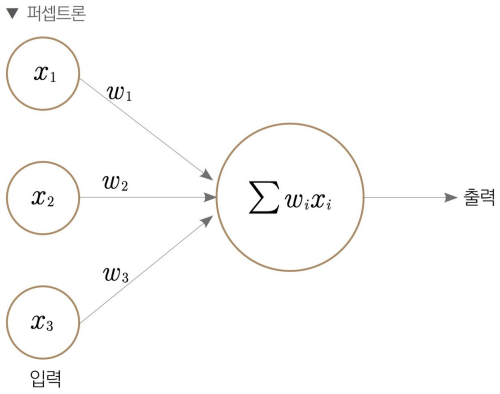
딥러닝 주요 개념

송태영

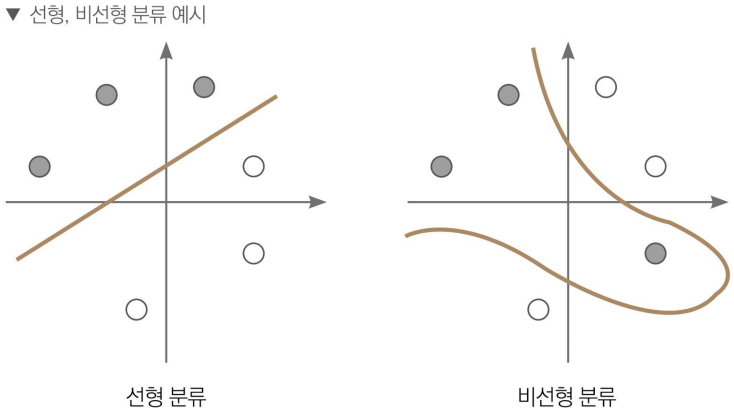
인공 신경망

퍼셉트론

- 퍼셉트론(perceptron)은 뉴런의 원리를 본떠 만든 인공 구조.
  - 입력값과 가중치를 곱한다
  - 곱한 값들의 총합을 구한다
  - 총합이 0을 넘으면 1, 아니면 0 출력



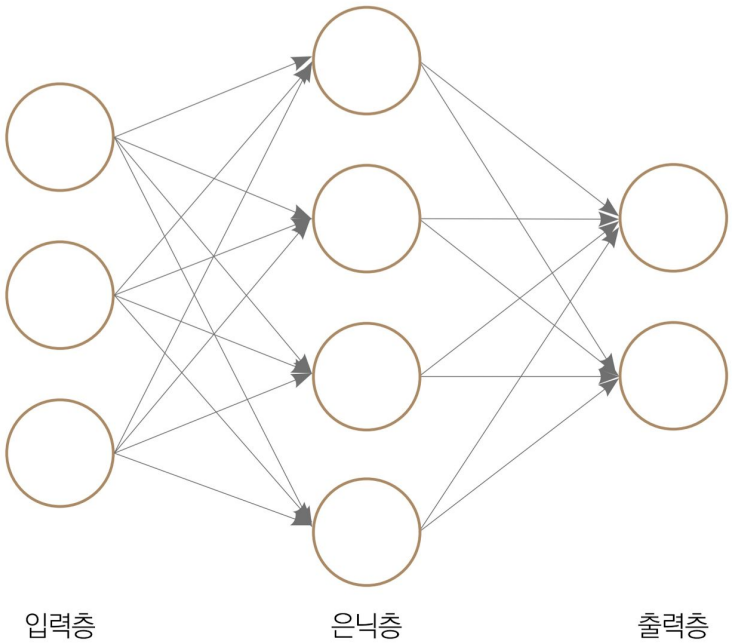
- 선형 분류 밖에 풀지 못한다는 한계
- 여러층을 쌓아 다층 퍼셉트론을 만들어야 비선형 분류 가능



신경망

- 신경망은, 입력층, 은닉층(중간층), 출력층으로 구성
- 신호는 입력층 → 은닉층 → 출력층으로 흐른다.
- 은닉층은 아예 존재하지 않거나 1개층 이상 존재

