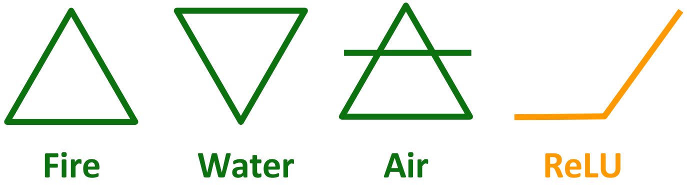


Understanding *Generalization in Deep Learning*

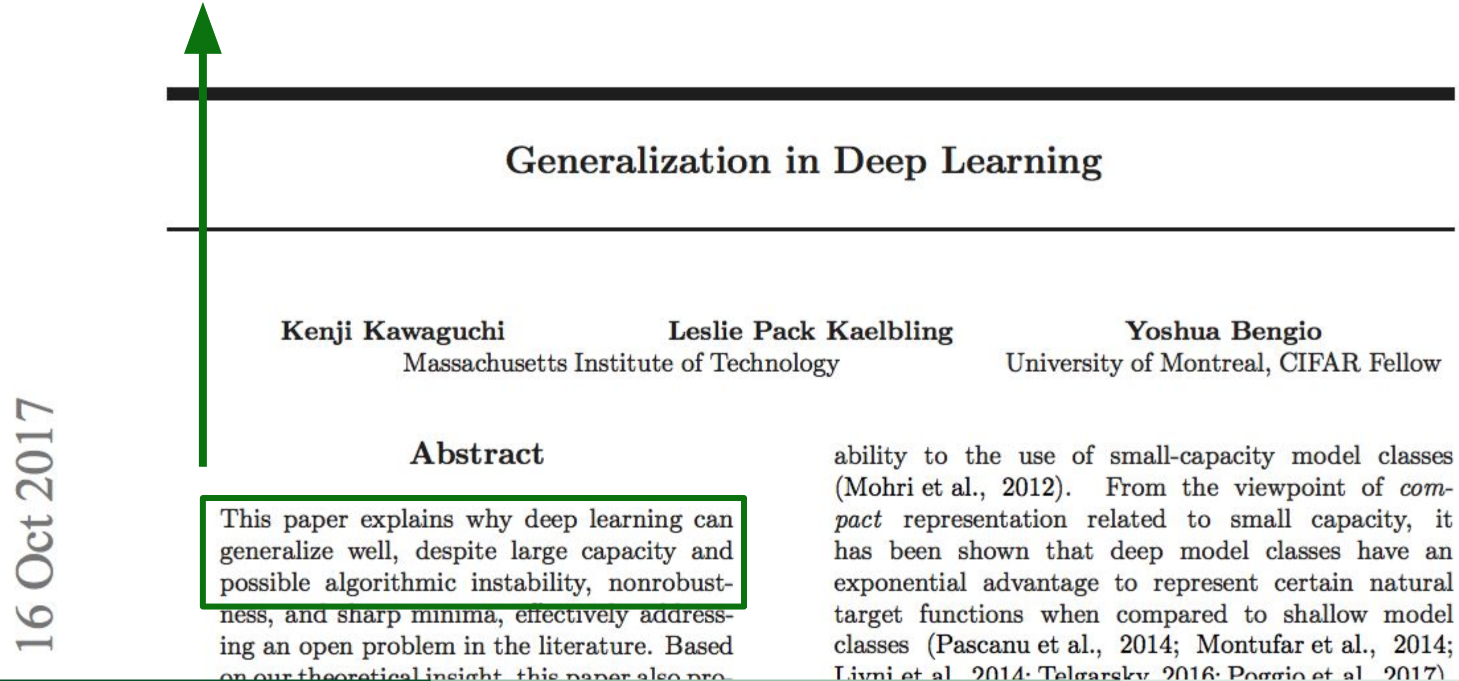
Ou Changkun

WiSe 2017/2018
University of Munich, Seminar *Deep Learning*

hi@changkun.us

February 1, 2018

“This paper explains why deep learning can generalize well”



16 Oct 2017

Generalization in Deep Learning

Kenji Kawaguchi **Leslie Pack Kaelbling**
Massachusetts Institute of Technology

Yoshua Bengio
University of Montreal, CIFAR Fellow

Abstract

This paper explains why deep learning can generalize well, despite large capacity and possible algorithmic instability, nonrobustness, and sharp minima, effectively addressing an open problem in the literature. Based on our theoretical insight, this paper also pro-

ability to the use of small-capacity model classes (Mohri et al., 2012). From the viewpoint of *compact* representation related to small capacity, it has been shown that deep model classes have an exponential advantage to represent certain natural target functions when compared to shallow model classes (Pascanu et al., 2014; Montufar et al., 2014; Livni et al., 2014; Telgarsky, 2016; Poggio et al., 2017).

Outline

1 Motivation

- Backgrounds Introduction
- Recap of Traditional Generalization Theory

2 Rethinking of Generalization

3 Bounds of Generalization Gap

4 DARC Regularizer

What is Generalization?

Definition: Generalization Bound

Given an sample dataset $S_m = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_m, y_m)\}$generated by i.i.d from an unknown distribution $\mathcal{D}$, let ℓ be a loss function. Suppose we have a hypothesis *h* learned from model by optimization algorithm *A* and the dataset, the expected risk $R[h_{\mathcal{A}(S_m)}] = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(h(\boldsymbol{x}), y)]$ and empirical risk $\hat{R}_m[h_{\mathcal{A}(S_m)}] = \frac{1}{m}\sum_{I=1}^{m}\ell(h(\boldsymbol{x_i}), y_i)$. The generalization bound defined as follows:

$$G_{h,\ell,S_m} = R[h] - \hat{R}_m[h]$$

Informal:

- **hypothesis: a function you learned**
- **hypothesis set (class) = all possible functions you can learn**
- **generalization: “knowledge”**
- **generalization bound = test error - training error**

Goals of Generalization Theory

"There is nothing more practical than a good theory"

- Role 1: Expected risk guarantee (**estimating test error**)
- Role 2: Generalization bound (**generalization quality, determine good models**)
- Role 3: Practical guidance (**archive optimal rapidly**)

Informal: **Generalization theory is all about:** Test error - Training error = ????

