FIT5215 Final Exam

Trung Le

Part A - Multiple Choices (12 questions, 25 marks)

Q1

- *Consider an image classification task with five classes {cat=1, car=2, lion=3, dolphin=4, cow=5, dog = 6}. Consider an image x. Assume that a Convolutional Neural Network gives a* ***prediction probabilities*** $f(x) = [0.1, 0.2, 0.1, 0.2, 0.3, 0.1]$ *and* ***categorial ground-truth label*** *of x is cow. What is the cross-entropy loss suffered by this prediction?*

- *A)* $-\log 0.3$ ***[x]***
- *B)* $\log 0.3$
- *C)* $-\log 0.2$
- *D)* $\log 0.2$
- *E)* $\log 0.1$
- *F)* $-\log 0.1$

Q2

- *Assume that we have **5 classes** in **{cat = 1,dolphin = 2,monkey = 3, dog = 4, elephant = 5}**. Given a data example x with **ground-truth label monkey**, assume that a feed-forward NN gives **discriminative scores** to this x as $h_1 = -2$, $h_2 = 1$, $h_3 = 5$, $h_4 = 2$, $h_5 = 4$. What is the probability to predict x as dolphin or $p(y = dolphin \mid x)$?*

- A. $\frac{e^2}{e^{-2}+e^1+e^5+e^2+e^4}$
- B. 0.5
- C. $\frac{e^{-2}}{e^{-2}+e^1+e^5+e^2+e^4}$
- D. $\frac{e^1}{e^{-2}+e^1+e^5+e^2+e^4}$ **[x]**
- E. $\log \frac{e^2}{e^{-2}+e^1+e^5+e^2+e^4}$

Q3

- Consider the LeakyReLU activation function

$$\sigma(z) = \begin{cases} z & if\ z \geq 0 \\ 0.2z & otherwise \end{cases}$$

- Assume that $h = \sigma(\bar{h})$ with $\bar{h} = [-1, 2, -3]$. What is the derivative $\frac{\partial h}{\partial \bar{h}}$?

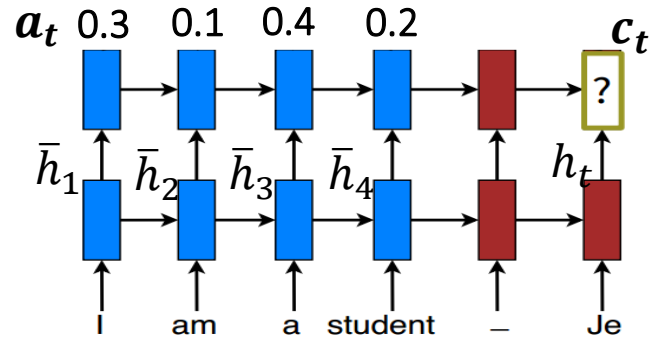
- A) diag([-1, 2, -3])
- B) diag([0, 1 ,0])
- C) diag([0.2, 1 ,0.2]) **[x]**
- D) [0, 1, 0]
- E) [0.2, 1 ,0.2]

Q4

- Given the function $f(w) = \frac{1}{1000}\sum_{i=1}^{1000}(w - x_i)^2$ where $x_i = i, \forall i = 1, \dots, 1000$. We need to solve $\min_w f(w)$ using stochastic gradient descent with the learning rate $\eta = 0.1$. Assume we sample a batch $b_1 = 1, b_2 = 5$ of indices and at the iteration t, we have $w_t = 5$. What is the value of w_{t+1} at the next iteration?
  - A) 4.6 **[X]**
  - B) 4.7
  - C) 5.1
  - D) 5.2
  - E) 4.8

Q5

Consider the below seq2deq model. We apply the global attention to compute the context vector c_t. What are correct?



A. The third word is more important than other words to the generation of the current output word.

B. The first word is more important than other words to the generation of the current output word.

C. $c_t = 0.3\bar{h}_1 + 0.1\bar{h}_2 + 0.4\bar{h}_3 + 0.2\bar{h}_4$

D. $c_t = 0.1\bar{h}_2 + 0.4\bar{h}_3 + 0.2\bar{h}_4$

E. $c_t = 0.2\bar{h}_1 + 0.4\bar{h}_2 + 0.1\bar{h}_3 + 0.3\bar{h}_4$

Q6

*Assume that the tensor before the last tensor of a CNN has shape [64, 32, 32, 16] and we apply **10 filters** each of which has the shape **[3,3,16]** and strides= **[2,2]** with padding = **'same'** to obtain the last tensor. We flatten this tensor to a **fully connected** (FC) layer. What is the **number of neurons** on this FC layer?*

- A. 16 x 16 x 10 **[x]**
- B. 64 x 16 x 16 x 10
- C. 15 x 15 x 10
- D. 64 x 15 x 15 x 10

