Undergraduate Research Internship in Affective AI LAB.

혼자 공부하는 머신러닝 & 딥러닝

6주차 : ch07. 딥러닝 개요

컴퓨터공학과 12201682 공지윤

목차

인공신경망 (ANN)

- 퍼셉트론 (Perceptron)
- 다층신경망 (MLP)

딥러닝 학습

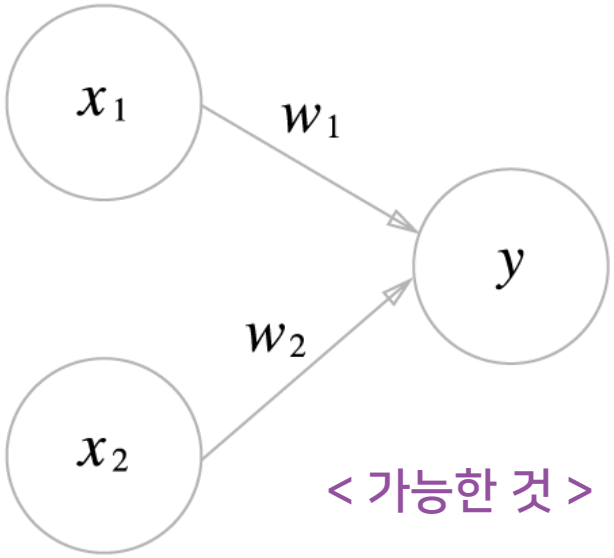
- 경사하강법
- 오차역전파법 (Backpropagation)
- 활성화 함수
- 최적화 (Optimization)

Keras Framework

- Multi-backend
- Tensorflow 2.x

1. 인공신경망 (ANN)

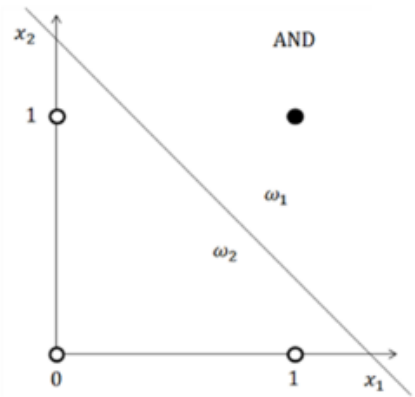
[퍼셉트론 (Perceptron)]

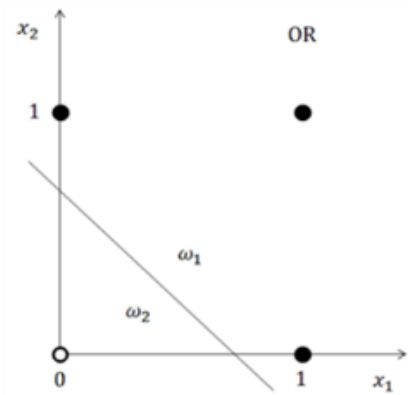


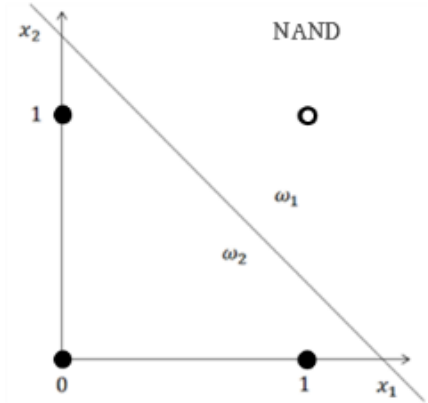
- Perceptron = Perception + Neuron
  - 인공 신경망의 구성 요소
- 다수의 값을 입력 받아 정해진 규칙에 따라 하나의 값으로 출력하는 알고리즘

< 가능한 것 >

- AND, OR, NAND와 같은 선형 영역 표현
- 직선 분리 문제

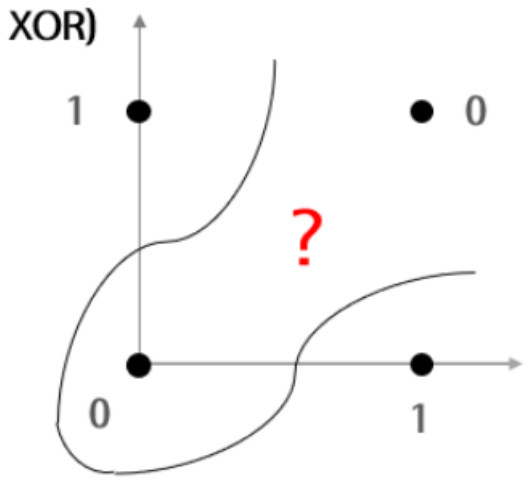






< 불가능한 것 >

- XOR과 같은 비선형 영역 표현



1. 인공신경망 (ANN)