The Force of Generalization

- Traditional wisdom (theories)
  - Occam’s Razor & No Free Lunch Theorem;
  - Algorithmic stability & robustness;
  - Flat region generalize better (unproved).
- Paper [Zhang et al. 2017] (empirical)
  - Deep models generalize via memorization
  - Explicit regularization (weights norm, L2) is unnecessary
  - Optimization algorithms (SGD) can implicit regularize
- **“Paradox”: why deep learning sometimes generalize well** despite its large capacity, instability, non-robustness, and sharp minima**?**

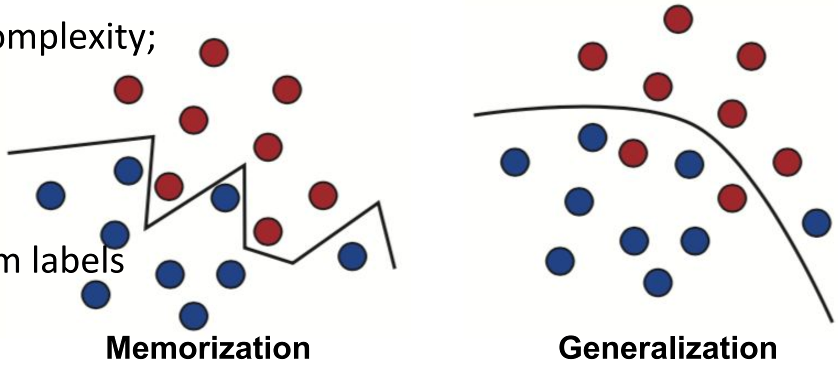
This paper:

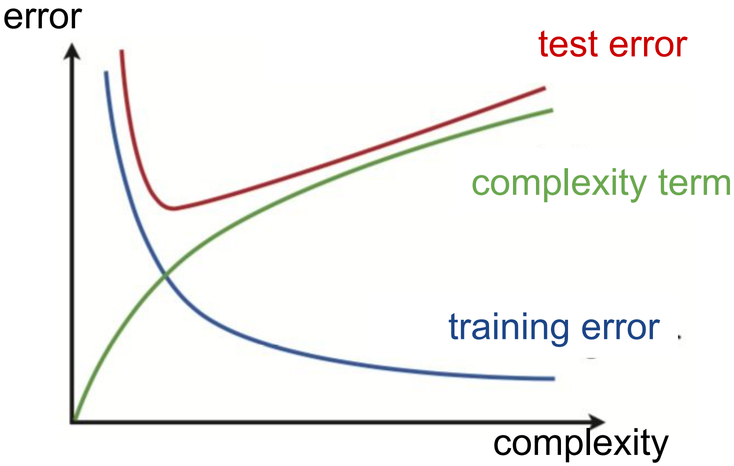
- Decomposition of generalization theory
- Understanding generalization failures

Recap I: Model Complexity

Images souce [Mohri et al. 2012]

- Traditional theory uses **VC dimension** measures model complexity;
- However, VC dimension is data independent;
- And **Rademacher complexity is** data dependent;
- VC(deep nets) = O(#weights*log(#weights));
- Rademacher complexity measures ability of fitting random labels





Rademacher bound: $$R[h] \leq \hat{R}_m[h] + 2\Re(\mathcal{H}) + \sqrt{\frac{\ln \delta^{-1}}{2m}}$$

VC bound: $$R[h] \leq \hat{R}_m[h] + \sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}} + \sqrt{\frac{\ln \delta^{-1}}{2m}}$$

Informal: Test error ≤ Traning error + Complexity Penalty

Recap II: Algorithmic Stability & Robustness

- Algorithmic **stability** consider how **one data point perturbation** influence hypothesis function;
  - If an optimization algorithm is sensitive to a single datapoint, then the algorithm is unstable.
- Algorithmic **robustness** consider how sensitive apply same algorithm to **partitions of a dataset**.
  - If an algorithm is sensitive to different partitions, then the algorithm is non-robustness

Informal: Test error ≤ Traning error + Algorithm Penalty

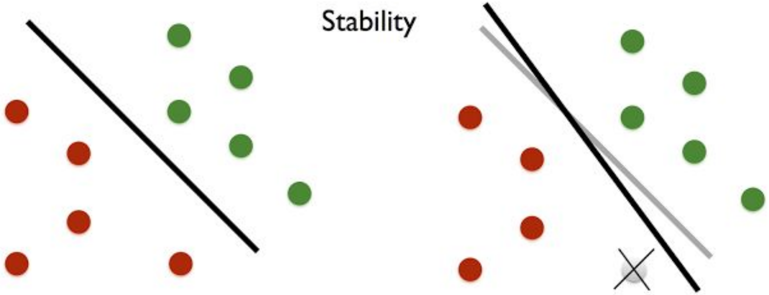
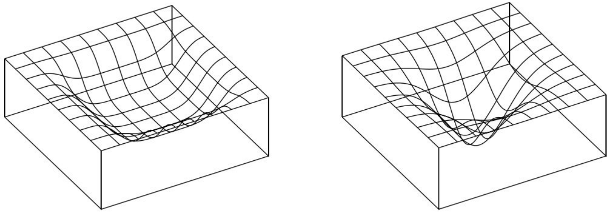


Image Source [Google Inc., 2015]

