北京大学
PEKING UNIVERSITY

Deep Learning: From Theory to Algorithm

王立威

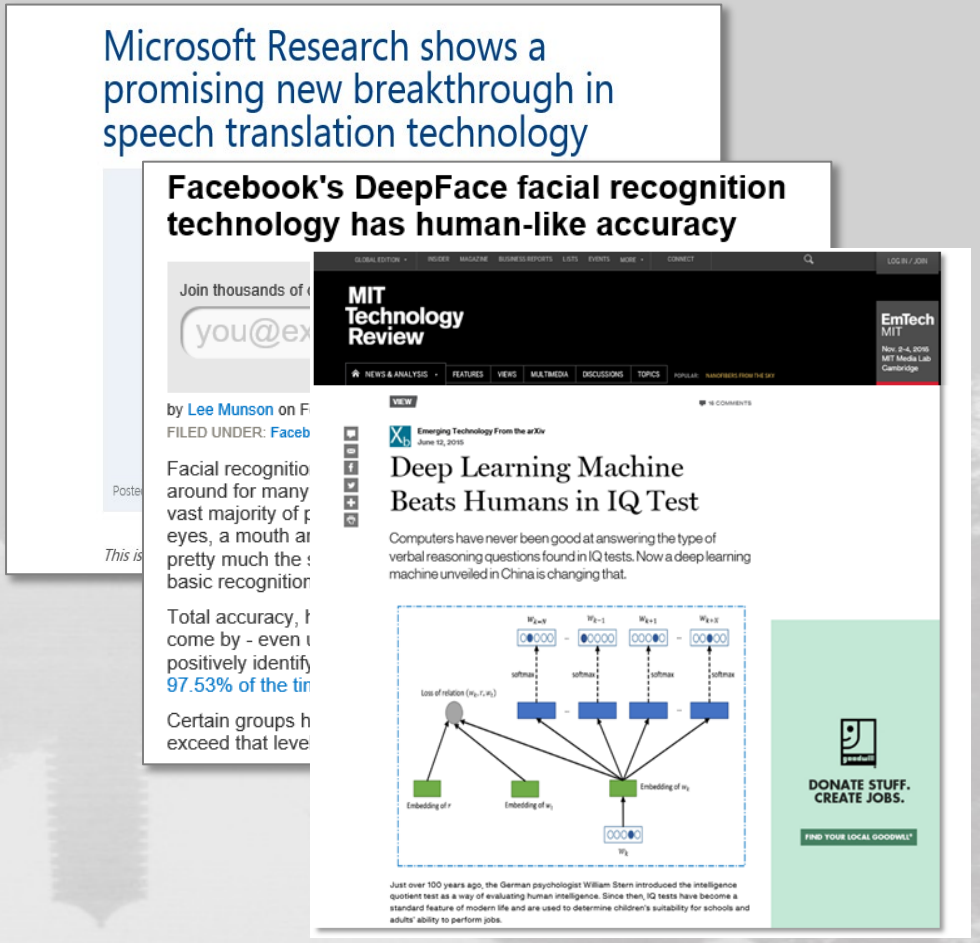
北京大学

Outline:

1. Overview of theoretical studies of deep learning

2. Optimization theory of deep neural networks
   1) Gradient finds global optima
   2) Gram-Gauss-Newton Algorithm

Success of Deep Learning

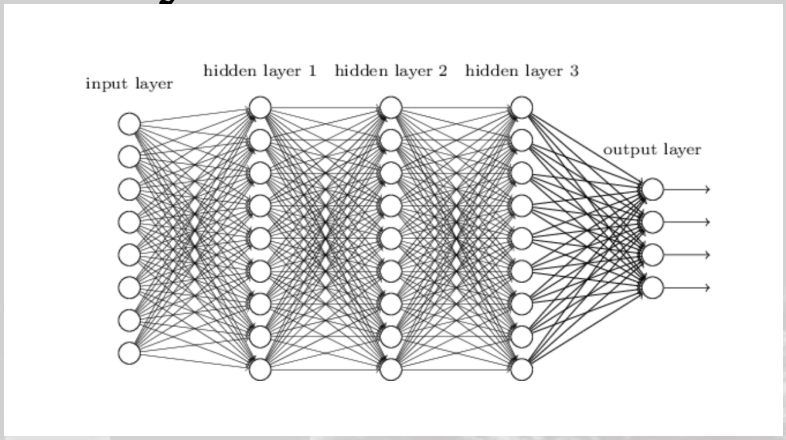


Mainly four areas:

