Understanding Deep Learning Requires Rethinking Generalization[1]

Presenter: Zhengze Zhou

Cornell University

zz433@cornell.edu

February 21, 2019

[1] based on [Zhang et al., 2016]

Quiz

1. Which of the following can not be viewed as a form of regularization?
    - A. Data augmentation
    - B. Dropout
    - C. Early stopping
    - D. Empirical risk minimization
2. Which of the following network architecture is not mentioned in the paper?
    - A. AlexNet
    - B. LeNet-5
    - C. Inception
    - D. MLP

Overview

Overfitting in Deep Learning

Bias Variance Tradeoff

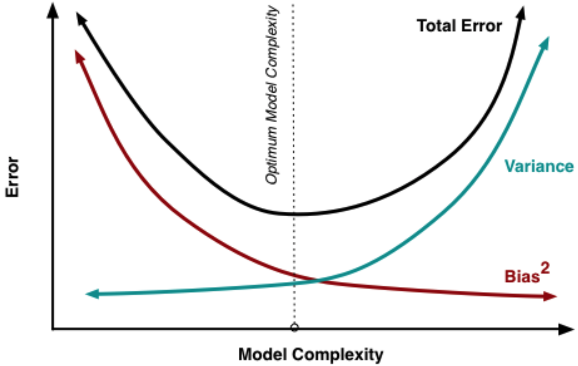

Figure: Bias and variance contributing to total error.[2]

[2]Source: http://scott.fortmann-roe.com/docs/BiasVariance.html

Overparameterization in Deep Learning

Number of total parameters[3]:

- LeNet-5 [LeCun et al., 1998]: 60,000

[3] Source: https://www.jeremyjordan.me/convnet-architectures/

Overparameterization in Deep Learning

Number of total parameters[3]:

- LeNet-5 [LeCun et al., 1998]: 60,000
- AlexNet [Krizhevsky et al., 2012]: 60 million

[3] Source: https://www.jeremyjordan.me/convnet-architectures/

Overparameterization in Deep Learning

Number of total parameters[3]:

- LeNet-5 [LeCun et al., 1998]: 60,000
- AlexNet [Krizhevsky et al., 2012]: 60 million
- VGG-16 [Simonyan and Zisserman, 2014]: 138 million

[3] Source: https://www.jeremyjordan.me/convnet-architectures/

Overparameterization in Deep Learning

Number of total parameters[3]:

- LeNet-5 [LeCun et al., 1998]: 60,000
- AlexNet [Krizhevsky et al., 2012]: 60 million
- VGG-16 [Simonyan and Zisserman, 2014]: 138 million
- Inception (GoogLeNet) [Szegedy et al., 2015] and [Szegedy et al., 2016]: 5 million (V1) and 23 million (V3)

[3] Source: https://www.jeremyjordan.me/convnet-architectures/

Overparameterization in Deep Learning

Number of total parameters[3]:

- LeNet-5 [LeCun et al., 1998]: 60,000
- AlexNet [Krizhevsky et al., 2012]: 60 million
- VGG-16 [Simonyan and Zisserman, 2014]: 138 million
- Inception (GoogLeNet) [Szegedy et al., 2015] and [Szegedy et al., 2016]: 5 million (V1) and 23 million (V3)
- ResNet [He et al., 2016]: 25 million (ResNet 50)

[3] Source: https://www.jeremyjordan.me/convnet-architectures/

Overparameterization in Deep Learning

Number of total parameters[3]:

- LeNet-5 [LeCun et al., 1998]: 60,000
- AlexNet [Krizhevsky et al., 2012]: 60 million
- VGG-16 [Simonyan and Zisserman, 2014]: 138 million
- Inception (GoogLeNet) [Szegedy et al., 2015] and [Szegedy et al., 2016]: 5 million (V1) and 23 million (V3)
- ResNet [He et al., 2016]: 25 million (ResNet 50)
- DenseNet [Huang et al., 2017]: 40 million (DenseNet-190, k=40)

[3] Source: https://www.jeremyjordan.me/convnet-architectures/

Overparameterization in Deep Learning

Number of total parameters[3]:

- LeNet-5 [LeCun et al., 1998]: 60,000
- AlexNet [Krizhevsky et al., 2012]: 60 million
- VGG-16 [Simonyan and Zisserman, 2014]: 138 million
- Inception (GoogLeNet) [Szegedy et al., 2015] and [Szegedy et al., 2016]: 5 million (V1) and 23 million (V3)
- ResNet [He et al., 2016]: 25 million (ResNet 50)
- DenseNet [Huang et al., 2017]: 40 million (DenseNet-190, k=40)
- ...