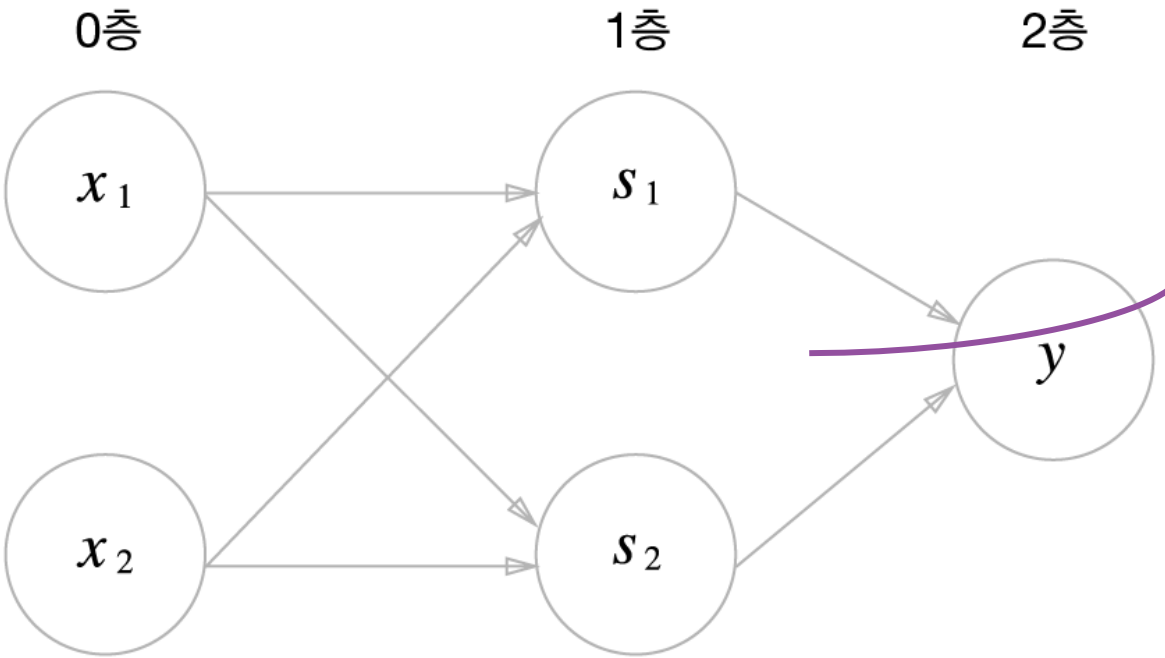
[퍼셉트론 (Perceptron)]

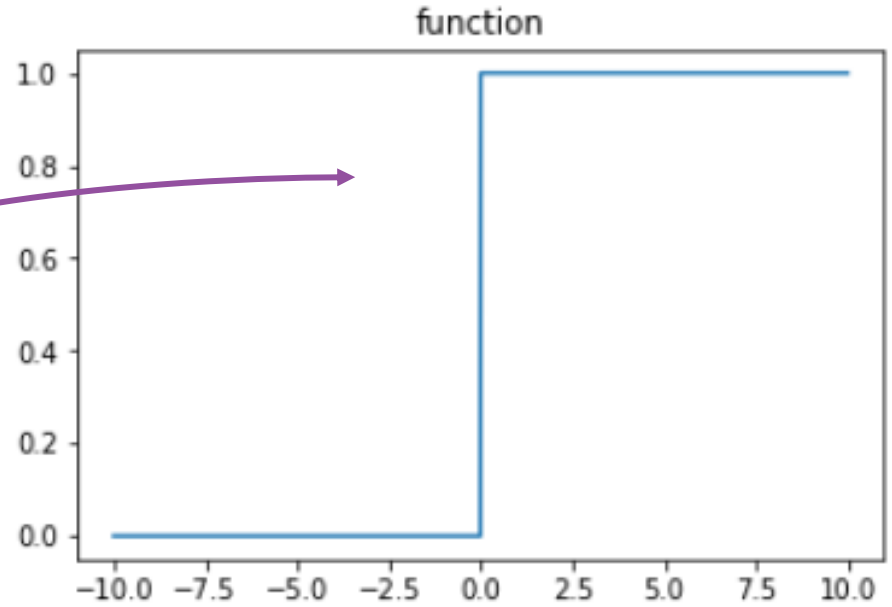
< 단층 퍼셉트론 >

- 비선형 영역을 표현할 수 없다는 한계가 존재

< 다층 퍼셉트론 >

- 입력층 + 은닉층 + 출력층으로 구성
- 활성화 함수 : 선형 방정식 (계단함수)



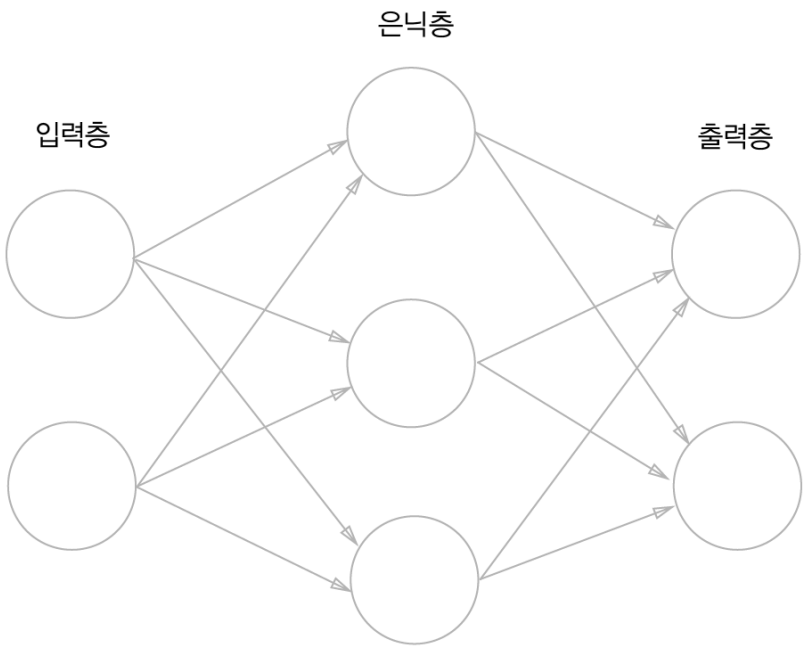


1. 인공신경망 (ANN)