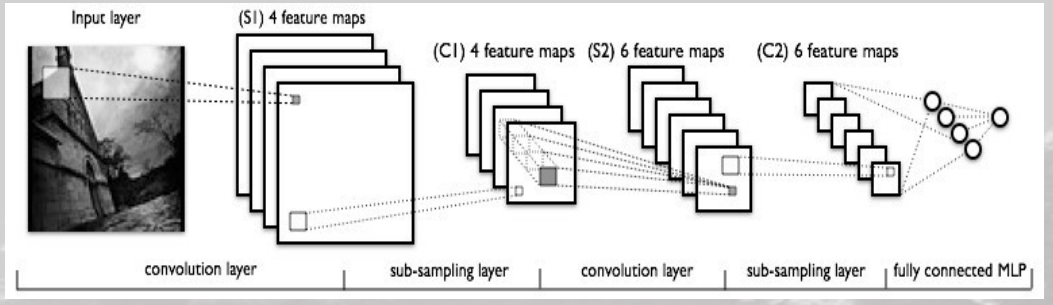
- Computer Vision
- Speech Recognition
- Natural Language Processing
- Deep Reinforcement Learning

Basic Network Structure

Fully Connected Network

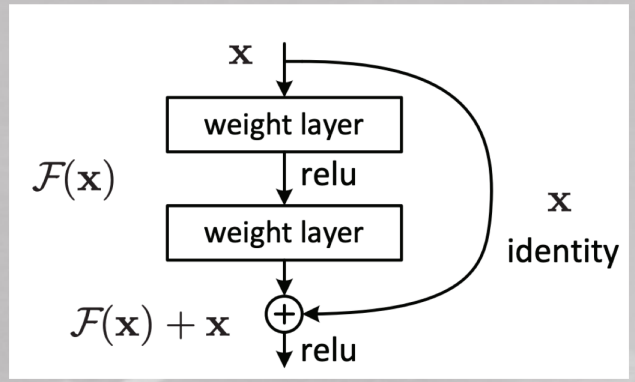


Convolutional Network



Further improvement:

Residual Network …
Recurrent Network (LSTM …)



Mystery of Deep Neural Network

For any kind of dataset, DNN achieves 0 training error easily.

Why do neural networks work so well?

A key factor:

Supervised Learning

Collect data: $\{(x_i, y_i) | \ i = 1, 2, \ldots, n\}$

Learn a model: f: $\mathcal{X} \rightarrow \mathcal{Y}$, $f \in \mathcal{H}$

Predict new data: $x \rightarrow f(x)$

All $(x, y) \sim \mathcal{D}$, where $\mathcal{D}$ is unknown

A common approach to learn:

ERM (Empirical Risk Minimization)

$$\min R_n(w) := \frac{1}{n} \sum_i l(w; x_i, y_i)$$

w: model parameters

$l(w; x, y)$: loss function w.r.t. data

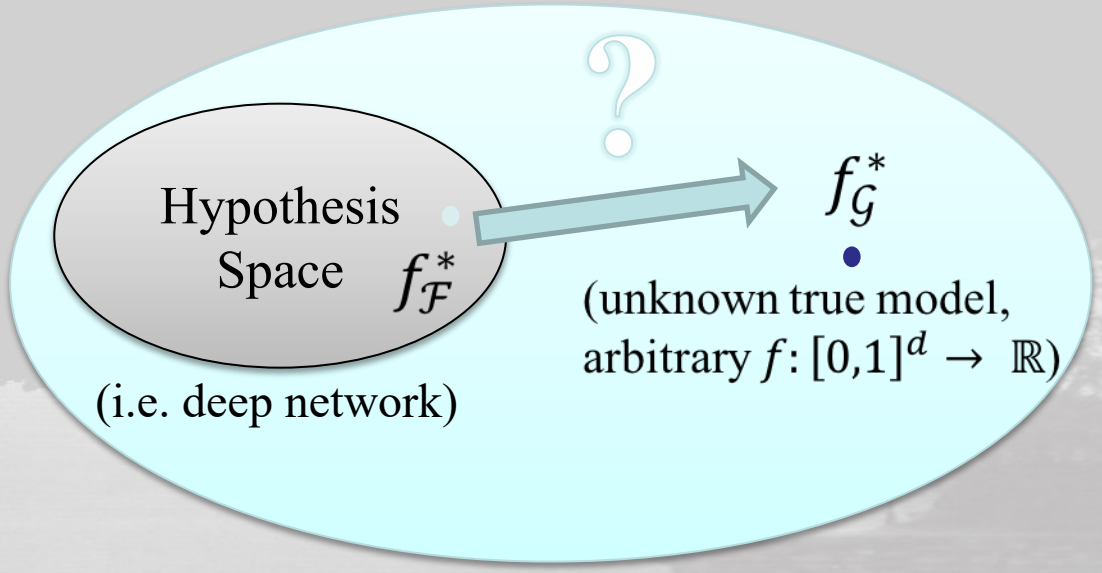
Population Risk: $R(w) := \mathrm{E}[l(w; x, y)]$

