AIO2024 - AI VIETNAM

# Bài thi đánh giá cuối module 6

Soạn nội dung: Trường Bình, Nguyễn Khôi, Đăng Nhã, Nguyễn Thịnh
Soạn Latex: Mai Hạnh
Tư vấn: Quang Vinh

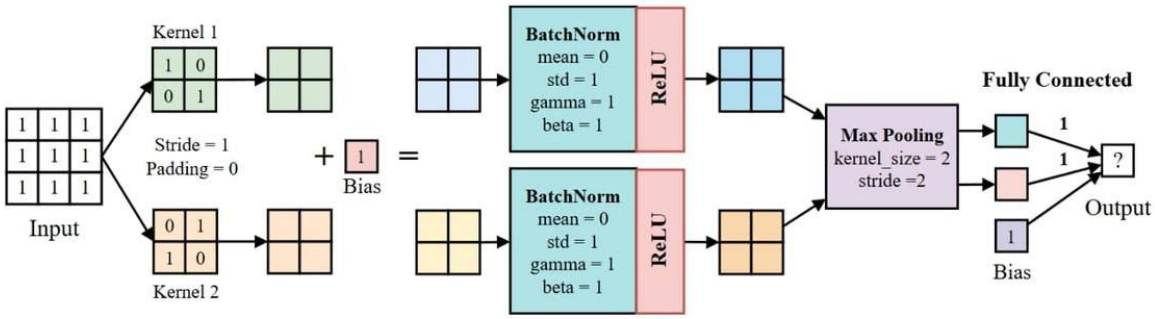
Ngày 05 tháng 01 năm 2025

## 1 Phần I: CNN

### Mô tả bài toán

1. Dữ liệu đầu vào: một ma trận
2. Mạng nơ ron có kiến trúc lần lượt gồm các lớp:
    - Convolution
    - Batch Normalization
    - ReLU
    - MaxPooling
    - Fully Connected

Với dữ liệu đầu vào và thông tin chi tiết về hình 1 dưới đây:




Hình 1: Mô hình CNN áp dụng từ câu 1 tới câu 5

## 1.1 Câu 1

**Câu hỏi:** Dựa vào mô tả trong hình 1, hãy tính toán đầu ra của mạng sau lớp Convolution (làm tròn đến 1 chữ số thập phân).

(A) $\begin{bmatrix} 3.0 & 3.0 \\ 3.0 & 3.0 \end{bmatrix} \quad \begin{bmatrix} 3.0 & 3.0 \\ 3.0 & 3.0 \end{bmatrix}$

(B) $\begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix} \quad \begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix}$

(C) $\begin{bmatrix} 3.0 & 3.0 \\ 3.0 & 3.0 \end{bmatrix} \quad \begin{bmatrix} 6.0 & 3.0 \\ 3.0 & 6.0 \end{bmatrix}$

(D) $\begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix} \quad \begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 3.0 \end{bmatrix}$

## 1.2 Câu 2

**Câu hỏi:** Với dữ liệu đầu vào và thông tin chi tiết về mạng được mô tả trong hình 1, hãy tính toán đầu ra của mạng sau lớp Batch Normalization và ReLU (làm tròn đến 1 chữ số thập phân).

(A) $\begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix} \quad \begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix}$

(B) $\begin{bmatrix} 4.0 & 4.0 \\ 4.0 & 4.0 \end{bmatrix} \quad \begin{bmatrix} 4.0 & 4.0 \\ 4.0 & 4.0 \end{bmatrix}$

(C) $\begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} \quad \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$

(D) $\begin{bmatrix} 3.0 & 3.0 \\ 3.0 & 3.0 \end{bmatrix} \quad \begin{bmatrix} 3.0 & 3.0 \\ 3.0 & 3.0 \end{bmatrix}$

## 1.3 Câu 3

**Câu hỏi:** Với dữ liệu đầu vào và thông tin chi tiết về mạng được mô tả trong hình 1, hãy tính toán đầu ra của mạng sau lớp MaxPooling (làm tròn đến 1 chữ số thập phân).

(A) $\begin{bmatrix} 2.0 \end{bmatrix} \quad \begin{bmatrix} 2.0 \end{bmatrix}$

(B) $\begin{bmatrix} 1.5 \end{bmatrix} \quad \begin{bmatrix} 1.5 \end{bmatrix}$

(C) $\begin{bmatrix} 3.0 \end{bmatrix} \quad \begin{bmatrix} 3.0 \end{bmatrix}$

(D) $\begin{bmatrix} 4.0 \end{bmatrix} \quad \begin{bmatrix} 4.0 \end{bmatrix}$

## 1.4 Câu 4

**Câu hỏi:** Với dữ liệu đầu vào và thông tin chi tiết về mạng được mô tả trong hình 1, hãy tính toán đầu ra cuối cùng sau lớp Fully Connected (làm tròn đến 1 chữ số thập phân).

(A) $\begin{bmatrix} 7.0 \end{bmatrix}$

(B) $\begin{bmatrix} 8.0 \end{bmatrix}$

(C) $\begin{bmatrix} 9.0 \end{bmatrix}$

(D) $\begin{bmatrix} 10.0 \end{bmatrix}$

## 1.5 Câu 5

**Câu hỏi:** Cho ma trận đầu vào có kích thước $10 \times 10$, áp dụng phép tích chập với kernel $5 \times 5$, stride = 2, padding = 1. Tính kích thước của ma trận đầu ra.

