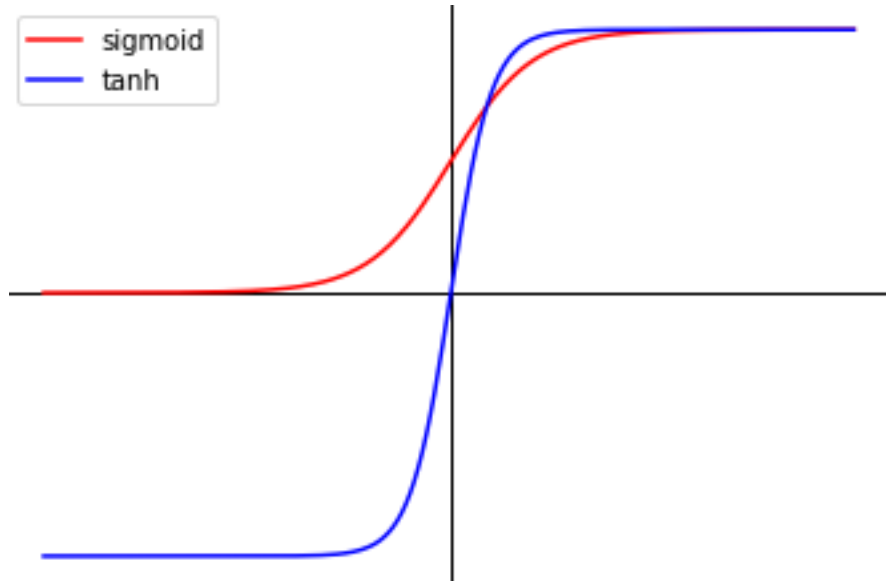
▶ 미분을 간단하게 표현할 수 있음



tanh Activation

(-1, 1) 사이의 실수 값으로 출력 값을 반환함.

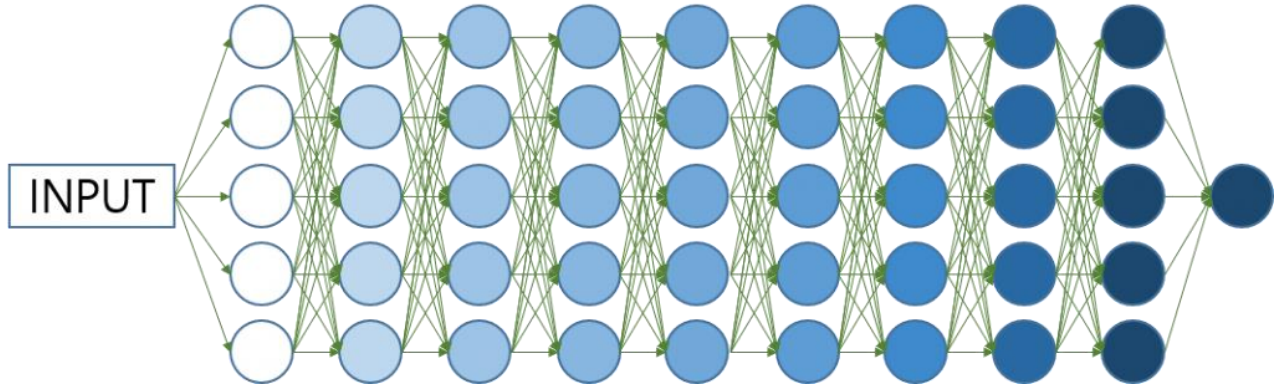
- ▶ 0에 가까운 경우, sigmoid보다 더 큰 미분 값을 가지기 때문에 weight값을 더 빠르게 수정
- ▶ 미분 역시 간단함



