AI 개론+응용 정기평가 대비 예상 문제

홍송은

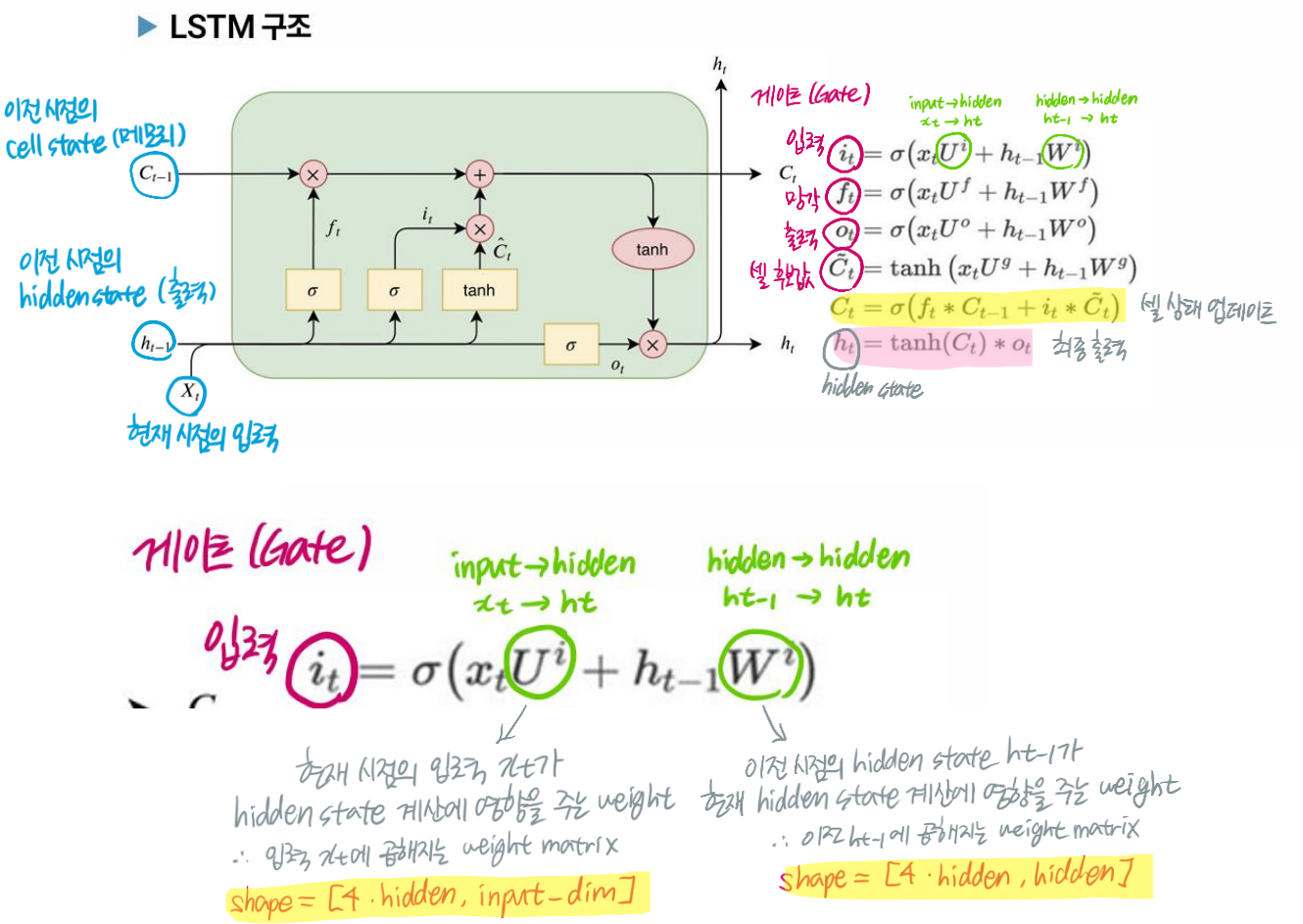
11 주차 스터디 컴퓨터비전 정기평가 대비 예상 문제 작성 및 풀이(개론) 중 문제 풀이 보완

(전효재 - 문제 20 번)

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)
```

(풀이) AI 개론 17 차시 강의자료 – LSTM 구조



총 파라미터 수 = 4 x (input_dim x hidden + hidden x hidden + 2 x hidden) = 360

1. 다음 중 실행 시 오류가 발생할 가능성이 있는 코드 조각을 모두 고르시오.

a.