(A) $3 \times 3$

(B) $4 \times 4$

(C) $5 \times 5$

(D) $6 \times 6$

## 1.6 Câu 6

**Câu hỏi:** Cho trước lớp convolution của nhánh skip connection có padding = 0, hãy tính toán số lượng kernel, kernel size và stride thật phù hợp cho lớp convolution này.

(A) Số lượng kernel: 3, kernel size: 7x7, stride: 1

(B) Số lượng kernel: 32, kernel size: 3x3, stride: 1

(C) Số lượng kernel: 32, kernel size: 5x5, stride: 2

(D) Số lượng kernel: 3, kernel size: 7x7, stride: 2

## 1.7 Câu 7

**Câu hỏi:** Kích thước tensor đầu ra cuối cùng của mạng là bao nhiêu?

(A) ‘[1, 3, 64, 64]‘

(B) ‘[1, 32, 32, 32]‘

(C) ‘[1, 3, 32, 32]‘

(D) ‘[1, 3, 29, 29]‘

## 1.8 Câu 8

**Câu hỏi:** Cơ chế Skip Connection lần đầu tiên được giới thiệu trong kiến trúc mạng nào?

(A) VGGNet

(B) AlexNet

(C) ResNet

(D) MobileNet

## 1.9 Câu 9

**Câu hỏi:** Kiến trúc GoogleNet giới thiệu khái niệm "Inception Module". Điểm khác biệt chính của Inception Module so với các kiến trúc mạng trước đó là gì?

(A) Thay thế lớp pooling bằng các lớp convolution $1 \times 1$.

(B) Tích hợp nhiều kích thước kernel khác nhau trong cùng một module.

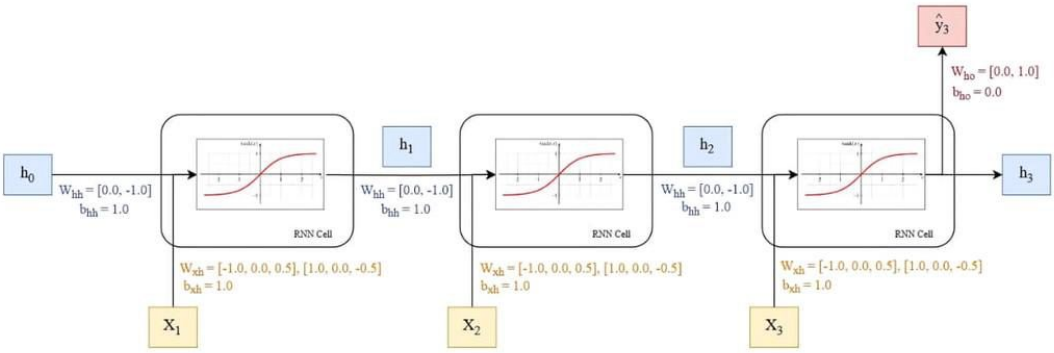
(C) Giảm số lượng lớp fully connected.

(D) Sử dụng depthwise separable convolution.

# 2 Phần II: RNN

## 2.1 Câu 10

Với dữ liệu đầu vào và thông tin chi tiết về hình 2 dưới đây:




Hình 2: Mô hình RNN áp dụng từ câu 10 và câu 11

**Câu hỏi:** Giả sử ta có một mạng RNN đơn giản với **input size = 3** và **hidden size = 2**. Trọng số của mạng chi tiết như hình 2.
Đầu vào của mạng là chuỗi gồm 3 timestep $(x_1, x_2, x_3)$, mỗi timestep có **input size = 3**:

$$x_1 = [1.0, 2.0, 3.0], \quad x_2 = [4.0, 5.0, 6.0], \quad x_3 = [7.0, 8.0, 9.0]$$

Hãy tính toán output của hidden state tại mỗi timestep $(h_1, h_2, h_3)$.

(A) $[0.9866, 0.9051], [0.0946, 0.9702], [-0.8996, 0.9983]$

(B) $[0.9866, 0.9051], [0.0946, 0.9702], [0.9114, 0.4899]$

(C) $[0.9866, 0.9051], [0.0946, 0.9720], [0.9999, 1.0000]$

(D) $[-0.4621, 0.9998], [2.4676 \times 10^{-4}, 0.9640], [0.9114, 0.4899]$

## 2.2 Câu 11

Tiếp tục với mạng RNN ở câu 10, ta thêm một lớp fully connected với **output size = 1** để thực hiện bài toán regression. Trọng số của lớp fully connected $W_{ho}$ và $b_{ho}$ được khởi tạo như hình.
**Câu hỏi:** Hãy tính toán output của mạng RNN sau khi đi qua lớp fully connected $\hat{y}_3$.

(A) $[0.4899]$

(B) $[0.9983]$

(C) $[1]$

(D) $[0.9051]$

### 2.3 Câu 12

Giả sử ta muốn dùng mạng RNN ở câu 11 để thực hiện bài toán sequence labeling. Ta thêm vào mỗi timestep một lớp Fully Connected để chuyển đổi giá trị hidden state thành giá trị dự đoán output $\hat{y}_i$.
**Câu hỏi:** Ba timestep $(x_1, x_2, x_3)$ lần lượt được mạng RNN trên dự đoán nhãn là gì?

