

## learning-AI101 : 인공지능 기초를 위한 FAQ

- 임규연 작성, 2024.06.21. ~ 현재
- 작성 단계
  - 1. 각 question에 대해 먼저 블로그, 논문, 저널 등을 survey한 후 키워드를 추려서 요약 정리함 (2024.06.21. ~ 2024.06.30.)
  - 2. 요약 정리를 llama-2 local 모델에 넣어서 자세한 설명을 요구함 (2024.06.30. ~ 2024.07.03)
    - token의 한계로 문자열을 잘라서 대입할 수 밖에 없었음 (그래서 시간 오래 걸림)
    - freund (<https://github.com/lky473736/freund>)에 대입함
  - 3. 본격적인 학습 돌입 (2024.07.04 ~ )

### 1. 인공지능에서 지능에 해당하는 기능은 무엇인가?

- **학습(Learning)**: 인공지능의 학습 능력은 데이터에서 패턴과 관계를 인식하고 이를 기반으로 새로운 정보에 대해 학습하는 것을 의미합니다. 머신러닝 알고리즘을 사용하여 다양한 유형의 데이터를 처리하고, 이를 통해 새로운 데이터를 예측하거나 분류하는 능력을 개발합니다. 예를 들어, 이미지 인식 모델은 수많은 이미지 데이터를 학습하여 새로운 이미지가 무엇을 나타내는지 인식할 수 있습니다.
- **추론(Reasoning)**: 추론 능력은 학습된 정보를 바탕으로 결론을 도출하는 과정입니다. 이는 논리적 사고를 통해 문제를 해결하거나 새로운 정보를 생성하는 능력을 포함합니다. 예를 들어, 체스 인공지능은 현재 체스판의 상태를 분석하고 가능한 수를 예측하여 최적의 수를 결정합니다.
- **문제 해결(Problem Solving)**: 문제 해결 능력은 주어진 문제를 분석하고 해결책을 찾아내는 과정을 포함합니다. 이는 최적화 알고리즘을 사용하여 복잡한 문제를 해결하거나, 제약 조건을 고려한 해결책을 찾아내는 능력을 의미합니다. 예를 들어, 물류 시스템에서 최적의 경로를 찾아내는 알고리즘은 여러 가지 변수와 제약 조건을 고려하여 최적의 경로를 계획합니다.
- **이해(Understanding)**: 이해 능력은 자연어 처리 기술을 통해 인간의 언어와 감정을 이해하는 것을 포함합니다. 이는 텍스트 분석, 감정 분석, 언어 번역 등의 기술을 통해 구현됩니다. 예를 들어, 챗봇은 사용자의 질문을 이해하고, 그에 맞는 적절한 답변을 제공합니다.

### 2. 인공지능의 종류 4가지에 대해서 설명하시오 (지도학습, 비지도학습, 반지도학습, 강화학습)

- **지도학습(Supervised Learning)**: 지도학습은 레이블이 있는 데이터를 사용하여 모델을 학습시키는 방법입니다. 이는 입력 데이터와 정답(레이블) 쌍을 사용하여 모델이 학습을 통해 패턴을 인식하고, 새로운 데이터에 대해 예측을 수행합니다. 예를 들어, 스팸 이메일 필터링에서는 대량의 이메일 데이터와 각 이메일이 스팸인지 아닌지에 대한 레이블을 사용하여 학습합니다. 주요 알고리즘으로는 선형 회귀, 로지스틱 회귀, 결정 트리, 서포트 벡터 머신(SVM) 등이 있습니다.
- **비지도학습(Unsupervised Learning)**: 비지도학습은 레이블이 없는 데이터를 사용하여 데이터의 구조를 학습하는 방법입니다. 이는 데이터를 클러스터링하거나, 차원을 축소하여 데이터의 패턴을 이해하는 데 사용됩니다. 예를 들어, 고객 세분화에서는 고객 데이터를 클러스터링하여 유사한 특성을 가진 고객 그룹을 식별합니다. 주요 알고리즘으로는 K-평균 클러스터링, 주성분 분석(PCA), t-SNE 등이 있습니다.
- **반지도학습(Semisupervised learning)**: 학습 시에 레이블이 일부 주어져서, 입력에 대한 결과값이 없는 미분류 데이터 (unlabeled data)를 지도학습에 사용하는 방법입니다. 반지도 학습의 가정은 총 3가지로 아래와 같습니다.
  - 평활성 가정 : 가까이 있는 점들은 서로 같은 부류에 속할 가능성 있음
  - 군집 가정 : 같은 군집에 속하는 데이터는 동일 부류에 속할 가능성 있음
  - 매니폴드 가정 : 원래 차원보다 낮은 차원의 매니폴드에 데이터가 분포할 가능성 있음 (매니폴드 : 데이터가 존재할 수 있는 일종의 공간)