Q7

*Assume that the tensor before the last tensor of a CNN has shape [128, 64, 64, 10] and we apply **16 filters** each of which has the shape **[3,3,10]** and strides= **[2,2]** with padding = '**valid** to obtain the last tensor. What is the **shape** of the output tensor?*

- ❑ A. [128, 31, 31, 16] **[x]**
- ❑ B. [128, 32, 32, 16]
- ❑ C. [31, 31, 16]
- ❑ D. [32, 32, 16]

Q8

Given a constraint of an adversarial example as follow: $x_{adv} \in B_\epsilon(x) = \{x' : \|x' - x\|_\infty \leq \epsilon\}$. *Which statements are correct? (MC)*

- A. This constraint to make sure that x_{adv} and x look very similar under human perspective **[x]**
- B. This constraint to make sure that x_{adv} and x look very different under human perspective
- C. This constraint to make sure that $argmax_{1 \leq m \leq M} f_m(x_{adv}) = argmax_{1 \leq m \leq M} f_m(x)$
- D. The highest absolute difference between pixels of x_{adv} and x is less than or equal ϵ **[x]**

Q9

According to the following code, what is the shape of h2?

- A. [None, 5, 64]
- B. [None, 5, 200]
- C. [None, 5, 8]
- D. [None, 5, 16] **[x]**

```
embed_size = 64
vocab_size = 200
x = tf.keras.Input(shape=[5],dtype= 'int64')
h1 = tf.keras.layers.Embedding(vocab_size, embed_size)(x)
h2 = tf.keras.layers.GRU(16, return_sequences= True)(h1)
h3 = tf.keras.layers.GRU(8, return_sequences= True)(h2)
h4 = tf.keras.layers.GRU(16, return_sequences= True)(h3)
h5 = tf.keras.layers.Flatten()(h4)
h6 = tf.keras.layers.Dense(10, activation='softmax')(h5)
```

Q10

Given a CBOW model with vocabulary size 1,000 and embedding size 250, we consider a target word with index 10 and context words with indices 15, 25, 35, 45 respectively. Let U and V be two weight matrices connecting input to hidden layers and hidden to output layers. What statements are correct? (MC)

- A. Shape of U is [1000,1000] and shape of V is [250,250]
- B. Shape of U is [1000,250] and shape of V is [250,1000] **[x]**
- C. Input to the network is one-hot vector 1_{10}.
- D. Input to the network is $\frac{1_{15}+1_{25}+1_{35}+1_{45}}{4}$. **[x]**
- E. The hidden value h is the average of rows $15,25,35,45$ of the matrix U **[x]**
- F. The hidden value h row 10 of the matrix U

Q11

- In the objective function of sparse auto-encoder: $\min_{\theta,\Phi} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}}\left[d(\boldsymbol{x}, g_\Phi(f_\theta(\boldsymbol{x}))\right] + \lambda\Omega(\boldsymbol{z})$ where $z = f_\theta(x)$, what are correct?

A. The first term is regularization term and the second one is reconstruction term
B. The second term is a regularization term and the first one is a reconstruction term ✓
C. The second term helps to eliminate redundant elements in the latent codes ✓
D. Setting higher values for λ leads to denser latent codes
E. The first term helps to regularize the latent codes
F. Setting too high values λ compromises the reconstructed images ✓

Q12

- What is the main source of mode collapsing problem of GANs?

A. The generator is too weak and cannot generate data to cover all data modes

B. The discriminator cannot classify well real and fake data

C. We cannot solve the min-max problem of GANs perfectly ✓

D. When updating the generator, there is not any constraints for it to generate data corresponding to all modes ✓

$$\max_D J(G,D) = \mathbb{E}_{\boldsymbol{x}\sim p_d(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}))]$$

$$\max_G \mathbb{E}_{\boldsymbol{z}}[\log(D(G(\boldsymbol{z}))] \Rightarrow$$

Just guide $G(z)$ to move toward some modes to gain high D values

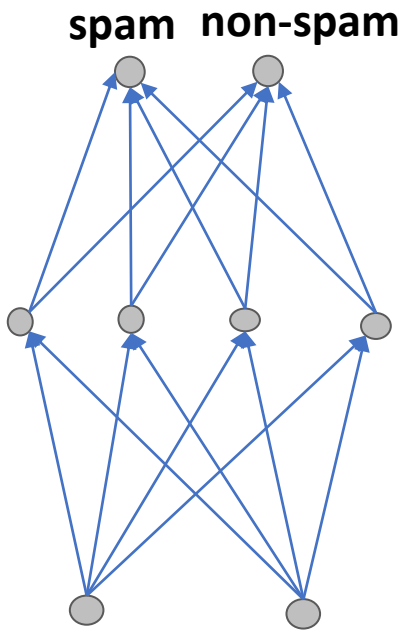
Not require to move generated examples to all data modes.

Part B Short Workout & Knowledge Questions (6 questions, 35 marks)

II.1. Computational process of MLP

- Consider a feed-forward neural network as shown in the figure for spam email detection with two labels (spam=1 and non-spam=2). Assume that we feed a feature vector $x = [1,1]^T$ with true label $y = 2$ to the network.

