**REPORT**

로고, 상징, 등록 상표, 텍스트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

|  |  |
| --- | --- |
| 제목 | 2주차 - 인공지능 기초를 위한 FAQ |

|  |  |
| --- | --- |
| 과목명 | 인공지능개론 |
| 학과 | 컴퓨터공학부(컴퓨터공학전공) |
| 학번 | 202334765 |
| 이름 | 하준휘 |

인공지능 기초를 위한 FAQ

**1. 인공지능에서 지능에 해당하는 기능은 무엇인가?**

인공지능의 ‘지능’은 인간의 사고 과정과 유사하게 작업을 처리하는 능력을 말한다. 곧 ‘인간의 지능’을 뜻하는 것이다. 이 ‘지능’을 구현하기 위한 기능으로는

1. 학습: 데이터를 분석하고 패턴을 찾아 스스로 개선.
2. 추론: 주어진 정보를 바탕으로 논리적 결론 도출.
3. 문제 해결: 문제를 해결하기 위한 최적의 방법 찾기.
4. 인지: 주변 환경을 감지, 이해 (예: 이미지, 소리 인식).
5. 자연어 처리: 사람의 언어를 이해, 작문.
6. 계획 및 의사 결정: 목표를 설정하고 최적의 행동을 결정.
7. 창의성: 새로운 아이디어나 콘텐츠를 생성.

등이 있다.

**2. 인공지능의 종류 3가지에 대해서 설명하시오 (지도학습, 반지도학습, 강화학습)**

a. 지도학습: 정답이 있는 데이터로 학습.

b. 반지도학습: 일부만 정답이 있는 데이터, 나머지는 정답이 없는 데이터. 실제로 사용하기 위한 모델을 학습시킬 때는 정답이 없는 데이터를 사용해야 할 때가 많은데, 이 때 정답이 있는 데이터를 살짝 섞어주면 학습 정확도가 개선되는 경향이 있다고 함.

c. 강화학습: 정답이 없지만, AI가 낸 결과에 따라 보상과 벌점을 주어 AI가 좋은 결과를 향해 다가갈 수 있도록 유도. 뚜렷한 정답이 없는 문제(게임 AI 등)에 주로 쓰임.

d. (추가)비지도학습: 정답이 없는 데이터로 학습. AI가 데이터의 구조, 특징을 스스로 찾아냄.

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

전통적인 프로그래밍은 인간이 직접 명확한 규칙을 코드로 작성하는 방식으로 이루어지고, 인공지능은 AI가 스스로 데이터를 학습하여 규칙을 찾는 방식으로 이루어진다.

|  |  |  |
| --- | --- | --- |
| 구분 | 전통적인 프로그래밍 | 인공지능 프로그래밍 |
| 규칙 설정 | 사람이 직접 작성 | AI가 데이터에서 학습 |
| 데이터 활용 | 입력 → 규칙 적용 → 출력 | 데이터 → 학습 → 예측 |
| 일반적인 적용 분야 | 명확한 규칙이 있는 문제 | 복잡한 패턴을 찾아야 하는 문제 |

**4. 딥러닝과 머신러닝의 차이점은 무엇인가?**

a. 머신러닝: AI의 하위 분류. AI가 데이터를 기반으로 스스로 학습하는 기술을 의미. 사람이 데이터의 특징을 직접 설정.

b. 딥러닝: 머신러닝의 하위 분류. 인공신경망 사용. AI가 데이터의 특징을 알아서 추출. 일반적으로 대량의 데이터가 필요하지만, 복잡한 문제도 해결 가능.

요약하자면, 머신러닝은 사람이 데이터의 특징을 설정하는 것, 딥러닝은 AI가 알아서 데이터의 특징을 알아내는 것이다.

**5. Classification과 Regression의 주된 차이점은?**

a. Classification(분류): 데이터를 미리 정해진 카테고리 내에서 분류하는 문제. 출력값이 이산적이고 서로 관련이 없음(예: ‘개’와 ‘고양이’는 서로 거리를 따질 수 없는 값).

b. Regression(회귀): 데이터를 기반으로 연속적인 숫자 범위 내에서 하나의 값을 예측해내는 문제. 출력값이 연속적이고 서로 관련이 있음(예: 50-51 보다 50-60이 더 먼 값).

