Towards Understanding the Implicit Regularization in Deep Learning

Lei Wu (leiwu@princeton.edu)

PACM, Princeton University

July 18, 2020

SJTU Online Summer School of Deep Learning Theory

Outline

1. Motivation
2. Linear problem
3. Convex problems
4. Two-layer neural networks
5. General non-convex problems

Explicit regularization

- Let

$$\hat{\mathcal{R}}_n(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h(\boldsymbol{x}_i), y_i)$$
$$\mathcal{R}(h) = \mathbb{E}_{\boldsymbol{x},y}[\ell(\boldsymbol{x}, y)]$$

  be the empirical and population risks, respectively.
- **"Explicit regularization":** Solve the "explicitly" regularized problem:

$$\min_h \hat{\mathcal{R}}_n(h) + \lambda\|h\|$$

  - Ridge regression: $\min_{\boldsymbol{\beta}} \|X\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$.
  - Lasso: $\min_{\boldsymbol{\beta}} \|X\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$.
  - Kernel ridge regression: $\min_{h\in\mathcal{H}_k} \hat{\mathcal{R}}_n(h) + \lambda\|h\|_{\mathcal{H}_k}$.
  - ...

What is the so called “Implicit Regularization”

Definition (Informal)

Let $\mathcal{H}_m$ be the hypothesis space. $\mathcal{A} : S \mapsto \mathcal{H}_m$ is an algorithm to solve the following un-regularized problem:

$$\min_{h \in \mathcal{H}_m} \hat{\mathcal{R}}_n(h).$$

Consider the over-parameterized setting, i.e. there exist many solutions such that $\hat{\mathcal{R}}_n(h) = 0$.

- **“Implicit bias”**: The property that $\mathcal{A}$ always pick up certain solutions.
- **“Implicit regularization”**: The property that $\mathcal{A}$ always pick up solutions with small population risk.

An one-dimensional example [1]

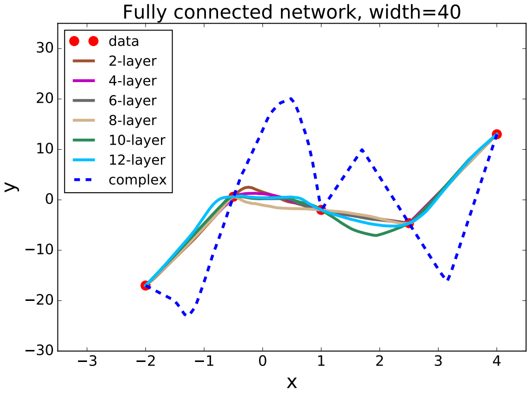


Figure: The algorithm is gradient descent (GD).

[1] Wu, Zhu and E, 2017

CIFAR-10 dataset

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

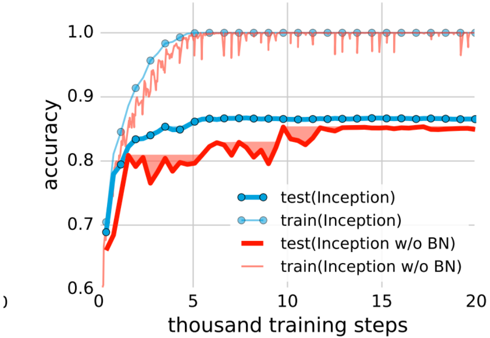


Figure: Figures from [Chiyuan Zhang, et al, ICLR2017]

Summary

- For neural network models, GD or SGD can always pick up solutions generalizing quite well.
- Explicit regularizations, such as weight decay, dropout, etc. only marginally improve the generalization performance, compared to implicit regularizations.
- Explicit regularizations could be critically important in some scenarios, such as highly noisy data, unsupervised learning (GAN), etc.

Let us start with the linear regression problem.

Linear regression

Consider the empirical risk:

$$\min \frac{1}{n} \|X\boldsymbol{\beta} - \boldsymbol{y}\|_2^2, \tag{1}$$

where $X \in \mathbb{R}^{n \times d}, \boldsymbol{\beta} \in \mathbb{R}^d$.
The GD algorithm for this problem is given by

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \frac{\eta}{n} X^T (X\boldsymbol{\beta}_t - \boldsymbol{y}). \tag{2}$$

Theorem

Consider the over-parameterized setting, i.e. $d > n$ and assume that $\boldsymbol{\beta}_0 = 0$. Then $\lim_{t \to \infty} \boldsymbol{\beta}_t = \boldsymbol{\beta}^*$, which is the minimum ℓ_2-norm solution given by

$$\boldsymbol{\beta}^* := \operatorname*{argmin}_{X\boldsymbol{\beta} = \boldsymbol{y}} \|\boldsymbol{\beta}\|^2$$

Proof

- Consider the continuous GD: $\dot{\boldsymbol{\beta}}(t) = -\frac{1}{n} X^T (X\boldsymbol{\beta}(t) - \boldsymbol{y})$.
- Let $X = U\Sigma V^T$ with $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$, $U \in \mathbb{R}^{n\times n}, V \in \mathbb{R}^{n\times d}$ be the SVD decomposition. Let $\boldsymbol{\beta}(t) = V\boldsymbol{a}(t) + V^{\perp}\boldsymbol{b}(t)$ with $\boldsymbol{a} \in \mathbb{R}^n, \boldsymbol{b} \in \mathbb{R}^{d-n}$.
- We have
$$V\dot{\boldsymbol{a}}(t) + V^{\perp}\dot{\boldsymbol{b}}(t) = -V\Sigma^2\boldsymbol{a}(t) + V\Sigma U^T\boldsymbol{y}.$$
- Let $\boldsymbol{z} = U^T\boldsymbol{y}$, we have that
$$\begin{aligned} \dot{\boldsymbol{b}}(t) &= 0 \\ \dot{\boldsymbol{a}}(t) &= -\frac{1}{n}\Sigma^2\boldsymbol{a}(t) + \frac{1}{n}\Sigma\boldsymbol{z}, \end{aligned}$$
which gives that $a_j(t) = a_j(0)e^{-t\sigma_j^2/n} + \frac{z_j}{\sigma_j}(1 - e^{-t\sigma_j^2/n})$.
- $\lim_{t\to\infty} \boldsymbol{\beta}(t) = \underbrace{V\Sigma^{-1}U^T\boldsymbol{y}}_{\text{minimum-norm solution}} + V^{\perp}\boldsymbol{b}(0)$.

The generalization error of GD solutions

Consider the label is generated by $\boldsymbol{y} = X\boldsymbol{\beta}^* + \boldsymbol{\xi}$ with $\boldsymbol{\beta}^* = V\boldsymbol{a}^* + V^\perp \boldsymbol{b}^*$.

- $\boldsymbol{z} = U^T\boldsymbol{y} = U^T(U\Sigma V^T\boldsymbol{\beta}^* + \boldsymbol{\xi}) = \Sigma\boldsymbol{a}^* + U^T\boldsymbol{\xi}$.
- For the zero initialization, $a_j(t) = \frac{a_j^*\sigma_j}{\sigma_j}(1 - e^{-\sigma_j^2 t/n}) + \frac{\boldsymbol{u}_j^T\boldsymbol{\xi}}{\sigma_j}(1 - e^{-\sigma_j^2 t/n})$.
- Let $e(t) := \|\boldsymbol{\beta}(t) - \boldsymbol{\beta}^*\|^2$. We have

$$\begin{aligned} e(t) &= \|\boldsymbol{b}(t) - \boldsymbol{b}^*\|^2 + \|\boldsymbol{a}(t) - \boldsymbol{a}^*\|^2 \\ &\leq \|\boldsymbol{b}^*\|^2 + 2\sum_{j=1}^{n}(a_j^*)^2 e^{-2\sigma_j^2 t/n} + 2\sum_j \frac{(\boldsymbol{u}_j^T\boldsymbol{\xi})^2}{\sigma_j^2}(1 - e^{-\sigma_j^2 t/n})^2 \end{aligned}$$

- $\mathbb{E}[e(\infty)] = \sum_j \frac{\mathbb{E}[(\boldsymbol{u}_j^T \boldsymbol{\xi})^2]}{\sigma_j^2} + \|\boldsymbol{b}^*\|^2 = \sum_j \frac{\varepsilon^2}{\sigma_j^2} + \|\boldsymbol{b}^*\|^2$. This minimum-norm solution can be very bad if σ_j is very small.
- Can we avoid this bad case.

$$\min_t \underbrace{\sum_{j=1}^{n} (a_j^*)^2 e^{-2\sigma_j^2 t/n}}_{\text{learning from signals}} + \underbrace{\sum_j \frac{(\boldsymbol{u}_j^T \boldsymbol{\xi})^2}{\sigma_j^2} (1 - e^{-\sigma_j^2 t/n})^2}_{\text{overfitting to the noise}}$$

- The noise is uniformally spread over the different eigenfunctions. The smallest is the σ_j, the slower is the overfitting to the noise.
- The signal typically concentrates on the top eigenfunctions, which enables the fast learning of singal.