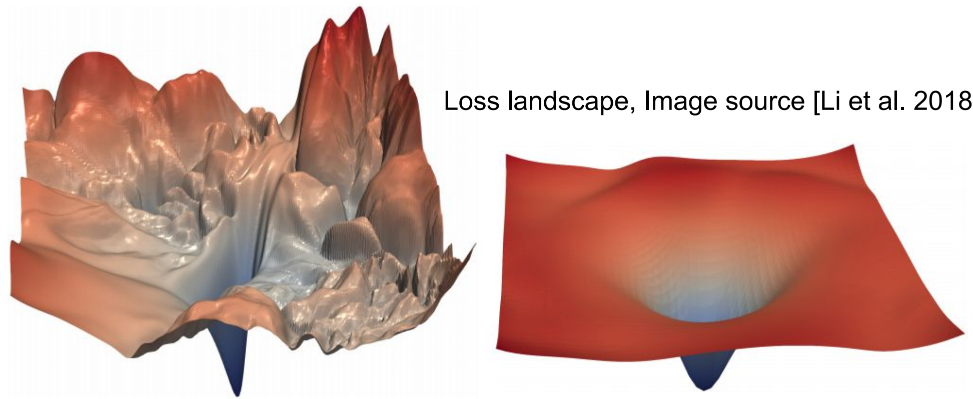
Recap III: Flat Region Conjecture

- (**empirical observation says**) Objective function optimization in high dimentional space:
  - #Saddle points tend to exponetially large than #minima;
  - Conjecture: Flat region tend to generalize well.

Image source [Dauphin et al. 2015]



“Flat region” “Sharp region”



Loss landscape, Image source [Li et al. 2018]

(a) without skip connections (b) with skip connections

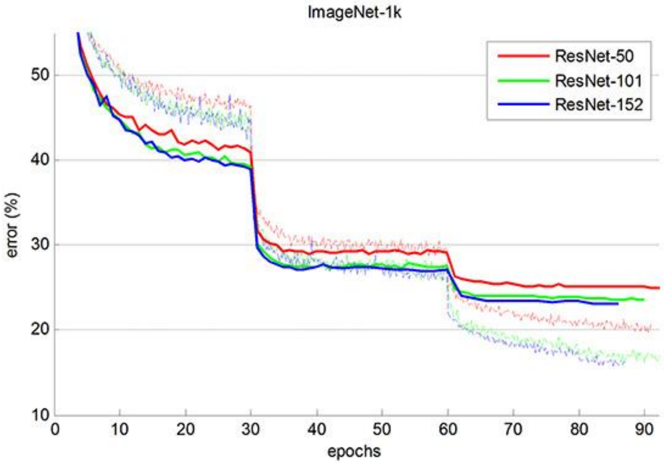


Image source [He et al. 2015]

Outline

1 Motivation

2 Rethinking of Generalization

- Understanding generalization failure
- Emperical risk guarantee

3 Bounds of Generalization Gap

4 DARC Regularizer

On the origin of “paradox”

In Model Compexity theory (same in other theories):

- *p*: model complexity is **appropriate** small
- *q*: generalization gap is small

- Model Complexity theory says: *p => q*
  - ○ *Namely, generalization gap becomes small* ***if*** *model complexity is* **appropriate** *small.*
- [Zhang et al 2017] says: We observed *q*, then we should have *p*. (*q => p*)
  - ○ *Namely, generalization gap becomes small* ***then*** *model complexity is* **appropriate** *small.*

In short: [Zhang et al. 2017] concluded nonsense. ~~Deep models generalize via memorization~~

Traditional theories doesn’t applied to Deep Learning *directly*.

Empirical Risk Guarantee (Estimating Test Error)

- The paper first proved the following bound, regarding deep learning practice:

Theorem: empirical risk estimation (for finite hypothesis class)

$$R[f] \leq \hat{R}_{val}[f] + \frac{2C\ln(\frac{|F_{val}|}{\delta})}{3m_{val}} + \sqrt{\frac{2\gamma^2\ln(\frac{|F_{val}|}{\delta})}{m_{val}}}.$$

where

- C and gamma depends on the quality of model, both equal to 1 in the wosest case;
- m_val is the number of validation set samples;
- |F_val| is the number of hypothesis when using validation set

In short: keep seeking validation model class guarantee “good” test error;

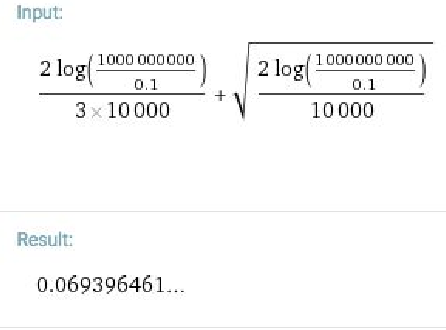
Test error independent with what algorithm you use, and how loss landscape looks like.

Empirical Risk Guarantee II: An Example

Theorem: empirical risk estimation (for finite hypothesis class)

$$R[f] \leq \hat{R}_{val}[f] + \frac{2C\ln(\frac{|F_{val}|}{\delta})}{3m_{val}} + \sqrt{\frac{2\gamma^2\ln(\frac{|F_{val}|}{\delta})}{m_{val}}}.$$

- Let $m_{val} = 10,000$ (MNIST or CIFAR-10) and $\delta = 0.1$ (90% confidence);
- In a worse case, C=1 and $\gamma^2 = 1$, and hypothesis class $|F_{val}| = 1,000,000,000$;
- $R[f] \leq \hat{R}_{val}[f] + 6.94\%$



Input:

$$\frac{2\log(\frac{1000000000}{0.1})}{3\times 10000} + \sqrt{\frac{2\log(\frac{1000000000}{0.1})}{10000}}$$

Result:

0.069396461...

In short: MNIST & CIFAR-10 always get good results because of small hypothesis class on validation set.

Outline

1 Motivation

2 Rethinking of Generalization

3 Bounds of Generalization Gap

- Data-dependent Bound
- Data-independent Bound

4 DARC Regularizer

“Good” Dataset

- The paper proves generalization bound in deep networks **strongly** depends on the **quality of dataset**;
- Thus, it provides a definition to determine what is a “good” dataset for a model:

Definition: Concentrated dataset

$$\lambda_{max}\left(\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T] - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{z}_i\boldsymbol{z}_i^T\right) \leq \beta_1, ||\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{y}_{ik}\boldsymbol{z}_i^T - \mathbb{E}[\boldsymbol{y}_k\boldsymbol{z}^T]||_\infty \leq \beta_2, \mathbb{E}[\boldsymbol{y}^T\boldsymbol{y}] - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{y}_i^T\boldsymbol{y}_i \leq \beta_3$$

where

- **z** = *some_function*(model, **x**), **x** is input;
- **y** is the label of the dataset.