일반적으로 Classification은 그룹을 나누는 문제(예: 스팸메일 or 정상메일), Regression은 숫자를 예측하는 문제(예: 5년 후 삼성전자 주가 예측, 학생 성적 예측)이다.

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

데이터의 차원이 증가할수록(= 특징이 많아질수록) 데이터 분석, 모델 학습이 어려워지는 것을 말한다.

[차원의 저주가 일어나는 이유]

a. 데이터 희소성 증가: 차원이 증가할수록 공간이 넓어져 점(데이터)이 없는 빈 공간이 많아짐(= 같은 크기 공간에서 데이터가 희소해짐).

b. 거리 계산이 어려워짐: 머신러닝 모델에서 사용하는 점(데이터)간 거리 계산 방식은, 차원이 증가할수록 점 사이 거리들의 차이가 점점 줄어듦. 결국 차원이 너무 크면 점 사이 거리가 전부 비슷비슷해짐.

c. 연산량 증가: 차원이 증가할수록 데이터 크기가 기하급수적으로 증가해 계산에 필요한 시간과 메모리 비용이 급격히 증가.

**7. Dimensionality Reduction는 왜 필요한가?**

Dimensionality Reduction(차원 축소)는 데이터의 특징 개수(차원)를 줄이는 것을 말한다. 이것이 필요한 이유는,

a. 차원의 저주를 줄일 수 있음.

b. 연산 속도 향상.

c. 과적합 방지: 불필요한 특징(노이즈, 원하는 답을 도출하는 데 방해되거나 도움이 되지 않는 특징)이 많으면 AI가 그런 특징까지 학습해서 성능이 나빠질 수 있음.

d. 데이터 시각화 가능: 2, 3차원까지 차원을 줄인다면 시각화 가능.

등이 있고, 추가로 차원 축소 방법에는

a. 주성분 분석(20번 문제): 데이터를 가장 잘 설명하는 주요 특징만 뽑기

b. t-SNE와 UMAP: 고차원 데이터를 2~3차원으로 줄여 시각화할 때 주로 사용

등이 있다.

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

Rigge, Lasso는 머신러닝의 규제(Regularization) 기법 중 하나다.

규제란, 머신러닝 모델이 과적합되는 것을 막기 위해 가중치에 제약 조건을 추가해 AI의 행동을 규체하는 것을 말한다. 일반적으로 규제를 넣으면 너무 큰 가중치를 줄여주므로, 모델을 단순하게 만들어 다양한 데이터에 대응할 수 있도록 할 수 있다.

가중치란, AI가 학습의 결과물로서 만들어 낸 계산 규칙을 말한다고 할 수 있다. AI가 학습할 데이터를 받으면, 그 데이터를 가지고 계산할 때 ‘어떤 값’을 넣어 계산해야 목표로 하는 출력에 가깝게 나오는지를 학습하게 된다. 여기서 그 ‘어떤 값’이 바로 가중치이다.

[Ridge와 Lasso의 공통점, 차이점]

공통점:

a. 과적합을 방지하기 위한 규제의 한 종류

b. 특징 스케일링 필요: 두 규제 모두 특징들의 크기 차이가 크면 성능이 나빠질 수 있음. 따라서 표준화 또는 정규화로 각 특징들의 값 크기를 비슷하게 맞춰주어야 함.

