ECE472 – Quiz 5

Jonathan Lam

October 7, 2020

1. *Describe one of the experiments in "Rethinking Generalization" and what the implications were.* (arXiv:1611.03530)

   The first experiment the authors describe is using standard neural network architectures to fit random labels. What they find is somewhat surprising: these networks, which work so well to classify images, can also classify both randomly-generated labels or randomly-generated images (in both cases, the labels are independent of the features) with high accuracy almost as if they were classifying the original image data (i.e., convergence is only slower by a linear factor). This is not surprising given the promising capability these networks have shown recently, but it is a problem when it comes to thinking about generalization: if it is capable enough to pick up patterns so well out of nonsensical data, but the same network also works well on true data, how do we truly know how well our model generalizes? In one case, the model completely overfits (because there is no relationship when the features or labels are randomized), but in another case, the same exact model works well.

   In other words, it is hard to tell when a model overfits and when it does not, because the same model is capable of both. The authors go on to describe that while the common regularization techniques used (e.g., L_2, L_1, dropout, batch-norm) often aid generalization, this likely makes up a small part of how well a model generalizes, and there is likely a lot still unknown about the exact nature of generalization.

2. *Compare and contrast Squeezenet with MobileNets.* (arXiv:1602.07360, arXiv:1704.04861)

   The motivation for both is roughly the same: smaller neural networks (in terms of memory) would likely be less computationally expensive to train and use, be easier to distill to other systems, and be smaller in size to transmit (e.g., in the case of software updates). Squeezenet focused more on the memory footprint, whereas MobileNets (which came a little later) was worried both about small network memory size and about minimizing the computational cost of the neural network during inference time. Both are also focused around the typical convolutional neural network architecture (e.g., AlexNet).

   Squeezenet took into account a few basic principles to try to reduce model parameters, namely by using smaller convolutional filters (mostly 1×1 filters and some 3×3 filters) and a small number of channels (i.e., size of the output of a convolutional layer). They messed around with the percentage of 3x3 filters and additional compression to achieve AlexNet-like accuracy with 50x fewer parameters. They were not, however, aiming to reduce computational cost like the MobileNet authors were, as this is usually a less strict constraint (memory is strictly limited by memory size, but computational cost is measured in time and is more flexible).

MobileNet, on the other hand, changed the nature of the convolutional layers themselves. Usually, the number of parameters for a convolutional layers is the product of the size of the filter, the number of input channels, and the number of output parameters, which can quickly add up. For a given convolutional layer with M input channels and N output channels and a filter with size D, this means MND parameters. A typical convolutional layer also involves convolving M different filters (one for each input channel) for each of the N output channels and then taking a linear combination, which can be very expensive. MobileNets reduces the computations necessary by "factoring" the convolutional layer into two steps: generate only one set of M different filters (one for each of the input channels), and then use a regular 1×1 convolutional layer to take the linear combination of the filters. This has two effects in reducing computation (decreasing the number of filters by N and decreasing filter size by a factor of D), and the authors only empirically experienced a trivial decrease in accuracy. The authors also provided two hyperparameters as a mechanism for a user to customize this network to meet more specific resource constraints. This caused MobileNet to outperform (in accuracy and in speed) Squeezenet when it had a similar number of parameters.

3. *What do you think of DenseNets?* (arXiv:1608.06993)

   I'm pretty surprised at their results – they exceed my expectations of what feels to me like a ResNet implementation with more skip connections. They more or less took the idea of ResNets and put it on steroids, along with some measures to make sure that the number of parameters doesn't explode (i.e., adding bottlenecking layers and constraining the dense connections to three smaller blocks to prevent too many skip connections), and it seems to outperform pretty much all of the other methods without needing to be too deep or have too many parameters (i.e., a 100-layer DenseNet outperforms and has fewer parameters than a 1000-layer ResNet).

   In one sense, this can be thought of as a continuation of "Identity Mappings in Deep Neural Networks" (arXiv:1603.05027), which explored the role of identity activations so that the gradient wouldn't be obfuscated as it traveled to lower layers (i.e., this architecture had the nice property that the overall gradient would be the sum of the gradients of each ResNet block). Here, they take it a step further by actually passing through earlier feature maps, not obfuscated by the earlier layers at all, which reinforces the aforementioned paper's claim that direct passthrough of information from earlier layers as skip connections work very well. I think it might be very interesting to look into other possible ways to similar self-reinforcing methods.