[퍼셉트론 (Perceptron)]

< 단층 퍼셉트론 >

- 비선형 영역을 표현할 수 없다는 한계가 존재



< 다층 퍼셉트론 >

- 입력층 + 은닉층 + 출력층으로 구성
- 활성화 함수 : 선형 방정식 (계단함수)

< 신경망 >

- 좀 더 복잡한 형태의 다층 퍼셉트론
- 활성화 함수 : Sigmoid, ReLU, Softmax

다층 퍼셉트론과 신경망이 동일하다고 보는 곳도 많아서 굳이 구분하지 않아도 될 듯!

1. 인공신경망 (ANN)

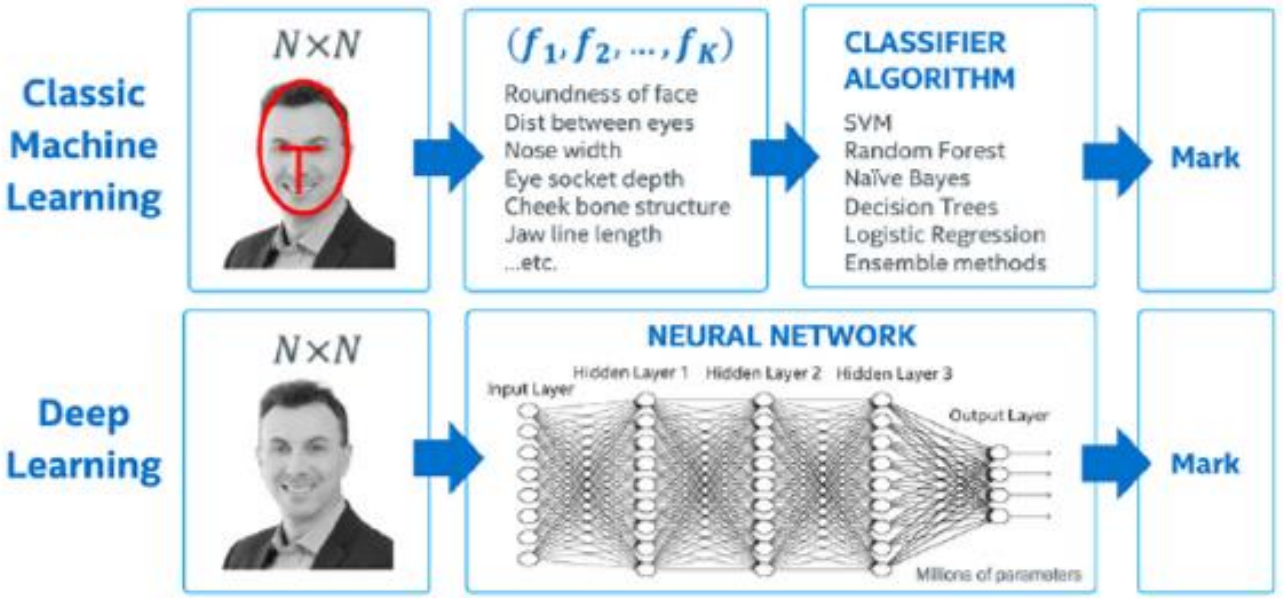
[머신러닝 vs. 딥러닝]

< 머신러닝 >

- Feature Extraction : 사람이
- Train : 기계가

< 딥러닝 >

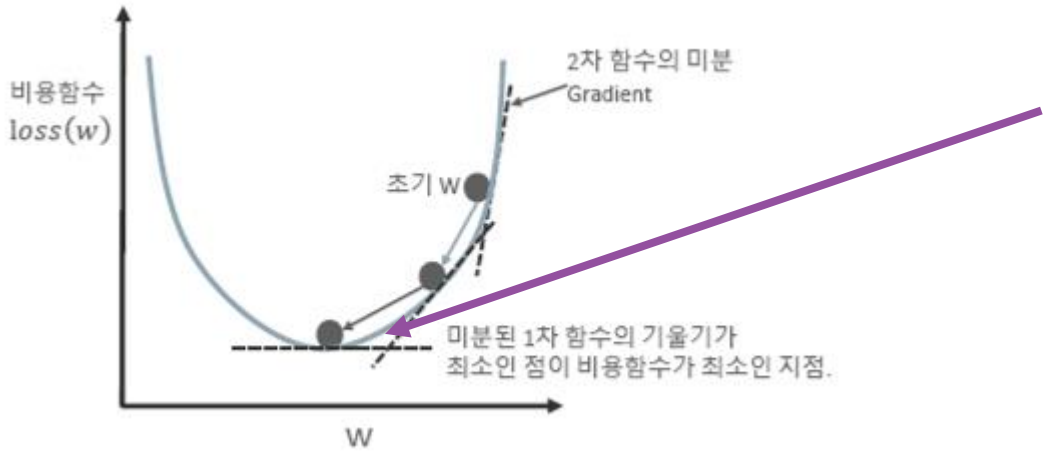
- Feature Extraction : 기계가
- Train : 기계가