1. What are the formulas and the values of
2. $\bar{h}^1, h^1$? **[3 points]**
3. What are the formulas and the values for the logit h^2 and the prediction probability p? **[3 points]**
4. What is the predicted label $\hat{y}$ and the cross-entropy loss for this prediction? Is it a correct or incorrect prediction? **[4 points]**



$$h^2, p = softmax(h^2)$$

$$W^2 = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 0 & 1 & -1 & 1 \end{bmatrix}, b^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\bar{h}^1, h^1 = ReLU(\bar{h}^1)$$

$$W^1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \\ 1 & -1 \\ -2 & 1 \end{bmatrix}, b^1 = \begin{bmatrix} -2 \\ 0 \\ -1 \\ 1 \end{bmatrix}$$

$$x = [1,1]^T, y = 2$$

Solution

1. $\bar{h}^1 = W^1 x + b^1 = [0\ 1\ 2]^T + [-2\ 0\ 0]^T = [-2\ 1\ 2]^T$ (**1.5 points**)

   $h^1 = ReLU(\bar{h}^1) = [0\ 1\ 2]^T$ (**1.5 points**)

2. $h^2 = W^2 h^1 + b^2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} [0\ 1\ 2]^T + [0\ 0]^T = [3\ 3]^T$ (**1.5 points**)

   $p = softmax(h^2) = [0.5\ 0.5]$ (**1.5 points**)

3. The prediction label $\hat{y} = 1$ or $\hat{y} = 2$ (students can make assumption about how to make prediction) (**2 points**)

$l(y, p) = CE(1_y, p) = -\log 0.5$ (**2 points**)

II.2. CNN

Assume that we conduct a Convolution Neural Network (CNN) with the configuration as shown in the below figure to predict the image dataset Cifar100 with 100 classes. We feed a batch of images with the shape [16, 32, 32, 3] our CNN. Answer the following questions.

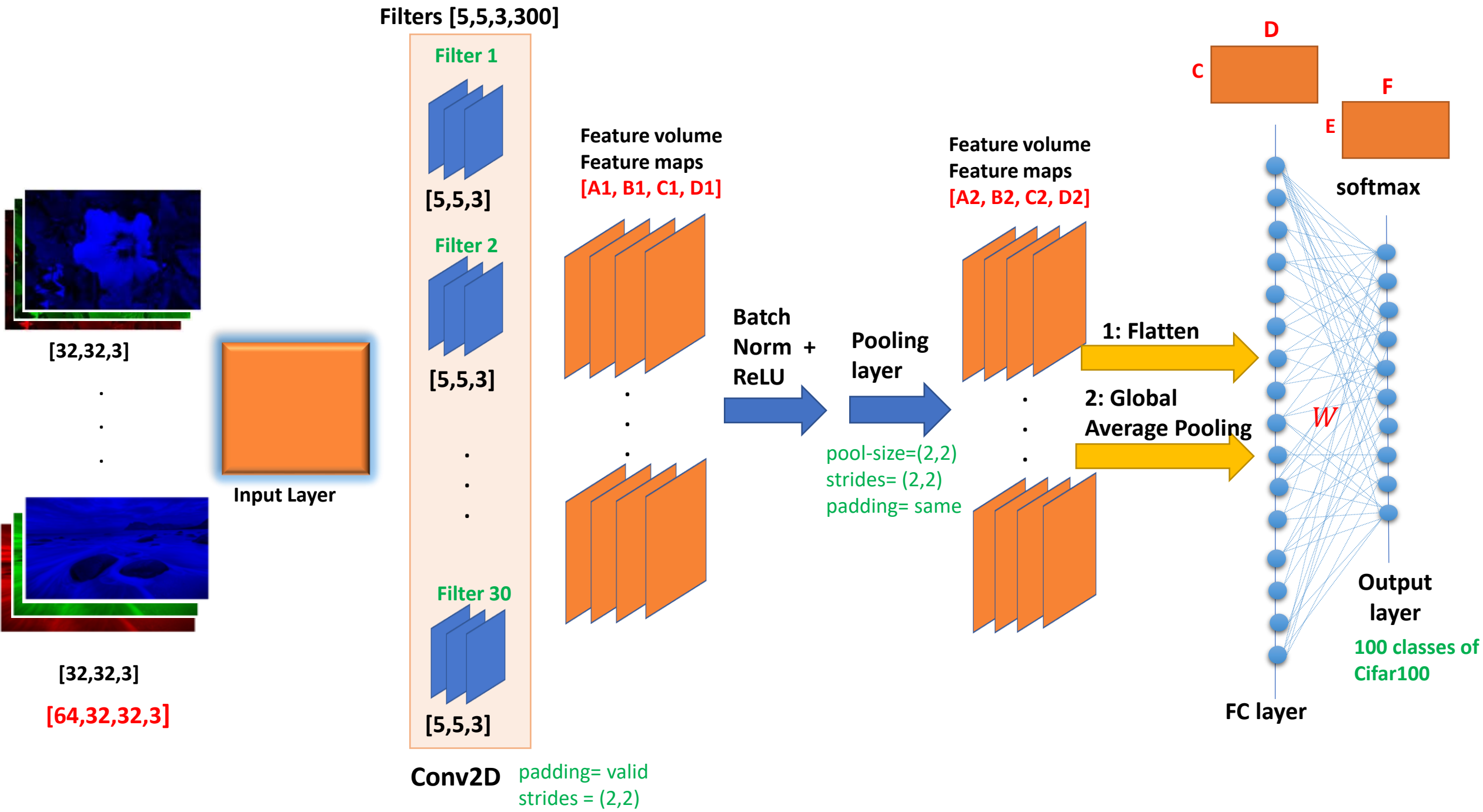
A) What is the shape of the feature maps [A1, B1, C1, D1]? Show the steps of your answer. (**3 points**)