(A) 1, 0, 1

(B) 1, 1, 0

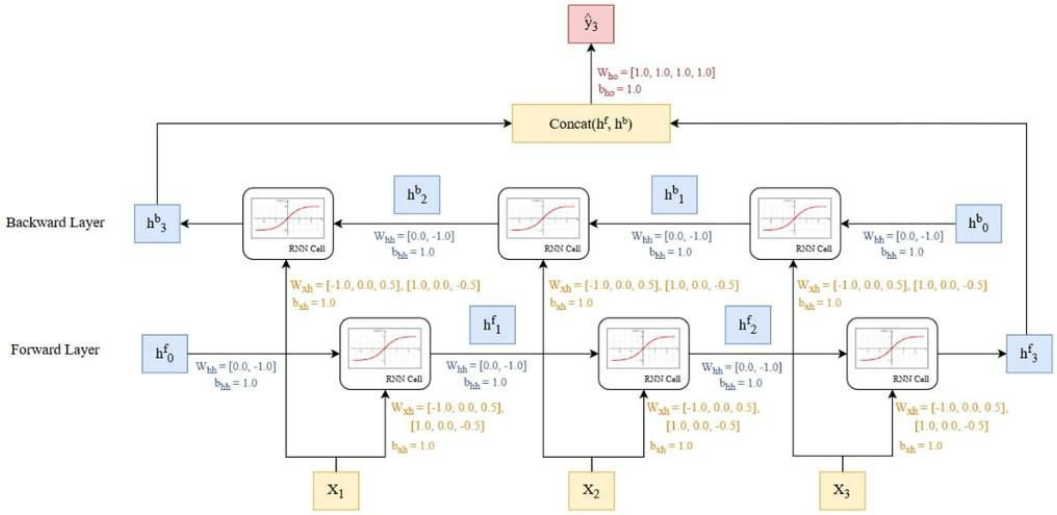
(C) 0, 0, 1

(D) 1, 0, 0

## 3 Phần III: LSTM

### 3.1 Câu 13

Ta có một mạng Bidirectional RNN với **input size = 3** và **hidden size = 2**. Trọng số của mạng được khởi tạo giống như trên hình cho cả chiều từ trái sang phải và chiều từ phải sang trái.
Với dữ liệu đầu vào và thông tin chi tiết về hình 3 dưới đây:




Hình 3: Mô hình LSTM áp dụng câu 13

Đầu vào của mạng là chuỗi 3 timesteps, mỗi timestep có **input size = 3** và có giá trị lần lượt là:

- $x_1 = [1.0, 2.0, 3.0]$
- $x_2 = [4.0, 5.0, 6.0]$

- $x_3 = [7.0, 8.0, 9.0]$

Đầu ra của mạng ở hai chiều sẽ được nối với nhau với thứ tự **(hidden state của mạng từ trái sang phải, hidden state của mạng từ phải sang trái)**.
**Câu hỏi:** Hãy tính **output cuối cùng** của mạng (không sử dụng softmax). Cho trước giá trị đầu ra cuối cùng của mạng từ phải sang trái là $[0.9114, 0.4899]$.

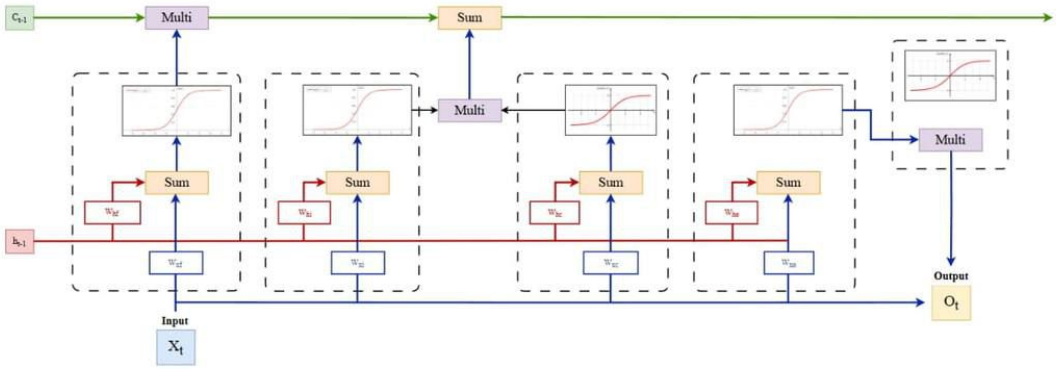
(A) 2.5

(B) 4.5

(C) 1.0

(D) 3

Với dữ liệu đầu vào và thông tin chi tiết về hình 4 dưới đây:




Hình 4: Mô hình LSTM áp dụng từ câu 14 tới câu 17

## 3.2 Câu 14

Giả sử ta có một mạng LSTM với input size = 3 và hidden size = 2. Cell state ban đầu: $C_{t-1} = [0.0, 0.0]$. Trọng số của mạng được khởi tạo như sau. Lưu ý rằng mô hình không có bias:

$$W_{xf} = \begin{bmatrix} 1.0 & 0.0 & 0.5 \\ 1.0 & 0.0 & -0.5 \end{bmatrix}, \quad W_{hf} = \begin{bmatrix} 0.0 & 1.0 \\ 0.0 & -1.0 \end{bmatrix}$$

Đầu vào của mạng là 1 timestep:

$$X = [1.0, 2.0, 3.0]$$

**Câu hỏi:** Hãy tính output của forget gate.

(A) $[2.5000, -0.5000]$

(B) $[0.5000, -2.5000]$

(C) $[-0.5000, -0.5000]$

(D) $[1.5000, -1.5000]$

## 3.3 Câu 15

Tiếp tục với mạng ở câu trên. Lưu ý rằng mô hình không có bias:

$$W_{xc} = \begin{bmatrix} 1.0 & 0.0 & -0.5 \\ 1.0 & 0.0 & -0.5 \end{bmatrix}, \quad W_{hc} = \begin{bmatrix} 0.0 & 1.0 \\ 0.0 & -1.0 \end{bmatrix}$$