2. 딥러닝 학습

[경사하강법]

- 딥러닝의 가중치의 개수는 매우 많기 때문에, 고차원의 방정식을 사용하더라도 최소가 되는 가중치를 구하는 것은 어려운 일이다.
- 경사하강법으로 점진적인 반복적 계산을 통해 오류값의 최소를 향해서 하강한다.
- 비용함수의 미분값은 증가 또는 감소의 방향성을 나타낸다.



미분된 1차함수의 기울기가 감소하지 않는 지점을 비용함수가 최소인 지점으로 간주.

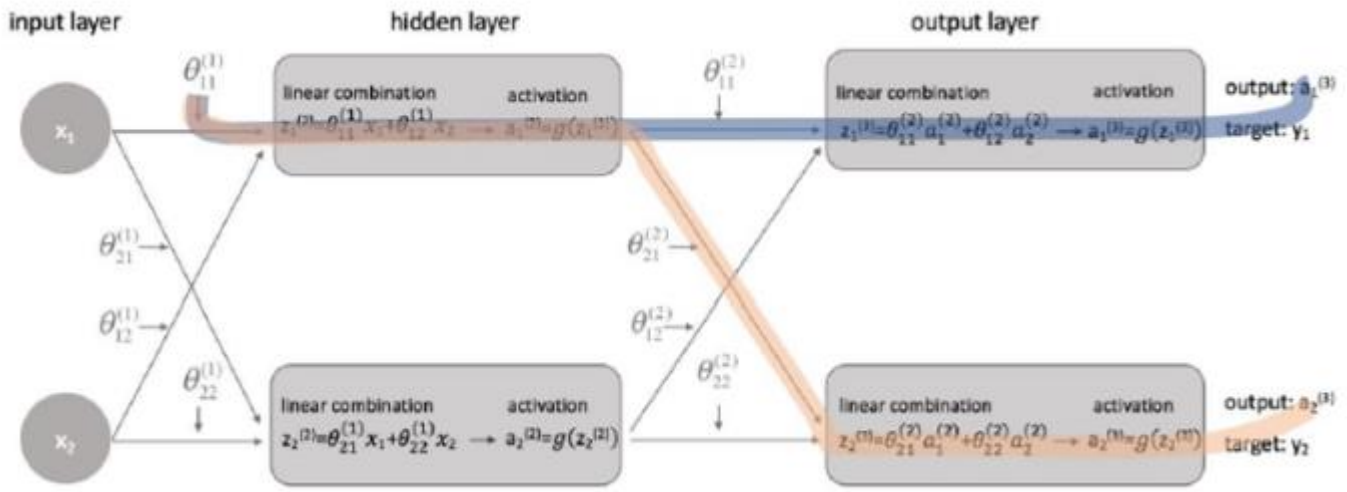
2. 딥러닝 학습

[오차역전파법]

- 출력층부터 역순으로 Gradient를 전달하여 전체 Layer의 가중치를 업데이트 하는 방식
- 다변수들 중에서 하나의 변수가 전체 크기에 어느 정도 영향을 주는지를 알고자 한다.

< 여러 뉴런이 있는 신경망 >

영향을 주는 모든 경로를 고려해야 한다.



$$\frac{\partial J(\theta)}{\partial \theta_{11}^{(1)}} = \left(\frac{\partial J(\theta)}{\partial a_1^{(3)}}\right)\left(\frac{\partial a_1^{(3)}}{\partial z_1^{(3)}}\right)\left(\frac{\partial z_1^{(3)}}{\partial a_1^{(2)}}\right)\left(\frac{\partial a_1^{(2)}}{\partial z_1^{(2)}}\right)\left(\frac{\partial z_1^{(2)}}{\partial \theta_{11}^{(1)}}\right) + \left(\frac{\partial J(\theta)}{\partial a_2^{(3)}}\right)\left(\frac{\partial a_2^{(3)}}{\partial z_2^{(3)}}\right)\left(\frac{\partial z_2^{(3)}}{\partial a_1^{(2)}}\right)\left(\frac{\partial a_1^{(2)}}{\partial z_1^{(2)}}\right)\left(\frac{\partial z_1^{(2)}}{\partial \theta_{11}^{(1)}}\right)$$

