Towards Demystifying Overparameterization in Deep Learning

Mahdi Soltanolkotabi
Department of Electrical and Computer Engineering

USC University of Southern California
Mathematics of Imaging Workshop # 3
Henri Poincare Institute
Collaborators:
Samet Oymak and Mingchen Li
Motivation (Theory)
Many success stories

Neural networks very effective at learning from data
Lots of hype

AI is the New Electricity
Dr. Andrew Ng

Andrew Ng @AndrewYNg

Should radiologists be worried about their jobs? Breaking news: We can now diagnose pneumonia from chest X-rays better than radiologists.

stanfordmlgroup.github.io/projects/chexn...
Some failures

Microsoft silences its new A.I. bot Tay, after Twitter users teach it racism [Updated]

Sarah Perez @sarahintampa / 3 years ago

The Grim Conclusions of the Largest-Ever Study of Fake News
Falsehoods almost always beat out the truth on Twitter, penetrating further, faster, and deeper into the social network than accurate information.

ROBINSON MEYER MAR 8, 2018
Need more principled understanding

Deep learning-based AI increasingly used in human facing services

Challenges:

- Optimization: Why can they fit?
- Generalization: Why can they predict?
- Architecture: Which neural nets?
This talk: Overparameterization without overfitting

Mystery

# of parameters >> # training data

overfitting

just right!
Surprising experiment I (stolen from B. Recht)

$p$ parameters, $n = 50,000$ training samples, $d = 3072$ feature size, and 10 classes

<table>
<thead>
<tr>
<th>Model</th>
<th>parameters</th>
<th>p/n</th>
<th>Train loss</th>
<th>Test error</th>
</tr>
</thead>
<tbody>
<tr>
<td>CudaConvNet</td>
<td>145,578</td>
<td>2.9</td>
<td>0</td>
<td>23%</td>
</tr>
<tr>
<td>CudaConvNet (with regularization)</td>
<td>145,578</td>
<td>2.9</td>
<td>0.34</td>
<td>18%</td>
</tr>
<tr>
<td>MicroInception</td>
<td>1,649,402</td>
<td>33</td>
<td>0</td>
<td>14%</td>
</tr>
<tr>
<td>ResNet</td>
<td>2,401,440</td>
<td>48</td>
<td>0</td>
<td>13%</td>
</tr>
</tbody>
</table>
Surprising experiment II-Overfitting to corruption

Add corruption

- Corrupt a fraction of training labels by replacing with another random label
- No corruption on test labels

![Diagram showing accuracy over percentage of label corruption](image)
Surprising experiment III-Robustness

Repeat the same experiment but stop early
Surprising experiment III—Robustness

Repeat the same experiment but stop early

[Graph showing 50% label corruption with accuracy on the y-axis and epoch on the x-axis. The graph illustrates the difference between train and test accuracy over epochs, with a section indicating overfitting to noise.]
Benefits of overparameterization for neural networks

- Benefit I: Tractable nonconvex optimization
- Benefit II: Robustness to corruption with early stopping
Benefit I: Tractable nonconvex optimization
One-hidden layer

\[ y_i = \mathbf{v}^T \phi(\mathbf{W} \mathbf{x}_i) \]
Theory for smooth activations

Data set \( \{ (x_i, y_i) \}_{i=1}^n \) with \( \| x_i \|_{\ell_2} = 1 \)

\[
\min_W \mathcal{L}(W) := \sum_{i=1}^n \left( v^T \phi(W x_i) - y_i \right)^2
\]
Theory for smooth activations

Data set \( \{(x_i, y_i)\}_{i=1}^n \) with \( \|x_i\|_2 = 1 \)

\[
\min_W \mathcal{L}(W) := \sum_{i=1}^{n} (v^T \phi(Wx_i) - y_i)^2
\]

- Set \( v \) at random or balanced (half +, half -)
- Run gradient descent \( W_{\tau+1} = W_{\tau} - \mu_{\tau} \nabla \mathcal{L}(W_{\tau}) \) with random initialization

