FIT3181/5215 Deep Learning

Quiz for:
Deep Learning for time-series and temporal data:
RNNs and LSTMs

Department of Data Science and AI
Faculty of Information Technology, Monash University
Email: **trunglm@monash.edu**

Question 1

Given a sequence of random variables x_1, x_2, …, x_n, which of the following equation expresses the first-order Markov assumption property?

☐A. $p(x_1,…,x_n) = 1$

☐B. $p(x_1,…,x_n) = p(x_1)p(x_2) … p(x_n)$

☐C. $p(x_1,…,x_n) = p(x_n|x_{n-1}) \, p(x_{n-1} \mid x_{n-2}) …. p(x_2 |x_1) \, p(x_1)$

☐D. $p(x_1,…,x_n) = p(x_n |x_{1:n-1}) \, p(x_{n-1} \mid x_{1:n-2}) …. p(x_2 |x_1) \, p(x_1)$

Question 1

Given a sequence of random variables x_1, x_2, …, x_n, which of the following equation expresses the first-order Markov assumption property?

☐A. $p(x_1,…,x_n) = 1$

☐B. $p(x_1,…,x_n) = p(x_1)p(x_2) … p(x_n)$

☐C. $p(x_1,…,x_n) = p(x_n|x_{n-1}) p(x_{n-1} | x_{n-2}) …. p(x_2 |x_1) p(x_1)$ **[x]**

☐D. $p(x_1,…,x_n) = p(x_n |x_{1:n-1}) p(x_{n-1} | x_{1:n-2}) …. p(x_2 |x_1) p(x_1)$

Question 2

Given the dimension of the input is 10, and the dimension of the hidden state is 5, what is the number of trainable parameters for this single instance. (SC)

$$\overline{h}_t = tanh(h_{t-1}W + x_tU + b)$$

- ❑ A. 10x5
- ❑ B. 10x5+5
- ❑ C. 2x(10x5+5) + 5x5
- ❑ D. 10x5 + 5 + 5x5

Question 2

Given the dimension of the input is 10, and the dimension of the hidden state is 5, what is the number of trainable parameters for this single instance. (SC)
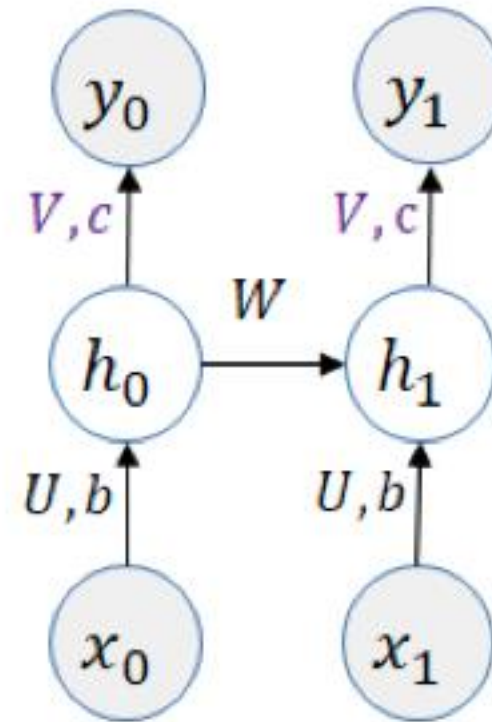
$$\bar{h}_t = tanh(h_{t-1}W + x_t U + b)$$

❑ A. 10x5

❑ B. 10x5+5

❑ C. 2x(10x5+5) + 5x5

❑ D. 10x5 + 5 + 5x5 **[x]**

Question 3

*Consider a simple RNN with two time-slices below. Assuming the **tanh** activation function is used and **classification task** is considering in all cases. Which of the following statements are correct (MC)*

- ❑ A. $h0=\tanh(x_0 U+b)$, $h1=\tanh(h_0 W+x_1 V+b)$
- ❑ B. $h0=\tanh(x_0 U+b)$, $h1=\tanh(h_0 W+x_1 U+b)$
- ❑ C. Assume the task is regression, then $y_1=h_1 U+c$
- ❑ D. Assume the task is classification using softmax function, then $y_0=\text{softmax}(h_0 V+c)$
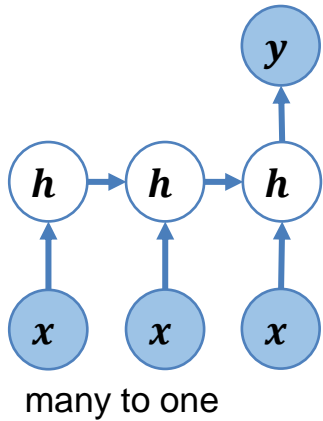
Question 3

*Consider a simple RNN with two time-slices below. Assuming the **tanh** activation function is used and **classification task** is considering in all cases. Which of the following statements are correct (MC)*

- ❑ A. $h0=\tanh(x_0 U+b)$, $h1=\tanh(h_0 W+x_1 V+b)$
- ❑ B. $h0=\tanh(x_0 U+b)$, $h1=\tanh(h_0 W+x_1 U+b)$ **[x]**
- ❑ C. Assume the task is regression, then $y_1=h_1 U+c$
- ❑ D. Assume the task is classification using softmax function, then $y_0=\text{softmax}(h_0 V+c)$ **[x]**
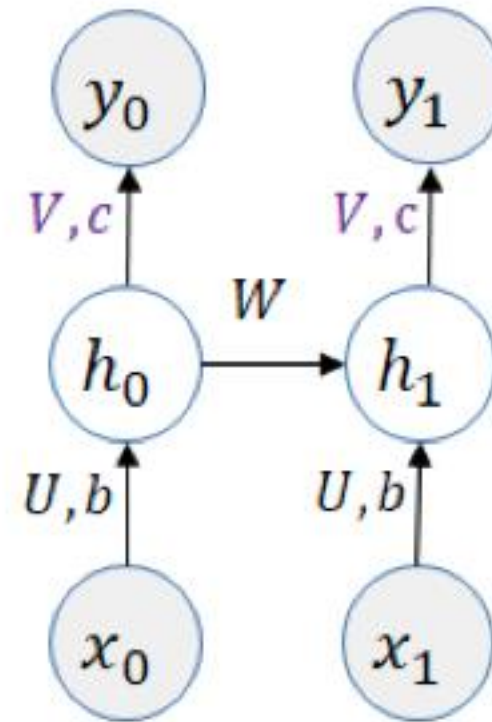
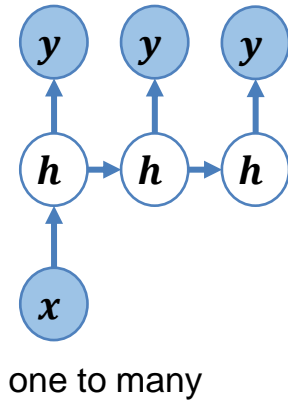# Question 4

Matching RNN topologies and applications.

