MONASH University

FIT5215 Deep Learning

Quiz for: Convolutional Neural Network

Tutor Team

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

GROUP OF EIGHT AUSTRALIA

Question 1

*Given an 3D input tensor with shape [32, 32, 3] over which we apply a conv2D with **16 filters** each of which has shape **[5,5]**, strides **[3,3]**, and padding **valid**. What is the shape of the output tensor?*

- A. [10, 10]
- B. [11, 11]
- C. [11, 11, 16]
- D. [10, 10, 16]

floor((32 - 5) / 3)+ 1
= 10

need to consider no. of filters!

Question 2

*Given an 3D input tensor with shape [32, 32, 3] over which we apply a conv2D with **16 filters** each of which has shape **[5,5]**, strides **[3,3]**, and padding **same**. What is the shape of the output tensor?*

- A. [10, 10]
- B. [11, 11]
- C. [11, 11, 16]
- D. [10, 10, 16]

floor((32 - 1) / 3)+ 1 = 11

need to consider no. of filters!

Question 3

*Given an 3D input tensor with shape [64, 64, 10] over which we apply a **max pooling** layer with kernel size **[3,3]**, strides **[3,3]**, and padding **same**. What is the **shape** of the **output tensor**?*

- A. [21, 21]
- B. [22, 22]
- C. [22, 22, 10]
- D. [22, 22, 3]

floor((64 - 1) / 3)+ 1 = 22

Question 4

*Assume that the tensor before the last tensor of a CNN has shape [32, 32, 32, 10] and we apply **5 filters** each of which has the shape **[5,5,10]** and strides= **[2,2]** with padding = **'same'** to obtain the last tensor. What is the **shape** of the output tensor?*

floor((32 - 1) / 2)+ 1 = 16

- A. [16, 16, 5]
- B. [14, 14, 5]
- C. [32, 14, 14, 5]
- D. [32, 16, 16, 5]

Question 5

*Assume that the tensor before the last tensor of a CNN has shape [32, 32, 32, 10] and we apply **5 filters** each of which has the shape **[5,5,10]** and strides= **[2,2]** with padding = **'valid'** 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?*

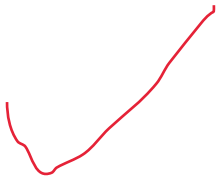
- A. 16 x 16 x5
- B. 14 x 14 x 5
- C. 32 x 16 x 16 x5
- D. 32 x 14 x 14 x 5

[32,32,32,10]
[32,14,14,5] ; floor((32-5)/2) + 1 = 14 ; 5 filters
[14,14,5]

Question 6

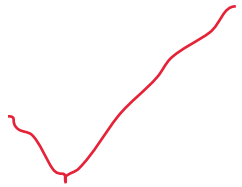
What likely happen if using a large filter (e.g., 7x7, 9x9) with a deep model (e.g., 20 layers) if there are few images?

- ❑ A. Overfitting
- ❑ B. Underfitting



Question 7

Which is a good CNN model architecture?



- ❑ A. Input layer → Convolutional layer (Activation) → Pooling layer → FC layer → Output
- ❑ B. Input layer → Pooling layer → Convolutional layer (Activation) → FC layer → Output
- ❑ C. Input layer → FC Layer → Pooling layer → Convolutional layer (Activation) → Output
- ❑ D. Input layer → Convolutional layer (Activation) → FC layer → Pooling layer → Output

Question 8

Given an implementation as below. What is the shape of h1?

```
X = Input(shape=(32, 32, 3))
h1 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='same')(X)
h1 = AveragePooling2D(pool_size=(2, 2), strides=(2, 2))(h1)
h2 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='same')(h1)
h2 = AveragePooling2D(pool_size=(2, 2), strides=(2, 2))(h2)
h3 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='same')(h2)
h4 = Flatten()(h3)
p = Dense(10)(h4)
print("h1", h1.shape)
```

- A. (16,16,3)
- B. (16,16,10)
- C. (None,16,16,3)
- D. (None, 16,16,10)

[32,32,10] ; floor((32-1)/1) + 1 =
32 after pooling: [16,16,10]

Question 9

Given an implementation as below. What is the shape of h1?

```
X = Input(shape=(32, 32, 3))
h1 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(X)
h1 = AveragePooling2D(pool_size=(2, 2), strides=(2, 2))(h1)
h2 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(h1)
h2 = AveragePooling2D(pool_size=(2, 2), strides=(2, 2))(h2)
h3 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(h2)
h4 = Flatten()(h3)
p = Dense(10)(h4)
print("h1", h1.shape)
```

- A. (None,16,16,10)
- B. (None,15,15,10)
- C. (None,14,14,10)
- D. (None, 13,13,10)

[30, 30, 10] ; floor((32-3)/1) + 1 = 30
after pooling: [15,15,10]

Question 10

Given an implementation as below. What are the shape of h1/h2/h3?

```
X = Input(shape=(32, 32, 3))
h1 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(X)
h1 = AveragePooling2D(pool_size=(2, 2), strides=(2, 2))(h1)
h2 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(h1)
h2 = AveragePooling2D(pool_size=(2, 2), strides=(2, 2))(h2)
h3 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(h2)
h4 = Flatten()(h3)
p = Dense(10)(h4)
print("h1", h1.shape)
print("h2", h2.shape)
print("h3", h3.shape)
```

floor((15 - 3)/1) + 1 = 13

after pool: floor(13-2)/2) + 1 = 6

- A. (None,15,15,10) / (None,6,6,10) / (None,3,3,10)
- B. (None,15,15,10) / (None,6,6,10) / (None,4,4,10)
- C. (None,15,15,10) / (None,7,7,10) / (None,3,3,10)
- D. (None,15,15,10) / (None,7,7,10) / (None,4,4,10)

floor((6-3)/1)+ 1 = 4

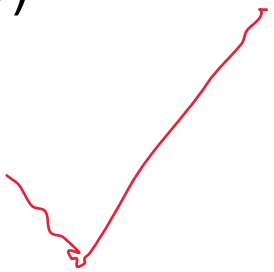
careless

Question 11

Given an implementation as below. What is the shape of h4?

```
X = Input(shape=(32, 32, 3))
h1 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(X)
h1 = AveragePooling2D(pool_size=(2, 2), strides=(2, 2))(h1)
h2 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(h1)
h2 = AveragePooling2D(pool_size=(2, 2), strides=(2, 2))(h2)
h3 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(h2)
h4 = Flatten()(h3)
p = Dense(10)(h4)
print("h4", h4.shape)
```

- A. (None,4,4,10)
- B. (None,90)
- C. (None,160)
- D. (90,)



Question 12

Given an implementation as below. What is the shape of W1 and b1?

```
X = Input(shape=(32, 32, 3))
h1 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(X) # Layer 1
h1 = AveragePooling2D(pool_size=(2, 2), strides=(2, 2))(h1)
h2 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(h1)
h2 = AveragePooling2D(pool_size=(2, 2), strides=(2, 2))(h2)
h3 = Conv2D(filters=10, kernel_size=(3, 3), strides=(1, 1), padding='valid')(h2)
h4 = Flatten()(h3)
p = Dense(10)(h4)

model = tf.keras.Model(inputs=X, outputs=p)

W1, b1 = model.layers[1].weights
print(W1.shape)
print(b1.shape)
```

width, length, depth = 3,3,3

no. filters = 10

- A. (3,3,3,10), (10,)
- B. (3,3,3,10), (3,3,10)
- C. (15,15,3,10), (10,)
- D. (15,15,3,10), (3,3,10)