

14. Activation function 함수를 사용하는 이유? Softmax, Sigmoid 함수의 차이는?

Activation function은 신경망에 비선형성을 부여하기 위해 사용한다.

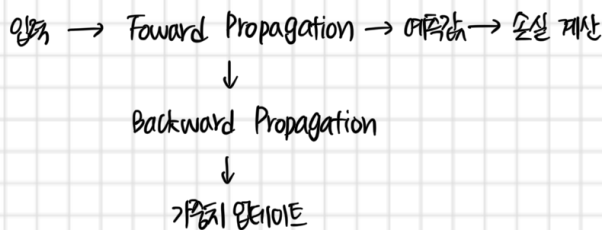
여러 층을 쌓아도 활성화 함수가 없다면 결국 하나의 선형모델과 같다 → 특별한 다른 학습 불가능

|       | Sigmoid 함수        | Softmax 함수            |
|-------|-------------------|-----------------------|
| 출력 범위 | 0 ~ 1             | 0 ~ 1 (각 출력값의 총합 = 1) |
| 사용 목적 | 0인 분류             | 다중 클래스 분류             |
| 출력 개수 | 1개                | 클래스 수 만큼              |
| 특징    | True/False 확률로 해석 | 여러 클래스 중 가장 높은 확률을 선택 |

15. Forward Propagation, Backward propagation이란?

Forward Propagation : 입력 데이터가 신경망의 각 층을 앞으로 통과하면서 예측값을 만들어내는 과정

Backward Propagation : 예측 오차를 바탕으로 각 층의 가중치에 대해 오차의 기울기를 계산하고, 이를 이용해 가중치를 업데이트하는 과정



16. 손실함수란 무엇인가? 가장 많이 사용하는 손실함수 4가지 종류는?

손실함수 : 모델의 예측값과 실제 정답 사이의 차이(오차)를 수치로 계산한 함수

↳ 이 값이 작을수록 모델이 더 정확하다는 뜻

이 값을 최소화하는 방향으로 모델을 학습한다

MSE (Mean Squared Error)

: 예측값과 실제 값의 차이를 제곱해서 평균 낸 값

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

MAE (Mean Absolute Error)

: 예측값과 실제 값의 차이의 절대값 평균

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

}

주로 레거리에 사용

Binary Cross-Entropy 0인 교차 엔트로피

: 두 클래스 (예: 고양이/개) 중 하나를 예측하는 0인 분류에 사용

$$\text{Binary Cross-Entropy} = -[y \log(\hat{y}) + (1-y) \log(1-\hat{y})]$$

확률 기반 예측에 적합, 확률이 정답에 가까울수록 loss가 낮음

Categorical Cross-Entropy 다중 클래스 교차 엔트로피

: 여러 클래스 중 하나를 고르는 다중 분류 문제에서 사용

정답 클래스의 확률만 영향을 준다.

$$CCE = - \sum_{i=1}^C y_i \log(\hat{y}_i)$$

17. 옵티마이저(optimizer)란 무엇일까? 옵티마이저와 손실 함수의 차이점은?

옵티마이저 : 모델의 가중치를 조정해서 손실 함수의 값을 최소화하도록 도와주는 알고리즘

신경망 학습 과정에서 손실함수를 가운다 가중치를 업데이트 한다. 경사하강법이나 그 변형된 방법들을 사용한다

손실 함수 : 모델의 예측값과 실제 값 사이의 차이를 수치로 표현하는 함수

손실함수는 밑바탕을 깔아주는 역할을 맡고 옵티마이저는 어떻게 그걸 줄일지 알려주는 도우미이다.

18. 정규화방법의 의미는? (학률적 정규화방법, 배치 정규화방법, 미니 배치 정규화방법)

정규화방법은 손실함수의 값을 최소화하기 위해 가중치를 변경하는 과정과 알고리즘이다.

현재 위치의 가중치를 구하며 가중치 업데이트 방향과 매개변수들을 업데이트하는 방법이다.

