Research Plan: A Complete view of Neural Network Representation Capabilities

Introduction

Neural networks are a function approximation tool based on running data through a network of nodes that perform simple computations to output a desired result. These networks are a widely applied technique in Machine Learning, but their capabilities are not well understood. We know, for instance, that neural networks with only a few layers can approximate any continuous function [Hornik et al., 1989] and that there are some functions that can be easily represented with three layer networks cannot easily be represented with only two [Eldan and Shamir, 2016]. However, these results do not describe fully the spaces of functions that can be written as a neural network of a particular size and depth. This is the main thrust of my research plan — to give a thorough description of the abilities of neural networks of certain sizes to represent functions. This question is of the utmost importance; one well-known principle about learning systems is that less complex models generalize better to new data. Therefore, if we want our neural networks to be able to accurately respond to new situations, it is important to understand how small these networks can be made. Especially in use cases such as self-driving cars or medical diagnosis, where a wrong answer can mean life-or-death, it is critical that we be able to guarantee that our algorithms are highly accurate.

Networks with One Nonlinear Layer

The first step in my plan is to give a complete picture of the basic unit of a neural network: a network with a single nonlinear layer. All neural networks, including the "deep networks" that are widely applied in many areas, are composed of many of these single layers. Each layer is composed of a linear transformation of the data from the previous layer, followed by a nonlinear "activation". This is done again and again, with each layer distilling more information about the data. The nonlinear transformation is what makes it possible for neural networks to be so powerful; without these nonlinearities, the network would only be able to tell apart two data sets if they were linearly separable (that is, only if a simple dividing line could be drawn to separate them). These nonlinear layers seem to be the essence of neural networks, which is why I want to study them in more detail. The goal of this first step will be to produce an explicit mathematical formulation of the complexity of representing functions in this single layer framework. More specifically, I will develop techniques for constructing one-layer approximations of functions and corresponding lower bounds, based on factors like the desired accuracy of the network, the metric in which that accuracy is measured, and bounds on the weights within the network. I will also consider additional assumptions on the structure of these functions, such as radial symmetry which preliminary work was shown may be useful in improving the bounds.

Multilayer Networks

Once I have a clear understanding of one layer networks, I will approach the problem of representation for multilayer networks. One established fact from this area of study is that neural networks with only l layers cannot represent the same set of functions as those with l^2 layers without an exponential blowup in width [Telgarsky, 2016]. One key question is to determine whether this l vs l^2 separation can be improved to kl for some constant k, or perhaps even $l + k$. Results about these cases are still unproven, but all previous work in the area seems to suggest that the power of

a network is related to the quantity w^l, where w is the width and l the number of layers. The goal of this step will be to rigorously show that there are containments between different classes based on this measure of complexity, or else to point out exceptions to this rule.

Training

So far, my goals in research have to do with clear view of the representation power of general neural networks — I only focus on what is possible in a general sense, considering all possible parameter configurations for the network. In the real world, parameters are found by training procedures that aren't always perfect. I would like to specifically examine neural networks that are produced using standard training methods such as gradient descent (which is by far the most popular training method). Gradient descent is a technique for optimizing the parameters of a neural network by incrementally changing those parameters. It is not clear, even if neural networks have a great degree of power, whether gradient descent can find the weights that optimally solve a given function approximation problem. What we know is that the optimization landscape for neural networks is very complex, and we are only guaranteed to find a local optimum when we do gradient descent, rather than the configuration of the network which is truly the best. But if optimizing neural networks is so hard, why do they work in practice at all? I feel that there must be some underlying explanation for the good performance of gradient descent as a technique. The question of why neural networks succeed in training is discussed in Frankle and Carbin [2018], which posits a theory — that structures existing in the random initialization of networks allow training to succeed. However, this may only be true for highly overparametrized networks. Giving rigorous constraints on the types of networks for which this is true is the goal of this step.

References

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *COLT*, 2016.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Training pruned neural networks. *CoRR*, abs/1803.03635, 2018. URL `http://arxiv.org/abs/1803.03635`.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, july 1989.

Matus Telgarsky. benefits of depth in neural networks. In *29th Annual Conference on Learning Theory*, volume 49, pages 1517–1539, 2016.