Đầu vào của mạng là 1 timestep:

$$X = [1.0, 2.0, 3.0]$$

**Câu hỏi:** Hãy tính output của cell gate.

(A) $[2.5000, -0.5000]$

(B) $[0.5000, -2.5000]$

(C) $[-0.5000, -0.5000]$

(D) $[1.5000, -1.5000]$

## 3.4 Câu 16

Tiếp tục với mạng ở câu trên. Lưu ý rằng mô hình không có bias:

$$W_{xc} = \begin{bmatrix} 1.0 & 0.0 & -0.5 \\ 1.0 & 0.0 & -0.5 \end{bmatrix}, \quad W_{hc} = \begin{bmatrix} 0.0 & 1.0 \\ 0.0 & -1.0 \end{bmatrix}$$

Đầu vào của mạng là 1 timestep:

$$X = [1.0, 2.0, 3.0]$$

**Câu hỏi:** Hãy tính output của cell gate.

(A) $[2.5000, -0.5000]$

(B) $[0.5000, -2.5000]$

(C) $[-0.5000, -0.5000]$

(D) $[1.5000, -1.5000]$

## 3.5 Câu 17

Tiếp tục với mạng ở câu trên. Lưu ý rằng mô hình không có bias:

$$W_{xo} = \begin{bmatrix} 1.0 & 1.0 & 0.5 \\ 1.0 & -1.0 & -0.5 \end{bmatrix}, \quad W_{ho} = \begin{bmatrix} 0.0 & 1.0 \\ 0.0 & -1.0 \end{bmatrix}$$

Đầu vào của mạng là 1 timestep:

$$X = [1.0, 2.0, 3.0]$$

**Câu hỏi:** Hãy tính output cuối cùng của mô hình.

(A) $[-0.2769, -0.0027]$

(B) $[0.2769, -0.0027]$

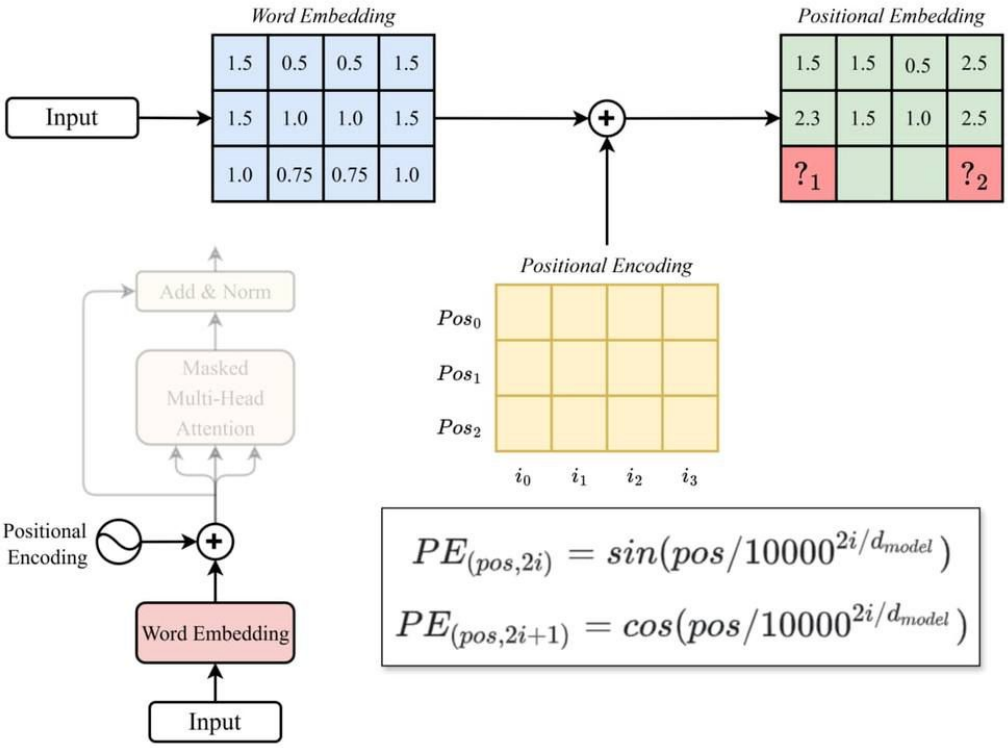
(C) $[-0.2769, 0.0027]$

(D) $[0.2769, 0.0027]$

# 4 Phần IV: Transformer

## 4.1 Câu 18

Với dữ liệu đầu vào và thông tin chi tiết về hình 5 dưới đây:




Hình 5: Hình ảnh minh hoạ mô hình Transformer cho câu 18

**Câu hỏi:** Các giá trị thiếu (các ô ký hiệu là "?") trong hình lần lượt ("?$_1$", "?$_2$") là:

- ***Chỉ kết quả cuối cùng*** được làm tròn đến chữ số thập phân **thứ nhất**.
- *Những* ***ô đỏ không có ký hiệu*** ? là những câu đã tính ở câu trước.
- *Những giá trị trên hình đã đủ để tính mà* ***không phụ thuộc vào câu trước***.