1. Video classification at frame level
2. Sentiment analysis
3. Machine translation
4. Image captioning

Question 4
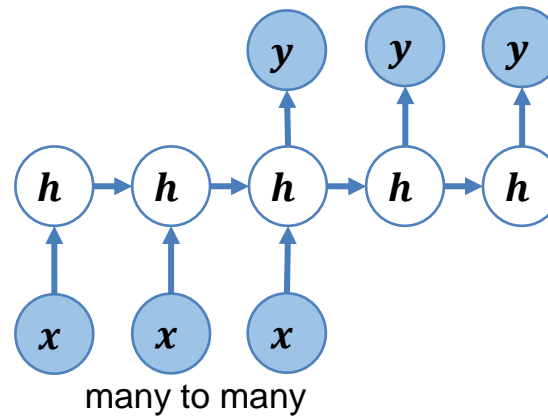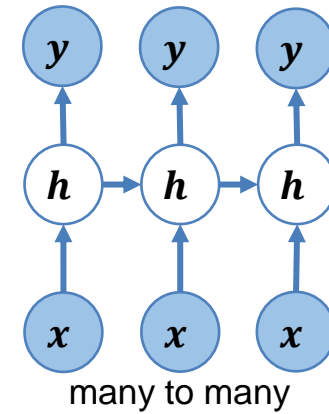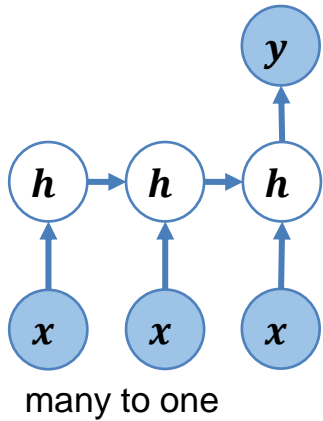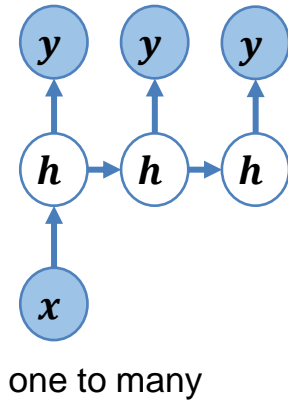
Matching RNN topologies and applications.
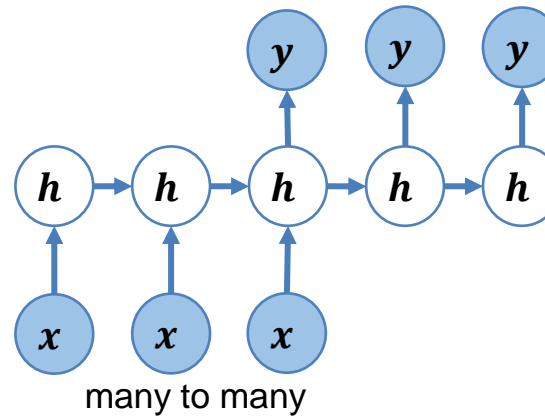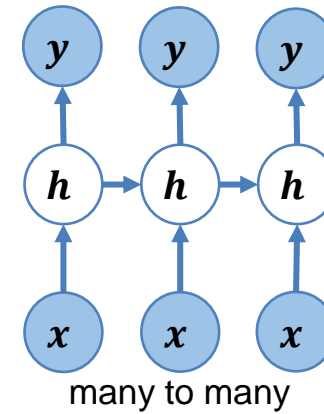
1. Video classification at frame level (D)

2. Sentiment analysis (A)

3. Machine translation (C)

4. Image captioning (B)

Question 5

Given an RNN with three hidden layers of GRU cells, assume that the state sizes of hidden states on the hidden layers are 15, 20, and 25 respectively and sequence length of input and hidden layers is 5, and the embedding size of embedding layer is 10. We feed to our network a mini-batch of shape [2,5]. What are the shape of the output tensor of embedding layer?

❑A. [2, 5, 10]

❑B. [2, 5, 15]
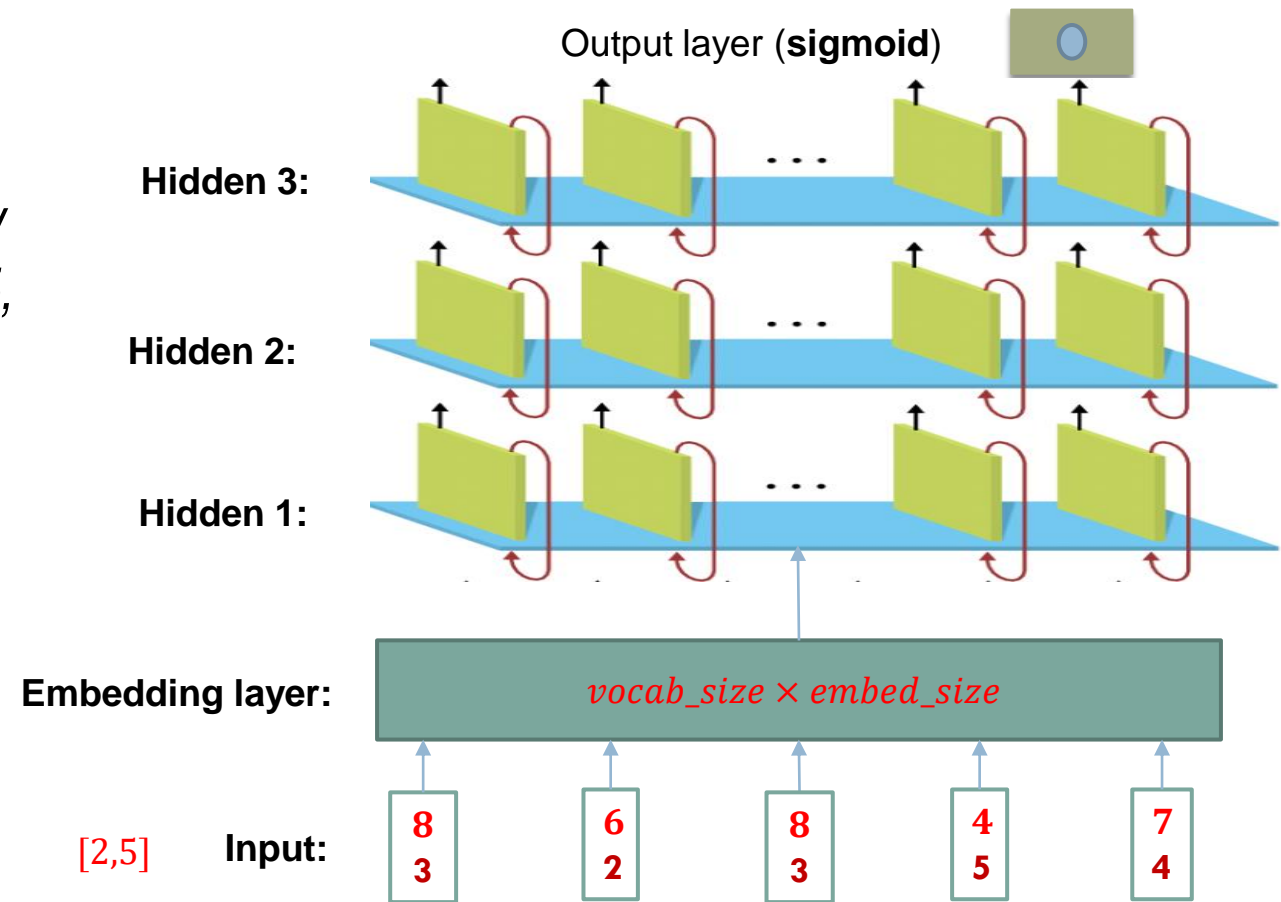
❑C. [2, 5, 20]

❑D. [2, 5, 25]

Output layer (**sigmoid**)

Hidden 3:

Hidden 2:

Hidden 1:

Embedding layer: $vocab\_size \times embed\_size$

[2,5]   Input: 8 3   6 2   8 3   4 5   7 4

Question 5

Given an RNN with three hidden layers of GRU cells, assume that the state sizes of hidden states on the hidden layers are 15, 20, and 25 respectively and sequence length of input and hidden layers is 5, and the embedding size of embedding layer is 10. We feed to our network a mini-batch of shape [2,5]. What are the shape of the output tensor of embedding layer?