[3] Source: https://www.jeremyjordan.me/convnet-architectures/

Why Deep Learning Does not Overfit?

- Effective capacity?

Why Deep Learning Does not Overfit?

- Effective capacity?
- Regularization?

Why Deep Learning Does not Overfit?

- Effective capacity?
- Regularization?
- Optimization? ([Gunasekar et al., 2017], [Ma et al., 2017])

Effective Capacity of Neural Networks

Key Takeaways

Deep neural networks easily fit random labels!

Randomization Tests

- Take a candidate architecture (Inception, AlexNet, MLP, . . .).
- Train on the true data and on a copy of the data with random labels.
- Experiments setting:
    - true label
    - partially corrupted labels
    - random labels
    - shuffled pixels
    - random pixels
    - Gaussian

Learning Curves

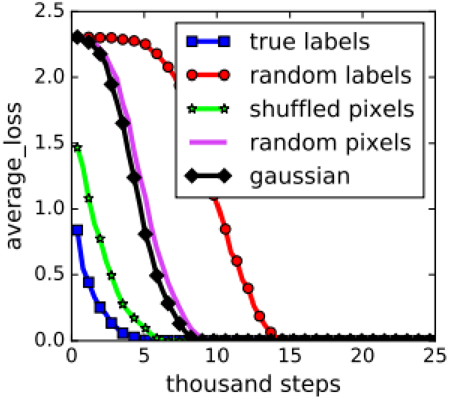

Figure: shows the learning curves of the Inception model on the CIFAR10 dataset under various settings.

Convergence Slowdown

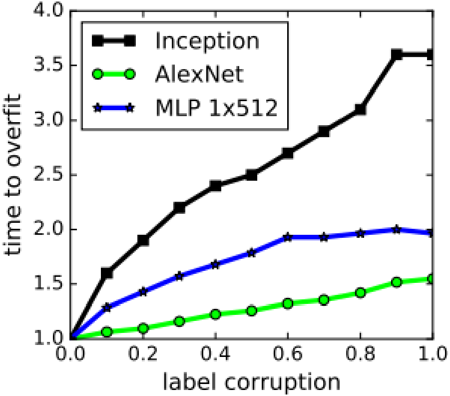

Figure: shows the slowdown of the convergence time with increasing level of label noises.

Generalization Error Growth

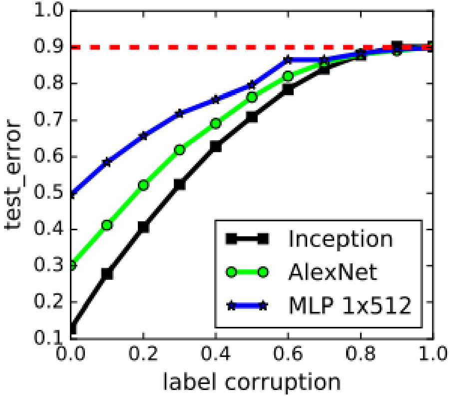

Figure: shows the test error under different label corruptions.

Finite-Sample Expressivity

Theorem

There exists a two-layer neural network with ReLU activations and $2n + d$ weights that can represent any function on a sample of size n in d dimensions.

Remark

The theorem is not stated at the **population level***, which is to show what functions of the entire domain can and cannot be represented by certain classes of neural networks with the same number of parameters.*

Finite-Sample Expressivity

Lemma

For any two interleaving sequences of n real numbers $b_1 < x_1 < b_2 < x_2 \ldots < b_n < x_n$, the $n \times n$ matrix $A = [\max\{x_i - b_j, 0\}]_{ij}$ has full rank. Its smallest eigenvalue is $\min_i\{x_i - b_i\}$.

Finite-Sample Expressivity

Proof.

- $c(x) = \sum_{j=1} \omega_j \max\{< a, z > -b_j, 0\}$ can be expressed by a depth 2 network with ReLU activations. $(\omega, b \in \mathbb{R}^n, a \in \mathbb{R}^d)$

$\square$

Finite-Sample Expressivity

Proof.

- $c(x) = \sum_{j=1} \omega_j \max\{< a, z > -b_j, 0\}$ can be expressed by a depth 2 network with ReLU activations. $(\omega, b \in \mathbb{R}^n, a \in \mathbb{R}^d)$
- Fix sample $S = \{z_1, \ldots, z_n\}$ and target vector $y \in \mathbb{R}^n$.

$\square$

Finite-Sample Expressivity

Proof.