**Theorem (Oymak and Soltanolkotabi 2019)**

Assume

- Smooth activations \( |\phi'(z)| \leq B \) and \( |\phi''(z)| \leq B \)
- Overparameterization \( \sqrt{kd} \gtrsim \kappa(X)n \)
- Initialization with i.i.d. \( \mathcal{N}(0,1) \) entries

Then, with high probability

- Zero training error: \( \mathcal{L}(W_{\tau}) \leq (1 - c\frac{d}{n})^{2\tau} \mathcal{L}(W_0) \)
- Iterates remain close to initialization: \( \frac{\|W - W_0\|_F}{\|W_0\|_F} \lesssim \frac{\sqrt{kd}}{\sqrt{n}} \)
Dependence on data?

Diversity of input data is important...

\[
X = \begin{bmatrix}
    x_1^T \\
    x_2^T \\
    \vdots \\
    x_n^T
\end{bmatrix}
\]

\[
\kappa(X) := \sqrt{\frac{d}{n}} \frac{\|X\|}{\lambda(X)}
\]

Definition (Neural network covariance matrix and eigenvalue)

- Neural net covariance matrix

\[
\Sigma(X) := \frac{1}{k} \mathbb{E}_{W_0} \left[ \mathcal{J}(W_0) \mathcal{J}^T(W_0) \right]
= \mathbb{E}_{w} \left[ (\phi'(Xw)\phi'(Xw)^T) \odot (XX^T) \right].
\]

- Eigenvalue \( \lambda(X) := \lambda_{\text{min}}(\Sigma(X)) \)
Lemma

Let $\{\mu_r(\phi')\}_{r=0}^{\infty}$ be the Hermite coefficients of $\phi'$. Then,

$$\Sigma(X) = \sum_{r=0}^{+\infty} \mu_r^2(\phi') (XX^T) \odot \ldots \odot (XX^T) \succeq \mu_2^2(\phi') (XX^T) \odot (XX^T) \quad (E[\phi''(g)])^2$$

arbitrary activation $\iff$ quadratic activation

Conclusion

For generic data e.g. $x_i$ i.i.d. uniform on the unit sphere $\kappa(X)$ scales like a constant
Theory for ReLU activations

Data set \( \{(x_i, y_i)\}_{i=1}^n \) with \( \|x_i\|_2 = 1 \)

\[
\min_{W} \mathcal{L}(W) := \sum_{i=1}^{n} (v^T \phi(Wx_i) - y_i)^2
\]

- Set \( v \) at random or balanced (half +, half −)
- Run gradient descent \( W_{\tau+1} = W_{\tau} - \mu_{\tau} \nabla \mathcal{L}(W_{\tau}) \) with random initialization

**Theorem (Oymak and Soltanolkotabi 2019)**

**Assume**
- ReLU activation \( \phi(z) = ReLU(z) = \max(0, z) \)
- Overparameterization \( \sqrt{kd} \gtrsim \kappa^3(X) \frac{n}{d} \times n \)
- Initialization with i.i.d. \( \mathcal{N}(0, 1) \) entries

Then, with high probability
- Zero training error: \( \mathcal{L}(W_{\tau}) \leq \left(1 - c \frac{d}{n}\right)^{2\tau} \mathcal{L}(W_0) \)
- Iterates remain close to initialization: \( \frac{\|W-W_0\|_F}{\|W_0\|_F} \lesssim \frac{\sqrt{kd}}{\sqrt{n}} \)
Theory for SGD

Data set $\{(x_i, y_i)\}_{i=1}^n$ with $\|x_i\|_{\ell_2} = 1$

$$\min_{W} \mathcal{L}(W) := \sum_{i=1}^{n} (v^T \phi(W x_i) - y_i)^2$$

- Set $v$ at random or balanced (half $+$, half $-$)
- Run gradient descent $W_{\tau+1} = W_{\tau} - \mu_{\tau} \nabla \mathcal{L}(W_{\tau})$ with random initialization

Theorem (Oymak and Soltanolkotabi 2019)

