

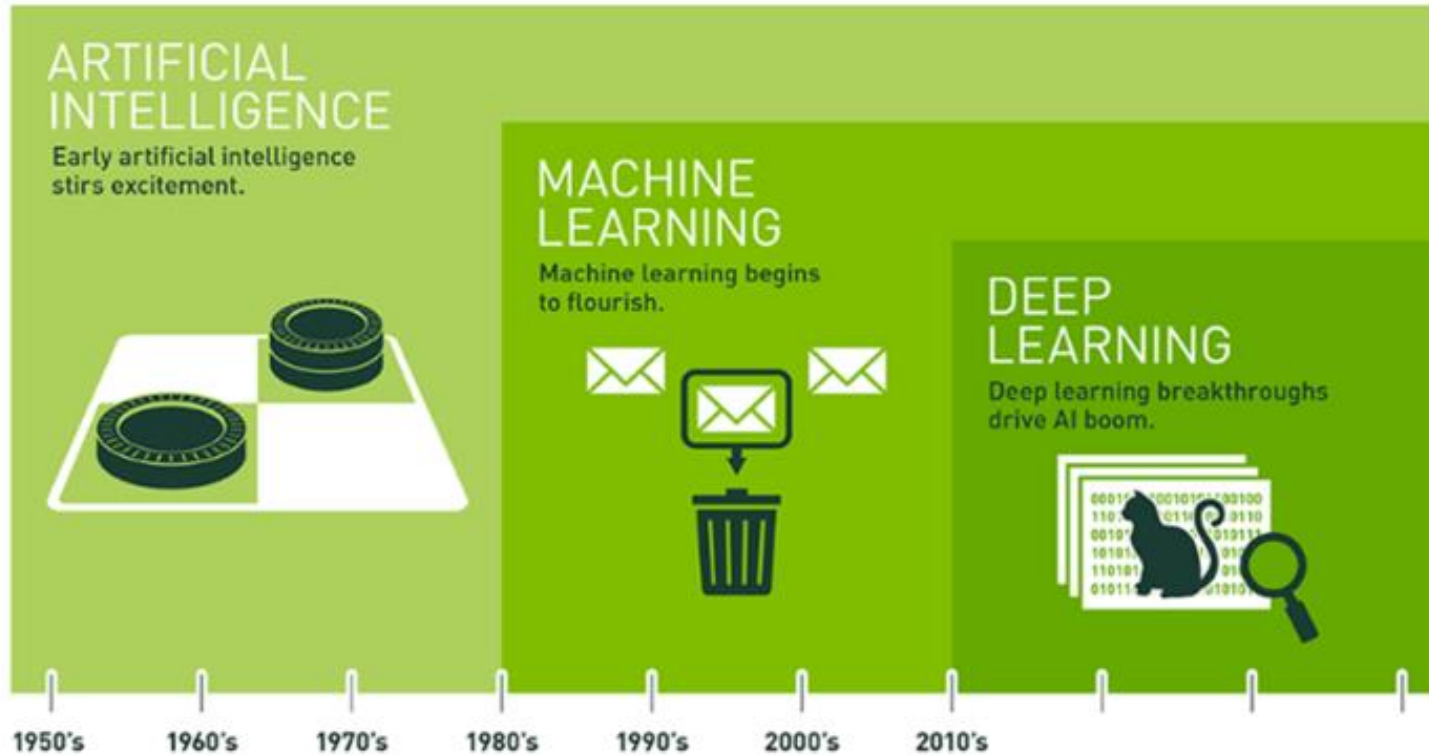
딥러닝 필수개념

Text

010010



머신러닝과 딥러닝의 차이점



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

■ **인공 지능: 인간의 지능을 기계로 구현하다**

인간의 감각, 사고력을 지닌 채 인간처럼 생각하는 인공 지능이라고 하며, 현재의 기술 발전 수준에서 만들 수 있는 인공지능은 Narrow AI의 개념에 포함됩니다.

■ **머신 러닝: 인공 지능을 구현하는 구체적 접근 방식**

알고리즘을 이용해 데이터를 분석하고, 분석을 통해 학습하며, 학습한 내용을 기반으로 판단이나 예측을 합니다.

인공 지능을 구현하는 과정 전반에 일정량의 코딩 작업이 수반된다는 한계점이 있습니다.

■ **딥러닝: 완전한 머신 러닝을 실현하는 기술**

딥러닝은 인공신경망에서 발전한 형태의 인공 지능으로, 뇌의 뉴런과 유사한 정보 입출력 계층을 활용해 데이터를 학습합니다.

기본적인 신경망도 많은 양의 연산을 필요로 합니다.

학습(learning)

■ 학습(learning)

- 데이터를 특별한 알고리즘에 적용해 머신러닝 모델의 정의된 문제에 최적화 하는 과정

■ 지도학습(supervised learning)

- 학습시 정답을 알려 주면서 진행하는 학습으로, 학습시 데이터와 레이블(정답)이 함께 제공됩니다.
- 레이블(Label) = 정답, 실제값, 타겟, 클래스, y
- 예측된 값 = 예측값, 분류값, \hat{y} (y hat)
- 데이터마다 레이블을 달기 위해 많은 시간을 투자해야 합니다.
- 지도학습의 대표적인 예는 분류(이진분류, 다중분류)와 회귀(주가예측 등)가 있습니다.

■ 비지도학습(unsupervised learning)

- 레이블(정답) 없이 진행되는 학습으로, 데이터 자체에서 패턴을 찾아내야 할 때 사용합니다.
- 비지도학습의 대표적인 예는 군집화와 차원축소가 있다.

