

1. 다음 결과를 예측하시오.

```
A = np.array([[1, 2, 3, 4],
              [5, 6, 7, 8],
              [9, 10, 11, 12]])

B = A.reshape(4, -1)
result = B[1:, :3]

print(result)
```

```
[[ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

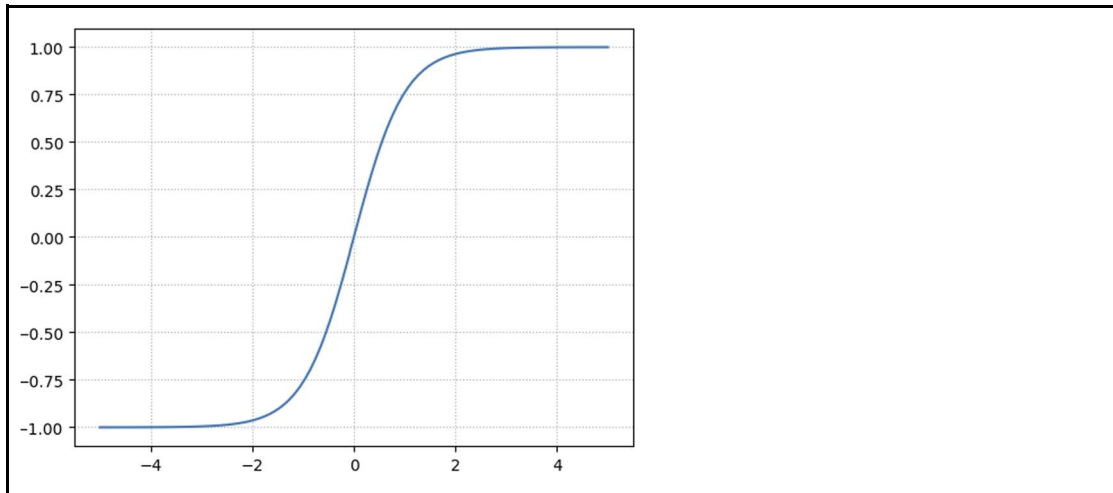
2. 다음 PyTorch 텐서 a 와 b 에 대해 a + b 의 결과는?

```
a = torch.tensor([[1],
                  [2]])