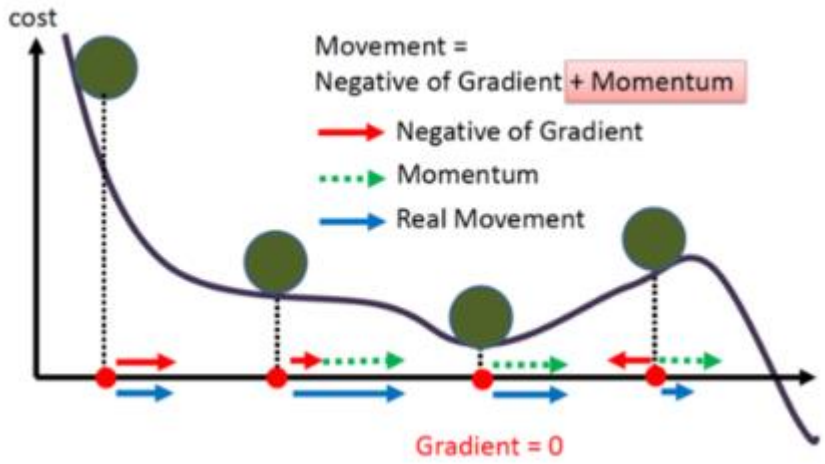
2. 딥러닝 학습

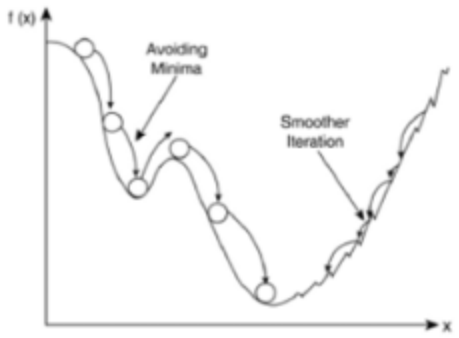
[최적화]

- 옵티마이저 (Optimizer) : 더 빠르고 안정적으로 최적화해주는 것
- 종류 : SGD with batch, Momentum, AdaGrad, RMSProp, Adam

< Momentum >

- 과거의 가중치를 반영하여 새로운 가중치를 계산한다.
- 이전에 이동했던 방향을 기억하면서 이전 기울기의 크기를 고려하여 어느 정도 추가로 이동시킨다.
- SGD보다 더 빠른 속도로 가중치를 업데이트한다.



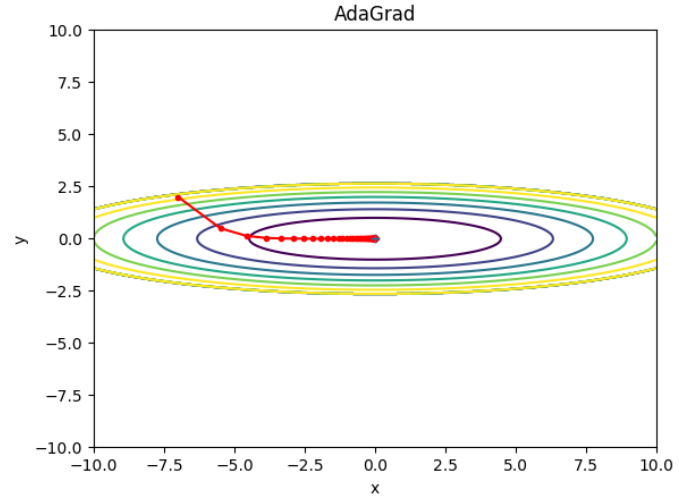


2. 딥러닝 학습

[최적화]

< AdaGrad >

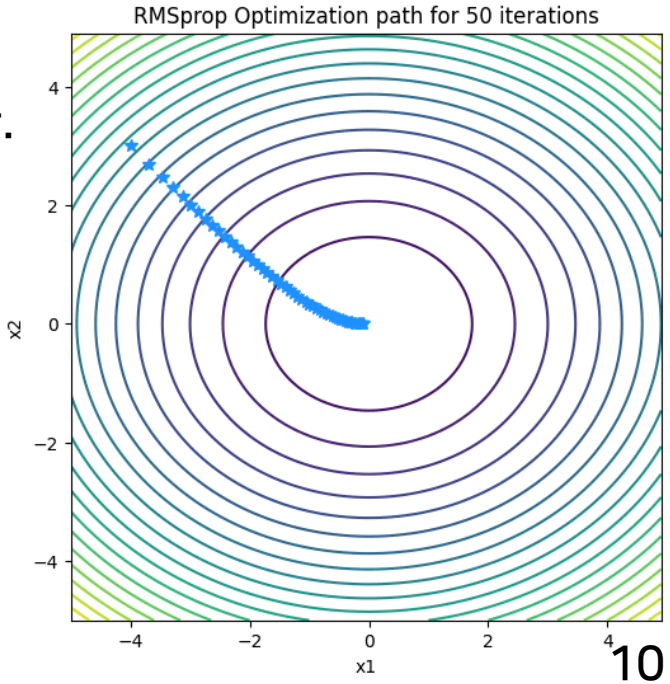
- 가중치 별로 다른 learning rate를 동적으로 사용하는 방식
- 처음에는 큰 learning rate가, 최저점에 가까울 수록 learning rate가 작아진다.
- 결론적으로 너무 작은 learning rate가 된다.