Slow deterioration of the generalization error

Theorem

Consider $\boldsymbol{\beta}^* = \boldsymbol{v}_1$. Then we have the GD solutions satisfy that

$$\mathbb{E}[e(t)] \lesssim \underbrace{e^{-2\sigma_1^2 t/n}}_{\textit{Exponential learning}} + \underbrace{\varepsilon^2 t}_{\textit{Linear deterioration}} .$$

Taking $T_0 = \frac{n}{2\sigma_1^2}\log(\frac{2\sigma_1^2}{n\varepsilon^2})$, *we have*

$$\mathbb{E}[e(T_0)] \lesssim \frac{n\varepsilon^2}{\sigma_1^2}\log^2(\frac{2\sigma_1^2}{n\varepsilon^2})$$

Up to log terms:

$$\frac{1}{n}\|\boldsymbol{\beta}(T_0) - \boldsymbol{\beta}^*\| \lesssim \varepsilon^2$$

The same theorem for random feature model can be found in [Ma, Wu and E, MSML 2020].

Proof:

$$\begin{aligned}\mathbb{E}[e(t)] &\lesssim e^{-2\sigma_1^2 t/n} + \sum_j \frac{\varepsilon^2}{\sigma_j^2}(1 - e^{-\sigma_j^2 t/n})^2 \\ &\lesssim e^{-2\sigma_1^2 t/n} + \varepsilon^2 \sum_j \frac{1}{\sigma_j^2} \min\{1, \frac{\sigma_j^4 t^2}{n^2}\} \\ &\lesssim e^{-2\sigma_1^2 t/n} + \varepsilon^2 \sum_j \min\{\frac{1}{\sigma_j^2}, \frac{\sigma_j^2 t^2}{n^2}\} \\ &\lesssim e^{-2\sigma_1^2 t/n} + \varepsilon^2 \sum_j \sqrt{\frac{t^2}{n^2}} \\ &= e^{-2\sigma_1^2 t/n} + \varepsilon^2 t\end{aligned}$$

The third inequality follows from that $1 - e^{-x} \leq \min\{1, x\}$; the fourth inequality is due to $\min\{a, b\} \leq \sqrt{ab}$.

Random feature models

$$f_m(\boldsymbol{x};\boldsymbol{a}) := \sum_{j=1}^{m} a_j \sigma(\boldsymbol{w}_j^T \boldsymbol{x})$$

with $\{\boldsymbol{w}_j\}$ i.i.d. sampled from a fixed distribution π_0. The empirical risk is given

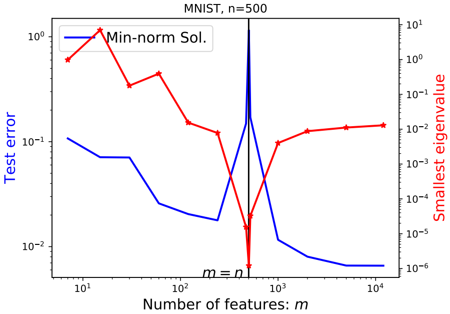
$$\hat{\mathcal{R}}(\boldsymbol{a}) := \frac{1}{n}\|\Psi\boldsymbol{a} - \boldsymbol{y}\|^2,$$

where $\Psi = (\sigma(\boldsymbol{w}_j^T \boldsymbol{x}_k)) \in \mathbb{R}^{n\times m}$.
The generalization error is affected by the spectrum of the Gram matrix $G = \frac{1}{n}\Psi^T\Psi \in \mathbb{R}^{m\times m}$

$$G_{i,j} = \frac{1}{n}\sum_{k=1}^{n} \sigma(\boldsymbol{w}_i^T \boldsymbol{x}_k)\sigma(\boldsymbol{w}_j^T \boldsymbol{x}_k).$$

Generalization error curve

- The following figure show how the test errors of GD solutions depend on the number of features.



- The double descent [2] phenomenon happens even when the label is clean. In this case, the intrinsic noise comes from the approximation error, i.e. $f^*(\boldsymbol{x}) = f_m(\boldsymbol{x}; \boldsymbol{a}^*) + \varepsilon(\boldsymbol{x})$.

[2] Belkin, Hsu, Ma and Mandal, PNAS2019

The slow deterioration of the generalization error

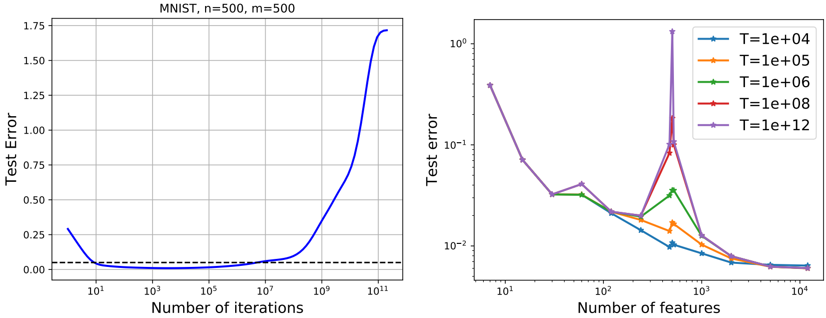


Figure: figures from [Ma, Wu and E, MSML2020]

Does GD still converge to the minimum ℓ_2 norm solution for general convex problems ?

Implicit bias of GD for general convex problems

Theorem

Assume that empirical risk $\hat{\mathcal{R}}_n(\cdot)$ is convex. Let θ_t is the GD solution at time t and $\bar{\theta}$ is the minimum-norm solution. Then we have

$$\|\theta_t\| \leq \|\theta_0\| + 2\|\bar{\theta}\|$$

Proof: Define

$$J(t) = t(\hat{\mathcal{R}}_n(\theta_t) - \hat{\mathcal{R}}_n(\bar{\theta})) + \frac{1}{2}\|\theta_t - \bar{\theta}\|^2$$

Using the convexity of $\hat{\mathcal{R}}$, we have

$$\frac{dJ(t)}{dt} = \hat{\mathcal{R}}_n(\theta_t) - \hat{\mathcal{R}}_n(\bar{\theta}) + \langle \theta_t - \bar{\theta}, -\nabla \hat{\mathcal{R}}_n(\theta_t)\rangle - t\|\nabla \hat{\mathcal{R}}_n(t)\|^2 \leq 0.$$

So $J(t) \leq J(0)$, i.e.

$$t(\hat{\mathcal{R}}_n(\theta_t) - \hat{\mathcal{R}}_n(\bar{\theta})) + \frac{1}{2}\|\theta_t - \bar{\theta}\|^2 \leq \frac{1}{2}\|\theta_0 - \bar{\theta}\|^2.$$

This gives us

$$\|\theta_t - \bar{\theta}\| \leq \|\theta_0 - \bar{\theta}\|.$$

Let us go beyond the convexity!

Two-layer neural networks

Consider two-layer neural networks:

$$f_m(\boldsymbol{x};\boldsymbol{a},\boldsymbol{B})=\sum_{j=1}^{m}a_j\sigma(\boldsymbol{b}_j^T\boldsymbol{x})=\boldsymbol{a}^T\sigma(\boldsymbol{B}\boldsymbol{x}). \tag{3}$$

The empirical and population error is given by

$$\hat{\mathcal{R}}_n(\boldsymbol{a},\boldsymbol{B})=\frac{1}{n}\sum_{i=1}^{n}(f_m(\boldsymbol{x}_i;\boldsymbol{a},\boldsymbol{B})-f^*(\boldsymbol{x}_i))^2 \tag{4}$$

$$\mathcal{R}(\boldsymbol{a},\boldsymbol{B})=\mathbb{E}_{\boldsymbol{x}}[(f_m(\boldsymbol{x};\boldsymbol{a},\boldsymbol{B})-f^*(\boldsymbol{x}))^2]. \tag{5}$$

GD with Xavier-type initialization

To optimize the empirical error, the typical optimizer used is the GD dynamics:

$$\dot{\boldsymbol{a}}_t = -\frac{1}{n}\sum_{i=1}^{n}(f_m(\boldsymbol{x}_i;\boldsymbol{a},\boldsymbol{B}) - f^*(\boldsymbol{x}_i))\sigma(\boldsymbol{B}_t\boldsymbol{x}_i) \tag{6}$$

$$\dot{\boldsymbol{B}}_t = -\frac{1}{n}\sum_{i=1}^{n}(f_m(\boldsymbol{x}_i;\boldsymbol{a},\boldsymbol{B}) - f^*(\boldsymbol{x}_i))\left(\boldsymbol{a}_t \circ \sigma'(\boldsymbol{B}_t\boldsymbol{x}_i)\right)\boldsymbol{x}_i^T \tag{7}$$

with the Xavier-type initialization

$$\boldsymbol{a}_j(0) \sim \mathcal{N}(0,\beta^2), \qquad \boldsymbol{b}_j(0) \sim \mathcal{N}(0, I/d). \tag{8}$$