- **강화학습(Reinforcement Learning):** 강화학습은 에이전트가 환경과 상호작용하며 보상을 최대화하는 방향으로 학습하는 방법입니다. 이는 에이전트가 행동을 선택하고, 그 행동의 결과로 보상을 받으며, 이를 통해 최적의 행동을 학습합니다. 예를 들어, 게임 AI는 게임 환경에서 다양한 행동을 시도하고, 승리할 수 있는 전략을 학습합니다. 주요 알고리즘으로는 Q-러닝, 딥 Q-네트워크(DQN), 정책 그라디언트 등이 있습니다.

### 3. 전통적인 프로그래밍 방법과 인공지능 프로그램의 차이점은 무엇인가?

- **전통적인 프로그래밍:** 전통적인 프로그래밍에서는 프로그래머가 명시적인 규칙과 로직을 작성합니다. 이는 입력 데이터에 따라 미리 정해진 결과를 출력하는 방식입니다. 예를 들어, if-else 조건문을 사용하여 특정 조건에 따라 다른 동작을 수행하는 프로그램을 작성할 수 있습니다. 이러한 프로그램은 명확한 규칙과 조건에 따라 작동하며, 예측 불가능한 상황에서는 동작이 제한적입니다.
- **인공지능 프로그램:** 인공지능 프로그램에서는 데이터를 통해 패턴과 규칙을 학습하여 예측 모델을 생성합니다. 이는 입력 데이터에 기반하여 학습된 모델이 결과를 예측하는 방식입니다. 예를 들어, 스팸 이메일 필터링 프로그램은 대량의 이메일 데이터를 학습하여 스팸 이메일을 필터링하는 규칙을 자동으로 학습합니다. 이러한 프로그램은 데이터에서 학습된 패턴을 기반으로 예측을 수행하므로, 새로운 데이터나 예측 불가능한 상황에서도 유연하게 동작할 수 있습니다.

### 4. 딥러닝과 머신러닝의 차이점은 무엇인가?

- **머신러닝(Machine Learning):** 머신러닝은 데이터를 사용하여 패턴을 학습하는 알고리즘을 개발하는 분야입니다. 다양한 알고리즘이 포함되며, 데이터에서 특징을 추출하고 이를 기반으로 예측 모델을 학습합니다. 예를 들어, 간단한 회귀 모델은 주어진 데이터에서 독립 변수와 종속 변수 간의 관계를 학습하여 새로운 데이터를 예측합니다. 주요 알고리즘으로는 선형 회귀, 로지스틱 회귀, 결정 트리, 서포트 벡터 머신(SVM), K-최근접 이웃(K-NN) 등이 있습니다.
- **딥러닝(Deep Learning):** 딥러닝 또한 머신러닝의 일종으로, 인공신경망을 이용한 머신러닝이라고 보면 됩니다. 딥러닝은 인공신경망, 특히 다층 신경망을 사용하여 복잡한 패턴을 학습하는 분야입니다. 대량의 데이터와 높은 계산 능력이 필요하며, 이미지 인식, 음성 인식, 자연어 처리 등 다양한 분야에서 뛰어난 성능을 보입니다. 예를 들어, 컨볼루션 신경망(CNN)은 여러 층의 신경망을 사용하여 이미지의 특징을 추출하고, 이를 기반으로 이미지를 분류합니다. 딥러닝의 주요 알고리즘으로는 CNN, 순환 신경망(RNN), 장단기 메모리 네트워크(LSTM) 등이 있습니다.

### 5. Classification과 Regression의 주된 차이점은?

- **분류(Classification):** 분류는 데이터를 특정 카테고리로 분류하는 작업을 의미합니다. 출력 값은 이산적(discrete) 값이며, 각 카테고리에 속하는지 여부를 예측합니다. 예를 들어, 이메일을 스팸 또는 비스팸으로 분류하는 작업은 분류 문제입니다. 주요 알고리즘으로는 로지스틱 회귀, 결정 트리, 랜덤 포레스트, 서포트 벡터 머신(SVM) 등이 있습니다.
- **회귀(Regression):** 회귀는 연속적인 값을 예측하는 작업을 의미합니다. 출력 값은 연속적(continuous) 값이며, 주어진 독립 변수들에 기반하여 종속 변수의 값을 예측합니다. 예를 들어, 주택의 면적, 위치 등의 정보를 기반으로 주택 가격을 예측하는 작업은 회귀 문제입니다. 주요 알고리즘으로는 선형 회귀, 다항 회귀, 결정 트리 회귀, 랜덤 포레스트 회귀 등이 있습니다.

