Learning Two-layer Neural Networks with Symmetric Inputs

Rong Ge*, Rohith Kuditipudi*, Zhize Li§, Xiang Wang*

*Duke University, §Tsinghua University

Motivation

**Can we efficiently train a neural network?**
- Non convex optimization is NP-Hard in general.
- Require assumptions on distribution and network.

**What’s the right assumption on distribution?**
- Spherical Gaussian distribution. [T17] [LY17]
- Over-parameterization:
  - Kernel methods:
    1. Far from practice.
    2. Symmetric input distribution.
  - Robustness to noise.
  - Sample efficiency.

Challenges

- Even for Gaussian input, bad local minima exist. [GLM17] [SS17]. Gradient descent fails.
- Single neuron network: $y = \sigma(w^T x)$.

Warm-up: Single Neuron Network

- Single neuron network: $y = \sigma(w^T x)$.
- Observation [GKM18]: $E[\sigma(w^T x)] = \frac{1}{2} E[\sigma(w^T x)]$ $w$
- Proof. $2E[\sigma(w^T x)] = E[\sigma(w^T x) x] - \sigma(-w^T x) x = E[w^T x] x = E[\sigma(w^T x)] w$

Algorithm

$$E[\sigma(w^T x)] = \frac{1}{2} E[\sigma(w^T x)]$$

Model

- **Teacher Network.** Generate data: $x \sim \mathcal{D}$, $y = g(w^T x)$
- **Student Network.** Objective: $f(W) = E[\|g(W^T x) - g(x)\|^2]$
- Assumptions:
  1. Hidden layer width is smaller than both input and output. (multi-class classification, autoencoder)
  2. Symmetric input distribution.
  3. Weight matrices and input distribution are in "general position".

Reduce Two-layer to Single Neuron

- If we can find a direction where $u^T A = e_1^T$, then $u^T y = u^T A \sigma(W x) = \sigma(w^T x)$
- Finding a special direction can reduce the problem to a single neuron.

High Level Algorithm

1. Find special directions $u_i$
2. For each $u_i$, solve the single neuron problem with samples $(x, u_i^T y)$

Robust & Smoothed Analysis

**Theorem 1.** Suppose weight matrices $A, W$ and input distribution are not degenerate, our algorithm works with poly(d, 1/ε) time and number of samples.

**Theorem 2.** If $A, W, D$ are perturbed, with high probability, our algorithm works with poly(d, 1/ε, 1/λ) time and number of samples.

Experiments

- Sample efficiency. $A, W$ 3x3 random orthonormal matrices
- Robustness to noise. Gaussian noise; $A, W$ 10x10 mixture of Gaussian distribution; 10,000 samples

Linearization

- Problem: each entry of $pure(u)$ is a quadratic form in $u$. Solving a system of quadratic equations is NP-Hard in general.
- Linearization, inspired by FOOBI [DCC07].

Pure Neuron Detector

- **Pure neuron detector:** a function $pure(u)$ that is 0 if and only if $z = u^T y$ is a pure neuron.
  - Claim: when $z$ is a pure neuron $E[z^2(x \otimes x)] = 2E[z(x \otimes x)E[xx^T](x \otimes x)]$.
  - If $z$ is mixed, we have $RHS - LHS = \sum_{1 \leq i < k} c_{ij} E[w^T z] [w^T z] (x \otimes x)(w^T z)(w^T z) < 0$.

Le lemma. Let $pure(u) = E[(u^T y)(x \otimes x)] - 2E[u^T y E[(x \otimes x)E[xx^T]]] (x \otimes x)$. Then $pure(u)$ equals to 0 iff $u^T y$ is a pure neuron!

References