< RMSProp >

- AdaGrad처럼 모든 gradient의 합을 반영하는 것이 아니라, 최근 것에 가중치를 부여한다.
- 지수 가중 평균법 (EWA, Exponentially Weighted Average)

  : 시간의 흐름에 따라 오래된 데이터에 대한 영향력이 지수적으로 감쇠하도록 설계한 방식



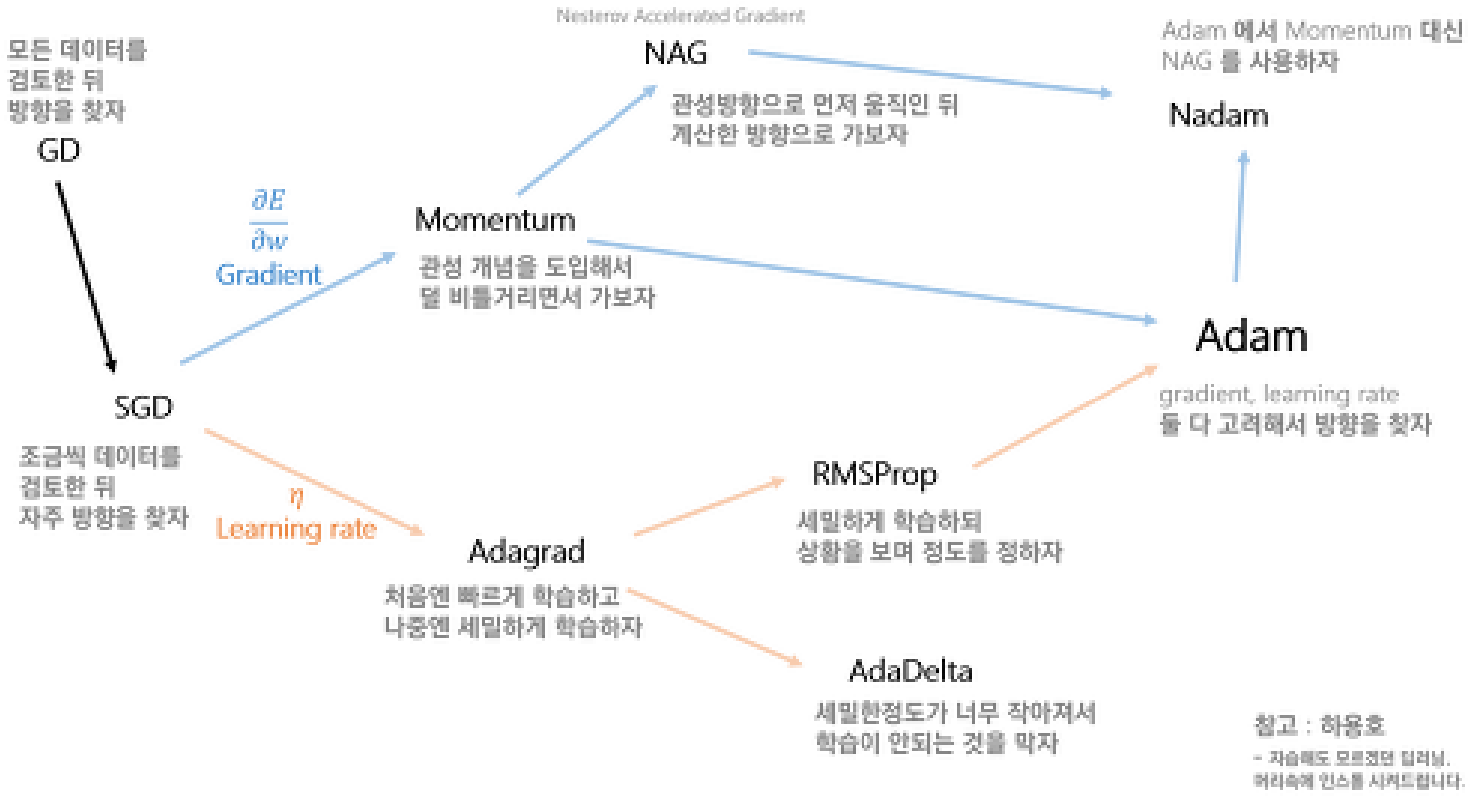
2. 딥러닝 학습

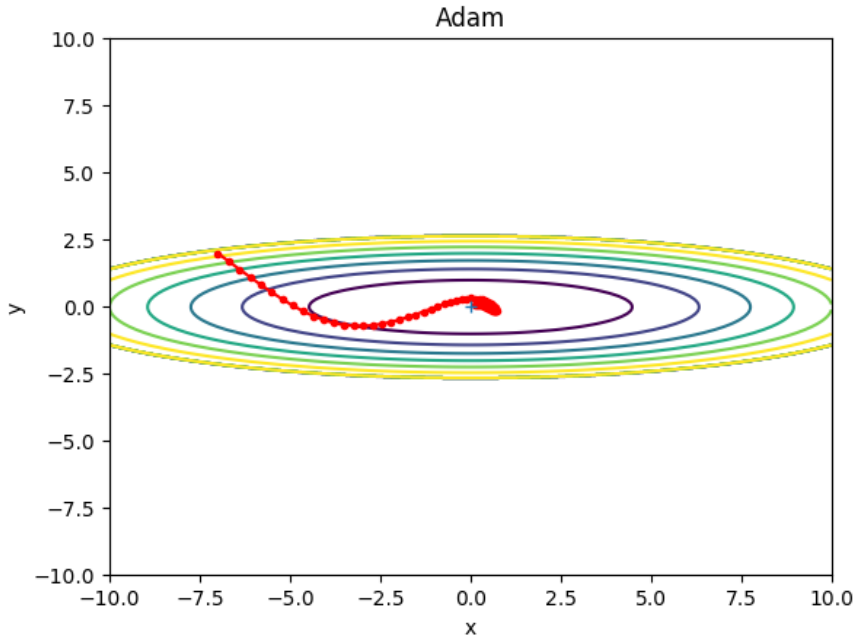
[최적화]

