On Implicit Bias of Optimization Algorithms: A New Era of Interpolation

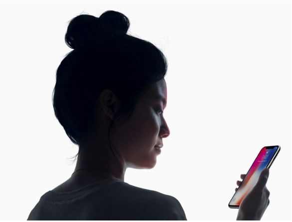
Yan Li

Georgia Tech

Background and Motivations

Successes of Deep Learning (DL)

- Face recognition
  - bio-authentification
  - security



- Natural language processing
  - machine translation
  - machine understanding
  - text generation

- Planning
  - autonomous driving

Deep Learning for Classification

Given n observations $\{x_i, y_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, ..., K\}$, we train a classifier $f(\cdot, \theta)$ by

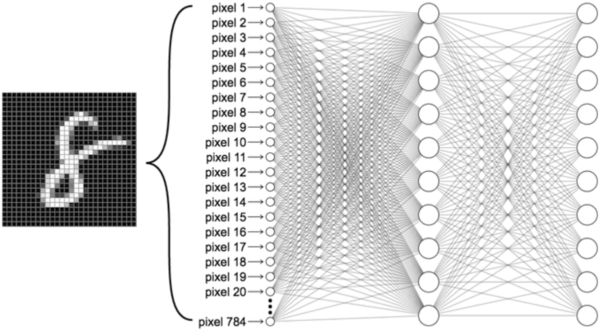
$$\widehat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i, \theta), y_i),$$

where $\ell(\cdot)$ is a proper loss function for classification.

Handwritten digits from the MNIST dataset.

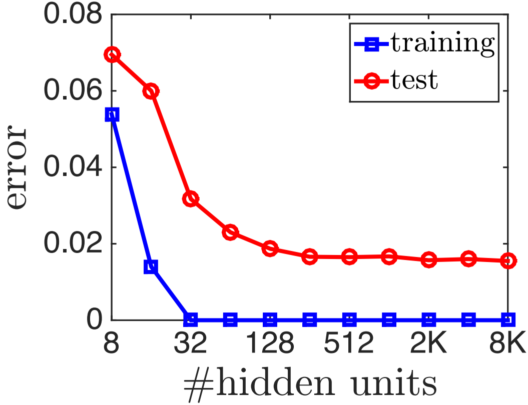
Neural Networks with One Hidden Layer

When there are more 64 neurons, the number of parameters in the neural network ($\geq 784 \times 64 + 64 \times 10 = 50,816$) is already larger than the training sample size $(50,000)$.



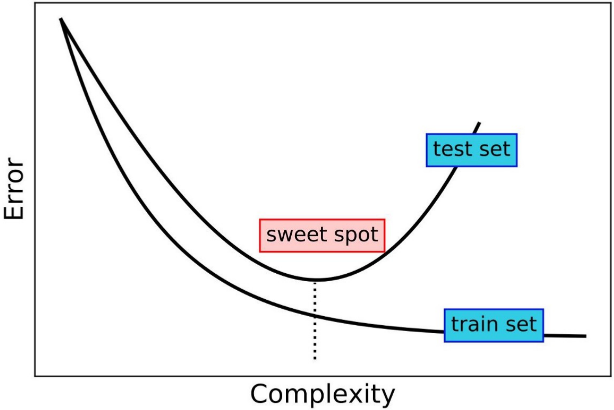
Benign Overfitting

Neural networks can interpolate data and achieve a zero training error, but can still generalize well.



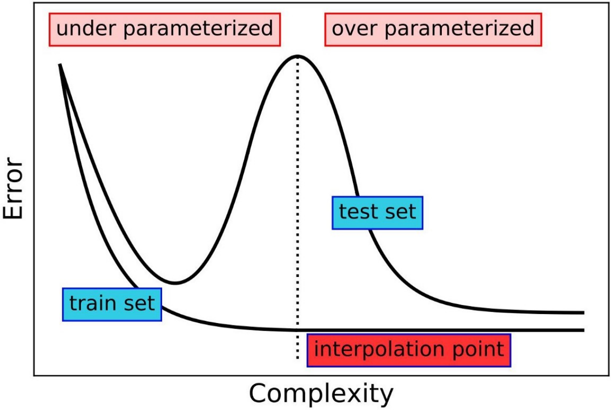
Conventional Wisdom

Bias-Variance Tradeoff: If we control the model complexity by regularization, we reduce the variance and increase the bias.



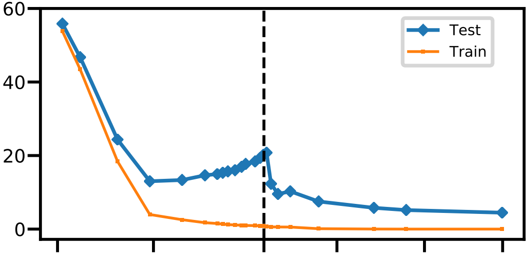
Double Descent and Overparameterization

When the number of parameters exceeds certain interpolation point, the testing error starts to decrease again!



