Generalization Ability

Over-parameterized models generalize well without overfitting by maximizing margins

Generalization Error

- Consider the empirical risk minimization under i.i.d. samples

$$\hat{R}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i;\theta)) + \mathcal{R}(\theta)$$

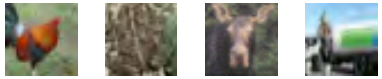
- The population risk with respect to unknown distribution

$$R(\theta) = \mathbf{E}_{x,y\sim P}\ell(y, f(x;\theta))$$

- Fundamental Theorem of Machine Learning (for 0-1 misclassification loss, called 'errors' below)

$$R(\theta) = \underbrace{\hat{R}_n(\theta)}_{\text{training loss/error}} + \underbrace{R(\theta) - \hat{R}_n(\theta)}_{\text{generalization loss/error}}$$

Why big models generalize well?



CIFAR10

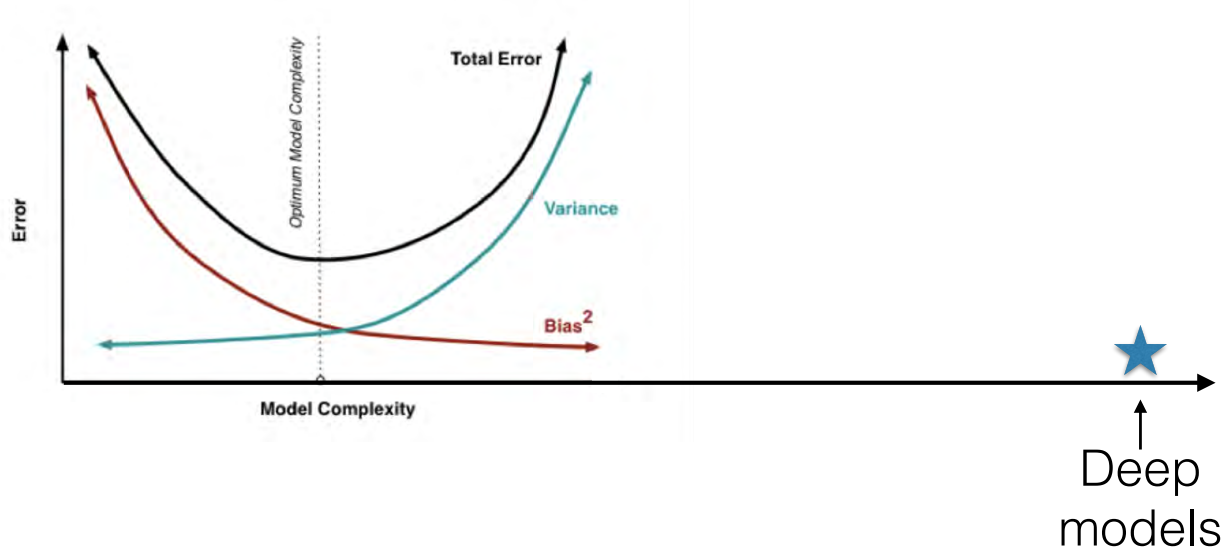
n=50,000
d=3,072
k=10

What happens when I turn off the regularizers?

Model	parameters	p/n	Train ***loss***	Test ***error***
CudaConvNet	145,578	2.9	0	23%
CudaConvNet (with regularization)	145,578	2.9	0.34	18%
MicroInception	1,649,402	33	0	14%
ResNet	2,401,440	48	0	13%