(A) 1.9, 2.0

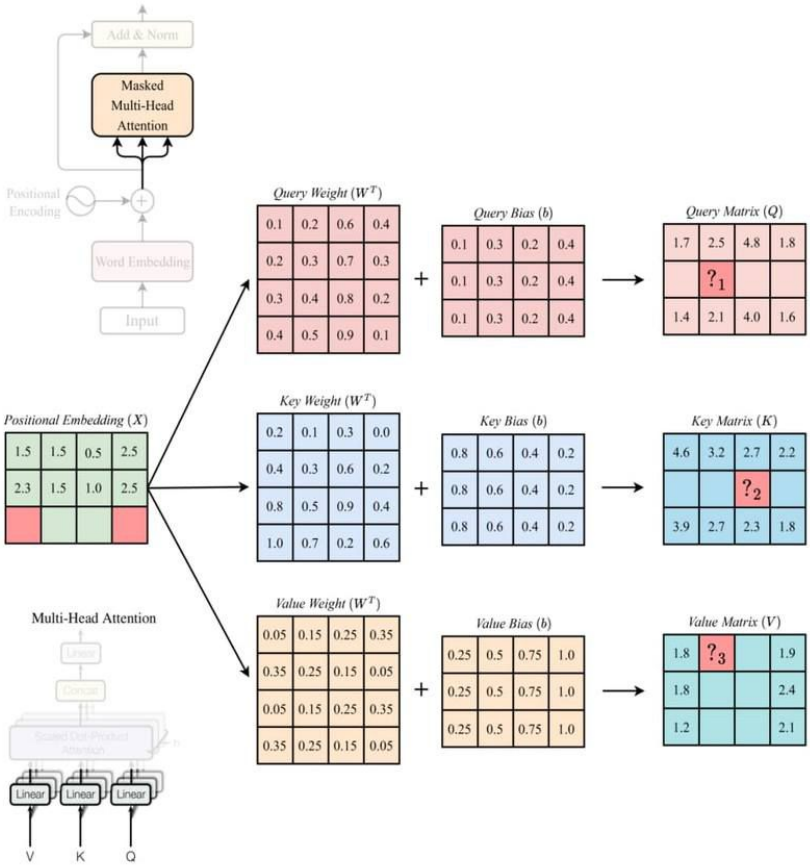
(B) 1.9, 0.8

(C) 0.8, 2.0

(D) 0.3, 0.8

## 4.2 Câu 19

Với dữ liệu đầu vào và thông tin chi tiết về hình 6 dưới đây:




Hình 6: Hình ảnh minh hoạ mô hình Transformer cho câu 19

**Câu hỏi:** Các giá trị thiếu (các ô ký hiệu là "?") trong hình lần lượt ("?$_1$", "?$_2$", "?$_3$") là:

- ***Chỉ kết quả cuối cùng*** được làm tròn đến chữ số thập phân **thứ nhất**.
- *Những* ***ô đỏ không có ký hiệu*** ? là những câu đã tính ở câu trước.
- *Những giá trị trên hình đã đủ để tính mà* ***không phụ thuộc vào câu trước***.

(A) 3.4, 2.9, 1.8

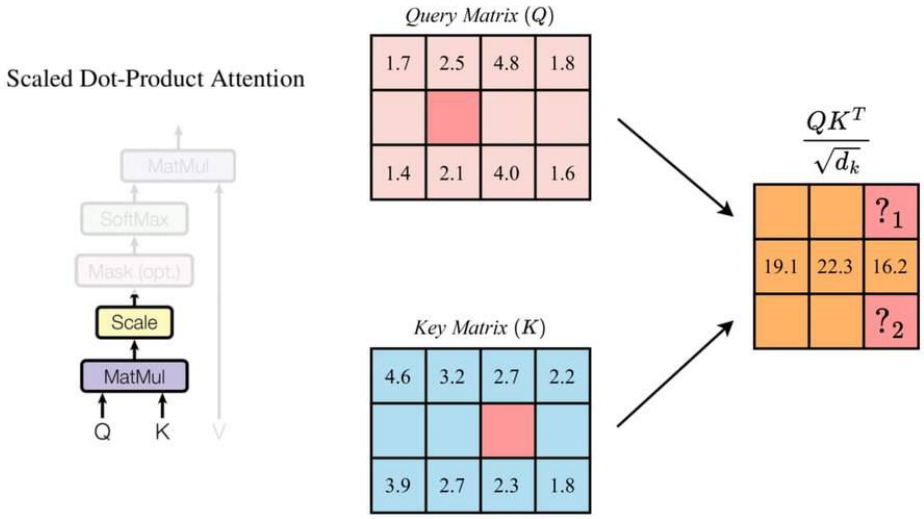
(B) 2.9, 1.8, 3.4

(C) 2.9, 3.4, 1.8

(D) 1.8, 3.4, 2.9

### 4.3 Câu 20

Với dữ liệu đầu vào và thông tin chi tiết về hình 7 dưới đây:




Hình 7: Hình ảnh minh hoạ mô hình Transformer cho câu 20

**Câu hỏi:** Tính các giá trị thiếu (các giá trị "?") trong hình lần lượt ("$?_1$", "$?_2$") là:

- ***Chỉ kết quả cuối cùng*** được làm tròn đến chữ số thập phân **thứ nhất**.
- *Những* ***ô đỏ không có ký hiệu*** ? là những câu đã tính ở câu trước.
- *Những giá trị trên hình đã đủ để tính mà* ***không phụ thuộc vào câu trước***.

