

14. Activation function 함수를 사용하는 이유는? Softmax, Sigmoid 함수의 차이?

딥러닝에서 각 인공신경망에 누적된 계산을 하나의 결과값으로 계산하기 위해서 사용하는 함수이다.

- Sigmoid는 하나의 출력값을 0~1 사이의 실수 값으로 변환한다.
→ 이진 분류에 사용된다.
- Softmax는 여러 개의 출력값을 받아 각 값을 0~1 사이의 확률로 변환한다.
전체 출력의 합이 1이 되도록 정규화한다.
→ 다중 클래스 분류에 사용된다.

15. Forward Propagation - Backward Propagation이란?

Forward Propagation이란 가중치와 편향을 이용해 정방향으로 인공신경망 계산을 하여 결과 값에 대한 예측을 수행하는 딥러닝의 학습방법이다.

Back Propagation이란 예측값과 결과값을 비교하여 계산된 오차를 바탕으로 역방향으로 계산하여 feature에 대한 가중치를 수정하는 방법이다.

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

손실함수란 예측값과 실제 결과값 사이에 오차를 수치로 나타내는 함수이다.
오차 바탕으로 가중치를 수정하기 위한 경사하강법에서 필요한 함수이다.

① Mean Squared Error (MSE) - 회귀에 사용

실제 값과 예측 값의 제곱 오차 평균을 계산한다. 오차가 클수록 큰 벌점을 부과한다.

② Mean Absolute Error (MAE) - 회귀에 사용

절댓값 기준의 오차 평균을 계산한다. MSE보다 이상치에 덜 민감하다.

③ Binary Cross Entropy (BCE) - 이진분류에 사용

예측 확률이 정답에서 멀어질수록 손실이 커진다.

④ Categorical Cross Entropy (CCE) - 다중클래스 분류에 사용

정답 클래스에 해당하는 예측 확률이 높을수록 손실이 작아짐

17. 옵티마이저(optimizer)란 무엇일까? 옵티마이저와 손실함수의 차이점은?

손실함수는 모델이 얼마나 잘못 예측했는지를 수치화한다. 모델이 예측값을 생성하면 손실함수가 결과값과 비교하여 손실 값을 출력한다.

옵티마이저는 손실 함수의 값을 최소화하는 방향으로 가중치를 업데이트한다. 손실함수를 미분해서 기울기를 계산하고 이를 기반으로 가중치를 수정한다. 즉, 손실함수는 지금 예측이 얼마나 다른가를 알려주는 좌이고 옵티마이저는 그 손실을 줄이기 위해 실제로 업데이트를 수행하는 엔진이다.

21. CNN의 합성곱의 역할은?

합성곱은 입력이미지에서 특징(feature)을 추출하는 역할을 한다.

필터(커널)을 통해 이미지의 공간적 구조(예: 엣지, 모양, 질감등)를 감지하고, 이를 통해 중요한 정보를 보존하면서 입력의 차원을 줄인다.

18. 경사하강법 의미는? (확률적 경사하강법, ^{배치}배치 경사하강법, 미니 경사하강법)

경사하강법은 손실함수의 미분값이 나타내는 gradient를 계산해서 수정하며 손실함수의 최솟값을 찾아가는 계산법이다.

- 확률적 경사하강법은 데이터 한 개를 사용해 기울기를 계산하고 업데이트 한다. 속도가 빠르고 최솟값 탈출 가능성이 높다. 불안정적 수렴을 보인다.
- 배치 경사하강법은 전체 데이터셋을 사용해 한 번의 기울기를 계산하고 가중치를 업데이트 한다. 안정적 수렴을 보인다.
- 미니 배치 경사하강법은 전체 데이터를 작은 묶음으로 나눠서 학습한다. 배치 경사하강법의 안정성과 확률적 경사하강법의 빠른 속도를 가지는 방법이다. 병렬 처리가 가능한 방법이다.

(단, 배치의 크기에 따라 성능이 달라지기에 하이퍼파라미터 튜닝이 필요하다)

22. CNN의 풀링층의 역할은?

풀링층은 공간적 크기를 줄이고 연산량을 감소시키며, 특징의 위치 변화에 대한 불변성 (translation invariance)을 제공합니다. 대표적인 방법으로 Max Pooling과 Average Pooling이 있으며, 국소적인 영역에서 대표값을 추출합니다.

