Understanding Deep Learning Requires Rethinking Generalization: A Review

Ashutosh Jha (ME14B148)

March 2017

1 Introduction

Deep Learning has changed the way people look at Machine Learning however, a theoretical understanding of why is it that Deep Neural Networks actually work is still lacking.
Well designed ANN's show really small generalization error however you could always make a neural network really small and its generalization error would shoot up. So model architecture is something which obviously determines generalization of a model.
This paper sheds some light on how the traditional statistical theory understanding of "generalization" fail to work in the case of Deep Neural Nets.

2 The Funda

Traditional Statistical Learning has defined complexity measures which it believes are capable of controlling generalization errors. The paper shows that these methods are incapable of differentiating ANN's which generalize well versus those that don't.
How they do this is by randomly shuffling the original data in many different ways and show that Deep Neural Networks are essentially able to fit those easily giving 0 training error. On the other hand the test error would essentially be no better than random because the network hasn't learned anything meaningful. **As a result, the generalization error of a given architecture can be made to shoot up by just shuffling the labels.**This essentially shows that traditional methods for measuring generalization like Rademacher Complexity which depend on the model fitting random variables are **useless** when used in the Deep Learning framework.

3 Too Much? The 3 line summary

The authors themselves summarize the entire set of contributions in 3 lines (which is kind of really cool. authors rarely do such things for rookies):

- Deep neural networks easily fit random labels.
- Explicit regularization may improve generalization performance, but is neither necessary nor by itself sufficient for controlling generalization error.
- There exists a two-layer neural network with ReLU activations and 2n + d weights that can represent any function on a sample of size n in d dimensions.

4 The Devil in the Details

4.1 Effective Capacity of Neural Networks

The authors perform experiments on the effects of different levels of randomization on CIFAR-10 and ImageNet. They use 3 model architectures: AlexNet, Inception V3, MLP (1 hidden layer with 512 neurons).
The different levels of random shuffling used were:

- True Labels: Original Dataset
- Partially Corrupted Class: For each image with a fixed probability p, the label is set to any of labels with uniform probability.
- Random Labels: All labels are randomly shuffled.
- Shuffled Pixels: A random permutation of all the pixels is fixed. This permutation is now applied to all the images with labels kept intact.
- Random Pixels: A different random permutation is applied to each image independantly.
- Gaussian: A gaussian with mean and variance matching with original image is used to generate random pixels for each image.

The authors report that the ANN's are able to still fit the data perfectly (˜0 training error) by using SGD with **unchanged** hyperparameters. This implies that DNN's have the capacity to memorize the entire training set.

This essentially flies in the face of applying existing methods to realize generalization in the Deep Learning framework because:

- The empirical Rademacher complexity of a hypothesis class H on a dataset $\{x_1, \cdots, x_n\}$ is defined as the eqn. 1 in the paper. The idea is that it measures the ability of a model class (could be ANN,SVM anything) to fit random label assignments. Since it is empirically seen that DNN can fit any random labelling, R(H) = 1. Thus, you cannot use Rademacher complexity to differentiate well generalizing models to those not generalizing well.

- The VC Dimension of a classification algo is the cardinality of the largest set of points that the algorithm can shatter. This is again ruled out as the paper shows empirically that a DNN can shatter quite large datasets.
- Uniform Stability of an algorithm A measures how sensitive A is to replacement of a single training example. This is independent of the labelling of the data. Thus, it cannot be used to distinguish a model trained on true data to a model trained on randomly shuffled data.

4.2 The Role of Regularization

The authors consider 3 types of explicit regularization: data augmentation, L2 Weight Decay and Dropout along with implicit regularization techniques such as Batch Normalization and Early Stopping.
The experiments by authors try to include each regularization technique one at a time and try out different combinations of regularizers. The experiments suggest that data augmentation is the strongest type of regularizer and that regularization in general helps reducing generalization error. The authors however note that by just changing model architecture (w/o regularization), same generalization as of old model along with regularization can be acheived. This shows that while regularization certainly helps in improving generalization there is something more fundamental(probably in the model architecture) that is the cause of generalization.

4.3 Finite Sample Complexity

The universal approximation theorem already states that MLP's of 1 hidden layer have the capacity to express any arbitrary function. So the first thing that crossed my mind is why put this section? The point is that they assume that the amount of training data available is infinite. Also more importantly just because ANN's can express any function doesn't mean a training algo can find it. [The 2nd point remains a problem even after this paper. There are no theoretical guarantees that an algorithm will necessarily find the solution by UAT]
The authors show in the paper that there exists a two-layer neural network with ReLU activations and 2n + d weights that can represent any function on a sample of size n in d dimensions and they give expressions for weights which will perfectly fit the training data.
The way they do it is pretty simple math. Given a sample, they essentially find the expressions for the weight matrices. They do this by first defining a 2 layer MLP and then show that the relations obtained for weights can be written as a c = Aw where A is full rank.

The authors also show that the results hold for a MLP with more than two layers where the funda is that at every layer you partition the activations into disjoint intervals and apply the original proof for each interval. (Math oh Math! Why art thou so painful?)

4.4 Implicit Regularization: An Appeal to Linear Models

The authors show in this part of the paper that SGD by itself is an implicit regularization method. The idea is that if you start SGD with $w_0 = 0$ (which by itself is bugging me a lot. You never initialize the weights to 0. Its always close to 0 and distinct to avoid co-adaptation), we can show that the weights obtained by SGD is given by $w = X^T \alpha$ and $XX^T \alpha = y$.
It can be shown that the solution to $Xw = y$ using the kernel trick(the second equation) would essentially give minimum L2 norm solution.
The authors however do note that this notion of min norm is not related to generalization performance as the norm of the solutions with better test performance empirically had larger norms.

5 Drawbacks and Directions for Future Research

The paper presents a negative take on the existing ways of looking at generalization showing how it is wrong. It however offers no alternatives to the present methods. Like the title of the paper suggests, we need to rethink how we define generalization.

Another important point that this paper presents is that Deep Neural Nets have the capacity to memorize the entire training data. But is this truly what a deep neural net does? Turns out it is not. An ICLR 2017 workshop submission (https://openreview.net/pdf?id=rJv6ZgHYg) infact extends the rethinking generalization paper to show that even though the network does have the capacity to memorize, it doesn't actually memorize the data.

Another important direction for research which the authors kept implicitly suggesting was the role of the model architecture in generalization. Finding reasons to why certain models generalize better than others is a clear direction for further research.