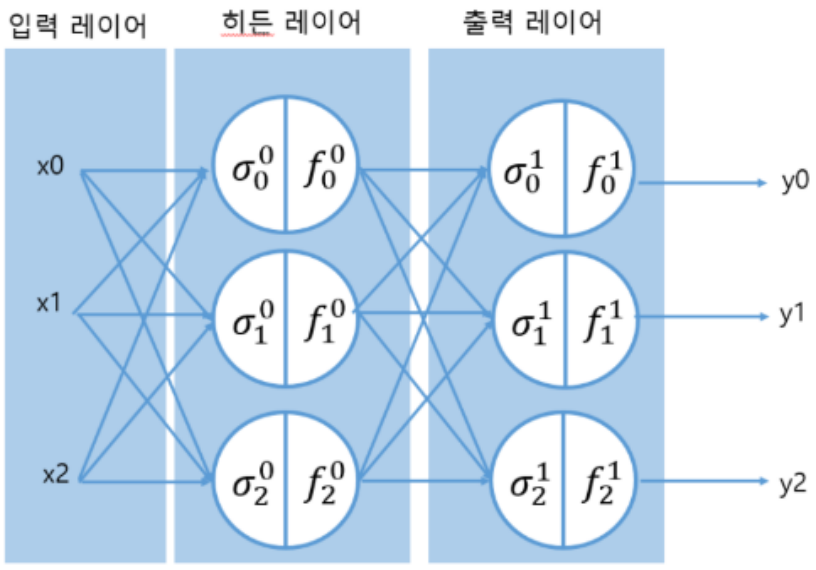
2주차 (1)

CNN 이론

완전 연결 계층

- 완전 연결 계층(Fully Connected Layer)이란?
  - 한층의 모든 뉴런이 다음 층에 전부 연결되어있다는 상태이다.
  - Dense Layer라고도 한다.
  - 입력이 1차원으로 변환된다.



  - 문제점
    - 2차원이나 3차원 이미지를 완전 연결 계층으로 네트워크를 구성하면 입력이 flatten 된다.
      - flatten: 다차원 배열을 1차원으로 만든다.
    - 공간적으로 가까운지/먼지와 연관이 있는 픽셀 정보들이 flatten 과정에서 데이터 형상이 무시된다.

합성곱

- 합성곱(Convolution)이란?
  - 합성곱은 함수 f, g에 대하여 하나를 반전시키고 전이시켜 곱한 것을 적분하는 계산이다.
- 합성곱 연산

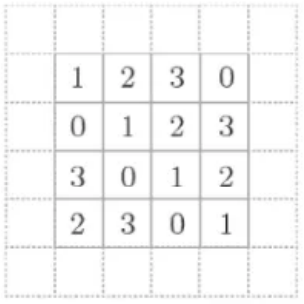
$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau$$

  t에 대해 한 함수 g를 반전과 전이한 식은 g(t-τ)이다.

- 완전연결계층의 문제점
  - CNN의 마지막 계층으로, 이전 계층들로부터 뽑은 정보를 flatten하므로 이미지가 크면 계산이 비효율적이고 feature를 잘 뽑아낼 수 없다.
  - ex) 3*3 Feature Map인 경우, flatten을 수행하면 9*1이 된다.

padding

- padding이란?
  - padding은 데이터 주위에 여백이나 값을 채워 출력의 크기를 늘리는 방식이다.
  - 패딩이 필요한 이유
    1. n * n 입력이 합성곱 계층을 지날때마다 작아져 1이 되버린다.
    2. 중요한 정보가 엣지쪽에 몰려있는 경우가 있다.



(4, 4)
입력 데이터(패딩 : 1)

- 패딩이 1인 경우 4*4에서 6*6로 output 크기가 2만큼 증가한다.
- 패딩을 적용시키고 3*3 필터로 연산을 한번 하면 입력의 크기가 된다.

- padding 종류
  - Zero padding : 패딩을 0으로 채운다.
  - Valid padding : 패딩을 붙이지 않는다.
  - Full padding : (필터크기-1) 만큼 패딩을 붙인다
  - Same padding : 출력크기=입력크기가 되도록 패딩을 붙인다.
    - (필터크기-1)/2 = 패딩 크기

pooling

- 사용 이유
  - Map의 크기를 줄이거나 특징을 추출하기 위함이다.

특징 및 장단점

- 특징
  - 학습할 parameter가 없다. (최댓값, 평균 계산만 수행)
  - Channel(3차원) 단위로 계산하여 Channel 수가 같다.
  - 입력의 크기에 영향을 덜 받는다.
- 장점
  - Channel 수가 변하지 않고 Map 크기만 줄일 수 있다.
  - Map크기를 줄여 자원을 절약한다.
  - Parameter 수를 줄여 오버피팅을 방지한다.
- 단점
  - 많은 정보를 활용하고 학습 속도를 높이는 알고리즘이 많아져 사용빈도가 줄었다.
  - 특징을 손상시킬 수 있다.
    - 최대 풀링의 경우, 전체적인 입력 데이터 특징에 손실이 발생할 수 있다.
    - 평균 풀링의 경우, 디테일한 특징이 손실될 수 있다.

Conv(n)D [n = 1, 2]

- Convolution(합성곱) 연산을 n차원으로 진행한다.
- 차원 → filter의 방향성을 결정
- 인자로 filter, kernel_size, activation, input_data 주로 이용

입력데이터의 차원, kernel

- Conv1D
  - 입력 데이터를 받아 1차원으로 합성곱 연산 수행
  - 3차원 input 필요

    - Sequence모델, 자연어 처리에서 주로 활용
    - 단어에 대해 한다면 한 문장에 대해서 합성곱을 진행하므로 1D 사용
- Conv2D
    - 입력데이터를 받아 2차원으로 합성곱 연산 수행
    - 3차원 input 필요
    - 이미지에 대해서 한다면 이미지 1장의 가로, 세로 공간에 대해 합성곱을 진행하므로 2D 사용
- Kernel
    - CNN에서의 Hyperparameter(특히 weight의 역할)이다.
    - 커널(필터)는 입력 데이터의 특징을 잘 뽑아낸 map을 만들어내야한다.
    - 따라서 적절한 사이즈의 커널로 합성곱을 최적으로 반복 수행하는 것이 중요하다.

keras에서 MLP 구현하는법 (이론)

1. 모델을 .Sequential()로 생성
2. .add 함수로 입력층 추가 → 입력을 Flatten 함수 내부에 추가 - 입력이 Flatten되어 추가됨
3. .add 함수로 은닉층 추가

   → 이전 뉴런과 완전연결계층으로 구성하기 위해 .Dense함수로 출력 뉴런 수, 활성화 함수 지정
4. .add 함수로 출력층 추가

   → 은닉층과 동일, 활성화 함수는 이진분류, 다중클래스 분류같이 목적에 맞는 함수로 지정
5. .summary()로 분석하여 의도한대로 만들어진 모델인지 확인
6. 모델의 비용함수, 최적화 방법, 평가지표를 지정한다. (.compile())
7. .fit으로 학습시킨다.(훈련/출력/검증데이터, 학습횟수(eporch) 등 지정)
8. fit으로 얻은 history 객체로 그래프 출력(시각화)