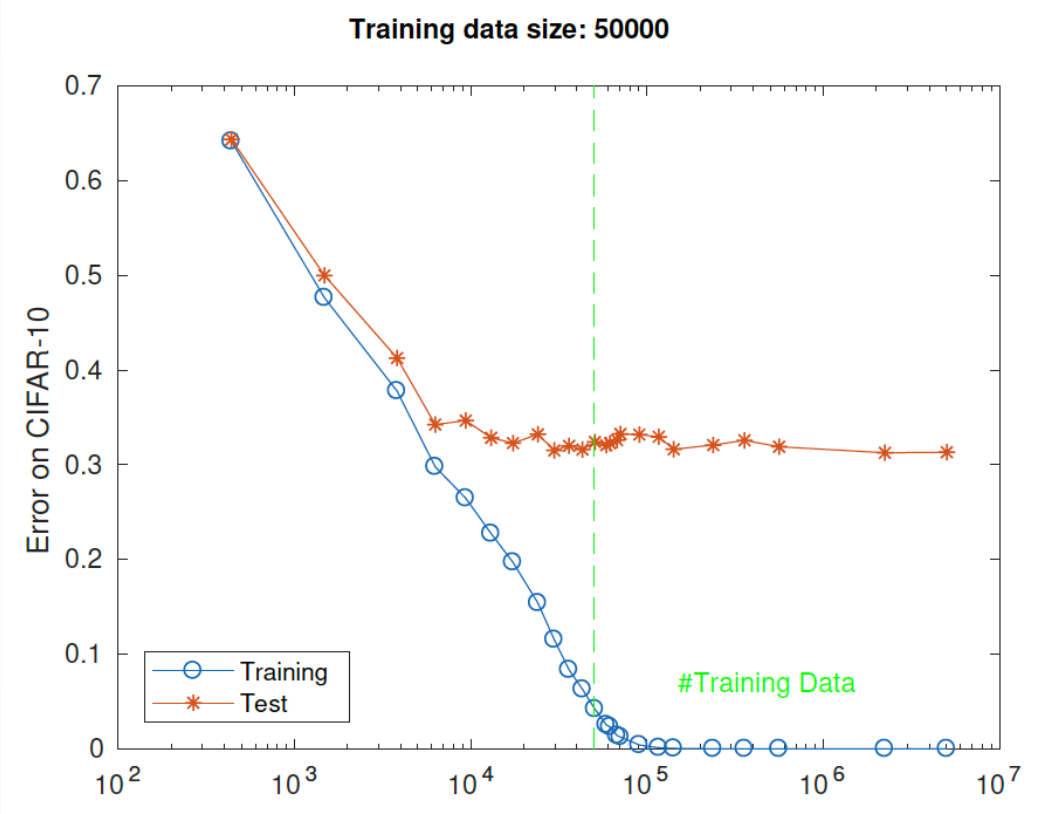
Ben Recht et al. 2016

The Bias-Variance Tradeoff?



Models where p>20n are common

Over-parameterized models

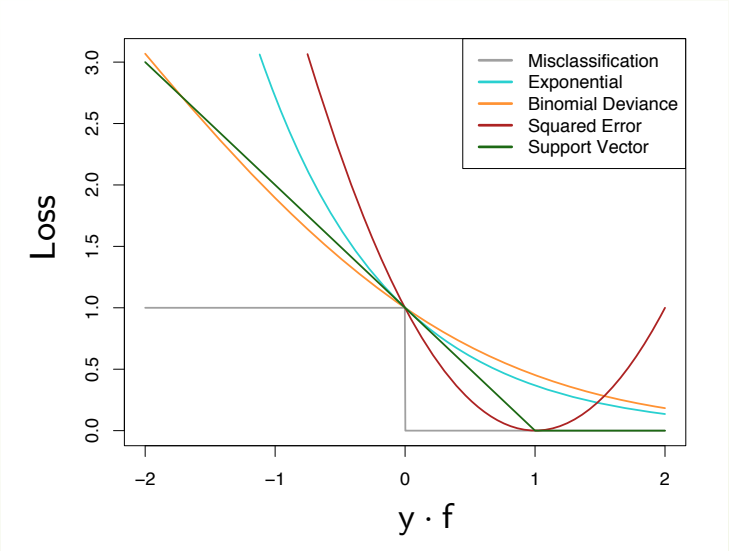


As model complexity grows (p>n), training error goes down to zero, but test error does not increase. Why overparameterized models do not overfit here? -- Tommy Poggio, 2018