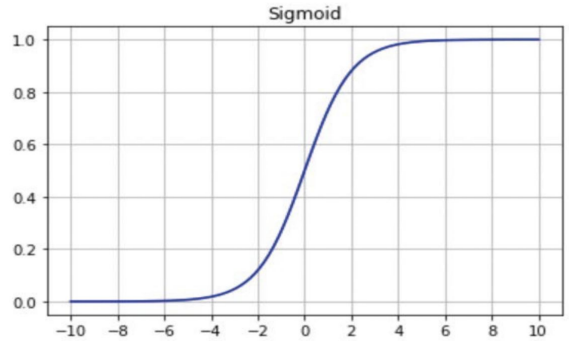
▼ 신경망 구조



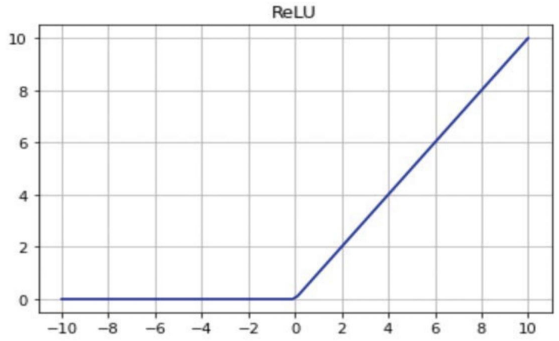
활성화 함수

- 입력값을 어떤 값으로 변환해 출력할지 결정, 입력값과 가중치를 곱한 값들은 활성 함수를 거쳐 출력
- 시그모이드 : S자를 그리는 함수
- ReLU(rectified linear unit) 함수 : 0 보다 크면 그대로 출력, 작으면 0 출력
- Leaky ReLU 함수 : 입력이 0이하일 때 약간의 음수값 출력

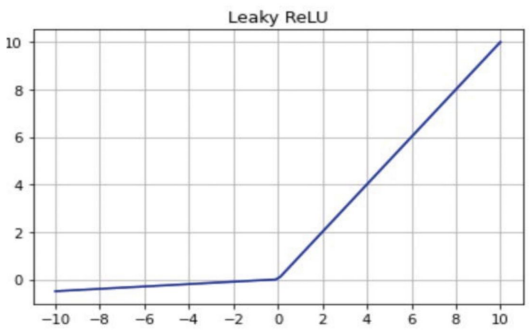
▼ 시그모이드 함수 그래프



▼ ReLU 함수 그래프

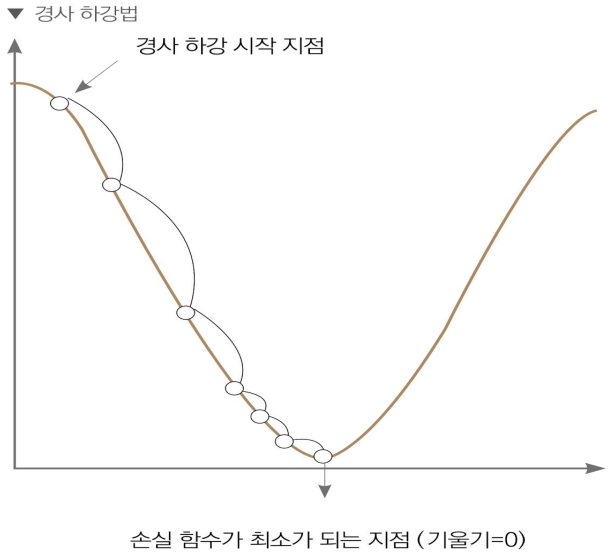


▼ Leaky ReLU 함수 그래프



경사 하강법

- 신경망 훈련의 목표는 최적의 파라미터 찾기
- 최적의 파라미터는 손실 함수가 최솟값일 때의 파라미터
- 손실 함수 : 모델의 성능이 얼마나 나쁜지 측정하는 함수. 모델의 예측값과 실젯값 사이의 차이
- 경사 하강법(gradient descent) : 현 위치의 경사(기울기)를 구해 기울기와 반대 방향으로 이동한다. 손실 함수가 최소가 될 때 까지
- 기울기 반대 방향으로 얼마나 이동할 지를 정하는 것을 학습률 이라고 한다.
- 즉, '학습률과 기울기를 곱한 값'을 뺀 값이 다음 가중치 임.
- 확률적 경사 하강법, 배치 경사 하강법, 미니 배치 경사 하강법 등



