Generalization and Memorization in Neural Networks

SMAI Project - Team 43

Table of contents

1. Inspired From
2. Generalization Vs Memorization
3. Metrics for Quantitative Estimation
4. Role of Regularizers
5. Generalization using IB Principle

Inspired From

Papers

- Understanding Deep Learning Requires Rethinking Generalization (ICLR 2017 best paper)

  Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, Oriol Vinyals

- A Closer Look at Memorization in Deep Networks (ICML 2015)

  University de Montreal

- Deep Learning and the Information Bottleneck Principle (2015)

  Naftali Tishby, Noga Zaslavsky

Primary Questions

In a Neural Network (NN)

$$\#params > \#samples$$

- What is the capacity of a NN to memorize ?
- Are the Neural Networks learning the data or Memorizing ?
- How helpful are the regularization techniques ?
- What can be a Quantitative Estimation of memorization in NN ?

Investigate

Dataset 1

True Labels

The original dataset without modification

- Generalization Error = low
- Train Error = low
- Test Error = low
- Train time = less

Dataset 2

Random Labels

Labels replaced with random labels.

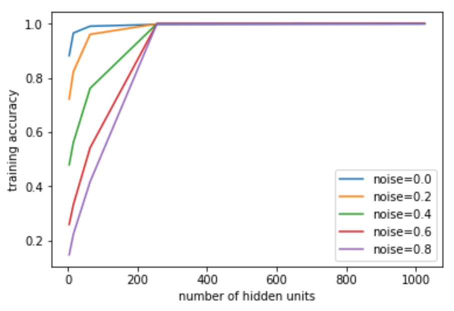
- Generalization Error = high
- Train Error = low
- Test Error = high
- Train time = high

Generalization error = Train Error - Test Error

Generalization Vs Memorization

First Claim

NNs Easily fit Random Labels given enough complexity



MNIST Trained on 2-layered MLP with varying number of hidden units.

Finite Sample Expressivity

Theorem

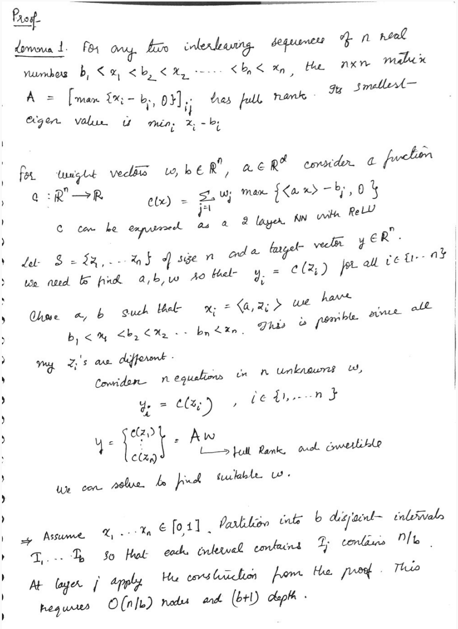
There exists a two-layer neural network with ReLU activations and 2n + d weights that can represent any function on a sample of size n in d dimensions

This results in a very wide neural network but it can be shown that,

Corollary

For every $k \geq 2$, there exists neural network with ReLU activations of depth k, width O(n/k) and O(n+d) weights that can represent any function on a sample of size n in d dimensions.

Proof



Proof

Lemma 1. For any two interleaving sequences of n real numbers $b_1 < x_1 < b_2 < x_2 \cdots\cdots < b_n < x_n$, the $n \times n$ matrix $A = [\max\{x_i - b_j, 0\}]_{ij}$ has full rank. Its smallest eigen value is $\min_i x_i - b_i$

for weight vectors $w, b \in \mathbb{R}^n$, $a \in \mathbb{R}^d$ consider a function $c : \mathbb{R}^n \longrightarrow \mathbb{R}$

$$c(x) = \sum_{j=1} w_j \max\{\langle a\, x\rangle - b_j, 0\}$$

c can be expressed as a 2 layer NN with ReLU

Let $S = \{z_1, \ldots z_n\}$ of size n and a target vector $y \in \mathbb{R}^n$. we need to find a, b, w so that $y_i = c(z_i)$ for all $i \in \{1 .. n\}$

Chose a, b such that $x_i = \langle a, z_i \rangle$ we have $b_1 < x_1 < b_2 < x_2 \,..\, b_n < x_n$. This is possible since all my z_i's are different.

