축하합니다! 통과하셨습니다!

받은 학점 100% 최신 제출물 학점 100% 통과 점수: 80% 이상

23h 54m 후에 과제를 다시 풀어보세요.

다음 항목으로 이동

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

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix}$$

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

더 보기

맞습니다
Correct. Notice that there is a high delta between the values in the top left part and the ones in the bottom right part. When convolving this filter on a grayscale image, the edges forming a 45-degree angle with the horizontal will be detected.

2. Suppose your input is a 128 by 128 color (RGB) image, and you are not using a convolutional network. If the first hidden layer has 64 neurons, each one fully connected to the input, how many parameters does this hidden layer have (including the bias parameters)? 1 / 1점

- 3145792
- 1048640
- 1048576
- 3145728

더 보기

맞습니다
Correct, the number of inputs for each unit is $128 \times 128 \times 3$ since the input image is RGB, so we need $128 \times 128 \times 3 \times 64$ parameters for the weights and 64 parameters for the bias parameters, thus $128 \times 128 \times 3 \times 64 + 64 = 3145792$.

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 / 1점

- 1233125504
- 18944
- 18816
- 6400

더 보기

맞습니다
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 / 1점

- $118 \times 118 \times 16$
- $118 \times 118 \times 32$
- $40 \times 40 \times 16$
- $40 \times 40 \times 32$

더 보기

맞습니다
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 / 1점

- 64x64x32
- 64x64x35
- 67x67x32
- 61x61x35

더 보기

맞습니다
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 / 1점

- 7
- 2
- 3
- 1

더 보기

맞습니다
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 66x66x21, and apply max pooling with a stride of 3 and a filter size of 3. What is the output volume? 1 / 1점

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

더 보기

맞습니다
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. Because pooling layers do not have parameters, they do not affect the backpropagation (derivatives) calculation. 1 / 1점

- ( ) True
- (●) False

더 보기

맞습니다
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 true about convolutional layers? (Check all that apply) 1 / 1점

- [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.

- [x] Convolutional layers provide sparsity of connections.

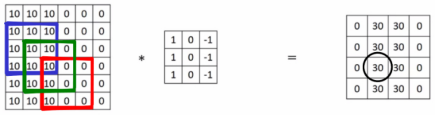
  ✓ **Correct**
  Yes. this happens since the next activation layer depends only on a small number of activations from the previous layer.

- [ ] It allows parameters learned for one task to be shared even for a different task (transfer learning).
- [ ] It speeds up the training since we don't need to compute the gradient for convolutional layers.

더 보기

맞습니다
Great, you got all the right answers.

10. The following image depicts the result of a convolution at the right when using a stride of 1 and the filter is shown right next. 1 / 1점



On which pixels does the circled pixel of the activation at the right depend?

- ( ) It depends on the pixels enclosed by the red square.
- ( ) It depends on the pixels enclosed by the blue square.
- ( ) It depends on all the pixels of the image on the left.
- (●) It depends on the pixels enclosed by the green square.

더 보기

맞습니다
Yes, this is the position of the filter when we move it two pixels down and one to the right.