손실 함수 미분값(기울기)

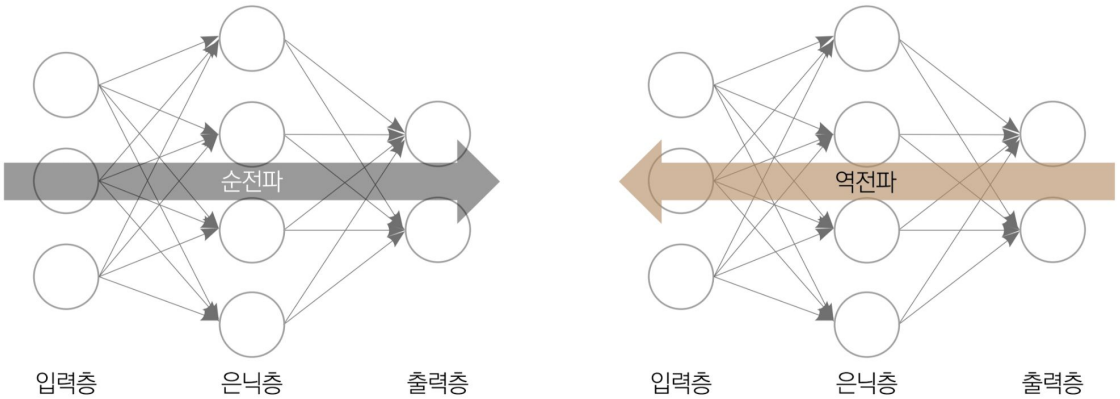
$$W = W - \eta \frac{\partial L}{\partial W}$$

갱신할 가중치 학습률

순전파와 역전파

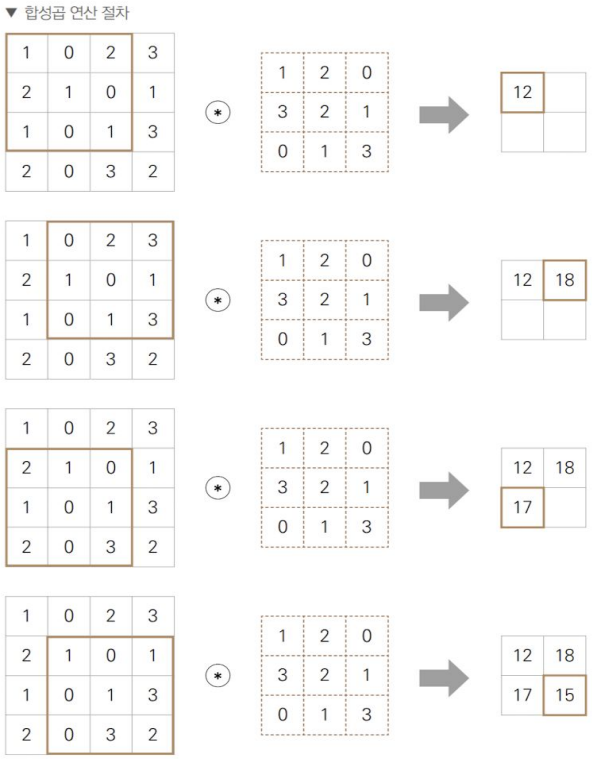
- 순전파 (forward propagation) : 신경망에서 입력값이 입력층과 은닉층을 거쳐 출력층에 도달하기 까지의 계산과정, 타깃 예측값과 실제 타깃의 차이(손실) 계산
- 역전파 (back propagation) : 순전파의 반대 개념, 손실을 입력층 방향으로 보내면서