Vanishing Gradient Problem

Layer가 깊어질 수록 gradient 값이 점점 작아짐

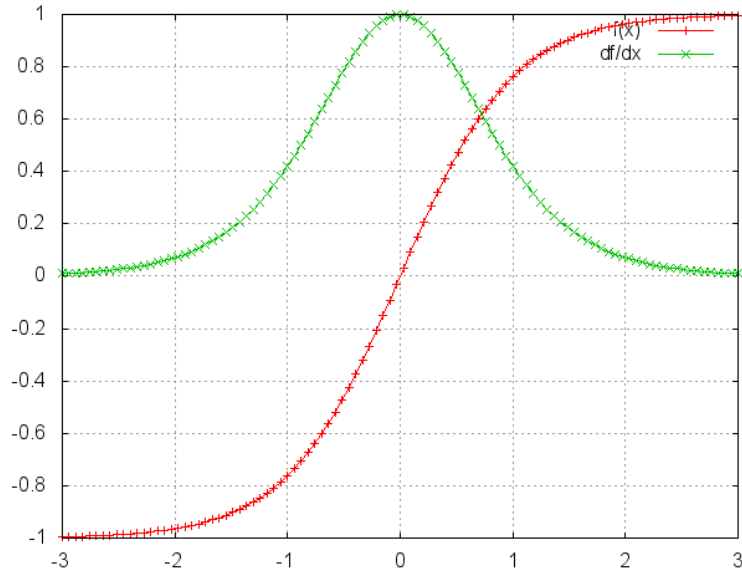
▶ Output Layer에서 멀어질 수록 weight값이 거의 업데이트 되지 않음



Vanishing Gradient Problem

Layer가 깊어질 수록 gradient 값이 점점 작아짐

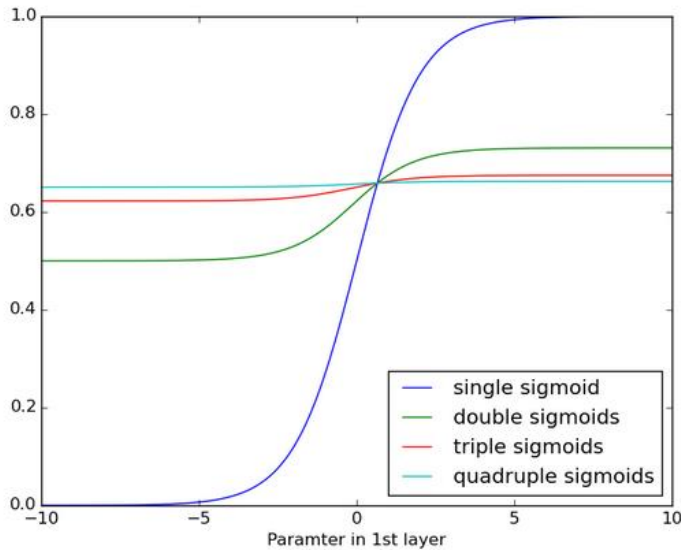
▶ Output Layer에서 멀어질 수록 weight값이 거의 업데이트 되지 않음



Vanishing Gradient Problem

Layer가 깊어질 수록 gradient 값이 점점 작아짐

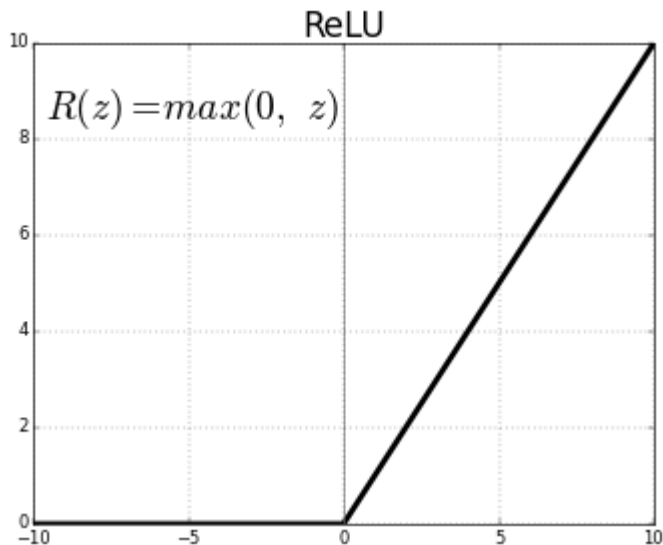
▶ Output Layer에서 멀어질 수록 weight값이 거의 업데이트 되지 않음



ReLU Activation

$(-\infty, \infty)$ 사이의 실수 값으로 출력 값을 반환함.

