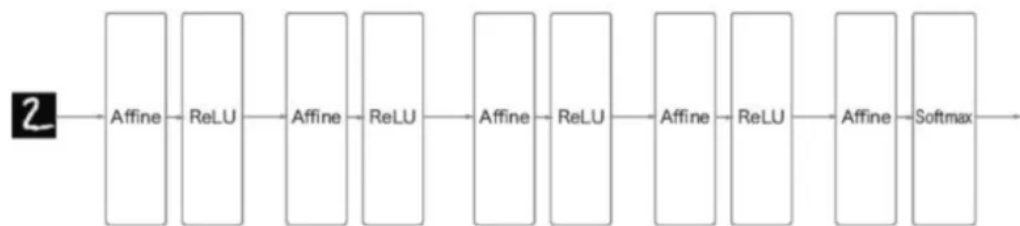


250711

1주차-2

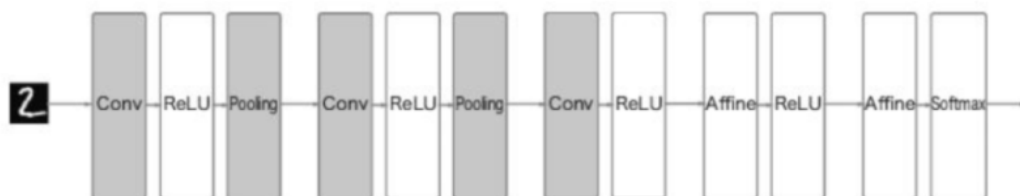
metrics = {'accuracy'}

- CNN 이론
 - 완전 연결 계층
 - 완전 연결 계층이란?
 - 신경망은 인접하는 계층의 모든 뉴런과 결합되어 있으며, 완전 연결 (fully-connected, 전 결합)이라고 하며, 완전히 연결된 계층을 **Affine 계층**이라는 이름으로 구현



완전연결 계층(Affine 계층)으로 이뤄진 네트워크의 예

- 완전 연결 신경망은 Affine 계층 뒤에 활성화 함수를 갖는 ReLU 계층(혹은 Sigmoid 계층)이 이어짐
- 이 그림에서는 Affine-ReLU 조합이 4개가 쌓였고, 마지막 5번째 층은 Affine 계층에 이어 소프트 맥스 계층에서 최종 결과 (확률)을 출력



CNN으로 이뤄진 네트워크의 예 : 합성곱 계층과 풀링 계층이 새로 추가(회색)

- 완전연결 계층에서는 인접하는 계층의 뉴런이 모두 연결되고 출력의 수는 임의로 정할 수 있음

- 문제점

- 데이터의 형상이 무시

- 입력 데이터가 이미지인 경우를 예로 들면, 이미지는 통상 세로, 가로, 채널(색상)로 구성된 3차원 데이터이고 형상에는 소중한 공간적 정보가 있음
→ 그러나 완전 연결 계층에 입력할 때는 3차원 데이터를 평평한 1차원 데이터로 평탄화해줘야함
 - 예를 들어 공간적으로 가까운 픽셀은 값이 비슷하거나, RGB의 각 채널은 서로 밀접하게 관련되어 있거나, 거리가 먼 픽셀끼리는 별 연관이 없는 등, 3차원 속에서 의미를 갖는 본질적인 패턴이 숨어 있음 → 그러나 완전 연결 계층은 형상을 무시하고 모든 입력 데이터를 동등한 뉴런(같은 차원의 뉴런)으로 취급하여 형상에 담긴 정보를 살릴 수 없음

- 합성곱

- 합성곱 (Convolution)

- 소리 신호 필터링, 영상 처리 등과 같이 입력과 이에 대한 출력이 존재하는 경우, 입력을 목적에 따라 가공해서 원하는 출력을 얻기 위해서 사용하는 연산
 - 입력 데이터에 필터를 일정한 간격으로 이동하며 단일 곱셈-누산