☐A. [2, 5, 10] **[x]**

☐B. [2, 5, 15]

☐C. [2, 5, 20]

☐D. [2, 5, 25]

Output layer (**sigmoid**)

Hidden 3:

Hidden 2:

Hidden 1:

Embedding layer: $vocab\_size \times embed\_size$

[2,5] Input:

| 8 | 6 | 8 | 4 | 7 |
| 3 | 2 | 3 | 5 | 4 |

Question 6

Given an RNN with three hidden layers of GRU cells, assume that the state sizes of hidden states on the hidden layers are 15, 20, and 25 respectively and sequence length of input and hidden layers is 5, and the embedding size of embedding layer is 10. We feed to our network a mini-batch of shape [2,5]. What are the shape of the output tensor of hidden 1 layer?

- ❑A. [2, 5, 10]
- ❑B. [2, 5, 15]
- ❑C. [2, 5, 20]
- ❑D. [2, 5, 25]

Output layer (**sigmoid**)

Hidden 3:

Hidden 2:

Hidden 1:

Embedding layer: *vocab_size × embed_size*

[2,5]   **Input:**

| 8 | 6 | 8 | 4 | 7 |
| 3 | 2 | 3 | 5 | 4 |

Question 6

Given an RNN with three hidden layers of GRU cells, assume that the state sizes of hidden states on the hidden layers are 15, 20, and 25 respectively and sequence length of input and hidden layers is 5, and the embedding size of embedding layer is 10. We feed to our network a mini-batch of shape [2,5]. What are the shape of the output tensor of hidden 1 layer?
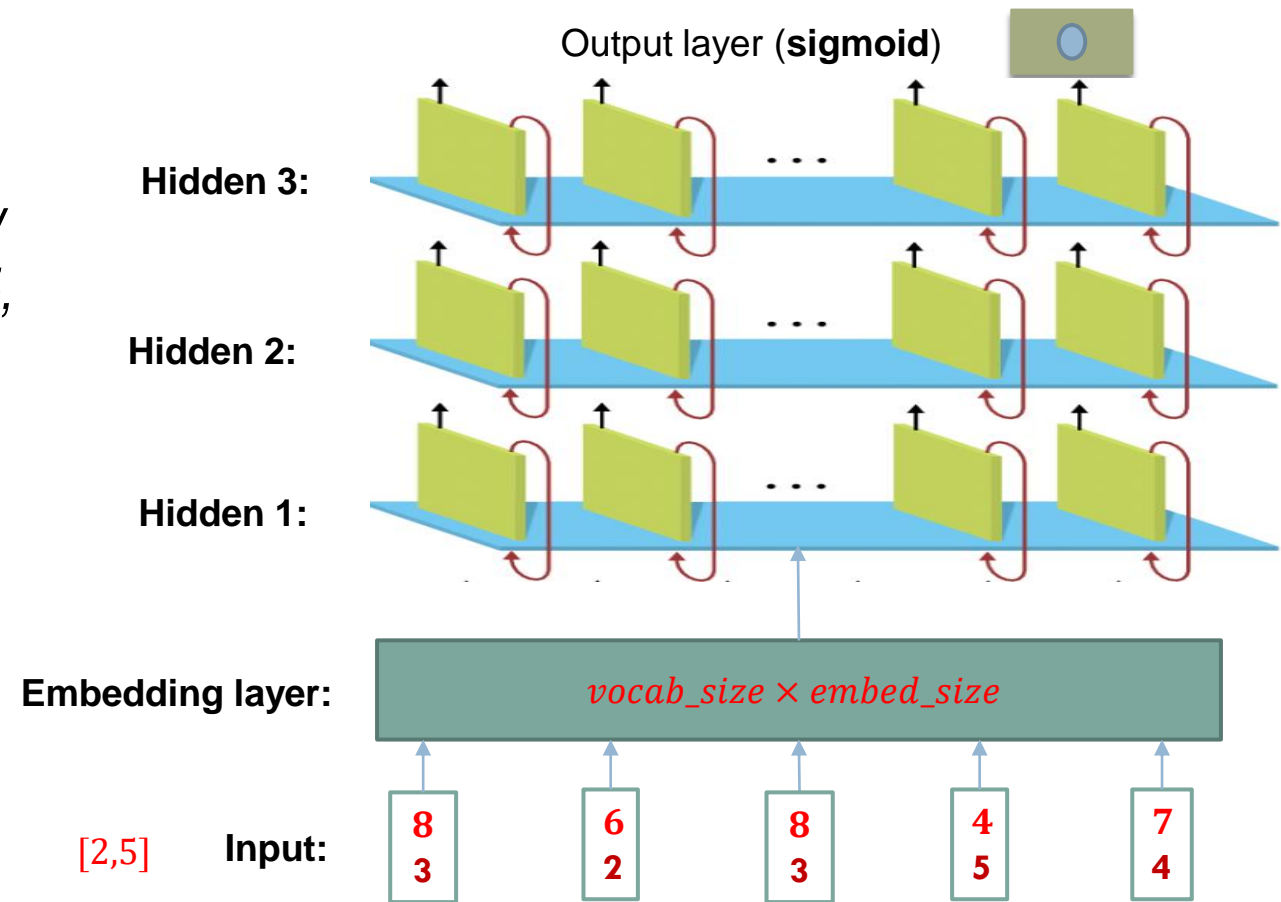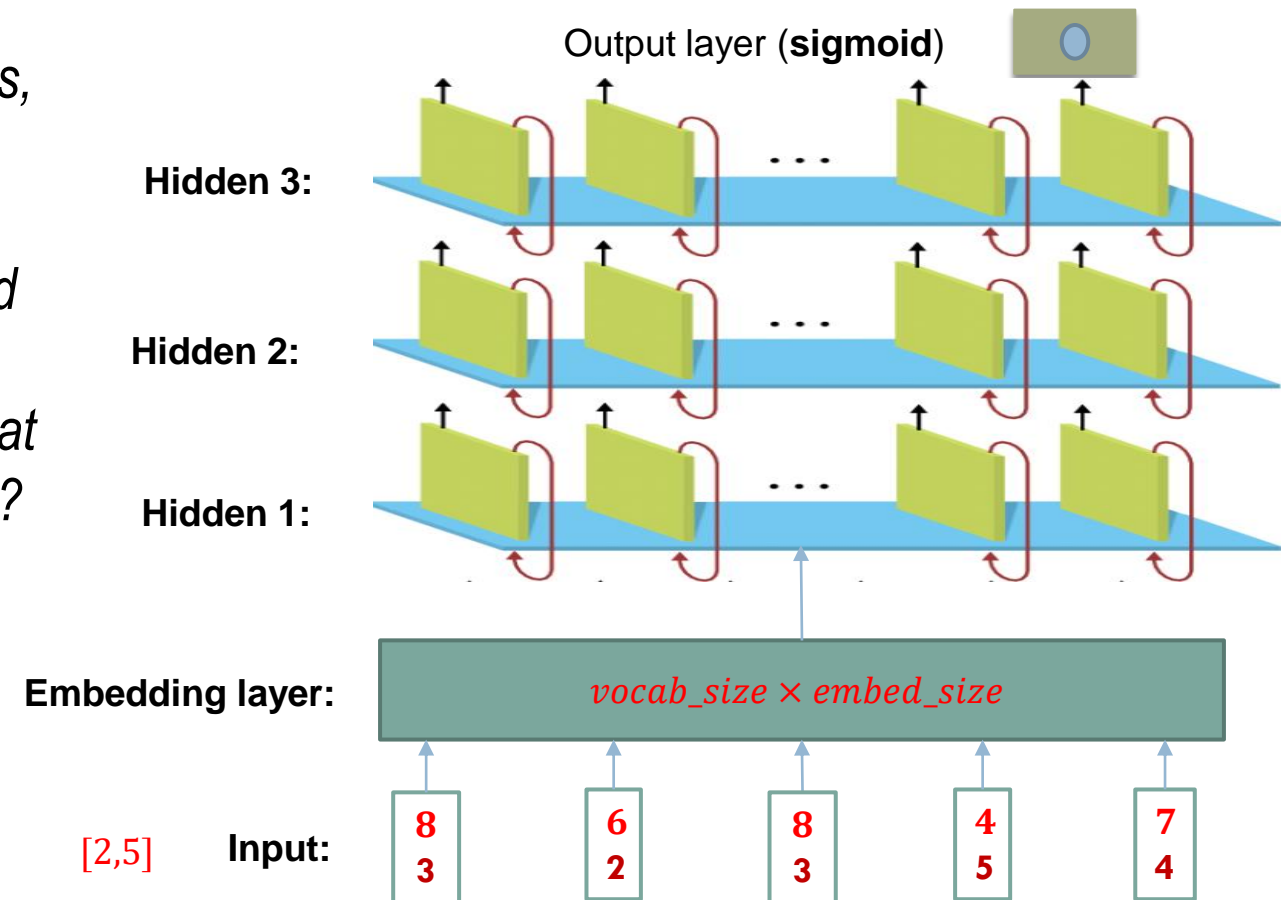
☐A. [2, 5, 10]