B) What is the shape of the feature maps [A2, B2, C2, D2]? Show the steps of your answer. (**3 points**)

C) What is the shape of the 2D tensor [C,D]? Explain your answer. (**2 points**)

D) What is the shape of the 2D tensor [E,F]? Explain your answer. (**2 points**)

[10 points]


Filters [5,5,3,300]
Filter 1
[5,5,3]
Filter 2
[5,5,3]
Filter 30
[5,5,3]
[32,32,3]
[32,32,3]
[64,32,32,3]
Input Layer
Conv2D
padding= valid
strides = (2,2)
Feature volume
Feature maps
[A1, B1, C1, D1]
Batch
Norm +
ReLU
Pooling
layer
pool-size=(2,2)
strides= (2,2)
padding= same
Feature volume
Feature maps
[A2, B2, C2, D2]
1: Flatten
2: Global
Average Pooling
D
C
F
E
softmax
W
Output
layer
100 classes of
Cifar100
FC layer

Solution

(A) A1= 16 is batch size, D1 = 16 is number of used filters, B1=C1= floor((32-1)/2) +1 = 16 → [16, 16, 16, 20] (**2 points**)

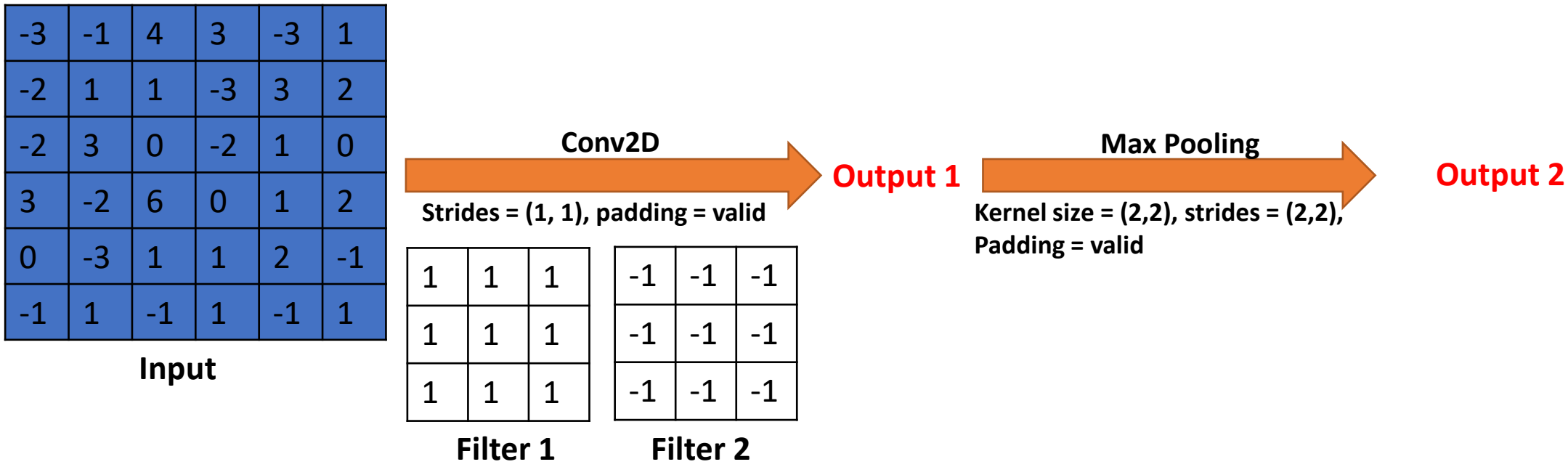
(B) A2= 16 is batch size, D2 = 16, B2 = C2 = floor((16-2)/2) + 1 = 8 → [16, 8, 8, 20] (**2 points**)

(C) If flattening, C = 16, D = 8 x 8 x 20 = 1280 (**2 points**)

(D) If using global pooling, C= 16, D = 20 (**2 points**)