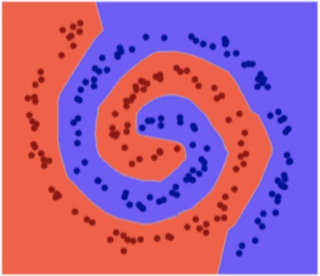
More Refined Experiments on MNIST

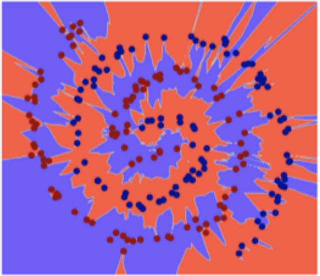
When we carefully train the neural networks for interpolation, double decent indeed happens.



What actually Happened?

Neural networks are highly flexible and are capable of perfectly classifying all the training data points, when there are sufficiently many parameters.





Training neural networks using (stochastic) gradient descent prefers a smooth decision boundary (Left) to a bumpy one (Right).

Common Belief: Implicit Bias

There exist many large neural network models, which can perfectly classify all the training data points.

Open Question: Which model will the algorithm pick?

Conjecture: The (stochastic) gradient descent algorithm tends to pick a low-complexity model. This is known as implicit bias/regularization of GD/SGD.

Reality: Large neural networks are usually trained by not only SGD but also many other tricks, which may also contribute to generalization.

Warm Up: Overaparameterized Linear Regression

Overparameterized Linear Regression (OLR)

Given n observations $\{x_i, y_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$ and $n < d$, we train a linear model $f(x, \theta) = x^\top \theta$ by

$$\widehat{\theta} = \arg\min_{\theta} \frac{1}{2n} \sum_{i=1}^{n} (y_i - x^\top \theta)^2 = \arg\min_{\theta} \underbrace{\frac{1}{2n} \|y - X\theta\|_2^2}_{\mathcal{L}(\theta)}.$$

There are infinitely many optima:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = X^\top X\theta - X^\top y = 0 \Rightarrow \widehat{\theta} = (X^\top X)^\dagger X^\top y + u,$$

where $A^\dagger$ is the pseudo-inverse of A, and u is any vector in $\mathbb{R}^d$ satisfying $X^\top u = 0$.

Gradient Descent for OLR

We initialize at θ_0. At the t-th iteration, Gradient Descent takes

$$\theta_t = \theta_{t-1} - \eta_t X^\top (X\theta_{t-1} - y),$$

where $\eta_t > 0$ is the step size.

Remark 1: The iterative increment always stays in the row space of X denoted by $\mathcal{X}$.

Remark 2: By decomposing

$$\theta_0 = \Pi_{\mathcal{X}}(\theta_0) + \Pi_{\mathcal{X}_\perp}(\theta_0),$$

we can easily show

$$\theta_t \to \Pi_{\mathcal{X}_\perp}(\theta_0) + (X^\top X)^\dagger X^\top y.$$

Gradient Descent for OLR

When we initialize at $\theta_0 = 0$, we can show

$$\theta_t \to (X^\top X)^\dagger X^\top y.$$

This is equivalent to finding

$$\widehat{\theta} = \arg\min_{\theta} \|\theta\|_2^2 \quad \text{subject to } \mathcal{L}(\theta) = 0.$$

Remark 3: Gradient Descent finds the minimum 2-norm model, which can interpolate n data points.

Overparameterized Linear Classification

Overparameterized Linear Classification (OLC)

Given n observations $\{x_i, y_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$ and $n < d$, we train a linear model $f(x, \theta) = x^\top \theta$ by

$$\widehat{\theta} = \arg\min_{\theta} \mathcal{L}(\theta), \quad \text{where} \quad \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i x^\top \theta),$$

and $\ell(\cdot)$ is a properly chosen light-tail loss function.

No finite minimizer: Assume that there are no identical x_i's. When $d > n$, the data are linearly separable: There exist infinitely many θ's such that

$$y_i x_i^\top \theta > 0 \quad \text{and} \quad \lim_{c \to \infty} \mathcal{L}(c\theta) = 0.$$

Overparameterized Linear Classification

The scale of θ does not matter in terms of the classification accuracy: Given a testing data point $\widetilde{x}$, we predict its label by

$$\widetilde{y} = \mathrm{sign}(\widetilde{x}^\top \theta).$$

Recall the support vector machine for linearly separable data.

