1. Consider the task of Video QA where given a video and the question (example, “What is the person in the video doing?’) the task is to generate an answer (example, “Walking.”). Assume all videos are of the same length T and the answers contain a single word picked from a fixed vocabulary. We can model this task using the encoder-decoder framework as shown below:

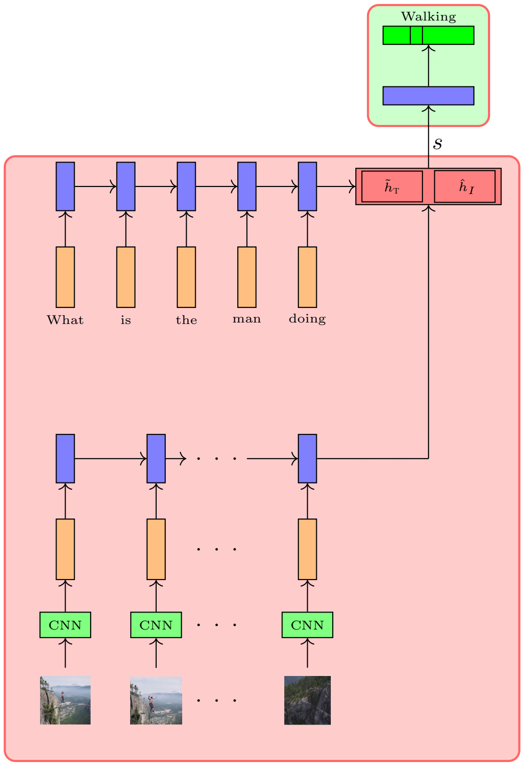

Figure 1: Video Question Answering

- **Task:** Video Question Answering
- **Data:** $\{x_i = \{video, q\}_i,\ y_i = Answer_i\}_{i=1}^{N}$
- **Model:**
  - **Encoder:**

$$\hat{h}_t = \_\_\_\_\_\_,\quad \tilde{h}_t = \_\_\_\_\_\_$$

$$s = [\tilde{h}_T; \hat{h}_T]$$

  - **Decoder:**

- **Loss:**

$$P(y|q, Video) = \_\_\_\_\_\_$$

$$\mathscr{L}(\theta) = \_\_\_\_\_\_$$

- **Algorithm:** Gradient descent with backpropagation

Given the above model, what will be a natural choice for the encoder, or what will be a natural choice for $\hat{h}_t$ and $\tilde{h}_t$, where $\hat{h}_t$ represents the video encoding while $\tilde{h}_t$ represents the question encoding.

A. $\hat{h}_t = CNN(Video_{it})$, $\tilde{h}_t = RNN(\tilde{h}_{t-1}, q_{it})$

B. $\hat{h}_t = RNN(\hat{h}_{t-1}, Video_{it})$, $\tilde{h}_t = RNN(\tilde{h}_{t-1}, q_{it})$

C. $\hat{h}_t = RNN(\hat{h}_{t-1}, CNN(Video_{it}))$, $\tilde{h}_t = RNN(\tilde{h}_{t-1}, q_{it})$

Solution: Option C is the correct option.

2. In the Video Question Answering task defined in Question 1, what will be the equation of the decoder?

A. $P(y|q, Video) = sigmoid(Vs + b)$

B. $P(y|q, Video) = softmax(Vs + b)$

C. $P(y|q, Video) = Vs + b$

Solution: Option B is the correct option.

3. In the Video Question Answering task defined in Question 1, what will be an appropriate loss function ?

A. $\mathscr{L}(\theta) = -\log P(y = \ell|video)$

B. $\mathscr{L}(\theta) = -\log P(y = \ell|video, q)$

C. $\sum_{t=1}^{T} \mathscr{L}_t(\theta) = -\sum_{t=1}^{T} \log P(y_t = \ell_t|y_1^{t-1}, video, q)$

Solution: Option B is the correct option.