Theoretical Viewpoints of Deep Learning

- Model (Architecture)
  - CNN for images, RNN for speech…
  - Shallow (but wide) networks are universal approximator (Cybenko, 1989)
  - Deep (and thin) ReLU networks are universal approximator (LPWH**W**, 2017)
- Optimization on Training Data
  - Learning by optimizing the empirical loss, nonconvex optimization
- Generalization to Test Data
  - Generalization theory

Representation Power of DNN

Goal: find unknown true function



Universal Approximation Theorem

NN can approximate any continuous function arbitrarily well:

$$\forall f \in \mathcal{F}, \exists NN \in \mathcal{N}, s.t.$$
$$\forall x \in [0,1]^d, |NN(x) - f(x)| \leq \epsilon$$

1. Depth bounded (Cybenko, 1989)
2. Width bounded (LPWH**W**, 2017)

Issue: only show existence, ignore the algorithmic part

Cybenko, Approximation by superpositions of a sigmodial function, 1989

Lu et al. The Expressive Power of Neural Networks: A View from the Width, NIPS17

Some Observations of Deep Nets

- # of parameters >> # of data, hence easy to fit data
- Without regularization, deep nets also have benign generalization
- For random label or random feature, deep nets converge to 0 training error but without any generalization

How to explain these phenomena?

ICLR 2017 Best Paper:

"Understanding deep learning requires rethinking generalization"

Traditional Learning Theory Fails

Common form of generalization bound (in expectation or high probability)

$$R(w) \leq R_n(w) + \sqrt{\frac{Capacity\ Measure}{n}}$$

Capacity Measurement	Complexity
VC-dimension	$VC \leq O(\lvert E\rvert \log \lvert E\rvert)$
$\mathcal{E}$-Covering number	$\log_2 N_{l_1}(\mathcal{F}, \epsilon, m) \leq O\left(\frac{(AL_\phi)^{L(L+1)}}{\epsilon^{2L}}\right)$
Rademacher Average	$R_m(\mathcal{F}) \leq O(\mu^L)$

$|E|$: # of edges

L: # of layers

All these measurements are far beyond the number of data points!

Generalization of DL: margin theory

Bartlett et al. (NIPS17):

Main idea	Normalize Lipschitz constant (product of spectral norms of weighted matrices) by margin
Final bound	$\Pr\left[\arg\max_j F_{\mathcal{A}}(x)_j \neq y\right] \leq \widehat{\mathcal{R}}_\gamma(F_{\mathcal{A}}) + \widetilde{\mathcal{O}}\left(\frac{\|X\|_2 R_{\mathcal{A}}}{\gamma n}\ln(n)\ln(W) + \sqrt{\frac{\ln(1/\delta)}{n}}\right)$

where $R_{\mathcal{A}}$ is the spectral complexity

Remark:

(1) nearly has no dependence on # of parameters

(2) a multiclass bound, with no explicit dependence on # of classes

The Generalization Induced by SGD: Train faster generalize better

In nonconvex case, there are some results, but very weak

Hardt et al. (ICML15) , for SGD:

Assuming Lipschitz and β-smooth, then

$$\varepsilon_{stab} \leq O\left(\frac{T^{1-\frac{1}{\beta c+1}}}{n}\right)$$

which maybe linear dependent on training iterations

Our Results

From the view of stability theory:

Under mild conditions of (surrogate) loss function, the generalization error of SGLD at N-th round satisfies

$$E[l(w_S, z)] - E_S[l(w_S, z)] \leq O\left(\frac{1}{n}\left(k_0 + L\sqrt{\beta \sum_{k=k_0+1}^{N} \eta_k}\right)\right)$$

where L is the Lipschitz constant, and $k_0 := \min\{k: \eta_k \beta L^2 < 1\}$

If consider high probability form, there is an additional $\tilde{O}\left(\sqrt{1/n}\right)$ term

Our Results

From the view of PAC-Bayesian theory:

For regularized ERM with $R(w) = \lambda||w||^2/2$. Under mild conditions, with high probability, the generalization error of SGLD at N-th round satisfies

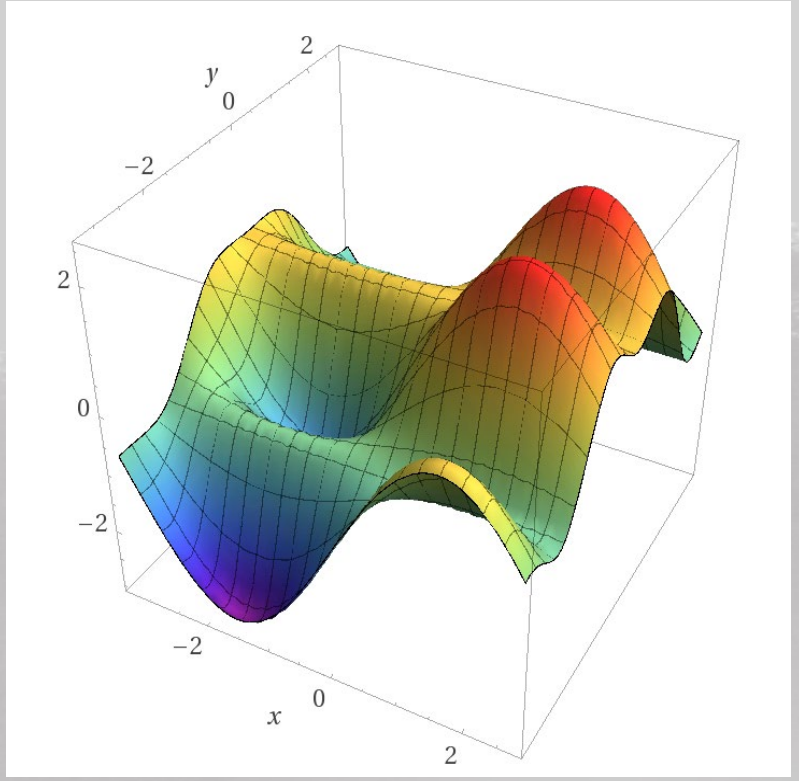
$$E[E[l(w_S, z)]] - E_S[E[l(w_S, z)]] \leq O\left(\sqrt{\frac{\beta}{n}\sum_{k=1}^{N}\eta_k e^{-\lambda(T_N - T_k)/2} E[||g_k||^2]}\right)$$

where $T_k = \sum_{j=1}^{k}\eta_k$, g_k is the stochastic gradient in each round.

Optimization for Deep Neural Network

- Loss functions for DNN is highly non-convex
- Common SG methods (such as SGD) work well

What‘s the reason behind above facts?



Our Results (DLLWZ, 2019)

Notations:

n: the number of training data; L: network depth; $R_n(w)$: empirical square loss;
λ_0: some constant that depends on network structure and initialization scheme;
w_t: network parameters obtained by GD

Theorem: For fully connected network with smooth acti

$$if\ width\ m = poly\left(n, 2^L, \frac{1}{\lambda_0}\right)\ and\ step\ size\ \eta = O\left(\frac{\lambda_0}{n^2 2^L}\right),$$

then with high probability, there is

$$R_n(w_t) \leq (1-\eta\lambda_0)^t R_n(w_0)$$

GD finds global minima in a linear convergence rate!