■ 모델(model)

- Binary Classification Model, Multiclass Classification Model, Regression Model

모델 평가

스켈터랩스는 자사의 AI 기반 대화엔진의 성능이 질문의 의도를 파악하는 테스트에서 글로벌 기업보다 더 나은 정확도를 나타냈다고 전했다. 스켈터랩스는 6월 한국정보화진흥원이 AI 학습용으로 공개한 250만건 데이터 중 소상공인 분야에 대한 총 10만건의 한국어 대화 데이터를 활용해 테스트를 실시했다. 총 600개의 인텐트(intent, 질문의 의도)를 대상으로 스켈터랩스와 글로벌 기업 A, B사의 엔진을 비교했다.

스켈터랩스에 따르면 스켈터랩스의 대화엔진은 72%, 글로벌 A, B사의 대화엔진은 65%의 F1 스코어를 기록했다. F1 스코어는 정밀도 및 재현율을 모두 종합한 값으로, F1 스코어가 높다는 것은 그만큼 질문의 의도를 정확히 파악한다는 것을 의미한다.

스켈터랩스는 자사의 대화엔진이 규칙 기반 의도 분류와 머신러닝 기반 의도 분류를 함께 활용한 하이브리드 모델을 적용해 타사 엔진 대비 높은 정확도를 기록할 수 있었다고 설명했다.

대화형 AI 엔진 F1 스코어 '72%'

혼동 행렬(Confusion Matrix)

모델의 성능을 평가할 때 사용되며, 대략적인 성능확인과 모델의 성능을 혼동행렬을 기반으로 수치로 표현할 수 있습니다.

True Positive(TP) : 실제 True인 정답을 True라고 예측 (정답)

True Negative(TN) : 실제 False인 정답을 False라고 예측 (정답)

False Positive(FP) : 실제 False인 정답을 True라고 예측 (오답)

False Negative(FN) : 실제 True인 정답을 False라고 예측 (오답)

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

TP(True Positive)

데이터를 입력했을 때 데이터의 실제값을 올바르게 예측한 케이스로 혼동행렬의 대각선 부분이 TP입니다.

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

TP(True Positive) : 맞는것을 올바르게 예측한 것

$$\text{정확도(Accuracy)} = (9 + 15 + 24 + 15) / 80 = 0.78$$

TN(True Negative)

틀린 것을 올바르게 예측한 것을 TN(True Negative)이라고 하며, 각 클래스별로 TN이 따로 있습니다.

■ A클래스의 TN : A가 아닌 클래스들을
A가 아니라고 예측한 모든 값

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

■ B클래스의 TN : B가 아닌 클래스들을
B가 아니라고 예측한 모든 값

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

TN(True Negative) - 틀린것을 올바르게 예측한 것

FP(False Positive)

틀린 것을 맞다고 예측한 것을 FP(False Positive)라고 하며, 각 클래스별로 FP가 있습니다.

■ A클래스의 FP

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

■ B클래스의 FP

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

- FP(False Positive) - 틀린것을 맞다고 **잘못** 예측한 것

FN(False Negative)

맞는 것을 틀렸다고 잘못 예측한 것을 FN(False Negative)라고 하며, 각 클래스별로 FN이 있습니다.

■ A클래스의 FN

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

TP, TN은 잘 한 경우,
FP, FN은 잘못된 경우입니다.

그런데 FP, FN 중에 뭐가 낫냐고 하면,
그건 Case By Case 입니다.

예를 들어
암이 아닌데 암이라고 진단하거나(False Positive),
암인데 암이 아니라고 진단하는(False Negative) 케이스는
어떤 경우가 더 나쁠까요?

- FN(False Negative)- 맞는 것을 틀렸다고 잘못 예측한 것

모델성능 평가지표

■ 정밀도(Precision)

모델이 True라고 분류한 것 중에서 실제 True인 것의 비율
날씨 예측 모델이 맑다로 예측했는데, 실제 날씨가 맑았는지는 나타낸 지표입니다.

$$(Precision) = \frac{TP}{TP + FP}$$

■ 재현율(Recall)

실제 True인 것 중에서 모델이 True라고 예측한 것의 비율
실제 날씨가 맑은 날 중에서 모델이 맑다고 예측한 비율을 나타낸 지표입니다.

$$(Recall) = \frac{TP}{TP + FN}$$

■ 정확도(Accuracy)

가장 직관적으로 모델의 성능을 나타낼 수 있는 평가 지표
혼동행렬상에서는 대각선(TP)을 전체 셀로 나눈 값에 해당합니다.
한달 동안에 맑은 날이 28일이고 비가 오는날이 이틀인 경우, 비가 오는 것을 예측하는 성능은 매우 낮을 수 밖에 없으므로 이를 보완할 지표가 필요합니다.

$$(Accuracy) = \frac{TP + TN}{TP + FN + FP + TN}$$

■ F1 점수(F1 score)

정밀도와 재현율의 조화평균입니다.

$$(F1-score) = 2 \times \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

텐서(Tensor)

딥러닝은 텐서(Tensor)를 기반으로 진행되는 연산의 연속이며, 텐서는 다차원 배열을 처리하기 위한 데이터 구조입니다.

