A Mean Field Analysis Of Deep ResNet:
Towards Provable Optimization Via Overparameterization From Depth

Joint work with Chao Ma, Yulong Lu, Jianfeng Lu and Lexing Ying

Presenter: Yiping Lu
Contact: yplu@stanford.edu, https://web.stanford.edu/~yplu/
Global Convergence Proof Of NN

- Neural Tangent Kernel ([Jacot et al. 2019]): Linearize the model

\[ f_{NN}(\theta) = f_{NN}(\theta_{init}) + \left< \nabla_{\theta} f_{NN}(\theta_{init}), \theta - \theta_{init} \right> \]
Global Convergence Proof Of NN

- Neural Tangent Kernel ([Jacot et al. 2019]): Linearize the model
  \[ f_{\text{NN}}(\theta) = f_{\text{NN}}(\theta_{\text{init}}) + \langle \nabla_{\theta} f_{\text{NN}}(\theta_{\text{init}}), \theta - \theta_{\text{init}} \rangle \]
  - **Pro**: can provide proof of convergence for any structure of NN. ([Li et al. 2019])
  - **Con**: Feature is lazy learned, i.e. not data dependent. ([Chizat and Bach 2019.][Ghorbani et al. 2019])

Mean Field Regime ([Bengio et al. 2006][Bach et al. 2014][Suzuki et al. 2015]): We consider properties of the loss landscape with respect to the distribution of weights
  \[ L(\rho) = \|E_{\theta \sim \rho} g(\theta, x) - f(x)\|_2^2, \]
  the objective is a convex function
  - **Pr**: SGD = Wasserstein Gradient Flow ([Mei et al. 2018][Chizat et al. 2018][Rotskoff et al. 2018])
  - **Con**: Hard to generalize beyond two layer
Global Convergence Proof Of NN

- **Neural Tangent Kernel** ([Jacot et al. 2019]): Linearize the model

\[ f_{NN}(\theta) = f_{NN}(\theta_{init}) + \langle \nabla_{\theta} f_{NN}(\theta_{init}), \theta - \theta_{init} \rangle \]

  - **Pro**: can provide proof of convergence for any structure of NN. ([Li et al. 2019])
  - **Con**: Feature is lazy learned, i.e. not data dependent. ([Chizat and Bach 2019][Ghorbani et al. 2019])

- **Mean Field Regime** ([Bengio et al. 2006][Bach et al. 2014][Suzuki et al. 2015]): We consider properties of the loss landscape with respect to the distribution of weights \( L(\rho) = \| \mathbb{E}_{\theta \sim \rho} g(\theta, x) - f(x) \|_2^2 \), the objective is a convex function.
Global Convergence Proof Of NN

- **Neural Tangent Kernel ([Jacot et al. 2019]):** Linearize the model
  \[ f_{\text{NN}}(\theta) = f_{\text{NN}}(\theta_{\text{init}}) + < \nabla_{\theta} f_{\text{NN}}(\theta_{\text{init}}), \theta - \theta_{\text{init}} > \]
  - **Pro:** can provide proof of convergence for any structure of NN. ([Li et al. 2019])
  - **Con:** Feature is lazy learned, i.e. not data dependent. ([Chizat and Bach 2019.][Ghorbani et al. 2019])

- **Mean Field Regime ([Bengio et al. 2006][Bach et al. 2014][Suzuki et al. 2015]):** We consider properties of the loss landscape with respect to the distribution of weights \( L(\rho) = \| \mathbb{E}_{\theta \sim \rho} g(\theta, x) - f(x) \|_2^2 \), the objective is a convex function
  - **Pro:** SGD = Wasserstein Gradient Flow ([Mei et al. 2018][Chizat et al. 2018][Rotskoff et al. 2018])
  - **Con:** Hard to generalize beyond two layer
Neural Ordinary Differential Equation

Figure: ResNet can be seen as the Euler discretization of a time evolving ODE

\[
\frac{dx}{dt} = F(x, t),
\]
\[
x_i+1 = x_i + \gamma F(x_i, t_i),
\]
\[
x_0 = x(0), x_i = x(\gamma i), ... \gamma \text{ - step size}
\]
Neural Ordinary Differential Equation

Limit of depth $\rightarrow \infty$

This analogy does not directly provide guarantees of global convergence even in the continuum limit.
Mean Field ResNet

**Our Aim:** Provide a **new** continuous limit for ResNet with good limiting landscape.

**Idea:** We consider properties of the loss landscape with respect to the distribution of weights.
Mean Field ResNet

Our Aim: Provide a new continuous limit for ResNet with good limiting landscape.

