I will discuss the extent to which learning theory, as we know it today, can explain modern machine learning practice.

The starting point is that linear function classes are the main tool that learning theory currently has for providing guarantees on learning. Indeed, most known efficiently learnable function classes are either linear or contained in a linear class. Furthermore, the picture arising from various hardness results, is that it is hard to learn non-linear function classes.

In light of that, I will present recent work, which associates a linear function class to a network architecture, and that shows that modern neural network (NN) algorithms are guaranteed to learn a function that is at least as good as the best function in that class. This result provides the first polynomial time and distribution-free guarantees on modern NN learning algorithms, and applies to a relatively rich family of network architectures.

Yet, our results are still far from constituting a satisfying explanation to the success of NN, as they are quantitatively weak for the regime of parameters that is used in practice. I will end the talk with a short discussion on how better understanding could possibly be achieved.

Based on joint work with Roy Frostig, Vineet Gupta and Yoram Singer.

1 Learning theory in the age of neural networks

We want to learn $h^* : X \to Y$. Assume the input space is $X = \{\pm 1\}^n$ and the output space is $Y = \{\pm 1\}$. We have a data source $(x, h^*(x))$, $x \sim D$ where D, h^* are unknown.

The goal is to find $h : X_n \to \{\pm 1\}$ with small 0-1 loss

$$L_D^{0-1}(h) = \mathbb{P}_{x\sim D}(h(x) \neq h^*(x)).$$

A network N is a DAG with n inputs and 1 output. Each non-input v is labeled by an activation function $\sigma_v : \mathbb{R} \to \mathbb{R}$. Assume $\mathbb{E}_{X\sim N(0,1)} \sigma^2(X) = 1$.

Sample weights $w_{uv} \sim N(0, \frac{1}{\deg^+(v)})$ (indegree). This is Javier initialization.

If you fix 1 input example, and look at a particular neuron, this initialization makes sure that the L^2 norm of the random variable is 1.

Define

$$L_D(w) = \mathop{\mathbb{E}}_{x\sim D} l(h^*(x)h_w(x))$$

for $l(z) = \ln(1 + e^{-z})$. Minimize $L_D(w)$ using SGD:

$$w_{t+1} = w_t - \eta L_S(w_t) \tag{1}$$

$$L_S(w) = \frac{1}{m}\sum_{i=1}^{m} l(y_i h_w(x_i)). \tag{2}$$

How do we analyze learning algorithms in general? It is impossible to efficiently learn any function $h^* : X \to Y$, but it may be possible under conditions on h^*, D. We want conditions that are clean and intuitive and valid in many real-world scenarios.

In PAC learning assume $h^* \in H$. This is valid in many cases when H is large enough.

What hypothesis classes are learnable? For $\Psi : X_n \to \mathbb{R}^k$,

$$\text{Linear}(\Psi) = \left\{\text{sign} \circ h_w \circ \Psi : h_w(x) = \langle w, x\rangle \text{ for } w \in \mathbb{R}^k\right\}.$$

Large margin linear classes are learnable:

$$\text{Margin}(\Psi, M) := \left\{\overline{\text{sign}} \circ h_w \circ \Psi : \|w\| \le M\right\}.$$

$\mathbb{F}_q$-linear classes are learnable (using Gaussian elimination).

Beyond linear, with a few exceptions, virtually all known learnable classes are linear or contained in learnable linear class. Exception: decision lists with parity leaves. Open: find more exceptions!

In the statistical queries model, all known learnable classes are linear or contained in learnable linear class. Open: find exceptions or prove that no exceptions exist.

Very simple nonlinear classes are hard to learn. Hardness implies hardness of predicting better than random (boosting).

How to analyze neural networks? One idea is to formulate new models and conditions on D. Instead, we give guarantees for known learnable classes such as linear classes. We associate a linear class to a network.

Previous work involves

- Depth 2 networks (Whilliams98)
- Deep fully connected networks (Cho, Saul09)
- Deep convolutional networks (Mairal, Konlusz, Harchaoui, Schmid14)

SGD on top layer for depth-2 has guaranttes for depth-2 nets.

Networks often have a concise skeleton. We can break up into clusters, such that presence of an edge depends on the clusters that the 2 endpoints are in. The connectivity of these clusters is the **skeleton**.