■ tensor

rank	type	example
0	scalar	[1]
1	vector	[1,1]
2	matrix	[[1,1], [1,1]]
3	3 tensor	[[[1,1], [1,1]], [[1,1], [1,1]]]
n	N tensor	

■ tensor example in NLP

sentence	vector representation
hi John	[[1,0,0,0], [0,1,0,0]]
hi James	[[1,0,0,0], [0,0,1,0]]
hi Brian	[[1,0,0,0], [0,0,0,1]]

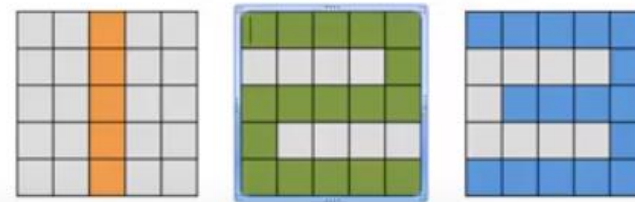
Unique word dictionary

word	index	one hot encoding vector
hi	0	[1,0,0,0]
John	1	[0,1,0,0]
James	2	[0,0,1,0]
Brian	3	[0,0,0,1]

■ tensor example in rgb color image

(3, 5, 5, 3) 4d tensor!

you have 3 images 5 rows 5 columns red, green, blue



■ tensor example in rgb color video

(3, 5, 5, 5, 3) 5d tensor!

you have 3 frames 5 images 5 rows 5 columns red, green, blue

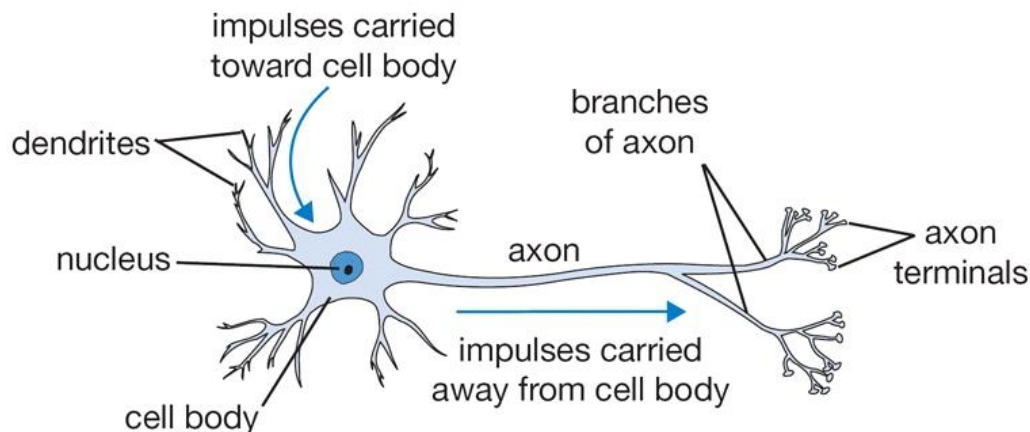


퍼셉트론(Perceptron)

신경망(딥러닝)의 기원이 되는 알고리즘으로 다수의 신호(흐름이 있는)를 입력으로 받아 하나의 신호를 출력합니다.

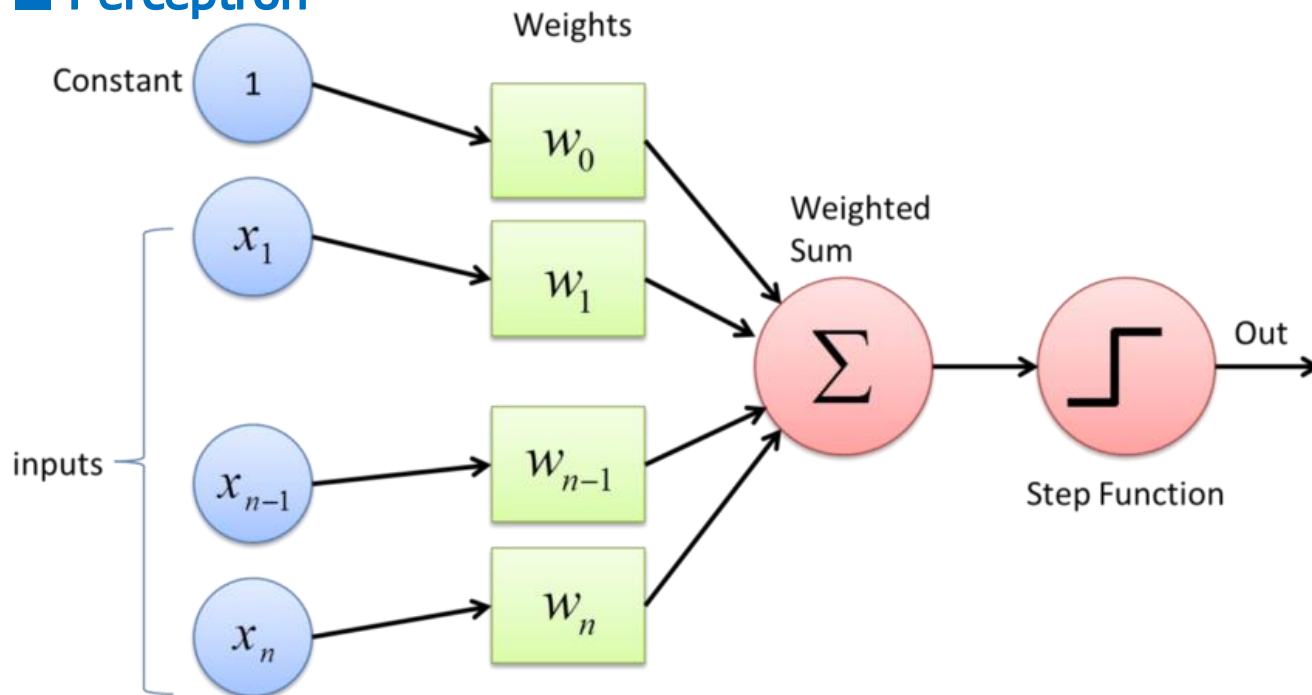
■ Neuron

인간의 신경계에서 약 860 억 개의 뉴런이 있으며, 대략 $10^{14} \sim 10^{15}$ 개의 시냅스와 연결되어 있습니다.