```
nn.Conv2d(1, 32, kernel_size=5, padding='same')
```

b.

```
nn.Linear(784, 256)
x = torch.randn(32, 784)
x = x.view(784)
```

c.

```
x = torch.randn(1, 28, 28)
x = x.unsqueeze(0)
```

d.

```
x = torch.randn(32, 3, 32, 32)
model = nn.Conv2d(3, 16, kernel_size=3)
y = model(x)
```

2. 아래 CNN 의 출력 텐서 크기는?

```
nn.Conv2d(1, 8, kernel_size=3, stride=1, padding=0)  # 입력: (1, 28, 28)
```

a. (8, 28, 28)

b. (8, 26, 26) # (C, H, W)

c. (1, 28, 28)

d. (8, 24, 24)

e. (1, 26, 26)

3. OpenCV 를 이용하여 엣지를 검출한 후, 윤곽선을 추출하고 중심 좌표를 반환하는 코드를 작성하시오.

```
def get_object_center(img):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray, 100, 200)
    contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL,
                    cv2.CHAIN_APPROX_SIMPLE)
    if len(contours) == 0:
        return None
    largest = max(contours, key=cv2.contourArea)
    M = cv2.moments(largest)
    cx = int(M["m10"] / M["m00"])
    cy = int(M["m01"] / M["m00"])
    return (cx, cy)
```

4. 다음 중 HSV 색공간의 장점으로 적절한 것을 모두 고르시오.
    a. H 는 색상, S 는 채도, V 는 명도를 의미한다.
    b. HSV 는 조명 변화에 민감하다. → 덜 민감하다.
    c. HSV 는 RGB 보다 사람의 색 인식 방식에 가깝게 표현한다.
    d. OpenCV 에서 BGR → HSV 변환은 cv2.cvtColor 로 수행된다.
    e. HSV 는 RGB 보다 빠른 연산을 지원한다. → 연산 속도가 느리다.
    f. HSV 는 색상 기반 객체 추적에 자주 사용된다.

5. 다음 CamShift 를 사용한 추적 알고리즘 구현에서 빈칸을 채워 전체 로직을 완성하시오.

```
def camshift_tracking(frame, rc, hist):
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    backproj = cv2.calcBackProject([hsv], [0], hist, [0, 180], 1)
    term_crit = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 1)
    ret, rc = cv2.CamShift(backproj, rc, term_crit)
    cv2.ellipse(frame, ret, (0, 255, 0), 2)
    return frame
```

6. 다음 중 영상 처리에서 기하학적 변환(Geometric Transformation)에 해당하지 않는 것은?
    a. 이미지 회전
    b. 이미지 확대
    c. 영상 이진화 → 픽셀 기반 연산
    d. 전단(Shearing)
    e. 아핀 변환

7. cv2.filter2D() 함수로 구현 가능한 필터링 기법을 모두 고르시오.
    a. 평균값 필터
    b. 가우시안 블러 → cv2.GaussianBlur()
    c. 샤프닝 필터
    d. Sobel 필터

8. 다음 중 Lucas-Kanade Optical Flow 알고리즘의 특징으로 옳은 것들은?
    a. 모든 픽셀에 대해 흐름을 계산한다 → Dense Optical Flow
    b. 특징점 기반 추적 방식이다
    c. 작은 움직임 추적에 강하다
    d. 연산량이 크고 느리다
    e. 다중 해상도(Pyramid) 기반 개선이 가능하다

9. Depthwise Separable Convolution 의 장점으로 올바른 설명은?
    a. 채널 별 연산 분리를 통해 연산량 감소
    b. 3×3 필터만 사용 가능함
    c. 파라미터 수 감소
    d. Pointwise convolution 은 채널 간 정보를 통합
    e. 연산량은 일반 Conv 보다 약 8~9 배 적다
    f. 항상 정확도가 향상된다

10. 다음 중 Word2Vec 의 Skip-gram 방식 설명으로 올바른 것은?
    a. 전체 문서의 단어 분포를 벡터로 표현한다
    b. 주변 단어를 이용해 중심 단어를 예측한다
    c. 중심 단어로부터 주변 단어를 예측한다
    d. TF 와 IDF 를 곱하여 중요도를 반영한다
    e. 문장의 길이를 기준으로 벡터를 설정한다

11. TF-IDF 를 기반으로 텍스트 데이터를 벡터화하는 코드를 완성하시오.