(E) E = 16 is batch size, F = 100 is the number of classes. (**2 points**)

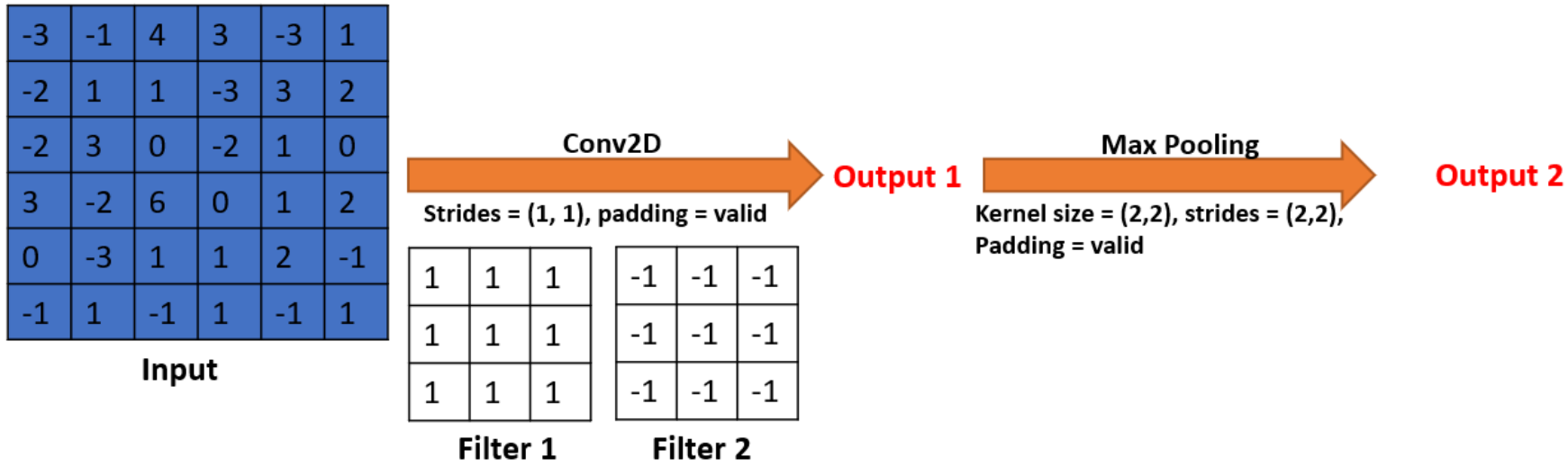
II.3. CNN



- Assume that we have [6,6,1] input tensor as shown below. We first apply a Conv2D layer with two filters (i.e., filter 1 and filter 2), strides = (1,1), and padding = valid to obtain the output 1. We then apply max pooling with kernel size = (2,2), strides = (2,2), and padding= valid to obtain the output 2.

1. What are the values of the feature maps in Output 1? **[5 points]**
2. What are the values of the feature maps in Output 2? **[5 points]**

Solution



1. Output 1 has two feature maps **[5 points]**

1	6	4	2
8	4	7	4
6	4	10	4
4	4	10	6

-1	-6	-4	-2
-8	-4	-7	-4
-6	-4	-10	-4
-4	-4	-10	-6

2. Output 2 has two feature maps **[5 points]**

8	7
6	10

-1	-2
-4	-4

II.4. Reading code for RNN

Read the following code and provide the shapes of the tensors x, h1, h2, h3, h4, h5, h6. Note that the shapes should contain one dimension with the Value None for batch size

```
embed_size = 256
vocab_size = 512
x= tf.keras.Input(shape = [20], dtype = 'int64')
h1 = tf.keras.layers.Embedding(vocab_size, embed_size)(x)
h2 = tf.keras.layers.LSTM(32, return_sequences = True)(h1)
h3 = tf.keras.layers.LSTM(64, return_sequences = True)(h2)
h4 = tf.keras.layers.LSTM(32, return_sequences = False)(h3)
h5 = tf.keras.layers.Flatten()(h4)
h6 = tf.keras.layers.Dense(15, activation = 'softmax')(h5)
```

[5 points]

Solution

```
embed_size = 256
vocab_size = 512
x= tf.keras.Input(shape = [20], dtype = 'int64')
h1 = tf.keras.layers.Embedding(vocab_size, embed_size)(x)
h2 = tf.keras.layers.LSTM(32, return_sequences = True)(h1)
h3 = tf.keras.layers.LSTM(64, return_sequences = True)(h2)
h4 = tf.keras.layers.LSTM(32, return_sequences = False)(h3)
h5 = tf.keras.layers.Flatten()(h4)
h6 = tf.keras.layers.Dense(15, activation = 'softmax')(h5)
```

- x: [None, 20] (**1 point**)
- h1: [None, 20, 256] (**0.5 point**)
- h2: [None, 20, 32] (**0.5 point**)
- h3: [None, 20, 64] (**0.5 point**)
- h4: [None, 32] (**1 point**) (return_sequences = False)
- h5: [None, 32] (**1 point**)
- h6: [None, 15] (**0.5 point**)