- axon (축삭돌기) : 팔처럼 몸체에서 뻗어나와 다른 뉴런의 수상돌기와 연결됩니다.
- dendrite (수상돌기) : 다른 뉴런의 축삭 돌기와 연결되며, 몸체에 나뭇가지 형태로 붙어 있습니다.
- synapse (시냅스) : 축삭돌기와 수상돌기가 연결된 지점입니다. 여기서 한 뉴런이 다른 뉴런으로 신호가 전달됩니다.

■ Perceptron

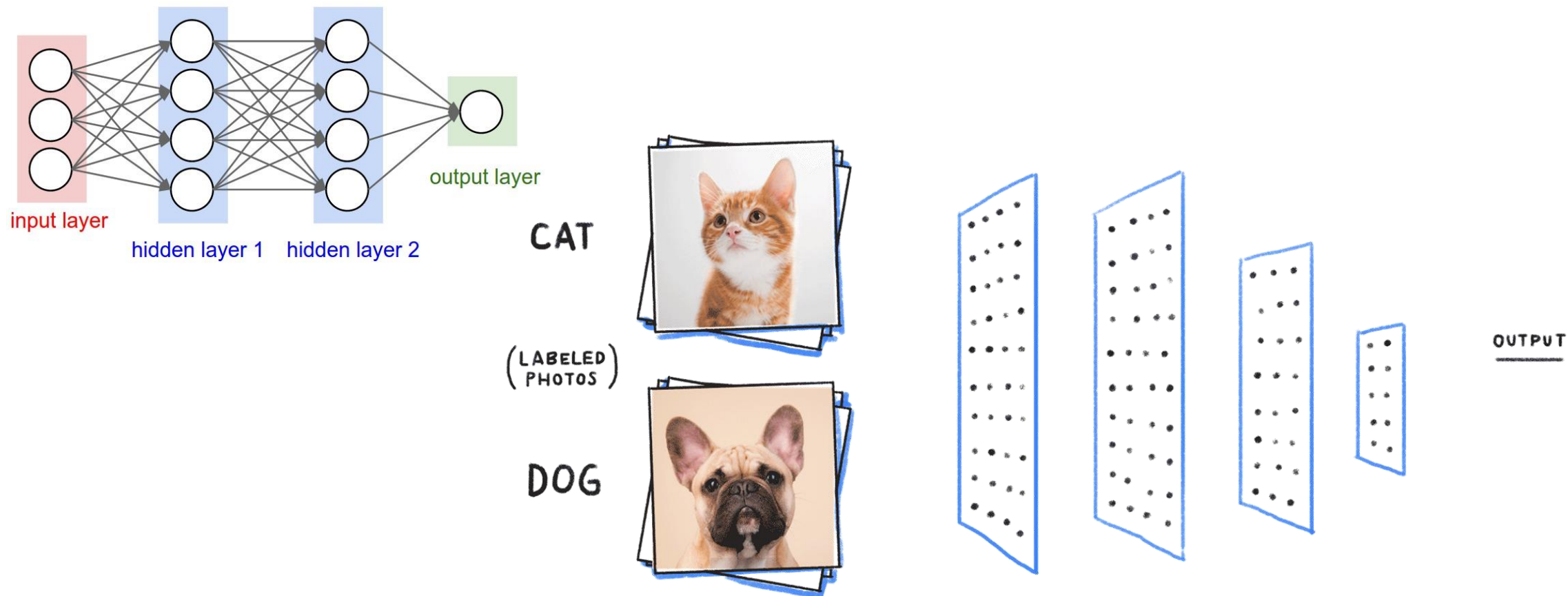


The perceptron consists of 4 parts.

1. Input values or One input layer
2. Weights and Bias
3. Net sum
4. Activation Function

신경망(Neural Network)

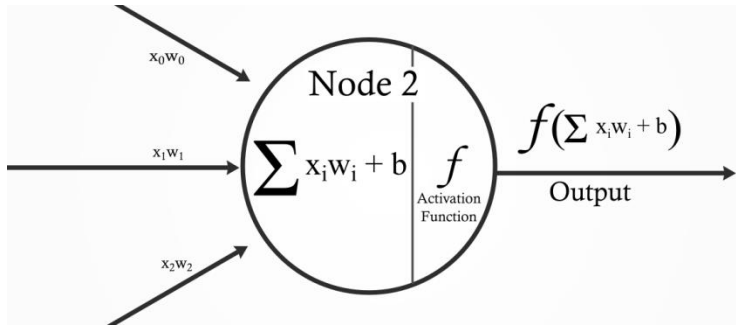
퍼셉트론은 단일 레이어 신경망이며, 레이어를 여러 개 연결해서 구성한 멀티 레이어 퍼셉트론을 신경망이라고 합니다. 구성에 따라 X-NN이라 하며 CNN, RNN이 대표적입니다.