차이점:

|  |  |  |
| --- | --- | --- |
|  | Ridge (L2 규제) | Lasso (L1 규제) |
| 규제 방식 | 가중치(w)의 제곱합을 규제 (L2 정규화) | 가중치(w)의 절댓값을 규제 (L1 정규화) |
| 특징 선택 | 일부 특징의 가중치를 0에 가깝게 만들지만, 완전히 0이 되진 않음 (즉, 모든 특징을 유지) | 일부 특징의 가중치를 완전히 0으로 만들어 특징을 일부만 적용(특징 선택) |
| 사용 예시 | 모든 변수의 영향력이 중요할 때 (예: 경제 예측, 날씨 예측) | 중요하지 않은 특징을 자동으로 제거하고 싶을 때 (예: 희소한 데이터, 유전자 분석) |

**9. Overfitting vs. Underfitting**

Overfitting, Underfitting 둘 모두 AI를 이용할 때 원하는 만큼 성능이 나오지 않는 경우를 말한다. 하지만 훈련 데이터에서는 성능이 좋았었는지, 나빴었는지에 따라 경우를 둘로 나눈다.

|  |  |  |
| --- | --- | --- |
|  | Overfitting (과적합) | Underfitting (과소적합) |
| 의미 | 훈련 데이터에 너무 치중해, 훈련 데이터에서는 성능이 높지만, 다른 데이터에서는 성능이 떨어지는 상태 | AI가 제대로 학습하지 못해, 훈련 데이터에서조차 성능이 떨어지는 상태. 말 그대로 성능 부족. |
| 원인 | 모델이 너무 복잡(특징이 많거나 가중치가 큼), 노이즈 데이터까지 학습 | 모델이 너무 단순(특징 또는 학습량 부족) |
| 해결 방법 | - 모델 단순화(특징 줄이기)  - 규제  - test 데이터 양 늘리기  - 노이즈 데이터 없애기 | - 더 복잡한 모델 사용  - 특징 추가  - 데이터 양 늘리기 |

**10. Feature Engineering과 Feature Selection의 차이점은?**

Feature Engineering은 데이터의 특징을 추가하는 것, Feature Selection은 데이터의 특징을 줄이는(일부만 선택하는) 것을 말한다.

|  |  |  |
| --- | --- | --- |
|  | Feature Engineering (특징 엔지니어링) | Feature Selection (특징 선택) |
| 의미 | 새로운 특징을 만들거나 변형하는 것 | 기존 특징 중 중요한 것만 선택해 남기는 것 |
| 목적 | 데이터를 더 풍부하게 만들어 모델이 데이터를 더 잘 이해하도록 함 | 불필요한 특징을 제거해 모델이 더 적은 계산량으로 필요한 정보만 학습하도록 함 |

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

전처리란, 모델이 데이터를 더 잘 이해할 수 있도록 데이터를 정리, 변형하는 과정을 말한다. 데이터에는 노이즈, 이상치, 결측치 등 문제가 있을 수 있다.

a. 노이즈(Noise): 데이터에서 원하는 출력과 상관이 없는 부분. 예: 사람 음성 데이터의 배경음과 잡음

해결방법

1. 평균 필터: 특정한 특징은 여러 데이터의 평균값을 사용

2. 이상치 제거: 일정 기준 이상으로 벗어난 값 제거

b. 이상치(Outlier): 데이터에서 너무 튀는 값. 예: 키 데이터의 300cm 같은 잘못된 값

해결방법

1. IQR(사분위수 범위) 분석

2. Z-score(표준점수) 분석

3. 대체(Imputation): 이상치를 평균 or 중앙 or 최빈값 등으로 변경

c. 결측치(Missing Value): 데이터에서 누락된 값. 예: 설문조사에서 일부 문항에 답을 하지 않은 사람이 있음

해결방법

1. 삭제: 해당 데이터 전체 삭제

2. 대체: 결측치를 평균 or 중앙 or 최빈값 등으로 채움

3. 예측: 다른 AI 모델로 결측값 예측

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

EDA(탐색적 데이터 분석, Exploratory Data Analysis)란, 데이터를 시각화하고 통계 분석해 특징과 레이블의 전체적인 특성을 파악하는 과정이다. **EDA를 잘 해야 좋은 모델을 만들 수 있다!**