배치 정규화방법 : 전체 데이터를 사용해 가중치를 계산하고 안정적이지만 느림

학률적 정규화방법 : 샘플 하나씩 가중치를 계산하고 빠른지만 진동이 큼

미니 배치 정규화방법 : 소문 묶음 단위로 가중치를 계산하고 속도와 안정성이 균형 잡혀 있음

19. 교차 검증, k-fold 교차 검증의 의미와 차이

교차 검증을 통한 모델을 여러 폴드로 나누어 하나의 폴드가 검증 세트의 역할을 하고 나머지 폴드에서는 모델을 훈련한다.

k-fold 교차 검증을 전체 데이터를 k개의 폴드로 나눈 그중 하나를 검증용으로, 나머지를 학습용으로 사용해서 모델을 학습한다.

이 과정을 k번 반복해서, 각각 다른 검증용 데이터를 선택한다.

교차검증은 방법을 선택해야 하는 개념적인 프레임워크인 k-fold 교차 검증은 구현 가능한 구체적인 방식이다.

20. 하이퍼 파라미터 튜닝이란 무엇인가?

머신러닝이나 딥러닝 모델의 성능을 높이기 위해 하이퍼파라미터 값을 최적화 하는 과정이다.

21. CNN의 학습공의 역할은?

① 특징 추출 : 학습공은 입력 이미지의 국소적인 (소셜) 패턴을 탐지하고 추출한다.

작은 영역 (예:  $3 \times 3$ ,  $5 \times 5$  필터)을 스캔하여 이미지의 경계, 선, 모서리, 색상 변화 등을 감지한다.  
여러층의 학습공 연산을 거치면서 저수준에서 고수준까지 점진적으로 복잡한 특징을 학습한다.

② 파라미터 공유 : 학습공 연산에서는 같은 필터 (커널)를 전체 이미지에 적용한다.

→ 학습해야 할 파라미터 수 ↓, 학습 효율 ↑

③ 공간 불변성 : 입력이 이니치에 있을 같은 패턴을 탐지한다.

⇒ CNN에서 학습공은 국소적인 특징을 추출하고, 파라미터 수를 줄이며, 공간적으로 동일한 패턴을 감지하여 계층적으로 복잡한 특징을 학습하는 역할을 한다.

22. CNN에서 풀링층의 역할은?

① 특징의 공간 크기 축소 : 풀링층은 입력 특징 맵 (feature map)의 공간적 크기 (가로, 세로)를 줄인다.

⇒ 계산량 ↓, 모델의 메모리 사용 ↓

② 불변성 강화 : 풀링은 입력의 작은 위치 변화에도 특징을 유지하도록 도와준다.

최대 풀링 : 국소 영역의 최대값을 선택하여 가장 강한 특징을 강조  
최소 풀링 : 국소 영역의 평균값을 선택하여 정보를 부드럽게 요약

③ 과적합 방지 : 데이터의 다양성을 축소하고 중요한 정보만 남기고, 모델이 복잡해지는 것을 막는다.

④ 특징 요약 : 각 지역의 정보를 하나의 대표 값으로 요약하며, 다음 계층으로 중요한 정보만 전달한다.

⇒ CNN의 풀링층은 공간 크기를 줄여 계산량을 절약하고, 위치 변화에 강한 특징을 추출하며, 모델의 복잡도를 낮추는 역할을 한다.

## 23. CNN의 Dense Layer의 역할은?

① 분류 역할 : CNN의 앞부분은 주로 합성곱과 풀링층의 특징을 추출하고,

마지막에 위치한 Dense Layer는 이 추출된 특징을 기반으로 최종 출력을 예측한다.

② 전역 정보 처리 : 합성곱과 풀링은 3x3 (local) 영역만 처리하지만, Dense Layer는 모든 뉴런을 연결하여 이미지 전체의 정보를 활용한다.

③ 비선형 조합 : 입력된 특징을 여러 가중치와 활성화 함수를 통해 복잡하고 조합하여, 복잡한 패턴을 학습한다.

④ 저차원 공간으로 투영 : 추출된 특징을 저차원 공간으로 투영하여, 패턴 분리나 최종 분류를 수행한다.