☐B. [2, 5, 15] **[x]**

☐C. [2, 5, 20]

☐D. [2, 5, 25]

Output layer (**sigmoid**)

Hidden 3:

Hidden 2:

Hidden 1:

Embedding layer: $vocab\_size \times embed\_size$

[2,5]   Input:

| 8 | 6 | 8 | 4 | 7 |
| 3 | 2 | 3 | 5 | 4 |

Question 7

Given an RNN with three hidden layers of GRU cells, assume that the state sizes of hidden states on the hidden layers are 15, 20, and 25 respectively and sequence length of input and hidden layers is 5, and the embedding size of embedding layer is 10. We feed to our network a mini-batch of shape [2,5]. What are the shape of the output tensor hidden 2 layer?

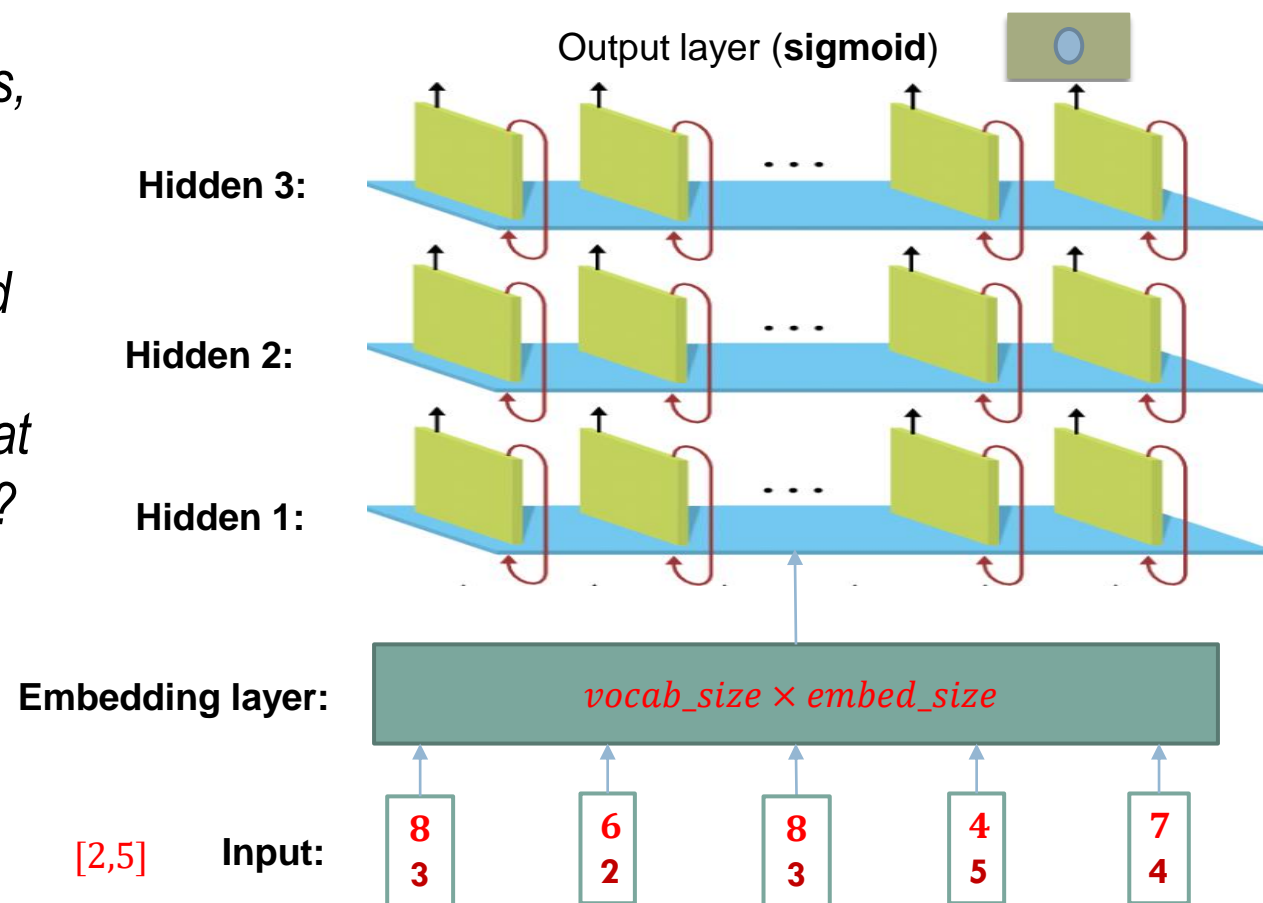- ❏A. [2, 5, 10]
- ❏B. [2, 5, 15]
- ❏C. [2, 5, 20]
- ❏D. [2, 5, 25]

Output layer (**sigmoid**)

Hidden 3:

Hidden 2:

Hidden 1:

Embedding layer: $vocab\_size \times embed\_size$

[2,5]   Input:

| 8 | 6 | 8 | 4 | 7 |
| 3 | 2 | 3 | 5 | 4 |

Question 7

Given an RNN with three hidden layers of GRU cells, assume that the state sizes of hidden states on the hidden layers are 15, 20, and 25 respectively and sequence length of input and hidden layers is 5, and the embedding size of embedding layer is 10. We feed to our network a mini-batch of shape [2,5]. What are the shape of the output tensor hidden 2 layer?

