## Congratulations! You passed!

**Grade received** 100% **Latest Submission Grade** 100% **To pass** 80% or higher

Go to next item

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

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 3 & 3 & 1 \\ -1 & -3 & -3 & -1 \\ 0 & -1 & -1 & 0 \end{bmatrix}$$

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

Expand

**Correct**
Correct. There is a high difference between the values in the top part from those in the bottom part of the matrix. When convolving this filter on a grayscale image, the horizontal edges will be detected.

**2.** Suppose your input is a 300 by 300 color (RGB) image, and you are not using a convolutional network. If the first hidden layer has 100 neurons, each one fully connected to the input, how many parameters does this hidden layer have (including the bias parameters)? **1 / 1 point**

- 9,000,001
- (selected) 27,000,100
- 9,000,100
- 27,000,001

Expand

**Correct**
Correct, the number of weights is $300 \times 300 \times 3 \times 100 = 27,000,000$, when you add the bias terms (one per neuron) you get $27,000,100$.

**3.** Suppose your input is a 256 by 256 color (RGB) image, and you use a convolutional layer with 128 filters that are each $7 \times 7$. How many parameters does this hidden layer have (including the bias parameters)? **1 / 1 point**

- (selected) 18944
- 1233125504
- 18816

- 6400

Expand

**Correct**
Yes, you have $7 \times 7 \times 3 + 1$ weights per filter with the bias. Given that you have 128 filters, you get $(7 \times 7 \times 3 + 1) \times 128 = 18944$.

**4.** You have an input volume that is $127 \times 127 \times 16$, and convolve it with 32 filters of $5 \times 5$, using a stride of 2 and no padding. What is the output volume? **1 / 1 point**

- $123 \times 123 \times 16$
- $62 \times 62 \times 32$
- $123 \times 123 \times 32$
- $62 \times 62 \times 16$

Expand

**Correct**
Correct, using the formula $n_H^{[l]} = \frac{n_H^{[l-1]} + 2 \times p - f}{s} + 1$ with $n_H^{[l-1]} = 127, p = 0, f = 5$, and $s = 2$ we get 62.

**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**

- 19x19x8
- 17x17x10
- 19x19x12
- 17x17x8

Expand

**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 a volume that is $121 \times 121 \times 32$, and convolve it with 32 filters of $5 \times 5$, and a stride of 1. You want to use a "same" convolution. What is the padding? **1 / 1 point**

- 0
- 5
- 3
- 2

Expand

**Correct**
Yes, when using a padding of 2 the output volume has $n_H = \frac{121-5+4}{1} + 1$.

**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**

- (•) $22 \times 22 \times 21$
- ( ) $66 \times 66 \times 7$
- ( ) $21 \times 21 \times 21$
- ( ) $22 \times 22 \times 7$

Expand

**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.** Which of the following are hyperparameters of the pooling layers? (Choose all that apply) **1 / 1 point**

- [ ] Average weights.
- [x] Whether it is max or average.

**Correct**
Yes, these are the two types of pooling discussed in the lectures, and choosing which to use is considered a hyperparameter.

- [ ] Number of filters.
- [x] Filter size.

**Correct**
Yes, although usually, we set $f = s$ this is one of the hyperparameters of a pooling layer.

Expand

**Correct**
Great, you got all the right answers.

**9.** Which of the following are true about convolutional layers? (Check all that apply) **1 / 1 point**

- [x] Convolutional layers provide sparsity of connections.

**Correct**
Yes, this happens since the next activation layer depends only on a small number of activations from the previous layer.

- [ ] It speeds up the training since we don't need to compute the gradient for convolutional layers.
- [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.

Expand

Correct
Great, you got all the right 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**

- False
- True

Expand

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.