Convolutional Neural Networks

Week 1: The Basics of ConvNets

1 What do you think applying this filter to a grayscale image will do? $\begin{bmatrix} 0 & 1 & -1 & 0 \\ 1 & 3 & -3 & -1 \\ 1 & 3 & -3 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}$ Answer:

Detect vertical edges.

Comment: 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)?

Answer: 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 7×7. How many parameters does this hidden layer have (including the bias parameters)?

Answer: 18944.

Comment: You have $7 \times 7 \times 3 + 1$ weights per filter with the bias. Given that you have 128 filters, you get $128(7 \times 7 \times 3 + 1)$.

4 You have an input volume that is $63 \times 63 \times 16$, and convolve it with 32 filters that are each 7×7, using a stride of 2 and no padding. What is the output volume?

Answer: $29 \times 29 \times 32$.

Comment: $(63 - 7 + 0 \times 2)/2 + 1$ and the number of channels should match the number of filters.

5 You have an input volume that is $61 \times 61 \times 32$, and pad it using pad=3. What is the dimension of the resulting volume (after padding)?

Answer: $67 \times 67 \times 32$.

Comment: If the padding is 3 you add 6 to the height dimension and 6 to the width dimension.

6 You have a volume that is $64 \times 64 \times 32$, and convolve it with 40 filters of 9×9, and stride 1. You want to use a same convolution. What is the padding?

Answer: 4.

Comment: When using a padding of 4 the output volume has $n_H = (64 - 9 + 2 \times 4)/1 + 1$.

7 Question 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?

Answer: $22 \times 22 \times 21$.

Comment: $n_H^{[l]} = (n_H^{[l-1]} + 2 \times p - f)/s + 1$ where p = 0, f = 3, s = 3, $n_H^{[l-1]} = 66$..

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

Answer: False.

Comment: 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 Version 1: Which of the following are true about convolutional layers? (Check all that apply)

Answer: It allows a feature detector to be used in multiple locations throughout the whole input volume.

Comment: Since convolution involves sliding the filter throughout the whole input volume the feature detector is computed over all the volume.

9 Version 2: In lecture we talked about parameter sharing as a benefit of using convolutional networks. Which of the following statement about parameter? (Check all that apply.)

Answer: It reduces the total number of parameters, thus reducing overfitting; It allows a feature detector to be used in multiple locations throughout the whole input image/input volume.

10 Version 1: 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?

Answer: True.

Comment: 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.

10 Version 2: In lecture we talked about sparsity of connections as a benefit of using convolutional layers. What does this mean?

Answer: Each activation in the next layer depends on only a small number of activations from the previous layer.