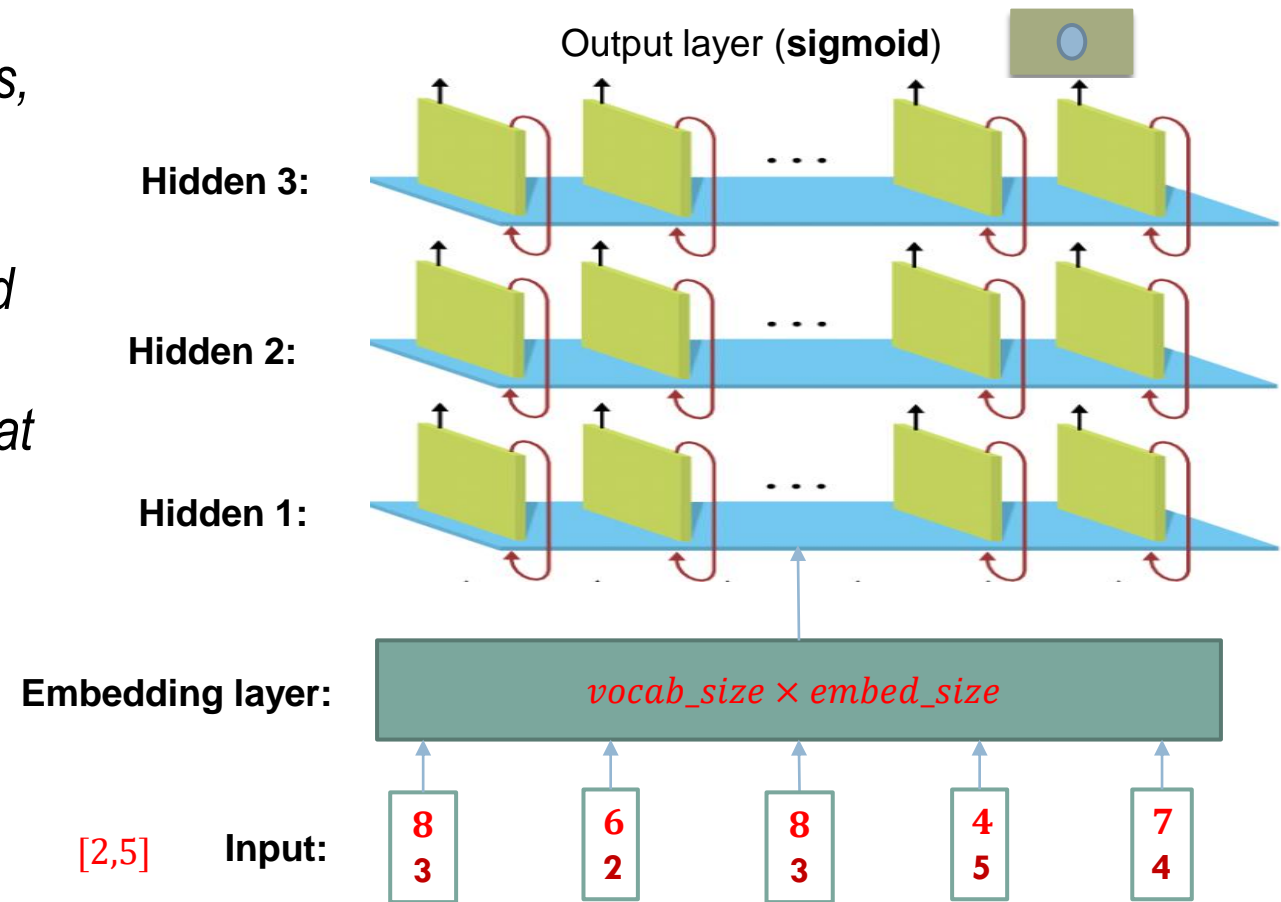❏A. [2, 5, 10]

❏B. [2, 5, 15]

❏C. [2, 5, 20] **[x]**

❏D. [2, 5, 25]

Output layer (**sigmoid**)

Hidden 3:

Hidden 2:

Hidden 1:

Embedding layer: $vocab\_size \times embed\_size$

[2,5] Input:

| 8 | 6 | 8 | 4 | 7 |
| 3 | 2 | 3 | 5 | 4 |

Question 8

Given an RNN with three hidden layers of GRU cells, assume that the state sizes of hidden states on the hidden layers are 15, 20, and 25 respectively and sequence length of input and hidden layers is 5, and the embedding size of embedding layer is 10. We feed to our network a mini-batch of shape [2,5]. What are the shape of the output tensor of hidden 3 layer?
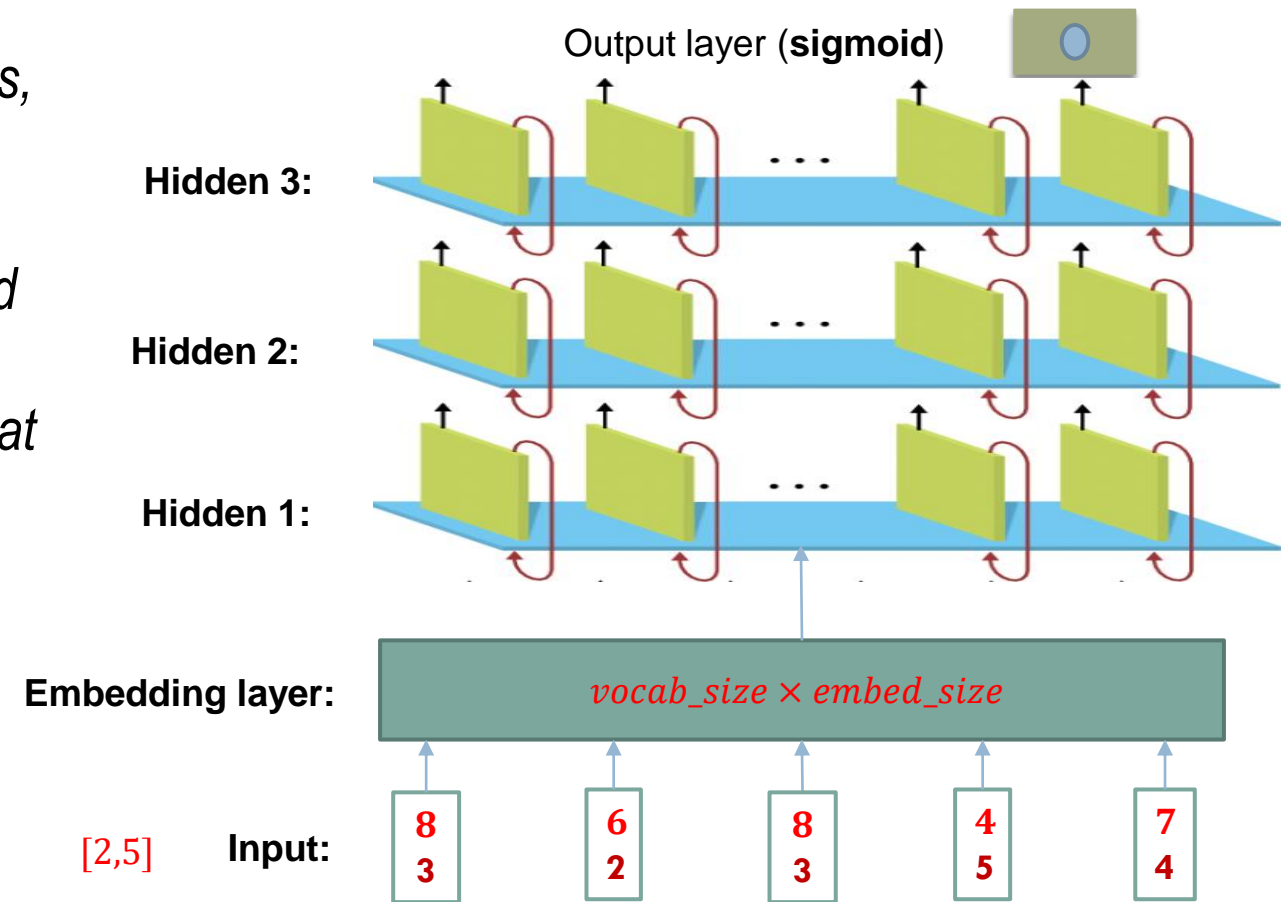
☐A. [2, 5, 10]

☐B. [2, 5, 15]

☐C. [2, 5, 20]

☐D. [2, 5, 25]