### 6. 머신러닝에서 차원의 저주(curse of dimensionality)란?

- 차원의 저주란 고차원 데이터 공간에서 발생하는 문제들을 의미합니다. 차원이 증가할수록 데이터 공간의 부피가 기하급수적으로 증가하여 데이터 포인트 간의 거리가 멀어지고, 모델의 학습이 어려워지는 현상이 발생합니다. 이는 데이터가 희소해져서 모델이 일반화되지 못하고 과적합(overfitting)되기 쉬워지는 문제를 야기합니다. 예를 들어, 2차원 공간에서 데이터 포인트가 밀집되어 있다면, 고차원 공간에서는 같은 수의 데이터 포인트

가 훨씬 더 넓게 분산되어 있어 유의미한 패턴을 학습하기 어렵습니다. 이를 해결하기 위해 차원 축소 기법 (PCA, t-SNE 등)을 사용하여 데이터의 차원을 줄이고, 중요한 특징만을 남기는 방법이 사용됩니다.

## 7. Dimensionality Reduction는 왜 필요한가?

차원 축소 (dimensionality reduction)은 고차원의 원본 데이터의 의미 있는 feature을 원래의 차원에 가깝게 유지할 수 있도록 고차원에서 저차원 공간으로 데이터를 변환하는 것을 의미합니다.

- **모델의 복잡성 감소:** 차원을 줄이면 모델의 복잡성이 감소하여 과적합(overfitting)을 방지할 수 있습니다. 이는 모델이 불필요한 특징에 대해 과도하게 학습하는 것을 방지하고, 더 일반화된 모델을 생성하는 데 도움을 줍니다. 예를 들어, 주성분 분석(PCA)을 사용하여 고차원 데이터에서 중요한 주성분만을 선택하여 차원을 축소할 수 있습니다.
- **계산 효율성 향상:** 차원 축소를 통해 데이터의 차원이 줄어들면 연산 속도가 향상되고, 모델의 학습 시간이 단축됩니다. 이는 대량의 데이터를 처리하거나, 실시간으로 예측을 수행해야 하는 경우에 특히 유용합니다. 예를 들어, 고차원 이미지 데이터를 2D 또는 3D로 축소하여 연산 속도를 높일 수 있습니다.
- **데이터 시각화 용이:** 차원을 축소하면 데이터를 시각화하기 쉬워집니다. 이는 데이터의 패턴을 이해하고, 분석하는 데 도움을 줍니다. 예를 들어, t-SNE를 사용하여 고차원 데이터를 2D 또는 3D로 시각화하여 데이터의 클러스터링 구조를 쉽게 파악할 수 있습니다.

## 8. Ridge와 Lasso의 공통점과 차이점? (Regularization, 규제, Scaling)

- **공통점:** Ridge와 Lasso는 모두 규제(regularization)를 통해 모델의 과적합을 방지하는 방법입니다. 규제는 비용 함수에 페널티 항을 추가하여 모델이 복잡해지는 것을 방지하고, 일반화 성능을 향상시키는 역할을 합니다.
  - **목적:** 모델의 일반화 성능을 향상시키고, 과적합을 방지하기 위함.
  - **사용:** 비용 함수에 페널티 항을 추가하여 가중치 값을 제한.
- **Ridge 회귀:** Ridge 회귀는 L2 정규화를 사용합니다. 이는 가중치의 제곱합을 페널티로 추가하여 모든 특성의 가중치를 작게 만듭니다.
  - **페널티 항:** 가중치의 제곱합.
  - **특징:** 모든 특성의 가중치를 작게 만들지만, 가중치가 0이 되지는 않음. 이는 모든 특성이 모델에 기여할 수 있도록 합니다.
- **Lasso 회귀:** Lasso 회귀는 L1 정규화를 사용합니다. 이는 가중치의 절댓값 합을 페널티로 추가하여 불필요한 특성의 가중치를 0으로 만듭니다.
  - **페널티 항:** 가중치의 절댓값 합.
  - **특징:** 일부 가중치를 0으로 만들어 불필요한 특성을 제거함. 이는 특성 선택의 효과를 가지며, 중요한 특성만을 남깁니다.