b = torch.tensor([4, 5])
```

- ① tensor([[5], [7]])
- ② tensor([[5, 6],[6, 7]])
- ③ tensor([5, 6, 6, 7])
- ④ tensor([[5, 5],[6, 6]])

3. 아래 그림은 어떤 활성화 함수의 그래프인가?



Tanh <-하이퍼볼릭 탄젠트(Hyperbolic Tangent)

4. 아래 그림은 어떤 활성화 함수의 수식인가?

$$\delta(x) = \begin{cases} x, & \text{if } x \geq 0 \\ ax, & \text{else} \end{cases}$$

Leaky ReLU

5. 다음 수식에서 $\nabla_{\theta} J(\theta^n)$, α 의미는 무엇인가?

$$\theta^{n+1} = \theta^n - \alpha \nabla_{\theta} J(\theta^n)$$

경사하강법

기울기(그래디언트) / 학습률

6. 다음 수식은 머신러닝에서 사용되는 손실 함수이다.
이 함수는 A 이며, 예측 문제 중 B 에서 사용된다. B 에서 출력층의 활성화 함수는 C 가 주로 사용 된다.
A, B, C 는?

$$J(\theta) = - \sum_{i=1}^n \sum_{k=1}^K y_i^{(k)} \log(h_{\theta}^{(k)}(x_i))$$

Cross-Entropy Loss

다중 클래스 분류 문제

Softmax

7. 어떤 분류 모델의 logit 점수가 다음과 같다:
Softmax 함수를 통해 확률로 변환했을 때, 다음 중 올바른 확률 값 조합은?

- 클래스 X: 2
- 클래스 Y: 1
- 클래스 Z: 0

- ① X: 0.500, Y: 0.300, Z: 0.200
- ② X: 0.400, Y: 0.400, Z: 0.200
- ③ X: 0.665, Y: 0.245, Z: 0.090
- ④ X: 0.600, Y: 0.250, Z: 0.150

8. 어떤 이메일 스팸 필터링 시스템의 예측 결과를 정리한 혼동 행렬은 다음과 같다.
이 모델의 다음 성능 지표들을 계산하시오.(스팸 기준)

	예측: 스팸	예측: 정상
실제: 스팸	10	2
실제: 정상	3	5

- a. 정확도 (Accuracy) : 0.75
- b. 재현율 (Recall) : 0.833
- c. 정밀도 (Precision) : 0.769

9. A, B, C 세 가지 클래스를 분류하는 딥러닝 모델의 출력층 결과값이 다음과 같습니다.
이 값을 **Softmax** 함수를 적용하여 확률값으로 변환하세요. (소수점 셋째 자리에서 반올림)

클래스	출력값 (Logits)
A	5
B	3
C	1

A : 0.867
B : 0.117
C : 0.016

10. 다음 중 ReLU 활성화 함수와 Sigmoid 함수에 대한 설명으로 옳은 것을 모두 고르시오.

- ① ReLU 는 음수 입력에 대해 0 을 출력하므로, 그래디언트 소실 문제가 발생할 수 있다.
- ② Sigmoid 는 출력 범위가 (0, 1)이므로 이진 분류 문제의 출력층에 적합하다.
- ③ ReLU 는 선형 함수보다 계산 효율성이 낮다.
- ④ Sigmoid 는 매우 큰 또는 작은 입력값에서 기울기가 거의 0 이 되어 학습이 느려질 수 있다.

11. 다음 결과는?

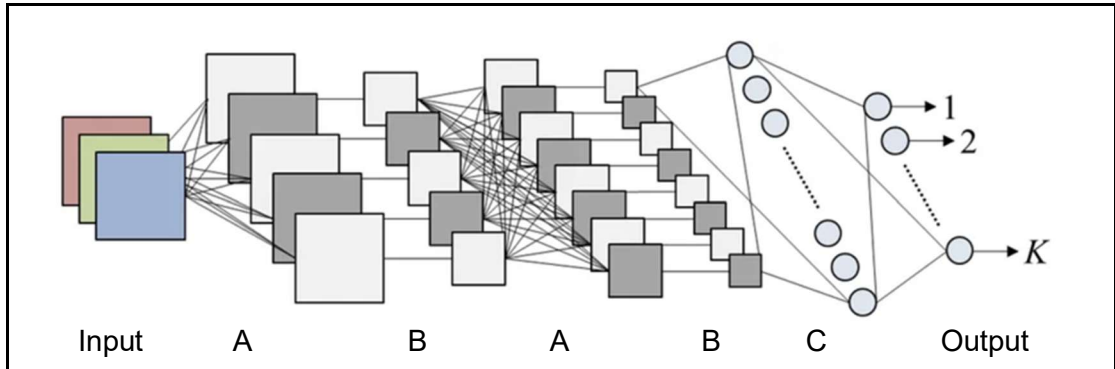
```
input_image = torch.tensor([
    [[0.2, 0.4],
     [0.6, 0.8]]
], dtype=torch.float32)

normalize = transforms.Normalize(mean=0.3, std=0.1)
output_image = normalize(input_image)

print(output_image)
```

tensor([[[[-1.0000, 1.0000],
 [3.0000, 5.0000]]]])

12. 아래 이미지에서 A,B,C 각각 무슨과정인가?



A : max pooling
B : Convolution
C : Fully-Connected

13. 256×256 RGB 영상 1 장을 다음 CNN 네트워크에 입력했을 때, outputs 의 shape 을 계산하시오.