[EDA의 목적]

a. 데이터에 어떤 패턴이 있는지 살펴봄

b. 이상치, 결측치 등 데이터의 문제점 파악

c. 변수 간 상관관계 분석

d. 모델을 위한 적절한 전처리 방법 결정

[EDA에서 주로 하는 작업]

a. 데이터 분포 확인: 데이터가 어떻게 퍼져 있고, 이상치 등 값에 이상이 있는지, 평균, 중앙값, 표준편차 등이 어떤 값인지 확인하는 작업. 히스토그램으로 값들의 빈도를 확인하거나, 박스플롯으로 데이터 특징의 중앙값, 이상치를 확인하는 등.

b. 변수 간 상관관계 분석: 데이터의 두 특징 간에 어떤 관계가 있는지 확인하는 작업. 두 변수의 관계를 산점도로 시각적으로 확인하거나, 상관계수로 -1~1의 값으로 확인할 수 있음(예: 공부 시간이 많을수록 점수가 높으면 양수(+), 공부 시간이 많을수록 점수가 낮으면 음수(-)).

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

회귀 분석(Resression)이란, 입력값을 사용해 출력값을 예측하는 모델을 말한다.

[회귀에서 절편, 기울기가 의미하는 바]

예를 들어, 회귀 분석 중 가장 간단한 선형 회귀 모델을 보자. 선형 회귀란, 분산되어 있는 데이터들의 중간 지점을 찾아, 입력 변수(X)와 출력 변수(Y) 사이의 관계를 가장 잘 나타내는 직선을 찾고, 이를 통해 X값이 주어졌을 때 Y값을 예측해내는 통계적 모델이다.

선형 회귀의 직선은 y = ax + b의 식을 갖는다. 여기서 a는 기울기, b는 y절편인데, AI 모델이 하는 일이 이 기울기와 절편을 찾아 직선을 구성하는 것이다. 즉 기울기와 절편은 회귀 분석의 결과가 되는 직선을 의미한다.

[딥러닝과 어떻게 연관되는가]

딥러닝의 기본 단위가 바로 선형 회귀다. 딥러닝에서는 선형 회귀처럼 입력 X를 가중치 W와 곱하고, 편향 b를 더한 후 활성화 함수를 적용한다. 즉 Y = WX + b 식을 이용하고, W가 선형 회귀의 기울기, b가 선형 회귀의 y절편이 되니, 결국 기울기(W)와 절편(b)이 딥러닝의 뉴런이 학습하는 핵심 값인 것이다!

추가로, 가중치(W)는 입력 신호의 중요도를 조정하는 값이고(예: 키가 몸무게에 미치는 영향이 클수록 W가 커짐), 편향(b)은 모델이 더 유연하게 학습하도록 하는 조정값이다(예: 모든 입력이 0이어도 편향값에 의해 결과값이 존재하도록 함).

**14. 교차검증, K-fold 교차검증의 의미와 차이**

a. 교차검증: 머신러닝 모델이 어떤 데이터에서든 잘 작동하는지 평가하기 위한 방법. 데이터를 여러 개로 나눠서 각각 검증하는 방식.

b. K-Fold 교차검증: 교차검증 중 가장 많이 쓰임. 데이터를 K개의 폴드(Fold)로 나누고, K번 반복 학습해 성능 평가.

[K-Fold 교차검증 과정]

1. 데이터를 K개 그룹(Fold)으로 나눔

2. 1번째 그룹을 test, 나머지 그룹을 train 데이터로 모델 학습 후 평가  
 이후 2번째 그룹을 test, 나머지 그룹을 train 데이터로 모델 학습 후 평가  
 이를 K번 반복 -> K개의 모델 성능 측정값이 나옴

3. 모델의 최종 성능 = 성능 측정값들의 평균

둘의 차이는, 교차검증은 데이터를 여러 번 나눠서 모델을 학습 및 테스트하는 모든 방법을 아우르는 말이고, K-Fold 교차검증은 교차검증 중 하나의 방법으로써 데이터를 K개로 나눠 모델을 학습 및 평가하는 방식이라는 것이다.

**15. 하이퍼파라미터 튜닝이란 무엇인가?**

하이퍼파라미터 튜닝(Hyperparameter Tuning)이란, AI 모델의 성능을 최적화하기 위해 최적 하이퍼파라미터 값을 찾는 과정을 말한다.