- ▶ 0보다 작을 경우 update X, 0보다 클 경우 항상 고정된 값으로 미분
- ▶ 0에서 미분이 불가능하다는 단점
- ▶ 초기 weight 값 지정에 영향을 많이 받음



Various ReLU Activation

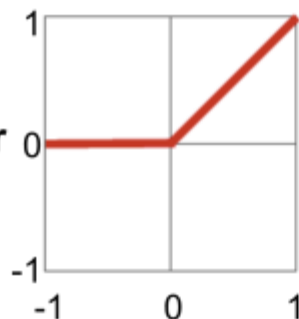
0보다 작을 경우 update X

0에서 미분이 불가능하다는 단점

▶ 두 가지 단점을 해결하기 위한 다양한 접근 방법들

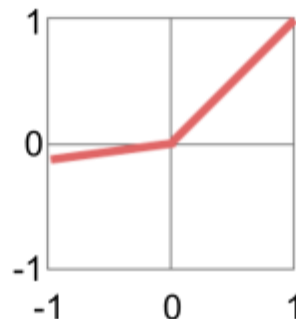
**Modern
Non-Linear
Activation
Functions**

**Rectified Linear Unit
(ReLU)**



$$y = \max(0, x)$$

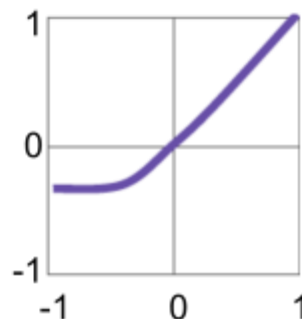
Leaky ReLU



$$y = \max(\alpha x, x)$$

α = small const. (e.g. 0.1)

Exponential LU



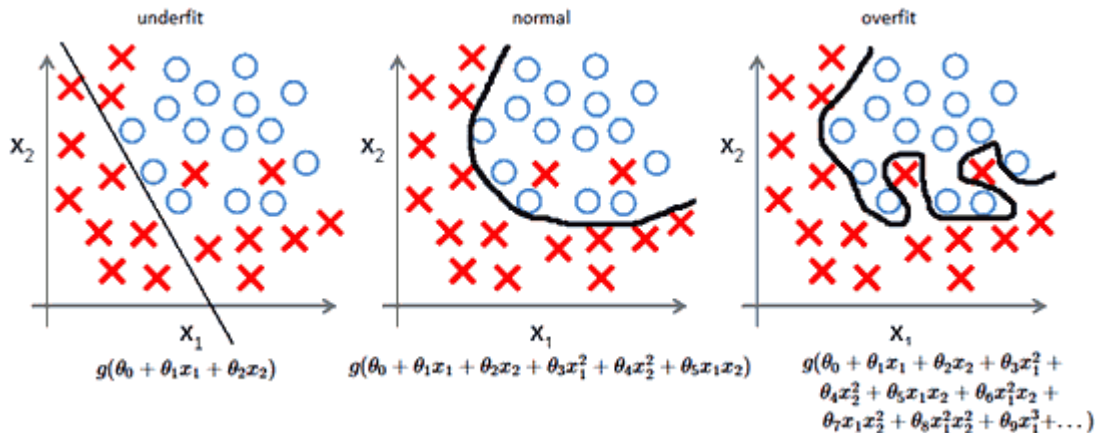
$$y = \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases}$$