## 9. Overfitting vs. Underfitting

- **Overfitting(과적합):** 과적합은 모델이 훈련 데이터에 너무 맞춰져서 새로운 데이터에 대한 예측 성능이 떨어지는 현상을 의미합니다. 이는 모델이 훈련 데이터의 노이즈까지 학습하여 일반화되지 못한 결과를 초래합니다.
  - **원인:** 너무 복잡한 모델, 충분하지 않은 데이터, 과도한 학습.
  - **해결 방법:** 규제 적용(Ridge, Lasso), 더 많은 데이터 수집, 모델 단순화, 교차 검증 사용.
- **Underfitting(과소적합):** 과소적합은 모델이 훈련 데이터의 패턴을 제대로 학습하지 못하는 현상을 의미합니다. 이는 모델이 너무 단순하여 데이터의 복잡한 관계를 파악하지 못한 결과를 초래합니다.
  - **원인:** 너무 단순한 모델, 충분하지 않은 학습, 중요한 특성 누락.
  - **해결 방법:** 더 복잡한 모델 사용, 더 많은 특징 추가, 모델의 학습 시간 증가.

## 10. Feature Engineering과 Feature Selection의 차이점은?

- **Feature Engineering:** Feature Engineering은 원시 데이터를 모델이 이해할 수 있도록 변환하고 새로운 특징을 생성하는 과정입니다. 이는 데이터의 의미를 더 잘 표현하도록 새로운 변수를 만들거나, 기존 변수를 변형하는 작업을 포함합니다.
  - **예시:** 날짜 데이터를 년, 월, 일로 분리하거나, 텍스트 데이터를 벡터로 변환.
  - **목적:** 모델이 데이터를 더 잘 이해하고, 성능을 향상시키도록 함.
- **Feature Selection:** Feature Selection은 모델 학습에 중요한 특징을 선택하고, 불필요한 특징을 제거하는 과정입니다. 이는 데이터의 차원을 줄이고, 모델의 학습 속도를 높이며, 과적합을 방지하는 역할을 합니다.
  - **예시:** 상관관계 분석을 통해 중요한 변수를 선택하거나, Lasso 회귀를 사용하여 중요하지 않은 특징 제거.
  - **목적:** 중요한 특징만을 남겨 모델의 성능을 최적화하고, 계산 효율성을 높임.

## 11. 전처리(Preprocessing)의 목적과 방법? (노이즈, 이상치, 결측치)

- **목적:** 전처리는 데이터의 품질을 개선하여 모델의 성능을 향상시키는 것을 목표로 합니다. 이는 데이터의 신뢰성을 높이고, 학습 과정에서 발생할 수 있는 오류를 줄이기 위함입니다.
- **방법:**
  - **노이즈 제거:** 불필요한 데이터를 제거하여 신호 대 잡음비를 개선합니다. 예를 들어, 이미지 데이터에서 노이즈를 제거하거나, 텍스트 데이터에서 불필요한 문자를 제거할 수 있습니다.
    - **기법:** 필터링, 스무딩, 노이즈 감소 알고리즘 사용.
  - **이상치 처리:** 이상치는 데이터에서 비정상적으로 벗어난 값을 의미합니다. 이를 처리하지 않으면 모델의 성능이 저하될 수 있습니다.
    - **기법:** 이상치를 제거하거나, 다른 값으로 대체(Imputation)하는 방법 사용.
  - **결측치 처리:** 결측치는 데이터에서 값이 누락된 부분을 의미합니다. 결측치를 처리하지 않으면 모델의 학습에 부정적인 영향을 미칠 수 있습니다.
    - **기법:** 결측치를 평균, 중앙값 등으로 대체하거나, 결측치가 포함된 행을 제거하는 방법 사용.
  - **정규화(Normalization):** 데이터를 일정한 범위로 변환하여 모델의 학습을 돕습니다. 예를 들어, 모든 값을 0과 1 사이로 스케일링할 수 있습니다.
    - **기법:** Min-Max 스케일링, Z-스코어 정규화.
  - **표준화(Standardization):** 데이터의 평균을 0, 표준편차를 1로 변환하여 정규 분포를 따르게 합니다.
    - **기법:** 평균과 표준편차를 사용한 표준화.