하이퍼파라미터란, 모델이 학습할 때 사전에 설정해야 하는 값을 말한다. 이 값은 학습 과정에서는 조정되지 않는다. 보통 사람이 직접 조정하거나 최적화 기법을 사용한다.

하이퍼파라미터예: DT의 트리 깊이(max\_depth), 최소 샘플 수(min\_samples\_split) 또는 신경망의 학습률(learning\_rate), 배치 크기(batch\_size), 레이어 개수

튜닝 방법 예:

a. Grid Search(그리드 서치): 미리 정한 값 조합을 하나하나 전부 탐색. 모든 경우를 볼 수 있지만 오래 걸림

b. Random Search (랜덤 서치): 지정된 범위 내 임의의 값 조합을 선택해 탐색. 빠르지만 최적값을 놓칠 수 있음

c. Bayesian Optimization (베이지안 최적화): 이전 결과를 바탕으로 더 나은 값을 탐색. 효율적이지만 설정이 복잡함

d. Gradient-based Optimization (그라디언트 기반 최적화): 그라디언트(기울기)를 이용해 값을 좋은 방향으로 조정, 딥러닝에서 자주 사용(학습률 등)

**16. 결정트리에서 불순도(Impurity) – 지니 계수(Gini Index)란 무엇인가?**

우선, 결정 트리(DT, Decision Tree)란 데이터를 트리 구조로 분류, 예측하는 머신러닝 모델이다. 특정 조건을 기준으로 True냐 False냐에 따라 데이터를 나뭇가지가 갈라져 퍼지듯이 분류해나간다. 여기서 조건을 기준으로 나누는 한 칸을 노드(Node)라고 한다.

[불순도, 지니 계수란]

a. 불순도: DT의 특정 노드 내에서 데이터가 섞여 있는 정도

b. 지니 계수: 불순도를 측정하는 지표 중 하나. 0에 가까울수록 불순도가 낮고, 1에 가까울수록 불순도가 높음.

**17. 앙상블이란 무엇인가?**

앙상블이란, 여러 개 모델을 조합해 성능을 높이는 기법이다. 대표적인 기법으로는

a. 배깅(Bagging, Bootstrap Aggregating(부트스트랩 집계)): 여러 모델을 독립적으로 돌린 뒤 평균 or 투표로 결과

b. 부스팅(Boosting): 이전 모델이 틀린 데이터를 다음 모델이 더 신경써서 학습

c. 스태킹(Stacking): 여러 모델의 예측을 메타 모델(최종 모델)에 입력해 예측. 다양한 모델의 강점을 조합할 수 있지만, 구현이 어렵고 연산량이 배깅, 부스팅보다 많음.

정도가 있다.

**18. 부트 스트랩핑(bootstrapping)이란 무엇인가?**

부트스트래핑이란, 주어진 데이터에서 여러 개의 샘플을 랜덤하게 뽑아 분석 및 학습하는 과정을 반복해 주어진 데이터를 더 잘 대표하는 추정값을 얻는 기법을 말한다. 특히 앙상블의 배깅 기법에서 자주 활용한다.

자세한 부트스트래핑의 사용 사례로는,

a. 머신러닝 – 배깅, RF 모델: 랜덤 포레스트(RF) 모델이 바로 위의 앙상블 - 배깅 기법을 활용하는 모델. 이 RF 모델이 여러 개 결정트리(DT)를 학습시킬 때, 각 DT를 부트스트랩 샘플을 이용해 독립적으로 학습시킴.

b. 통계학 - 신뢰구간 추정: 모집단의 분포를 모를 때, 부트스트래핑으로 평균, 분산 등의 신뢰구간 추정 가능.

c. A/B 테스트 결과 분석: A/B 테스트란 UI, 알고리즘 등 컨텐츠를 A안, B안으로 나눠 어떤 안이 조회수 등에서 더 효과적인지 체크하는 테스트임. 여기서 샘플을 반복 추출해 A, B안 사이 평균 차이가 유의미한지 검정.