In short, (β1, β2, β3) are constants, all depends on (model, dataset);

Data-dependent Bound (Generalization Quality)

- With a concentrated dataset, we have the following bound:

Theorem: Data-dependent Bound

$$R[f_{\mathcal{A}(S_m)}] - \hat{R}_m[f_{\mathcal{A}(S_m)}] \leq \beta_1 \sum_{k=1}^{d_y} ||\boldsymbol{w}_{\mathcal{A}(S_m),k}||_2^2 + 2\beta_2 \sum_{k=1}^{d_y} ||\boldsymbol{w}_{\mathcal{A}(S_m),k}||_1 + \beta_3$$

where

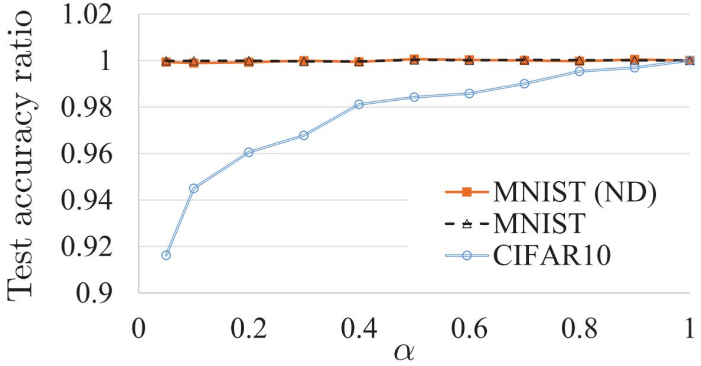
- **w**$_{\mathrm{A(Sm)}}$ is the weights learned by algorithm A over dataset Sm

In short:

- **test error - training error = *constant_of*(model, dataset);**
- **No relationship with algorithm (stability and robustness) and loss (flat or sharp);**
- **With this formula, you can determine dataset quality and estimate your test error.**

Two-phase Training Technique

- By study the properties of SGD, the author propose **two-phase training;**



** ratio = two-phase/base*

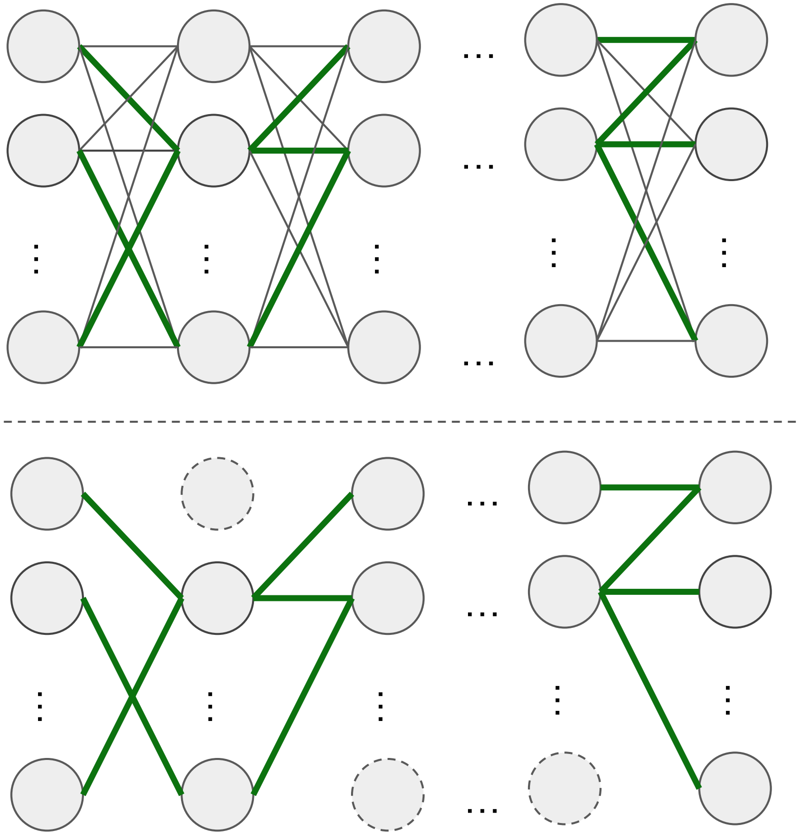
- Experiments: Small alpha archives compatitive performance.

“Standard” Phase

Train the network in standard way with αm samples of the dataset;

“Freeze” Phase

*Freeze **activation pattern** and keep training with the rest of (1-α)m samples.*



Data-independent Bound (Determine Good Model)

- With *two-phase training*, **weights** also **can be bounded**, author proved:

Theorem: Data-independent bound

$$R[f] \leq \hat{R}_{m_\sigma,\rho}[f] + \frac{2d_y^2 C_\sigma C_w}{\rho\sqrt{m_\sigma}} + \sqrt{\frac{\ln \delta^{-1}}{2m_\sigma}}$$

where

- C_rho and C_w are some bounded constants from z and w;
- ρ, m_sigma, delta are hyperparameters;

In short:

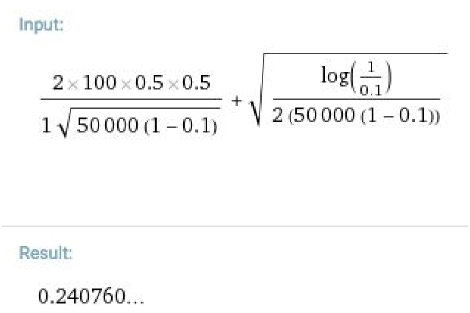
- **test error - training error = *constant_of*(model);**
- **With this formula, you can determine what is a good model architecture.**
- **The bound is independent with #params**

Data-independent Bound: Examples

Theorem: Data-independent bound

$$R[f] \leq \hat{R}_{m_\sigma,\rho}[f] + \frac{2d_y^2 C_\sigma C_w}{\rho\sqrt{m_\sigma}} + \sqrt{\frac{\ln \delta^{-1}}{2m_\sigma}}$$

