Towards Understanding the Role of Over-Parameterization in Generalization of Neural Networks

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, Nathan Srebro

Presented by,
Tanmayee Joshi, Sean Chung
Exploring Generalization in Deep Learning

• Overparameterization:
  ○ Traditional wisdom vs empirical evidence
  ○ Improved generalization performance even without regularization
  ○ Easily fit random labels

• Different Complexity Measures:
  ○ VC bounds, norm-based, sharpness, PAC-Bayes etc.

• Study of how these measures can ensure generalization
  ([Neyshabur et al](https://example.com))
Complexity Measures: Expectation

- Sufficient to ensure generalization
- Low complexity for the networks learned in practice
- Networks learned using real labels to have lower complexity than the ones using random labels
- Complexity to decrease when # hidden units increased
- Correlation between complexity measure and generalization ability for zero-training error models
Outcomes of the study

• Generalization behavior remains unexplained

• Some combination of expected sharpness and norms can explain the phenomenon, but still unclear

• Unresolved relationship between optimization and implicit regularization
Novel Complexity Measure: Inspiration

- Closeness between **learned weights** to initialization
  - Extreme setting: #hidden units go to infinity [Bengio et al, Bach et al]

- Large number of parameters represent all possible features for
  the optimization problem to select the right feature

- Over-parameterization reduces optimization algorithms’ work
  - Less work in tuning the weights
Setup

- Use two-layer ReLU network
  - Simplify the architecture while maintaining the property of interest
Setup

- Representation of two-layer neural network
  - Input dimension $d$, output dimension $c$ and the hidden units $h$
  - Output of the network: $f_{V,U}(x) = V[Ux]_+$ where $x \in \mathbb{R}^d$, $U \in \mathbb{R}^{h \times d}$, $V \in \mathbb{R}^{c \times h}$
- Margin operator,
  \[ \mu(f(x), y) = f(x)[y] - \max_{i \neq y} f(x)[i] \]
Setup

• Ramp loss: \((\gamma \text{ is margin } > 0)\)

\[
\ell_\gamma(f(x), y) = \begin{cases} 
0 & \mu(f(x), y) > \gamma \\
\frac{\mu(f(x), y)}{\gamma} & \mu(f(x), y) \in [0, \gamma] \\
1 & \mu(f(x), y) < 0.
\end{cases}
\]

• Expected margin loss: \(L_\gamma(f) = E_{(x,y) \sim D}[\ell_\gamma(f(x), y)]\)

• Empirical estimate: \(\hat{L}_\gamma(f) = \frac{1}{m} \sum_{i=1}^{m} \ell_\gamma(f(x_i), y_i)\)

• Write \(L_0(f), \hat{L}_0(f)\) as expected risk and training error respectively
Setup

• Rademacher complexity:
  ○ \[ R_S(H) = \frac{1}{m} \mathbb{E}_{\xi \sim \{\pm 1\}^m} \left[ \sup_{f \in H} \sum_{i=1}^{m} \xi_i f(x_i) \right] \]
  ○ Captures the ability of functions to fit random labels
  ○ Increases as the complexity of the class increases
  ○ \( S \): training set
  ○ \( f \): function in the function class \( H \)
  ○ \( m \): number of input examples
  ○ \( \xi \): Rademacher random variables
Empirical Investigation

- Observations for the first layer:
  - Frobenius distance $= \|U - U_0\|_F$
Empirical Investigation

- Observations for the first layer:
  - Per-unit distance to initialization ↓ with ↑ in #parameters
  - Distribution of angles from orthogonal to aligned
Empirical Investigation

- **Unit capacity** of hidden unit $i$:
  - Per unit distance to initialization
    - $\beta_i = \|u_i - u_i^0\|_2$
Empirical Investigation

- Observations for second layer:
  - Both Frobenius & distance decrease \(\rightarrow\) initialization has little impact
  - Impact of each classifier on final decision is shrinking at rate \(\geq \frac{1}{\sqrt{h}}\)
    - View each hidden unit as a linear separator
    - View the top layer as an ensemble over classifiers
Empirical Investigation

- **Unit impact** of hidden unit \(i\),
  - Magnitude of outgoing weights from unit \(i\)
  - \(\alpha_i = \|v_i\|_2\)
Setup: Reducing the Function Class Size

- Combining unit impact and capacity, the restricted set of parameter:

\[ W = \{ (V, U) \mid V \in \mathbb{R}^{c \times h}, U \in \mathbb{R}^{h \times d}, \|v_i\| \leq \alpha_i, \|u_i - u_i^0\|_2 \leq \beta_i \} \]

- Hypothesis class of neural networks:

\[ F_W = \{ f(x) = V[Ux] \mid (V, U) \in W \} \]
Generalization bound

- From previous work (Mohri et al),
  - With probability \(1 - \delta\), following generalization bound holds for any \(f \in H\):
    \[
    L_0(f) \leq \hat{L}_\gamma(f) + 2\mathcal{R}(\ell_\gamma \circ \mathcal{H}) + 3\sqrt{\frac{\ln(2/\delta)}{2m}}
    \]
  - Therefore, need to bound the Rademacher complexity
Generalization Bound