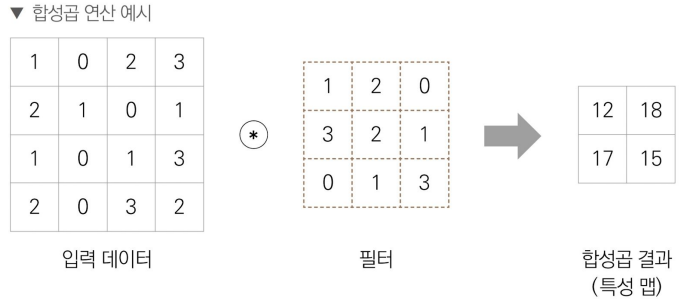
▼ 순전파와 역전파

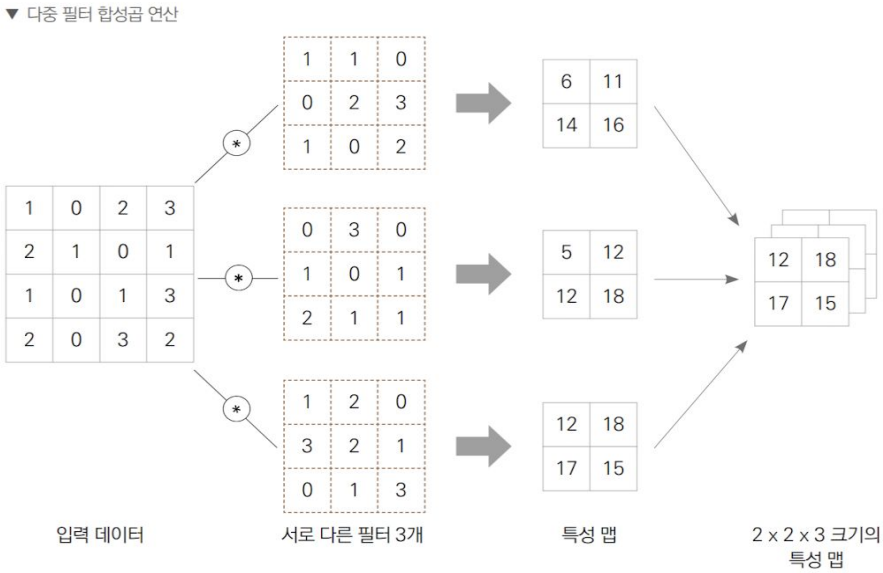


합성곱 신경망(CNN)

합성곱 계층


▼ 합성곱 연산 절차
1 0 2 3
2 1 0 1
1 0 1 3
2 0 3 2
1 2 0
3 2 1
0 1 3
12
18
17
15

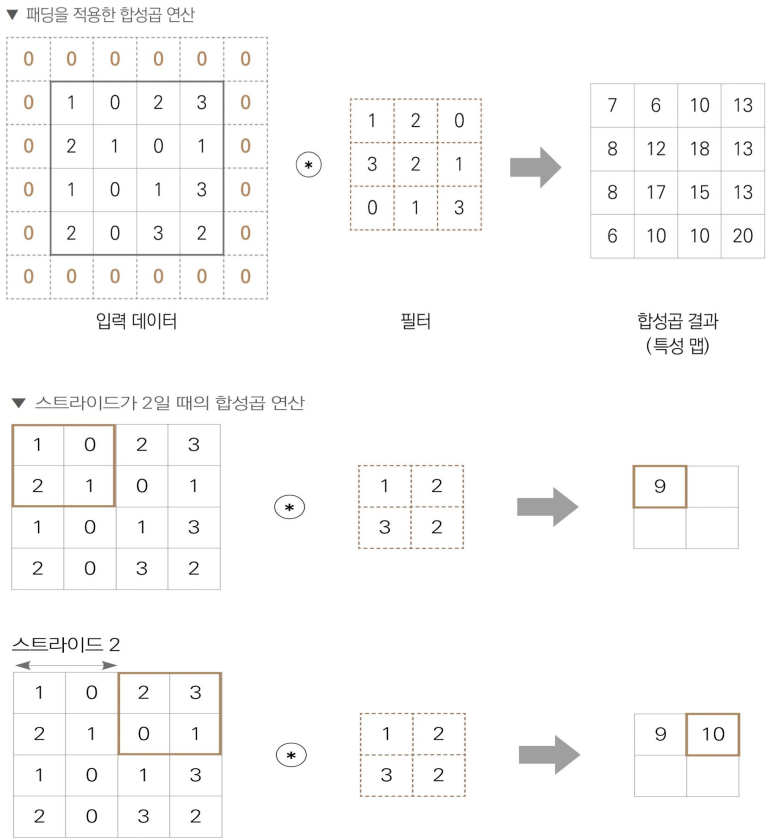

▼ 합성곱 연산 예시
1 0 2 3
2 1 0 1
1 0 1 3
2 0 3 2
1 2 0
3 2 1
0 1 3
12 18
17 15
입력 데이터
필터
합성곱 결과
(특성 맵)