Overfit and Underfit

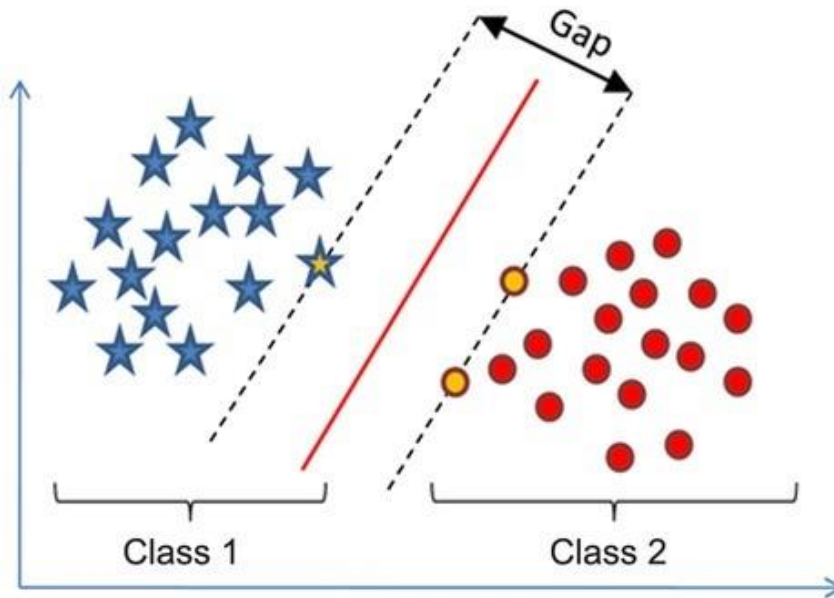
Underfit : 아직 모델이 제대로 학습되지 않았을 경우 생기는 문제