Assume

- **Smooth activations** $|\phi'(z)| \leq B$ and $|\phi''(z)| \leq B$
- **Overparameterization** $\sqrt{kd} \gtrsim \kappa(X)n$
- **Initialization** with i.i.d. $\mathcal{N}(0,1)$ entries

Then, with high probability

- **Zero training error**: $\mathbb{E}[\mathcal{L}(W_{\tau})] \leq (1 - c \frac{d}{n^2})^{2\tau} \mathcal{L}(W_0)$
- **Iterates remain close to initialization**: $\frac{\|W - W_0\|_F}{\|W_0\|_F} \lesssim \frac{\sqrt{kd}}{\sqrt{n}}$
Proof Sketch
Prelude: over-parametrized linear least-squares

\[
\min_{\theta \in \mathbb{R}^p} \mathcal{L}(\theta) := \frac{1}{2} \|X\theta - y\|_2^2 \quad \text{with} \quad X \in \mathbb{R}^{n \times p} \quad \text{and} \quad n \leq p.
\]

Gradient descent starting from \(\theta_0\) has three properties:

- Global convergence
- Converges to a global optimum which is closest to \(\theta_0\)
- Total gradient path length is relatively short
Over-parametrized nonlinear least-squares

\[
\min_{\theta \in \mathbb{R}^p} \mathcal{L}(\theta) := \frac{1}{2} \|f(\theta) - y\|_{\ell_2}^2,
\]

where

\[
y := \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n, \quad f(\theta) := \begin{bmatrix} f(x_1; \theta) \\ f(x_2; \theta) \\ \vdots \\ f(x_n; \theta) \end{bmatrix} \in \mathbb{R}^n, \quad \text{and} \quad n \leq p.
\]

Gradient descent: start from some initial parameter \(\theta_0\) and run

\[
\theta_{\tau+1} = \theta_\tau - \eta_\tau \nabla \mathcal{L}(\theta_\tau),
\]

\[
\nabla \mathcal{L}(\theta) = \mathcal{J}(\theta)^T(f(\theta) - y).
\]

Here, \(\mathcal{J}(\theta) \in \mathbb{R}^{n \times p}\) is the Jacobian matrix with entries \(\mathcal{J}_{ij} = \frac{\partial f(x_i, \theta)}{\partial \theta_j}\).
**Key lemma**

**Lemma**

Following assumptions on $B(\theta_0, R)$ with $R := \frac{4\|f(\theta_0) - y\|_{\ell_2}}{\alpha}$

- **Jacobian at initialization:** $\sigma_{\min}(\mathcal{J}(\theta_0)) \geq 2\alpha$
- **Bounded Jacobian spectrum:** $\|\mathcal{J}(\theta)\| \leq \beta$
- **Lipschitz Jacobian:** $\|\mathcal{J}(\tilde{\theta}) - \mathcal{J}(\theta)\| \leq L \|\tilde{\theta} - \theta\|_F$
- **Small initial residual:** $\|f(\theta_0) - y\|_{\ell_2} \leq \frac{\alpha^2}{4L}$

Then using step size $\eta \leq \frac{2}{\beta}$

- **Global geometric convergence:** $\|f(\theta_\tau) - y\|_{\ell_2}^2 \leq \left(1 - \frac{\eta \alpha^2}{2}\right)^\tau \|f(\theta_0) - y\|_{\ell_2}^2$
- **Iterates stay close to init.:** $\|\theta_\tau - \theta_0\|_{\ell_2} \leq \frac{4}{\alpha} \|f(\theta_0) - y\|_{\ell_2} \leq \frac{4 \beta}{\alpha} \|\theta^* - \theta_0\|_{\ell_2}$
- **Total gradient path bounded:** $\sum_{\tau=0}^{\infty} \|\theta_{\tau+1} - \theta_\tau\|_{\ell_2} \leq \frac{4}{\alpha} \|f(\theta_0) - y\|_{\ell_2}$

**Key Ideal**

Track dynamics of $V_\tau := \|r_\tau\|_{\ell_2} + \frac{1}{2} \left(1 - \eta \beta^2\right) \sum_{t=1}^{\tau-1} \|\theta_{t+1} - \theta_t\|_{\ell_2}$. 
Proof sketch (SGD)