▼ 다중 필터 합성곱 연산
1 0 2 3
2 1 0 1
1 0 1 3
2 0 3 2
1 1 0
0 2 3
1 0 2
0 3 0
1 0 1
2 1 1
1 2 0
3 2 1
0 1 3
6 11
14 16
5 12
12 18
12 18
17 15
12 18
17 15
입력 데이터
서로 다른 필터 3개
특성 맵
2 x 2 x 3 크기의
특성 맵

패딩과 스트라이드

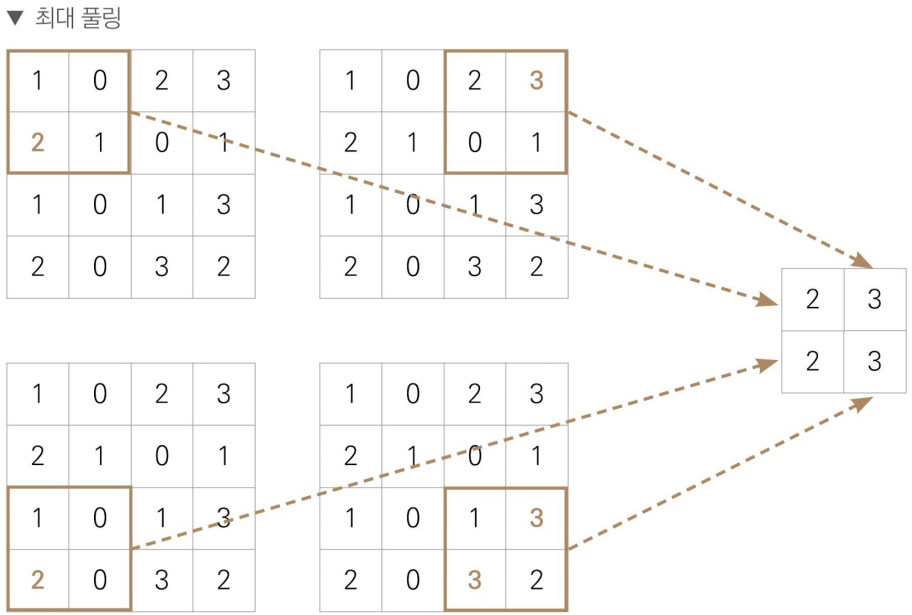
- 패팅 (padding) : 합성곱을 하게 되면 피쳐맵의 크기가 입력보다 작아짐, 크기를 유지 하기 위해, 입력 데이터 주변에 특정 값으로 채우는 것
- 필터를 입력값과 합성곱을 할 때 몇 칸 씩 이동하는 가
- 입력 데이터의 크기를 Nin, 필터 크기를 K, 패딩 크기를 P, 스터라이드 크기를 S, 출력 크기를 Nout 이 경우

$$N_{out} = \left\lfloor \frac{N_{in} + 2P - K}{S} \right\rfloor + 1$$



풀링

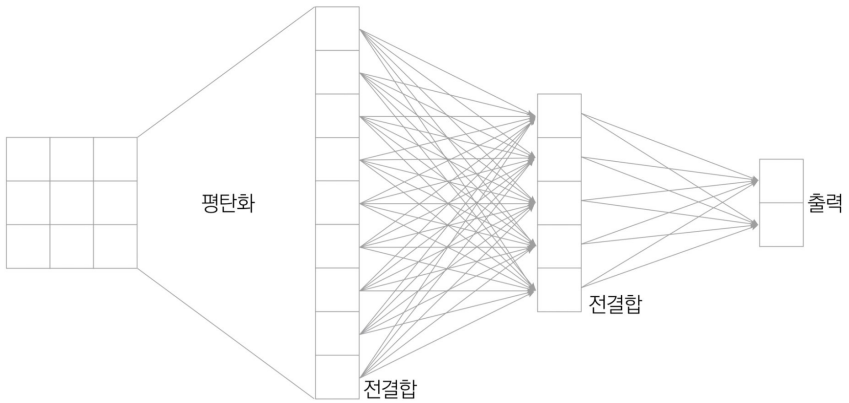
- 풀링(Pooling) : 특성 맵의 크기를 줄여 이미지의 요약 정보를 추출 하는 기능
- 풀링은 가중치(필터)가 필요 없다
- 특성 맵 크기가 줄면 연산 속도가 빨라진다
- Max pooling은, 풀링 영역의 가장 큰 값을 취한다.
- Average pooling은 평균을 구한다.
- 일반적으로, max pooling을 많이 쓴다.



