The Basics of ConvNets

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 horizontal edges
- Detect 45 degree edges
- Detect image contrast
- Detect vertical edges

Expand

Correct

Correct! As you can see the difference between values from the left part and values from the right of this filter is high. When convolving this filter on a grayscale image, the vertical 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
- 27,000,001
- (selected) 27,000,100
- 9,000,100

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 grayscale image, and you use a convolutional layer with 128 filters that are each 3×3. How many parameters does this hidden layer have (including the bias parameters)? **1 / 1 point**

- 1152
- (selected) 1280
- 75497600
- 3584

Expand

Correct
Yes, since the input volume has only one channel each filter has $3 \times 3 + 1$ weights including the bias, thus the total is $(3 \times 3 + 1) \times 128$.

4. You have an input volume that is 63x63x16, and convolve it with 32 filters that are each 7x7, using a stride of 2 and no padding. What is the output volume? **1 / 1 point**

- 16x16x32
- 29x29x16
- (selected) 29x29x32
- 16x16x16

Expand

Correct
Yes, $\frac{63-7+0\times2}{2} + 1 = 29$ and the number of channels should match the number of filters.

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**

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

Expand

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**

- 1
- 2
- 3
- 7

Expand

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**

- 32x32x8
- 15x15x16
- 16x16x8
- 16x16x16 (selected)

Expand

Correct

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

8. Because pooling layers do not have parameters, they do not affect the backpropagation (derivatives) calculation. **1 / 1 point**

- True
- False (selected)

Expand

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) **1 / 1 point**

- [ ] It allows parameters learned for one task to be shared even for a different task (transfer learning).
- [ ] It reduces the computations in backpropagation since we omit the convolutional layers in the process.
- [x] 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.

- [x] Convolutional layers are good at capturing translation invariance.

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

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

- (•) True
- ( ) False

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.