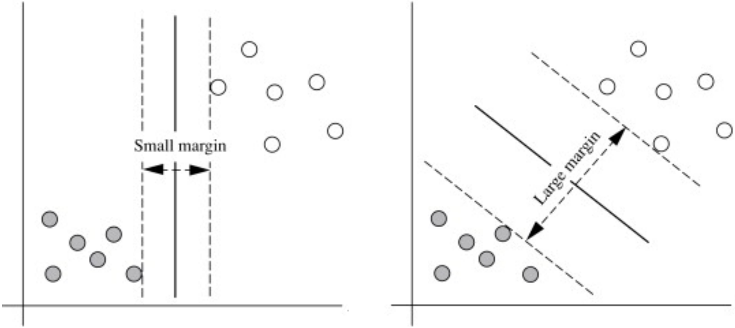
$$\widehat{\theta}_{\mathrm{SVM}} = \arg\min_{\theta} \ \|\theta\|_2^2$$
$$\text{subject to } y_i x_i^\top \theta \geq 1, \ i = 1, 2, ..., n.$$

This is equivalent to maximizing the normalized margin:

$$\max_{\theta} \min_{i} y_i \langle x_i, \theta / \|\theta\|_2 \rangle.$$

Maximum Margin Classifier

The maximum margin classifier (MMC, i.e., Support Vector Machine) is essentially the minimum 2-norm model, which can interpolate the data subject to the minimum margin value 1.



Gradient Descent for OLC

We initialize at θ_0. At the t-th iteration, Gradient Descent takes

$$\theta_t = \theta_{t-1} - \frac{\eta_t}{n} \sum_{i=1}^{n} \ell'(y_i x_i^\top \theta_{t-1}) y_i x_i,$$

where $\eta_t > 0$ is the step size.

Remark 1: Gradient descent will diverge to infinity, as there is no finite minimizer.

Remark 2: Most of existing optimization theories only consider the problems with finite minimizers.

Implicit Regularization of Gradient Descent

Assumptions and Notations

- x_i's are bounded:

$$\|x_i\|_2 \le 1, \quad i = 1, ..., n$$

- Light tail loss:

$$\ell(y_i x_i^\top \theta) = \exp(-y_i x_i^\top \theta)$$

- Maximum $\|\|_2$-norm margin:

$$\theta^* = \mathop{\arg\max}_{\|\theta\|_2 = 1} \min_i y_i x_i^\top \theta \quad \text{and} \quad \gamma = \min_i y_i x_i^\top \theta^*$$

- $\mathcal{X}$: The subspace spanned by x_i's

Convergence of Empirical Risk

Theorem (Ji et al. 2019)

Given $\eta \leq 1$ and $\theta_0 = 0$, we have

$$\mathcal{R}(\theta_T) - \inf_{\theta} \mathcal{R}(\theta) = \mathcal{O}\left(\frac{1}{T} + \frac{\log^2 T}{\gamma^2 T}\right)$$

Remarks:

- Allows the data to be not strictly separable (non-separable in a subspace).
- Acceleration is possible by using increasing stepsizes

Directional Convergence of Parameter

Theorem (Ji et al. 2019)

Given $\eta = 1$ *and* $\theta_0 = 0$, *we have*

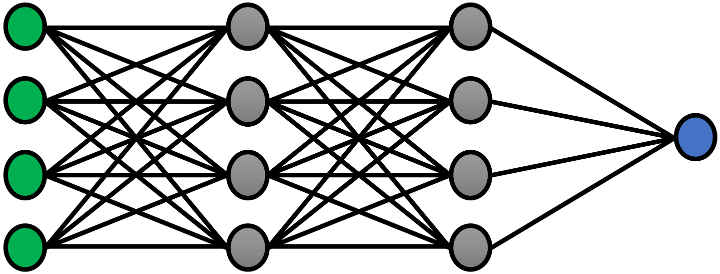
$$1 - \langle \theta^*, \theta_T / \|\theta_T\|_2 \rangle = \mathcal{O}\left(\frac{\log n + \log\log T}{\gamma^2 \log T} \right)$$

Remarks:

- Note the convergence is really slow!
- Acceleration is possible by using increasing stepsize (strictly separable data)

Fully Connected Linear Networks

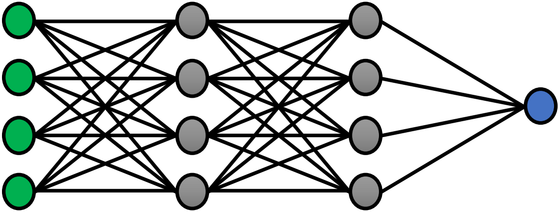
Fully Connected Linear Networks



$$f(x, \mathcal{W}) = w_L W_{L-1} \cdots W_1 x = x^\top \theta,$$

where $\theta = w_L W_{L-1} \cdots W_1$.