(A) 16.4, 13.7

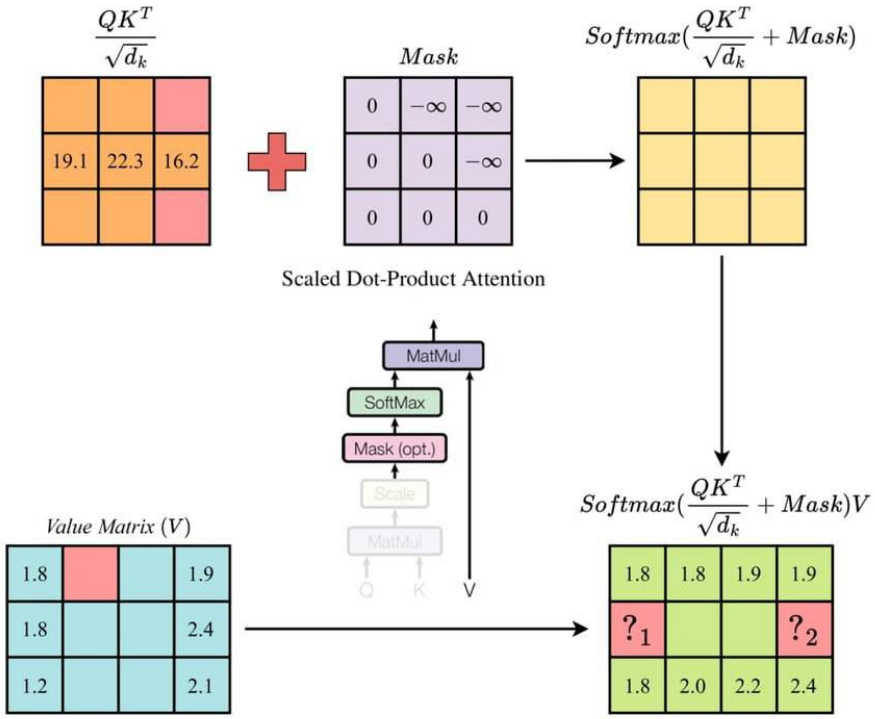
(B) 13.8, 13.7

(C) 16.4, 11.6

(D) 13.8, 11.6

## 4.4 Câu 21

Với dữ liệu đầu vào và thông tin chi tiết về hình 8 dưới đây:




Hình 8: Hình ảnh minh hoạ mô hình Transformer cho câu 21

**Câu hỏi:** Sử dụng kết quả từ câu trước, tính các giá trị thiếu (các giá trị "?") trong hình lần lượt ("?$_1$", "?$_2$") là:

- ***Chỉ kết quả cuối cùng*** được làm tròn đến chữ số thập phân **thứ nhất**.
- *Những* ***ô đỏ không có ký hiệu*** ? là những câu đã tính ở câu trước.
- *Những giá trị trên hình đã đủ để tính mà* ***không phụ thuộc vào câu trước***.

(A) 1.8, 1.8

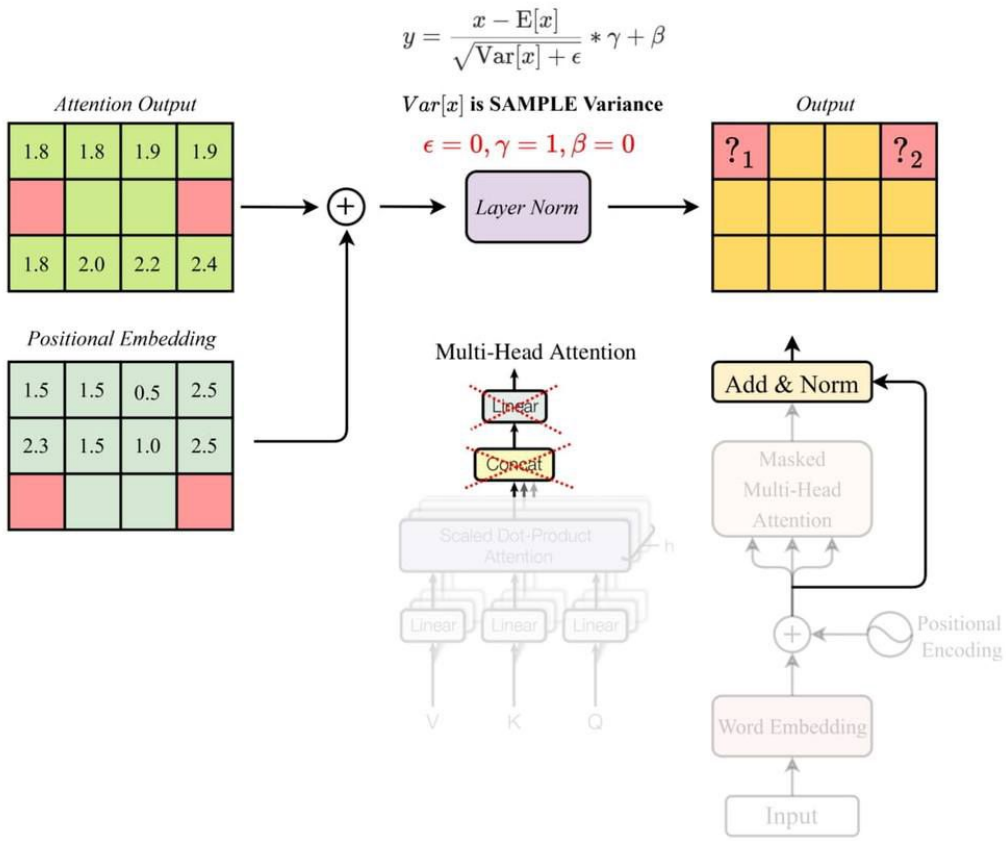
(B) 2.0, 2.4

(C) 1.6, 1.8

(D) 1.8, 2.4

## 4.5 Câu 22

Với dữ liệu đầu vào và thông tin chi tiết về hình 9 dưới đây:




Hình 9: Hình ảnh minh hoạ mô hình Transformer cho câu 22

**Câu hỏi:** Sử dụng kết quả từ câu trước, tính các giá trị thiếu (các giá trị "?") trong hình lần lượt ("?$_1$", "?$_2$") là:

- ***Chỉ kết quả cuối cùng*** được làm tròn đến chữ số thập phân **thứ hai**.
- *Những* ***ô đỏ không có ký hiệu*** ? là những câu đã tính ở câu trước.
- *Những giá trị trên hình đã đủ để tính mà* ***không phụ thuộc vào câu trước***.