Challenge: show that SGD remains in the local neighborhood

- Attempt I: Show $\|\theta_\tau - \theta_0\|_{\ell_2}$ is a super martingale (see also [Tan and Vershynin 2017])
- Attempt II: Show that $\|f(\theta_\tau) - y\|_{\ell_2} + \lambda \|\theta_\tau - \theta_0\|_{\ell_2}$ is a super martingale
- Final attempt: Show that

$$\frac{1}{K} \sum_{j=1}^{K} \|\theta_\tau - v_i\|_{\ell_2} + \frac{3\eta}{n} \|J^T(\theta_\tau)(f(\theta_\tau) - y)\|_{\ell_2}$$

is a super-martingale. Here, $v_i$ is a very fine cover of $B(\theta_0, R)$
Over-parametrized nonlinear least-squares for neural nets

\[
\min_{W \in \mathbb{R}^{k \times d}} \mathcal{L}(W) := \frac{1}{2} \| f(W) - y \|_{\ell_2}^2,
\]

where

\[
y := \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n, \quad f(W) := \begin{bmatrix} f(W, x_1) \\ f(W, x_2) \\ \vdots \\ f(W, x_n) \end{bmatrix} \in \mathbb{R}^n, \quad \text{and} \quad n \leq kd.
\]

Linearization via Jacobian

\[
\mathcal{J}(W) = X \ast \left( \phi' \left( X W^T \right) \text{diag}(v) \right)
\]
Key Techniques

- Hadamard product

\[ \mathcal{J}(W) \mathcal{J}^T(W) = (\phi'(XW^T)\phi'(WX^T)) \odot (XX^T) \]

**Theorem (Schur 1913)**

For two PSD \( A, B \in \mathbb{R}^{n \times n} \)

\[ \lambda_{\min} (A \odot B) \geq \left( \min_i B_{ii} \right) \lambda_{\min}(A) \]

\[ \lambda_{\max} (A \odot B) \leq \left( \max_i B_{ii} \right) \lambda_{\max}(A) \]

- Random matrix theory

\[ \mathcal{J}(W) \mathcal{J}^T(W) = \sum_{\ell=1}^{k} (\phi'(Xw_{\ell})\phi'(Xw_{\ell})^T) \odot (XX^T) \]
Side corollary: Nonconvex matrix recovery

- Features: \( A_1, A_2, \ldots, A_n \in \mathbb{R}^{d \times d} \).
- Labels: \( y_1, y_2, \ldots, y_n \).
- Solve Nonconvex matrix factorization

\[
\min_{U \in \mathbb{R}^{d \times r}} \frac{1}{2} \sum_{i=1}^{n} \left( y_i - \langle A_i, U U^T \rangle \right)^2
\]

Theorem (Oymak and Soltanolkotabi 2018)

Assume

- i.i.d. Gaussian \( A_i \)
- any label \( y_i \)
- Initialization at well conditioned matrix \( U_0 \)

Then, gradient descent iterations \( U_\tau \) converge with a geometric rate to a close global optima as soon as \( n \leq dr \).

- Burer-Monteiro and many others \( r \geq \sqrt{n} \)
- For Gaussian \( A_i \) we allow \( r \geq \frac{n}{d} \)

when \( n \approx dr_0 \)
- Burer-Monteiro: \( r \gtrsim \sqrt{dr_0} \)
- Ours: \( r \gtrsim r_0 \)
Previous work

- Unrealistic quadratic: [Soltanolkotabi, Javanmard, Lee 2018] and [Venturi, Bandeira, Bruna,...]
- Smooth activations: [Du, Lee, Li, Wang, Zhai 2018]
  \[ kd \gtrsim n^2 \text{ versus } k \gtrsim n^4. \]
- ReLU activation: [Du et. al. 2018]
  \[ k \gtrsim \frac{n^4}{d^3} \text{ versus } k \gtrsim n^6. \]
- Separation: [Li and Liang 2018] [ Allen-Zhu, Li, Song 2018]
  \[ k \gtrsim \frac{n^{12}}{\delta^4} \text{ versus } k \gtrsim n^{25}??.??. \]
- Begin to move beyond “lazy training” [Chizat & Bach, 2018];
- Faster convergence rate
- Mean field analysis for infinitely wide: [Mei et al., 2018]; [Chizat & Bach, 2018]; [Sirignano & Spiliopoulos, 2018]; [Rotskoff & Vanden-Eijnden, 2018]; [Wei et al., 2018].
Related recent literature