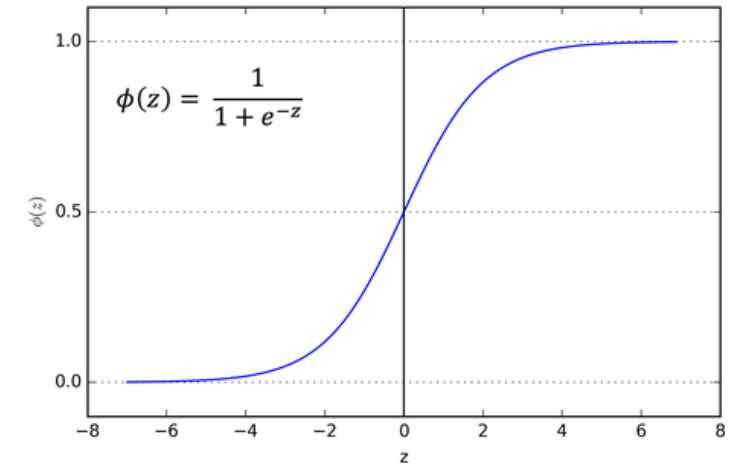
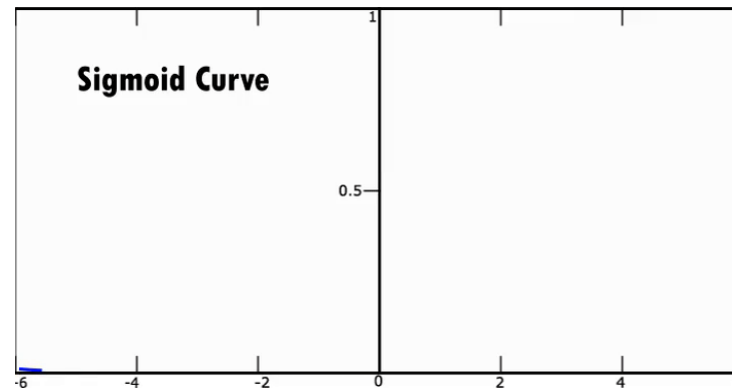
활성화 함수(Activation function)

‘예’ 또는 ‘아니오’와 같은 신경망의 출력을 결정하는 데 사용됩니다. 결과 값을 0-1 또는 -1-1 등으로 매핑합니다(함수에 따라 다름)

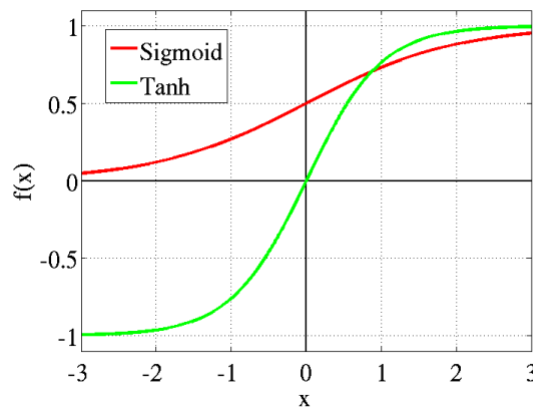
■ Activation Function



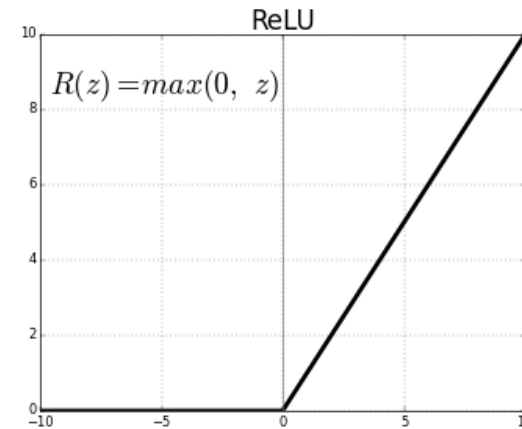
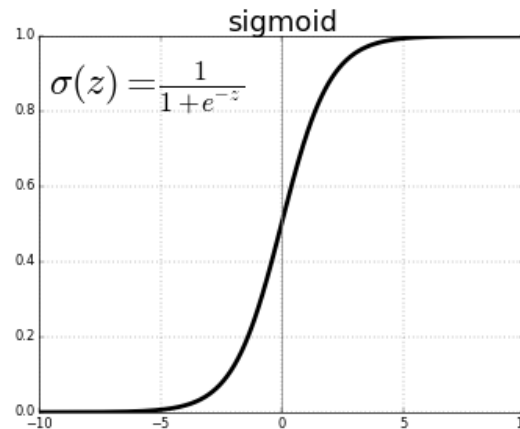
■ Sigmoid



■ tanH



■ ReLU (Rectified Linear Unit)



딥러닝(Deep Learning)

딥러닝 학습의 목표는 모델에 입력값을 넣었을 때의 출력값이 최대한 정답과 일치하게 하는 것입니다.