II.5. Reading the code for AE

- Read the following code and provide the shapes of the tensors x, h1, h2, z, hbar2, hbar1, xr.
- Note that the shapes should contain one dimension with the Value None for batch size

```
x = tf.keras.Input(shape = [32,32,3], dtype = 'float64')
h1 = tf.keras.layers.Conv2D(10, kernel_size=3, strides=[4,4], padding= "same", activation = 'relu')(x)
h2 = tf.keras.layers.Conv2D(20, kernel_size=3, strides=[2,2], padding= "same", activation = 'relu')(h1)
z= tf.keras.layers.Flatten()(h2)
hbar2 = tf.keras.layers.Reshape([h2.shape[1],h2.shape[2], h2.shape[3]])(h2)
hbar1 = tf.keras.layers.Conv2DTranspose(10, kernel_size = 3, strides = [2,2], padding= 'same', activation = 'relu')(hbar2)
xr = tf.keras.layers.Conv2DTranspose(3, kernel_size = 3, strides = [4,4], padding= 'same', activation = 'relu')(hbar1)
```

[5 points]

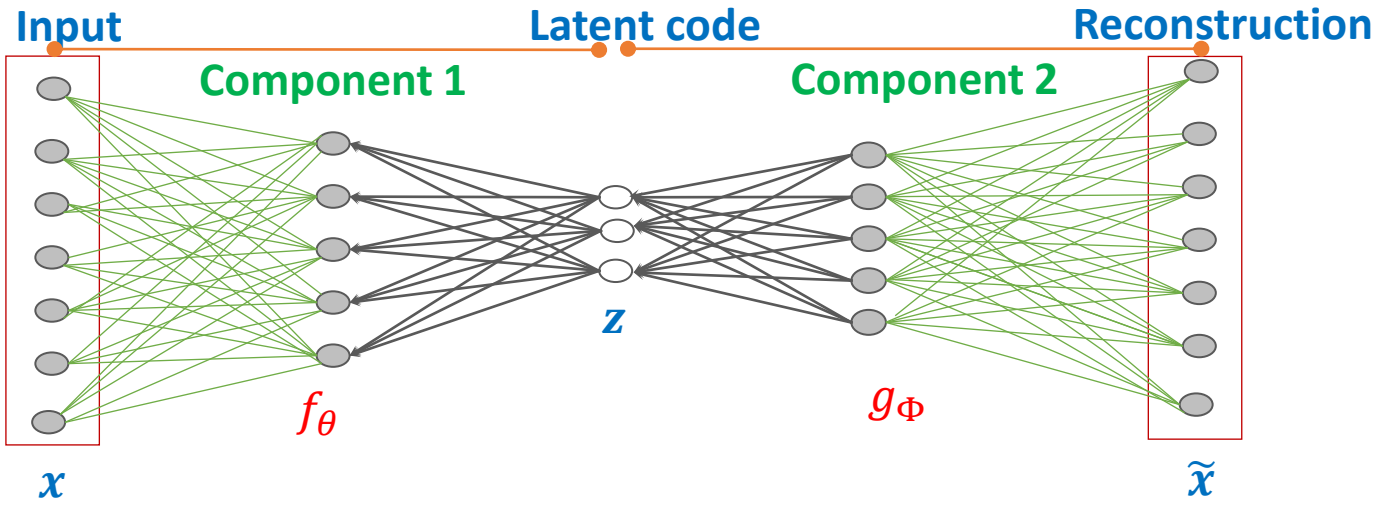
Solutions

```
x = tf.keras.Input(shape = [32,32,3], dtype = 'float64')
h1 = tf.keras.layers.Conv2D(10, kernel_size=3, strides=[4,4], padding= "same", activation = 'relu')(x)
h2 = tf.keras.layers.Conv2D(20, kernel_size=3, strides=[2,2], padding= "same", activation = 'relu')(h1)
z= tf.keras.layers.Flatten()(h2)
hbar2 = tf.keras.layers.Reshape([h2.shape[1],h2.shape[2], h2.shape[3]])(h2)
hbar1 = tf.keras.layers.Conv2DTranspose(10, kernel_size = 3, strides = [2,2], padding= 'same', activation = 'relu')(hbar2)
xr = tf.keras.layers.Conv2DTranspose(3, kernel_size = 3, strides = [4,4], padding= 'same', activation = 'relu')(hbar1)
```

- x: [None, 32, 32, 3] (**0.5 point**)
- h1: [None, 8, 8, 10] (**0.5 point**)
- h2: [None, 4, 4, 20] (**0.5 point**)
- z: [None, 320] (**0.5 point**)
- hbar2: [None, 4, 4, 20] (**1 point**)
- hbar1: [None, 8, 8, 10] (**1 point**)
- xr: [None, 32, 32, 3] (**1 point**)