전결합

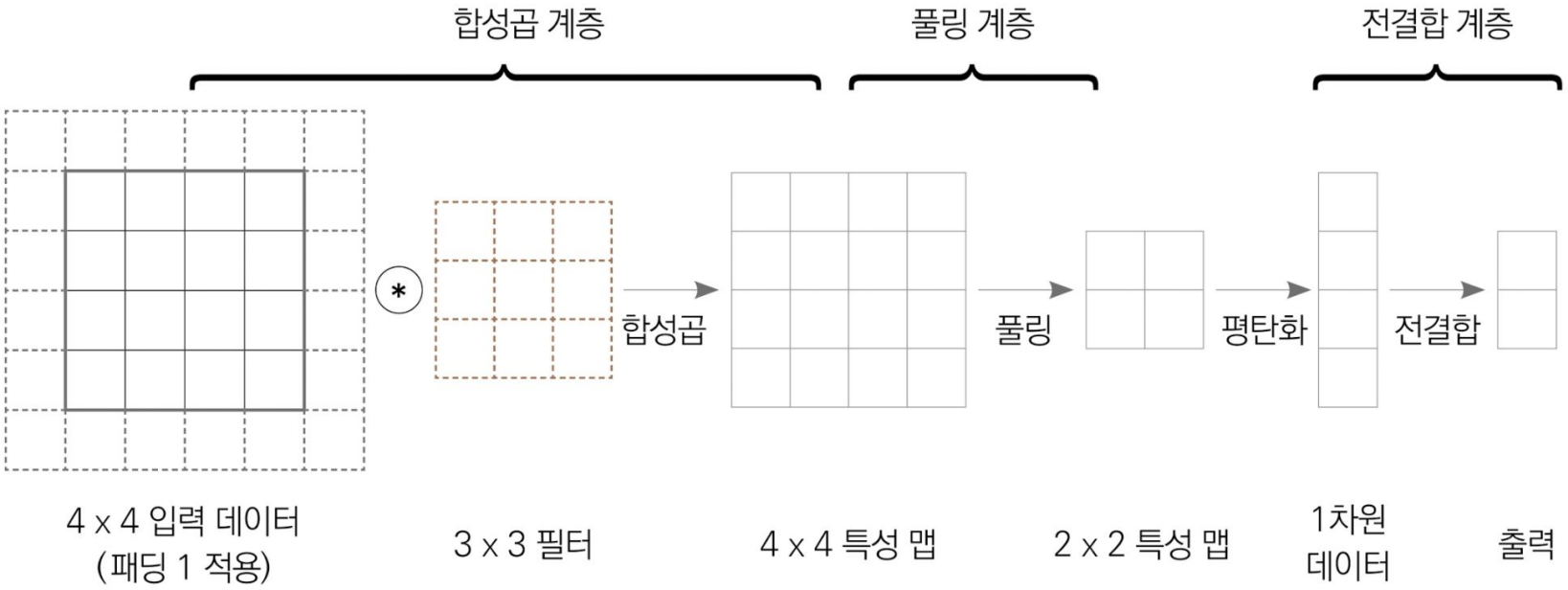
- 이전 계층의 모든 노드 각각이 다음 계층의 노드 전부와 연결된 결합을 전결합(fully connection)이라고 한다.
- 전결합으로 구성된 계측을 전결합 계층(fully-connected layer) 또는 밀집 계층(dense layer)라고 한다.
- CNN에서는 보통 맨 마지막 부분에 구현 됨. 단 평탄화 (2차원의 데이터를 1차원으로 바꾸는 것) 필요