■ 딥러닝의 학습방법

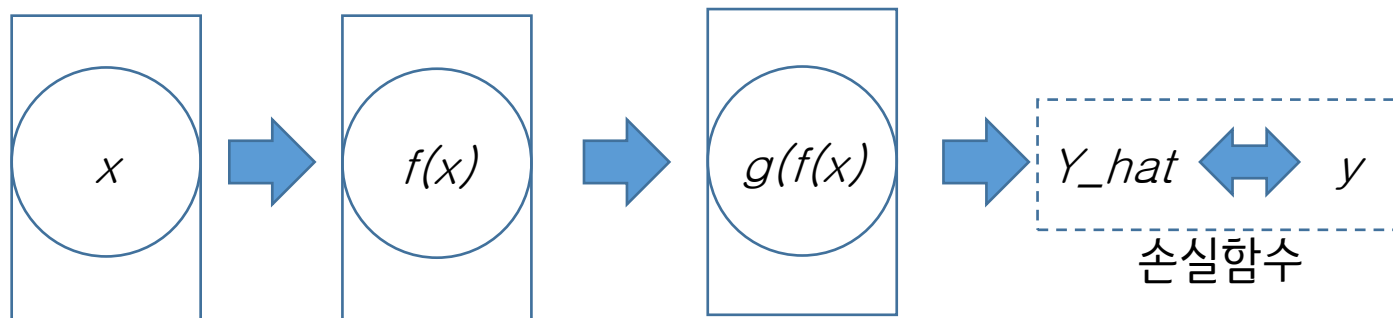
- 보통 최초 딥러닝 모델의 매개변수(weight, bias)를 무작위로 부여한 후, 반복학습을 통해 모델의 출력값을 최대한 정답과 일치하도록 매개변수를 조금씩 조정한다.

■ 순전파(Forward Propagation)

- 딥러닝 모델에 값을 입력해서 출력을 얻는 과정

■ 오차 역전파(Error Backpropagation)

- 결과 값을 통해서 다시 역으로 input 방향으로 오차를 다시 보내며 가중치를 재업데이트 하는 과정



■ 손실함수(Loss Function)

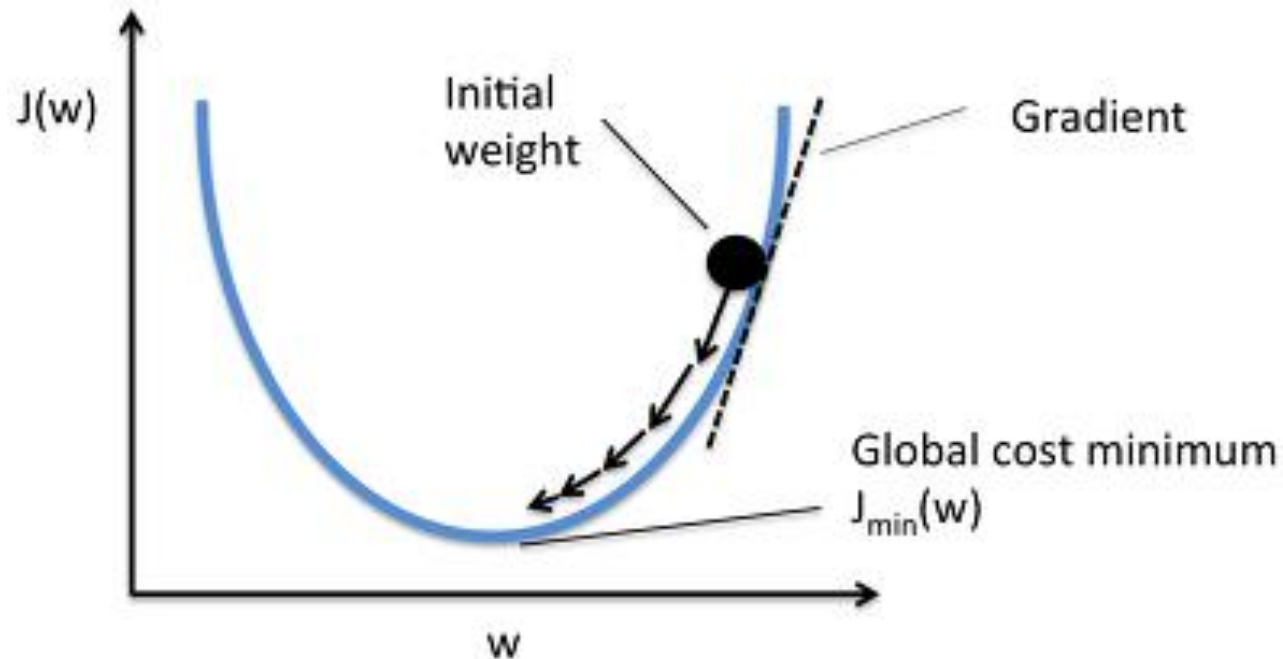
- 손실함수는 신경망 학습의 목적으로(목적함수) 출력값과 정답의 차이를 계산한다.
- 보통 회귀에는 평균제곱오차(MSE)를, 분류문제에는 크로스 엔트로피(신경망 출력을 확률로 간주할 수 있는 경우에 사용)를 손실함수로 사용한다.

■ 최적화(Optimization)

- 딥러닝 모델의 매개변수(weight, bias)를 조절해서 손실함수의 값을 최저로 만드는 과정
- 대표적인 방법은 경사하강법(Gradient Descent)이다.