4. Consider the task of Video Captioning where you want to generate a textual description given a video. For example, in the following example, we want to generate the caption "A man is walking on a rope." Assume all videos are of same length T and the caption generated for each video is of length J. We can model this task using an encoder, decoder and attention mechanism as shown below:

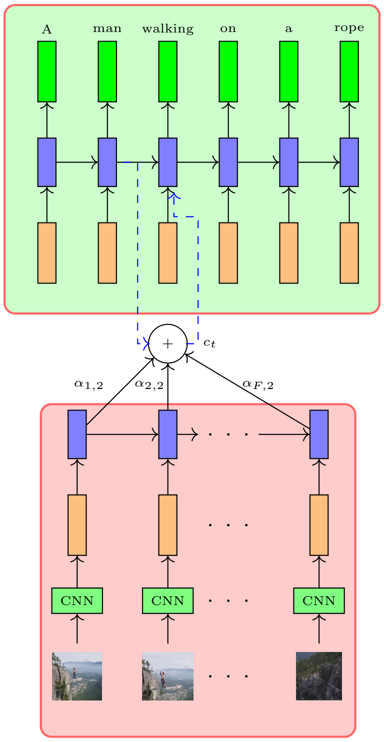

Figure 2: Video Captioning

- **Task:** Video Caption Generation (With attention)
- **Data:** $\{x_i = video_i,\ y_i = desc_i\}_{i=1}^{N}$
- **Model:**
  - **Encoder:**

$$h_t = \underline{\qquad\qquad}$$
$$s_0 = h_T$$

  - **Decoder:**

$$e_{jt} = V_{attn}^{T} tanh(U_{attn} h_j + W_{attn} s_t)$$
$$\alpha_{jt} = softmax(e_{jt})$$
$$c_t = \underline{\qquad\qquad}$$
$$s_t = \underline{\qquad\qquad}$$
$$l_t = softmax(V s_t + b)$$

- **Loss:**

$$\sum_{t=1}^{T} \mathscr{L}_t(\theta) = -\sum_{t=1}^{T} \log P(y_t = \ell_t | y_1^{t-1}, x)$$

- **Algorithm:** Gradient descent with backpropagation

What will be the encoder equation for this task, *i.e.*, what will be h_t?

A. $h_t = RNN(h_{t-1}, CNN(f_{attn}(x_{it})))$

B. $h_t = RNN(h_{t-1}, f_{attn}(CNN(x_{it})))$

C. $h_t = RNN(h_{t-1}, CNN(x_{it}))$

Solution: **Option C** is the correct option.

5. In the context of the Video Captioning task defined in Question 4, we have seen the equation,

$$e_{jt} = V_{attn}^T tanh(U_{attn}h_j + W_{attn}s_t)$$

where e_{jt} is a _____ which tells us how much attention should be given to the j-th input frame at time step t.

A. Scalar

B. Vector

C. Matrix

D. Tensor

Solution: **Option A** is the correct option.

6. In the context of the Video Captioning task defined in Question 4, what will be the equation to calculate c_t which is the context being passed to the decoder at timestep t?

A. $c_t = \alpha_{jt}s_j$, for $j = 1$ to T

B. $c_t = \alpha_{jt}h_j$, for $j = 1$ to T

C. $c_t = \sum_{j=1}^{T} \alpha_{jt}s_j$

D. $c_t = \sum_{j=1}^{T} \alpha_{jt}h_j$

Solution: **Option D** is the correct option.

7. In the context of the Video Captioning task defined in Question 4, what will be the equation to calculate s_t which is the hidden state of the decoder at time step t?

A. $s_t = \text{RNN}(s_t, [e(\hat{y}_t), c_{t-1}])$

B. $s_t = \text{RNN}(s_{t-1}, [e(\hat{y}_{t-1}), c_{t-1}])$

C. $s_t = \text{RNN}(s_{t-1}, [e(\hat{y}_{t-1}), c_t])$

D. $s_t = \text{RNN}(s_t, [e(\hat{y}_t), c_t])$

Solution: Option C is the correct option.

8. Consider the output of the 4^{th} convolutional layer of VGGNet network given in Slide 50 of Lecture 15, which is a $28 \times 28 \times 512$ size feature map. If we were to use this model as an encoder and then introduce an attention mechanism then how many locations will the model have to learn to attend to? ?

A. 196 locations

B. 512 locations

C. 784 locations

Solution: Option C is the correct option.