

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

A. 활성화 함수를 사용하는 이유는 만약에 선형 함수를 사용하는 경우에는 신경망이 층을 쌓아도 결국 선형적인 계산만 하는 모델이 되기 때문에 복잡한 형태의 데이터를 학습하기 어렵다.
그래서 활성화 함수를 사용하여 각 층마다 계산된 값을 비선형으로 바꾸주어, 층이 쌓이면 쌓일수록 복잡하고 다양한 특성을 뽑아낼 수 있게 한다. Sigmoid 함수는 출력을 0~1 사이로 변환을 시켜 이진 분류에서 사용되고 Softmax는 다중 클래스 분류에서 각 클래스별 확률 분포를 동시에 계산할 때 활용된다.

Q 15. Forward propagation, Backward propagation이란?

A. Forward propagation은 입력 데이터를 신경망의 첫 번째 층부터 마지막 출력층까지 차례대로 전달하면서 각 층의 가중치와 활성화 함수를 통해 출력값을 계산하는 과정이다. 이 과정을 통해 모델이 예측값을 만들고, 그 결과가 손실 함수에 들어가 실제 값과의 차이를 계산할 수 있다.

Backward propagation은 출력층에서 계산된 오차를 다시 신경망을 거슬러 올라가며 각 가중치의 기울기, 즉 그래디언트를 구한 이를 바탕으로 가중치를 업데이트하는 과정이다. 손실 함수에 대한 편미분을 통해 각 층의 가중치가 얼마나 조정되어야 하는지 계산하고, 이를 이용해 경사하강법 같은 알고리즘으로 가중치를 수정함으로써 학습이 이루어진다.

Q 16. 손실 함수란 무엇인가? 가장 많이 사용되는 손실 함수 4가지 종류는?

A. 손실 함수는 모델이 예측한 값과 실제 값 사이의 차이를 수치적으로 평가하는 함수로, 학습 과정에서 이 값을 최소화하도록 가중치를 조정하는 기준이다. 즉, 모델이 얼마나 잘못 예측했는지 계산해주는 척도라고 볼 수 있으며 학습이 진행될수록 손실값이 줄어드는 방향으로 파라미터가 업데이트 되도록 한다. 손실 함수 중에서 많이 사용되는 함수는 평균 제곱오차(MSE), 평균 절대오차(MAE), 이진 교차 엔트로피, 범주형 교차 엔트로피가 있는데 MSE, MAE는 회귀 문제에서 주로 사용되고 이진 교차 엔트로피는 이진 분류, 범주형 교차 엔트로피는 다중 클래스 분류에서 주로 사용된다.

Q17. 옵티마이저(Optimizer)란 무엇일까? 옵티마이저와 손실 함수의 차이점은?

A. 옵티마이저는 신경망이 학습할 때 손실 함수 값을 최소화하도록 가중치와 편향을 조정해 주는 알고리즘이다. 즉, 손실 함수로 계산된 오차를 줄여가며 파라미터를 업데이트하는 역할을 한다. 반면 손실 함수는 현재 모델이 예측한 값과 실제 값 사이의 차이를 수치화해서 어느 정도로 틀렸는지 평가하는 함수이다. 따라서 손실 함수는 모델의 잘못된 정도를 계산하는 측정 기준이고, 옵티마이저는 그 손실 값을 줄이기 위해 파라미터를 어떻게 조절할지 결정하는 방법이다.

Q18. 경사하강법의 특징은? (확률적 경사하강법, 배치 경사하강법, 미니 배치 경사하강법)

A. 경사하강법은 손실 함수를 최소화하기 위해 가중치 방향으로 조금씩 조정해 나가는 방법이다.

전체 데이터를 한번에 사용하여 가중치를 구하는 방식은 배치 경사하강법이라고 한다.

반면에 매번 데이터를 하나씩 무작위로 골라서 샘플을 만들고 그 샘플에 대한 기울기로 가중치를 업데이트하는 방식은 확률적 경사하강법이다. 이 방식은 계산량은 적지만 값이 불안정하다.

마지막으로, 데이터를 작은 묶음으로 나눠 그 묶음마다 기울기를 계산해 업데이트하는 방식은 미니 배치 경사하강법이라 한다.

Q19. 교차검증, k-fold 교차검증의 의미와 차이

A. 교차검증은 모델이 어떤 데이터든 안정하게 적용되는지, 즉 일반화 성능을 평가하기 위해 데이터를 여러개의 부분 집합으로 나눠서 훈련과 검증을 반복하는 기법이다.

그 중에서 k-fold 교차검증은 전체 데이터를 균등하게 k개의 폴드로 나누고, 그 중 하나를 검증 세트로 사용하고 나머지 k-1개를 훈련 세트로 사용하는 과정을 k번 반복하는 방식이다.

