

인공지능 기초를 위한 FAQ

1. Activation Function 함수를 사용하는 이유? softmax, sigmoid 함수의 차이?

활성화 함수는 인공 신경망에서 입력 신호에 비선형성을 부여하여 단순한 선형회로는 해결할 수 없는 복잡한 문제(이미지, 음성, 자연어 등)의 패턴을 학습할 수 있게 해준다.

- sigmoid 함수는 입력 값을 0과 1 사이의 확률로 변환하여 이진 분류에 적합하다.
- softmax 함수는 여러 클래스가 있을 때 각 클래스에 속할 확률을 출력하며, 출력값의 총합이 1이 되도록 정규화 한다. 따라서 다중 클래스 분류에 사용된다.

2. Forward propagation, Backward propagation이란?

- Forward : 입력층에서 시작하여 각 층의 가중치와 편향, 활성화 함수를 거쳐 순차적으로 출력을 계산하는 과정이다. 최종적으로는 모델의 예측값을 생성하며, 이 과정을 통해 모델이 현재 입력에 대해 어떤 출력을 내놓는지를 알 수 있다.
- Backward : 모델의 예측값과 실제값과의 차이를 손실함수를 통해 계산한 후, 이 오차를 출력층에서 입력층 방향으로 역전파하여 각 층의 가중치와 편향을 수정한다. 이는 경사하강법을 통해 손실을 줄이는 방향으로 가중치를 업데이트 하는 핵심 메커니즘이다.

3. 손실함수란 무엇인가? 가장 많이 사용하는 손실함수 4가지는?

모델이 얼마나 잘못 예측하고 있는지를 수치적으로 평가하는 함수이다. 예측값과 실제값 간의 차이를 측정하며, 이 값을 줄이기 위해 모델이 학습된다. 예를 들어 회귀 문제에서는 MSE, 분류 문제에서는 Cross-Entropy가 주로 사용된다.

- MSE (Mean Squared Error): 평균제곱오차, 회귀 문제에 사용.
- MAE (Mean Absolute Error): 절대값 평균오차, 이상치에 덜 민감.
- Cross-Entropy: 분류 문제에서 확률 기반 예측 성능 측정.
- Hinge Loss: 주로 SVM에 사용하는 손실함수, 마진 기반 분류에 적합.

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

옵티마이저는 손실 함수를 최소화 하기 위해 네트워크의 가중치와 편향을 조정하는 알고리즘이다. 대표적인 옵티마이저는 SGD, Adam, RMSprop 등이 있으며, 학습률 조절과 모멘텀 등 다양한 기법이 결합되어 있다.

손실 함수는 현재 모델이 얼마나 틀렸는지를 정량화 하여 알려준다. 옵티마이저는 이 손실 값을 바탕으로 가중치를 어떻게 업데이트 할지를 결정한다. 즉, 손실은 '문제의 정도', 옵티마이저는 '해결 방법'이다.

5. 정사화장법 의미는? (확률적 정사화장법, 배치 정사화장법, 미니 배치 정사화장법)

가중치를 따라 손실 함수의 값을 줄이기 위해 가중치를 반복적으로 조정하는 방법이다. 손실 함수의 기울기가 작아지는 방향으로 파라미터를 이동하며 최적값을 찾는다.

- 확률적 정사화장법 (SGD)

매번 하나의 샘플만을 사용해서 가중치를 업데이트하여 빠르게 학습하지만 손실함수의 값이 불안정하게 변화한다. 속도는 빠르지만 노이즈가 있다.

- 배치 정사화장법

전체 데이터를 사용해서 손실 함수의 기울기를 계산하고 업데이트 하는 방식으로, 안정적인 학습이 가능하지만 계산 비용이 크다.

- 미니 배치 정사화장법

데이터 셋을 여러 개로 작은 배치로 나누어 학습하는 방식으로, 학습의 안정성과 속도 간 균형을 맞춘 방식이다. 일반적으로 가장 많이 사용된다.