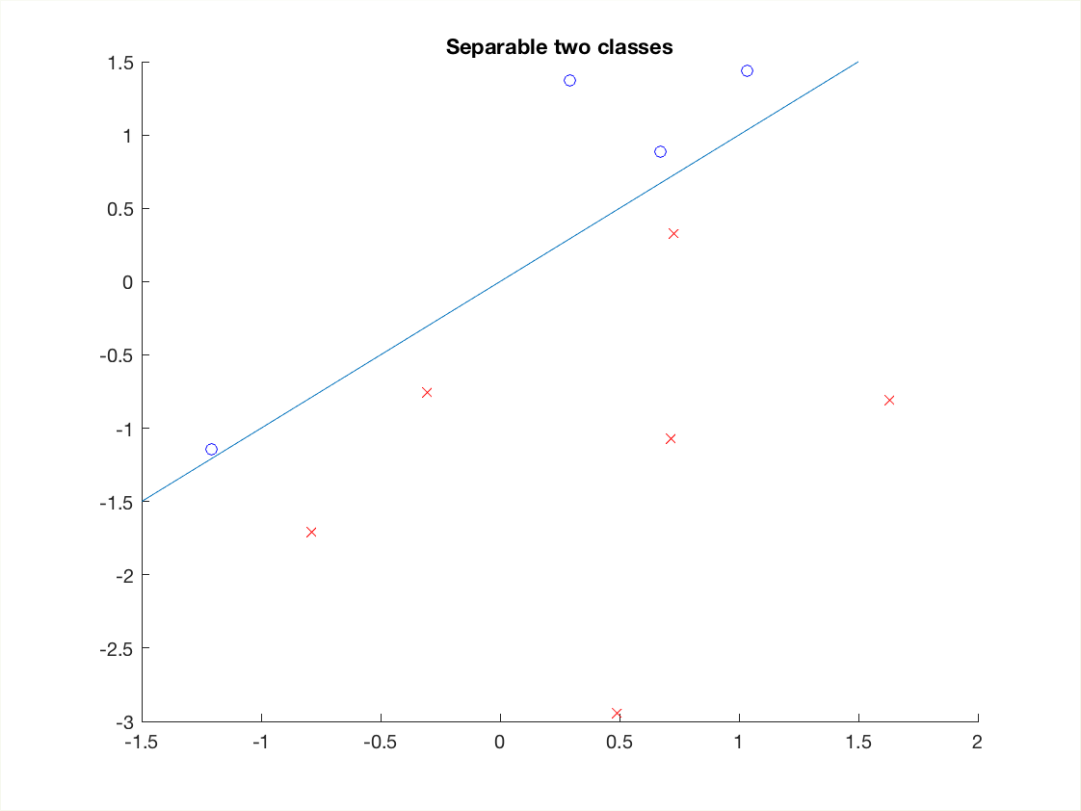
Binary Classification Problem

Consider a dataset $\{\mathbf{x}_n, y_n\}_{n=1}^N$, with $\mathbf{x}_n \in \mathbb{R}^d$ and binary labels $y_n \in \{-1, 1\}$. We analyze learning by minimizing an empirical loss of the form

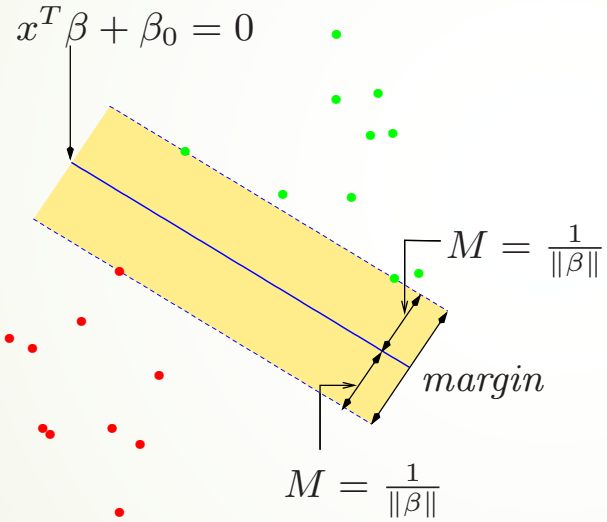
$$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N} \ell\left(y_n \mathbf{w}^\top \mathbf{x}_n\right) . \tag{1}$$



Separable Classification



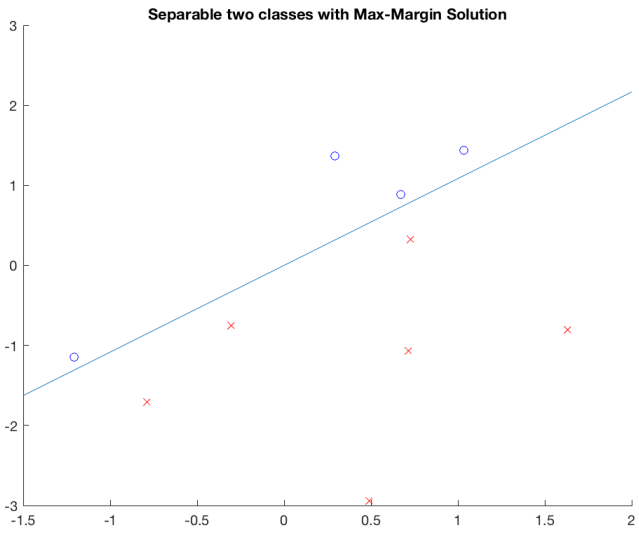
Max-Margin Classifier (SVM)