Consider n equations in n unknowns w,

$$y_i = c(z_i) \quad , \quad i \in \{1, \ldots n\}$$

$$y = \begin{Bmatrix} c(z_1) \\ \vdots \\ c(z_n) \end{Bmatrix} = A\underset{\hookrightarrow \text{Full Rank and invertible}}{w}$$

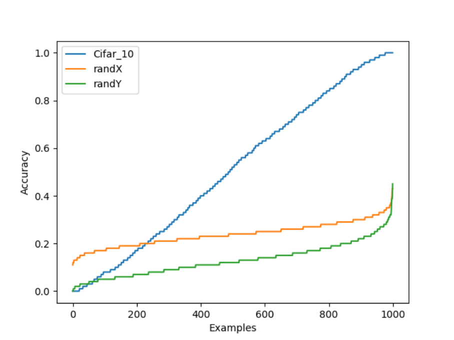
We can solve to find suitable w.

⇒ Assume $x_1 \ldots x_n \in [0, 1]$. Partition into b disjoint intervals $I_1 \ldots I_b$ so that each interval contains I_j contains n/b.

At layer j apply the construction from the proof. This requires $O(n/b)$ nodes and $(b+1)$ depth.

Second Claim

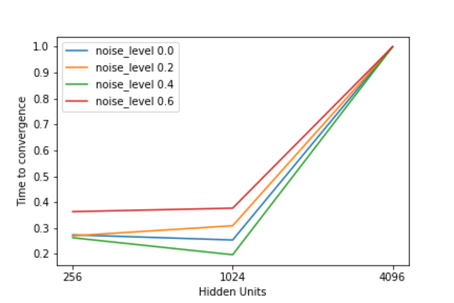
NNs learn simple patters first

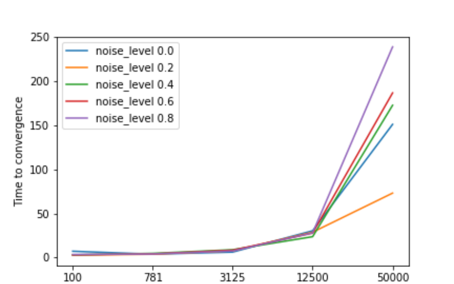


In Real data examples are consistently classified (in)correctly after a single epoch.

For noise data, the difference between examples is much less.

Convergence with Noise



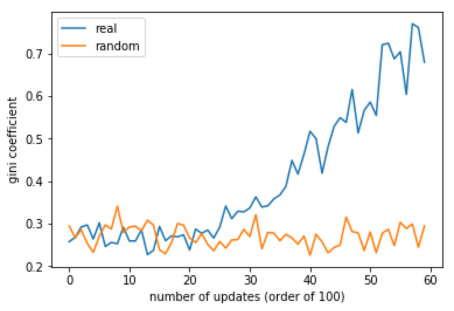


Metrics for Quantitative Estimation

Loss Sensitivity

Effect of sample on average loss

$$g_x^t = \| \partial L_t / \partial x \|_1$$



Real data: High value on a subset Random data: High for all values

Critical Sample Ratio (CSR)

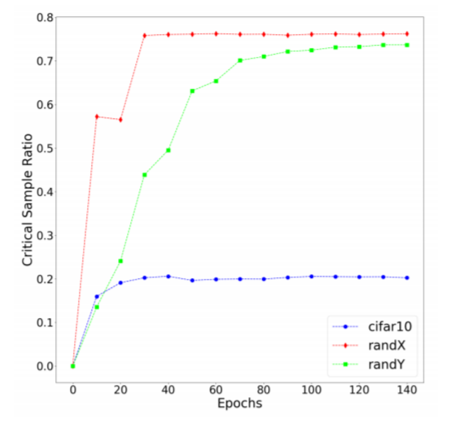
Critical Sample is a subset of data with an adversarial example in its proximity :

$$\underset{i}{argmax}\, f_i(x) \neq \underset{j}{argmax}\, f_j(\hat{x})$$

$$s.t. \parallel x - \hat{x} \parallel_{\infty} \leq r$$

$$CSR = \frac{\#criticalsamples}{\#datapoints}$$

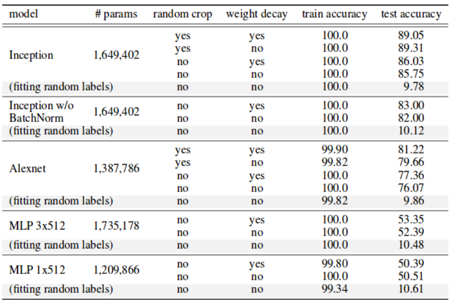
High CSR $\Rightarrow$ complex hypothesis



The higher number of CSRs on the noise data suggest a more complex learned decision surface

Role of Regularizers

Explicit Regularizers



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

Regularizers improve generalization

Regularizers are neither necessary nor sufficient.