Overfit : 학습데이터를 과하게 학습할 경우 생기는 문제

▶ 두 경우에 빠지지 않았을 경우를 general한 경우라고 일컫음



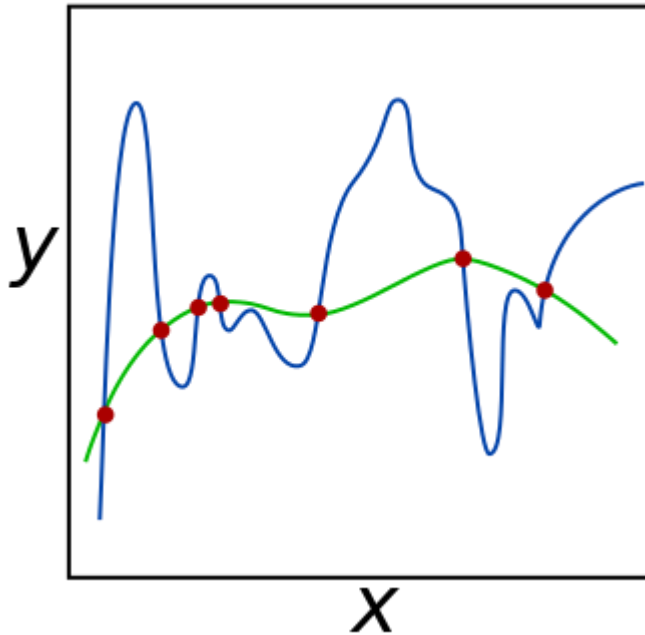
SVM and generalization



Regularization

과적합을 방지하기 위해 모델에 패널티를 부여

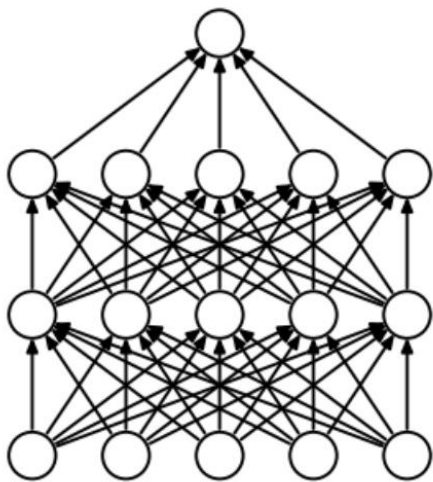
- ▶ 일반적으로 L1 또는 L2 regularization을 사용



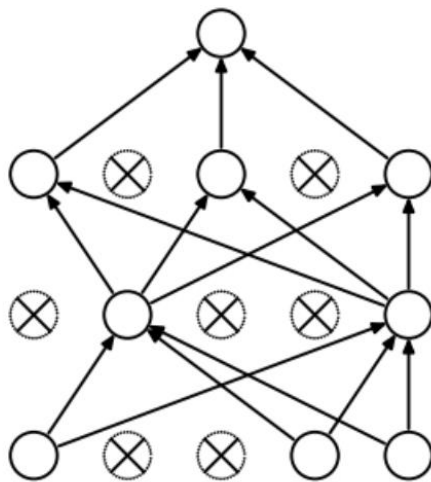
Dropout

매 epoch(mini-batch)마다 임의의 node를 제외하고 학습 진행

- ▶ Co-adaption 회피
- ▶ Ensemble 효과



(a) Standard Neural Net

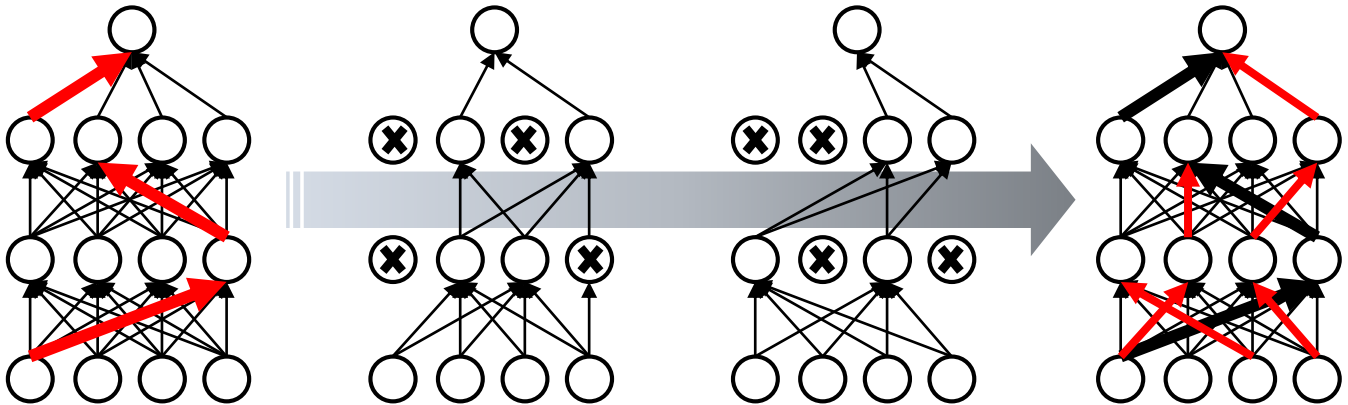


(b) After applying dropout.