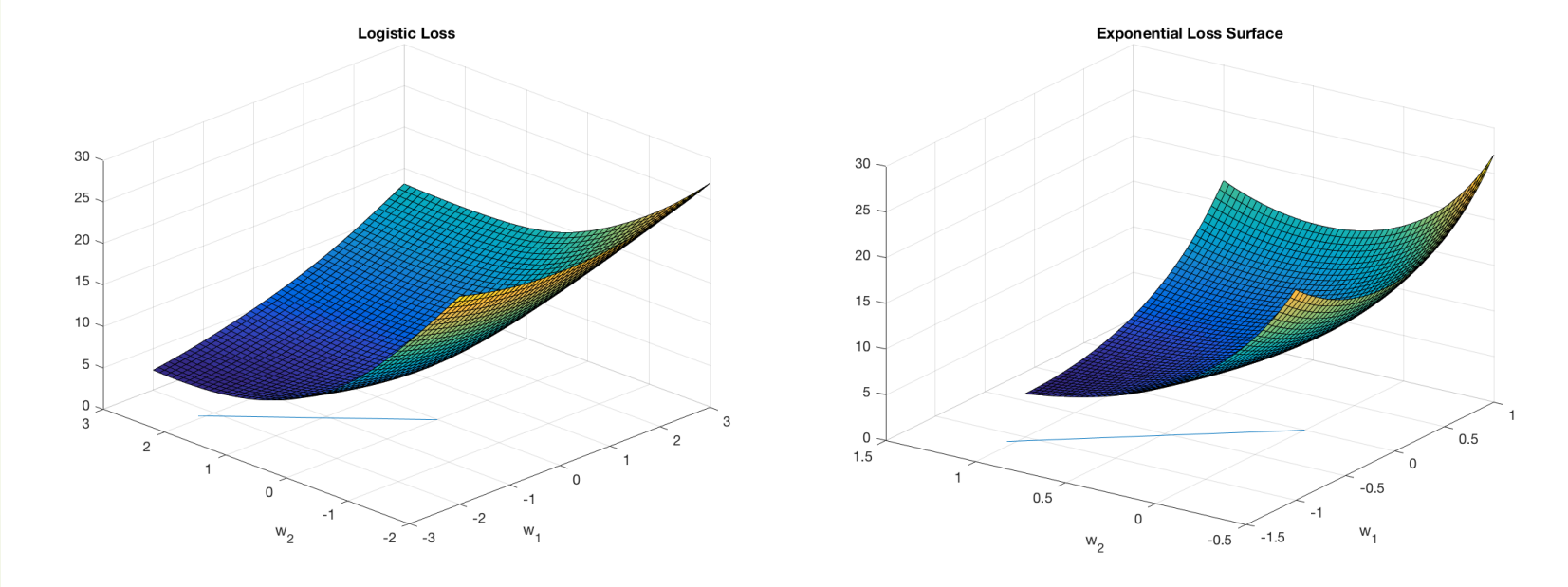
Vladmir Vapnik, 1994

$$\text{minimize}_{\beta_0,\beta_1,\ldots,\beta_p} \|\beta\|^2 := \sum_j \beta_j^2$$

$$\text{subject to } y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq 1 \text{ for all } i$$



Landscape of Logistic/Exponential Loss



The minimizers are at infinity, asymptotically in the direction of max-margin classifier

[Soudry, Hoffer, Nacson, Gunasekar, Srebro, 2017]

Theorem 3 *For any dataset which is linearly separable (Assumption 1), any β-smooth decreasing loss function (Assumption 2) with an exponential tail (Assumption 3), any stepsize $\eta < 2\beta^{-1}\sigma_{\max}^{-2}(\mathbf{X})$ and any starting point $\mathbf{w}(0)$, the gradient descent iterates (as in eq. 2) will behave as:*

$$\mathbf{w}(t) = \hat{\mathbf{w}} \log t + \boldsymbol{\rho}(t) , \tag{3}$$

where $\hat{\mathbf{w}}$ is the L_2 max margin vector (the solution to the hard margin SVM):

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}} \|\mathbf{w}\|^2 \ \text{s.t.}\ \mathbf{w}^\top \mathbf{x}_n \geq 1, \tag{4}$$

and the residual grows at most as $\|\boldsymbol{\rho}(t)\| = O(\log\log(t))$, and so

$$\lim_{t\to\infty} \frac{\mathbf{w}(t)}{\|\mathbf{w}(t)\|} = \frac{\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|}.$$

Furthermore, for almost all data sets (all except measure zero), the residual $\rho(t)$ is bounded.