- Consider CIFAR-10, m=50000, d_y = 10, a worse case: rho = 1, C_sigma = C_w = 0.5.
- With *two-phase training*, alpha = 0.1 (90% samples for freeze phase) and 90% confidence:
  - R[f] ≤ \hat{R}[f] + 24%

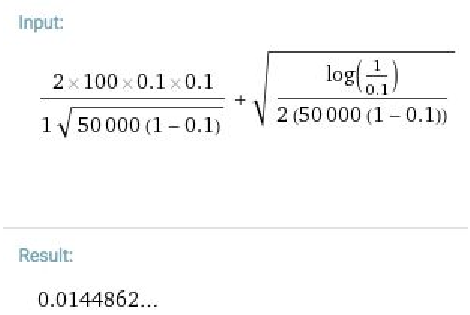


Input:

$$\frac{2\times 100\times 0.5\times 0.5}{1\sqrt{50\,000\,(1-0.1)}}+\sqrt{\frac{\log\left(\frac{1}{0.1}\right)}{2\,(50\,000\,(1-0.1))}}$$

Result:

0.240760...



Input:

$$\frac{2\times 100\times 0.1\times 0.1}{1\sqrt{50\,000\,(1-0.1)}}+\sqrt{\frac{\log\left(\frac{1}{0.1}\right)}{2\,(50\,000\,(1-0.1))}}$$

Result:

0.0144862...

- An optimal case, C_sigma=C_w=0.1
  - R[f] ≤ \hat{R}[f] + **1.4% !!!**

Outline

1 Motivation

2 Rethinking of Generalization

3 Bounds of Generalization

4 DARC Regularizer

- Directly Approximately Regularizing Complexity
- DARC1 Experiments

Directly Approximately Regularizing Complexity (DARC) Family

Data-independent bound derive from Radmacher complexity over ***mini-batch****.* Thus the following formula describe DARC family:

$$\text{loss} = \text{original loss} + \frac{\lambda}{\bar{m}} \hat{\mathbb{E}}_{S_m,\xi}\left[\sup_k \sum_{i=1}^{\bar{m}} \xi_i h_k^{(H+1)}(x_i)\right]$$

The simplest version:

DARC1 Regularization

$$\text{loss} = \text{original loss} + \frac{\lambda}{\bar{m}} \max_k \sum_{i=1}^{\bar{m}} |h_k^{(H+1)}(x_i)|$$

In short, it's a implicit regularizer with explicit expression in loss function.

Experiment results: DARC1 Regularizer on MNIST & CIFAR-10

DARC1 implementation (Keras)

$$\frac{\lambda}{\bar{m}} \max_{k} \sum_{i=1}^{\bar{m}} |h_k^{(H+1)}(x_i)|$$

```
def darc1_loss(model, lamb=0.001):
    def _loss(y_true, y_pred):
        original_loss = K.categorical_crossentropy(y_true, y_pred)
        custom_loss = lamb*K.max(K.sum(K.abs(model.layers[-1].output), axis=0))
        return original_loss + custom_loss
    return _loss
```

Promising results (Base line is the state-of-the-art error):

Test error ratio	MNIST (ND)		MNIST		CIFAR-10	
	mean	stdv	mean	stdv	mean	stdv
DARC1/Base	0.89	0.61	0.95	0.67	0.97	0.79

TL;DR: Only Model Architecture & Dataset Matter

Traditional theories fail to explain generalization for over-parameterized models;

Deep Networks can generalize well:

- because of **large hypothesis capacity (but why SGD family find quickly?)**;
- because of small **validation hypothesis class search**;
- **independent** with #params;
- In a specific task, test error depends on **dataset quality and model architecture**;
- In general, test error only depends on **model architecture.**

DARC regularization family:

- Implicit regularizer to help you improve generalization.

This paper is all about: Test error ≤ Traning error + (Model, Data) Penalty

Thank you

Questions?

Outline

1 Motivation

2 Rethinking of Generalization

3 Bounds of Generalization

4 DARC Regularizer

5 **Related Readings**

Related Readings I: Previous Researches

- [Shalev-Shwartz et al. 2014]
  Understanding machine learning from theory to algorithms
  Cambridge University Press.

- [Neyshabur et al. 2017]
  Exploring generalization in deep learning
  arXiv preprint: 1706.08947, NIPS 2017

- [Xu et al. 2010]
  Robustness and generalization
  arXiv preprint: 1005.2243, JMLR 2012

- [Bousquet et al. 2000]
  Stability and generalization
  JMLR 2000

Related Readings II: Rethinking Generalization

- [Zhang et al. 2017]
  Understanding deep learning requires rethinking generalization
  arXiv preprint: 1611.03530, ICLR 2017

- [Krueger et al. 2017]
  Deep nets don’t learn via memorization
  ICLR 2017 Workshop

- [Kawaguchi et al. 2017]
  Generalization in deep learning
  arXiv preprint: 1710.05468

Related Readings III: The Trick of Deep Path

- [Baldi, et al. 1989]
  Neural networks and principal component analysis: Learning from examples without local minima
  Journal of Neural networks, 1989, Vol.2 52–58

- [Choromanska et al. 2015]
  The Loss Surfaces of Multilayer Networks
  arXiv preprint: 1412.0233, JMLR 2015

- [Kawaguchi 2016]
  Deep learning without poor local minima
  NIPS 2016, oral presentation paper

Related Readings IV: Recent Research

- [Dziugaite et al. 2017]
  Computing Non-vacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data
  arXiv preprint: 1703.11008

- [Bartlett et al. 2017]
  Spectrally-normalized margin bounds for neural networks
  arXiv preprint: 1706.08498, NIPS 2017

- [Neyshabur et al. 2018]
  A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks
  arXiv preprint: 1707.09564, ICLR 2018

- [Neyshabur et al. 2018]
  Certifiable Distributional Robustness with Principled Adversarial Training
  arXiv preprint: 1710.10571, ICLR 2018 Best Paper

Backstage Slides

What’s the difference between .....???