Fully Connected Linear Networks



Theorem (Gunasekar et al. 2018)

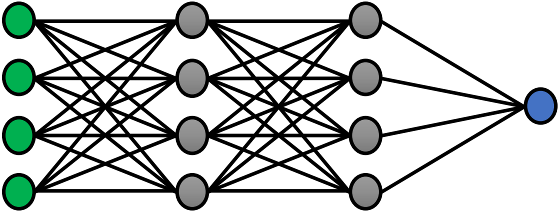
Under some regularity conditions, we have

$$\lim_{T\to\infty} \frac{\theta_T}{\|\theta_T\|_2} = \frac{\theta^*}{\|\theta^*\|_2},$$

where

$$\theta^* = \arg\min_{\theta} \|\theta\|_2^2 \quad \textit{subject to} \quad y_i x_i^\top \theta \geq 1,\ \forall\, i = 1, ..., n.$$

Fully Connected Linear Networks



Theorem (Ji et al. 2019)

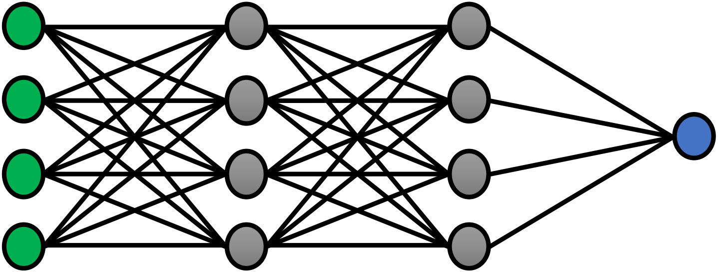
Under some regularity conditions, we have

$$\lim_{T\to\infty} \frac{\|W_T^j\|_2}{\|W_T^j\|_{\mathrm{F}}} = 1 \quad \textit{and} \quad \lim_{T\to\infty} \langle v_T, \theta^* \rangle = 1,$$

where v_T is the first right singular vector of W^1.

Homogenous Nonlinear Fully Connected Networks

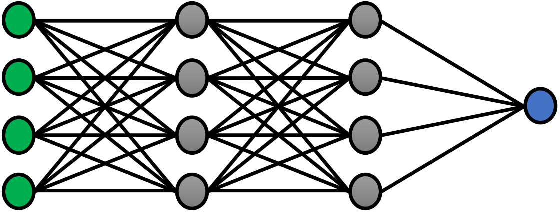
Homogenous Nonlinear Networks



$$f(x, \mathcal{W}) = w_L \sigma(W_{L-1} \cdots \sigma(W_1 x)),$$

where $\sigma(\cdot)$ is the homogenous activation satisfying $\sigma(tx) = t\sigma(x)$.

Homogenous Nonlinear Networks



Theorem (Lyu et al. 2018)

Under some regularity conditions, we have

$$\lim_{T\to\infty} \frac{\mathcal{W}_T}{\|\mathcal{W}_T\|_2} = \frac{\mathcal{W}^*}{\|\mathcal{W}^*\|_2},$$

where $\mathcal{W}^*$ is in the vector form and some KKT point to the following nonconvex optimization problem

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} \|\mathcal{W}\|_2^2 \quad \textit{subject to} \quad y_i f(x_i, \mathcal{W}) \geq 1,\ \forall\ i = 1, ..., n.$$

Data-dependent Algorithmic Regularization

Motivation: Bregman Proximal Point in Applications

An emerging set of algorithms: self-training, self-distillation

A lot of them can be described by Bregman proximal point algorithm

$$\theta_{t+1} = \arg\min_\theta L(\theta) + 1/(2\eta_t)\mathcal{D}(\theta, \theta_t).$$

Popular choices of divergence function

- $D_{\mathrm{LS}}(\theta, \theta_t) = \mathbb{E}_{\mathcal{D}} \left\| f_\theta(x) - f_{\theta_t}(x) \right\|_2^2$
- $D_{\mathrm{KL}}(\theta, \theta_t) = \mathbb{E}_{\mathcal{D}} \mathrm{KL}\left(f_{\theta'}(x) \| f_\theta(x) \right)$

State-of-art performance in language models and image classification models.

The role of divergence?

Question: How does $\mathcal{D}$ affect the learned model?

Key observation: divergence $\mathcal{D}$ can be data-dependent.

Approach: connecting algorithmic regularization with (potentially data-dependent) divergence $\mathcal{D}$.

Linear Separable Classification

Task: Learn a linear classifier for linearly separable data

Learning Objective:

$$\min_{\theta} L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i x_i^\top \theta)$$

Algorithm (BPPA):

$$\theta_{t+1} = \arg\min_{\theta} L(\theta) + 1/(2\eta_t) D_w(\theta, \theta_t).$$

Remark:

- Use tight exponential tail loss (exp/logistic).
- No finite minimizer.
- Our later results holds also for mirror descent.

Minimal Assumption

Only assumption: The distance generating function of Bregman divergence $D_w(\cdot, \cdot)$ is L_w-smooth and μ_w-strongly convex w.r.t. $\|\cdot\|$-norm.

$$\frac{\mu_w}{2} \left\|\theta - \theta'\right\|^2 \leq \underbrace{w(\theta) - w(\theta') - \left\langle \nabla w(\theta'), \theta - \theta' \right\rangle}_{D_w(\theta, \theta')} \leq \frac{L_w}{2} \left\|\theta - \theta'\right\|^2.$$

Remark

- D_w can be data-dependent, so does the $\|\cdot\|$.
- Will provide a concrete example.