```
from sklearn.feature_extraction.text import _____

text_data = ["나는 사과를 좋아해", "나는 바나나를 좋아해"]
vectorizer = TfidfVectorizer()
vectorizer. fit(text_data)
X = vectorizer. transform(text_data).toarray()
```

12. 다음 중 TF-IDF 에서 고려하는 정보로 올바른 항목을 모두 고르시오.
    a. 단어의 빈도
    b. 단어의 길이
    c. 단어가 나타나는 문서 수
    d. 전체 문서 수
    e. 단어의 철자 수

13. 다음 중 Neural Style Transfer(NST)에 사용되는 구성 요소를 모두 고르시오.

a. GAN 기반 생성자
b. Content loss
c. Style loss
d. Gram matrix
e. VGG16 feature extractor
f. TfidfVectorizer

14. 다음 Gram Matrix 계산 함수에 대한 설명으로 옳은 것을 모두 고르시오.

```
def gram_matrix(ip):
    B, C, H, W = ip.size()
    feats = ip.view(B * C, H * W)
    gram = torch.mm(feats, feats.t())
    return gram / (B * C * H * W)
```

a. Gram Matrix 는 채널 간 상관관계를 나타낸다
b. Feature map 의 Flatten 결과를 바탕으로 계산된다
c. 스타일 손실 계산에 사용되는 주요 행렬이다
d. 연산 결과는 항상 정방행렬이다
e. 이미지의 색상만 반영된다 → 공간/채널 간 연관 모두 포함한다.

15. CAM 시각화 기법이 정상적으로 작동하기 위한 모델 구조 요건은?

a. Dropout Layer 후 Fully Connected Layer
b. 마지막 Convolution + Global Average Pooling + FC
c. FC Layer 만 있는 구조
d. 입력 이미지의 크기 제한 없음
e. BatchNorm 이 반드시 포함되어야 함

16. 다음 중 3D Rendering 에서 NeRF 의 입력 및 출력 요소로 옳은 것을 고르시오.

a. (x, y, z) 위치
b. 시선 방향 (θ, φ)
c. RGB 색상
d. 텍스처 좌표
e. 밀도 σ

17. Gaussian Splatting 의 렌더링 품질과 효율을 조절하는 핵심 기법은?

a. Noise Regularization
b. Adaptive Density Control
   → 각 Gaussian 의 밀도와 크기를 조정해 표현 품질을 유동적으로 제어
c. Positional Encoding
d. GAN Loss Adjustment
e. Image Pyramid Sampling

18. DQN 의 안정성을 높이기 위한 기법으로 적절한 것은?

a. Experience Replay → 데이터 decorrelation
b. ε-soft Policy → 탐험/이용 균형
c. Target Network → 안정적 Q 업데이트
d. One-hot Encoding
e. Dynamic Threshold Scaling

19. Experience Replay 에서 사용하는 주요 기능은?

a. 상태 전이 확률 계산
b. Q-table 직접 업데이트
c. 메모리 버퍼 샘플링
d. 연관성 낮은 샘플로 학습
e. 예측과 타깃 네트워크 일치

20. 아래 코드의 출력을 예측하시오.

```
gamma = 0.9
rewards = [1, 0, 0, 2]
G = 0
for r in reversed(rewards):
    G = r + gamma * G
print(round(G, 2))
```

a. 2.81

b. 3.61

c. 2.46

d. 1.90

e. 4.39

$G_3 = 2$

$G_2 = 0 + 0.9 \times 2 = 1.8$

$G_1 = 0 + 0.9 \times 1.8 = 1.62$

$G_0 = 1 + 0.9 \times 1.62 = 2.458 \approx 2.46$

21. 다음은 Q-learning 의 핵심 업데이트 식이다. 다음 조건을 만족하는 update_q 함수를 구현하시오.

- Q 는 상태-행동 값 테이블 (numpy 2D array)
- 입력: 상태, 행동, 보상, 다음 상태, 학습률, 감가율
- 출력: Q[state, action]을 업데이트

```
def update_q(Q, state, action, reward, next_state, alpha, gamma):
    max_next_q = np.max(Q[next_state])
    Q[state, action] += alpha * (reward + gamma * max_next_q - Q[state, action])
```