To go from a skeleton to a network: a replication parameter r and a skeleton induce a network. Replace each noninput node in the skeleton with r nodes.

Define embedding to higher dimension. For $x \in X$ and $A \subseteq [n]$ let $x^A = \prod_{i\in A} x_i$. For $\mu = \{\mu_A\}_{A\subseteq[n]}$ with $\mu_A \ge 0$ and $\sum_{A\subseteq[n]} \mu_A = 1$ define

$$\Psi_\mu : \{\pm 1\}^n \to \mathbb{S}^{2^n-1} \tag{3}$$

$$\Psi_\mu(x) = (\sqrt{\mu_A x^A})_{A\subseteq[n],\mu_A>0}. \tag{4}$$

Think of this as a family of kernels, distribution says which are important. In this case,

$$\text{MARGIN}(\Psi_\mu, M) = \left\{\overline{\text{sign}}(f) : \|f\|_\mu \le M\right\}$$

where

$$\|f\|_\mu = \sqrt{\sum_A \frac{\widehat{f}(A)^2}{\mu_A}} \tag{5}$$

$$f(x) = \sum_{A\subseteq[n]} \widehat{f}(A) x^A. \tag{6}$$

Ex. if μ is uniform over degree 1 monomials, then $\|f\|_\mu = \sqrt{n \sum_{i=1}^n \widehat{f}(i)^2}$. If μ is uniform, then $\|f\|_\mu = 2^{\frac{n}{2}} \|f\|_2$.

$$\mu_i(i) = 1 \tag{7}$$

$$\widetilde{\mu}_v = \frac{1}{|in(v)|} \sum_{u \in in(v)} \mu_u \tag{8}$$

$$\mu_v = \sum_{i=0}^{\infty} a_i^2 (\widetilde{\mu}_v)^{*i} \tag{9}$$

where $\sigma_v = \sum_{i=0}^{\infty} a_i h_i$ is Hermite expansion of σ_v, $\mu^{*i}(A) = \sum_{A_1 \Delta \cdots \Delta A_i = A} \mu(A_1) \cdots \mu(A_i)$.

Fix constant depth, poly-sized skeleton S with C-bounded activations ($\|\sigma\|_\infty, \|\sigma'\|_\infty, \|\sigma''\|_\infty \leq C$).

Theorem 1.1. *For replcation $r \geq \text{poly}(n, M, \frac{1}{\varepsilon})$, SGD wrt the network $N(S, r)$ efficiently learns MARGIN(Ψ_S, M).*

Fully connected skeletons contain polynomials.

Fact 1.2: Constant degree polynomials with poly-bounded coefficients are contained in MARGIN$(\Psi_S, \text{poly}(n))$.

Corollary 1.3. *SGD on fully connected nets are guaranteed to learn large margin polynomial threshold functions, constant length DNFs, and constant length decision lists.*

[1]

Fully connected skeletons contain only polynomials. By symmetry, $\mu_A = \mu(|A|)$. Hence if $\mu(A) \leq \frac{1}{\binom{n}{|A|}}$. Then

$$\|f\|_S^2 \geq \sum_{A \subseteq [n]} \binom{n}{|A|} |\widehat{f}(A)|^2 \tag{10}$$

$$\left\|f - f^{\leq d}\right\|^2 \leq \frac{\|f\|_S^2}{\binom{n}{d+1}} \tag{11}$$

Convolutional skeletons contain complex local functions. $f : X_n \to \mathbb{R}$ is n'-local if $f = \sum_{i=1}^m f_i$ where each f_i depends on $\leq n'$ adjacent coordinates.

Super constant degree n'-local polys with poly-bounded weights are contained in MARGIN.

Weight sharing is neglected here.

What is captured by our theory?

- Structure: theory adapts to network's architecture.

[1] What's the benefit vs. linearizing? There is none. The goal is to provide some bounds on what NN to do. We have a followup paper about how to train those kernels.

What is missing?l

- Quantitative bounds: for networks with m parameters, guarantee with VC dimension $m^{\frac{1}{12}}$.

  Challenge: does depth-2 NN with n hidden neurons learn some class of VC dimension $\gg n$?

- Hiearchical learning:

  Inherent limitation of linear methods: linear classes amount to fixing embedding and learning linear threshold on top of that. You don't learn the embedding.

- Challenge: find tractable model for hierarchical learning. Doesn't have to be function class; can be assumption on D.