```
conv1 = nn.Conv2d(3, 64, kernel_size=3)
relu = nn.ReLU(inplace=True)
conv2 = nn.Conv2d(64, 128, kernel_size=3)
maxpool = nn.MaxPool2d(kernel_size=2)

features = nn.Sequential(
    conv1,
    relu,
    conv2,
    relu,
    maxpool
)

inputs = torch.randn(1, 3, 256, 256)
outputs = features(inputs)
print(outputs.shape)
```

(1, 64, 254, 254)
(1, 64, 254, 254)
(1, 128, 252, 252)
(1, 128, 252, 252)
(1, 128, 126, 126)

14. 다음 입력 텐서에 Batch Normalization 을 적용한 값은?

```
X = torch.tensor([[4.0, 6.0, 8.0]])  
epsilon = 0.001  
gamma = 1.5  
beta = 0.5
```

1) 평균(μ) 계산 6.0

2) 분산 계산 2.6667

3) 정규화

• $x = 4.0$:

$$\frac{4.0 - 6.0}{\sqrt{2.6667 + 0.001}} \approx \frac{-2.0}{1.633} \approx -1.225$$

• $x = 6.0$:

$$\frac{6.0 - 6.0}{1.633} = 0.0$$

• $x = 8.0$:

$$\frac{8.0 - 6.0}{1.633} \approx 1.225$$

4) 스케일(γ)과 시프트(β) 적용

$BN(x) = \gamma \cdot \text{정규화된 값} + \beta$

$\text{tensor}([-1.337, 0.5, 2.337])$

15. 다음 4×4 입력 행렬에 대해 2×2 크기의 맥스 풀링(Max Pooling)을 스트라이드(stride) 1 로 적용할 경우, 결과로 얻어지는 출력 행렬의 크기와 각 요소의 값을 구하시오.

3	5	2	1
0	4	8	6
2	7	1	9
3	2	5	4

588

789

779

16. 다음 중 ResNet18 과 VGG19 에 대한 설명으로 옳지 않은 것은 무엇인가요?

- ① ResNet18 은 잔차 연결(Residual Connection)을 사용하여 깊은 네트워크에서도 기울기 소실 문제를 완화한다.
- ② VGG19 는 3x3 합성곱 필터만 반복적으로 사용하여 모델의 깊이를 증가시켰다.
- ③ ResNet18 은 Fully Connected Layer 를 사용하지 않아 파라미터 수가 매우 적다.
- ④ VGG19 는 Batch Normalization 을 적용하지 않아 학습 안정성이 ResNet18 보다 상대적으로 낮을 수 있다.

17. 다음 중 YOLOv3 의 Anchor Box 작동 방식에 대한 설명으로 옳지 않은 것은?

- ① YOLOv3 는 k-means 클러스터링으로 데이터셋에 최적화된 9 개의 Anchor 크기를 미리 정의한다
- ② 각 Anchor Box 는 신경망이 예측한 조정값에 따라 위치와 크기가 변화한다
- ③ Anchor Box 는 예측 과정에서 객체 크기에 맞게 자동으로 변형된다.
- ④ 서로 다른 스케일의 예측 레이어(13×13, 26×26, 52×52)마다 다른 크기의 Anchor 가 할당된다
- ⑤ 최종 예측 박스는 Anchor 를 기준으로 조정된 값(b_x, b_y, b_w, b_h)으로 계산된다

18. 다음 중 기울기 소실 문제를 가장 잘 설명한 것은?

- ① 학습률이 점점 커지는 현상
- ② 역전파 시 기울기가 점점 작아져 사라지는 현상
- ③ 가중치가 너무 빨리 갱신되는 현상
- ④ 활성화 함수 값이 항상 1 이 되는 현상

19. 다음 중 RNN 계열 모델이 잘 처리할 수 있는 작업은?

- ① 이미지 분류
- ② 시계열 예측
- ③ 문서 분류(단일 입력)
- ④ 정적 데이터 클러스터링

20. 다음은 LSTM 의 인스턴스를 만드는 코드이다.

lstm 모델에서 학습해야 할 파라미터의 개수는 몇 개인지 구하시오.

```
input_dim = 7  
hidden_size = 6  
lstm = nn.LSTM(input_dim, hidden_size, batch_first=True)
```