Towards Understanding Optimization of Deep Learning

Tianle Cai

Peking University -> Princeton University

Joint work with: Siyu Chen, Ruiqi Gao, Di He, Cho-jui Hsieh, Jikai Hou, Jason Lee, Haochuan Li, Dong Wang, Liwei Wang, Zhihua Zhang

Outline

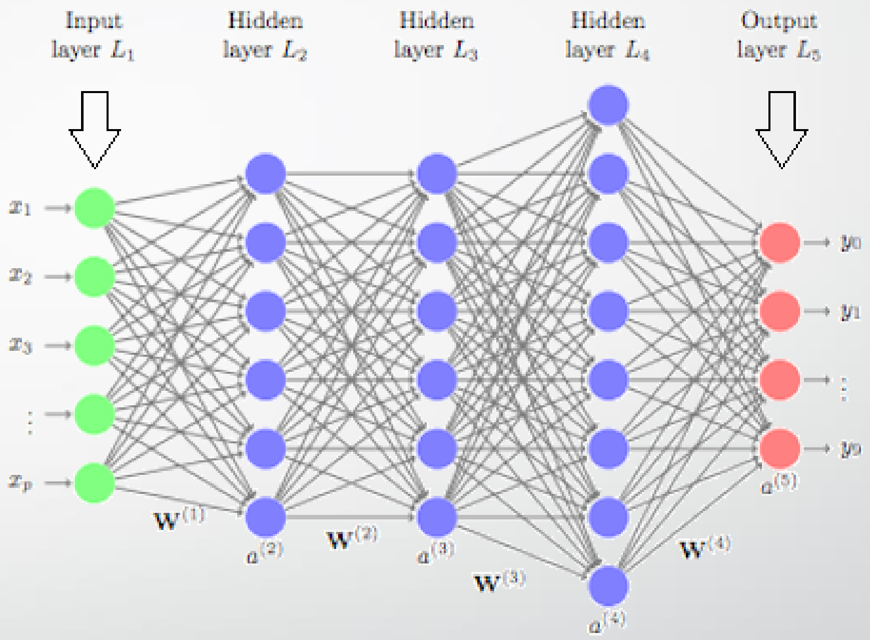
- A step towards understanding optimization of deep learning
- Convergence of a harder problem than classic supervised learning
- Algorithmic insights from the understanding

Supervised learning

- Sample: $\{x_i\}_{i=1}^{n}$,
- Label: $\{y_i\}_{i=1}^{n}$,
- Model: $f(w, x)$ where w denotes the parameter,
- Loss: $\ell(f(w, x), y)$,
- ERM: $\min_{w} \sum_{i=1}^{n} \ell(f(w, x_i), y_i)$.

Remark: Optimization is performed by optimizer such as gradient descent

Deep learning



- Use overparameterized deep neural network as model.

Optimization of Deep Learning

Theory:

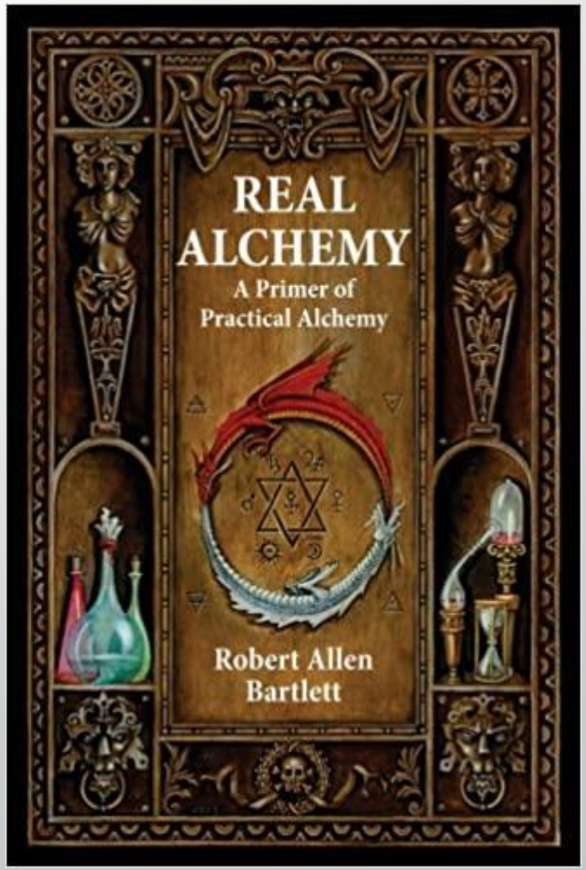
- Highly non-convex optimization problem
- Hard to get global convergence result

VS

Practice:

- Optimizing quite well
- Can be optimized to fit even random labels

How to understand optimization of deep learning (neural networks)?

