Your grade: 100%

Next item →

Your latest: **92.50%** • Your highest: **100%** • To pass you need at least 80%. We keep your highest score.

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 horizontal edges.
- ( ) Detecting image contrast.
- ( ) Detect 45-degree edges.
- ( ) 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 128 by 128 grayscale image, and you are not using a convolutional network. If the first hidden layer has 256 neurons, each one fully connected to the input, how many parameters does this hidden layer have (including the bias parameters)?

1 / 1 point

- (●) 4194560
- ( ) 4194304
- ( ) 12582912

- ○ 12583168

Correct

Correct, the number of inputs for each unit is 128×128 since the input image is grayscale, so we need $128 \times 128 \times 256$ parameters for the weights and 256 parameters for the bias thus $128 \times 128 \times 256 + 256 = 4194560$.

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

1 / 1 point

- ○ 6400
- ◉ 18944
- ○ 1233125504
- ○ 18816

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 $121 \times 121 \times 16$, 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 61x61x32, and pad it using "pad=3". What is the dimension of the resulting volume (after padding)?

1 / 1 point

- (•) 67x67x32
- ( ) 64x64x35
- ( ) 64x64x32
- ( ) 61x61x35

Correct

Yes, if the padding is 3 you add 6 to the height dimension and 6 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

- (•) 3
- ( ) 2
- ( ) 1
- ( ) 7

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 32x32x16, and apply max pooling with a stride of 2 and a filter size of 2. What is the output volume?

1 / 1 point

- ○ 16x16x8
- ◉ 16x16x16
- ○ 32x32x8
- ○ 15x15x16

Correct
Correct, using the following formula: $n_H^{[l]} = \frac{n_H^{[l-1]}+2\times p-f}{s} + 1$

8. Which of the following are hyperparameters of the pooling layers? (Choose all that apply)

0.8 / 1 point

- [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.

- [x] Filter size.

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

- [ ] Average weights.

- [x] Number of filters.

This should not be selected
No, pooling layers keep the depth dimension of the volume, we don't need to specify the number of filters with a pooling layer.

9. In lecture we talked about “parameter sharing” as a benefit of using convolutional networks. Which of the following statements about parameter sharing in ConvNets are true? (Check all that apply) — 0.5 / 1 point

- [x] It allows parameters learned for one task to be shared even for a different task (transfer learning).

This should not be selected
No, transfer learning is not bound to ConvNets and can be used with other types of models as you've seen in Course 1-3.

- [ ] It reduces the total number of parameters, thus reducing overfitting.
- [ ] It allows gradient descent to set many of the parameters to zero, thus making the connections sparse.
- [x] It allows a feature detector to be used in multiple locations throughout the whole input image/input volume.

Correct
Yes, by sliding a filter of parameters over the entire input volume, we make sure a feature detector can be used in multiple locations.

10. In lecture we talked about “sparsity of connections” as a benefit of using convolutional layers. What does this mean? — 1 / 1 point

- ( ) Regularization causes gradient descent to set many of the parameters to zero.

This should not be selected
No, transfer learning is not bound to ConvNets and can be used with other types of models as you've seen in Course 1-3.

- [ ] It reduces the total number of parameters, thus reducing overfitting.
- [ ] It allows gradient descent to set many of the parameters to zero, thus making the connections sparse.
- [x] It allows a feature detector to be used in multiple locations throughout the whole input image/input volume.

Correct
Yes, by sliding a filter of parameters over the entire input volume, we make sure a feature detector can be used in multiple locations.

10. In lecture we talked about "sparsity of connections" as a benefit of using convolutional layers. What does this mean? 1 / 1 point

- ( ) Regularization causes gradient descent to set many of the parameters to zero.
- ( ) Each layer in a convolutional network is connected only to two other layers
- ( ) Each filter is connected to every channel in the previous layer.
- (•) 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.