

1주차 퀴즈

1. 컴퓨터 비전(**Computer Vision**)의 의미는 무엇인가요?

- ① 컴퓨터가 인간의 언어를 이해하여 해석하고, 복잡한 문제를 해결하는 인공지능의 한 분야
ex) gpt, gemini
- ② 컴퓨터가 인간의 시각처럼 시각 데이터를 이해하여 의미있는 정보를 추출하는 인공지능의 한 분야
ex) 자율주행 자동차, 로봇틱스의 시각부분

2. 회귀와 분류에 대한 설명으로 옳지 않은 것은?

- ① 회귀는 연속된 수를 예측하는 문제로써 주로 주택 가격이나 주식 가격 등을 예측하는 문제에 적용된다.
- ② 분류는 연속된 수를 예측하는 문제로써 주로 주택 가격이나 주식 가격 등을 예측하는 문제에 적용된다.

3. 인공 신경망에서 선형 층(**Linear Layer**)만 쓰지 않고 비선형 활성화 함수(**Activation Function**)를 사용하는 주된 이유는 무엇인가요?

- ① 모델의 학습 속도를 획기적으로 낮추기 위해서
- ② 여러 개의 선형 층을 쌓아도 결국 하나의 선형 함수로 표현되는 것을 방지하기 위해서
- ③ 데이터의 차원을 축소시키기 위해서
- ④ 메모리 사용량을 줄이기 위해서
- ⑤ 데이터를 모두 양수로 만들기 위해서

4. 손실 함수(**Loss Function**)의 역할로 가장 적절한 설명은 무엇인가요?

- ① 모델이 예측한 값과 실제 정답 사이의 차이를 수치화한다.
- ② 데이터의 정답(Label)을 자동으로 생성한다.
- ③ 모델의 가중치를 무작위로 초기화한다.
- ④ 활성화 함수의 출력을 0과 1 사이로 변환한다.
- ⑤ 경사하강법의 학습률을 자동으로 결정한다.

5. 경사하강법(**Gradient Descent**)에서 가중치를 업데이트할 때, 손실 함수의 기울기(**Gradient**)를 사용하는 이유는 무엇인가요?

- ① 손실 값이 가장 큰 지점을 찾기 위해서
- ② 가중치를 항상 양수 값으로 유지하기 위해서
- ③ 기울기의 반대 방향으로 이동하여 손실 함수가 최소가 되는 지점을 찾기 위해서
- ④ 기울기가 0인 지점에서 멀어지기 위해서
- ⑤ 데이터의 평균값을 계산하기 위해서

6. 오차 역전파(**Backpropagation**) 알고리즘의 핵심적인 원리는 무엇인가요?

- ① 가중치를 앞에서부터 순서대로 더해가는 과정
- ② 연쇄 법칙(Chain Rule)을 이용하여 출력층의 오차를 입력층 방향으로 전달하는 과정
- ③ 데이터의 순서를 무작위로 섞는 과정
- ④ 입력 데이터에 노이즈를 추가하여 학습하는 과정
- ⑤ 모델의 층을 최대한 많이 쓰는 과정

7. 파이토치에서 `torch.tensor([1, 2, 3])`을 실행했을 때 생성되는 객체의 성격으로 옳은 것은?

- ① 단순한 파이썬의 리스트(List)와 동일하게 동작한다.
- ② 다차원 배열을 계산하기 위한 파이토치의 기본 자료구조이다.
- ③ 한 번 생성하면 내부의 값을 절대 변경할 수 없다.
- ④ 수치 데이터가 아닌 텍스트 데이터만 저장할 수 있다.
- ⑤ 신경망 모델 그 자체를 의미한다.

8. 파이토치에서 여러 개의 층(Layer)을 순차적으로 쓸 때 하나의 모델로 정의할 때 사용하는 클래스는 무엇인가요?