Previous results on GD dynamics for neural networks

- **Highly over-parameterized Regime:** $m \geq Cn^2/\lambda_n^4$ [W. E, C. Ma and L. Wu 2019] [3]

$$\sup_{\boldsymbol{x}\in\mathbb{S}^{d-1}} |f_m(\boldsymbol{x};\boldsymbol{a}_t,\boldsymbol{B}_t) - f_m(\boldsymbol{x};\boldsymbol{a}_t,\boldsymbol{B}_0)| \leq O\left(\frac{\lambda_n^{-1}}{\sqrt{m}}\right) \tag{9}$$

$$\sup_{t\geq 0} \|\boldsymbol{B}_t - \boldsymbol{B}_0\|_F \leq O\left(\frac{\lambda_n^{-1}}{\sqrt{m}}\right) \tag{10}$$

Key Observation: Time scale separation

$$\dot{a}_j(t) \sim O(\|\boldsymbol{b}_j\|) = O(1) \tag{11}$$

$$\dot{\boldsymbol{b}}_j(t) \sim O(|a_j|) = O\left(\frac{\lambda_n^{-1}}{m}\right) \tag{12}$$

Conclusion: NN degenerates to Random feature model in the highly over-parametrized regime.

[3] See also Sanjeev Arora, Simon S Du et al. 2019

People do observe that NN models perform better in practice than RF models.

Questions:

- Implicit regularization in “mildly” over-parametrized regime?
- Qualitative behavior of the GD dynamics

What happens to the time scale separation?

Investigation of the regime: $m < \infty, n = \infty$

In this regime, when $\boldsymbol{x} \sim \text{Uniform}(\mathbb{S}^{d-1})$, the population loss has analytic expression [4]:

$$\mathcal{R}(\boldsymbol{a}, \boldsymbol{B}) = \mathbb{E}_{\boldsymbol{x}}[(\sum_{j=1}^{m} a_j \sigma(\boldsymbol{b}_j \cdot \boldsymbol{x}) - \int a^*(\boldsymbol{b}) \sigma(\boldsymbol{b}^* \cdot \boldsymbol{x}) d\pi^*(\boldsymbol{b}))^2] \tag{13}$$

$$= \sum_{j_1, j_2 = 1}^{m} a_{j_1} a_{j_2} k(\boldsymbol{b}_{j_1}, \boldsymbol{b}_{j_2}) - 2 \sum_{j=1}^{m} \int a_j a^*(\boldsymbol{b}') k(\boldsymbol{b}_j, \boldsymbol{b}) d\pi^*(\boldsymbol{b})$$

$$+ \int a^*(\boldsymbol{b}) a^*(\boldsymbol{b}') k(\boldsymbol{b}, \boldsymbol{b}') d\pi^*(\boldsymbol{b}) d\pi^*(\boldsymbol{b}), \tag{14}$$

where

$$k(\boldsymbol{b}, \boldsymbol{b}') := \mathbb{E}_{\boldsymbol{x}}[\sigma(\boldsymbol{b} \cdot \boldsymbol{x}) \sigma(\boldsymbol{b}' \cdot \boldsymbol{x})] = \|\boldsymbol{b}\| \|\boldsymbol{b}'\| \left(\sin\theta + (\pi - \theta)\cos\theta\right), \tag{15}$$

with $\theta = \arccos(\langle \hat{\boldsymbol{b}}, \hat{\boldsymbol{b}}' \rangle)$.

[4] The derivation of the kernel can be found in [Youngmin Cho and Lawrence K. Saul, NIPS2009]

Single neuron: $f^*(\boldsymbol{x}) = \sigma(x_1)$

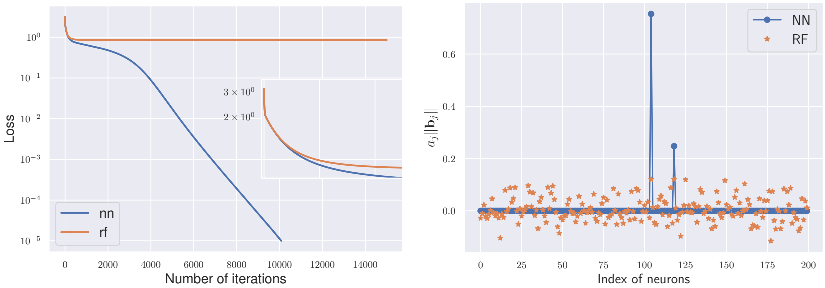


Figure: **(Left)** The dynamic behavior of the loss ; **(Right)** The magnitude of each neurons of the convergent solution. In this experiment, $m = 200, d = 100$.

Observation

- Initially, NN is close to the RF model
- NN and RF depart around the time when RF's loss saturates.
- The final NN solution is very sparse, and only one neuron contributes to the model in this case.

The dynamics of neurons

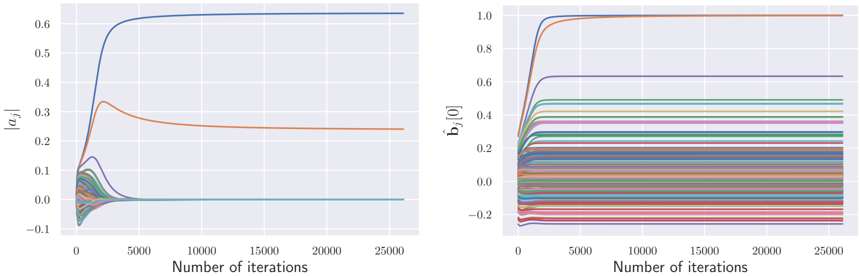


Figure: The dynamics of Top-5 neurons. The top-1 neuron keeps activated, while the other 4 neurons die slowly. Click for video.

Observation:

We can observe that:

- **Phase 1:** NN behave like random feature. This phase ends around the time when RF's loss saturates.
- **Transition:** A small portion of neurons are activated, since their $\boldsymbol{b}$'s begin to move significantly.
- **Phase 2:**
  - Activation: one-neuron keeps activated and its magnitude keeps increasing.
  - Deactivation: Other activated neurons die slowly.
- **Final Solution:** Only two neurons contribute to the model.

Linear target function

- $f^*(\boldsymbol{x}) = x_1 = \sigma(x_1) - \sigma(-x_1)$. $m = 200, d = 100$. Moreover, $\|f^*\|_{\mathcal{H}_{k_{\pi_0}}} \propto \sqrt{d}$, and $\|f^*\|_{\mathcal{B}} \leq 2$.

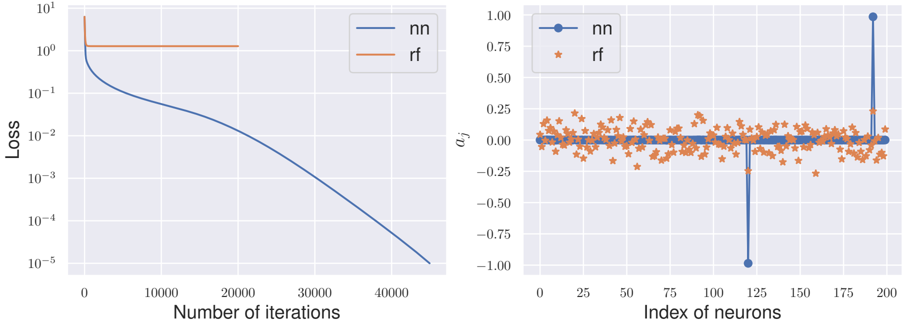


Figure: RKHS functions can also exhibit the activation process.

Linear target function

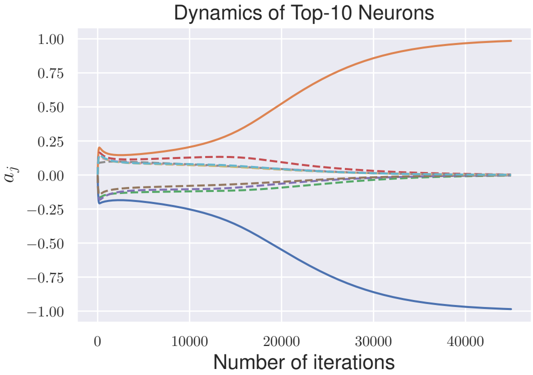


Figure: Watch the videos showing the dynamics all the neurons by clicking link1 and link2.

Finite neurons target function

Learning finite neurons $f^*(\boldsymbol{x}) = \sum_{j=1}^{m^*} a_j^* \sigma(\boldsymbol{b}_j^* \cdot \boldsymbol{x})$.

