Understanding Deep Learning Requires Rethinking Generalization

ChiyuanZhang[1] Samy Bengio[3] Moritz Hardt[3] Benjamin Recht[2]
Oriol Vinyals[4]

[1]Massachusetts Institute of Technology

[2]University of California, Berkeley

[3]Google Brain

[4]Google DeepMind

ICLR, 2017

Presenter: Arshdeep Sekhon

Generalization Error

1. *Generalization error = test error − training error*
2. A network that generalizes well has comparable performance on the test and training set
3. $p >> n$ in neural networks, still low generalization error
4. *Question: What makes a NN with good generalization different from one that generalizes poorly?*

Traditional View of generalization

1. Model Family
2. Complexity Measures:
   1. Rademacher Complexity
   2. Uniform Stability
   3. VC dimension
3. Regularization
   1. Explicit Regularization: weight decay, dropout,etc
   2. Implicit Regularization: early stopping, batch norm,etc

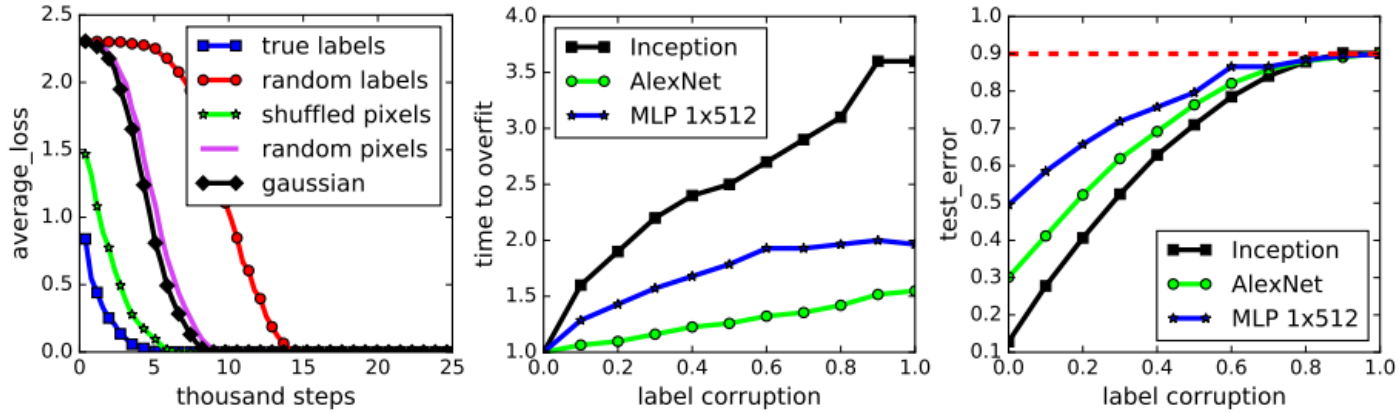
Effective Capacity of Neural Networks

Experiments with the following modifications of input and labeled data:

1. original data
2. partially corrupted labels: independently with probability p, the label of each image is corrupted as a uniform random class
3. Randomize labels completely: No relationship between data and labels
4. shuffled pixels: same random permutation of pixels to all images
5. Random Pixels: different random permutation of pixels to all images
6. Gaussian: Use gaussian to generate random pixels

Ideally, should affect training procedure as there is no relationship between input and output.

Results



(a) learning curves (b) convergence slowdown (c) generalization error growth

Figure: Randomization tests results

1. Training Error zero: fits the data perfectly/Overfitting
2. No changes in training procedure
3. more corruption slows convergence

Implications

1. Rademacher Complexity:

$$E_\sigma \left[\sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right] \tag{1}$$

where $\sigma_1, \sigma_1, \sigma_1, \in +1, -1$ are iid random variables
Indicates how well a model in the hypothesis class fits a random assignment.

Implications

1. Rademacher Complexity:

$$E_\sigma \left[\sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right] \tag{1}$$

where $\sigma_1, \sigma_1, \sigma_1, \in +1, -1$ are iid random variables
Indicates how well a model in the hypothesis class fits a random assignment.

2. Because the NNs fit the training data perfectly, $R(H) \approx 1$. But, this is the upper bound for Rademacher complexity.*generalization is between zero and the worst case.*

Implications

1. Rademacher Complexity:

$$E_\sigma \left[\sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right] \tag{1}$$

where $\sigma_1, \sigma_1, \sigma_1, \in +1, -1$ are iid random variables
Indicates how well a model in the hypothesis class fits a random assignment.

2. Because the NNs fit the training data perfectly, $R(H) \approx 1$. But, this is the upper bound for Rademacher complexity.*generalization is between zero and the worst case.*

3. Uniform Stability: Uniform stability of an algorithm A measures how sensitive the algorithm is to the replacement of a single example. A property of the algorithm/Has no relationship to data/distribution of labels

Regularization and generalization

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

1.

Figure: Regularization and Generalization

2. Key Observations:
   1. Even with regularization, networks generalize fine.
   2. Even with regularization, training error is still zero: fit perfectly.