Implicit Regularization of BPPA

Good Conditionedness = Good Seperation:

$$\lim_{t\to\infty} \min_{i\in[n]} \left\langle \frac{\theta_t}{\|\theta_t\|}, y_i x_i \right\rangle \geq \sqrt{\frac{\mu_w}{L_w}} \gamma_{\|\cdot\|_*},$$

Interpretation:

$$u_{\|\cdot\|_*} = \arg\max_{\|u\|\leq 1} \min_{i\in[n]} \langle u, y_i x_i \rangle, \quad \gamma_{\|\cdot\|_*} = \max_{\|u\|\leq 1} \min_{i\in[n]} \langle u, y_i x_i \rangle.$$

Remarks:

- Lower bound is tight for a class of problems
- Works for general norm $\|\cdot\|$ instead of ℓ_2 norm
- Non-asymptotic convergence is still slow $\mathcal{O}(1/\log t)$
- Can be accelerated to $\mathcal{O}(1/\sqrt{t})$ using increasing stepsizes

Data-dependent implicit regularization in action

Mixture of Sphere: $y_i \sim \text{Bernoulli}(1/2),\ x_i \sim \text{Unif}\left(\mathbb{S}_{y_i\mu}(r)\right)$, where $\mathbb{S}_z(r)$ denotes the sphere centered at z with radius r in $\mathbb{R}^d$.

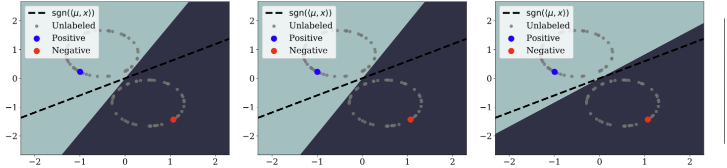
Limited labeled, Abundant unlabeled data: n labeled data $\{(x_i, y_i)\}_{i=1}^n$. m unlabeled data $\{\widetilde{x}_j\}_{j=1}^m$.

Three divergences:

- Data-independent: $D^{(1)}(\theta, \theta') = \|\theta - \theta'\|_2^2$
- Data-dependent: $D^{(2)}(\theta, \theta') = (\theta - \theta')^\top \widehat{\Sigma}(\theta - \theta')$
- Data-dependent: $D^{(3)}(\theta, \theta') = (\theta - \theta')^\top \widehat{\Sigma}^{-1}(\theta - \theta')$
- $\widehat{\Sigma} = \frac{1}{m}\sum_{j=1}^m \widetilde{x}_j \widetilde{x}_j^\top$

Data-dependent implicit regularization in action

Visualization of Decision Boundary



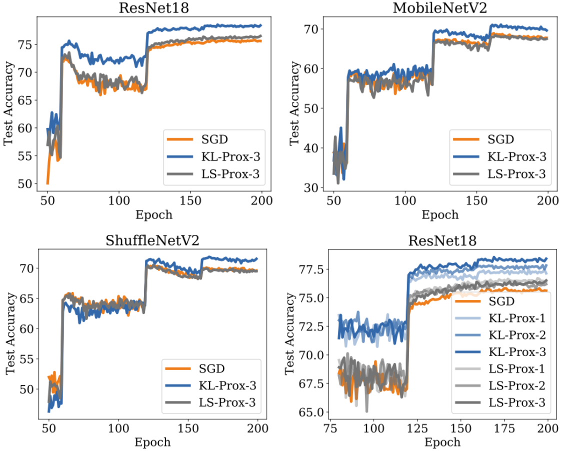
Left ($D^{(1)}$), Middle ($D^{(2)}$), Right ($D^{(3)}$)

Explanation

- All three divergence have conditioned number 1, but with different norm.
- Leads to maximal margin solutions wrt different norms.
- $D^{(3)}$ gives the best norm.

Data-dependent implicit regularization in action

BPPA on Neural Networks: ResNet-18, MobileNetV2, ShuffleNetV2 on CIFAR-100 dataset.

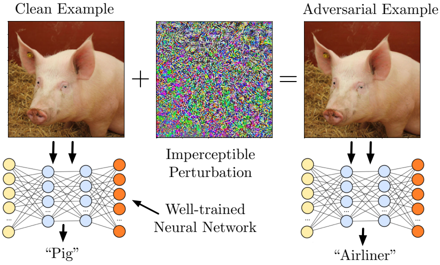


Divergence matters!

Understanding Adversarial Training via Algorithmic Regularization

Blindspots of DL: Adversarial Examples

Deep neural networks are vulnerable to adversarial examples (Goodfellow et al. 2014).