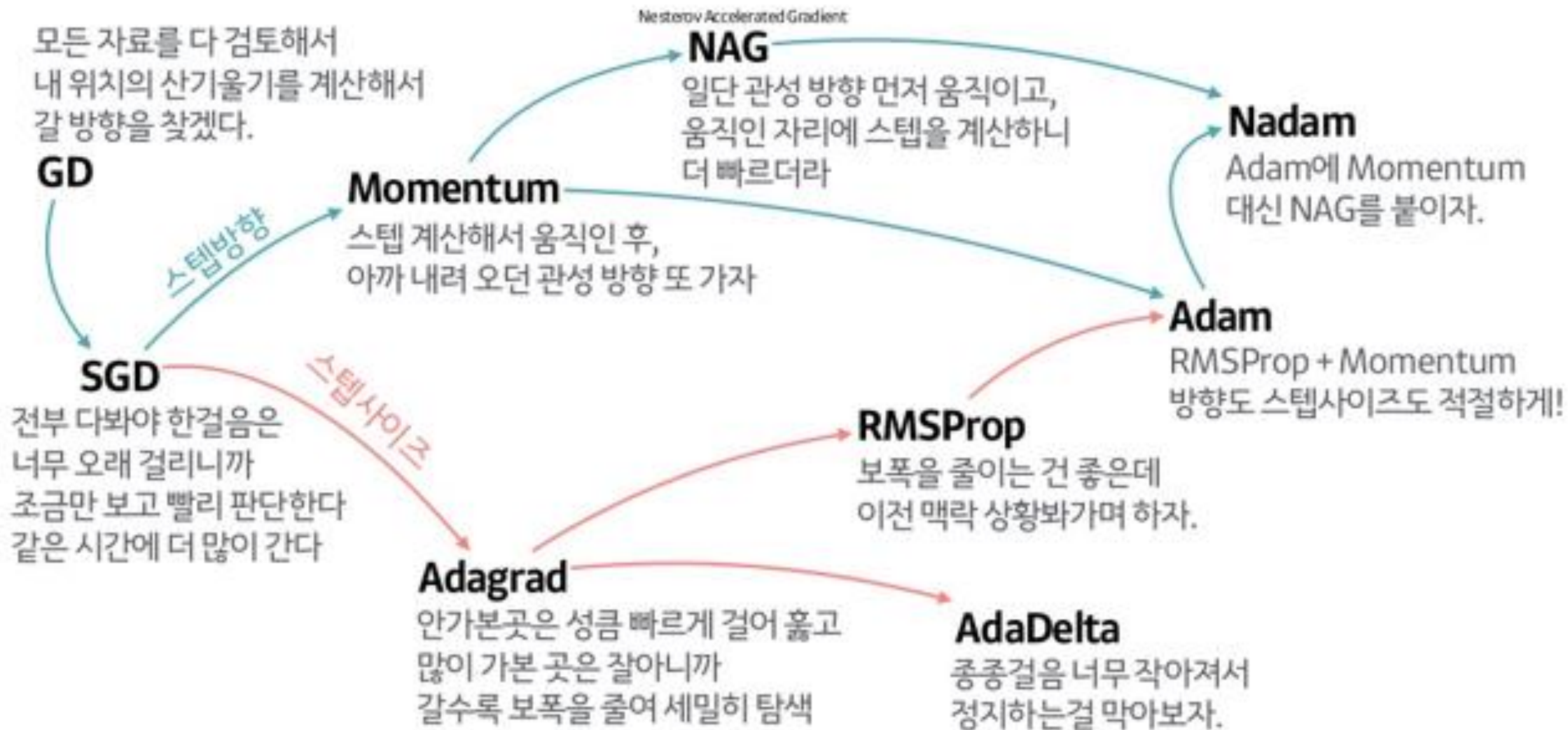
경사하강법(Gradient Descent)

반복적으로 손실함수에 대한 모델 매개변수의 미분값을 구한 후, 그 미분값을 반대 방향으로 매개변수를 조절해 나가면 결국에는 최저 손실함수값에 도달하게 합니다.