▼ 전결합



전체 구조

▼ 합성곱 신경망 전체 구조

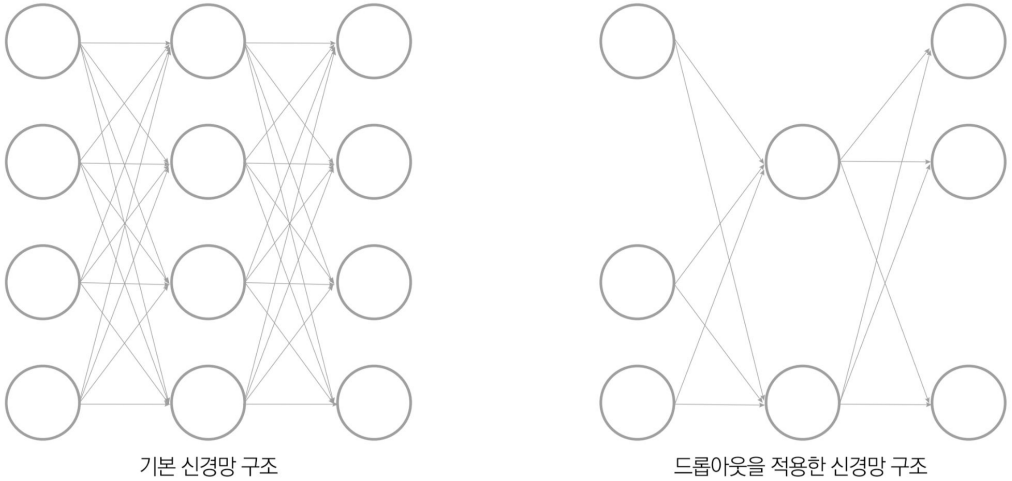


성능 향상을 위한 딥러닝 알고리즘

드롭아웃

- 과대적합을 막기 위해 신경망 훈련 과정에서 무작위로 일부 뉴런을 제외
- 매 이터레이션 마다 랜덤으로 드롭할 노드 결정 (앙상블과 유사)

▼ 일반적인 신경망 구조와 드롭아웃을 적용한 신경망 구조 비교



기본 신경망 구조

드롭아웃을 적용한 신경망 구조

배치 정규화

- 과대적합 방지 및 훈련 속도 향상을 위해 사용
- 내부 공변량 변화(신경망 계층 마다 입력 데이터의 분포가 다른 현상)를 해결
- 신경망 계층마다 입력 데이터 분포가 다르면 훈련 속도가 느려짐
- 배치 : 미니 배치 의미,
- 수행 단계
  - 1. 입력 데이터 미니배치를 평균이 0, 분산이 1이 되게 정규화
  - 2. 정규화한 데이터의 스케일을 조정 및 이동

- 정규화 : 미니 배치의 평균이 0 분산이 1

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

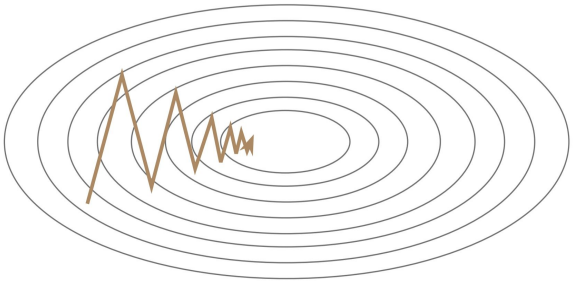
- 정규화를 하면 대부분 값이 0 근처로 몰리는 데, 이 런 값을 활성 함수에 입력하면 선형성에 빠지게 된다. (시그모이드 함수는 0 근처가 선형임)
- 활성 함수는 비선형 적이여야 성능이 좋으므로, 스케일 조정 및 이동을 해준다.