Dropout

Co-adaption 회피

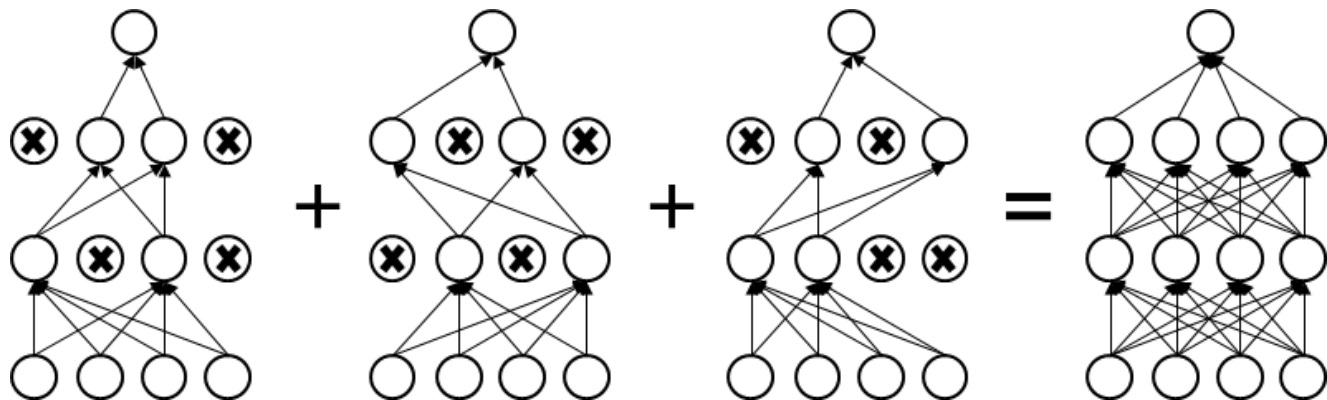
- ▶ 특정 노드의 영향력이 지나치게 강해지면 다른 노드는 학습이 진행 안 됨
- ▶ 영향력이 강한 노드를 제외시켜서 다른 노드를 학습시킴