< Adam >

- Gradient에 Momentum 적용 + Learning Rate에 RMSProp 적용

  (단, Gradient에 Momentum 적용하는 부분에서 지수 가중 평균 적용)





2. 딥러닝 학습

[활성화 함수]

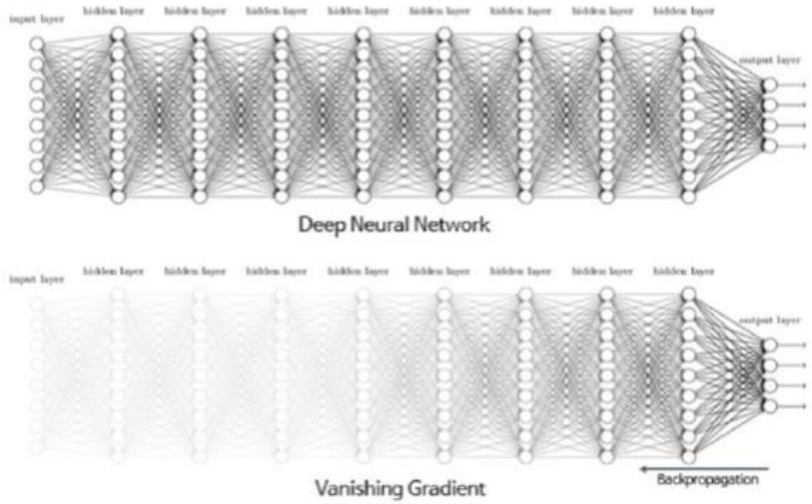
< 활성화 함수의 주요 목적 >

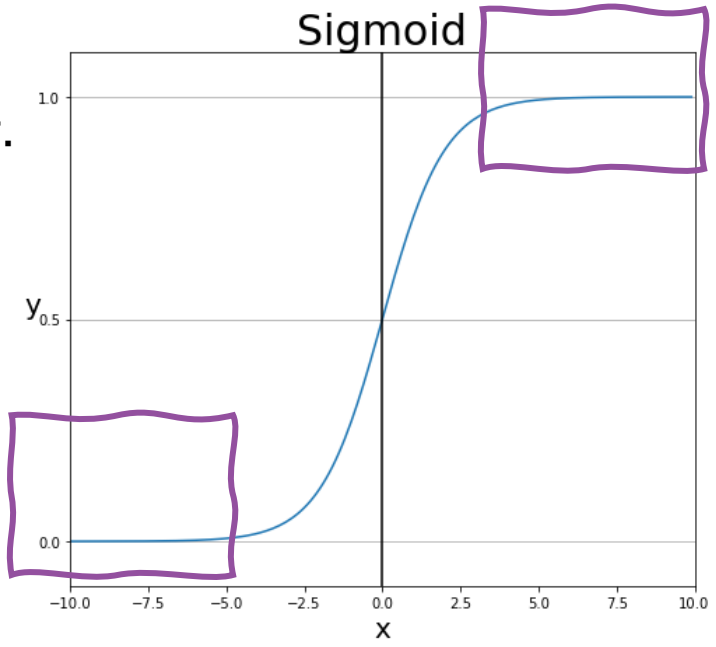
1. 딥러닝 네트워크에 비선형성을 적용한다.
2. 복잡한 함수를 근사할 수 있도록 만들어준다.

   (단, 비선형성을 가진다고 무조건 좋은 것은 아니다. 필연적으로 오버피팅 수반!)

< Sigmoid >

- 마법의 확률 곡선이라고 불리는 오래된 활성화 함수
- 은닉층에 더 이상 활성화 함수로 사용되지 않는데, 그 이유로는 기울기 소실 문제가 있다.



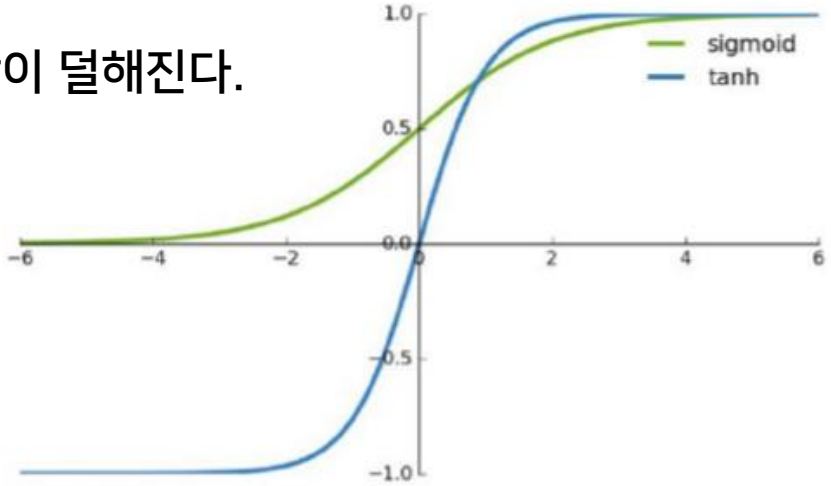


2. 딥러닝 학습

[활성화 함수]

