COMP 474/6741 Intelligent Systems (Winter 2024)

Worksheet #9: Introduction to Deep Learning

Task 1. Let's compute the *cross-entropy loss* for a binary classification task, where a neural network has to classify an image into *hot dog* (label 1) vs. *not hot-dog* (label 0), using the loss function

$$H(y,p) = -\left(y\ln(p) + (1-y)\ln(1-p)\right)$$

Assume for a given input image, we obtained an output value p (using the sigmoid activation function) of 0.84. The expected output y for this image is 1. Compute the cross-entropy loss in this case: .....

What's the loss if the network had predicted 0.2 instead (same image, expected result is still 1): .....

Task 2. Ok, let's now compute the *categorical cross-entropy loss* for a classification task where a network has to distinguish between three classes: *cat*, *dog*, and *hot-dog*. The network outputs a probability distribution across these classes for a given image, using the *softmax* activation function in the output layer. The loss function is defined as:

$$L(y,p) = -\sum_{i=1}^{N} y_i \ln(p_i)$$

For a given image, the network predicted the following probabilities: *cat* = 0.2, *dog* = 0.5, *hot-dog* = 0.3. The true class of the image is *dog*, represented as a one-hot encoded vector $y = [0,1,0]$.

Calculate the categorical cross-entropy loss for this scenario: .....

Task 3. Consider the following matrix that represents an image. This image will be fed into a convolutional neural network (CNN):

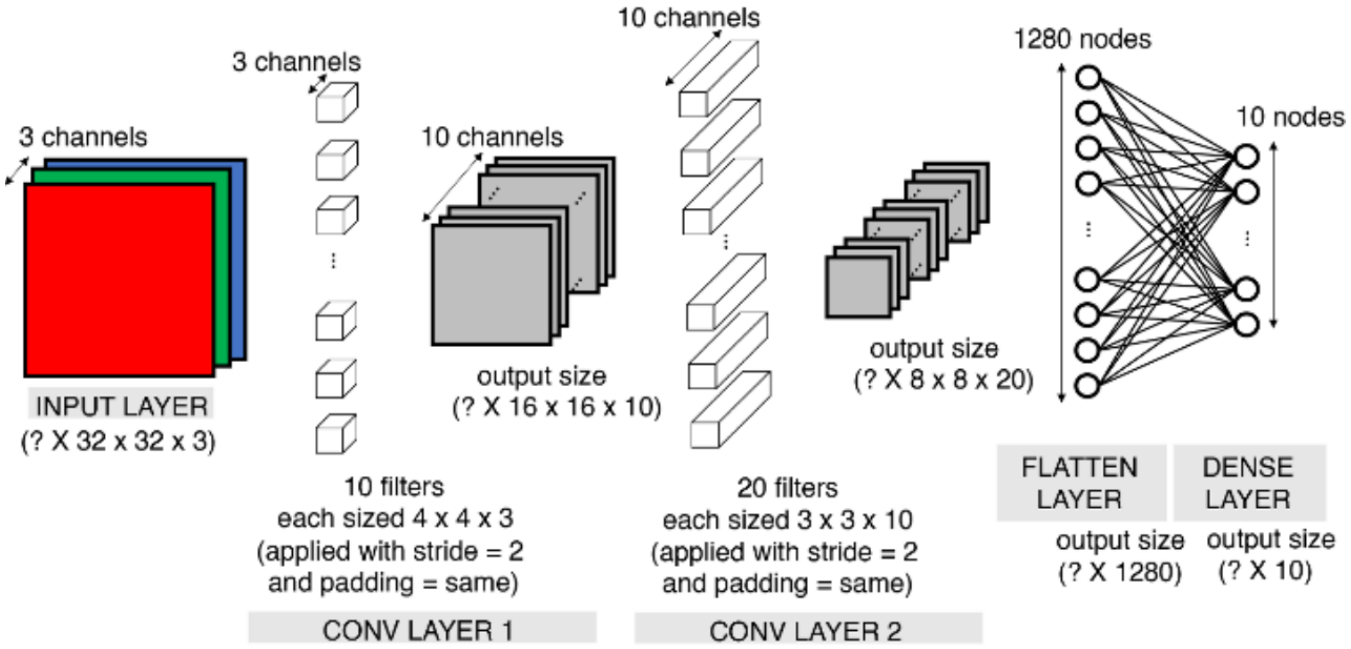
1	1	2	2	2	0	0
2	0	1	1	2	1	2
0	1	0	0	1	1	2
0	2	1	2	0	2	2
1	2	0	0	1	0	1
0	0	0	0	1	2	1
2	0	0	0	2	1	1

Assume that we use the following convolution filter with a stride of 2 (no padding):

0	1	1
0	1	0
0	-1	-1

What will be the size of the activation map? .....

What will be the resulting activation map?



Task 4. The task here is to understand the structure of the CNN (shown at the top) we are building: Our first convolution layer has 10 filters, each sized $4\times4\times3$ (`kernel_size = (4,4)`), thus $(4\times4\times3+1)\times10 = 490$ weights (parameters to train).

How do we obtain the output shape of this layer? The general formula you can use is (for `padding = "same"`, meaning the size of the kernel is the same as the input, padded with zeros):[1]

$$\text{output\_shape} = \left(\text{None}, \frac{\text{input\_height}}{\text{stride}}, \frac{\text{input\_width}}{\text{stride}}, \text{filters}\right)$$

Now compute the output tensor shape of the first convolution layer using the formula above:
output_shape = ...

In the second convolution layer, we want to apply 20 filters of size 3×3 and a depth of 10.
How many weights do we have to train? .....

Compute again the output tensor shape using the formula above:
output_shape = ...

Task 5. What will be the output of a pooling layer with a size of 2×2 and a stride of 1, on the activation map of Task 3 above, if we use the following strategies:

1. Average pooling:

2. Max pooling:

[1] In a Keras model input shape, "None" represents a flexible batch size dimension, allowing for input batches of any size.