- 합성곱 연산

- **CNN 신경망**의 핵심으로 **이미지 처리**에 사용되는 연산
 - **더하기(합성)**와 **곱**만을 사용
 - Conv-ReLU 중 Conv계층에서 일어나는 연산으로 이미지 처리에서 말하는 필터 연산에 해당
 - 커널(kernel) 또는 필터(filter) 라는 $n \times m$ 크기의 행렬로 높이(height) \times 너비(width) 크기의 이미지를 처음부터 끝까지 겹치며 훑으면서, $n \times m$ 크기의 겹쳐지는 부분의 각 이미지와 커널의 원소의 값을 곱해서 모두 더한 값을 출력으로 하는 것

- padding

- padding

- 합성곱 연산 전에 입력 데이터 주변에 특정 값을 채우는 것

- 합성곱 연산 전에 입력 데이터 주변
- 주로 출력 데이터의 크기를 조정할 목적으로 사용
- 1픽셀리 패딩이란 데이터 사방에 1픽셀을 특정 값으로 채운다는 의미

■ padding 종류

- Valid padding (유효 패딩)
필터를 통과시키면 **항상 입력 사이즈보다 작게 됨**
합성곱 연산시 입력데이터의 모든 원소가 같은 비율로 사용되지 않는 문제
- Full padding (전체 패딩)
모든 원소가 합성곱 연산에 같은 비율로 참여하도록 하는 패딩 방식
- Same padding (동일 패딩)
출력 크기를 **입력 크기와 동일하게 유지**
풀패딩의 절반 개념이므로 **절반 패딩(half padding)**
- Mirror Padding (미러 패딩)
경계 밖 값들은 **경계에 반사된 값**을 가짐
경계 픽셀들이 **디테일이 있는 이미지**일 경우 적용가능 하지만 **계산 비용이 증가함**
- Replicate Padding (복제 패딩)
주변 픽셀 값을 복제하여 패딩하는 방법
입력 배열의 경계 픽셀 값을 패딩 영역에 복사하여 사용
주변 픽셀 값이 상수이거나 **픽셀 값을 보존**하는 데 유용
- Constant Padding (상수 패딩)
주변에 상수 값을 사용하여 패딩하는 방법
주어진 입력 배열의 주변에 사용자가 지정한 상수 값을 추가 (일반적으로
는 이미지 경계를 처리하는 데 사용)
- Zero padding (제로 패딩)
주변에 0으로 채워지는 패딩 방식
간단하고 **계산 비용**이 낮아 많이 사용되는 패딩 방법 중 하나
- Circular Padding (원형 패딩)
입력 배열을 원형으로 감싸는 방식으로 패딩

배열의 마지막 픽셀 다음에 첫 번째 픽셀이 오도록 패딩

주파수 도메인에서의 연산이나 순환적인 패턴 처리에 유용한 패딩 방법 그러나 정보 손실과 계산 비용 측면 에서도 고려

- pooling

- 사용 이유

- sub sampling은 해당하는 image data를 작은 size의 image로 줄이는 과정
 - 앞선 layer들(CONV layer, Activation, etc...)을 거치고 나서 나온 output feature map의 모든 data가 필요하지 않기 때문 → 다시말해, inference (추론, ex. 사과를 보고 사과라고 맞추는 것)를 하는데 있어 적당량의 data만 있어도 되기 때문

- 특징 및 장단점

- 특징

- training을 통해 train되어야 할 parameter가 없음
 - Pooling의 결과는 channel 수에는 영향이 없으므로 channel 수는 유지(independent)
 - input feature map에 변화(shift)가 있어도 pooling의 결과는 변화가 적음 (robustness)

- 장단점

- 장점 : 맵 크기를 줄임으로써 중요한 데이터만 추출하므로 노이즈를 제거하는 효과가 있다는 점
 - 단점 : 전체 정보를 반영하지 못하므로 추출되지 않은 정보가 중요한 정보인 경우 중요한 정보를 소실하여 성능을 저하시킬 수 있다는 점

- conv(n)d [n = 1, 2]

??

- 입력 데이터의 차원, kernel

-

- keras에서 MLP 구현하는 법 (이론)
 - 1. 텐서플로2 설치
 - 2. Sequential API로 이미지 분류기 구현하기
 - a. 데이터 다운로드
 - b. Sequential Model 구축
 - c. 모델 알아보기
 - d. 모델 컴파일하기
 - e. 모델 학습하기
 - f. 학습 시각화하기
 - g. 테스트 데이터로 평가하기
 - h. 예측하기