CSE 446: Machine Learning **Lecture**

Convolutions and convolutional neural nets...

Instructor: Sham Kakade

1 A Convolution

A convolution is a basic idea in signal processing. Let us review the 1-dimensional version. Suppose we have a one dimensional function $\mathcal{I}(t)$ and filter $f(t)$; (we can think of $\mathcal{I}$ as the image though it easier to present the definition when we think of the image as a one dimensional function). The convolution of $\mathcal{I}$ with f is defined as:

$$(\mathcal{I} \star f)(t) = \int_{\tau} \mathcal{I}(\tau) f(t-\tau) d\tau$$

It is helpful to think about what "low pass" and a "high pass" filter is.

2 A one hidden layer, convolutional neural network

2.1 the black and white case

Let us give an example on a convnet, say for mnist. Suppose that:

- image x is size 28×28
- suppose we have 64 filters of size 8×8. So an array of size $8 \times 8 \times 64$.
- we can view these filters as the parameters of the first layer.
- we also have second layer weights

Given the parameters, let us describe the forward pass, e.g. how to obtain $\hat{y}(x)$:

1. convolve the filter with the image. this will result in an image of size $28 \times 28 \times 64$ or $7 \times 7 \times 64$ (if we used a stride of 4). we say the image now has 64 *channels*.
   (a) this implicitly convolves each of the 64 filters with the image.
   (b) this also 'pads' the image with zeros so that you get back a 28×28 image for each convolution.
   (c) you could also use a stride of 4, i.e. only apply the filter with a center of every other pixel, to get an image of size 7×7.
   (d) you can do this with a batch of images as well
2. Then you can apply the relu to the image, still with a $7 \times 7 \times 64$ image.

3. now can 'pool' more to bring things down to $2 \times 2 \times 64$
   (a) we can 'average' or take the 'max' in subregions.
   (b) you could use regions, the pool size, of 4×4
   (c) you could strides this over by 4
   (d) with padding, this would give you a 2×2 size image (for each 'channel').
4. for the next 10 output nodes, we can treak the image like a just a layer with dimension $2 * 2 * 64$ nodes. and then the activations are just linear
5. we could then take the ReLu to get out 10 outputs.

2.2 the color case (which is also the multi-layer case!

Note that the first hidden layer has 64 channels. How might we add in another layer? To address this issues, you must understand the case where we start with a color image of size $28 \times 28 \times 3$.

Read the ConvNet links from the webpage. The extension to the case of colors is not difficult if one understand the black and white case. And if you understand how to deal with colors, then you understand the multi-layer case1

3 Weight sharing

Do you see why we can view a CNN as a particular neural network that 'shares' weights across nodes? Do you see why this is a powerful idea for statistical reasons? (i.e. for generalization?)