Our Results (DLLWZ, 2019)

Theorem: For ResNet or Convolutional ResNet with smooth activation, if width $m = poly\left(n, L, \frac{1}{\lambda_0}\right)$, *and step sitze* $\eta = O\left(\frac{\lambda_0}{n^2}\right)$, *then with high probability there is*

$$R_n(w_t) \leq (1 - \eta\lambda_0)^t R_n(w_0)$$

Note there is an exponential improvement about the network width compared with fully connected network!

Du et al. Gradient Descent Finds Global Minima of Deep Neural Networks, ICML19

Concurrent Results

Concurrently, Allen-Zhu et al. [1] and Zou et al. [2] proved (Stochastic) GD converges to global optimum under some similar but a little different assumptions.

When width of network is infinite, gradient descent converges to the solution of a kernel regression, which is characterized by Neural Tangent Kernel (NTK) [3].

$$\text{NTK: } K(x, x') = \langle \nabla_w f(w, x), \nabla_w f(w, x') \rangle$$

[1] Allen-Zhu et al. A convergence theory for deep learning via over-parameterization

[2] Zou et al. Stochastic gradient descent optimizes over-parameterized deep ReLU networks

[3] Jacot and Gabriel, Neural Tangent Kernel: convergence and generalization in neural networks (NIPS18)

Critical Facts

1. There is a global optimum inside this neighbor.

2. $f(w, x_i)$ is approximately linear w.r.t w in a neighbor of initialization, which is implicitly implied in the proof of GD;

Can we design faster algorithm than (stochastic) GD?

Second Order Algorithm for DNN

In classic convex optimization, second order algorithm achieves much faster convergence rate.

Main idea: use second order information (Hessian matrix) to accelerate training at the price of additional computational cost.

Second order algorithm for DNN is much more challenging:

1. Loss function is highly non-convex;
2. High dimensional parameter space (which is usually ignored in classic convex optimization).

Classic Gauss-Newton Method

Non-linear least square:

$$\min_w F(w) := \frac{1}{2}\sum(f(w, x_i) - y_i)^2$$

First order approximation at w_t:

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

$$w_{t+1} = argmin_w \frac{1}{2}||f_t + J_t(w - w_t) - y_t||_2^2$$

Notation:

$$f_t := \left(f(w_t, x_1), \ldots, f(w_t, x_n)\right)^T$$

$$J_t := \left(\nabla_w f(w_t, x_1), \ldots, \nabla_w f(w_t, x_n)\right)^T \quad \text{(Jacobian)}$$

$$H_t = J_t^T J_t \qquad y_t := (y_1, \ldots, y_n)^T$$

$$w_{t+1} = w_t - H_t^{-1} J_t^T (f_t - y)$$

Approximation of Newton's method
(using H_t to approximate Hessian)

Potential Issues

For DNN, parameter dimension $d \gg n$

$J_t \in \mathbb{R}^{n \times d}, \quad H_t = J_t^T J_t \in \mathbb{R}^{d \times d}$

1. Approximate Hessian H_t is too large to be stored;
2. H_t is not invertible;
3. Computational complexity may be expansive compared with SGD.

Key Observation

Theorem: *Let* $J \in \mathbb{R}^{n\times d}, r \in \mathbb{R}^n$, *where* $d \gg n$ *and* J *has full rank, then* $\Delta w := -J^T(JJ^T)^{-1}r$ *is a solution of the equation*

$$J^T J\Delta w = -J^T r$$

with minimum norm $\left|\left|\Delta w\right|\right|_2^2$.

$$w_{t+1} = w_t - H_t^{-1}J_t^T(f_t - y) \quad \Rightarrow \quad w_{t+1} = w_t - J_t^T(J_tJ_t^T)^{-1}(f_t - y)$$

$J_tJ_t^T \in \mathbb{R}^{n\times n}$: Gram matrix of NTK $K(x,x') = \langle \nabla_w f(w,x), \nabla_w f(w,x')\rangle$

Gram-Gauss-Newton (GGN) Algorithm

Mini-batch extension: $$w_{t+1} = w_t - J_{t,B_t}^T \left(J_{t,B_t} J_{t,B_t}^T\right)^{-1} (f_{t,B_t} - y_{B_t})$$

Stable version: $$w_{t+1} = w_t - J_{t,B_t}^T \left(\lambda J_{t,B_t} J_{t,B_t}^T + \alpha I\right)^{-1} (f_{t,B_t} - y_{B_t})$$