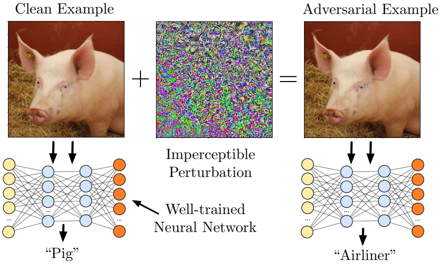
Given model $f(\cdot, \theta)$, loss function $\ell(\cdot, \cdot)$, data point (x, y), a (specified) perturbation set $\mathcal{B}$.

$$\widehat{x} = \underset{x' \in \{x\} \bigoplus \mathcal{B}}{\arg\max} \ \ell\left(f(x', \theta), y\right).$$

$\bigoplus$ denotes direct sum.

Blindspots of DL: Adversarial Examples

Deep neural networks are vulnerable to adversarial examples (Goodfellow et al. 2014).



Given model $f(\cdot, \theta)$, loss function $\ell(\cdot, \cdot)$, data point (x, y), a (specified) perturbation set $\mathcal{B}$.

$$\widehat{x} = \underset{x' \in \{x\} \bigoplus \mathcal{B}}{\arg\max} \; \ell\left(f(x', \theta), y\right).$$

$\bigoplus$ denotes direct sum.

Defend Against Adversarial Examples

Provable Defense:

- Discrete optimization: (Tjeng et al. 2017).
  – heavy computation, no scalable
- Randomized smoothing: (Cohen et al. 2019).
  – scalable to ImageNet level dataset.
  – hard to defend against ℓ_∞ attack: $\mathcal{B} = \{\delta : \|\delta\|_\infty \le \epsilon\}$.

Summary: Limited practical performance, assumes adversary has infinite computational power (reasonable?).

Adversarial Training (AT), (Madry et al 2017)

$$\min_{\theta} \mathcal{L}_{\text{adv}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \max_{\delta_i \in \mathcal{B}} \ell(f(\underbrace{x_i + \delta_i}_{\text{Adversarial Example:}\widehat{x}_i}; \theta), y_i).$$

- Robust optimization (Ben-Tal; Nemirovski, 1998)
  – non-convex max, non-concave min, no-convergence guarantees
  – solving min problem (approximately) with projected gradient descent (common practice)
- Great empirical performance. Building block for most defense methods. Matches state-of-art algorithm with early-stopping (Rice et al. 2020)
- Adaptive robustness: defending against stronger attack $\rightarrow$ more robust model (Gao et al. 2019).
  – gradient descent based adversary (GDAT).
- Lack of theoretical guarantees.

Outline

Address the following question, (tentatively)

- **Q**: Does AT really achieves robustness?

  A: Understand AT through its algorithmic implicit bias.

Road to robustness

Previous results: SVM can be viewed as AT (Xu et al. 2009).
The following are equivalent

$$\min_{w,b} \max_{\sum_{i=1}^n \|\delta_i\|_* \le c} \sum_{i=1}^n \max\left[1 - y_i(w^\top(x_i + \delta) + b), 0\right] \quad \text{(AT)}$$

$$\min_{w,b} c\|w\| + \sum_{i=1}^n \max\left[1 - y_i(w^\top(x_i + \delta) + b), 0\right] \quad \text{(SVM)}$$

- robustness and regularization:
- non-separable data, finite minimizer
- What about separable data? – infinitely many perfect-accuracy classifiers!

Question: does the optimization algorithm have any preference among infinitely solutions for under-determined system? – implicit bias

Example: Solving under-determined least square:

$$\min_{x \in \mathbb{R}^d} \|Ax - b\|_2^2, \quad A \in \mathbb{R}^{n \times d}, d \gg n, b \in \mathbb{R}^n$$

with gradient descent, initialized at $x_0 = 0$, converges to minimum ℓ_2 norm solution.

– quick proof: $x_t \in \operatorname{span}(A^\top)$ with GD update, let $x^* = \lim_{t\to\infty} x_t = A^\top c_*$, we have $AA^\top c_* = b$, then $x^* = A^\top c_* = A^\top (AA^\top)^{-1} b$, the minimum ℓ_2 norm solution.

Training linear classifier with AT

Problem setup: linearly separable data $(x_i, y_i)_{i=1}^n$, tight exponential tail loss (e.g., logistic/exp loss), ℓ_q perturbation: $\mathcal{B} = \{\delta : \|\delta\|_q \leq c\}$.

Learning robust linear classifier:

$$\min_\theta \mathcal{L}_{\text{adv}}(\theta) = \frac{1}{n} \sum_{i=1}^n \max_{\|\delta\|_q \leq c} \ell(y_i(x_i + \delta)^\top \theta).$$

Observations:

- When $c < \gamma_q$, no finite solution, $\|\theta_t\| \to \infty$!
- When $c = 0$, standard training, converges in direction to ℓ_2 SVM (Soudry et al. 2018).

Training linear classifier with AT

AT on Separable Data with ℓ_q Perturbation

Input: Data points $\{(x_i, y_i)\}_{i=1}^n$, perturbation level $c < \gamma_q$ and step sizes $\{\eta^t\}_{t=0}^{T-1}$.

Init: Set $\theta^0 = 0$.