Implicit Regularization and Generalization

1. Early Stopping
2. Batch Normalization

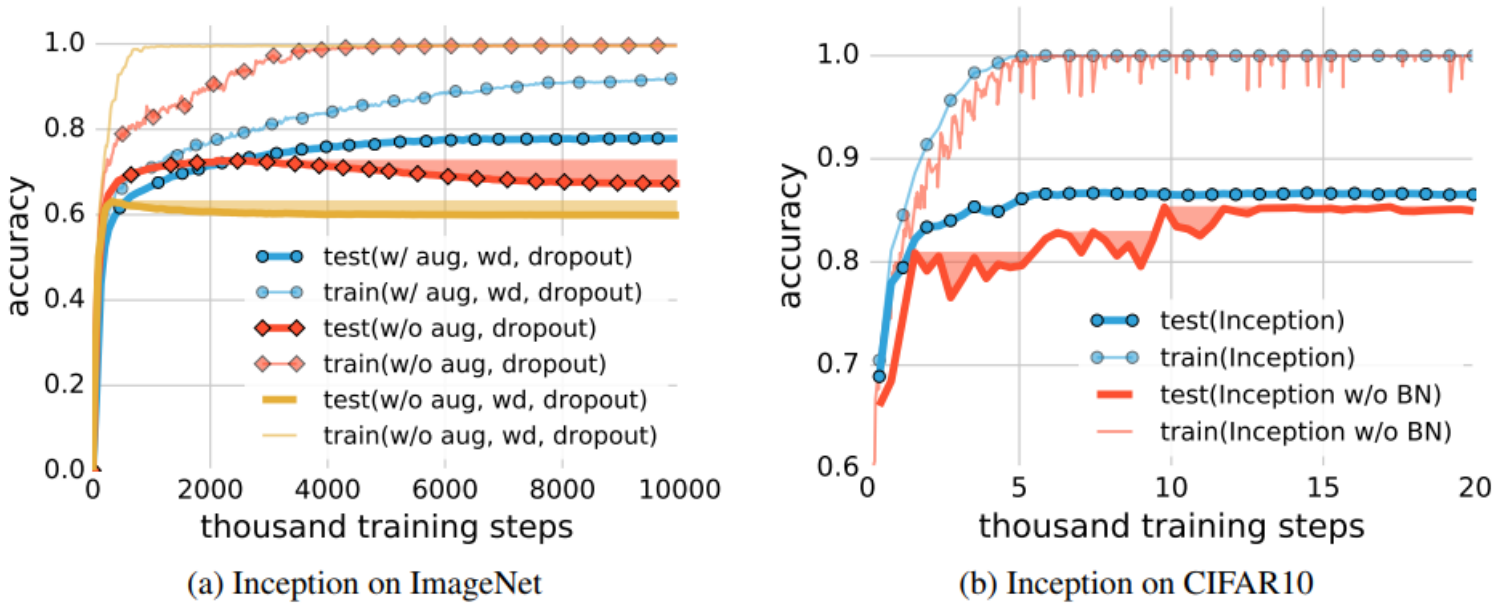


Figure: Implicit Regularization

3. Continue to perform well without regularization

Regularization for Generalization: Key Insights

1. Regularization improves generalization ability.
2. Not the key reason for generalization.

Model Expressivity

1. Old/Previous View: What functions can be expressed by certain classes of neural networks?
2. Finite Sample Expressivity: Given n samples of d dimension, parameters required to express any function?

Theorem: 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.

Proof:

Lemma 1:

For any interleaving sequences of n real numbers, $b_1 < x_1 < b_2 <, \cdots, b_n < x_n$, the $n \times n$ matrix $A = max[x_i - b_j, 0]$ has full rank.

Proof:

$$A = \begin{bmatrix} \max\{x_1 - b_1, 0\} & \max\{x_1 - b_2, 0\} & \cdots & \max\{x_1 - b_n, 0\} \\ \max\{x_2 - b_1, 0\} & \max\{x_2 - b_2, 0\} & \cdots & \max\{x_2 - b_n, 0\} \\ \vdots & \ddots & \ddots & \vdots \\ \max\{x_n - b_1, 0\} & \max\{x_n - b_2, 0\} & \cdots & \max\{x_n - b_n, 0\} \end{bmatrix}$$

$$\overset{(i)}{=} \begin{bmatrix} x_1 - b_1 & 0 & 0 & \cdots & 0 \\ x_2 - b_1 & x_2 - b_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x_{n-1} - b_1 & x_{n-1} - b_2 & \ddots & \cdots & 0 \\ x_n - b_1 & x_n - b_2 & x_n - b_3 & \cdots & x_n - b_n \end{bmatrix}$$

Theorem: Finite Sample Expressivity

consider function:

$$c(x) = \sum_{j=1}^{n} w_j \left[max < a, x > -b_j, 0 \right] \tag{2}$$

This can be expressed as a 2 layer ReLU network

$S = z_1, \cdots, z_n$

$x_i = < a, z_i >$

Choose a,b such that the interleaving property
$b_1 < x_1 < b_2 <, \cdots, b_n < x_n$, is satisfied

Reduces to $y = Aw$

because A is invertible by the lemma,

Find suitable weights w

Key contributions

1. Traditional Views fail to explain generalization
2. Regularization methods are not sufficient or necessary for explaining generalization
3. Optimization is easy even if the resulting model does not generalize well