- $c(x) = \sum_{j=1} \omega_j \max\{< a, z > -b_j, 0\}$ can be expressed by a depth 2 network with ReLU activations. $(\omega, b \in \mathbb{R}^n, a \in \mathbb{R}^d)$
- Fix sample $S = \{z_1, \ldots, z_n\}$ and target vector $y \in \mathbb{R}^n$.
- Choose a and b such that with $x_i =< a, z_i >$ we have the interleaving property $b_1 < x_1 < b_2 < \ldots < b_n < x_n$.

$\square$

Finite-Sample Expressivity

Proof.

- $c(x) = \sum_{j=1} \omega_j \max\{< a, z > -b_j, 0\}$ can be expressed by a depth 2 network with ReLU activations. $(\omega, b \in \mathbb{R}^n, a \in \mathbb{R}^d)$
- Fix sample $S = \{z_1, \ldots, z_n\}$ and target vector $y \in \mathbb{R}^n$.
- Choose a and b such that with $x_i =< a, z_i >$ we have the interleaving property $b_1 < x_1 < b_2 < \ldots < b_n < x_n$.
- $y_i = c(z_i), i \in \{1, \ldots, n\}$ is equivalent yo $y = A\omega$, where $A = [\max\{x_i - b_j, 0\}]_{ij}$.

$\square$

Finite-Sample Expressivity

Proof.

- $c(x) = \sum_{j=1} \omega_j \max\{< a, z > -b_j, 0\}$ can be expressed by a depth 2 network with ReLU activations. $(\omega, b \in \mathbb{R}^n, a \in \mathbb{R}^d)$
- Fix sample $S = \{z_1, \ldots, z_n\}$ and target vector $y \in \mathbb{R}^n$.
- Choose a and b such that with $x_i =< a, z_i >$ we have the interleaving property $b_1 < x_1 < b_2 < \ldots < b_n < x_n$.
- $y_i = c(z_i), i \in \{1, \ldots, n\}$ is equivalent yo $y = A\omega$, where $A = [\max\{x_i - b_j, 0\}]_{ij}$.
- The linear system $y = A\omega$ is solvable by Lemma.

$\square$

Key Takeaways

Deep neural networks easily fit random labels!

The Role of Regularization

What is a regularizer?

A regularizer is anything that hurts the training process.

A list of regularizers

- Data augmentation: augment the training set via domain-specific transformations.
- Weight decay: equivalent to ℓ_2 regularizer on the weights.
- Dropout: mask out each element of a layer output randomly with a given dropout probability.
- Early stopping: stop training at the "right" time.
- Batch normalization: an operator that normalizes the layer responses within each mini-batch.

Empirical Results

Table 1: The training and test accuracy (in percentage) of various models on the CIFAR10 dataset. Performance with and without data augmentation and weight decay are compared. The results of fitting random labels are also included.

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75
(fitting random labels)		no	no	100.0	9.78
Inception w/o BatchNorm	1,649,402	no	yes	100.0	83.00
		no	no	100.0	82.00
(fitting random labels)		no	no	100.0	10.12
Alexnet	1,387,786	yes	yes	99.90	81.22
		yes	no	99.82	79.66
		no	yes	100.0	77.36
		no	no	100.0	76.07
(fitting random labels)		no	no	99.82	9.86
MLP 3x512	1,735,178	no	yes	100.0	53.35
		no	no	100.0	52.39
(fitting random labels)		no	no	100.0	10.48
MLP 1x512	1,209,866	no	yes	99.80	50.39
		no	no	100.0	50.51
(fitting random labels)		no	no	99.34	10.61

Empirical Results

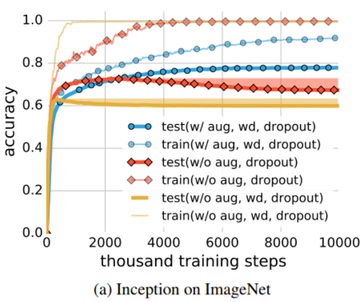

Figure: Effects of implicit regularizers on generalization performance.

An Appeal to Linear Models

- Consider the empirical risk minimization problem: $\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f(\omega^T x_i, y_i)$, where $d \geq n$.

An Appeal to Linear Models

- Consider the empirical risk minimization problem: $\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f(\omega^T x_i, y_i)$, where $d \geq n$.
- The quality of the minima is the curvature of the loss function at the solution. In the linear case, the curvature of all optimal solutions is the same.

An Appeal to Linear Models

- Consider the empirical risk minimization problem: $\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f(\omega^T x_i, y_i)$, where $d \geq n$.
- The quality of the minima is the curvature of the loss function at the solution. In the linear case, the curvature of all optimal solutions is the same.
- SGD update: $\omega_{t+1} = \omega_t - \eta_t e_t x_{i_t}$.