< Tanh >

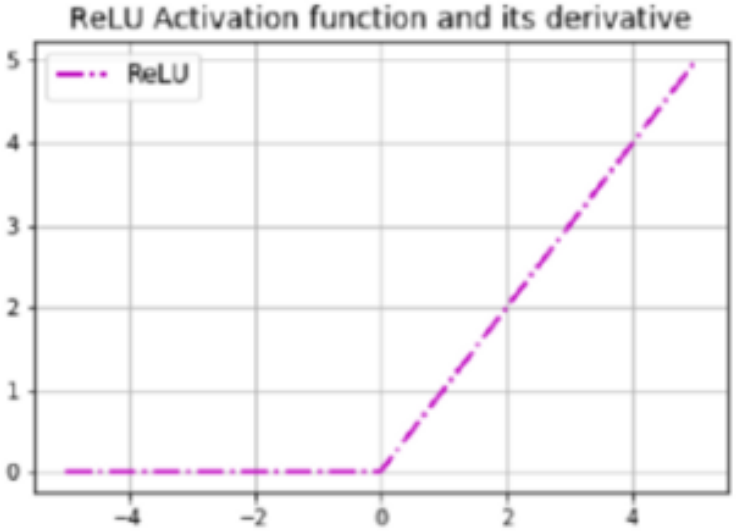
- 출력값의 평균이 0이 되면서 minimum이 지그재그로 수렴하는 현상이 덜해진다.
- 여전히 기울기 소실 문제는 남아있다.



$$\phi(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

< ReLU >

- 대표적인 은닉층의 활성함수
- 입력값이 0보다 작을 때 출력이 0, 0보다 클 때 입력값을 출력



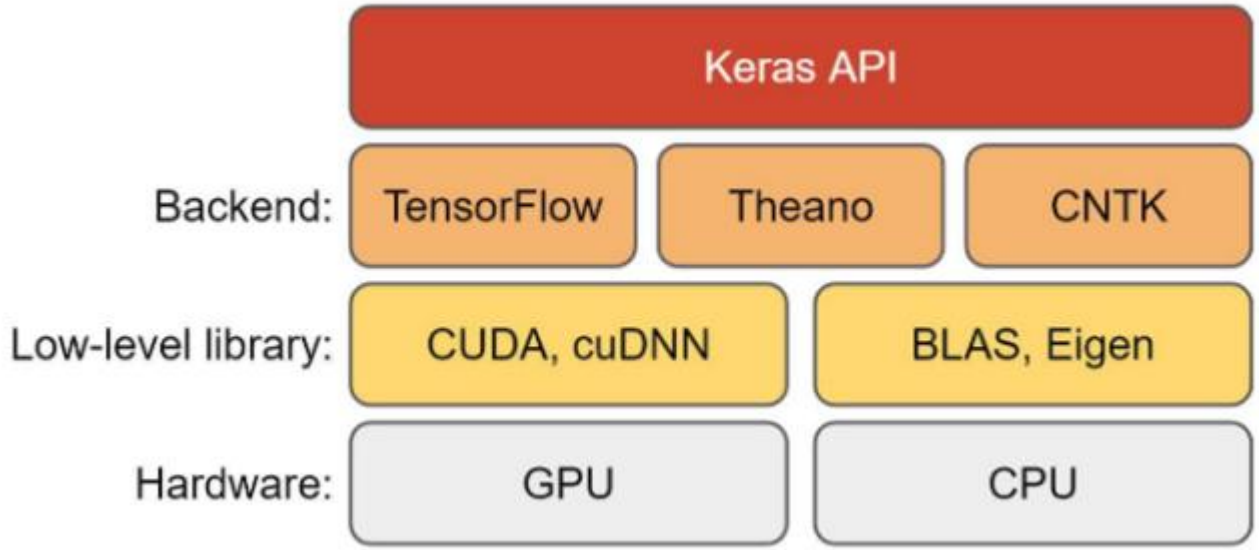
3. Keras Framework

< Keras와 멀티 백엔드 >

- Keras는 기본적으로 high level API이다.

  딥러닝의 아주 기본적인 연산 또는 CPU, GPU, TPU와 같은 HW를 동작시키는 작업은 Keras에서 구현하지 않음.
- Backend Engine에서 제공하는 Tensor Library가 연산을 맡아서 한다.

  예) Tensorflow, Theano, CNTK
- “Keras는 Multi-Backend를 지원한다.”



3. Keras Framework

< Tensorflow 2.x >

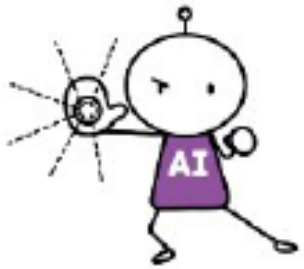
- Tensorflow 2.0 내부에 Keras가 패키지로 들어갔다.
- 과거 Keras에서는 백엔드 지원을 멀티 백엔드로 지원했다면, Tensorflow 2.0에서는 Keras 단독 백엔드를 지원한다.

```
from keras.layers import Dense
```

```
from tensorflow.keras.layers import Dense
```

Undergraduate Research Internship in Affective AI LAB.

감사합니다 :>



컴퓨터공학과 12201682 공지윤