$$z_i = \gamma \hat{x}_i + \beta$$

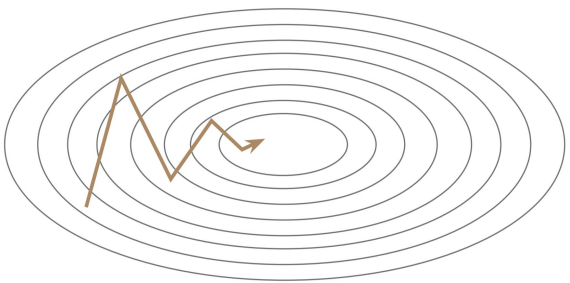
옵티아미저

- 신경망의 최적 가중치를 찾아주는 알고리즘을 옵티마이저(optimizer)라고 한다.
- 모멘텀 : SGD 옵티마이저는 최적의 경로를 찾아가는 경로가 비효율적, 모멘텀(momentum)은 SGD에 물리학의 관성 개념을 추가. 진행 하던 방향을 기억하여 일정 비율로 현 단계에 반영, 지그재그
- Adagrad : 학습률이 너무 크고 작으면 최적해를 찾는데 오래 걸림, 최적 파라미터에 도달 할 수록 학습률을 낮추도록 한 옵티마이저. 적응적 학습률(adaptive learning rate)
- RMSProp : Adagrad와 달리, 최근 기울기에 가중치를 더 줘서 학습률을 조절
- Adam : 모멘텀과 RMSProp을 합친것,

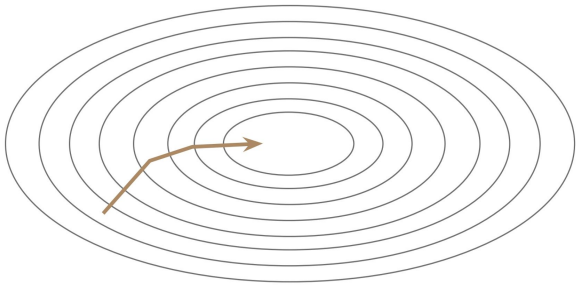
▼ SGD 옵티마이저로 최적 파라미터를 찾는 경로



▼ 모멘텀 옵티마이저로 최적 파라미터를 찾는 경로

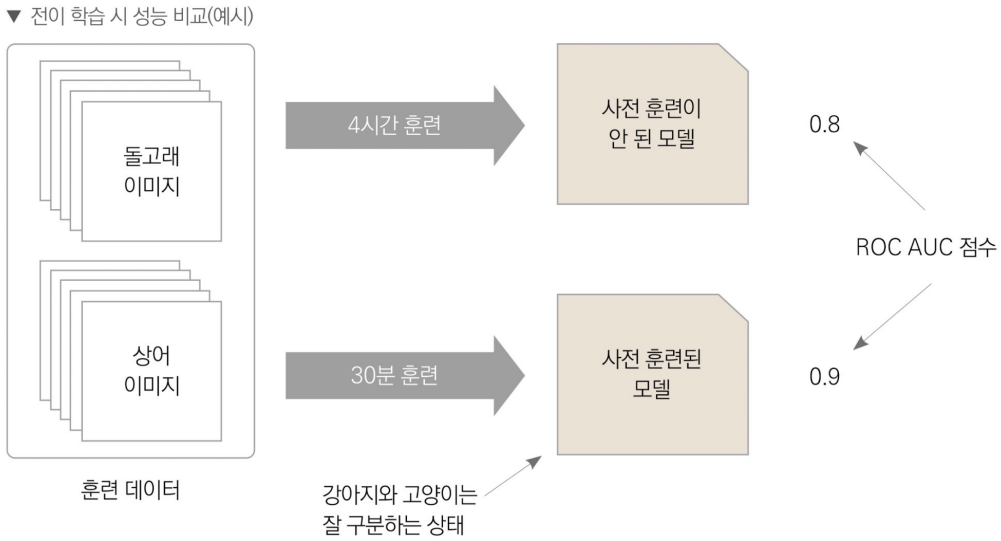


▼ Adagrad 옵티마이저로 최적 파라미터를 찾는 경로



전이 학습

- 한 영역의 사전 훈련 모델을 약간의 추가 학습을 더해 다른 영역에 활용
- 양질의 학습용 데이터를 확보하기 어려울 때 사용



전이 학습의 종류

- 신경망 전체 중 어느 정도를 학습 할지에 따라 나뉜다.
- 일부 계층만 (보통 출력층에 가까운 레이어들) 학습 시켜도 유용한 이유는, 신경망의 각 계측이 고유한 역할을 담당하고 있기 때문
- 계층을 올라 갈 수록 추상적인 특징을 가지고 있다.
- 따라서, 출력에 가까운 계층만 가지고 있는 데이터로 학습 시켜, 기존의 모델을 활용 할 수 있다.