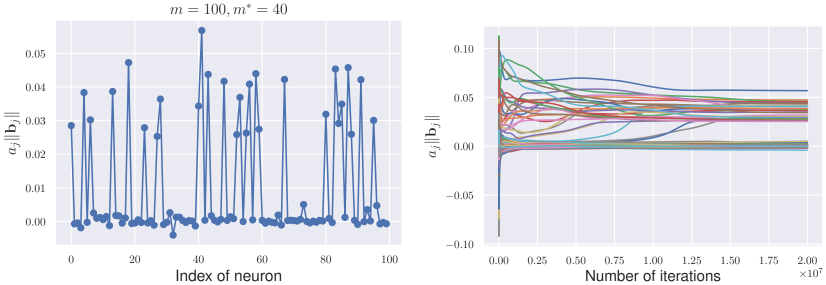


Figure: $m = 100, m^* = 40$ Learning finite neurons with $a_j^* = 1/m^*, \boldsymbol{b}_j^* \sim \pi_0$.

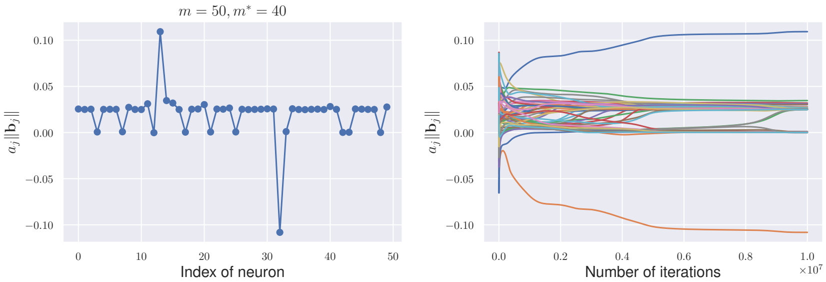


Figure: $m = 50, m^* = 40$

- Clear active and background neurons separation.
- GD tends to find sparse solutions.

Circle neuron target function

One target function which cannot be exactly expressed by finite neurons.

$$f_2^*(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{b}\sim\pi_2}[\sigma(\boldsymbol{b}^T\boldsymbol{x})], \tag{16}$$

where π_2 is the uniform distribution over the unit circle

$$\Gamma = \{\boldsymbol{b} : b_1^2 + b_2^2 = 1 \text{ and } b_i = 0 \ \forall i > 2\}.$$

Training curve

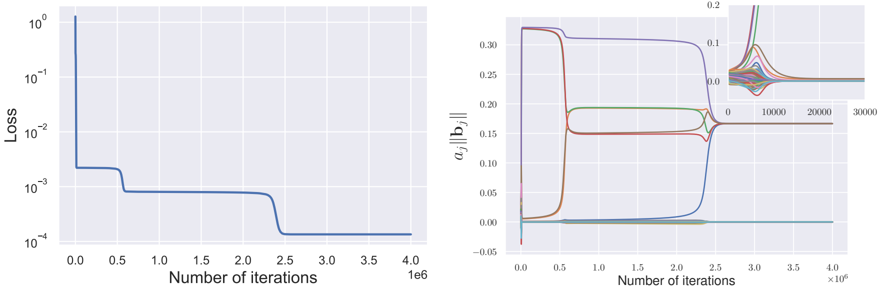


Figure: $m = 100, d = 100$.

Observations from the loss curve:

- The loss decreases in a “step-like” fashion in the second phase, suggesting an activation process.

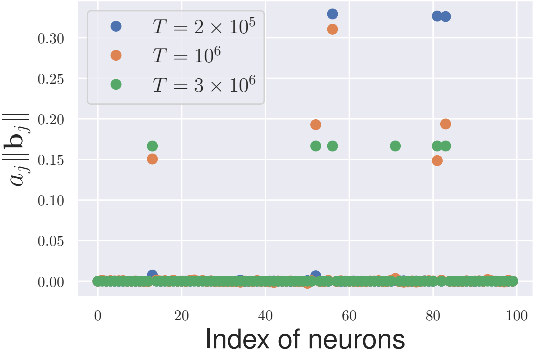


Figure: Solutions at the three steps.

What happens when data are finite?

- **Highly over-parameterized regime:** $m \gg n$.
- **Mildly over-parameterized regime:** $n/d \leq m \lesssim n$.
- **Under-parameterized regime:** $m < n/d$.

Highly over-parameterized regime: $m = cn^2$

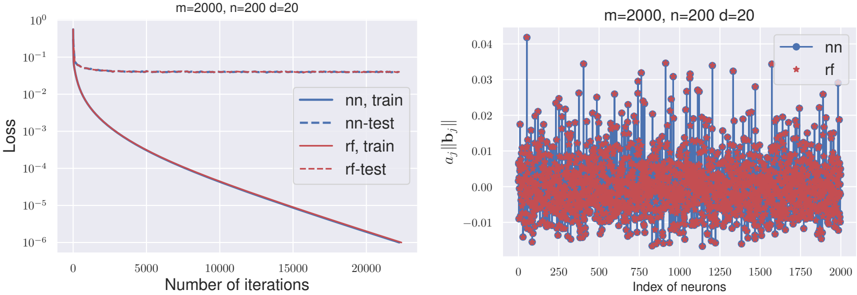


Figure: $m = 0.5n^2$, target function $f^*(\boldsymbol{x}) = \sigma(x_1)$. The numerical result is consistent with theoretical prediction in [E, Ma, Wu, 2019].

Under-parameterized regime: $m < n/d$

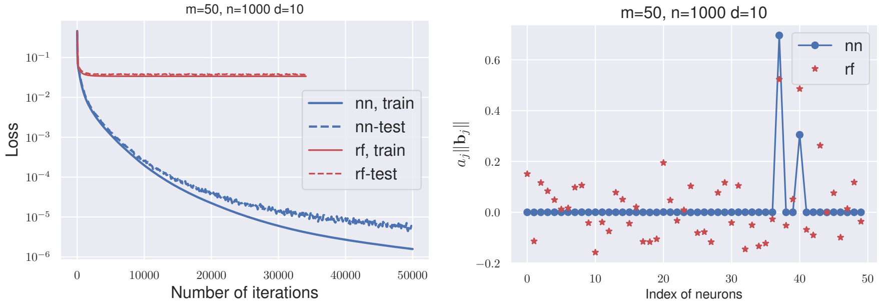


Figure: **(Left)** The dynamics of training and test errors. **(Right)** The final solutions. $m = 0.5n/d$. Here, $f^*(\boldsymbol{x}) = \sigma(x_1)$.

Midly over-parameterized regime: $m = cn/d$

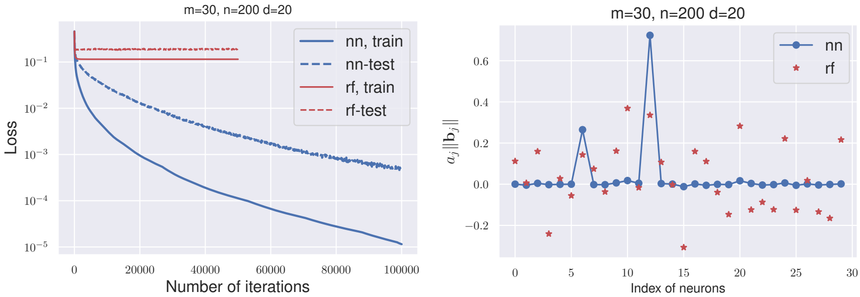


Figure: $m = 3n/d$. Here, $f^*(\boldsymbol{x}) = \sigma(x_1)$.

Mildly over-parameterized regime: $m = Cn$

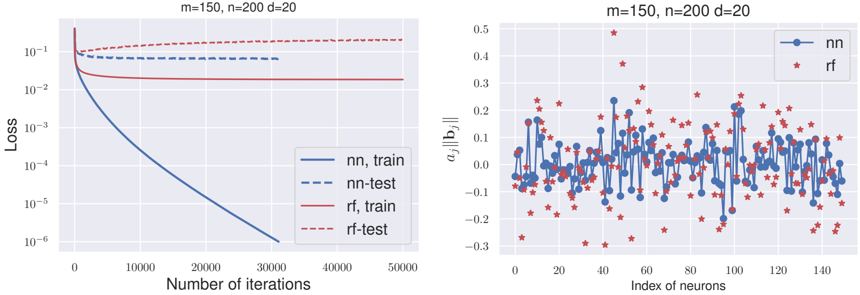


Figure: $m = 0.75n$. Here, $f^*(\boldsymbol{x}) = \sigma(x_1)$.