Dropout

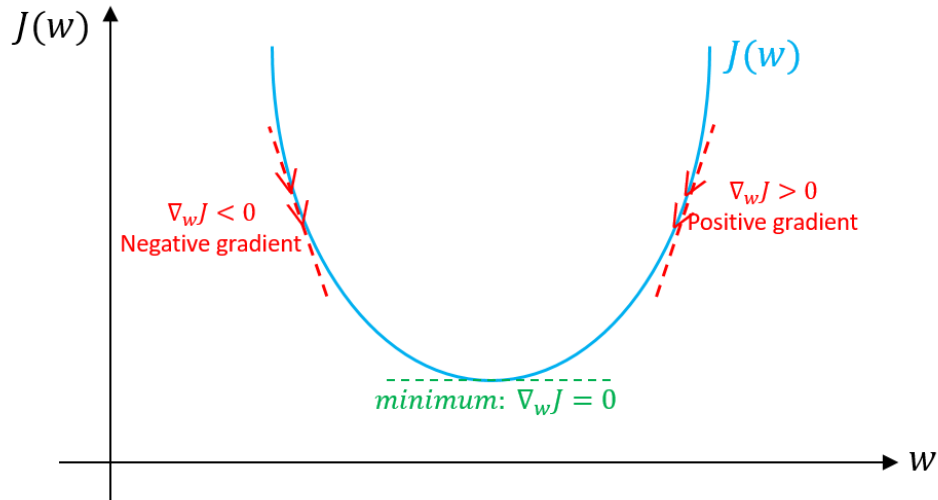
Ensemble 효과

- ▶ 각 batch에 따라 학습된 여러 neural network의 ensemble 관점으로 접근 가능



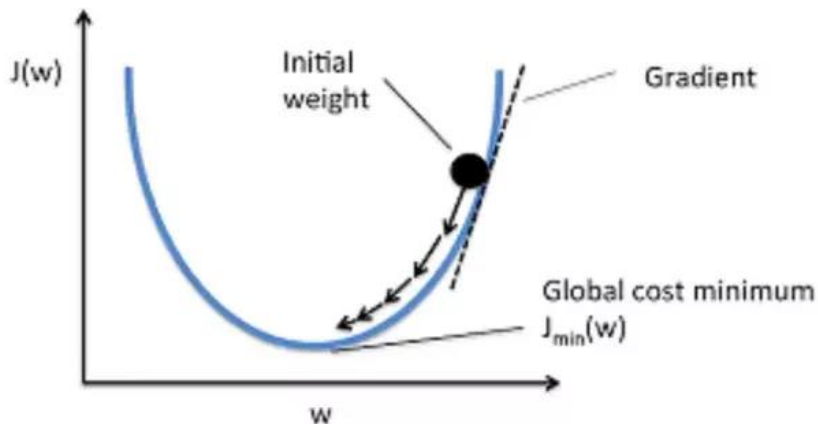
Optimizer

Error값을 최소화하는 방향으로 model을 학습시키는 기법



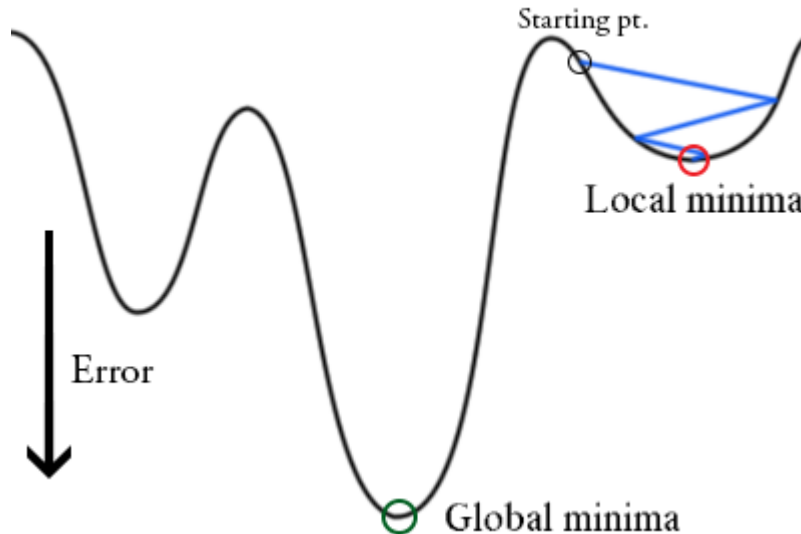
Gradient descent

매 epoch마다 error값을 계산하고, error의 미분 값에 따라 error값이 점점 작아지는(또는 커지는)형태로 모델을 수정하는 기법



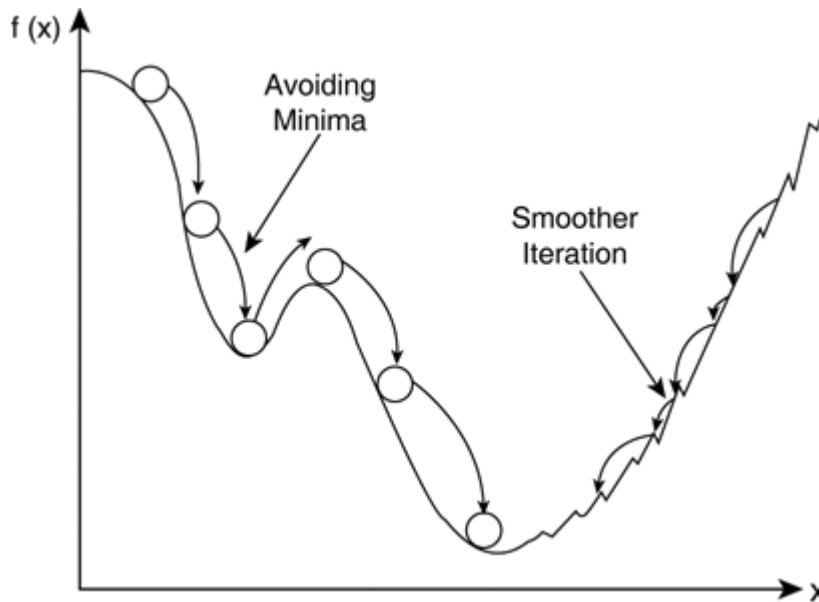
Global Minima vs Local Minima

Error가 작아지는 방향으로 내려가기 때문에, 지역 최소값에 빠지는 문제점이 발생



Momentum Optimizer

가속도 값을 이용해서 원래 이동하려는 방향으로 계속 나아가려는 성질을 통해, 얕은 local minima를 빠져나갈 수 있도록 학습



Various Optimizer

그 외에 다양한 optimizer 기법들이 존재

▶ (2017년도 기준) Adagrad기법 혹은 Rmsprop가 성능이 뛰어나다는 실험결과들이 있음