6. 교차검증, k-폴더 교차검증의 의미와 차이

교차검증이란 데이터를 여러 조각으로 나누어, 각 조각을 한 번씩 검증용으로 사용하면서 여러 모델을 훈련하는 기법이다. 데이터에 과적합 하지 않도록 일반화 성능을 평가 하는 데 유용하다. k-폴더 교차검증은 데이터를 k개의 폴드로 나눈, 그 중 하나를 검증용도, 나머지를 학습용으로 사용하며 k번 학습 및 평가를 수행한다. 최종 모델 평가는 k개의 평가 결과 평균으로 판단한다.

7. 하이퍼 파라미터 튜닝이란 무엇인가?

모델 구조나 학습 알고리즘에서 사람이 설정해주는 파라미터(예: 학습률, 배치크기, 층수 등)를 실험적으로 조절하여 최적의 성능을 내도록 설정하는 과정이다. 그리드 서치, 랜덤 서치, 베이지안 최적화 등 있다.

8. CNN의 합성곱의 역할은?

입력 이미지에서 국소적인 특징(에지, 색상 변화 등)을 추출한다. 필터(커널)가 입력과 연산하여 특징 맵을 생성하며, 이를 통해 객체 인식 및 분류에 적합한 특징을 자동으로 학습한다.

9. CNN의 풀링층의 역할은?

합성곱으로 생성된 특징맵에서 가장 중요한 정보를 유지하면서 공간적 크기를 줄여 계산량을 감소시키고, 과적합을 방지하는 효과가 있다. 주로 Max Pooling과 Average Pooling 이 사용된다.

10. CNN의 Dense Layer의 역할은?

합성곱과 풀링을 통해 추출된 특징을 바탕으로 최종 분류 또는 예측을 수행한다. 이 층은 각 뉴런이 이전 층의 모든 뉴런과 연결되어 있고, 일반적인 다층 퍼셉트론과 같다.

11. CNN의 state, filter의 역할? 필터의 가중치는 어떻게 결정되는가?

state란 필터가 한 번에 얼마나 많이 이동할지를 결정하여 출력 크기에 영향을 준다.

필터 (커널)은 특징을 감지하는 도구이며, 필터가 많을수록 다양한 특징을 학습할 수 있다.

필터의 가중치는 처음에는 무작위로 설정 되지만, 학습을 거치면서 오차를 줄이는 방향으로 방향을 찾아서 점점 정교해 조정된다.

12. RNN을 사용하는 이유와 한계점은?

문장, 음성, 시계열 데이터처럼 순차적인 데이터에서 앞뒤 맥락을 반영하여 예측할 수 있다.

일반적인 신경망과 달리 시간 흐름을 고려할 수 있다. 한계점은 긴 시퀀스를 처리할 수록 과거 정보가 희미해지는 '장기 의존성' 문제가 있으며, 기울기 소실이나 폭주로 인해 학습이 어려울 수 있다.

13. LSTM을 사용하는 이유와 한계점은?

게이트 구조(입력, 삭제, 출력 게이트)를 통해 정보를 선택적으로 기억하거나 잊을 수 있어 RNN보다 더 긴 시퀀스를 안정적으로 처리할 수 있다. 한계점은 구조가 복잡해 연산량이 많고 학습 시간이 오래 걸린다. 또한, 레이어가 깊지 않으면 과적합 가능성도 존재한다.

14. GRU를 사용하는 이유와 차별성은?

LSTM과 비슷한 성능을 내면서도 구조가 단순하며 계산이 더 빠르고 구현이 쉬우며, 적은 레이어에서도 효과적이다. 차별성은 게이트 수가 두 개 (업데이트, 리셋)로 간편하고 계산이 적어 효율성이 높으며, 메모리 사용량도 적다.

15. Dense Layer란 무엇인가?

입력 뉴런과 출력 뉴런이 모두 연결된 층으로, 고차원의 특징을 조합해 최종적인 분류나 회귀 등 결과의 결정을 내린다. CNN이나 RNN 등 다양한 구조에서 마지막 예측 단계를 위한 사용된다.