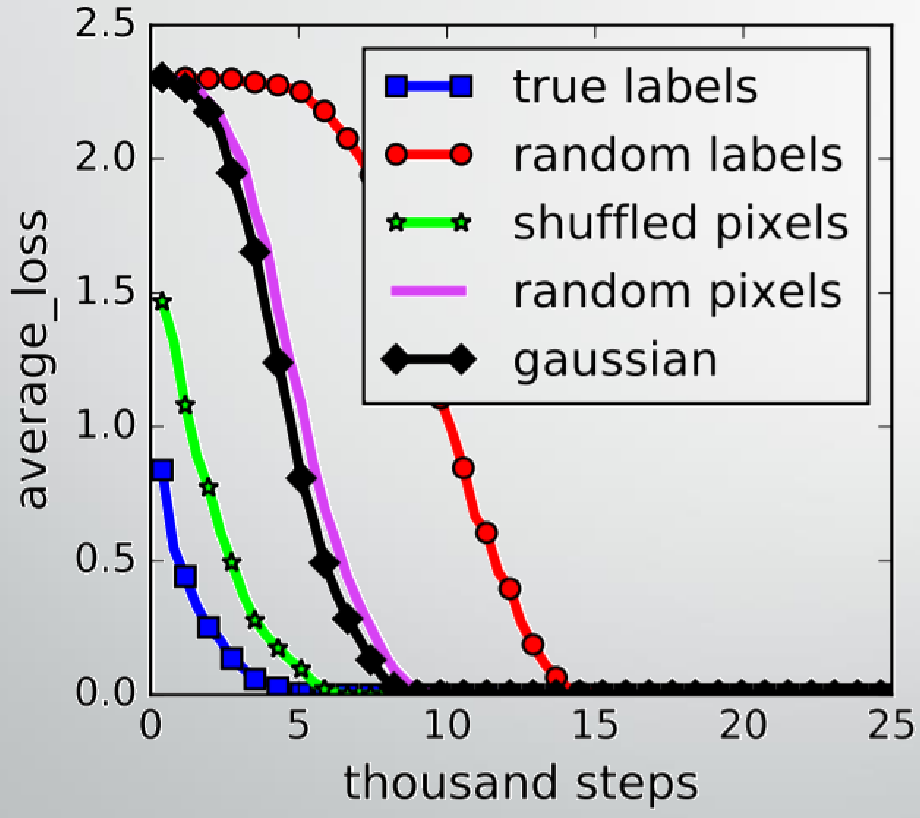


One attempt: through the lens of **overparameterization**

parameters ≫ # data

Effects of overparameterization:
Strong expressivity -> Easy to optimize

[Zhang et al. 2017]



(a) learning curves

- Overparameterized networks can approximate any function (in some certain sense).
- Overparameterized networks can be trained to overfit non-sense data.

Insights from overparameterization

- Overparameterized networks are very expressive.
- Maybe a small change to the parameter w is suffice to make the model fit the data.
- Only need to consider the optimization problem within a neighbor of the initial parameter w_0.

Inspired by these insights, we have …

- Overparameterized networks provably converge to zero training loss using gradient descent. [Jacot et al., 2018, Du et al., 2018, Allen-Zhu et al., 2018, Zou et al., 2018]
- Key idea:
  - Overparameterized networks are approximately **linear** w.r.t. the parameters in a neighbor of initialization (in the parameter space).
  - There is a global optima inside this neighbor.

Neural Taylor Expansion

Let $f(w, x)$ denote the network with parameter $w \in \mathbb{R}^m$ and input $x \in \mathbb{R}^d$. We assume the output of $f(w, x)$ is a scalar.

Neural Taylor Expansion/ Feature Map of Neural Tangent Kernel (NTK)

Within a neighbor of the initialization w_0, $\nabla_w f(w, x) \approx \nabla_w f(w_0, x)$. Then we have the approximate neural Taylor expansion [Chizat and Bach, 2018]:

$$f(w, x) \approx \underbrace{f(w_0, x)}_{\text{bias term}} + \underbrace{(w - w_0)}_{\text{linear parameters}} \cdot \underbrace{\nabla_w f(w_0, x)}_{\text{feature of } x},$$

where $\nabla_w f(w, x)$ is the gradient of $f(w, x)$ w.r.t. w.