Q20. 하이퍼파라미터 튜닝이란 무엇인가?

A. 하이퍼파라미터 튜닝은 모델을 학습하기 전에 사용자가 직접 설정해 주어야 하는 값을 최적화하기 위해 여러 값을 시퀀스하며 성능을 비교하는 과정을 말한다. 하이퍼파라미터의 예시로는 학습률, 배치 크기, 은닉층 수 등이 있다. 하이퍼파라미터 튜닝을 하는 이유는 모델의 일반화 성능을 높이기 위해서이다.

Q 21. CNN 합성곱의 역할은?

A. CNN에서 합성곱 연산은 입력 이미지에서 작은 영역을 일정한 크기의 필터로 훑으면서, 요소별 곱셈과 덧셈을 통해 국소적인 부분을 추출하는 역할을 한다. 이 과정에서 모델은 이미지 전체를 한번에 처리하는 대신 먼저 저수준의 특징을 먼저 감지하고, 이후에 여러 층을 거치며 점차 복잡한 형태의 고수준 특징을 학습한다.

Q 22. CNN 풀링층의 역할은?

A. 풀링층은 합성곱층에서 추출된 특징 맵의 가로 세로 크기를 줄여서 연산량과 파라미터 수를 감소시켜 컴퓨터가 처리해야 할 일이 줄어들어 계산 속도가 빨라지고, 이미지가 조금 움직이거나 모양이 바뀌어도 같은 특징을 가리키기 쉬워진다. 하지만 풀링층을 너무 자주 쓰면 중요한 정보까지 사라질 수 있으므로 합성곱층과 풀링층을 적절하게 사용하는 것이 중요하다.

Q 23. CNN의 Dense Layer의 역할은?

A. Dense Layer는 CNN에서 마지막에 위치에 있는 층으로, 앞서 여러 층에서 얻은 특징은 한데 모아 결과를 만들어 주는 역할을 한다. Dense Layer 안에 뉴런은 각각 어떤 특징이 중요한지 가중치를 매겨, 모든 정보를 합쳐 최종 예측을 한다.

Q 24. CNN의 stride, fitter의 역할? 필터의 가중치는 어떻게 결정되는가?

A. stride는 필터가 이미지를 위에서 한번에 얼마나 이동할지를 정하는 값이다.

stride가 클수록 연산해야 할 숫자가 줄어들지만 세세한 정보는 놓칠 수 있고 작을수록 더 세세한 정보를 얻는다. 필터는 이미지 위를 작은 사각형 모양으로 움직이며 그 영역 안에 픽셀값을 곱하고 더해서 특징을 뽑아내는 역할을 한다. 이 작은 사각형을 커널이라고도 부른다. 필터 안에 가중치는 처음에는 작은 랜덤 값으로 정해져있다. 학습이 진행되면서 정답 레이블과 필터가 뽑아낸 출력값을 비교한 뒤 오차가 작아지도록 가중치를 조정한다.

Q 25. RNN을 사용하는 이유와 한계점?

A. RNN은 입력한 데이터가 시간이나 순서에 따라 연결되어 있을 때, 이전 정보를 다음 단계로 전달하면서 처리하는 구조를 가진 신경망이다. RNN은 과거에 입력된 내용을 기억하여 새로운 입력과 함께 학습한다. 이런 구조로 RNN은 앞뒤 맥락을 파악하는 작업에 적합하다. 하지만 RNN은 한계점이 존재하는데 첫번째로는 입력이 길어질수록 과거의 정보가 점점 사라지는 기울기 소실 문제가 생긴다. 두번째로는 각 단계가 순차적으로 계산되어야해 학습속도가 느려진다. 또 단순한 RNN 구조로는 복잡한 장기 의존성을 학습하기 어려워 아주 긴 문에는 한계가 있다.

Q 26. LSTM을 사용하는 이유와 한계점?

A. LSTM은 RNN의 기울기 소실 문제를 해결하기 위해 만든 구조이다. RNN은 시간이 지날수록 과거 정보가 사라지지만 LSTM은 셀 상태라는 기억 공간과 정보를 열고 닫는 게이트를 사용해 중요한 정보는 오래 보관하고 불필요한 정보는 버린다. 하지만 LSTM은 구조가 복잡해서 계산량이 많아서 학습속도가 느린 단점이 있다.

Q 27. GRU를 사용하는 이유와 차별성?

A. GRU는 RNN의 기울기 소실 문제를 해결하면서 구조를 최대한 간단하게 만든 순환 신경망이다. LSTM이 3개의 게이트와 셀 상태를 사용해 정보를 조정하는데 비해 GRU는 업데이트 게이트와 리셋 게이트 2개만 둔다. 이로 인해 LSTM보다 학습속도가 빠르다는 장점이 있다.