(A) $-0.06, 1.28$

(B) $-0.06, -1.16$

(C) $0.06, 1.28$

(D) $0.06, -1.16$

## 4.6 Câu 23

**Câu hỏi:** Transformer MultiHead kết nối đầu ra của các head như nào?

(A) Cộng các đầu ra của các Head lại với nhau.

(B) Nhân các đầu ra của các Head với nhau.

(C) Nối các đầu ra của các Head theo chiều dọc (xếp chồng lên nhau).

(D) Nối các đầu ra theo chiều ngang (nối tiếp vào nhau).

## 4.7 Câu 24

**Câu hỏi:** Trong cơ chế Self-Attention của Transformer, phát biểu nào sau đây là đúng?

(A) Số cột của ma trận $Q$, $K$, $V$ bắt buộc phải bằng nhau.

(B) Số cột của ma trận $Q$, $K$ bắt buộc phải bằng nhau, nhưng $V$ có thể khác.

(C) Số cột của ma trận $Q$, $V$ bắt buộc phải bằng nhau, nhưng $K$ có thể khác.

(D) Số cột của ma trận $K$, $V$ bắt buộc phải bằng nhau, nhưng $Q$ có thể khác.

## 4.8 Câu 25

**Câu hỏi:** Trong cơ chế Cross-Attention của Transformer, Query $Q$, Key $K$ và Value $V$ có nguồn gốc từ đâu?

(A) $Q$ từ đầu ra của Masked Multi-Head Attention trong Decoder. $K$ và $V$ từ đầu ra của Encoder.

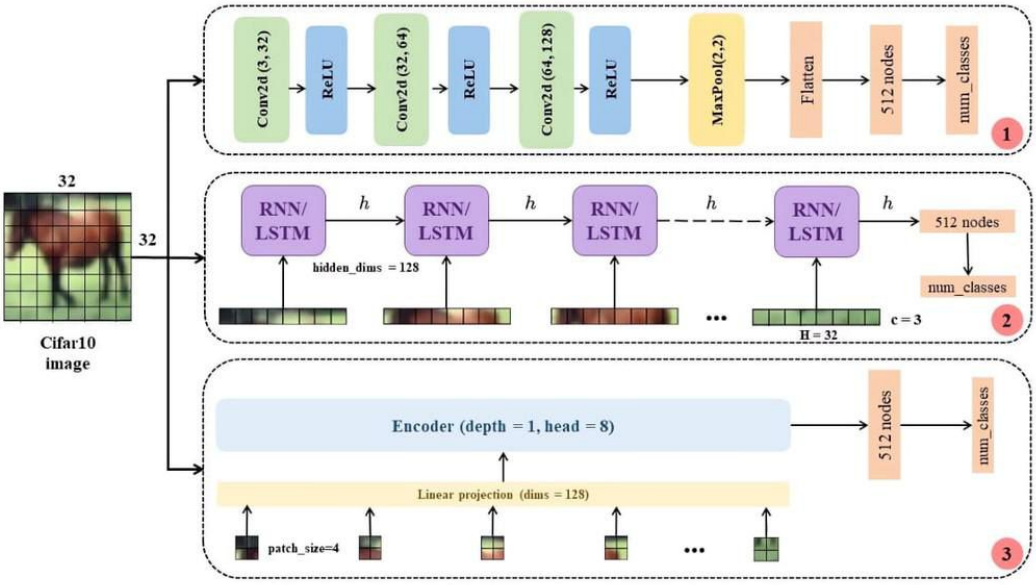
(B) $K$ và $V$ từ đầu ra của Masked Multi-Head Attention trong Decoder. $Q$ từ đầu ra của Encoder.

(C) $V$ từ đầu ra của Masked Multi-Head Attention trong Decoder. $Q$ và $K$ từ đầu ra của Encoder.

(D) $Q$ và $V$ từ đầu ra của Masked Multi-Head Attention trong Decoder. $K$ từ đầu ra của Encoder.

# 5 Phần V: Models Implementation

Với thông tin chi tiết về hình 10 dưới đây:




Hình 10: Hình ảnh minh hoạ áp dụng các mô hình từ câu 26 đến câu 30

## Mô tả bài toán

Bài toán phân loại ảnh dựa trên bộ dữ liệu CIFAR-10 là một bài toán kinh điển trong lĩnh vực Computer Vision. CIFAR-10 bao gồm 60.000 ảnh màu với kích thước $32 \times 32$ pixel, được chia thành 10 lớp, mỗi lớp có 6.000 ảnh.

Chúng ta sẽ thực hiện chia bộ Dataset thành tập train (50.000 ảnh) và tập test (10.000 ảnh). Đồng thời, các hyperparameters và pipeline đã được cài đặt sẵn trong file colab sau: link.

Nhiệm vụ của chúng ta là tải hoặc copy file colab, triển khai các mô hình khác nhau (CNN, RNN, LSTM, ViT) với các tham số được mô tả chi tiết trong hình dưới đây để huấn luyện cho tác vụ phân loại ảnh (lưu ý sử dụng GPU cho quá trình huấn luyện).