- **Approximation capability**
  [Barron 1994], [Telgarsky 2016], [Bolcskei, Grohs, Kutyniok, and Petersen 2017]

- **More over-parameterization** ($n \leq ck$)
  [Poston, Lee, Choie, and Kwon 1991], [Haeffele and Vidal 2015], [Nguyen and Hein 2017]

- **Under-parameterized with resampling**
  [Oymak 2018], [Ge, Ma, Lee 2017], [Zhong, Song, Jain, Bartlett, and Dhillon 2017]
  [Brutzkus and Globerson 2017] and [Li and Yuan 2017]

- **Other learning methods (Tensors, kernels, etc)**
  [Janzamin, Sedghi, and Anandkumar 2015], [Goel and Klivans 2017]

- **Generalization**
  [Hardt, Benjamin Recht, Yoram Singer 2016],
  [Brutzkus, Globerson, Malach, and Shalev-Shwartz 2017],
  [Golowich, Rakhlin, Shamir 2017],
  [Dziugaite and Roy 2017], [Bartlett, Foster, Telgarsky 2017],
  [Neysahbur, Bhojanapalli, McAllester, Srebro 2017]
  [Arora, Ge, Neyshabur, and Zhang 2018]
  [Arora, Cohen, Hazan 2018], [Azzian, Hassibi 2018]

- **Interface with statistical physics**
  [Choromanskaya, Henaff, Mathieu, Arous, LeCun 2015],
  [Lee, Bahri, Novak, Schoenholz, Pennington, Sohl-dickstein 2018],
  [Novak, Bahri, Abolafia, Pennington, Sohl-Dickstein 2018],

- **Many others...**
The need for overparameterization beyond width

Simple exercise: initialize $W$ at random and just fit output layer weights

$$
\mathcal{L}(v) := \frac{1}{2} \sum_{i=1}^{n} \left( v^T \phi(Wx_i) - y_i \right)^2 = \frac{1}{2} \left\| \phi(XW^T)v - y \right\|_{\ell_2}^2 ,
$$

Simple least-squares problem

$$
\hat{v} := \Phi^T (\Phi \Phi^T)^{-1} y \quad \text{where} \quad \Phi := \phi(XW^T).
$$

**Theorem (Oymak and Soltanolkotabi 2019)**

*Fitting the output layer perfectly interpolates the data w.h.p. as soon as*

$$
k \gtrsim n
$$
There is still a huge gap!
Benefit II: Robustness to corruption
Surprising experiment III-Robustness
Repeat the same experiment but stop early

![Graph showing the relationship between test accuracy, train accuracy, and train acc w.r.t. true labels and the percentage of label corruption. The graph indicates a decrease in accuracy as the percentage of label corruption increases.]
Model (without corruption)

**clean data**: \((\epsilon_0, \delta)\)-clusterable data

input/label pairs \(\{(x_i, y_i)\}_{i=1}^n \in \mathbb{R}^d \times [-1, 1]\)

\(L\) clusters and \(K\) classes
Robustness to corruption

Clean data points \( \{(x_i, \bar{y}_i)\}_{i=1}^n \), corrupt \( s := \rho n \) to get corrupted data \( \{(x_i, y_i)\}_{i=1}^n \).