Question 8

Given an RNN with three hidden layers of GRU cells, assume that the state sizes of hidden states on the hidden layers are 15, 20, and 25 respectively and sequence length of input and hidden layers is 5, and the embedding size of embedding layer is 10. We feed to our network a mini-batch of shape [2,5]. What are the shape of the output tensor of hidden 3 layer?
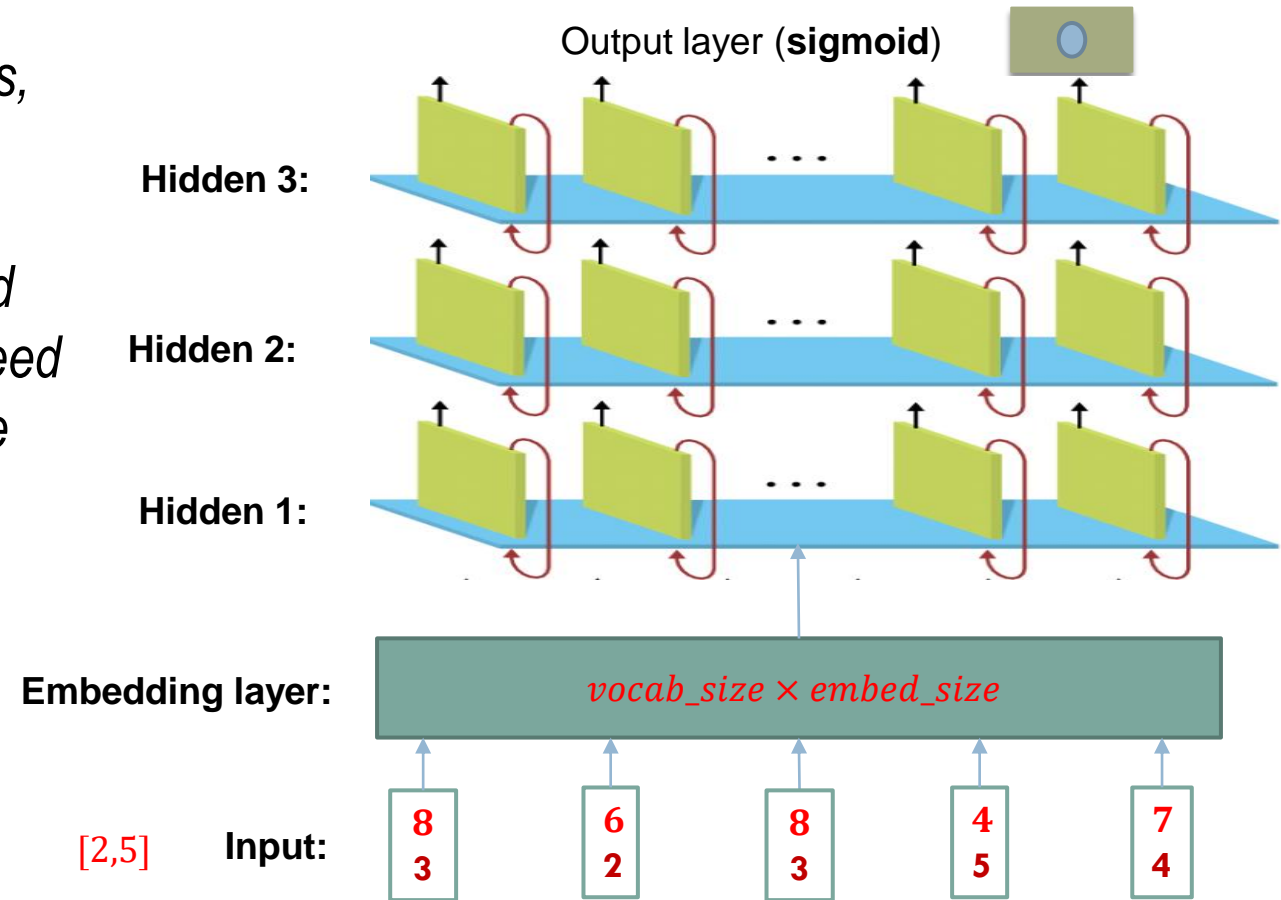
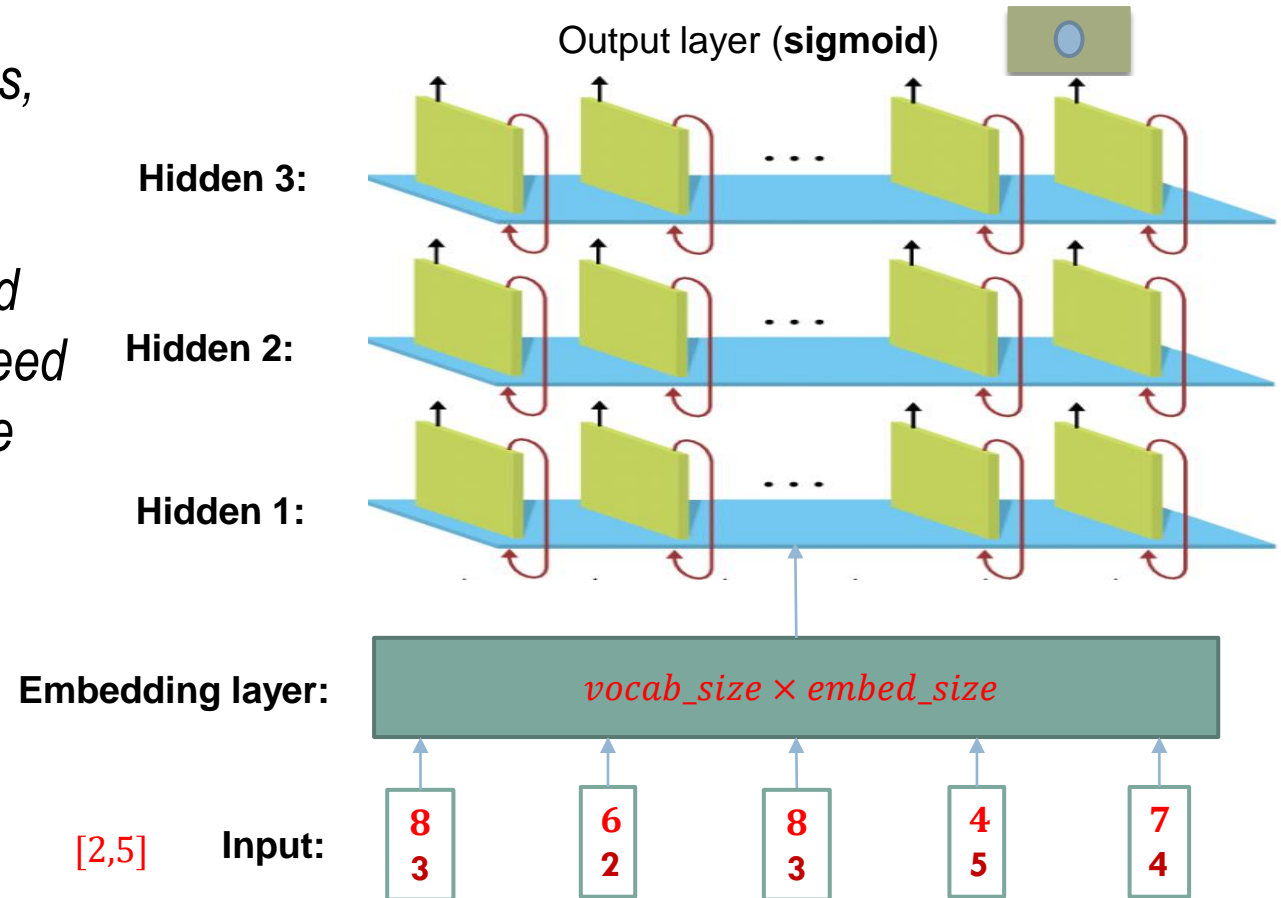☐A. [2, 5, 10]

☐B. [2, 5, 15]

☐C. [2, 5, 20]

☐D. [2, 5, 25] **[x]**

Output layer (**sigmoid**)

Hidden 3:

Hidden 2:

Hidden 1:

Embedding layer: $vocab\_size \times embed\_size$

[2,5]   **Input:**   8 3   6 2   8 3   4 5   7 4

Question 9

According to the following code, given the input shape [32, 5] where the sequence length is 5 and embedding size is 64, what is the shape of h1?