Part C - Mixed & Written Answer Questions (7 questions, 35 marks)

Q1



- Consider a deep learning model with the network architecture as shown below.
- A) What are the names of the deep learning model, component 1, and component 2? Write down the objective function to train this deep learning model with the explanations of the meaning of each term in this objective function. (**5 points**)
- B) To be able to perform denoising, we add some Gaussian noises to inputs and aim to reconstruct benign inputs from noisy inputs. Write down the objective function to train this denoising variant with the explanations of the meaning of each term in this objective function. (**2.5 points**)
- C) To gain sparser latent codes z, we apply a spare regularization term to latent codes z. . Write down the objective function to train this sparse variant with the explanations of the meaning of each term in this objective function. (**2.5 points**)
- D) To strengthen this model, we leverage with the principle of generative adversarial networks (GAN). To this end, we can view g_Φ as a generator and devise a discriminator D_γ (i.e., γ is its parameters) to discriminate $\tilde{x}$ and x. Give your further thoughts and comments about this extension. Write down the purposes of the discriminator D_γ and the component 2 (i.e., g_Φ) in this context. Write down the objective function to train D_γ. Write down the objective function to train f_θ and g_Φ when leveraging with the GAN principle. (**5 points**)

Solutions

- A) Auto-encoder (**0.5 point**), component 1 is encoder f_θ (**0.5 point**), component 2 is decoder g_Φ (**0.5 point**)
  - $\min_{\theta,\Phi} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}}\left[d(\boldsymbol{x},\tilde{\boldsymbol{x}})\right] = \min_{\theta,\Phi} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}}\left[d(\boldsymbol{x}, g_\Phi(f_\theta(\boldsymbol{x}))\right]$
  - $\min_{\theta,\Phi} \frac{1}{N}\sum_{i=1}^{N} d(\boldsymbol{x}_i,\tilde{\boldsymbol{x}}_i) = \min_{\theta,\Phi} \frac{1}{N}\sum_{i=1}^{N} d(\boldsymbol{x}_i, g_\Phi(\boldsymbol{z}_i)) =$ $\min_{\theta,\Phi} \frac{1}{N}\sum_{i=1}^{N} d(\boldsymbol{x}_i, g_\Phi(f_\theta(\boldsymbol{x_i}))$ (**1.75 point**)
  - $z = f_\theta(\boldsymbol{x})$ is the latent code for x, $\tilde{x} = g_\Phi(f_\theta(\boldsymbol{x})$ is the reconstructed image. The aim is to minimize the reconstruction error on the training set. (**1.75 point**)

  (**5 points**)

Solutions

- B) Denoising Auto-Encoder
  - $\min_{\theta,\Phi} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}}\left[\mathbb{E}_{\boldsymbol{x}'\sim N(\boldsymbol{x},\eta I)}[d(\boldsymbol{x}, g_\Phi(f_\theta(\boldsymbol{x}')))]\right]$ **(1 point)**
  - Add Gaussian noises to clean images to obtain noisy images and try to reconstruct clean images from noisy images. (**1.5 points**)

  (**2.5 points**)
- C) Sparse Auto-Encoder
  - $\min_{\theta,\Phi} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}}\left[d(\boldsymbol{x}, g_\Phi(f_\theta(\boldsymbol{x})))\right] + \lambda\Omega(\boldsymbol{z})$ **(1.5 points)**
  - $\Omega(\boldsymbol{z})$ is a regularization which is usually a norm over $\boldsymbol{z}, \lambda > 0$ **is** regularization parameter**.**
  - The first term is to reconstruct a clean image from its latent code (**0.5 points**)
  - The second term is to encourage sparse latent codes by eliminating redundant and irrelevant dimensions in latent codes. (**0.5 points**)

  (**2.5 points**)

Q2

- Consider a Convolution Neural Network (CNN) with the model parameter θ. Specifically, given an image x with the ground-truth label y, the CNN returns the prediction probabilities $f(x;\theta)$ over M classes and suffers the loss $l(f(x;\theta), y)$ where l is a loss function (e.g., the cross-entropy loss).
- A) Adversarial examples are a serious issue of CNNs. Give a definition of adversarial examples. Give a practical example to explain why adversarial examples circumvent the applications of CNNs in reality. (**3 points**)
- B) Given a benign example x and the ϵ-ball $B_\epsilon = \{x' : \|x' - x\|_\infty \leq \epsilon\}$, describe and give the formula to find out a targeted adversarial example for x. (**3 points**)
- C) Given a benign example x and the ϵ-ball $B_\epsilon = \{x' : \|x' - x\|_\infty \leq \epsilon\}$, describe and give the formula to find out a untargeted adversarial example for x. (**3 points**)
- D) Given a mini-batch $B = \{(x_1, y_1), \dots, (x_b, y_b)\}$ at an iteration, describe to perform adversarial training for this mini-batch to improve model robustness. (**3 points**)
- E) How adversarial training is similar to data augmentation? How adversarial training is different to data augmentation? (**3 points**)