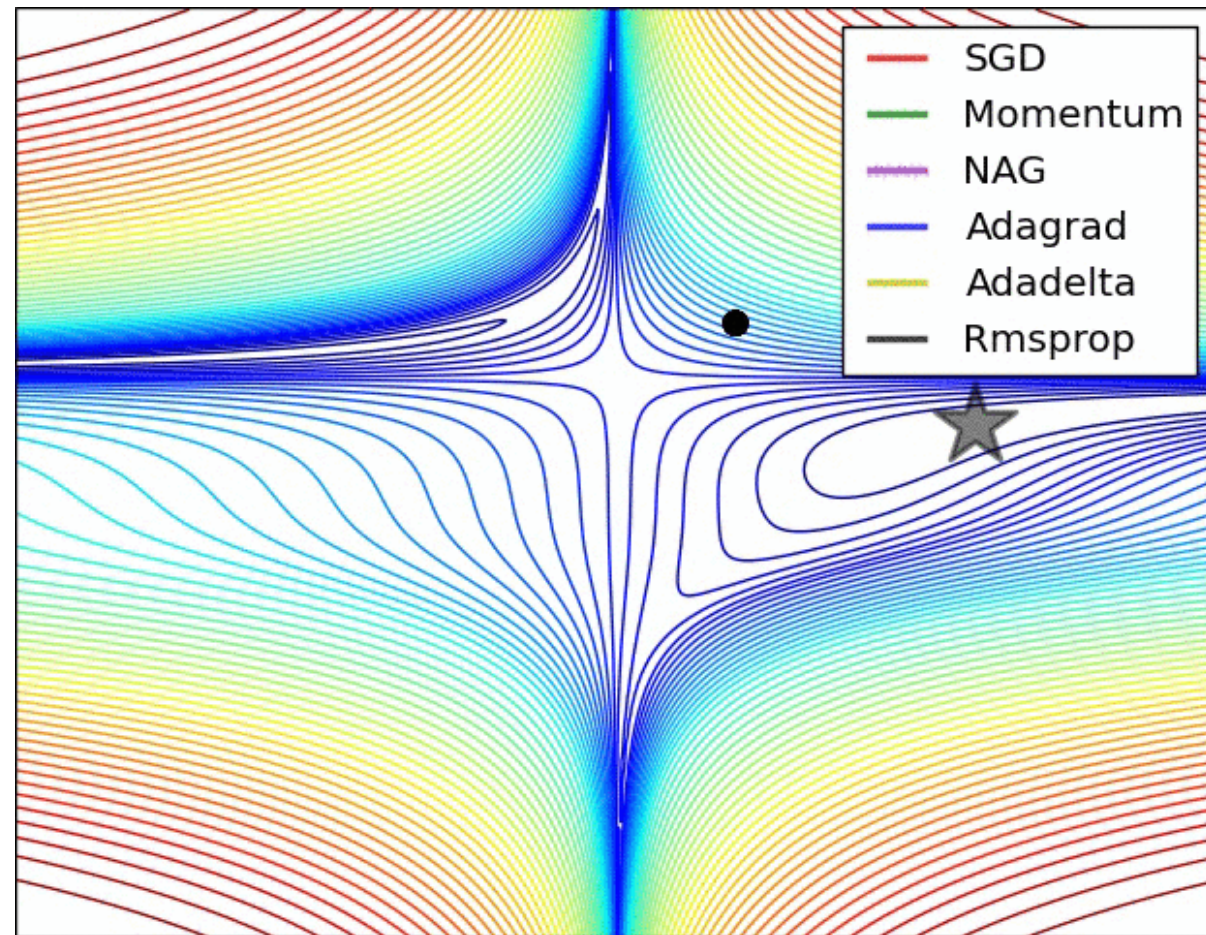
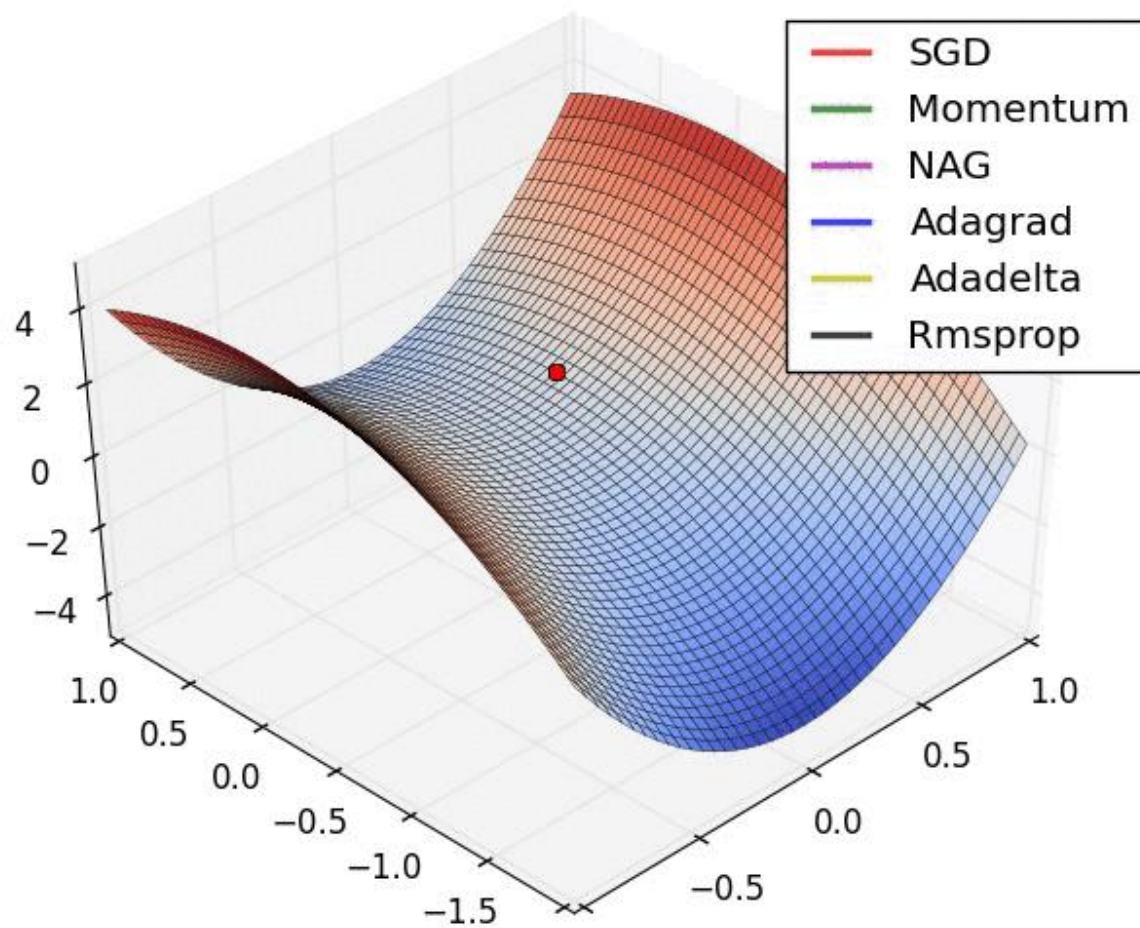


$$\text{weight의 업데이트} = \text{에러 낮추는 방향 (decent)} \times \text{한발자국 크기 (learning rate)} \times \text{현 지점의 기울기 (gradient)}$$
$$- \gamma \nabla F(\mathbf{a}^n)$$

최적화 알고리즘(Optimization Algorithm)



최적화 알고리즘(Optimization Algorithm)



Thank you