## 12. EDA(Exploratory Data Analysis)란? 데이터의 특성 파악(분포, 상관관계)

- **정의:** 탐색적 데이터 분석(EDA)은 데이터 세트를 분석하여 주요 특성을 요약하고, 이를 통해 패턴, 관계 및 이상치를 발견하는 과정입니다. 이는 데이터 과학 프로젝트의 초기 단계에서 수행되며, 데이터의 구조를 이해하고, 적절한 모델링 전략을 세우는 데 도움이 됩니다.
- **목적:** 데이터의 분포, 이상치, 상관관계 등 다양한 특성을 시각화하고 이해하여, 데이터에 대한 직관적 이해를 돕고, 데이터 전처리 및 모델링에 필요한 통찰을 제공합니다.
- **방법:**
  - **데이터 요약:** 통계적 요약(평균, 중앙값, 표준편차 등)을 통해 데이터의 기본 특성을 파악합니다.
  - **시각화:** 히스토그램, 박스플롯, 산점도 등 다양한 시각화 기법을 사용하여 데이터의 분포와 관계를 직관적으로 파악합니다.
  - **상관관계 분석:** 상관 행렬, 히트맵 등을 통해 변수 간의 상관관계를 분석하고, 중요한 변수들을 식별합니다.
  - **이상치 탐지:** 박스플롯, z-점수 등을 사용하여 데이터의 이상치를 탐지하고, 이를 처리하는 방법을 고려합니다.

## 13. 회귀에서 절편과 기울기가 의미하는 바는? 딥러닝과 어떻게 연관되는가?

- **절편(Intercept):** 회귀 모델에서 절편은 독립 변수의 값이 0일 때 종속 변수의 예상 값을 의미합니다. 즉, 회귀 직선이 y축과 만나는 점입니다. 예를 들어, 주택 가격 예측 모델에서 절편은 주택 면적이 0일 때 예상 주택 가격을 나타냅니다.
- **기울기(Slope, coef):** 기울기는 독립 변수의 값이 한 단위 증가할 때 종속 변수의 예상 값이 얼마나 증가하거나 감소하는지를 나타냅니다. 기울기는 회귀 직선의 경사를 나타내며, 변수 간의 관계 강도를 나타냅니다.
- **딥러닝과의 연관성:** 딥러닝의 기본 구성 요소인 신경망에서도 가중치와 편향(절편과 유사한 역할)을 사용합니다. 가중치는 입력 데이터가 출력에 미치는 영향을 조절하며, 편향은 모델이 예측을 조정하는 데 사용됩니다. 회귀 분석의 절편과 기울기는 단일 뉴런의 가중치와 편향으로 해석될 수 있습니다.

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

- **이유:** 활성화 함수는 신경망의 각 뉴런에서 입력 신호를 처리하여 비선형성을 추가합니다. 이를 통해 신경망이 복잡한 패턴을 학습하고, 다양한 문제를 해결할 수 있게 합니다. 활성화 함수가 없으면 신경망은 단순한 선형 모델에 불과합니다.
- **Sigmoid 함수:** Sigmoid 함수는 입력 값을 0과 1 사이의 값으로 변환합니다. 주로 이진 분류 문제에서 출력 뉴런에 사용됩니다.
  - 수식:  $\sigma(x) = \frac{1}{1 + e^{-x}}$
  - 특징: 출력 값이 0과 1 사이로 제한되어 있으며, 확률 값을 예측할 때 유용합니다. 그러나 경사 소실 (vanishing gradient) 문제를 야기할 수 있습니다.
- **Softmax 함수:** Softmax 함수는 여러 클래스 중 하나를 선택하는 다중 클래스 분류 문제에서 사용됩니다. 입력 값을 각 클래스에 대한 확률로 변환합니다.
  - 수식:  $\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$
  - 특징: 출력 값이 0과 1 사이의 값으로 변환되며, 모든 출력 값의 합이 1이 되도록 합니다. 이를 통해 클래스 확률을 예측할 수 있습니다.

#### 15. Forward propagation, Backward propagation이란?

- **Forward Propagation(순전파):** 순전파는 입력 데이터가 신경망을 통해 전달되어 출력 값을 계산하는 과정입니다. 각 뉴런의 출력 값은 활성화 함수를 통해 계산되며, 이는 다음 층의 뉴런으로 전달됩니다. 순전파는 입력 데이터로부터 예측 값을 생성하는 데 사용됩니다.
- **Backward Propagation(역전파):** 역전파는 출력 값과 실제 값 간의 오차를 계산하고, 이를 통해 신경망의 가중치와 편향을 조정하는 과정입니다. 이는 오차의 기울기를 계산하여 가중치를 업데이트하는 데 사용됩니다. 역전파는 순전파의 반대 방향으로 오차를 전파하며, 경사 하강법을 사용하여 가중치를 최적화합니다.