General Concpet Questions

- **What’s the difference between “function” and “hypothesis”?**
  - Function is a general concept represent a mathematical mapping, and a hypothesis is a function you learned by using machine learning model.
- **What’s the difference between “cost”, “loss” and “objective” function?**
  - A loss function is part of cost function, which is an objective function.
- **What’s the difference between “risk”, “loss”, “accuracy” and “error”?**
  - “accuracy” and “error” are different form of risk, which is the expectation of loss.

Why traditional theory also failure?

Properties of (over-parameterized) linear models

Theorem: Generalization failure on linear models

Consider a linear model with the training output matrix $Y = Xw$ where X is a fixed input feature. Let $Y_{test} = X_{test}w$ where X_{test} is a fixed input feature of X_{test}. Let $M = [X^T, X_{test}^T]^T$ and w^* be the ground truth parameters. Then,

- For any target $Y \in \mathbb{R}^{m \times d_y}$, there exist parameters w such that $\hat{Y} = Y$, if $n > m$, $rank(X) \geq m$ and $rank(M) < n$
- For any $\epsilon, \delta \geq 0$, there exist parameters w and pairs of training dataset (Φ, Y) and test dataset $(\Phi_{\text{test}}, Y_{\text{test}})$ such that
  - $\hat{Y} = Y + \epsilon A$ for some matrix A with $||A||_2 \leq 1$
  - $\hat{Y}_{\text{test}} = Y_{\text{test}} + \epsilon B$ for some matrix B with $||B||_2 \leq 1$
  - $||w||_2 \geq \delta$ and $||w - w^*||_2 \geq \delta$

On the origin of “paradox”

Proposition

Given an unknown distribution $(x, y) \sim \mathcal{D}$ and sample dataset S_m, suppose $\exists f^* \in F, \epsilon \geq 0$ such that $R[f^*] - \hat{R}_m[f^*] \leq \epsilon$. Then,

1. For any model class F whose model complexity is large enough to memorize any dataset and which includes f^* possibly at an arbitrarily sharp minimum, there exits $(\mathcal{A}, S_m)$ such that the generalization gap is at most ϵ;

2. For any dataset S_m there exist arbitrarily unstable and arbitrarily non-robust algorithms $\mathcal{A}$ such that the generalization gap of $f_{\mathcal{A}(S_m)}$ is at most ϵ.

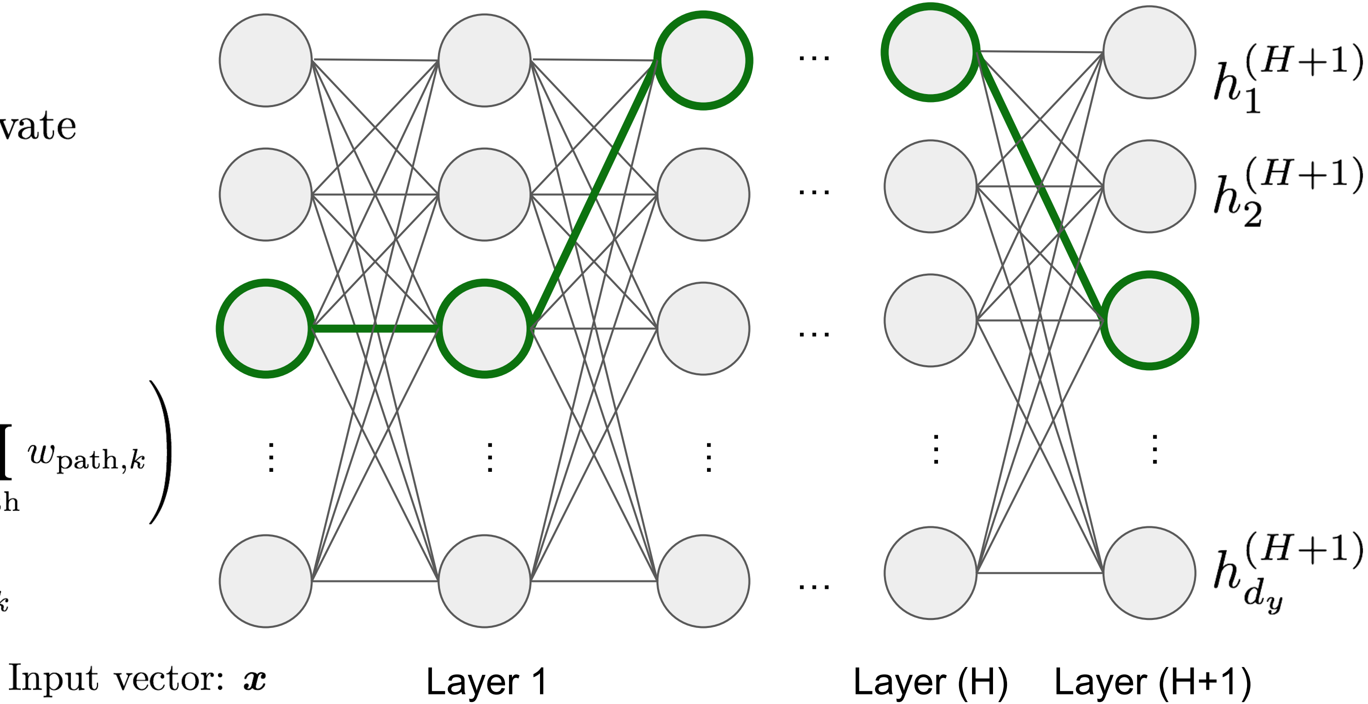
How can I evalute my dataset?

