

## Your grade: 100%

Your latest: 100% • Your highest: 100% • 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

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

- ☐ Detect 45-degree edges.
- ☒ Detect horizontal edges.
- ☐ Detecting image contrast.
- ☐ Detect vertical edges.

✓ **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)?

- ☒ 27,000,100
- ☐ 27,000,001
- ☐ 9,000,001
- ☐ 9,000,100

✓ **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)?

- ☐ 18816
- ☐ 1233125504
- ☐ 6400
- ☒ 18944

✓ **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?
- ☐  $123 \times 123 \times 16$
  - ☐  $123 \times 123 \times 32$
  - ☐  $62 \times 62 \times 16$
  - ☒  $62 \times 62 \times 32$

✔ Correct

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

5. You have an input volume that is  $15 \times 15 \times 8$ , and pad it using "pad=2". What is the dimension of the resulting volume (after padding)?
- ☐  $17 \times 17 \times 10$
  - ☐  $17 \times 17 \times 8$
  - ☒  $19 \times 19 \times 8$
  - ☐  $19 \times 19 \times 12$

✔ 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  $64 \times 64 \times 32$ , and convolve it with 40 filters of  $9 \times 9$ , and stride 1. You want to use a "same" convolution. What is the padding?
- ☐ 6
  - ☐ 0
  - ☒ 4
  - ☐ 8

✔ Correct

Yes, when using a padding of 4 the output volume has  $n_H = \frac{64 - 9 + 2 \times 4}{1} + 1$ .

7. You have an input volume that is 128x128x12, and apply max pooling with a stride of 4 and a filter size of 4. What is the output volume?

- ☐  $64 \times 64 \times 12$
- ☐  $128 \times 128 \times 3$
- ☒  $32 \times 32 \times 12$
- ☐  $32 \times 32 \times 3$

✔ Correct

Yes, using the formula  $n_H^{[l]} = \frac{n_H^{[l-1]} + 2 \times p - f}{s} + 1$  with  $p = 0$ ,  $f = 4$ ,  $s = 4$  and  $n_H^{[l-1]} = 32$ .

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

- ☐ 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 the benefits of using convolutional layers? (Check all that apply)

- ☐ It reduces the computations in backpropagation since we omit the convolutional layers in the process.
- ☒ It reduces the total number of parameters, thus reducing overfitting through parameter sharing.

✔ Correct

Yes, a convolutional layer uses parameters sharing and has usually a lot fewer parameters than a fully-connected layer.

- ☒ Convolutional layers are good at capturing translation invariance.

✔ Correct

Yes, this is due in part to applying the same filter all over the image.

10. In lecture we talked about “sparsity of connections” as a benefit of using convolutional layers. What does this mean?

- ☐ Regularization causes gradient descent to set many of the parameters to zero.
- ☐ Each filter is connected to every channel in the previous layer.
- ☐ Each layer in a convolutional network is connected only to two other layers
- ☒ Each activation in the next layer depends on only a small number of activations from the previous layer.

✔ Correct

Yes, each activation of the output volume is computed by multiplying the parameters from **only one filter** with a volumic slice of the input volume and then summing all these together.