Test error curve

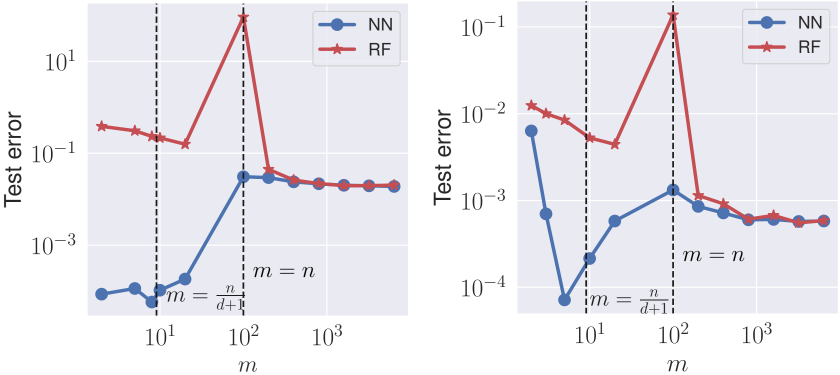


Figure: (Left) Single neuraon; (Right) Circle neuron.

Heatmap of test errors

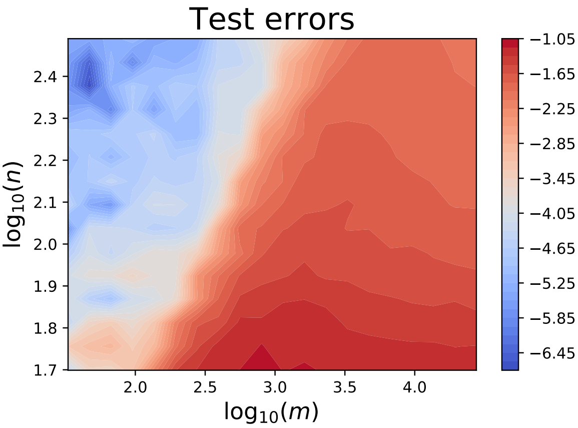


Figure: Single neuron.

Generalization Error: Avoiding the curse of dimensionality (CoD)

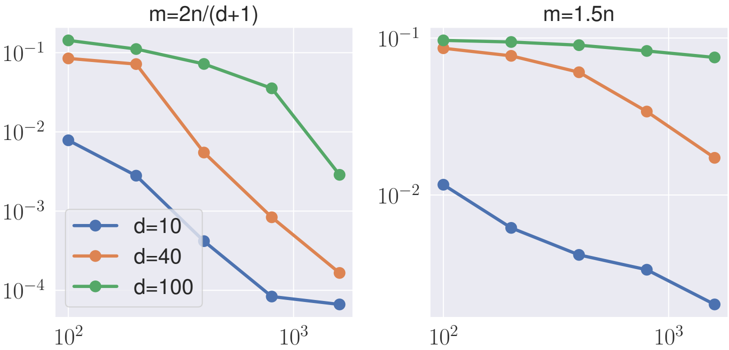


Figure: The test errors of GD solutions for the circle neuron target function. The GD is stopped when the training error becomes smaller than 10^{-5}.

Summary:

- **Under-parameterized regime:** $m < n/d$. Dynamics in this regime behaves like the case where $n = \infty$.
- **Mildly over-parameterized regime:** In this regime, a transition happens as m increases:
    - $m \sim n/d$, the training behaves like NN, the gen-err does not suffer from the curse of dimensionality.
    - $m \sim n$, the training behaves like kernel and the gen-err suffers CoD.
    - The transition from "NN-like" to "kernel-like" as we increase m.
- **Highly over-parameterized regime:** $m \gg n$. Dynamics in this regime behaves like that of the kernel method, and tends to the random feature model as m tends to infinity.

GD dynamics under mean-field scaling

$$f_m(\boldsymbol{x};\boldsymbol{a},\boldsymbol{B})=\frac{1}{m}\sum_{j=1}^{m}a_j\sigma(\boldsymbol{b}_j^T\boldsymbol{x}), \tag{17}$$

and the corresponding GD dynamics is given by

$$\begin{aligned}\dot{a}_j(t)&=-\frac{1}{mn}\sum_{i=1}^{n}(f_m(\boldsymbol{x}_i;\boldsymbol{a}(t),\boldsymbol{B}(t))-f^*(\boldsymbol{x}_i))\sigma(\boldsymbol{b}_j(t)^T\boldsymbol{x}_i)\\ \dot{\boldsymbol{b}}_j(t)&=-\frac{1}{mn}\sum_{i=1}^{n}(f_m(\boldsymbol{x}_i;\boldsymbol{a}(t),\boldsymbol{B}(t))-f^*(\boldsymbol{x}_i))a_j(t)\sigma'(\boldsymbol{b}_j(t)^T\boldsymbol{x}_i)\boldsymbol{x}_i.\end{aligned} \tag{18}$$

Dynamical behavior of the mean-field GD

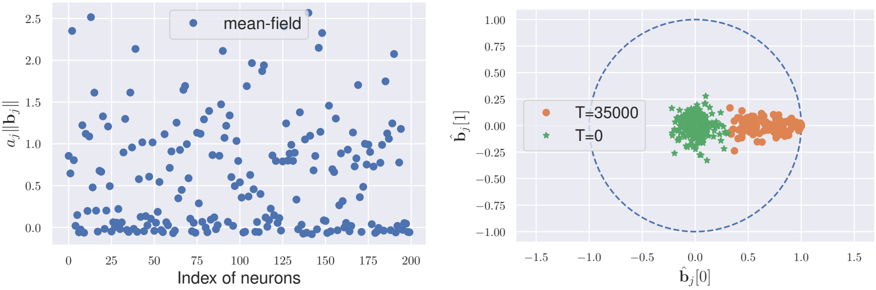


Figure: Single neuron target function.

Generalization error

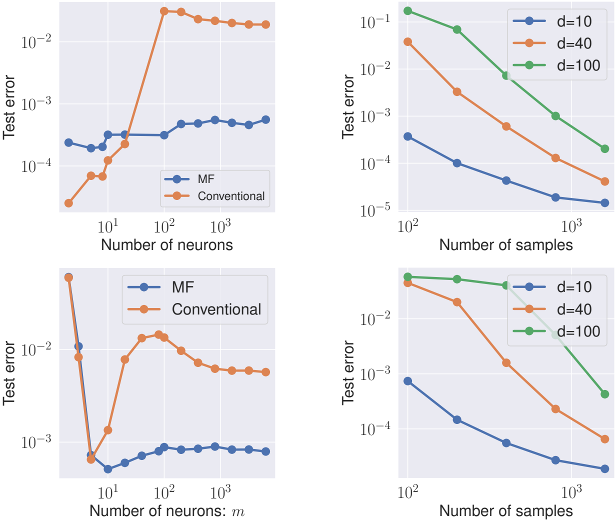


Figure: (1th row) the single neuron target function; (2nd row) the circle neuron target function.

Comparison between two scalings

- GD-conventional to pick up sparse solution, while GD-MF does not.
- GD-conventional is very sensitive to the network widths, while GD-MF is quite robust.
- GD-conventional with the optimal hyperparameter performs similar as GD-MF.

Connection to the network pruning

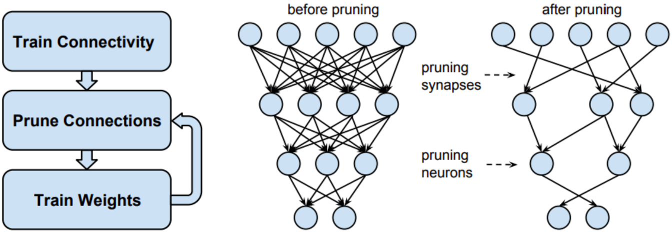


Figure: [Song Han, et al, NIPS2015]

- The implicit bias of GD-conventional provides an explanation why the above pruning method works.
- This pruning method may not work for models trained with GD-MF.

Practitioners observe that SGD always outperform GD in terms of generalization.