#### 16. 손실함수란 무엇인가? 가장 많이 사용하는 손실함수 4가지 종류?

- **정의:** 손실함수는 모델의 예측 값과 실제 값 간의 차이를 측정하는 함수입니다. 이는 모델의 성능을 평가하고, 가중치를 업데이트하는 데 사용됩니다. 손실함수 값이 작을수록 모델의 예측이 정확함을 의미합니다.
- **종류:**
  - **Mean Squared Error (MSE):** 회귀 문제에서 많이 사용되며, 예측 값과 실제 값 간의 차이의 제곱 평균을 계산합니다.
    - 수식:  $\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
  - **Cross-Entropy Loss:** 분류 문제에서 많이 사용되며, 예측된 확률 분포와 실제 분포 간의 차이를 측정합니다.
    - 수식:  $\text{Cross-Entropy} = -\sum_i y_i \log(\hat{y}_i)$
  - **Mean Absolute Error (MAE):** 예측 값과 실제 값 간의 차이의 절대값 평균을 계산합니다. MSE와 달리 이상치에 덜 민감합니다.
    - 수식:  $\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$

- **Hinge Loss:** 주로 서포트 벡터 머신(SVM)에서 사용되며, 예측 값과 실제 값 간의 마진을 계산합니다.

- 수식: 
$$\text{Hinge Loss} = \sum_{i=1}^n \max(0, 1 - y_i \hat{y}_i)$$

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

- **옵티마이저:** 옵티마이저는 손실함수를 최소화하기 위해 모델의 가중치를 업데이트하는 알고리즘입니다. 이는 경사 하강법을 기반으로 하여, 손실함수의 기울기를 계산하고, 이를 통해 가중치를 조정하여 최적의 모델을 찾습니다.
- **차이점:**
  - **손실함수:** 모델의 예측 성능을 평가하는 함수로, 예측 값과 실제 값 간의 차이를 측정합니다.
  - **옵티마이저:** 손실함수를 최소화하기 위해 가중치를 업데이트하는 알고리즘으로, 손실함수의 기울기를 사용하여 최적의 가중치를 찾습니다.

## 18. 경사하강법 의미는? (확률적 경사하강법, 배치 경사하강법, 미니 배치 경사하강법)

- **경사하강법(Gradient Descent):** 경사하강법은 손실함수를 최소화하기 위해 기울기를 따라 가중치를 업데이트하는 최적화 알고리즘입니다. 이는 기울기가 가장 가파르게 내려가는 방향으로 가중치를 조정하여 최적의 모델을 찾습니다.
- **종류:**
  - **배치 경사하강법(Batch Gradient Descent):** 전체 데이터 세트를 사용하여 기울기를 계산하고, 한 번에 가중치를 업데이트합니다. 이는 정확한 기울기를 계산할 수 있지만, 큰 데이터 세트에서는 계산 비용이 많이 듭니다.
  - **확률적 경사하강법(Stochastic Gradient Descent, SGD):** 각 데이터 포인트에 대해 기울기를 계산하고, 가중치를 업데이트합니다. 이는 계산 속도가 빠르지만, 기울기가 불안정하여 수렴 속도가 느릴 수 있습니다.
  - **미니 배치 경사하강법(Mini-batch Gradient Descent):** 데이터 세트를 작은 배치로 나누어 각 배치에 대해 기울기를 계산하고, 가중치를 업데이트합니다. 이는 배치 경사하강법과 확률적 경사하강법의 장점을 결합하여, 계산 비용을 줄이면서 안정적인 수렴을 도모합니다.

## 19. 교차검증, K-fold 교차검증의 의미와 차이

- **교차검증(Cross-Validation):** 교차검증은 모델의 일반화 성능을 평가하기 위해 데이터를 여러 번 나누어 학습하고, 검증하는 방법입니다. 이는 데이터를 학습 세트와 검증 세트로 나누어, 모델이 과적합되는 것을 방지하고, 성능을 평가합니다.
- **K-fold 교차검증:** K-fold 교차검증은 데이터를 K개의 폴드로 나누어, 각 폴드마다 한 번씩 검증 세트로 사용하고, 나머지는 학습 세트로 사용하는 방법입니다. 이를 통해 K번의 학습과 검증을 수행하여, 모델의 평균 성능을 평가합니다.
  - **차이점:** 교차검증은 단순히 데이터를 나누어 학습하고 검증하는 반면, K-fold 교차검증은 데이터를 K개의 폴드로 나누어 여러 번 학습과 검증을 반복합니다. 이는 모델의 성능을 더 안정적으로 평가할 수 있도록 합니다.

## 20. 하이퍼파라미터 튜닝이란 무엇인가?

- **정의:** 하이퍼파라미터 튜닝은 머신러닝 모델의 성능을 최적화하기 위해 하이퍼파라미터의 값을 조정하는 과정입니다. 하이퍼파라미터는 모델 학습 과정에서 수동으로 설정해야 하는 값으로, 학습률, 규제 강도, 은닉층의 수 등이 포함됩니다.
- **방법:**
  - **그리드 서치(Grid Search):** 모든 가능한 하이퍼파라미터 조합을 탐색하여 최적의 값을 찾는 방법입니다. 이는 계산 비용이 많이 들지만, 가장 최적의 조합을 찾을 수 있습니다.