Implicit Regularizers (SGD)

Consider linear model, n distinct data points (x_i, y_i) where x_i are d-dimensional. $d \geq n$

$$min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} loss(w^T x_i, y_i)$$

$Xw = y$ has infinite solutions

So is it possible to converge to one unique global optima ? SGD gives $w = X^T \alpha$, that is the w lies in the span of the data points. This along with $Xw = y$ gives us,

$$XX^T \alpha = y$$

Here α has unique solution. So therefore, any set of labels can be fit by forming the Gram matrix $K = XX^T$ and solving $K\alpha = y$.

Huge size of Kernel Matrix!

SGD will often converge to the solution with minimum norm.

$$\underset{w}{min} \parallel w \parallel_2^2$$

s.t

$$y = Xw$$

Using Lagrangian,

$$\underset{w}{min}(\parallel w \parallel_2^2 + \lambda \parallel y - Xw \parallel_2^2)$$

First order condition,

$$w(I + \lambda X^T X) = \lambda X^T y$$

As $\lambda \to \infty$

$$w* = (X^T X)^{-1} X^T y$$

Using the kernel solution $y = XX^T \alpha$ we get,

$$w* = (X^T X)^{-1} X^T XX^T \alpha = X^T \alpha = w(SGD)$$

Generalization using IB Principle

In Neural Networks

Tishby *et. al.* venture into quantifying the DNN using mutual information between the layers and the input and the output variables.

”The remarkable success of DNNs in learning to extract relevant features is mainly attributed to the sequential processing of the data, namely that each hidden layer operates as the input to the next one, which allows the construction of higher level distributed representations.”

$$I(X, \hat{X}) = H(X) - H(X|\hat{X}) = \sum p(x, \hat{x}) \log \frac{p(x, \hat{x})}{p(x)p(\hat{x})}$$

where H is the entropy

Sufficient Statistics

A **statistic** is any function $T(X)$. If θ parametrizes the distribution of X. Then for any statistic we have the Markov chain

$$\theta \leftarrow X \leftarrow T(X)$$

Data processing inequality tells us that $I(\theta, T(X)) \leq I(\theta, X)$
A statistic $T(X)$ is **sufficient** for a parameter θ if

$$\theta \leftarrow T(X) \leftarrow X$$

that is $I(\theta, T(X)) \geq I(\theta, X)$

Hence

$$I(\theta, T(X)) = I(\theta, X)$$

Data Processing Inequality

Given $X \rightarrow Y \rightarrow Z$ $\quad$ *To prove* $I(X;Y) \geq I(X;Z)$

$$p(x,y,z) = p(x)p(y|x)p(z|x,y)$$

From the markov chain, Z is independent of X.

$$Hence, p(x,y,z) = p(x)p(y|x)p(z|y)$$

$$p(x,z|y) = \frac{p(x,y,z)}{p(y)} = \frac{p(x)p(y|x)p(z|y)}{p(y)}$$
$$= \frac{p(x,y)}{p(y)}p(z|y) = p(x|y)p(z|y)$$

By the definition of Mutual Information:

$$I(X;Y,Z) = I(X;Z) + I(X;Y|Z) = I(X;Y) + I(X;Z|Y)$$

As X and Z are conditionally independent on Y

$$I(X;Z|Y) = 0 \text{ and } I(X;Y|Z) \geq 0$$

$$Hence\ I(X;Y) \geq I(X;Z)$$

Inforamtion Bottleneck Principle

$$\hat{X} = \underset{T}{argmin}\ I(X, T(X))\ \ s.t.\ \ I(Y, T(X)) \geq \gamma$$

The Lagrangian is given by,

$$\underset{T}{min}\ I(X, T) - \beta I(Y, T)\ \ \beta \geq 0$$

As $\beta \rightarrow \infty$, T preserves all information in X related to Y

T serves as an information bottleneck between X and Y. Extracting information from X that is relevant to Y

Generalization using IB Principle

$I(X, Y)$

Risk term, measuring the performance of a hypothesis on the sample data

$I(X, \hat{X})$

Regularization term, which penalizes complex hypotheses and so ensures reasonable generalization to unseen data.

Thus minimizing the Lagrangian ensures

- higher generalization as $\hat{X}$ is a maximally compressed representation of X
- higher performance as $\hat{X}$ stores the most relevant information about Y