Solutions

- A) The adversarial example x_{adv} of x looks similar to x from human perspective, but it can fool the model f, i.e., the model f predicts x_{adv} and x with different labels or the model f predicts x_{adv} to the label y_{adv} different from the ground-truth label y of x
  - $argmax_i f_i(x;\theta) \neq argmax_i f_i(x_{adv};\theta)$ **(1.5 points)**
- Some examples (**1.5 points**)
  - Autonomous car driving system
  - Authentication system based on human faces or biological characteristics
  - So on
  - Just need one example

Solutions

B) $\boldsymbol{x_{adv} = argmin_{x' \in B_\epsilon(x)} l(f(x'; \theta), y_{\neq})}$ **(1.5 points)**

x_{adv} in the ball around x that minimizes the loss function w.r.t. a targeted label $y_{\#} \neq y$ where y is the ground-truth label of x (**1.5 points**)

C) $\boldsymbol{x_{adv} = argmax_{x' \in B_\epsilon(x)} l(f(x'; \theta), y)}$

x_{adv} in the ball around x that maximizes the loss function w.r.t. y where y is the ground-truth label of x (**1.5 points**)

Solutions

- for epoch in n_epochs
  - for iter in range(n_iter_per_epoch)
    - Sample mini-batch $(x_1, y_1), \dots, (x_b, y_b)$ from the training set
    - Find PGD untargeted adversarial examples $x_1^{adv}, \dots, x_b^{adv}$ for $x_1, \dots, x_b$ w.r.t labels $y_1, \dots, y_b$
    - $batch\_loss = \frac{1}{b}\sum_{i=1}^{b} l(f(x_i;\theta), y_i) + \frac{1}{b}\sum_{i=1}^{b} l(f(x_i^{adv};\theta), y_i)$
    - $\theta = \theta - \eta \frac{\partial batch\_loss}{\partial \theta}$

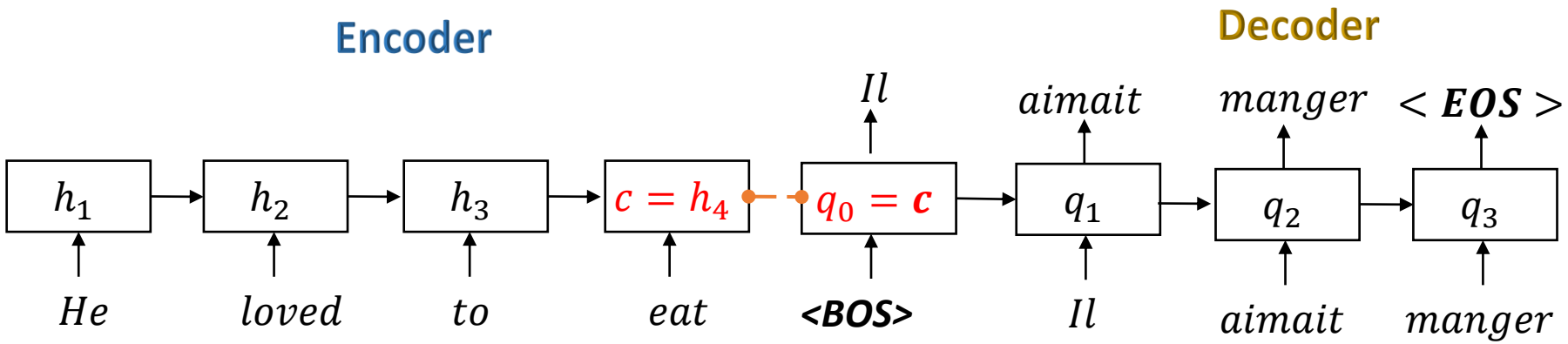
- Student can use words to describe. Give them full marks if it is correct. (**3 points**)

Solutions

- Similarity (**1.5 points**)
  - Augment the datasets with additional and challenging images and require the models to predict well on augmented images

- Difference (**1.5 points**)
  - For adversarial training, adversarial images are crafted from benign images
  - For data augmentation, augmented images are conducted from benign images through transformations (rotation, translation, skew, brightness adjustment)

Q3



- Consider the below seq2seq model for machine translation.
- A) Explain why the context vector c can summarize the content of the input sentence. (**3 points**)
- B) Describe the process to generate the output sentence from the context vector c. (**2 points**)

Solutions

A) c is a function of “eat” and h_3

- h_3 is a function of “to” and h_2
- h_2 is a function of “loved” and h_1
- h_1 is a function of “he”
- Therefore, c contains the information of “he loved to eat” and can be viewed as its lossy summary (**3 points**)

B) Input BOS to q_0 and try to predict/generate Il

- Input Il to q_1 and try to predict/generate aimat
- Input aimat to q_2 and try to predict/generate manger
- Input manger to q_3 and try to predict/generate EOS

(**2 points**)