Assumptions on General Loss Functions

Assumption 1 *The dataset is linearly separable:* $\exists \mathbf{w}_*$ *such that* $\forall n: \mathbf{w}_*^\top \mathbf{x}_n > 0$.

Assumption 2 $\ell(u)$ *is a positive, differentiable, monotonically decreasing to zero*[1]*, (so* $\forall u: \ell(u) > 0, \ell'(u) < 0$ *and* $\lim_{u\to\infty} \ell(u) = \lim_{u\to\infty} \ell'(u) = 0$*) and a* β*-smooth function,* i.e. *its derivative is* β*-Lipshitz.*

Assumption 2 includes many common loss functions, including the logistic, exp-loss[2], probit and sigmoidal losses. Assumption 2 implies that $\mathcal{L}(\mathbf{w})$ is a $\beta\sigma_{\max}^2(\mathbf{X})$-smooth function, where $\sigma_{\max}(\mathbf{X})$ is the maximal singular value of the data matrix $\mathbf{X} \in \mathbb{R}^{d\times N}$.

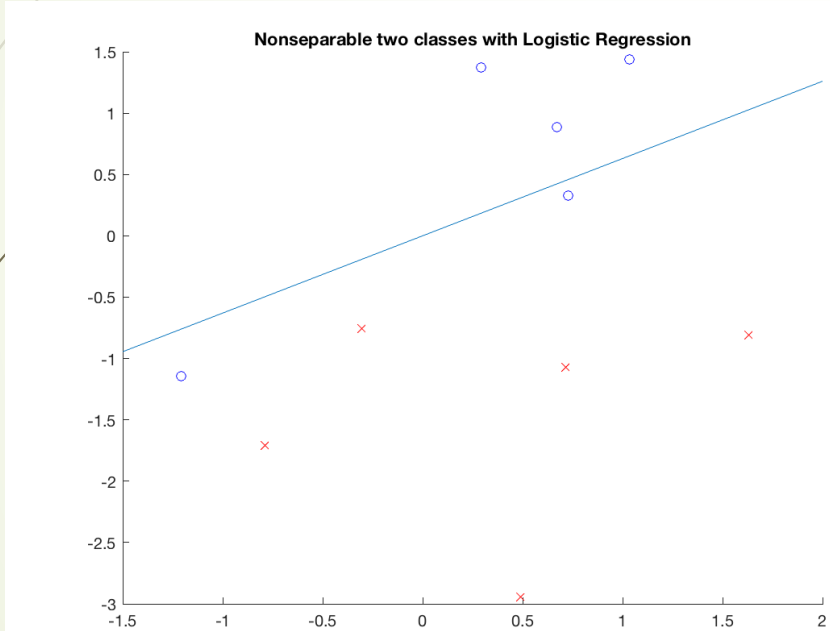
Definition 2 *A function* $f(u)$ *has a "tight exponential tail", if there exist positive constants* c, a, μ_+, μ_-, u_+ *and* u_- *such that*

$$\forall u > u_+ : f(u) \le c\left(1 + \exp\left(-\mu_+ u\right)\right) e^{-au}$$
$$\forall u > u_- : f(u) \ge c\left(1 - \exp\left(-\mu_- u\right)\right) e^{-au}.$$

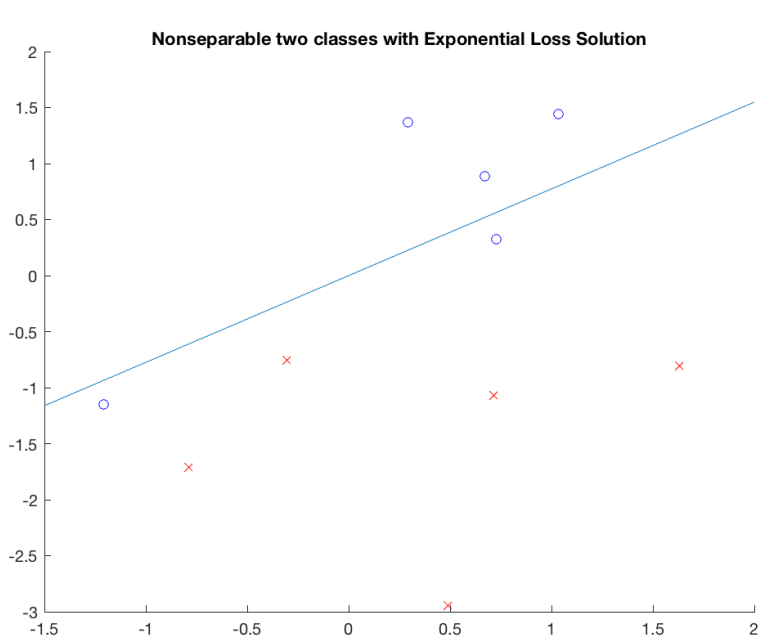
Assumption 3 *The negative loss derivative* $-\ell'(u)$ *has a tight exponential tail (Definition 2).*