Path Trick in ReLU Net [Choromanska et al. 2015]

$$\boldsymbol{\sigma} = \begin{cases} 1 & \text{if path activate} \\ 0 & \text{otherwise} \end{cases}$$

$$h_k^{(H+1)} = \sum_{\text{path}} \left(x_{\text{path}} \sigma_{\text{path}} \prod_{\text{path}} w_{\text{path},k} \right)$$

$$= [\boldsymbol{x} \circ \boldsymbol{\sigma}]^T \boldsymbol{w}_k = \boldsymbol{z}\boldsymbol{w}_k$$



Concentrated Dataset

$$
\begin{aligned}
R[f_{\mathcal{A}(S_m)}] - \hat{R}_m[f_{\mathcal{A}(S_m)}] &= \mathbb{E}(\ell(h(\boldsymbol{x})^{(H+1)}, y)) - \frac{1}{m}\sum_{i=1}^{m} \ell(h(\boldsymbol{x})^{(H+1)}, y) \\
&= \sum_{k=1}^{d_y}\left[\boldsymbol{w}^T_{\mathcal{A}(S_m),k}\left(\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T] - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{z}_i\boldsymbol{z}_i^T\right)\boldsymbol{w}_{\mathcal{A}(S_m),k}\right] \\
&+ 2\sum_{k=1}^{d_y}\left[\left(\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{y}_{ik}\boldsymbol{z}_i^T - \mathbb{E}[\boldsymbol{y}_k\boldsymbol{z}^T]\right)\boldsymbol{w}_{\mathcal{A}(S_m),k}\right] + \mathbb{E}[\boldsymbol{y}^T\boldsymbol{y}] - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{y}_i^T\boldsymbol{y}_i
\end{aligned}
$$

Definition 3: “good” dataset

$(\beta_1, \beta_2, \beta_3)$-concentrated dataset:

$$
\lambda_{max}\left(\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T] - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{z}_i\boldsymbol{z}_i^T\right) \leq \beta_1, ||\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{y}_{ik}\boldsymbol{z}_i^T - \mathbb{E}[\boldsymbol{y}_k\boldsymbol{z}^T]||_\infty \leq \beta_2, \mathbb{E}[\boldsymbol{y}^T\boldsymbol{y}] - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{y}_i^T\boldsymbol{y}_i \leq \beta_3
$$

Data-dependent Guarantee

$$
\begin{aligned}
R[f_{\mathcal{A}(S_m)}] - \hat{R}_m[f_{\mathcal{A}(S_m)}] &= \mathbb{E}(\ell(h(\boldsymbol{x})^{(H+1)}, y)) - \frac{1}{m}\sum_{i=1}^{m} \ell(h(\boldsymbol{x})^{(H+1)}, y) \\
&= \sum_{k=1}^{d_y} \left[\boldsymbol{w}_{\mathcal{A}(S_m),k}^T \left(\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T] - \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{z}_i^T \right) \boldsymbol{w}_{\mathcal{A}(S_m),k} \right] \\
&+ 2\sum_{k=1}^{d_y} \left[\left(\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}_{ik}\boldsymbol{z}_i^T - \mathbb{E}[\boldsymbol{y}_k \boldsymbol{z}^T] \right) \boldsymbol{w}_{\mathcal{A}(S_m),k} \right] + \mathbb{E}[\boldsymbol{y}^T\boldsymbol{y}] - \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y}_i^T \boldsymbol{y}_i
\end{aligned}
$$

Proposition 4: Data-dependent Bound

$$
R[f_{\mathcal{A}(S_m)}] - \hat{R}_m[f_{\mathcal{A}(S_m)}] \leq \beta_1 \sum_{k=1}^{d_y} ||\boldsymbol{w}_{\mathcal{A}(S_m),k}||_2^2 + 2\beta_2 \sum_{k=1}^{d_y} ||\boldsymbol{w}_{\mathcal{A}(S_m),k}||_1 + \beta_3
$$

How z? Why z?

Why z?

Bernstein Inequality

Independent zero-mean random variables $z_1, z_2, \dots, z_m$, $\mathbb{E}[z_i^2] \leq \gamma^2, |z_i| \leq C$.
Bernstein inequality:

$$\mathbb{P}\left(\frac{1}{m}\sum_{i=1}^{m} z_i > \epsilon\right) \leq \exp - \frac{\epsilon^2 m/2}{\gamma^2 + \epsilon C/3}$$

- Also have matrix form: [Tropp, 2012, Theorem 1.4]
- Basically, all generalization bound infer from concentration inequalities, thingking about how to construct a random variable.

What’s the SGD property supports two-phase training?

SGD, z, w; relations?

Did you forget that sigma depends on w?

No, they could be independent.

$$h_k^{(H+1)} = \sum_{\text{path}} \left(x_{\text{path}} \sigma_{\text{path}} \prod_{\text{path}} w_{\text{path},k} \right) = [\boldsymbol{x} \circ \boldsymbol{\sigma}]^T \boldsymbol{w}_k = \boldsymbol{z}\boldsymbol{w}_k$$

Consider Taylor approximation:

$$\ell(w)|_{w=w^*} = \ell(w^*) + \frac{\partial \ell(w^*)}{\partial w}(w - w^*) + o(w - w^*)$$

Then,

$$\ell(w^* + \epsilon) = \ell(w^*) + \left[\frac{\partial \ell(w^*)}{\partial w}\right]^T \epsilon + o(\epsilon)$$

Observation of gredient direction ϵ :

$$\epsilon = -\frac{\partial \ell(w^*)}{\partial w}$$

Consider chain rule and $h = [\boldsymbol{x} \circ \boldsymbol{\sigma}]^T \boldsymbol{w} = \boldsymbol{z}\boldsymbol{w}$:

$$dh = d([\boldsymbol{x} \circ \boldsymbol{\sigma}]^T)\boldsymbol{w} + [\boldsymbol{x} \circ \boldsymbol{\sigma}]^T d\boldsymbol{w} = \boldsymbol{w}d\boldsymbol{z} + \boldsymbol{z}d\boldsymbol{w}$$