For each iteration do

1. Sample a mini-batch B_t from dataset
2. Calculate Jacobian matrix J_{t,B_t} and Gram matrix $G_{t,B_t} = J_{t,B_t} J_{t,B_t}^T$
3. Update $w_{t+1} = w_t - J_{t,B_t}^T \left(\lambda G_{t,B_t} + \alpha I\right)^{-1} (f_{t,B_t} - y_{B_t})$

Computational Complexity

Space complexity:

Jacobian matrix J_{t,B_t}: $b \times d$, where b is the batch size

Gram matrix G_{t,B_t}: $b \times b$

Time complexity:

Mainly compute Gram matrix G_{t,B_t} and its inverse

$O(b^2 d + b^3)$

Compared with SGD:

Nearly the same computational cost, except keeping track of the derivative of every data point in mini-batch, instead of their average in SGD.

Theoretical Guarantee (CGHHW)

Theorem: For two layer ReLU network, if width $m = poly\left(\frac{n}{\lambda_0}\right)$, *then with hig*

1. *Gram matrix* G_t *at each iteration is invertible;*
2. *Empirical square loss converges to* 0:

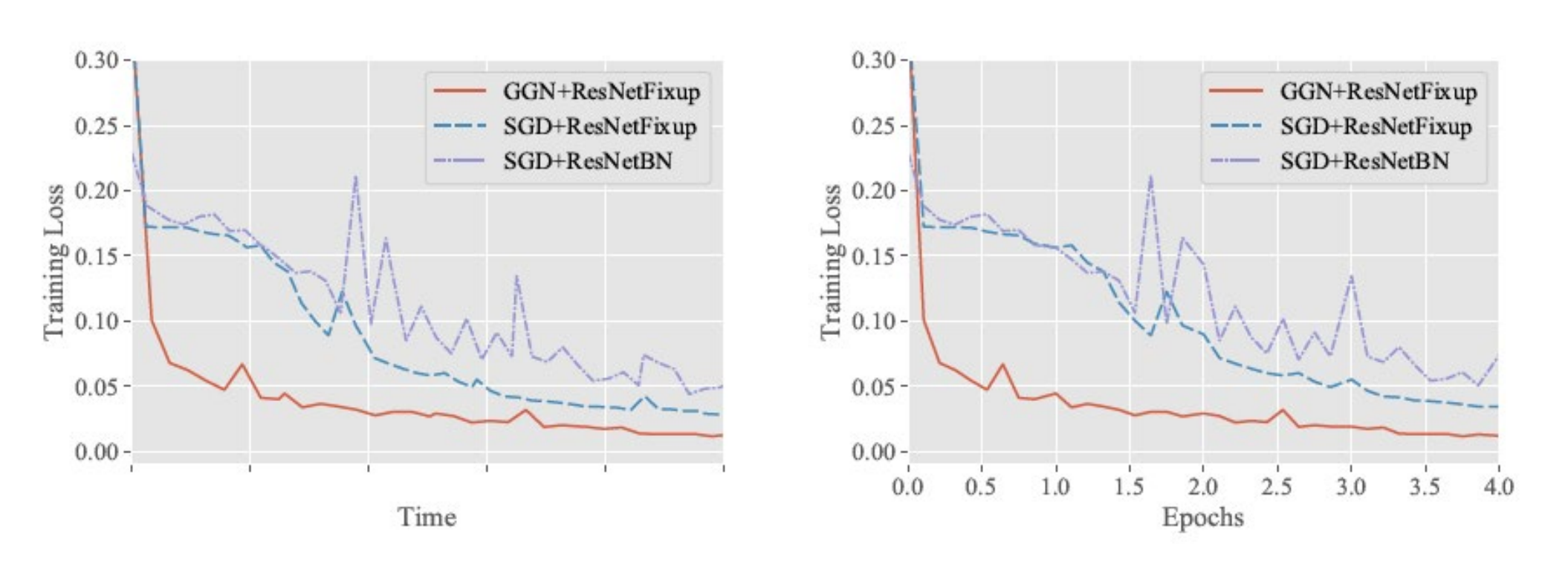
$$R_n(w_{t+1}) \leq \frac{Cn}{m} R_n(w_t)^2$$

1. Quadratic convergence;
2. Conclusion holds for general networks like GD.

Cai et al. A Gram-Gauss-Newton Method Learning Over-Parameterized Deep Neural Networks for Regression Problems, Arxiv19