Idea: We consider properties of the loss landscape with respect to the distribution of weights.

Here:

- Input data is the initial condition $X_\rho(x, 0) = \langle w_2, x \rangle$
- $X$ is the feature, $t$ represents the depth.
- Loss function: $E(\rho) = \mathbb{E}_{x \sim \mu} \left[ \frac{1}{2} \left( \langle w_1, X_\rho(x, 1) \rangle - y(x) \right)^2 \right]$. 
Adjoint Equation

To optimize the Mean Field model, we calculate the gradient $\frac{\delta E}{\delta \rho}$ via the adjoint sensitivity method.

Model

The loss function can be written as

$$E_{x \sim \mu} E(x; \rho) := \mathbb{E}_{x \sim \mu} \frac{1}{2} |\langle w_1, X_\rho(x, 1) \rangle - y(x) |^2$$

(1)

where $X_\rho$ satisfies the equation $\dot{X}_\rho(x, t) = \int_\theta f(X_\rho(x, t), \theta) \rho(\theta, t) d\theta$. 

[Image of Stanford University logo]
**Adjoint Equation**

To optimize the Mean Field model, we calculate the gradient $\frac{\delta E}{\delta \rho}$ via the *adjoint sensitivity method*.

**Model**

The loss function can be written as

$$E_{x \sim \mu} E(x; \rho) := E_{x \sim \mu} \frac{1}{2} |\langle w_1, X_\rho(x, 1) \rangle - y(x)|^2$$  \hspace{1cm} (1)

where $X_\rho$ satisfies the equation $\dot{X}_\rho(x, t) = \int_\theta f(X_\rho(x, t), \theta) \rho(\theta, t) d\theta$.

**Adjoint Equation.** The gradient can be represented as a second backwards-in-time augmented ODE.

$$\dot{p}_\rho(x, t) = -\delta_x H_\rho(p_\rho, x, t)$$

$$= -p_\rho(x, t) \int \nabla_x f(X_\rho(x, t), \theta) \rho(\theta, t) d\theta,$$

Here the Hamiltonian is defined as $H_\rho(p, x, t) = p(x, t) \cdot \int f(x, \theta) \rho(\theta, t) d\theta$. 
Theorem

For $\rho \in \mathcal{P}^2$ let $\frac{\delta E}{\delta \rho}(\theta, t) = \mathbb{E}_{x \sim \mu} f(X_\rho(x, t), \theta))p_\rho(x, t)$, where $p_\rho$ is the solution to the backward equation $\dot{p}_\rho(x, t) = -p_\rho(x, t) \int \nabla_x f(X_\rho(x, t), \theta)\rho(\theta, t) d\theta$. Then for every $\nu \in \mathcal{P}^2$, we have

$$E(\rho + \lambda(\nu - \rho)) = E(\rho) + \lambda \left\langle \frac{\delta E}{\delta \rho}, (\nu - \rho) \right\rangle + o(\lambda)$$

for the convex combination $(1 - \lambda)\rho + \lambda\nu \in \mathcal{P}^2$ with $\lambda \in [0, 1]$. 

Adjoint equation is equivalent to the back propagation
Deep Residual Network Behaves Like an Ensemble Of Shallow Models

\[ x^1 = x^0 + \frac{1}{L} \int_{\theta^0} \sigma(\theta^0 x^0) \rho^0(\theta^0) d\theta^0. \]
Deep Residual Network Behaves Like an Ensemble Of Shallow Models

\[ X^1 = X^0 + \frac{1}{L} \int_{\theta^0} \sigma(\theta^0 X^0) \rho^0(\theta^0) d\theta^0. \]

