67750: Advanced Practical Machine Learning	2019/20

Solution 3 — Convnets Theoretical Solutions

Or Sharir *TA: Omri Bloch*

1 Theoretical Questions

1.1 Parameterized ReLU

Define the following function:

$$f_i(o;t) = \max\{t, o_i\}$$

Will the incorporation of this function into a network define a larger hypothesis class as opposed to simply using ReLU activations? explain your answer and/or give an example.

Solution

This question wasn't defined properly and some of you didn't understand it, so we won't be deducting points for it. The idea was that t is shared along all of the neurons, and that we replace the ReLU activations in a network by this activation function and then we ask whether the new network is more expressive.

In this case, we can express the new network using a ReLU network by playing around with the biases. For a layer l, we need to subtract t to all of the bias weights so that the output of the network is equivalent to the new network (only shifted by t). Next, to make sure the next layer is still equivalent, we need to change the biases of the next layer (layer $l+1$) - this can be done using the following formula:

$$b_i \leftarrow b_i + t \sum_j W_{i,j}$$

The above process can be done repeatedly for all of the layers, where we don't subtract t from the bias in the final, and this gives us an equivalent ReLU network for every network with the new activation, so it doesn't increase our expressiveness.

1.2 Sigmoid Derivative

So far we talked mainly about the ReLU activation function, but another activation function which used to be very popular is the sigmoid function:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

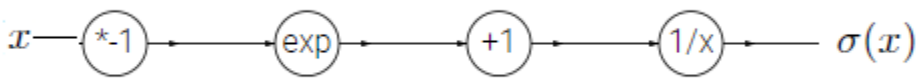
We saw in the recitation how we can easily calculate the derivative of the ReLU function by remembering which neuron had a positive activation, but calculating the derivative of the sigmoid function is potentially less pleasant.

Write the sigmoid activation as a computational graph with an input x and an output $\sigma(x)$.

Taking the gradient of the sigmoid function means back-propagating through quite a few nodes in the computational graph. Derive the gradient of the sigmoid function and show that it can be expressed using the sigmoid function itself. This will show we can calculate the sigmoid derivative easily by remembering the activation value during the forward pass.

Solution

First, we can write the computational graph for the sigmoid function in the following way:



This is quite a long computational graph for a function that will appear in every layer of our network, so we should want to simplify it by using values from the forward pass... However, we can express the gradient as a function of the sigmoid itself, which means we can just use the value we saved in the forward pass:

$$\frac{\partial \sigma(x)}{\partial x} = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{-1+1+e^{-x}}{(1+e^{-x})^2} = -\frac{1}{(1+e^{-x})^2} + \frac{1}{1+e^{-x}} = \sigma(x) - \sigma(x)^2$$

$$\frac{\partial \sigma(x)}{\partial x} = \sigma(x)(1-\sigma(x))$$