Proof pipeline of the global convergence

- Overparameterized networks are approximately linear w.r.t. the parameters in a neighbor of initialization (in parameter space).

⇒ Gradient descent is applied within a "not so nonconvex" regime.

⇒ Gradient descent can approximately find the optima within this regime.

- There is a global optima inside this neighbor.

⇒ The optima reached by gradient descent is an (approximate) global optima :)

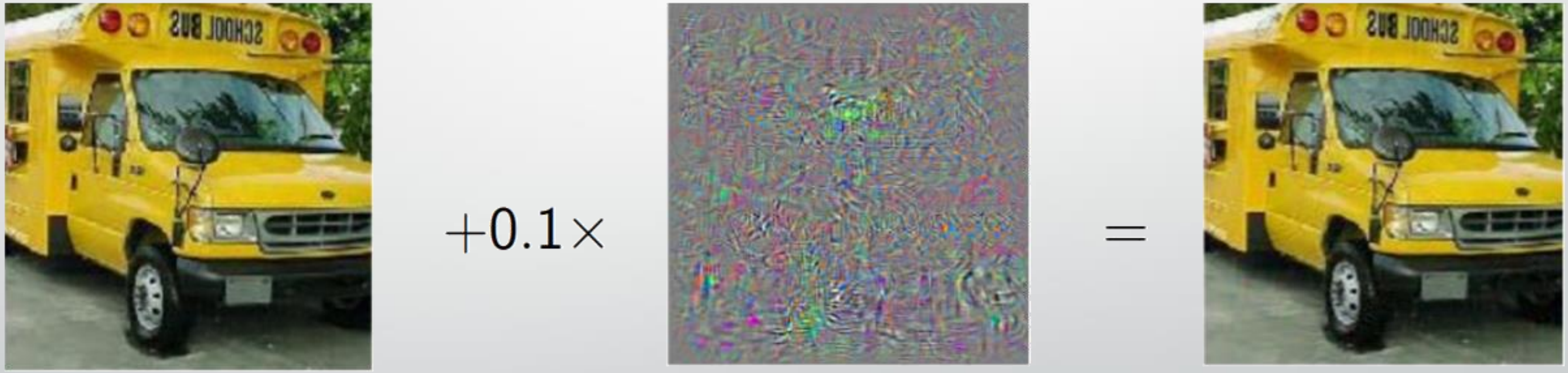
What's next?

- Global convergence of harder optimization problem,
- Better convergence rate for supervised learning.

Part I: harder optimization problem

- Appearance of adversary

Deep learning models are vulnerable to *adversarial attacks*.



(a) Schoolbus (b) Perturbation (c) Ostrich

Figure: Szegedy et al. (2014)

Adversarial attack

- Given: model $f(w, x)$, input data x,
- Adversarial attack find $A(w, x) \in B(x)$ where $B(x)$ is the allowed perturbation set at x, e.g. ℓ_2 or ℓ_∞ ball centered at x.

Algorithm to obtain robust model (w.r.t. adversarial attack)

- Adversarial training:

$$\min_{\mathrm{w}} \sum_{i=1}^{n} \ell(f(w, A(w, x_i), y_i)$$

that is, the loss evaluated at the perturbed data generated by A.

Convergence of adversarial training in overparameterized networks

- Can be non-smooth.
- Can be minmax optimization. (If the adversary can find the maxima within the perturbation set)

Our result: adversarial training can converge to global optima in overparameterized networks

- Still consider the optimization process in the neighbor of the initial parameter w_0.
- Main obstacles:
  - The optimization problem cannot be approximated by linear regression anymore,
  - The existence of optima of this harder problem is unclear.

We show that

- The optimization within the neighbor of initial weights can be approximated by a convex optimization problem. Analysis of this problem can guarantee to find an optimum in the neighbor;
- There existence a good optimum with near zero loss in the neighborhood, which is proved by random feature techniques.

Part II: Better convergence rate

- Prior work: gradient descent have linear convergence rate on overparameterized networks.