\[ X^2 = X^0 + \frac{1}{L} \int_{\theta^0} \sigma(\theta^0 X^0) \rho^0(\theta^0) d\theta^0 + \int_{\theta^1} \sigma(\theta^1 X^0) + \frac{1}{L} \int_{\theta^0} \sigma(\theta^0 X^0) \rho^0(\theta^0) d\theta^0) \rho^1(\theta^1) d\theta^1 \]

\[ = X^0 + \frac{1}{L} \int_{\theta^0} \sigma(\theta^0 X^0) \rho^0(\theta^0) d\theta^0 + \frac{1}{L} \int_{\theta^1} \sigma(\theta^1 X^0) \rho^1(\theta^1) d\theta^1 + \frac{1}{L^2} \int_{\theta^1} \nabla \sigma(\theta^1 X^0) \theta^1 (\int_{\theta^0} \sigma(\theta^0 X^0) \rho^0(\theta^0) d\theta^0) \rho^1(\theta^1) d\theta^1 + h.o.t. \]
Deep Residual Network Behaves Like an Ensemble Of Shallow Models

\[ X^1 = X^0 + \frac{1}{L} \int_{\theta^0} \sigma(\theta^0 X^0) \rho^0(\theta^0) d\theta^0. \]

\[ X^2 = X^0 + \frac{1}{L} \int_{\theta^0} \sigma(\theta^0 X^0) \rho^0(\theta^0) d\theta^0 + \frac{1}{L} \int_{\theta^1} \sigma(\theta^1(X^0 + \frac{1}{L} \int_{\theta^0} \sigma(\theta^0 X^0) \rho^0(\theta^0) d\theta^0)) \rho^1(\theta^1) d\theta^1 \]

\[ = X^0 + \frac{1}{L} \int_{\theta^0} \sigma(\theta^0 X^0) \rho^0(\theta^0) d\theta^0 + \frac{1}{L} \int_{\theta^1} \sigma(\theta^1 X^0) \rho^1(\theta^1) d\theta^1 + \frac{1}{L^2} \int_{\theta^1} \nabla \sigma(\theta^1 X^0) \theta^1 \left( \int_{\theta^0} \sigma(\theta^0 X^0) \rho^0(\theta^0) d\theta^0 \right) \rho^1(\theta^1) d\theta^1 + h.o.t. \]

Iterating this expansion gives rise to

\[ X^L \approx X^0 + \frac{1}{L} \sum_{a=0}^{L-1} \int \sigma(\theta X^0) \rho^a(\theta) d\theta + \frac{1}{L^2} \sum_{b>a} \int \int \nabla \sigma(\theta^b X^0) \theta^b \sigma(\theta^a X^0) \rho^b(\theta^b) \rho^a(\theta^a) d\theta^b \theta^a + h.o.t. \]

Deep Residual Network Behaves Like an Ensemble Of Shallow Models

Difference of back propagation process of two-layer net and ResNet.

Two-layer Network

\[ \frac{\delta E}{\delta \rho}(\theta, t) = \mathbb{E}_{x \sim \mu}f(x, \theta))(X_\rho - y(x)) \]

ResNet

\[ \frac{\delta E}{\delta \rho}(\theta, t) = \mathbb{E}_{x \sim \mu}f(X_\rho(x, t), \theta))p_\rho(x, t) \]

We aim to show that the two gradient are similar.
Deep Residual Network Behaves Like an Ensemble Of Shallow Models

Difference of back propagation process of two-layer net and ResNet.

Two-layer Network

\[ \frac{\delta E}{\delta \rho} (\theta, t) = \mathbb{E}_{x \sim \mu} f(x, \theta)(X_\rho - y(x)) \]

ResNet

\[ \frac{\delta E}{\delta \rho} (\theta, t) = \mathbb{E}_{x \sim \mu} f(X_\rho(x, t), \theta)) p_\rho(x, t) \]

We aim to show that the two gradient are similar.