• Intuition:
  ○ Previous works decompose the complexity of the network into the complexity of layers
  ○ Issue?
    • Ignore the linear structure of each individual layer
  ○ Solution
    • Calculated by decomposing to complexity of hidden units
Generalization Bound

Theorem 1: Given a training set $S = \{x_i\}_{i=1}^m$ and $\gamma > 0$, following bound is derived for the Rademacher complexity on loss $\ell_\gamma$ and class $\mathcal{F}_\mathcal{W}$ ($\alpha, \beta$ fixed before training):

$$\mathcal{R}_S(\ell_\gamma \circ \mathcal{F}_\mathcal{W}) \leq \frac{2\sqrt{2c} + 2}{\gamma m} \sum_{j=1}^h \alpha_j (\beta_j \|X\|_F + \|u_j^0 X\|_2)$$
Generalization Bound

Theorem 2: for any $h \geq 2$, $\gamma > 0$, $\delta \in (0, 1)$ and previously defined settings, with probability $1 - \delta$,

$$L_0(f) \leq \hat{L}_{\gamma}(f) + \tilde{O}
\left(
\sqrt{c \left\| V \right\|_F (\left\| U - U^0 \right\|_F + \left\| U^0 \right\|_2)} \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left\| x_i \right\|_2^2} + \sqrt{\frac{h}{m}}
\right)$$

Selected parts are associated with the number of hidden units (of interest)
Recall: Empirical Investigation
Generalization Bound

- Constructed measure from the generalization bound

\[ \tilde{\Theta} \left( \| U_0 \|_2 \| V \|_F + \| U - U^0 \|_F \| V \|_F + \sqrt{h} \right) \]
Comparison with Other Measures

| #  | Reference            | Measure                                           |
|----|----------------------|---------------------------------------------------|
| (1) | Harvey et al. 2017   | $O(dh)$                                           |
| (2) | Bartlett and Mendelson 2002 | $\tilde{O}\left(\|U\|_{\infty,1} \|V\|_{\infty,1}\right)$ |
| (3) | Neyshabur et al. 2015 | $O\left(\|U\|_F \|V\|_F\right)$                   |
| (4) | Bartlett et al. 2017  | $\tilde{O}\left(\|U\|_2 \|V - V_0\|_{1,2} + \|U - U_0\|_{1,2} \|V\|_2\right)$ |
| (5) | Neyshabur et al. 2018 | $\tilde{O}\left(\|U\|_2 \|V - V_0\|_F + \sqrt{h} \|U - U_0\|_F \|V\|_2\right)$ |
| (6) | ours                 | $\tilde{O}\left(\|U_0\|_2 \|V\|_F + \|U - U^0\|_F \|V\|_F + \sqrt{h}\right)$ |
Comparison with Other Measures
Comparison of Normalized Capacity
Experimental Comparison
Lower Bound

• The lower bound derived under certain set of assumptions implies that the upper bound for Rademacher complexity is actually tight!
Contribution

- Prove tighter generalization bounds on two-layer ReLU
- Proposed complexity measure could explain the effect of over-parameterization on generalization of neural networks
- Improved lower bound than the best existing result
Limitations and Future Work

• Results are based only on two-layer networks
• Still a very loose bound
  ○ Larger than the number of training examples
• No answer to whether optimization algorithms converge to low complexity networks
• Effect of the choice of different hyperparameters on complexity
• Implicit regularization in optimization algorithms
Discussion: Quiz Questions

1. What are the reasons suggested by the authors that could explain why over-parameterization improves generalization error?
   - Lower difference between initial and final weights
   - Faster convergence
   - More features are included
   - Less likely to be overfitting
   - Impact of each parameter of the last layer for the final decision is less influential as we increase the number of hidden units
Discussion: Quiz Questions

2. Which of the following statements are correct?
   ○ Previous works decompose the complexity of the network into that of the hidden units
   ○ Prior work mentioned in the paper is able to show that over-parameterization improves generalization error
   ○ The rate of outgoing weights from a parameter diminishes is faster than $\sqrt{h}$
   ○ It is easier to use 2-layer ReLU to show the feature that over-parameterization improves generalization error
   ○ The angle between initial and trained weights in first layer becomes orthogonal as we increase parameters
Discussion: Quiz Questions

3. Which of the following is observed for a complexity measure that explains the generalization in deep learning?
   ○ It should increase with the increase in number of hidden units
   ○ **It should decrease with the increase in number of hidden units**
   ○ Margin normalized by the measure should be higher for random labels than the true labels making it a harder problem
   ○ **Margin normalized by the measure should be higher for true labels than random labels**
Thank You!