Theorem 4.1 (Convergence Rate of Gradient Descent). *Under the same assumptions as in Theorem 3.2, if we set the number of hidden nodes* $m = \Omega\left(\frac{n^6}{\lambda_0^4 \delta^3}\right)$, *we i.i.d. initialize* $\mathbf{w}_r \sim N(\mathbf{0}, \mathbf{I})$, $a_r \sim \text{unif}\left[\{-1, 1\}\right]$ *for* $r \in [m]$, *and we set the step size* $\eta = O\left(\frac{\lambda_0}{n^2}\right)$ *then with probability at least* $1 - \delta$ *over the random initialization we have for* $k = 0, 1, 2, \ldots$

$$\|\mathbf{u}(k) - \mathbf{y}\|_2^2 \leq \left(1 - \frac{\eta \lambda_0}{2}\right)^k \|\mathbf{u}(0) - \mathbf{y}\|_2^2 .$$

[Du et al. 2018]

Can we design algorithm that is provable faster?

Key insights:

- The optimization of overparameterized can be approximated by a neural tangent kernel regression problem (linear problem) in the neighbor of initial parameter.
- This approximate problem can be solved by explicit formula other than applying gradient descent.

Recall

Neural Taylor Expansion

Let $f(w,x)$ denote the network with parameter $w \in \mathbb{R}^m$ and input $x \in \mathbb{R}^d$. We assume the output of $f(w,x)$ is a scalar.

Neural Taylor Expansion/ Feature Map of Neural Tangent Kernel (NTK)

Within a neighbor of the initialization w_0, $\nabla_w f(w,x) \approx \nabla_w f(w_0,x)$. Then we have the approximate neural Taylor expansion [Chizat and Bach, 2018]:

$$f(w,x) \approx \underbrace{f(w_0,x)}_{\text{bias term}} + \underbrace{(w - w_0)}_{\text{linear parameters}} \cdot \underbrace{\nabla_w f(w_0,x)}_{\text{feature of } x},$$

where $\nabla_w f(w,x)$ is the gradient of $f(w,x)$ w.r.t. w.

Linear approximation at w_t

- $f(w,x) \approx f(w_t,x) + (w - w_t)\cdot \nabla_w f(w_t,x)$

Directly solve $f(w_{t+1},x_i) = y_i$,

We get

$$w_{t+1} = w_t - \left(J_t^\top J_t\right)^{-1} J_t^\top (f(w_t) - y),$$

where J_t is the Jacobian matrix, $f(w_t) = \left(f(w_t,x_1),\ldots,f(w_t,x_n)\right)^\top$ and $y = (y_1,\ldots,y_n)^\top$

Quadratic convergence rate

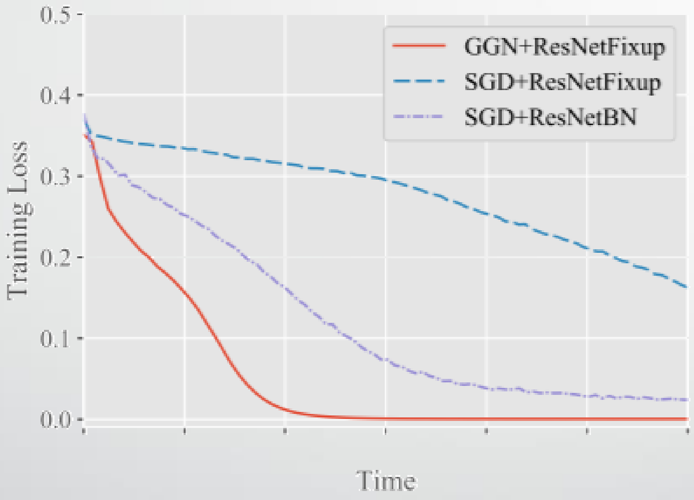
- Using update rule $w_{t+1} = w_t - \left(J_t^\top J_t\right)^{-1} J_t^\top (f(w_t) - y)$,
- We prove that the optimization of overparameterized network has a quadratic convergence rate.

Bonus

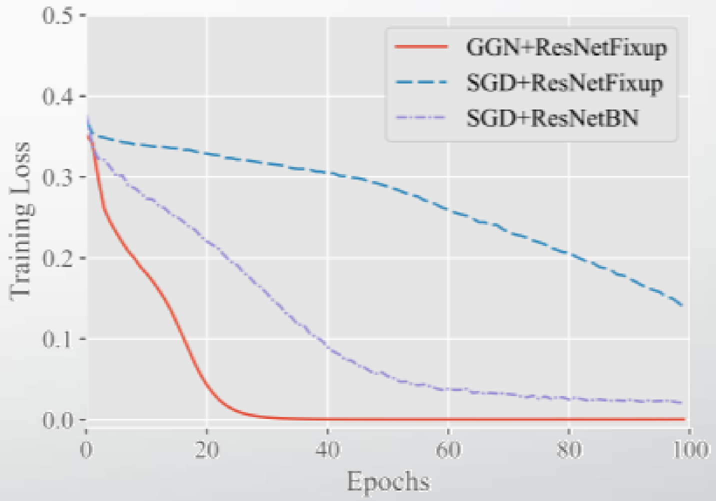
- The update rule $w_{t+1} = w_t \; - \left(J_t^\top J_t\right)^{-1} J_t^\top (f(w_t) - y)$ only involves the matrix $J_t^\top J_t$ whose size is #data times #data. (Comparing to classic Newton-type method that use (approximate) Hessian)
- This matrix is smaller than Hessian in overparameterized setting.
- Using mini-batch scheme can further reduce the size of matrix in the update.