For $t = 0 \ldots T-1$:

For $i = 1 \ldots n$, $\widehat{\delta}_i = \arg\max_{\|\delta_i\|_q \le c} \ell(y_i(x_i + \delta_i)^\top \theta^t)$.

Set $\widetilde{x}_i = x_i + \widehat{\delta}_i$, for $i = 1 \ldots n$.

Update $\theta^{t+1} = \theta^t - (\eta^t/n) \cdot \sum_{i=1}^n \nabla \ell(y_i \widetilde{x}_i \theta^t)$.

Questions: Does AT possess implicit bias, and whether it relates to robustness?

A Robust SVM

Standard ℓ_q-norm SVM.

- θ_2 (and generally θ_q) = the optimal $\ell_2(\ell_q)$ margin SVM,

$$\theta_q = \underset{\|\theta\|_p=1}{\arg\max} \min_{i=1,\dots,n} y_i x_i^\top \theta, \quad 1/p + 1/q = 1, p, q \in [1, \infty].$$

- γ_q = optimal ℓ_q margin (max of RHS).

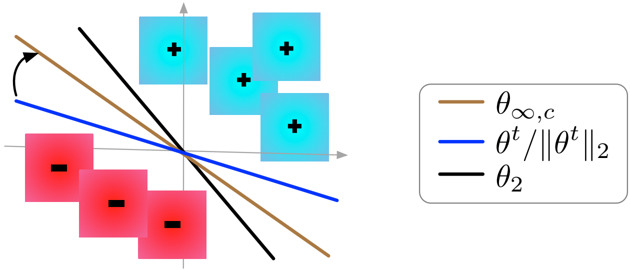
Robust SVM: adapts to adversary geometry $\mathcal{B} = \{\delta : \|\delta\|_q \le c\}$

$$\theta_{q,c} = \underset{\|\theta\|_2=1}{\arg\max} \min_{i=1,\dots,n} \min_{\|\delta_i\|_q \le c} y_i (x_i + \delta_i)^\top \theta.$$

Robustness: $\theta_{q,c}$ is in the same direction to the solution of

$$\min_{\theta \in \mathbb{R}^d} \|\theta\|_2 \quad \text{s.t. } y_i \widetilde{x}_i^\top \theta \ge 1 \text{ for all } \|\widetilde{x}_i - x_i\|_q \le c, \forall i = 1 \dots n.$$

A Robust SVM



Minimum mix-norm solution: $\theta_{q,c}$ is in the same direction to the solution of (here $1/p + 1/q = 1$)

$$\min_{\theta \in \mathbb{R}^d} \|\theta\|_2 + \eta(c)\|\theta\|_p \quad \text{s.t. } y_i x_i^\top \theta \geq 1, \forall i = 1 \ldots n.$$

GDAT Adapts to Adversary Examples

Theorem (Li et al. 2019)

Let perturbation level $c < \gamma_q$, Then

$$1 - \left\langle \theta^t / \left\| \theta^t \right\|_2, \theta_{q,c} \right\rangle = \mathcal{O}(\log n / \log t).$$

Remarks:

- Guaranteed robustness against ℓ_q perturbation bounded by c.
- Adaptive implicit bias. Converges to the most ℓ_2 robust linear classifier with ℓ_q margin at least c.
  Special case: $q = 2$, converge to ℓ_2 SVM.
- Complementary of well known results on non-separable data:
  SVM + AT $\Rightarrow$ Robust SVM (Xu el al. 2009).

AT Accelerates Convergence ($q = 2$)

Theorem (Li et al. 2019)

Let c and number of iterations T satisfy $\gamma_2 - c = \mathcal{O}\left(\frac{\log^2 T}{T}\right)^{1/2}$,
We have $\theta_{2,c} = \theta_2$, and

$$1 - \left\langle \theta^T / \left\|\theta^T\right\|_2, \theta_2 \right\rangle = \mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right).$$

Exponential Acceleration by AT!

Corollary: Convergence on clean loss by AT is almost exponentially faster than GD.

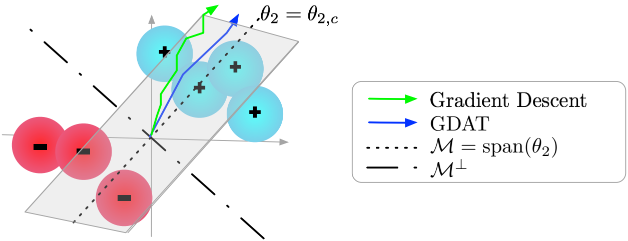
- AT: $\mathcal{L}_{\text{clean}}(\theta_T) = \mathcal{O}\left(\exp(-\sqrt{T}/\log T)\right)$
- GD: $\mathcal{L}_{\text{clean}}(\theta_T) = \mathcal{O}\left(1/T\right)$

Intuition: We have $\theta_{2,c} = \theta_2$: implicit bias of GD coincides with explicit bias of AT!

