BRIEFLY: Convolutional Neural Networks & The VC-Dimension

This post was made to provide some brief answers to questions below.

1. What are convolutional neural networks (or CNNs)?
2. Why do they achieve their results?
3. What are they good for?

1. What are convolutional neural networks (or CNNs)?

Convolutional neural networks are a type of deep learning network. (Already lost? Hopefully you can learn something from Part 1, but Part 2 is quite technical. Section 3 has some cool applications. Enjoy!) CNNs are starkly different from the more traditional fully connected neural nets (or Feed-Forward Neural Nets). CNNs are able to capture spatial data and temporal data, but they aggregate this data in an advantageous way.

Think about the pixels on your computer screen. Looking at only one pixel and trying to figure out how it relates to other pixels sounds really hard, and it is hard – even for computers! Maybe if we looked at all the pixels individually, but related them to each other in some way, then looking at one or two pixels individually can tell us about what is on the screen? Not really.

Trying to weigh the connected web of all the pixels on your screen is an extremely simplified, and personified, version of how fully connected neural nets try to learn image data. One may find patterns but it's clear that information about one pixel can not tell us too much about other pixels.This is why feed-forward neural nets do not perform well on image data.

Now imagine you can look at multiple pixels at once. The information of multiple pixels can be convoluted to tell us something about an area of the screen. Once we have a good understanding for an area or two, understanding what is happening on the screen might be plausible.

This convolution of image data reduces the number of pixels (or parameters) we have to think about. Yet, each convolution returns meaningful data.

By reducing the number of parameters into a smaller, more meaningful, set of parameters we can focus on learning what is important for deciphering the image. This is how CNNs get their success.

This automated image convolution was inspired by biological processes! When us humans see a bike we know it's a bike whether it is laying on the ground, moving through the street, sitting a dumpster, or falling from a tall building. This is because our eyes and brains just piece all the visual information about the bike together (wheels, material, handlebars, etc.), telling us it is a bike. These CNNs look at grouped data as well. Now, theoretically, when the CNN notices the wheels, handlebars, etc., it knows the image is a bike (even if the bike is doing something strange like falling from a building).

2. Why do they achieve their results?

Why do CNNs achieve good results? The short answer is CNNs have a lower VC-Dimension than fully-connected neural nets, so they are less prone to overfitting, and hence return better results on the test set.

Why exactly do CNNs overfit less? What are the details?

To answer this question, let's recall an important fact:

If $L_D(h_s)$ is the true error of our distribution, and $L_D(h^*)$ is the smallest error possible on the training set, then $L_D(h_s) - L_D(h^*) \leq O(\sqrt{\frac{VCDim(\mathcal{H})}{m}})$
*I know this fact courtesy of Mike Izbicky's Data Mining class at Claremont McKenna College.

So, if we decrease the VCDim(H) then we have decreased the difference between the true error and the training set error.

Notice when the VCDim(H) is large, we can not get close to matching the true error because our test error is large.

In general, it is known the VC-Dimension for binary classifier class is $\leq O(d^2 * E^2)$ where d is the total number of layers and E is the total number parameters in the network (or edges). For deep learning, this value $d^2 * E^2$ gets very large, very quickly, as the number of layers increases. Remember each neuron is connected to every other layer's neuron in fully connected neural nets.

In convolution neural nets, E is restricted based on the filter bank sublayer. This filter takes in the layer with dimension n x p, filters it with convolutions, and outputs a layer n' x p' where n' $<$ n, p' $<$ p. One can define q(d) to be the complexity of computing this restriction for every layer on the filtering choices, and the input layers. Generally, $q(d) << E^2$.

"On the Size of Convolutional Neural Networks and Generalization Performance" by Kabkab et. al. defines q(d) more formally.

"On the Size of Convolutional Neural Networks and Generalization Performance" by Kabkab et. al. also creates a bound for VCDim(H) when H is a CNN binary classifier:

$VCDim(H) \leq (d^2 * q(d)^2)$

Notice $q(d)^2 << E^2 \implies (d^2 * q(d)^2) < d^2 * E^2 \implies O(\sqrt{\frac{VCDim(\mathcal{H})}{m}})$ decreases in the CNN deep learning implementation.

Finally, we have a better understanding of how CNNs change their VC-Dimensions to reduce overfitting.

3. What are CNNs good for?

Inspired by biological image processing, it makes sense CNNs are a go-to network for image processing. There are other good uses for CNNs too.

Here are three cool results from CNNs:

Diagnosing Skin Caner with CNNs: https://www.nature.com/articles/nature21056

Computers Better Than Humans?: https://www.forbes.com/sites/michaelthomsen/2015/02/19/microsofts-deep-learning-project-outperforms-humans-in-image-recognition/?sh=2c3e3672740b

Weather Prediction: https://www.nature.com/articles/s41598-020-57897-9

It should be noted that CNNs only achieve these great results when there is plenty of training data, and one can wait for the CNN computation. But all deep learning networks require a lot of data and computation time to get good results, so the statement above is relative.

References:

- https://towardsdatascience.com/convolutional-neural-networks-the-biologically-inspired-model-f2d23a301f71
- "Understanding Machine Learning: From Theory to Algorithms," 2014 by Shai Shalev-Shwartz and Shai Ben-David
- https://ttic.uchicago.edu/ tewari/lectures/lecture12.pdf
- http://www.sontaglab.org/FTPDIR/vc-expo.pdf
- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53