PROBLEM 27

Consider the following 1D input and two convolutional filters:

1	2	-1	1	-3		1	0	-1		-1	1	-1

(a) Convolve the first filter with the input.

Solution: 2 1 2

(b) Convolve the second filter with the input.

Solution: 2 -4 5

(c) What would the output of the convolutional layer look like with both filters?

Solution: A 3x2 matrix with columns 2 1 2 and 2 -4 5.

(d) In the previous part, the spatial size of the output decreased from that of the original input -ie. it had fewer neurons. In some settings, we preserve the size by applying zero padding. What would the output of the convolutional layer look like in this case?

Solution: A 5x2 matrix with columns -2 2 1 2 1 and -1 2 -4 5 -4

(e) Continuing to use zero padding of size 1, we now use a stride length of 2. What is the output of the convolutional layer now? Compare how this relates to the case with no stride.

Solution: A 3x2 matrix with columns -2 1 1 and -1 -4 -4

This contains alternating weights from the output of the previous part.

(f) If our filters were size 5, rather than size 3, with stride 1, how much padding would you apply to maintain output size?

Solution: Pad each side with 2 zeros. In general, set to (F-1)/2.

(g) What is a general expression for the output size (number of neurons) in a convolutional layer, in terms of the filter size (F), stride length (S), amount of padding (P), and the input size (W)?

Solution: $(W - F + 2P)/S + 1$

(h) Consider a 10x10 grayscale image input (no RGB channel). If you made a 1-layer fully connected network, with 1 hidden, how many weight parameters would be required in the hidden layer including bias terms (ignore the output layer)?

> **Solution:** 100 weights for the first layer + 1 bias terms = 101

(i) Consider a 10x10 grayscale image input (no RGB channel). If you made a 2-layer fully connected network, with 10 and 5 hidden units respectively, how many weight parameters would be required in these hidden layers including bias terms (ignore the output layer)?

> **Solution:** 1000 weights for the first layer + 10 bias terms and 50 weights + 5 bias terms for the second layer

(j) With the same 10x10 input image, you now use a convolutional layer with F = 2 (square filters), S = 2 (in both dimensions), and P = 0. How is the output volume of the first convolutional layer with 2 filters, including bias terms?

> **Solution:** (10 - 2 + 2*0)/2 + 1 = 5 in each dimension. The convolutional layer has output volume 5x5x2.

(l) One advantage of convolutional networks is that the number of free parameters can be controlled by parameter sharing for a filter across spatial dimensions. Each filter is replicated across the entire visual field (image). That is, if a filter is used at one spatial coordinate (x_1, y_1), the same weights are also used in a different position (x_2, y_2). Adopting the parameter sharing scheme, how many parameters does the convolutional layer have?

> **Solution:** With parameter sharing, we effectively don't have to have different weights across the 5x5 dimension. So rather than 5x5x2x5 = 250 parameters, we have $2 * 5 = 10$ parameters.

Additional Explanations

The solutions assume we are not zero-padding the input, so the output is smaller size than the input. Note this is what happens when you use padding=valid rather than padding=same in Keras.

Technically, a convolution requires first flipping the filter, but in CNNs we typically don't worry about doing this. Some of you may have flipped the filter and thus gotten different answers, which is okay. On an exam we would be very clear about our expectations regarding this.

a) First value = $(1*1)+(2*0)+(-1*-1) = 2$. Second value = $(2*1)+(-1*0)+(1*-1) = 1$. Third value = $(-1*1)+(1*0)+(-3*-1) = 2$.

b) First value = $(1*-1)+(2*1)+(-1*-1) = 2$. Second value = $(2*-1)+(-1*1)+(1*-1)=-4$. Third value=$(-1*-1)+(1*1)+(-3*-1) = 5$.

g) Our effective input length is $W + 2P$ when we take into account the padding (which adds P zeros on each side of the input). If we had a stride of 1, let's think about what our output length would be. Output length is just the number of places across which we drag the filter. If we drag a length-3 filter across $W + 2P = 5$ values, we get output length 3. If we drag a length-3 filter across $W + 2P = 7$ values, we get output length 5. From this we can deduce the pattern that the output length is $W + 2P - F + 1$. Working out some examples where stride is greater than 1 shows how to incorporate the stride into this expression, and we get $(W + 2P - F)/S + 1$.

i) The answer is,thus,1000+10+50+5=1065.