Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer NNs

Sanjeev Arora  
Princeton & IAS

Simon S. Du  
CMU

Wei Hu  
Princeton

Zhiyuan Li  
Princeton

Ruosong Wang  
CMU
“Rethinking generalization” Experiment  [Zhang et al ‘17]

True Labels:  2  1  3  1  4

Random Labels:  5  1  7  0  8
“Rethinking generalization” Experiment [Zhang et al ‘17]

Unexplained phenomena

① SGD achieves nearly 0 training loss for both correct and random labels (overparametrization!)

② Good generalization with correct labels

③ Faster convergence with correct labels than random labels.
“Rethinking generalization” Experiment [Zhang et al ‘17]

Unexplained phenomena

① SGD achieves nearly 0 training loss for both correct and random labels (overparametrization!)

② Good generalization with correct labels

③ Faster convergence with correct labels than random labels.

No good explanation in existing generalization theory:

generalization gap \leq \sqrt{\frac{\text{model complexity}}{\# \text{training samples}}}

1. SGD achieves nearly 0 training loss for both correct and random labels (overparametrization!).
2. Good generalization with correct labels.
3. Faster convergence with correct labels than random labels.

No good explanation in existing generalization theory:

generalization gap \leq \sqrt{\frac{\text{model complexity}}{\# \text{training samples}}}

Experiment [Zhang et al ‘17]
“Rethinking generalization” Experiment [Zhang et al ‘17]

Unexplained phenomena

① SGD achieves nearly 0 training loss for both correct and random labels (overparametrization!)

② Good generalization with correct labels

③ Faster convergence with correct labels than random labels.

No good explanation in existing generalization theory:

generalization gap \leq \sqrt{\frac{\text{model complexity}}{\# \text{ training samples}}}

This paper: Theoretical explanation for overparametrized 2-layer nets using label properties
Setting: **Overparam** Two-Layer ReLU Neural Nets

**Unexplained phenomena**

1. SGD achieves nearly 0 training loss for both correct and random labels (overparametrization!)

2. Good generalization with correct labels

3. Faster convergence with correct labels.

- Overparam: # hidden nodes is large
- Training obj: $\ell_2$ loss, binary classification
- Init: i.i.d. Gaussian
- Opt algo: GD for the first layer, $W$
Setting: **Overparam** Two-Layer ReLU Neural Nets

Unexplained phenomena

1. SGD achieves nearly 0 training loss for both correct and random labels (overparametrization!)
2. Good generalization with correct labels
3. Faster convergence with correct labels.

[Du et al., ICLR’19]:
GD converges to 0 training loss
Explains phenomenon ①, but not ② or ③

Overparam: # hidden nodes is large
Training obj: $\ell_2$ loss, binary classification
Init: i.i.d. Gaussian
Opt algo: GD for the first layer, $W$
**Setting:** Overparam Two-Layer ReLU Neural Nets

**Unexplained phenomena**

1. **SGD** achieves nearly 0 training loss for both correct and random labels (overparametrization!)
2. **Good generalization** with correct labels
3. **Faster convergence** with correct labels.

[Du et al., ICLR’19]:
GD converges to 0 training loss
Explains phenomenon ①, but not ② or ③

**This paper: for ② and ③**
- Faster convergence with true labels
- A data-dependent generalization bound (distinguish random labels from true labels).

\[ x \rightarrow f(W, x) \]

Overparam: # hidden nodes is large
Training obj: \( \ell_2 \) loss, binary classification
Init: i.i.d. Gaussian
Opt algo: GD for the first layer, \( W \)
Training Speed

Theorem:

\[
\text{loss( iteration } k \text{ )} \approx \left\| (I - \eta H)^k \cdot y \right\|^2
\]

- \( y \): vector of labels
- \( H \): kernel matrix ("Neural Tangent Kernel"),

\[
H_{ij} = E_W \left\langle \nabla_{\!W} f(W, x^{(i)}), \nabla_{\!W} f(W, x^{(j)}) \right\rangle = \frac{\pi - \arccos(x_i^T x_j)}{2\pi} x_i^T x_j
\]
Training Speed

Theorem:

\[ \text{loss(\text{iteration } k)} \approx \| (I - \eta H)^k \cdot y \|^2 \]

- \( y \): vector of labels
- \( H \): kernel matrix ("Neural Tangent Kernel"),
  \[ H_{ij} = E_W \langle \nabla_W f(W, x^{(i)}), \nabla_W f(W, x^{(j)}) \rangle = \frac{\pi - \arccos(x_i^T x_j)}{2\pi} x_i^T x_j \]

Implication:

- Training speed determined by projections of \( y \) on eigenvectors of \( H \): \( \langle y, v_1 \rangle, \langle y, v_2 \rangle, \langle y, v_3 \rangle, \ldots \)
- Components on top eigenvectors converge to 0 faster than components on bottom eigenvectors

Explains different training speeds on correct vs random labels
Explaining Generalization despite vast overparametrization

**Theorem:** For 1-Lipschitz loss, 
\[
\text{test error} \leq \sqrt{\frac{2y^\top H^{-1}y}{\# \text{ training samples}}} + \text{small terms}
\]

**Corollary:** Simple functions are provably learnable (eg, linear function and even-degree polynomials).
Explaining Generalization despite vast overparametrization

**Theorem:** For 1-Lipschitz loss,

\[
\text{test error} \leq \sqrt{\frac{2y^\top H^{-1}y}{\# \text{ training samples}}} + \text{small terms}
\]

**Corollary:** Simple functions are provably learnable (e.g., linear function and even-degree polynomials).

Poster #75 tonight
Explaining Generalization despite vast overparametrization

**Theorem:** For 1-Lipschitz loss, 
\[
\text{test error} \leq \sqrt{\frac{2y^\top H^{-1}y}{\# \text{ training samples}}} + \text{small terms}
\]

“Distance to Init”

“Min RKHS norm for training labels”

“data dependent complexity”

**Corollary:** Simple functions are provably learnable (eg, linear function and even-degree polynomials).

Poster #75 tonight