SGD v.s. GD

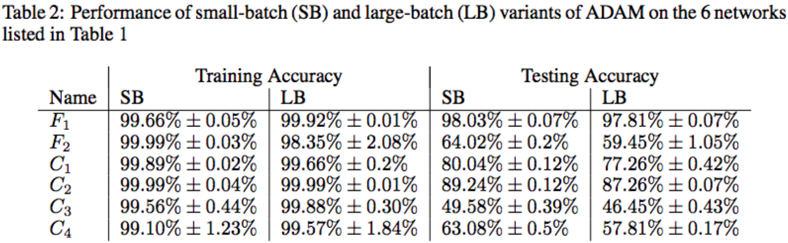


Table 2: Performance of small-batch (SB) and large-batch (LB) variants of ADAM on the 6 networks listed in Table 1

	Training Accuracy		Testing Accuracy	
Name	SB	LB	SB	LB
F_1	$99.66\% \pm 0.05\%$	$99.92\% \pm 0.01\%$	$98.03\% \pm 0.07\%$	$97.81\% \pm 0.07\%$
F_2	$99.99\% \pm 0.03\%$	$98.35\% \pm 2.08\%$	$64.02\% \pm 0.2\%$	$59.45\% \pm 1.05\%$
C_1	$99.89\% \pm 0.02\%$	$99.66\% \pm 0.2\%$	$80.04\% \pm 0.12\%$	$77.26\% \pm 0.42\%$
C_2	$99.99\% \pm 0.04\%$	$99.99\% \pm 0.01\%$	$89.24\% \pm 0.12\%$	$87.26\% \pm 0.07\%$
C_3	$99.56\% \pm 0.44\%$	$99.88\% \pm 0.30\%$	$49.58\% \pm 0.39\%$	$46.45\% \pm 0.43\%$
C_4	$99.10\% \pm 1.23\%$	$99.57\% \pm 1.84\%$	$63.08\% \pm 0.5\%$	$57.81\% \pm 0.17\%$

Figure: Tables from [Nitish S. Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, ICLR2017]

Flatness

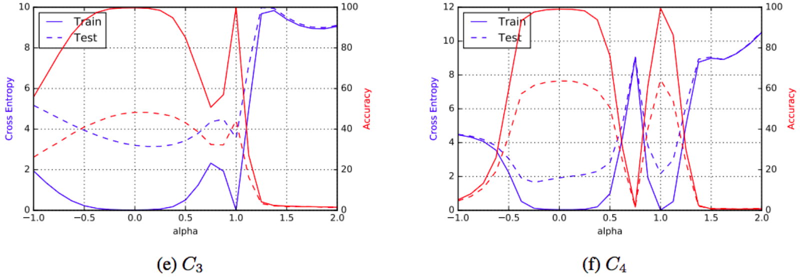


Figure: Figures from [Nitish S. Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, ICLR2017]

Flatness

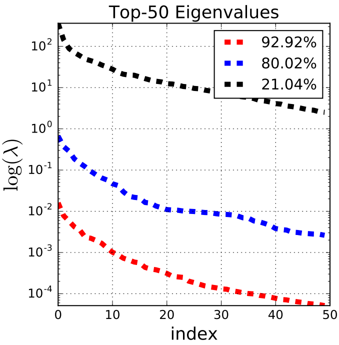


Figure: The spectrums of Hessian matrices of global minima with different generalization errors. Figures from [Lei Wu, Zhanxing Zhu, Weinan E, 2017]

Caution: Until now, there does not exist any “true theory” to connect the local geometric property to generalization performances.

Escape phenomenon

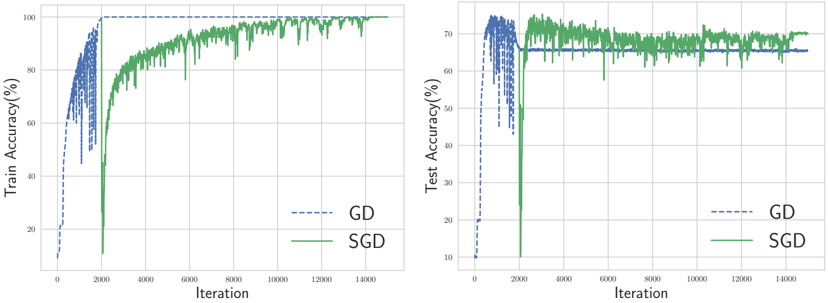
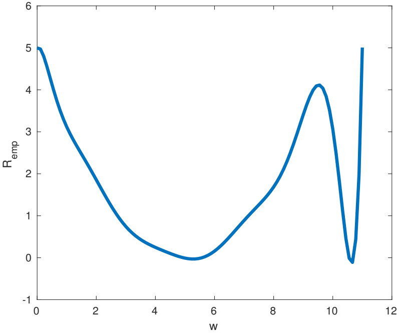


Figure: Fast escape phenomenon in fitting corrupted FashionMNIST. This escape phenomenon shows that the GD solutions are unstable for SGD dynamics.

Questions

Is it because GD picks up a sharp minima, which is not stable for SGD?



An illustrative counter example

Consider an one-dimensional problem $f(x) = \frac{1}{2}(f_1(x) + f_2(x))$ with

$$f_1(x) = \min\{x^2, 0.1(x-1)^2\},\ \ f_2(x) = \min\{x^2, 1.9(x-1)^2\}.$$

This function has two global minima at $x = 0$ and $x = 1$.

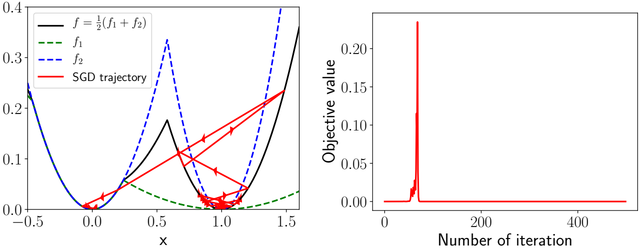


Figure: Motivating example. (Left) One trajectory of SGD with learning rate $\eta = 0.7, x_0 = 1 - 10^{-5}$, showing convergence to 0. GD with the same learning rate will converge to 1.

Explanation

- The two minima has the same sharpness $f'' = 1$, so the two minima are both stable for GD with $\eta <= 2/f'' = 2$.
- However, for SGD, in each iteration,it randomly picks one function from f_1 and f_2 and applies gradient descent to that function.
- Since $f_1''(1) = 0.1, f_2''(1) = 1.9$, SGD with the learning rate $\eta = 0.7$ is stable for f_1 but unstable for f_2. Thus $x = 1$ is not stable. In contrast, $\eta = 0.7$ is stable for both f_1 and f_2 around $x = 0$ since $f_1''(0) = f_2''(0) = 1$.

Dynamical stability analysis: One-dimensional case

- Consider the one-dimensional problem:

$$f(x) = \frac{1}{2n}\sum_{i=1}^{n} a_i x^2, \quad a_i \geq 0 \; \forall i \in [n] \tag{19}$$

Dynamical stability analysis: One-dimensional case

- Consider the one-dimensional problem:

$$f(x) = \frac{1}{2n}\sum_{i=1}^{n} a_i x^2, \quad a_i \geq 0 \; \forall i \in [n] \tag{19}$$

- The SGD iteration is given by,

$$x_{t+1} = x_t - \eta a_\xi x_t = (1 - \eta a_\xi) x_t, \tag{20}$$

Dynamical stability analysis: One-dimensional case

- Consider the one-dimensional problem:

$$f(x) = \frac{1}{2n}\sum_{i=1}^{n} a_i x^2, \quad a_i \geq 0 \ \forall i \in [n] \tag{19}$$

- The SGD iteration is given by,

$$x_{t+1} = x_t - \eta a_\xi x_t = (1-\eta a_\xi) x_t, \tag{20}$$

- So after one step update, we have

$$\begin{aligned} \mathbb{E}x_{t+1} &= (1-\eta a)\mathbb{E}x_t, && (21) \\ \mathbb{E}x_{t+1}^2 &= \left[(1-\eta a)^2 + \eta^2 s^2\right] \mathbb{E}x_t^2, && (22) \end{aligned}$$

where $a = \frac{1}{n}\sum_{i=1}^n a_i$, $s = \sqrt{\frac{1}{n}\sum_{i=1}^n a_i^2 - a^2}$. We call a: sharpness s: non-uniformity.

Stability Condition

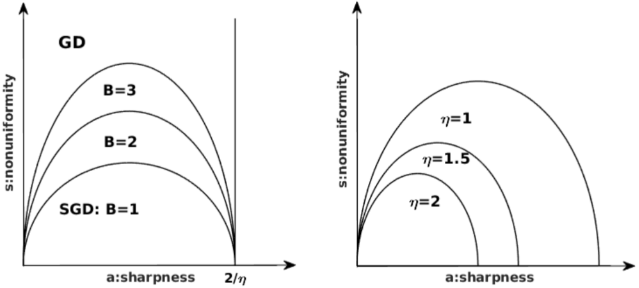
- Stability condition: $(1-\eta a)^2 + \eta^2 s^2 \leq 1$.
- Otherwise, a small perturbation will lead SGD to escape from 0.

$$\mathbb{E}[x_t^2] = [(1-\eta a)^2 + \eta^2 s^2]^t \mathbb{E}[x_0^2].$$

- SGD with batch size B:

$$(1-\eta a)^2 + \frac{\eta^2 (n-B)}{B(n-1)} s^2 \leq 1, \quad s \geq 0. \tag{23}$$

- Diagram:



High-dimensional Case

- Consider an optimization problem with quadratic objective function

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{2n}\sum_{k=1}^{n} x^T H_k x, \qquad H_k \succeq 0.$$

High-dimensional Case

- Consider an optimization problem with quadratic objective function

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{2n}\sum_{k=1}^{n} x^T H_k x, \qquad H_k \succeq 0.$$

- SGD with batch size B:

$$x_{t+1} = x_t - \tfrac{\eta}{B}\sum_{i=1}^{B} H_{\xi_i} x_t,$$

High-dimensional Case

- Consider an optimization problem with quadratic objective function

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{2n}\sum_{k=1}^{n} x^T H_k x, \qquad H_k \succeq 0.$$

- SGD with batch size B:

$$x_{t+1} = x_t - \tfrac{\eta}{B}\sum_{i=1}^{B} H_{\xi_i} x_t,$$

- We have

$$\begin{aligned}\mathbb{E}[x_{t+1}] &= \mathbb{E}\left[(I-\eta H)x_t\right]\\ \mathbb{E}\|x_{t+1}\|^2 &= \mathbb{E}x_t^T\left[(I-\eta H)^2 + \frac{\eta^2(n-B)}{B(n-1)}\Sigma\right]x_t,\end{aligned}$$

with$H = \frac{1}{n}\sum_{i=1}^{n} H_i$, $\Sigma = \frac{1}{n}\sum_{i=1}^{n} H_i^2 - H^2$.