- **랜덤 서치(Random Search):** 하이퍼파라미터 공간에서 임의의 조합을 샘플링하여 최적의 값을 찾는 방법입니다. 이는 그리드 서치보다 계산 비용이 적지만, 최적의 조합을 찾을 확률이 낮아질 수 있습니다.
- **베이지안 최적화(Bayesian Optimization):** 이전 평가 결과를 바탕으로 다음 평가할 하이퍼파라미터 조합을 선택하는 방법입니다. 이는 효율적으로 최적의 값을 찾을 수 있습니다.

## 21. CNN의 합성곱의 역할은?

- **정의:** 합성곱(convolution)은 입력 이미지에서 특징을 추출하는 연산입니다. 이는 이미지의 작은 영역에 필터를 적용하여, 해당 영역의 특징 맵(feature map)을 생성합니다.
- **역할:**
  - **특징 추출:** 입력 이미지의 다양한 특징(에지, 코너 등)을 효과적으로 추출합니다.
  - **공간 불변성:** 합성곱 연산은 이미지의 이동, 회전, 크기 변화에 대해 강건한 특징을 추출할 수 있습니다.
  - **차원 축소:** 입력 이미지의 크기를 줄이면서, 중요한 특징을 유지합니다. 이는 계산 효율성을 높이고, 모델의 복잡도를 줄이는 데 도움을 줍니다.

## 22. CNN의 풀링층의 역할은?

- **정의:** 풀링(pooling)은 입력 특징 맵의 공간 크기를 줄이는 연산입니다. 주로 최대 풀링(max pooling)과 평균 풀링(average pooling)이 사용됩니다.
- **역할:**
  - **차원 축소:** 입력 특징 맵의 크기를 줄여, 계산 비용을 감소시킵니다.
  - **특징 요약:** 중요한 특징을 유지하면서, 불필요한 세부 사항을 제거합니다.
  - **불변성 강화:** 입력 데이터의 이동, 회전, 크기 변화에 대해 강건한 특징을 추출합니다.

## 23. CNN의 Dense Layer의 역할은?

- **정의:** Dense Layer(완전 연결 층)는 각 입력 뉴런이 모든 출력 뉴런과 연결된 층입니다. 이는 주로 신경망의 마지막 단계에서 사용됩니다.
- **역할:**
  - **고차원 특징 결합:** 합성곱 층과 풀링 층에서 추출된 고차원 특징을 결합하여, 최종 예측을 수행합니다.
  - **복잡한 패턴 학습:** 모든 입력 뉴런이 출력 뉴런과 연결되어 있어, 복잡한 패턴을 학습할 수 있습니다.

## 24. CNN의 stride, filter의 역할? 필터의 가중치는 어떻게 결정되는가?

- **Stride(스트라이드):** 스트라이드는 필터가 입력 데이터를 이동하는 간격을 나타냅니다. 스트라이드 값이 클수록 출력 특징 맵의 크기가 작아집니다.
  - **역할:** 계산 효율성을 조절하고, 출력 특징 맵의 크기를 조정합니다.
- **Filter(필터):** 필터는 입력 데이터의 작은 영역에 적용되어 특징을 추출하는 매개변수입니다.
  - **역할:** 입력 데이터의 다양한 특징을 추출합니다.
  - **가중치 결정:** 필터의 가중치는 학습 과정에서 데이터에 의해 자동으로 조정됩니다. 초기에는 임의의 값으로 설정되며, 역전파를 통해 손실을 최소화하는 방향으로 업데이트됩니다.

## 25. RNN을 사용하는 이유와 한계점은?

- **이유:** 순환 신경망(RNN)은 순차적 데이터를 처리하고, 시간 종속성을 학습할 수 있습니다. 이는 텍스트, 시계열 데이터, 음성 등 순차적 데이터를 다루는 데 적합합니다.
  - **특징:** 이전 단계의 출력을 현재 단계의 입력으로 사용하여, 데이터의 순차적 의존성을 모델링합니다.
- **한계점:**
  - **기울기 소실 문제:** 역전파 과정에서 기울기가 점차 작아져, 장기 종속성을 학습하기 어려워질 수 있습니다.

- **계산 비용:** 순차적 데이터 처리가 필요하여, 계산 비용이 높아질 수 있습니다.