☐A. [32, 5, 64]

☐B. [32, 5, 200]

☐C. [32, 5, 8]

☐D. [32, 5, 16]

```python
class Model(nn.Module):
    def __init__(self, vocab_size, embed_size):
        super(Model, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embed_size, padding_idx=0)
        self.gru1 = nn.GRU(embed_size, 16, batch_first=True)
        self.gru2 = nn.GRU(16, 8, batch_first=True)
        self.gru3 = nn.GRU(8, 16, batch_first=True)
        self.flatten = nn.Flatten()
        self.fc = nn.LazyLinear(10)

    def forward(self, x):
        h1 = self.embedding(x)
        h2, _ = self.gru1(h1)
        h3, _ = self.gru2(h2)
        h4, _ = self.gru3(h3)
        h5 = self.flatten(h4)
        h6 = self.fc(h5)
        return h6
```

Question 9

According to the following code, given the input shape [32, 5] where the sequence length is 5 and embedding size is 64, what is the shape of h1?

A. [32, 5, 64] **[x]**

B. [32, 5, 200]

C. [32, 5, 8]

D. [32, 5, 16]

```python
class Model(nn.Module):
    def __init__(self, vocab_size, embed_size):
        super(Model, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embed_size, padding_idx=0)
        self.gru1 = nn.GRU(embed_size, 16, batch_first=True)
        self.gru2 = nn.GRU(16, 8, batch_first=True)
        self.gru3 = nn.GRU(8, 16, batch_first=True)
        self.flatten = nn.Flatten()
        self.fc = nn.LazyLinear(10)

    def forward(self, x):
        h1 = self.embedding(x)
        h2, _ = self.gru1(h1)
        h3, _ = self.gru2(h2)
        h4, _ = self.gru3(h3)
        h5 = self.flatten(h4)
        h6 = self.fc(h5)
        return h6
```

Question 10

According to the following code, given the input shape [32, 5] where the sequence length is 5 and embedding size is 64, what is the shape of h2?

❑A. [32, 5, 64]

❑B. [32, 5, 16]

❑C. [32, 5, 8]

❑D. [32, 80]

```python
class Model(nn.Module):
    def __init__(self, vocab_size, embed_size):
        super(Model, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embed_size, padding_idx=0)
        self.gru1 = nn.GRU(embed_size, 16, batch_first=True)
        self.gru2 = nn.GRU(16, 8, batch_first=True)
        self.gru3 = nn.GRU(8, 16, batch_first=True)
        self.flatten = nn.Flatten()
        self.fc = nn.LazyLinear(10)

    def forward(self, x):
        h1 = self.embedding(x)
        h2, _ = self.gru1(h1)
        h3, _ = self.gru2(h2)
        h4, _ = self.gru3(h3)
        h5 = self.flatten(h4)
        h6 = self.fc(h5)
        return h6
```

Question 10

According to the following code, given the input shape [32, 5] where the sequence length is 5 and embedding size is 64, what is the shape of h2?

❑A. [32, 5, 64]

❑B. [32, 5, 16] **[x]**

❑C. [32, 5, 8]

❑D. [32, 80]

```python
class Model(nn.Module):
    def __init__(self, vocab_size, embed_size):
        super(Model, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embed_size, padding_idx=0)
        self.gru1 = nn.GRU(embed_size, 16, batch_first=True)
        self.gru2 = nn.GRU(16, 8, batch_first=True)
        self.gru3 = nn.GRU(8, 16, batch_first=True)
        self.flatten = nn.Flatten()
        self.fc = nn.LazyLinear(10)

    def forward(self, x):
        h1 = self.embedding(x)
        h2, _ = self.gru1(h1)
        h3, _ = self.gru2(h2)
        h4, _ = self.gru3(h3)
        h5 = self.flatten(h4)
        h6 = self.fc(h5)
        return h6
```

Question 11

According to the following code, given the input shape [32, 5] where the sequence length is 5 and embedding size is 64, what is the shape of h4?

❑A. [32, 5, 8]

❑B. [32, 5, 16]

❑C. [32, 80]

❑D.[32, 10]

```python
class Model(nn.Module):
    def __init__(self, vocab_size, embed_size):
        super(Model, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embed_size, padding_idx=0)
        self.gru1 = nn.GRU(embed_size, 16, batch_first=True)
        self.gru2 = nn.GRU(16, 8, batch_first=True)
        self.gru3 = nn.GRU(8, 16, batch_first=True)
        self.flatten = nn.Flatten()
        self.fc = nn.LazyLinear(10)

    def forward(self, x):
        h1 = self.embedding(x)
        h2, _ = self.gru1(h1)
        h3, _ = self.gru2(h2)
        h4, _ = self.gru3(h3)
        h5 = self.flatten(h4)
        h6 = self.fc(h5)
        return h6
```

Question 11

According to the following code, given the input shape [32, 5] where the sequence length is 5 and embedding size is 64, what is the shape of h4?

A. [32, 5, 8]

B. [32, 5, 16] **[x]**

C. [32, 80]

D. [32, 10]

```python
class Model(nn.Module):
    def __init__(self, vocab_size, embed_size):
        super(Model, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embed_size, padding_idx=0)
        self.gru1 = nn.GRU(embed_size, 16, batch_first=True)
        self.gru2 = nn.GRU(16, 8, batch_first=True)
        self.gru3 = nn.GRU(8, 16, batch_first=True)
        self.flatten = nn.Flatten()
        self.fc = nn.LazyLinear(10)

    def forward(self, x):
        h1 = self.embedding(x)
        h2, _ = self.gru1(h1)
        h3, _ = self.gru2(h2)
        h4, _ = self.gru3(h3)
        h5 = self.flatten(h4)
        h6 = self.fc(h5)
        return h6
```

Question 12

According to the following code, given the input shape [32, 5] where the sequence length is 5 and embedding size is 64, what is the shape of h5?

```python
class Model(nn.Module):
    def __init__(self, vocab_size, embed_size):
        super(Model, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embed_size, padding_idx=0)
        self.gru1 = nn.GRU(embed_size, 16, batch_first=True)
        self.gru2 = nn.GRU(16, 8, batch_first=True)
        self.gru3 = nn.GRU(8, 16, batch_first=True)
        self.flatten = nn.Flatten()
        self.fc = nn.LazyLinear(10)

    def forward(self, x):
        h1 = self.embedding(x)
        h2, _ = self.gru1(h1)
        h3, _ = self.gru2(h2)
        h4, _ = self.gru3(h3)
        h5 = self.flatten(h4)
        h6 = self.fc(h5)
        return h6
```

❑A. [32, 16]

❑B. [32, 5, 16]

❑C. [32, 80]

❑D. [32, 10]

Question 12

According to the following code, given the input shape [32, 5] where the sequence length is 5 and embedding size is 64, what is the shape of h5?

❑A. [32, 16]

❑B. [32, 5, 16]

❑C. [32, 80] **[x]**

❑D. [32, 10]

```python
class Model(nn.Module):
    def __init__(self, vocab_size, embed_size):
        super(Model, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embed_size, padding_idx=0)
        self.gru1 = nn.GRU(embed_size, 16, batch_first=True)
        self.gru2 = nn.GRU(16, 8, batch_first=True)
        self.gru3 = nn.GRU(8, 16, batch_first=True)
        self.flatten = nn.Flatten()
        self.fc = nn.LazyLinear(10)

    def forward(self, x):
        h1 = self.embedding(x)
        h2, _ = self.gru1(h1)
        h3, _ = self.gru2(h2)
        h4, _ = self.gru3(h3)
        h5 = self.flatten(h4)
        h6 = self.fc(h5)
        return h6
```