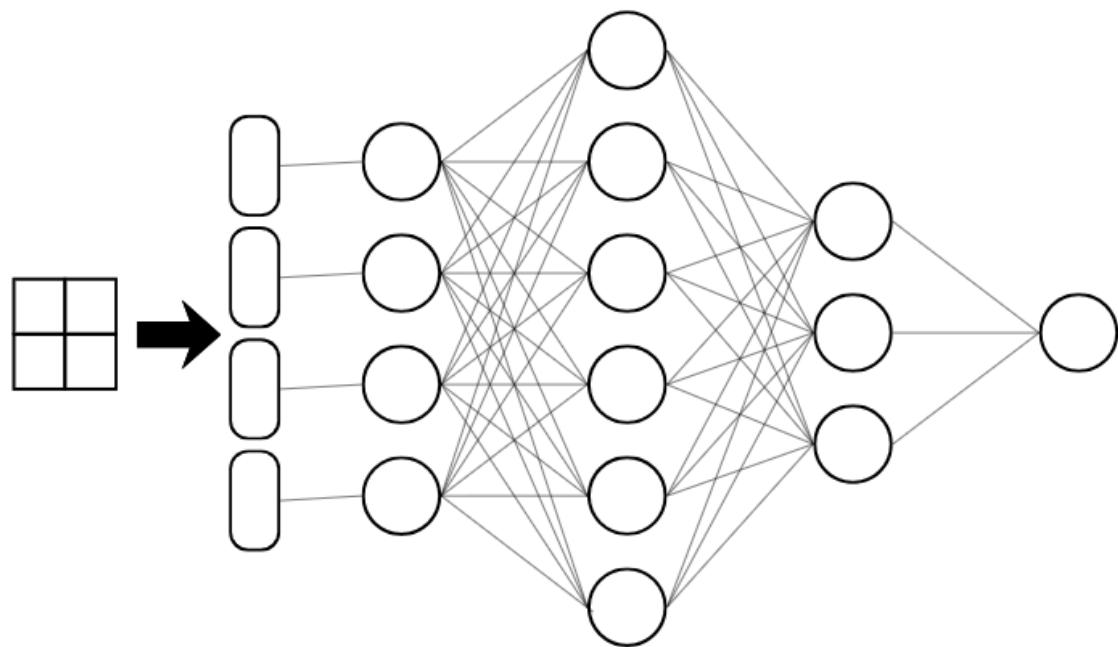
- ① nn.Parallel
- ② nn.LayerStack
- ③ torch.connect
- ④ nn.ModuleList
- ⑤ nn.Sequential

9. 다음 빈칸에 들어갈 코드로 가장 적절한 것은?

```
model = nn.Sequential(  
    nn.Linear(2, 4),  
    nn.ReLU(),  
    nn.Linear(4, 1)  
)  
input_data = torch.tensor([[1.0, 2.0]])  
output = ( _____ )
```

- ① model(input_data)
- ② input_data(model)
- ③ nn.Linear(input_data)
- ④ model.forward_all()
- ⑤ torch.tensor(model)

10. 다음 그림을 나타낸 코드로 가장 적절한 것은?



- ① nn.Sequential(
 nn.Linear(in_features=2, out_features=10),
 nn.Linear(in_features=10, out_features=6),
 nn.Linear(in_features=6, out_features=3)
)
- ② nn.Sequential(
 nn.Flatten(),
 nn.Linear(in_features=4, out_features=6),
 nn.Linear(in_features=6, out_features=3),
 nn.Linear(in_features=3, out_features=1)
)
- ③ nn.Sequential(
 nn.Flatten(start_dim=0, end_dim=-1),
 nn.Linear(in_features=4, out_features=6),
 nn.Linear(in_features=6, out_features=3),
 nn.Linear(in_features=3, out_features=1)
)
- ④ nn.Sequential(
 nn.Flatten(start_dim=0, end_dim=-1),
 nn.Linear(in_features=4, out_features=6),
 nn.Linear(in_features=8, out_features=3),
 nn.Linear(in_features=3, out_features=1)
)

11. 딥러닝 모델이 데이터를 학습하는 전체적인 순서로 옳은 것은 무엇인가요?

- ① 모델 예측 → 손실 계산 → 역전파 → 입력 데이터 → 가중치 업데이트
- ② 가중치 업데이트 → 입력 데이터 → 모델 예측 → 역전파 → 손실 계산
- ③ 입력 데이터 → 역전파 → 가중치 업데이트 → 모델 예측 → 손실 계산
- ④ 입력 데이터 → 모델 예측 → 손실(Loss) 계산 → 역전파(Backpropagation) → 가중치 업데이트
- ⑤ 손실 계산 → 역전파 → 모델 예측 → 가중치 업데이트 → 입력 데이터