An Appeal to Linear Models

- Consider the empirical risk minimization problem: $\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f(\omega^T x_i, y_i)$, where $d \geq n$.
- The quality of the minima is the curvature of the loss function at the solution. In the linear case, the curvature of all optimal solutions is the same.
- SGD update: $\omega_{t+1} = \omega_t - \eta_t e_t x_{i_t}$.
- If $\omega_0 = 0$, then the solution is of the form $\omega = X^T \alpha$.

An Appeal to Linear Models

- Consider the empirical risk minimization problem: $\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f(\omega^T x_i, y_i)$, where $d \geq n$.
- The quality of the minima is the curvature of the loss function at the solution. In the linear case, the curvature of all optimal solutions is the same.
- SGD update: $\omega_{t+1} = \omega_t - \eta_t e_t x_{i_t}$.
- If $\omega_0 = 0$, then the solution is of the form $\omega = X^T \alpha$.
- Since $X\omega = y$, we have $XX^T \alpha = y$, which has a unique solution.

An Appeal to Linear Models

- Consider the empirical risk minimization problem: $\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f(\omega^T x_i, y_i)$, where $d \geq n$.
- The quality of the minima is the curvature of the loss function at the solution. In the linear case, the curvature of all optimal solutions is the same.
- SGD update: $\omega_{t+1} = \omega_t - \eta_t e_t x_{i_t}$.
- If $\omega_0 = 0$, then the solution is of the form $\omega = X^T \alpha$.
- Since $X\omega = y$, we have $XX^T \alpha = y$, which has a unique solution.
- The kernel solution is equivalent to the minimum ℓ_2-norm solution of $X\omega = y$.

An Appeal to Linear Models

- Consider the empirical risk minimization problem: $\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f(\omega^T x_i, y_i)$, where $d \geq n$.
- The quality of the minima is the curvature of the loss function at the solution. In the linear case, the curvature of all optimal solutions is the same.
- SGD update: $\omega_{t+1} = \omega_t - \eta_t e_t x_{i_t}$.
- If $\omega_0 = 0$, then the solution is of the form $\omega = X^T \alpha$.
- Since $X\omega = y$, we have $XX^T\alpha = y$, which has a unique solution.
- The kernel solution is equivalent to the minimum ℓ_2-norm solution of $X\omega = y$.
- $\omega = X^T(XX^T)^{-1}y$, where $X^T(XX^T)^{-1}$ is often called pseudoinverse.

Key Takeaways

- Regularizers could help to improve the generalization performance when properly tuned.

Key Takeaways

- Regularizers could help to improve the generalization performance when properly tuned.
- It is unlikely that the regularizers are the fundamental reason for generalization, as the networks continue to perform well after all the regularizers removed.

Key Takeaways

- Regularizers could help to improve the generalization performance when properly tuned.
- It is unlikely that the regularizers are the fundamental reason for generalization, as the networks continue to perform well after all the regularizers removed.
- SGD will converge to the solution with minimum norm in the linear case, acting like a form of regularization. However, the notion of minimum norm is not predictive of generalization performance.

Conclusion

Conclusion

- This paper is focused on the question why large neural networks generalize well in practice.
- The effective capacity of several successful neural network architectures is large enough to shatter the training data.
- Regularizers are not the fundamental reason for generalization.
- Optimization may ast as an implicit form of regularization.
- A precise formal measure under which these enormous models are simple has not been discovered yet.

Discussion

Are there any experiments you think could be conducted to further investigate this phenomenon?

Discussion

Do you have other explanations behind the generalization behavior?

Thank you.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. (2017).
Implicit regularization in matrix factorization.
In *Advances in Neural Information Processing Systems*, pages 6151–6159.

He, K., Zhang, X., Ren, S., and Sun, J. (2016).
Deep residual learning for image recognition.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017).
Densely connected convolutional networks.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012).
Imagenet classification with deep convolutional neural networks.

In *Advances in neural information processing systems*, pages 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998).
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324.

Ma, C., Wang, K., Chi, Y., and Chen, Y. (2017).
Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion and blind deconvolution.
arXiv preprint arXiv:1711.10467.

Simonyan, K. and Zisserman, A. (2014).
Very deep convolutional networks for large-scale image recognition.
arXiv preprint arXiv:1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015).
Going deeper with convolutions.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016).
Rethinking the inception architecture for computer vision.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016).
Understanding deep learning requires rethinking generalization.
arXiv preprint arXiv:1611.03530.