Your grade: 96.66%

Your latest: **96.66%** • Your highest: **96.66%** • To pass you need at least 80%. We keep your highest score.

Next item

1. What do you think applying this filter to a grayscale image will do? **1 / 1 point**

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix}$$

- ( ) Detect horizontal edges.
- (•) Detect 45-degree edges.
- ( ) Detecting image contrast.
- ( ) Detect vertical edges.

Correct
Correct. Notice that there is a high delta between the values in the top left part and the ones in the bottom right part. When convolving this filter on a grayscale image, the edges forming a 45-degree angle with the horizontal will be detected.

2. Suppose your input is a 128 by 128 color (RGB) image, and you are not using a convolutional network. If the first hidden layer has 64 neurons, each one fully connected to the input, how many parameters does this hidden layer have (including the bias parameters)? **1 / 1 point**

- ( ) 1048576
- (•) 3145792
- ( ) 1048640
- ( ) 3145728

Correct
Correct, the number of inputs for each unit is $128 \times 128 \times 3$ since the input image is RGB, so we need $128 \times 128 \times 3 \times 64$ parameters for the weights and 64 parameters for the bias parameters, thus $128 \times 128 \times 3 \times 64 + 64 = 3145792$.

3. Suppose your input is a 300 by 300 color (RGB) image, and you use a convolutional layer with 100 filters that are each 5x5. How many parameters does this hidden layer have (including the bias parameters)? **1 / 1 point**

- (•) 7600
- ( ) 7500
- ( ) 2600
- ( ) 2501

Correct
Correct, you have $25 \times 3 = 75$ weights and 1 bias per filter. Given that you have 100 filters, you get 7,600 parameters for this layer.

4. You have an input volume that is $121 \times 121 \times 16$, and convolve it with 32 filters of 4×4, using a stride of 3 and no padding. What is the output volume? **1 / 1 point**

- (•) $40 \times 40 \times 32$
- ( ) $40 \times 40 \times 16$
- ( ) $118 \times 118 \times 16$
- ( ) $118 \times 118 \times 32$

Correct
Correct, using the formula $n_H^{[l]} = \frac{n_H^{[l-1]} + 2 \times p - f}{s} + 1$ with $n_H^{[l-1]} = 121, p = 0, f = 4$, and $s = 3$ we get 40

5. You have an input volume that is 15x15x8, and pad it using "pad=2". What is the dimension of the resulting volume (after padding)? **1 / 1 point**

- ( ) 17x17x8
- (•) 19x19x8
- ( ) 19x19x12
- ( ) 17x17x10

Correct
Correct, padding is applied over the height and the width of the input image. If the padding is two, you add 4 to the height dimension and 4 to the width dimension.

6. You have an input volume that is 63x63x16, and convolve it with 32 filters that are each 7x7, and stride of 1. You want to use a "same" convolution. What is the padding?

1 / 1 point

- ○ 2
- ○ 7
- ◉ 3
- ○ 1

Correct
Correct, you need to satisfy the following equation: $n_H - f + 2 \times p + 1 = n_H$ as you want to keep the dimensions between the input volume and the output volume.

7. You have an input volume that is 66x66x21, and apply max pooling with a stride of 3 and a filter size of 3. What is the output volume?

1 / 1 point

- ◉ $22 \times 22 \times 21$
- ○ $22 \times 22 \times 7$
- ○ $21 \times 21 \times 21$
- ○ $66 \times 66 \times 7$

Correct
Yes, using the formula $n_H^{[l]} = \frac{n_H^{[l-1]}+2\times p-f}{s} + 1$ with $p = 0, f = 3, s = 3$ and $n_H^{[l-1]} = 66$.

8. Because pooling layers do not have parameters, they do not affect the backpropagation (derivatives) calculation.

1 / 1 point

- ○ True
- ◉ False

Correct
Everything that influences the loss should appear in the backpropagation because we are computing derivatives. In fact, pooling layers modify the input by choosing one value out of several values in their input volume. Also, to compute derivatives for the layers that have parameters (Convolutions, Fully-Connected), we still need to backpropagate the gradient through the Pooling layers.

9. Which of the following are true about convolutional layers? (Check all that apply)

0.6666666666666666 / 1 point

- [ ] It speeds up the training since we don't need to compute the gradient for convolutional layers.
- [ ] Convolutional layers provide sparsity of connections.
- [x] It allows a feature detector to be used in multiple locations throughout the whole input volume.

Correct
Yes, since convolution involves sliding the filter throughout the whole input volume the feature detector is computed over all the volume.

You didn't select all the correct answers

10. The sparsity of connections and weight sharing are mechanisms that allow us to use fewer parameters in a convolutional layer making it possible to train a network with smaller training sets. True/False?

1 / 1 point

- ◉ True
- ○ False

Correct
Yes, weight sharing reduces significantly the number of parameters in a neural network, and sparsity of connections allows us to use a smaller number of inputs thus reducing even further the number of parameters.