Experiments

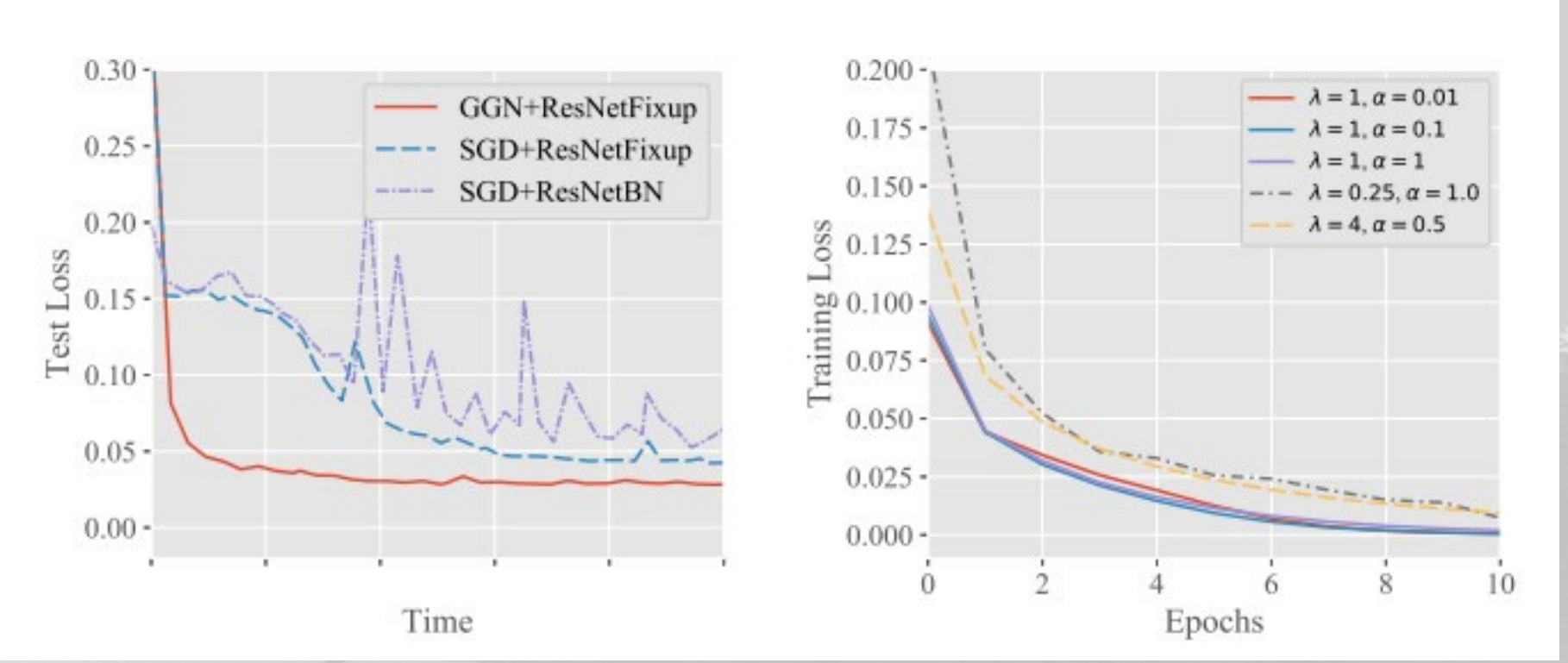
RSNA Bone Age task: predicting bone age by images



(a) Loss-time curve

(b) Loss-epoch curve

Experiments



(a) Test performance

(b) Training with different hyper-parameters

Take-aways

- We prove Gradient Descent achieves global optimum in a linear convergence rate for general over-parametrized neural network.

- We propose a novel quasi second order algorithm (GGN) for training network, which converges in quadratic order for general over-parametrized neural network and enjoys nearly the same computational complexity as SGD for regression task.

Related Paper

1. Gradient Descent Finds Global Minima of Deep Neural Networks, ICML19
2. A Gram-Gauss-Newton Method Learning Overparameterized Deep Neural Networks for Regression Problems, Arxiv, 2019
3. Lu et al. The Expressive Power of Neural Networks: A View from the Width, NIPS17

Thank you!