⇒ CNN의 Dense Layer는 앞단계에서 추출된 특징을 기반으로 최종 분류 (혹은 레거 등)를 수행하며,

전역 정보를 처리하고, 복잡한 패턴을 학습하는 역할을 한다.

## 24. CNN의 Stride, Filter의 역할? 필터의 가중치는 어떻게 결정되는가?

- ① Stride의 역할
- 합성곱 연산 시 필터(커널)를 얼마나 이동시킬지를 나타내는 값이다.
  - 일반적으로 stride = 1이면 한 칸씩 이동, stride = 2이면 두 칸씩 이동한다.
  - stride를 크게 하면
    - 출력 feature map의 크기가 작아진다
    - 계산량과 메모리 사용량을 줄일 수 있다.
    - 정보 손실이 발생할 수 있으므로 주의해야 한다

### ② Filter의 역할

입력의 3D 영역 (x, y, h, w)에 대해 합성곱을 수행하는 연산 단위

여러개의 필터를 사용하여, 각각 다른 특징 (모서리, 색상, 텍스처 등)을 추출한다.

필터마다 다른 가중치를 가지고, 여러 종류의 특징을 감지할 수 있다.

### ③ 필터의 가중치는 어떻게 결정되는가?

CNN 학습의 핵심 : 필터의 가중치를 학습하는 것.

주어진 필터의 가중치는 무작위로 초기화된다.

학습 과정에서 역전파와 경사하강법을 통해 오류율 (loss)를 계산하고, 손실 함수의 기울기를 계산해

필터의 가중치를 점점 더 나은 방향으로 업데이트한다.

이 과정에서 필터는 단순한 랜덤 값에서 유용한 특징을 감지하도록 진화한다.

## 25. RNN을 사용하는 이유와 한계점?

### ① 사용 이유

- 시퀀스 데이터 (연속적인 데이터) 예) 문장, 음성, 시계열 등을 처리하기 위해 고안된 구조
- 과거 정보를 기억하고 현재 출력에 반영할 수 있으며, 시퀀스 간의 관계를 모델링할 수 있다.

### ② 한계점

- 장기 의존성 문제 : 입력 시퀀스가 길어지면, 초기 입력 정보가 점점 희미해져서 학습이 어려워진다.  
이를 Gradient Vanishing 또는 Exploding이라고 한다.
- 계산량이 많고 학습 속도가 느려 병렬 처리가 어렵다.

26. LSTM을 사용하는 이유와 한계점은?

① 사용 이유

- RNN의 장기 의존성 문제를 해결하기 위해 고안된 구조이다.
- 입력 게이트, 출력 게이트, 삭제 게이트를 이용하여 중요한 정보를 장기간 유지하고, 덜 중요한 정보는 잊도록 조절한다.

② 한계점

- RNN보다 구조가 복잡하고 학습 시간이 오래 걸린다.
- 많은 파라미터로 인해 메모리 사용량이 많고, 계산 비용이 크다.

27. GRU를 사용하는 이유와 차별성은?

① 사용 이유

- LSTM의 복잡한 구조를 단순화하면서도 비슷한 성능을 유지하기 위해 고안된 구조이다.
- 게이트를 줄여 연산량과 메모리 사용량을 감소시켰다.

② 차별성

- GRU는 3개의 게이트를 2개 (업데이트, 리셋)로 통합해 구조를 간소화했다.
- LSTM보다 학습 속도가 빠르고 메모리 효율이 좋다.
- 그러나 긴 시퀀스에서는 LSTM이 더 좋은 성능을 보이기도 한다.

33. Dense Layer란 무엇인가?

- 입력층 모든 노드가 출력층 모든 노드와 연결되며, 입력 벡터의 각 요소에 가중치를 곱한 결과를 더해 하나의 출력으로 변환하는 층
- 이미지를 다룰 때는 평탄화 후 Dense Layer를 적용하고 픽셀 간 관계가 깨져 이미지 인식 성능이 떨어질 수 있다.