Question 13

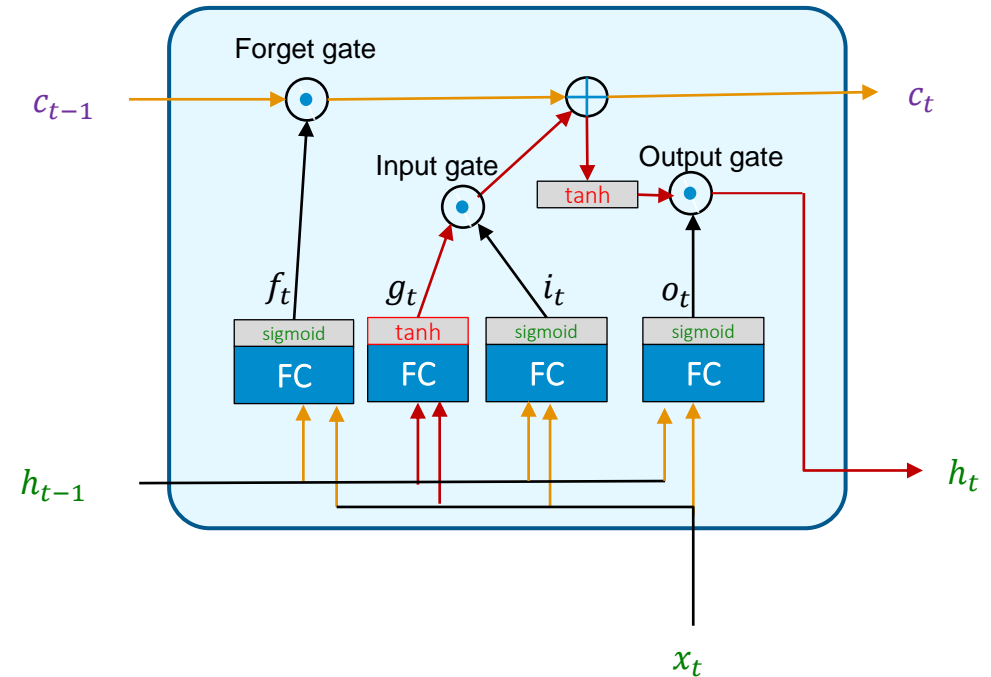*Consider LSTM cell which is shown in the following figure. What statements are correct for the forget gate?*

☐ A. We apply sigmoid activation when computing forget gate

☐ B. We apply tanh activation when computing forget gate

☐ C. It controls the proportion of information in long-term memory to be carried forward

☐ D. It controls the proportion of information in short-term memory to be carried forward
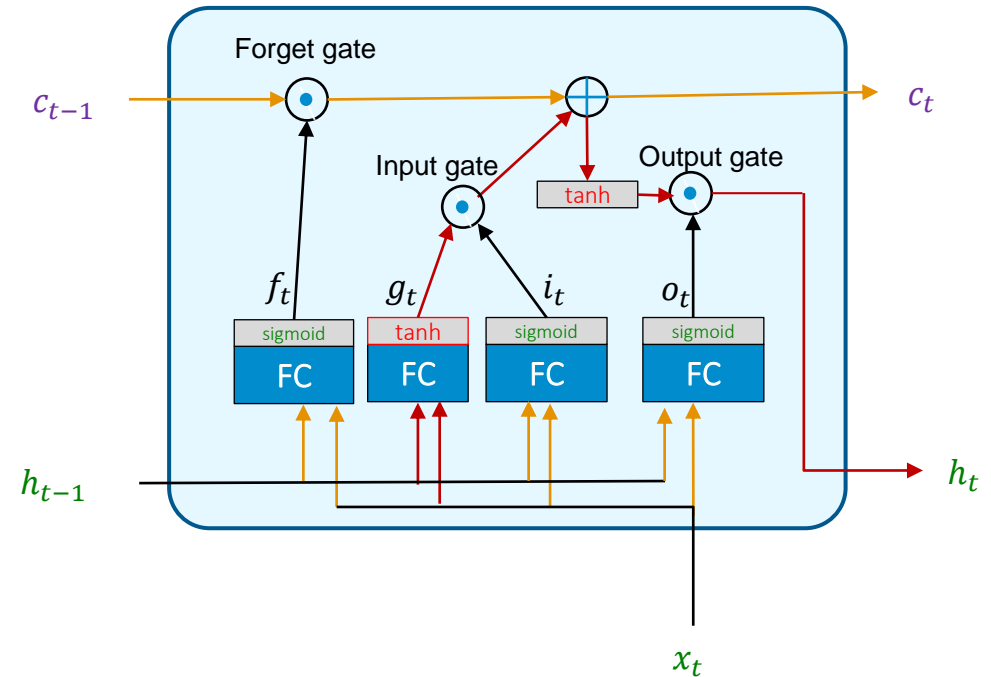
Question 13

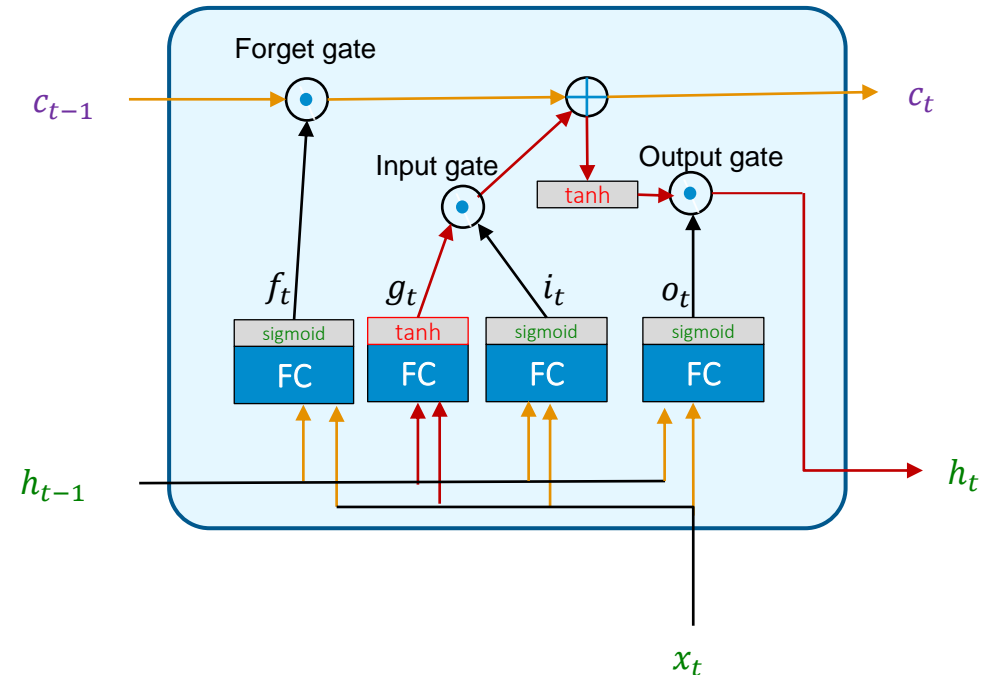*Consider LSTM cell which is shown in the following figure. What statements are correct for the forget gate?*

☐A. We apply sigmoid activation when computing forget gate **[x]**

☐B. We apply tanh activation when computing forget gate

☐C. It controls the proportion of information in long-term memory to be carried forward **[x]**

☐D. It controls the proportion of information in short-term memory to be carried forward

Question 14

Consider LSTM cell which is shown in the following figure. What statements are correct for the output gate?

☐A. We apply sigmoid activation when computing output gate

☐B. We apply tanh activation when computing output gate

☐C. It controls the proportion of information in current long-term memory to be carried forward to new short-term memory

☐D. It controls the proportion of information in current short-term memory to be carried forward to new long-term memory

Question 14

Consider LSTM cell which is shown in the following figure. What statements are correct for the output gate?

☐A. We apply sigmoid activation when computing output gate **[x]**

☐B. We apply tanh activation when computing output gate

☐C. It controls the proportion of information in current long-term memory to be carried forward to new short-term memory **[x]**

☐D. It controls the proportion of information in current short-term memory to be carried forward to new long-term memory