Stability Condition

- Global minimum 0 is stable for SGD if

$$\lambda_{\max}\left\{(I-\eta H)^2+\frac{\eta^2(n-B)}{B(n-1)}\Sigma\right\}\leq 1.$$

Stability Condition

- Global minimum 0 is stable for SGD if

$$\lambda_{\max}\left\{(I-\eta H)^2+\frac{\eta^2(n-B)}{B(n-1)}\Sigma\right\}\le 1.$$

- Let $a=\lambda_{\max}(H)$, $s^2=\lambda_{\max}(\Sigma)$, then a necessary condition is

$$0\le a\le \frac{2}{\eta},\quad 0\le s\le \frac{1}{\eta}\sqrt{\frac{B(n-1)}{n-B}}\approx\frac{\sqrt{B}}{\eta}.$$

Sharpness-non-uniformity diagram

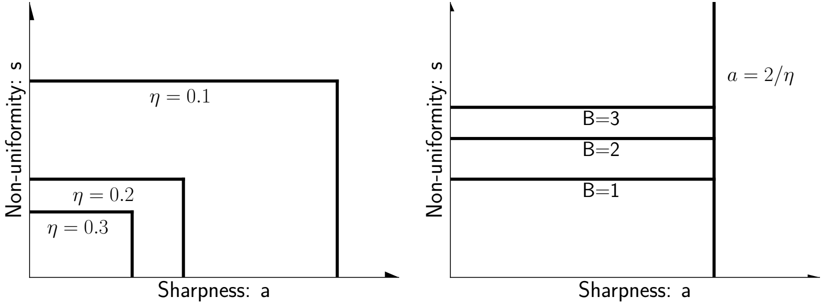


Figure: We use sharpness and non-uniformity to describe different global minima. The figures shows the condition on sharpness and non-uniformity for SGD with different batch-size and learning rate to be stable.

Dynamical Stability of a General Optimizer

- Consider a general optimizer

$$x_{t+1} = x_t - \eta G(x_t; \xi_t)$$

where ξ_t is a random variable independent of x_t.

- Let $G(x) = \mathbb{E}[G(x; \xi)]$. Let x^* be a fixed point, i.e.

$$G(x^*) = 0$$

- Over-parameterized assumption:

$$G(x^*; \xi) = 0$$

- Linearizing the dynamic gives us

$$x_{t+1} = x_t - \eta \nabla_x G(x^*; \xi_t)(x_t - x^*)$$

Theorem

Suppose x^* be the fixed point of interest, and let $A_\xi = \nabla_x G(x^*;\xi)$. x^* is linear stable if the following condition is satisfied,

$$|\lambda_{max}(I - \eta\mathbb{E}[A_\xi])| \leq 1$$

$$\lambda_{max}\left(\mathbb{E}_\xi[(I - \eta A_\xi)^T (I - \eta A_\xi)]\right) \leq 1$$

Proof.

- Let $B_\xi = 1 - \eta A_\xi$, then $x_n = \prod_{t=0}^{n} B_{\xi_t} x_t$
- $\mathbb{E}[x_n] = (\mathbb{E}[B_\xi])^n x_0$
- $$\begin{aligned}\mathbb{E}[\|x_n\|^2] &= \mathbb{E}[x_0^T B_{\xi_0}^T \cdots B_{\xi_n}^T B_{\xi_n} \cdots B_{\xi_0} x_0] \\ &\leq \mathbb{E}[x_0^T B_{\xi_0}^T \cdots B_{\xi_{n-1}}^T B_{\xi_{n-1}} \cdots B_{\xi_0} x_0] \\ &\leq x_0^T x_0\end{aligned}$$

□

Remark

- Roles of learning rate and batch size are different.

Remark

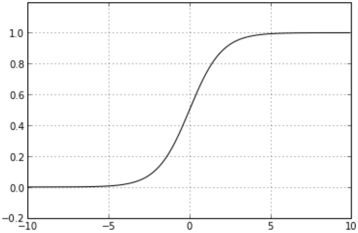
- Roles of learning rate and batch size are different.
  - Decreasing batch size forces SGD to choose solutions with smaller non-uniformity;

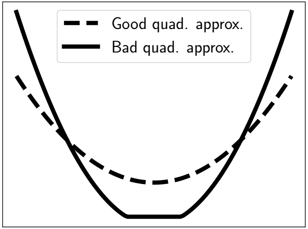
Remark

- Roles of learning rate and batch size are different.
  - Decreasing batch size forces SGD to choose solutions with smaller non-uniformity;
  - Increasing learning rate forces SGD to choose solutions with both smaller sharpness and smaller non-uniformity.

Remark

- Roles of learning rate and batch size are different.
  - Decreasing batch size forces SGD to choose solutions with smaller non-uniformity;
  - Increasing learning rate forces SGD to choose solutions with both smaller sharpness and smaller non-uniformity.
- Local quadratic approximation. $\nabla \ell(y, \bar{y}) = \ell'(y, \bar{y}) \frac{\partial y}{\partial \theta}$





Numerical Results.

- FashionMNIST and CIFAR10
- Quadratic loss
- Fixed learning rate

Learning rate is crucial for the sharpness of the selected minima

Table: Sharpness of the solutions found by GD with different learning rates. Each experiment is repeated for 5 times with independent random initialization.

η	0.01	0.05	0.1	0.5	1
FashionMNIST	53.5 ± 4.3	39.3 ± 0.5	19.6 ± 0.15	3.9 ± 0.0	1.9 ± 0.0
CIFAR10	198.9 ± 0.6	39.8 ± 0.2	19.8 ± 0.1	3.6 ± 0.4	-
prediction $2/\eta$	200	40	20	4	2

Influence of Batch Size

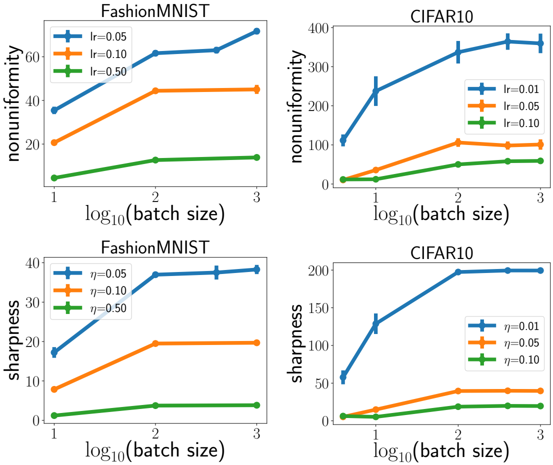


Figure: The influence of the batch size on the non-uniformity and sharpness.

The Selection Mechanism

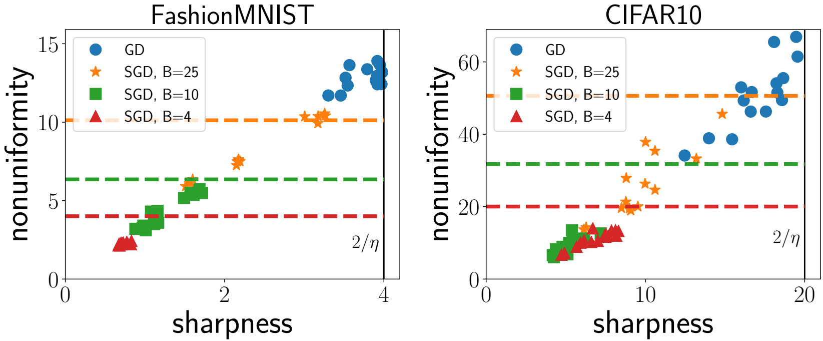


Figure: The sharpness-non-uniformity diagram for the minima selected by SGD.