**19. 배깅(Bagging)이란 무엇인가?**

배깅은 앙상블의 한 기법이다. 부트스트래핑을 활용해 여러 모델을 독립적으로 학습하고, 그 결과를 평균(회귀 문제) 또는 투표(분류 문제) 방식으로 결합해 낸다.

장점: 각 모델이 독립적으로 학습되므로, 특정 모델이 과적합하더라도 전체 모델 성능에는 크게 영향을 주지 않아 과적합 문제가 적게 발생.

단점: 결과를 해석하기가 어렵고, 계산 비용이 높음.

**20. 주성분 분석(PCA) 이란 무엇인가?**

주성분 분석(PCA, Principal Component Analysis)은 차원(특징) 축소 기법 중 하나다. 고차원 데이터의 여러 특징 중, 답을 내는 데 있어 중요한 특징만 남기고 나머지 특징은 없애는 방식이다.

[주성분 분석의 핵심 개념과 목표]

a. 차원이 축소되어야 함

b. 데이터의 분산(Variance)을 최대한 보존해야 함

\*예를 들어 사진을 흑백으로 변환한다고 해 보자. 컬러 사진은 일반적으로 한 픽셀에 R, G, B 3개 값이 포함되어 있다. 여기서 흑백 사진을 만드려면 이 3개 값 전부와 최대한 비슷한(= 3개 값 전부를 가장 잘 대표(표현)할 수 있는) 하나의 값을 찾아야 한다. PCA는 이처럼 데이터의 중요한 정보를 최대한 유지하면서 차원(특징)을 줄이는 과정이다.

c. 데이터 간 상관성을 제거해야 함(상관성(비례 등) 있는 특징들을 하나로 묶어야 함)

\*예를 들어 학생의 시험 성적 데이터를 분석한다고 해 보자. 수학 점수와 과학 점수는 서로 어느 정도 비례할 가능성이 크다. 그러므로 두 점수를 모두 사용할 필요가 없다. 대신 두 점수 모두를 대표(표현)할 수 있는 값 하나로 줄이면 더 효율적으로 분석할 수 있게 된다. PCA에서는 중복되는 부분을 제거하고 새로운 기준(주성분)으로 데이터를 정리해야 한다.

[주성분 분석 동작 과정]

1. 데이터 정규화: 각 특징(특성)들이 서로 다른 단위와 범위를 가지므로, 비교할 수 있도록 표준화 또는 정규화 수행

2. 공분산 행렬 계산: 데이터의 특성 간 상관관계를 분석하여 어떤 특성이 서로 강한 관계(비례관계 등)를 가지는지 파악

3. 고유값 분해 (or SVD): 상관성이 높은 특성들을 하나의 새로운 축(주성분)으로 묶고, 데이터를 가장 잘 설명하는 방향(주성분 축)을 찾음

4. 주성분 선택: 고유값이 큰 주성분(데이터의 분산을 많이 설명하는 축)부터 k개 선택

5. 새 주성분으로 데이터 변환: 기존 데이터를 선택한 \*\*주성분 축에 투영(projection)\*\*하여 변환 → 차원 축소 및 정보 압축

[주성분 분석 동작 과정(공간에 축을 놓는 경우라고 가정하고 쉽게 설명)]

a. 데이터 정규화: 여러 특징들의 축을 같은 공간에 거리 왜곡 없이 놓기 위해서 표준화 or 정규화

b. 공분산 행렬 계산: 특징들의 축을 공간에 놓았을 때 어떤 축이 어떤 축과 가깝고 어떤 축과 먼지(= 어떤 특징이 어떤 특징과 연관성(비례 관계)이 큰지) 계산

c. 고유값 분해: 축들이 놓인 이 공간에서 특정한 한 세트의 축 묶음을 최대한 잘 대표해 줄 수 있는 축 방향 찾기

d. k개의 주성분 선택: 1~3번을 통해 찾은, 축 묶음을 가장 잘 대표하는 k개의 축을 선택

e. 새 주성분으로 데이터 변환: 선택한 축에다, 그 축이 대표하는 축 묶음에 있는 모든 값(데이터)들을 규칙에 따라 하나의 값으로 압축해 넣어 줌