- Max Pooling은 특징 영역 내에서 최대값만 선택하여 출력으로 남기는 연산이다. 입력 이미지나 feature map의 공간적 크기는 줄여서 연산량은 감소시키며 (=차원 축소) 각 영역에서 가장 두드러지는 특징 (최대값)을 보존함으로써 중요한 특징을 보존한다. (=중요 특징 강조). 불필요한 세부 정보를 제거하며 일반화 성능이 향상되고 (=과적합 방지), 작고 국소적인 이동에 덜 민감해진다. (=위치 변화에 불변성)
- Average Pooling은 특징 영역 내에서 평균값만 선택하여 출력으로 남기는 연산이다. 입력의 공간적 차원은 줄여서 모델의 연산량은 감소시키며 (=차원 축소), 지역적인 대표값을 취해서 전체적인 정보나 패턴을 유지한다. (=정보의 일반화). 평균값을 사용함으로써 노이즈나 극단값에 덜 민감해진다. (=노이즈 완화)

23. CNN의 Dense Layer의 역할은?

Dense Layer는 합성곱 층과 풀링층에서 추출된 고수준의 특징들을 기반으로 분류나 회귀와 같은 최종 판단을 수행하는 역할을 한다. 전통적인 MLP와 유사하게 작동하며, 특징 간의 비선형 관계를 학습한다.

24. CNN의 stride, filter의 역할은? 필터의 가중치는 어떻게 결정되는가?

Stride: 필터가 입력 위를 이동하는 간격으로, stride가 클수록 출력 크기가 작아져 연산량이 줄고 특성 맵이 축소된다.

Filter (Kernel): 입력에서 특징을 추출하는 도구, 다양한 형태의 필터가 다양한 특징 (엣지, 코너 등)을 감지한다.

필터의 가중치는 학습과정 (~~Backtracking~~ ^{back propagation} + 경사하강법)을 통해 데이터에 맞게 자동으로 조절된다. 사람이 지정하지 않으며, 목표 함수 (예: cross entropy loss)를 최소화하도록 최적화된다.

25. RNN을 사용하는 이유와 한계점은?

시퀀스 (시간 순서가 있는 데이터. 예: 텍스트, 음성, 주가)를 처리할 수 있으며, 과거 정보를 기억하면서 순차적인 데이터에 적합하게 사용된다.

단, 긴 시퀀스에서 기울기 소실/폭주 문제 (Gradient vanishing/exploding problem)로 인해 장기 의존성 (long-term dependency) 학습이 어렵다.

- Gradient vanishing problem은 back propagation 과정에서 gradient가 점점 작아져서 초기층까지 전달되지 않는 현상이다. 특히 sigmoid와 tanh와 같은 포화 함수를 연속적으로 사용할 때 발생한다.

해결방법은

- (1) Relu 함수는 양수값의 기울기 값이 그대로 유지되기에 소실 완화가 가능하다.
- (2) 층별 분포를 정규화하며 기울기 흐름을 안정화한다.
- (3) LSTM 또는 GRU를 사용한다.

26. LSTM을 사용하는 이유와 한계점은?

LSTM은 Long Short Term Memory의 줄임말이다. RNN의 한계를 보완하여 장기 의존성을 효과적으로 학습할 수 있다. 과거의 출력값과 현재의 입력값을 모두 고려하여 현재의 출력값을 저장하기에 사용된다.

현재 상태와 gate의 구조 (입력, 출력, 망각 게이트)를 통해 정보 흐름을 조절한다.

- 입력 게이트는 현재의 입력값을 sigmoid 함수에 넣어 얼마나 기억할지를 결정한다.
- 망각 게이트는 과거의 출력값을 sigmoid 함수에 넣어 얼마나 잊을지를 결정한다.
- 출력 게이트는 현재의 입력값과 과거의 출력값을 sigmoid 함수에 넣어 어떤 정보를 현재의 출력값으로 선택할지를 결정한다.

단, 구조가 복잡하고 연산 비용이 크기에, 학습 시간이 오래 걸린다.

27. GRU를 사용하는 이유와 차별성은?

LSTM과 유사하게 장기 의존성을 학습할 수 있고, 구조가 더 단순하여 ~~높은~~ 높은 연산 비용과 빠른 학습에 장점이 있기에 사용된다. hidden state와 cell state 가 모두 존재한다.
(과거 출력값) (현재 입력값)

LSTM과 달리 hidden state 만을 유지한다.

- Reset 게이트는 과거의 출력값을 잊고 새로운 입력을 기억할지를 결정한다.
- Update 게이트는 과거의 출력값을 기억할지를 결정한다. (입력, 망각 게이트의 기능 통합)