## 26. LSTM을 사용하는 이유와 한계점은?

- **이유:** LSTM(Long Short-Term Memory)은 RNN의 기울기 소실 문제를 해결하기 위해 고안된 모델입니다. 이는 장기 종속성을 효과적으로 학습할 수 있습니다.
  - **특징:** 입력 게이트, 출력 게이트, 망각 게이트를 통해 정보를 선택적으로 기억하고, 잊을 수 있습니다.
- **한계점:**
  - **복잡성:** LSTM의 구조는 단순 RNN보다 복잡하여, 학습과 추론 시간이 길어질 수 있습니다.
  - **계산 비용:** 많은 파라미터를 가지므로, 계산 비용이 높아질 수 있습니다.

## 27. GRU를 사용하는 이유와 차별성은?

- **이유:** GRU(Gated Recurrent Unit)는 LSTM의 변형 모델로, 유사한 성능을 유지하면서 구조를 단순화하여 계산 비용을 줄입니다.
  - **특징:** 업데이트 게이트와 리셋 게이트를 통해 정보를 선택적으로 기억하고, 잊을 수 있습니다.
- **차별성:**
  - **구조적 단순성:** GRU는 LSTM보다 단순한 구조를 가지며, 학습과 추론 속도가 빠릅니다.
  - **성능:** 비슷한 상황에서 LSTM과 유사한 성능을 보이지만, 계산 비용이 적습니다.

## 28. 결정트리에서 불순도(Impurity) – 지니 계수(Gini Index)란 무엇인가?

- **불순도(Impurity):** 결정 트리에서 불순도는 노드의 순수성을 측정하는 지표입니다. 순수한 노드는 하나의 클래스만 포함하며, 불순한 노드는 여러 클래스가 혼합되어 있습니다.
- **지니 계수(Gini Index):** 지니 계수는 불순도를 측정하는 방법 중 하나입니다. 값이 0에 가까울수록 노드가 순수하며, 0.5에 가까울수록 불순합니다.
  - **수식:**  $Gini = 1 - \sum_{i=1}^n p_i^2$ , 여기서  $p_i$ 는 클래스  $i$ 의 비율입니다.

## 29. 앙상블이란 무엇인가?

- **정의:** 앙상블(Ensemble)은 여러 개의 모델을 결합하여, 단일 모델보다 더 나은 예측 성능을 얻는 방법입니다. 이는 각 모델의 예측을 조합하여, 전체적인 성능을 향상시킵니다.
- **방법:**
  - **배깅(Bagging):** 여러 모델을 병렬로 학습시키고, 예측 결과를 평균내거나 다수결로 결정합니다.
  - **부스팅(Boosting):** 여러 모델을 순차적으로 학습시키고, 이전 모델의 오차를 보완하는 방식으로 예측 성능을 향상시킵니다.

## 30. 부트 스트랩핑(Bootstrapping)이란 무엇인가?

- **정의:** 부트 스트랩핑은 원본 데이터 세트에서 중복을 허용하여 여러 번 샘플링하는 기법입니다. 이는 통계적 추정과 모델 평가에서 사용됩니다.
- **역할:** 모델의 안정성을 평가하고, 다양한 샘플에서의 성능을 확인하는 데 도움을 줍니다.

## 31. 배깅(Bagging)이란 무엇인가?

- **정의:** 배깅(Bootstrap Aggregating)은 부트 스트랩핑과 앙상블 학습을 결합한 방법입니다. 여러 모델을 병렬로 학습시키고, 예측 결과를 평균내거나 다수결로 결정합니다.
- **역할:** 모델의 분산을 줄이고, 예측 성능을 향상시킵니다. 배깅은 특히 결정 트리와 같은 고분산 모델에서 효과적입니다.

## 32. 주성분 분석(PCA) 이란 무엇인가?



- **정의:** 주성분 분석(PCA)은 고차원 데이터를 저차원으로 변환하는 기법입니다. 데이터의 분산을 최대한 보존하면서, 주요 축을 찾아 데이터를 투영합니다.
- **역할:**
  - **차원 축소:** 데이터의 주요 정보를 유지하면서, 차원을 축소하여 계산 비용을 줄입니다.
  - **특징 추출:** 데이터의 주요 특징을 추출하여, 모델의 성능을 향상시킬 수 있습니다.

### 33. Dense Layer란 무엇인가?

- **정의:** Dense Layer(완전 연결 층)는 각 입력 뉴런이 모든 출력 뉴런과 연결된 층입니다.
- **역할:** 합성곱 층과 풀링 층에서 추출된 특징을 결합하여, 최종 예측을 수행합니다. Dense Layer는 복잡한 패턴을 학습하고, 다양한 문제를 해결하는 데 사용됩니다.