Từ đó đưa ra đánh giá, so sánh tác dụng của từng loại mô hình cho dữ liệu ảnh.

### 5.1 Câu 26

Thực hiện triển khai mô hình CNN bằng Pytorch theo module (1) trong hình, với các tham số mặc định `padding=1`, `kernel_size=3`, `stride=1` và trả lời câu hỏi dưới đây:

**Câu hỏi:** Shape của lớp flatten bằng bao nhiêu?

(A) $128 \times 8$

(B) $128 \times 8 \times 8$

(C) $128 \times 16$

(D) $128 \times 16 \times 16$

## 5.2 Câu 27

Thực hiện triển khai mô hình RNN bằng Pytorch với các tham số được mô tả ở module (2) trong hình (lưu ý `batch_first=True`, `num_layers=1`) và trả lời câu hỏi dưới đây:
**Câu hỏi:** Shape của input X khi đưa vào mô hình RNN là bao nhiêu?

(A) `(64, 32, 32 * 128)`

(B) `(64, 128, 32 * 3)`

(C) `(64 * 32, 32, 128)`

(D) `(64, 32, 32 * 3)`

## 5.3 Câu 28

Thực hiện triển khai mô hình LSTM bằng Pytorch với các tham số được mô tả ở module (2) trong hình (lưu ý `batch_first=True`, `num_layers=1`) và trả lời câu hỏi dưới đây:
**Câu hỏi:** Output của mô hình LSTM gồm (`out, (h_n, c_n)`), đâu là shape đúng của `out`?

(A) `(batch_size, seq_len, hidden_dims)`

(B) `(num_layers, batch_size, hidden_dims)`

(C) `(batch_size, hidden_dims, seq_len)`

(D) `(batch_size, num_layers, hidden_dims)`

## 5.4 Câu 29

Thực hiện triển khai mô hình ViT với các tham số được mô tả ở module (3) - (những phần không vẽ của ViT như `[cls]`, `PoS` khai báo bình thường) trong hình và trả lời câu hỏi dưới đây:
**Câu hỏi:** Tổng cộng có bao nhiêu `patches`?

(A) 8

(B) 16

(C) 32

(D) 64

## 5.5 Câu 30

**Câu hỏi:** Sau khi đã triển khai toàn bộ 4 mô hình theo 3 modules và các tham số khai báo trong baseline, đâu là thứ tự đúng khi so sánh hiệu suất của các mô hình dựa trên tập test (cao nhất → thấp nhất)? (Nếu sự chênh lệch chưa tới 5% thì được xem như tương đương ˜)

(A) Transformer (ViT) → CNN → RNN ∼ LSTM

(B) CNN → LSTM ∼ Transformer → RNN

(C) CNN → RNN ∼ LSTM → Transformer

(D) CNN → Transformer → RNN ∼ LSTM

# 6 Đáp án

**Câu 1:** A.

$$\begin{bmatrix} 3.0 & 3.0 \\ 3.0 & 3.0 \end{bmatrix} \quad \begin{bmatrix} 3.0 & 3.0 \\ 3.0 & 3.0 \end{bmatrix}$$

**Câu 2:** B.

$$\begin{bmatrix} 4.0 & 4.0 \\ 4.0 & 4.0 \end{bmatrix} \quad \begin{bmatrix} 4.0 & 4.0 \\ 4.0 & 4.0 \end{bmatrix}$$

**Câu 3:** D.

$$\begin{bmatrix} 4.0 \end{bmatrix} \quad \begin{bmatrix} 4.0 \end{bmatrix}$$

**Câu 4:** C.

$$\begin{bmatrix} 9.0 \end{bmatrix}$$

**Câu 5:** B. $4 \times 4$
**Câu 6:** D. Số lượng kernel: 3, kernel size: $7 \times 7$, stride: 2
**Câu 7:** D. `[1, 3, 29, 29]`
**Câu 8:** C. ResNet
**Câu 9:** B. Tích hợp nhiều kích thước kernel khác nhau trong cùng một module.
**Câu 10:** A.

$$[0.9866, 0.9051], [0.0946, 0.9702], [-0.8996, 0.9983]$$

**Câu 11:** B. $[0.9983]$
**Câu 12:** D. $1, 0, 0$
**Câu 13:** A. 2.5
**Câu 14:** A. $[2.5000, -0.5000]$
**Câu 15:** B. $[0.5000, -2.5000]$
**Câu 16:** C. $[-0.5000, -0.5000]$
**Câu 17:** A. $[-0.2769, -0.0027]$
**Câu 18:** A. $1.9, 2.0$
**Câu 19:** C. $2.9, 3.4, 1.8$
**Câu 20:** D. $13.8, 11.6$
**Câu 21:** D. $1.8, 2.4$
**Câu 22:** A. $-0.06, 1.28$
**Câu 23:** D. Nối các đầu ra theo chiều ngang (nối tiếp vào nhau).
**Câu 24:** B. Số cột của ma trận $Q$, $K$ bắt buộc phải bằng nhau, nhưng $V$ có thể khác.
**Câu 25:** A. $Q$ từ đầu ra của Masked Multi-Head Attention trong Decoder. $K$ và $V$ từ đầu ra của Encoder.
**Câu 26:** D. $128 \times 16 \times 16$
**Câu 27:** D. `(64, 32, 32 * 3)`
**Câu 28:** A. `(batch_size, seq_len, hidden_dims)`
**Câu 29:** D. 64
**Câu 30:** B. CNN $\rightarrow$ LSTM $\sim$ Transformer $\rightarrow$ RNN