Empirical results

We conduct experiments on two regression datasets, AFAD-LITE task (human age prediction by image) and RSNA Bone Age regression.



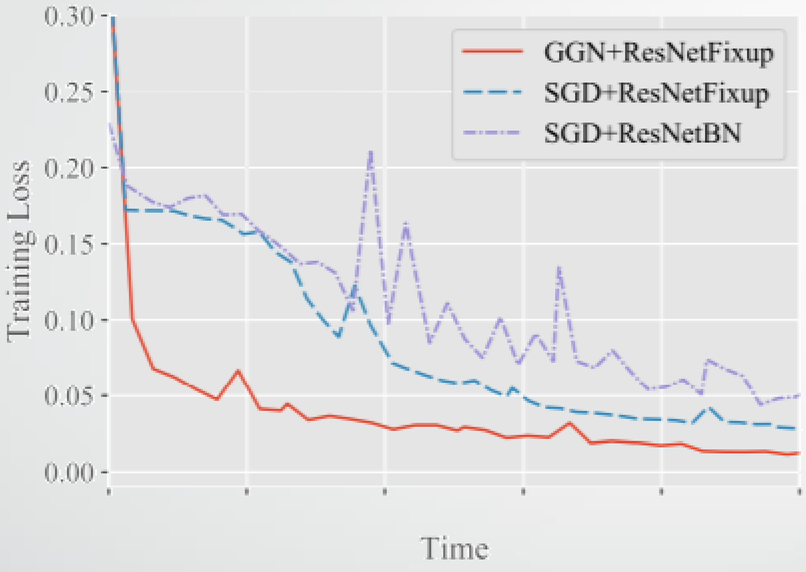
(a) Loss-time curve on AFAD-LITE



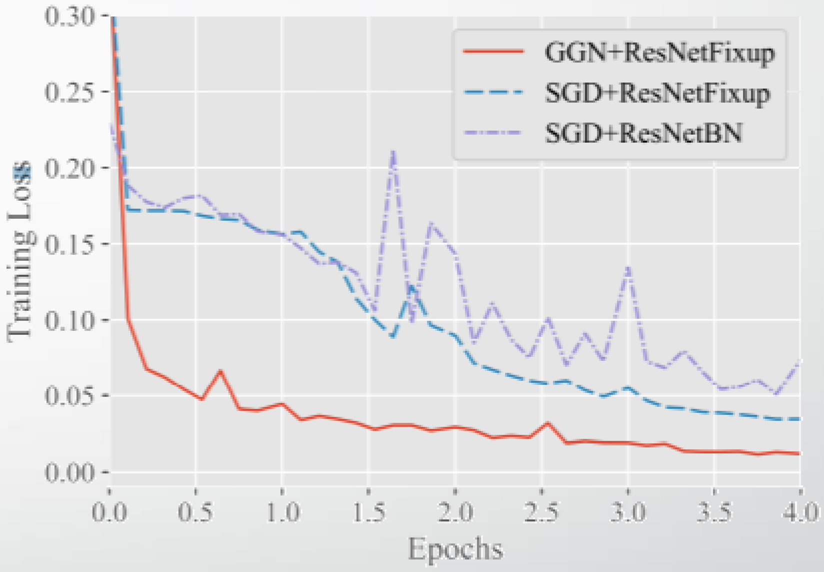
(b) Loss-epoch curve on AFAD-LITE

Figure: Training curves of GGN and SGD on two regression tasks.

Empirical results



(a) Loss-time curve on RSNA Bone Age



(b) Loss-epoch curve on RSNA Bone Age

Figure: Training curves of GGN and SGD on two regression tasks.

Thank you!

- Info: https://tianle.website/
- Contact: tianle.cai@princeton.edu