The derivative of $\boldsymbol{z}$

$$\boldsymbol{z} = \boldsymbol{x} \circ \boldsymbol{\sigma}$$

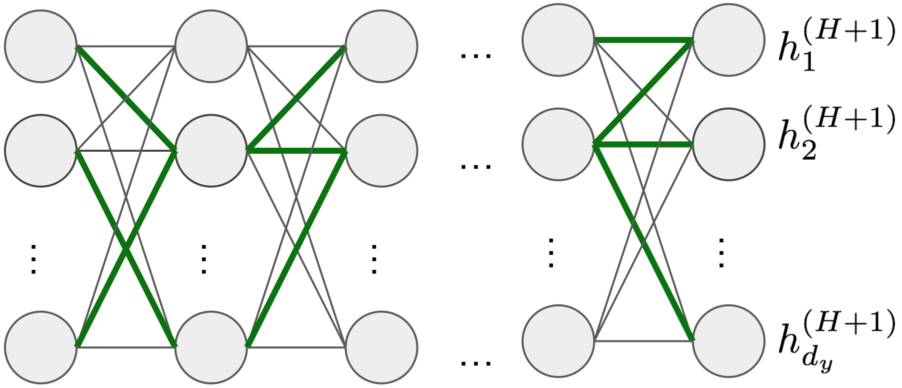
become a constant with respect to $\boldsymbol{w}$, i.e.

$$dh = \boldsymbol{z}d\boldsymbol{w}$$

Two-phase training

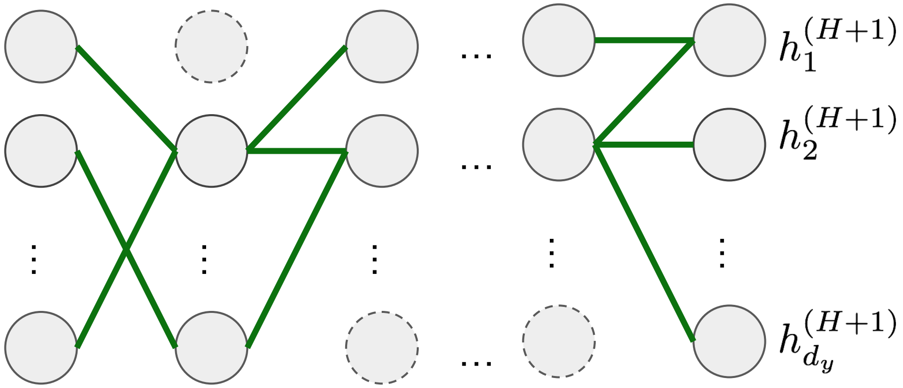
Standard Phase

Train the network in standard way with partial dataset;



Freeze Phase

Freeze activation pattern and keep training with the rest of dataset.



Why DARC helps implicit regularization?

Directly Approximately Regularizing Complexity (DARC)

Bound by Rademacher Complexity [Koltchinskii and Panchenko 2002]

Given a fixed ρ, with high probability $(1-\delta)$:

$$R[f] \leq \hat{R}_{m,\rho}[f] + \frac{2d_y^2}{\rho m}\mathfrak{R}'_m(F) + \sqrt{\frac{\ln\frac{1}{\delta}}{2m}}$$

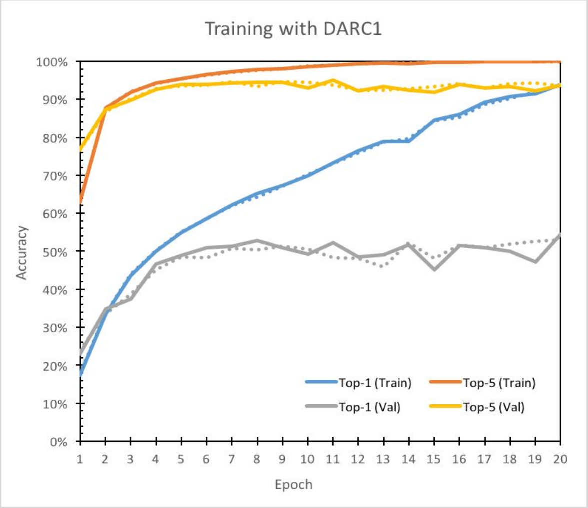
The approx. of Rademacher complx. converge to approx. of expect. over ***Sm.*** Thus:

$$\text{loss} = \text{original loss} + \frac{\lambda}{\bar{m}}\hat{\mathbb{E}}_{S_m,\xi}\left[\sup_k \sum_{i=1}^{\bar{m}} \xi_i h_k^{(H+1)}(x_i)\right]$$

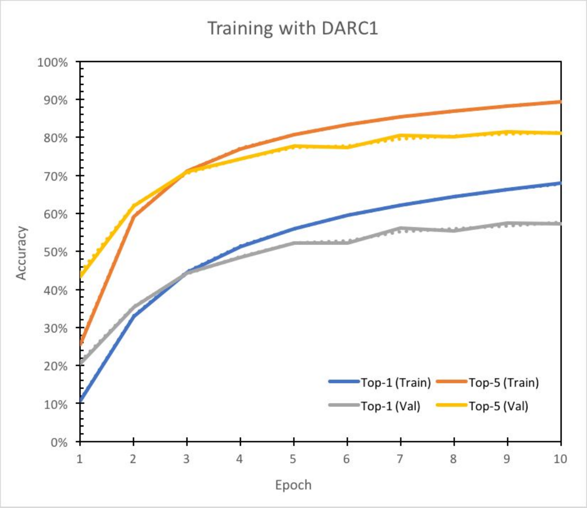
DARC1 on large dataset?

Experiment results: DARC1 Regularizer with ResNet-50 on ILSVRC2012

- **DARC1 Regularizer compare to baseline (ImageNet without DARC1 regularizer);**
  - Each line shows the average over 5 runs;
  - 10% training set held out as validation set;
  - Solid line uses DARC1 and dotted line is baseline.



10 Image classes



Entirely

Why SGD family?

Personal Opinions regarding SGD

- Deep networks only learned distribution of training data thanks to large nonlinearity activations, fortunately test data has same distribution;
- Generalization gap independent with traditional theory;
- However, global minima doens't mean best generalization;
- I think "flat minima hypothesis can generalize well" is true (sharp minima as well);
- SGD has the property to bypass saddle points (instead of stuck), therefore it can keep seek on flat surface;

We need more investigation.