Fit

\[
\mathcal{L}(W) := \frac{1}{2} \sum_{i=1}^{n} (f(W, x_i) - y_i)^2
\]

via gradient descent

**Theorem (Oymak and Soltanolkotabi 2019)**

**Assume**

- **Corruption level** \( \rho < \frac{1}{16} \)
- **Cluster radius** \( \epsilon \lesssim \frac{1}{L^2} \)
- **# Overparameterization** \( k \times d \gtrsim \kappa^2(C)L^4 \)

Starting from random initialization, after \( \tau \sim L \log(1/\rho) \) iterations, gradient descent finds a model with perfect accuracy, i.e.

\[
\text{closest label to } f(W_{\tau}, x_i) = \text{true label } \bar{y}_i
\]
Learning versus overfitting

Key insight: distance from initialization

![Graph showing accuracy over distance traveled from initial]

**Theorem (Oymak and Soltanolkotabi 2019)**

- *With early stopping* (\(\tau \sim L \log(\rho)\)) *distance is bounded* \(\|W - W_0\|_F \lesssim \sqrt{L}\)
- *To overfit to the corruption you have to travel far* \(\|W - W_0\|_F \propto \sqrt{s}\)
Proof Sketch
High-level intuition

- Intuition I: Network should learn when there is no corruption
- Intuition II: Network should not fit to the corruption
### Key Idea I

**Reminder**
- Gradient $\nabla L(\theta) = J(\theta) (f(\theta, X) - y)$
- Jacobian $J(\theta) = \begin{bmatrix} \frac{\partial f(\theta, x_1)}{\partial \theta} & \frac{\partial f(\theta, x_2)}{\partial \theta} & \cdots & \frac{\partial f(\theta, x_n)}{\partial \theta} \end{bmatrix} \in \mathbb{R}^{p \times n}$

### Key Ideal I

*If $\epsilon = 0$, there are only $L$ distinct inputs. $J$ has exactly rank $L$.***
Key Idea 1

Reminder

- Gradient $\nabla \mathcal{L}(\theta) = \mathcal{J}(\theta) (f(\theta, X) - y)$
- Jacobian $\mathcal{J}(\theta) = \begin{bmatrix} \frac{\partial f(\theta, x_1)}{\partial \theta} & \frac{\partial f(\theta, x_2)}{\partial \theta} & \ldots & \frac{\partial f(\theta, x_n)}{\partial \theta} \end{bmatrix} \in \mathbb{R}^{p \times n}$

Key Ideal 1

If $\epsilon$ is small, there are only $L$ distinct inputs. $\mathcal{J}$ has approximately rank $L$. 

![Diagram with clusters and Jacobian spectrum]
Key Idea II

Two complementary subspaces

- **Fast (data) subspace** $\mathcal{F}$: Subspace associated with top $L$ right singular vectors of $\mathcal{J}$
- **Slow (noise) subspace** $\mathcal{S}$: Complement of $\mathcal{F}$

Interaction of Jacobian and residual in the gradient $\nabla \mathcal{L}(\theta) = \mathcal{J}(\theta)(f(\theta, X) - y)$

Residual can be decomposed into two terms

$$r(\theta) := f(\theta, X) - y = f(\theta, X) - \bar{y} + \bar{y} - y$$

- Residual w.r.t. true labels falls mostly onto $\mathcal{F}$ and **quickly** goes to zero
- Corruption $y - \bar{y}$ falls mostly onto $\mathcal{S}$ and goes very slowly to zero
What about real data?
- Dataset: CIFAR10
- Model: ResNET20
- Task: Binary classification (airplane vs truck)
- \( n = 10,000 \) and \( p = 270,000 \)
Conclusion

Provable benefits of overparameterization

- More tractable optimization
- Robustness to corruption
Mandatory Postdoc Announcement
References

- Theoretical insights into the Optimization Landscape of Over-parameterized Shallow Neural Nets M. Soltanolkotabi, A. Javanmard, and J. D. Lee 2017.
- Over-parametrized nonlinear learning: Gradient descent follows the shortest path? S. Oymak and M. Soltanolkotabi
- Gradient Descent is Provably Robust to Label Noise for Overparameterized Neural Networks. S. Oymak and M. Soltanolkotabi
- Gradient Descent with Early Stopping is Provably Robust to Label Noise for Overparameterized Neural Networks. S. Oymak and M. Soltanolkotabi
Thanks!

Funding acknowledgment