Lemma

The norm of the solution to the adjoint equation can be bounded by the loss

\[ \|p_\rho(\cdot, t)\|_\mu \geq e^{-(C_1 + C_2 r)} E(\rho), \forall t \in [0, 1] \]
Local = Global

**Theorem**

If $E(\rho) > 0$ for distribution $\rho \in \mathcal{P}^2$ that is supported on one of the nested sets $Q_r$, we can always construct a descend direction $\nu \in \mathcal{P}^2$, i.e.

$$\inf_{\nu \in \mathcal{P}^2} \langle \frac{\delta E}{\delta \rho}, (\nu - \rho) \rangle < 0$$
Local = Global

Theorem
If $E(\rho) > 0$ for distribution $\rho \in \mathcal{P}^2$ that is supported on one of the nested sets $Q_r$, we can always construct a descend direction $\nu \in \mathcal{P}^2$, i.e.

$$\inf_{\nu \in \mathcal{P}^2} \left\langle \frac{\delta E}{\delta \rho}, (\nu - \rho) \right\rangle < 0$$

Corollary
Consider a stationary solution to the Wasserstein gradient flow which is full support (informal), then it’s a global minimizer.
Numerical Scheme

We may consider using a parametrization of $\rho$ with $n$ particles as

$$\rho_n(\theta, t) = \sum_{i=1}^{n} \delta_{\theta_i}(\theta) \mathbb{1}_{[\tau_i, \tau'_i]}(t).$$

The characteristic function $\mathbb{1}_{[\tau_i, \tau'_i]}$ can be viewed as a relaxation of the Dirac delta mass $\delta_{\tau_i}(t)$.

---

**Given:** A collection of residual blocks $(\theta_i, \tau_i)_{i=1}^{n}$

**while** training **do**

- Sort $(\theta_i, \tau_i)$ based on $\tau_i$ to be $(\theta^i, \tau^i)$ where $\tau^0 \leq \cdots \leq \tau^n$.
- Define the ResNet as $X^{\ell+1} = X^{\ell} + (\tau^\ell - \tau^{\ell-1})\sigma(\ell X^{\ell})$ for $0 \leq \ell < n$.
- Use gradient descent to update both $\theta^i$ and $\tau^i$.

**end while**
## Numerical Results

<table>
<thead>
<tr>
<th></th>
<th>Vanilla</th>
<th>mean-field</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet20</td>
<td>8.75</td>
<td>8.19</td>
<td>CIFAR10</td>
</tr>
<tr>
<td>ResNet32</td>
<td>7.51</td>
<td>7.15</td>
<td>CIFAR10</td>
</tr>
<tr>
<td>ResNet44</td>
<td>7.17</td>
<td>6.91</td>
<td>CIFAR10</td>
</tr>
<tr>
<td>ResNet56</td>
<td>6.97</td>
<td>6.72</td>
<td>CIFAR10</td>
</tr>
<tr>
<td>ResNet110</td>
<td>6.37</td>
<td>6.10</td>
<td>CIFAR10</td>
</tr>
<tr>
<td>ResNet164</td>
<td>5.46</td>
<td>5.19</td>
<td>CIFAR10</td>
</tr>
<tr>
<td>ResNeXt29(864d)</td>
<td>17.92</td>
<td>17.53</td>
<td>CIFAR100</td>
</tr>
<tr>
<td>ResNeXt29(1664d)</td>
<td>17.65</td>
<td>16.81</td>
<td>CIFAR100</td>
</tr>
</tbody>
</table>

**Table:** Comparison of the stochastic gradient descent and mean-field training (Algorithm 1.) of ResNet On CIFAR Dataset. Results indicate that our method our performs the Vanilla SGD consistently.
Take Home Message

- We propose a new continuous limit for deep resnet

\[ \dot{X}_\rho(x, t) = \int_\theta f(X_\rho(x, t), \theta) \rho(\theta, t) d\theta, \]

with initial \( X_\rho(x, 0) = \langle w_2, x \rangle \)

- Local minimizer is global in \( \ell_2 \) space.
- A potential scheme to approximate.
We propose a new continuous limit for deep resnet

\[
\dot{X}_\rho(x, t) = \int_\theta f(X_\rho(x, t), \theta)\rho(\theta, t)d\theta,
\]

with initial \( X_\rho(x, 0) = \langle w_2, x \rangle \)

- Local minimizer is global in \( \ell_2 \) space.
- A potential scheme to approximate.

**TO DO List.**

- Analysis of Wasserstein gradient flow. (Global Existence)
- Refined analysis of numerical scheme
- h.o.t in the expansion from ResNet to ensemble of small networks.
Thanks


Contact: yplu@stanford.edu