On the Computational Efficiency of Training Neural Networks

Roi Livni, Shai Shavli Shwartz and Ohad Shamir

INTRODUCTION.

- Neural Networks are almost the ultimate learner: Good generalization and high expressive power.
- A combination of algorithmic advancements, has led to a breakthrough in the effectiveness of training neural networks.
- The goal of this paper is to revisit and re-raise the question of neural network's computational efficiency, from a modern perspective.

NEURAL NETWORKS, MAIN CHALLENGE.

Neural Networks are formally hard to train. How can we circumvent hardness results?

- Over specified networks: While over specification seems to speedup training, formally hardness results are valid in the improper model.
- Changing the activation function: While changing the activation function from sigmoid to ReLU has lead to faster convergence of SGD methods, formally these networks are still hard.

Despite this theoretical pessimism, in practice, modern-day neural networks are trained successfully in many learning problems.

IN THIS WORK...

- We make a simple observation that for sufficiently over specified networks, global optima are in general easy to find.
- We prove that constant depth networks with constantly bounded weights are efficiently learnable.
- We show a provably efficient and practical algorithm for training two layers network with quadratic activation function, \( o(a) = a^2 \).

2-depth polynomial networks can be learnt via overspecification and a forward greedy selection method.

POLYNOMIAL NETWORKS.

In practice, the activation function of choice can have a significant effect on convergence rate. Choosing \( o(a) = a^2 \) leads to polynomial networks.

Polynomial network's expressiveness is comparable to classical neural networks.

- **Polynomial networks– expressive power 1**
  With \( O(\log T) \) layers and \( O(T^2) \) neurons, a polynomial network can implement any Turing machine with runtime \( T \).

- **Polynomial networks– expressive power 2**
  Polynomial networks with roughly \( O(t \log \log t) \) layers can approximate regular sigmoidal networks with \( t \) layers.

- **Polynomial networks– Training Time**
  Constant depth networks are contained in the class of constant degree polynomials and can be learnt efficiently. By expressive power 2, constant depth network with bounded weights are learnable.

EXPERIMENTS.

We considered pedestrian detection problem.

- We calculated HOG features and trained depth-2 Polynomials network with 40 hidden neurons.
- We’ve used both GECO and SGD.
- We also trained a similar network with ReLU activation function.

The second plot demonstrates the benefit of over-specification for SGD.

- We generated random examples and passed them through depth-2 network.
- We tried to fit new over-specified networks.

GENERALIZATION TO 3-DEGREE TENSORS.

- For \( k=1,...,O(\frac{1}{\epsilon}) \)
- Greedy Step: Add a new neuron: \( n_k(x) \)
- Optimize output layer: \( \sum_{k=1}^{O(\frac{1}{\epsilon})} a_k n_k(x) \)

\[
\text{What happens if we consider } n_k(x) = (w_k x) \text{? Suboptimal greedy step is enough. Guarantee: } O(\frac{1}{\epsilon}) \text{ neurons that competes with a network with } r \text{ neurons.}
\]

\[
\frac{1}{d^2} - \text{Suboptimal greedy step is enough. Guarantee: } O(\frac{1}{\epsilon}) \text{ neurons that competes with a network with } r \text{ neurons.}
\]

\[
\frac{1}{\sqrt{d}} \text{ max } < \ell(f(x,y)) x \geq 1 \frac{1}{\sqrt{d}} \text{ max } < \ell(f(x,y)) x \geq 1
\]

SUMMARY AND MAIN OPEN PROBLEMS.

- Deep networks pros: Optimal hypothesis class w.r.t expressiveness and sample complexity.
- Deep networks cons: No guarantees on training time.
- In this work: Networks of small depth and squared activation can be trained efficiently using greedy selection and over-specification.

Main problems:

- Why stochastic gradient descent methods work well in practice?
- To find a combination of network architecture and distributional assumptions that are useful in practice and lead to efficient algorithms.