Key Technical Ingredients:

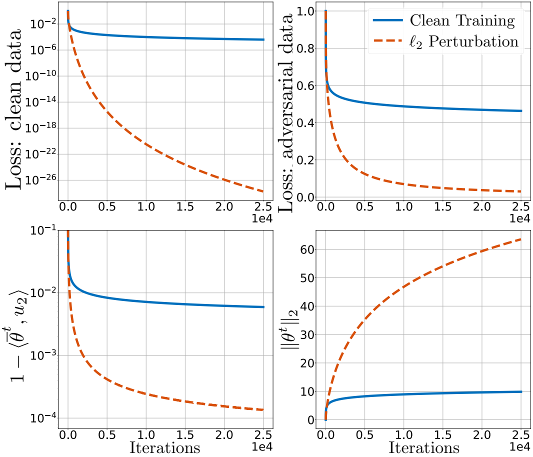
- Projection of θ^t onto the orthogonal space $\mathcal{M}^\perp = \{\theta : \langle \theta, \theta_2 \rangle = 0\}$ is bounded for all $t \geq 0$.
- For projection of θ^t onto the space $\mathcal{M} = \mathrm{span}(\theta_2)$, its increment satisfies **Generalized Perceptron Lemma**:

$$\left\langle \theta^{t+1} - \theta^t, \theta_2 \right\rangle \geq \eta \mathcal{L}_{\mathrm{adv}}(\theta^t)(\gamma_2 - c).$$



Empirical Study

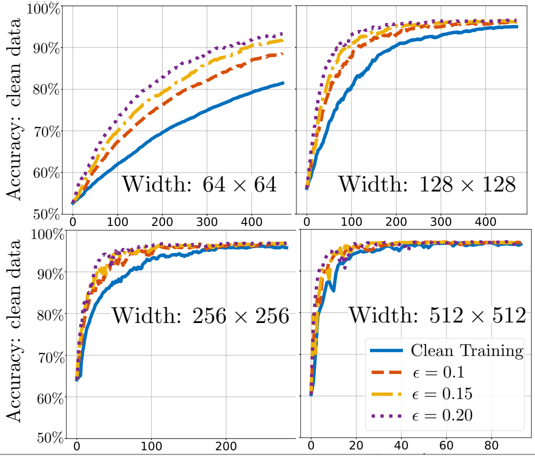
Linear Classifiers: We generate data with $\gamma_2 = 1$. We set $c = 0.95$. $\eta = 0.1$ for GDAT and $\eta = 1$ for standard training.



Clean Training v.s. GDAT (ℓ_2 perturbation)

Empirical Study

Neural Networks: We use MNIST dataset. The width of hidden layer varies in $\{64 \times 64, 128 \times 128, 256 \times 256, 512 \times 512\}$. We use ℓ_∞ perturbation with perturbation level $\epsilon \in \{0.1, 0.15, 0.20\}$.



Additional experimental results: (Xie et al. 2019) show acceleration effect of AT with practical deep networks.

Summary

- AT adapts the classifier to adversary geometry – provably for linear classifier.
- AT with ℓ_2 perturbation provides exponential speed-up on clean loss.

Deep non-linear networks is hard due to notorious non-convexity (future).

References

[1] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples."
[2] Madry, Aleksander, et al. "Towards deep learning models resistant to adversarial attacks."
[3] Rice, Leslie, Eric Wong, and J. Zico Kolter. "Overfitting in adversarially robust deep learning."
[4] Gao, Ruiqi, et al. "Convergence of Adversarial Training in Overparametrized Neural Networks."
[5] Tjeng, Vincent, Kai Xiao, and Russ Tedrake. "Evaluating robustness of neural networks with mixed integer programming."
[6] Cohen, Jeremy M., Elan Rosenfeld, and J. Zico Kolter. "Certified adversarial robustness via randomized smoothing."
[7] Montasser, Omar, Steve Hanneke, and Nathan Srebro. "VC classes are adversarially robustly learnable, but only improperly."
[8] Yin, Dong, Kannan Ramchandran, and Peter Bartlett. "Rademacher complexity for adversarially robust generalization."
[9] Khim, Justin, and Po-Ling Loh. "Adversarial risk bounds via function transformation."

References

[10] Soudry, Daniel, et al. "The implicit bias of gradient descent on separable data."
[11] Gunasekar, Suriya, et al. "Characterizing implicit bias in terms of optimization geometry."
[12] Ji, Ziwei, and Matus Telgarsky. "Gradient descent aligns the layers of deep linear networks."
[13] Gunasekar, Suriya, et al. "Implicit bias of gradient descent on linear convolutional networks."
[14] Xu, Huan, Constantine Caramanis, and Shie Mannor. "Robustness and regularization of support vector machines."
[15] Xie, Cihang, et al. "Adversarial Examples Improve Image Recognition."
[16] Schulman, John, et al. "Trust region policy optimization."
[17] Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning."

Thank You!