The Selection Mechanism

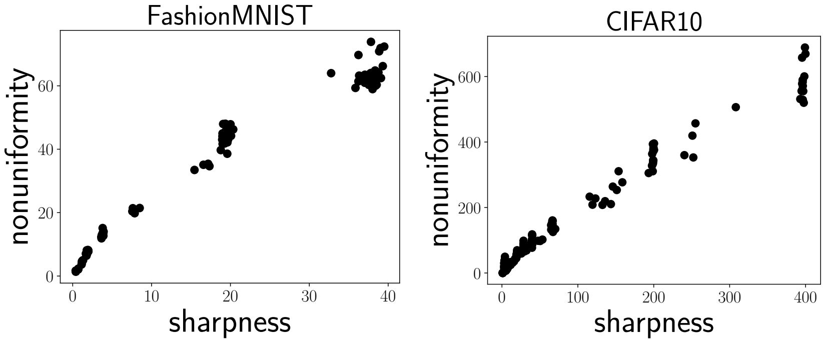


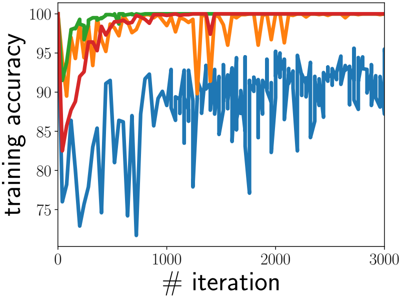
Figure: Scatter plot of sharpness and non-uniformity.

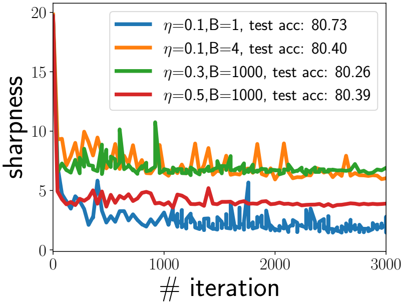
Back to the Escape Phenomenon

FashionMNIST

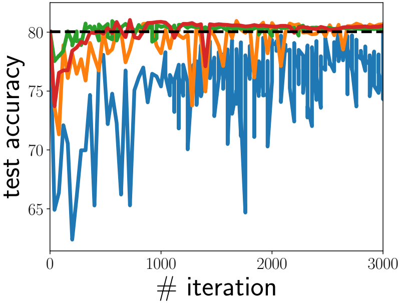
Table: Information for the initialization of the escape experiment.

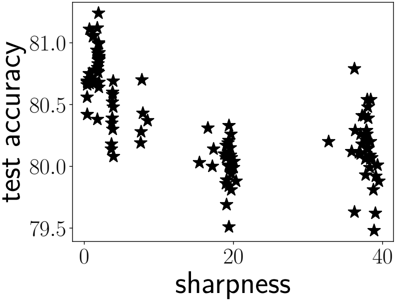
dataset	lr	test acc	sharpness	non-uniformity
FashionMNIST	0.1	80.04	19.7	45.2





FashionMNIST(cont'd)

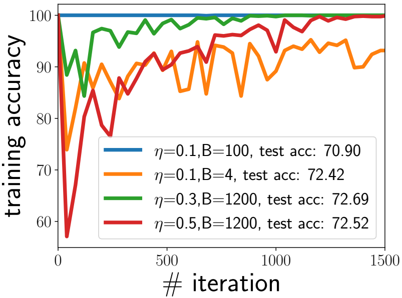


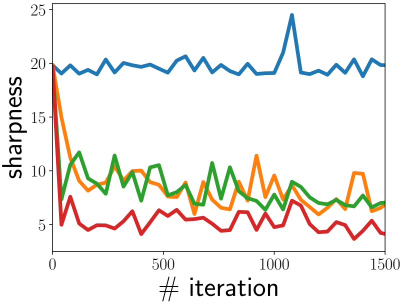


Corrupted FashionMNIST

Table: Information for the initialization of the escape experiment.

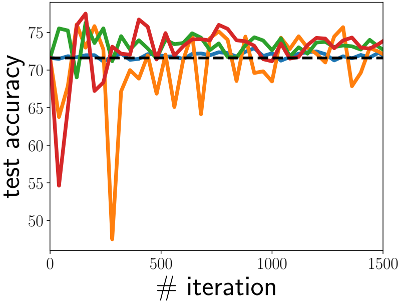
dataset	lr	test acc	sharpness	non-uniformity
Corrupted FashionMNIST	0.1	71.44	19.9	51.7

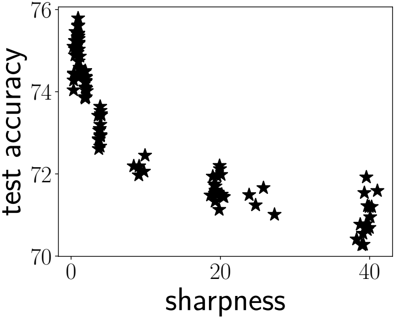




$$\sqrt{100}/0.1 = 100$$

Corrupted FashionMNIST(cont'd)





Extension: Quasi-Newton Method

Consider iteration:

$$x_{t+1} = x_t - \eta D^{-1} \nabla f(x_t),$$

whose stability condition is $\lambda_{max}(D^{-1}H) \leq 2/\eta$. For algorithms $D \approx H$, we have almost all the minima can be selected as long as $\eta \leq 2$.

Extension: Quasi-Newton Method

Consider iteration:

$$x_{t+1} = x_t - \eta D^{-1} \nabla f(x_t),$$

whose stability condition is $\lambda_{max}(D^{-1}H) \leq 2/\eta$. For algorithms $D \approx H$, we have almost all the minima can be selected as long as $\eta \leq 2$.

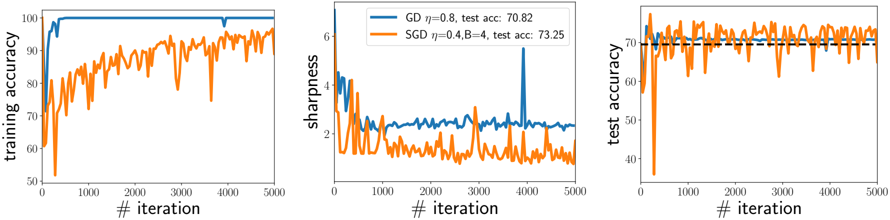


Figure: Escape of GD and SGD from the minima (test accuracy 69.5%) selected by well-tuned L-BFGS.

Extension: Quasi-Newton Method

Consider iteration:

$$x_{t+1} = x_t - \eta D^{-1} \nabla f(x_t),$$

whose stability condition is $\lambda_{max}(D^{-1}H) \leq 2/\eta$. For algorithms $D \approx H$, we have almost all the minima can be selected as long as $\eta \leq 2$.

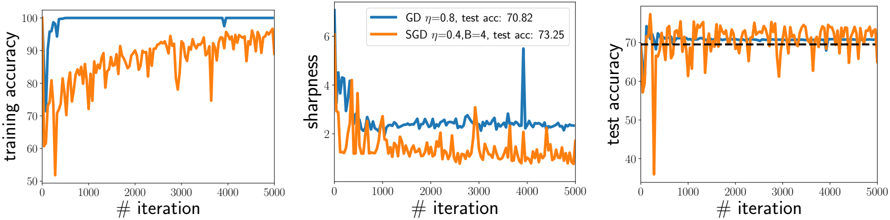


Figure: Escape of GD and SGD from the minima (test accuracy 69.5%) selected by well-tuned L-BFGS.

This might provide an explanation why adaptive gradient methods tend to pick up solutions generalizing worse (Wilson et al. [2017])

References I

- Lei Wu, Zhanxing Zhu, Weinan E. *Towards Understanding Generalization of Deep Learning: Perspective of Loss Landscape.* ICML 2017, Workshop on PADL
- Lei Wu, Chao Ma, and Weinan E . *How SGD selects the global minima in over-parameterized learning: A dynamical stability perspective.* NeurIPS 2018
- Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. *The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects.* ICML 2019
- Weinan E, Chao Ma, and Lei Wu . *A comparative analysis of the optimization and generalization property of two-layer neural network and random feature models under gradient descent dynamics.* arXiv preprint arXiv:1904.04326, 2019.

References II

- Weinan E, Chao Ma, and Lei Wu. *Machine learning from a continuous viewpoint.* arXiv preprint arXiv:1912.12777
- Chao Ma, Lei Wu and Weinan E. *The quenching-activation behavior of the gradient descent dynamics for two-layer neural network models.* Arxiv preprint arXiv:2006.14450
- Chao Ma, Lei Wu and Weinan E. *The slow deterioration of the generalization error of random feature models. MSML 2020*