For example, the exponential loss $\ell(u) = e^{-u}$ and the commonly used logistic loss $\ell(u) = \log\left(1 + e^{-u}\right)$ both follow this assumption with $a = c = 1$. We will assume $a = c = 1$ — without loss of generality, since these constants can be always absorbed by re-scaling $\mathbf{x}_n$ and η.

Nonseparable classification?

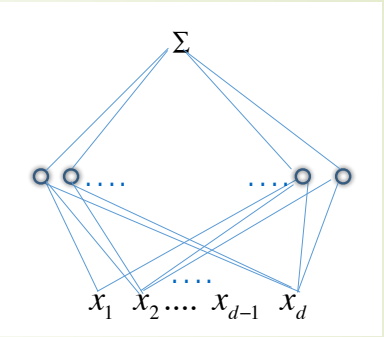


Left: GD solution for logistic loss



Right: GD solution for exponential loss

Deep Networks makes it separable

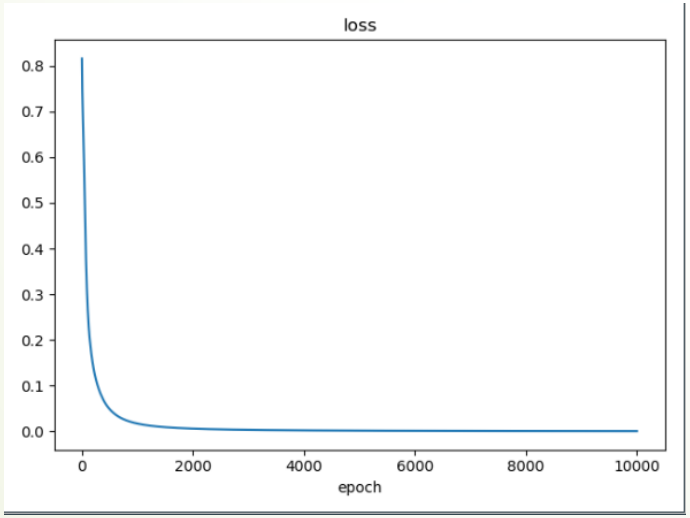


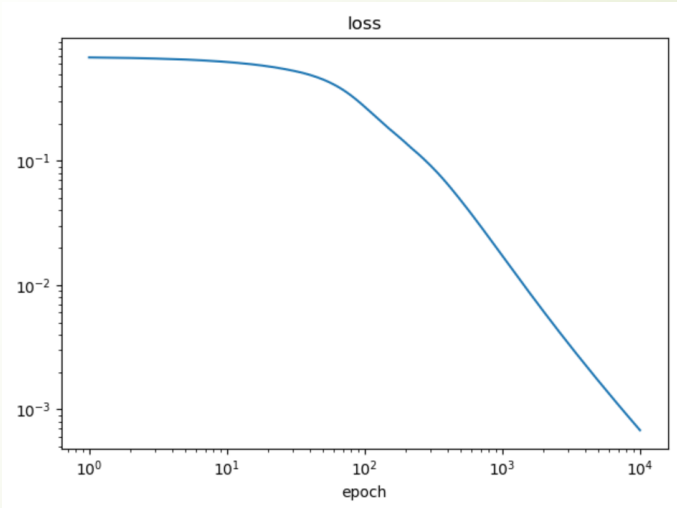
2-layer neural network:

$$f(x) = W_2\sigma(W_1 x)$$

where $\sigma(u) = \max(0, u)$ is ReLU, $W_1 \in R^{d\times q}$, and $W_2 \in R^{q\times 1}$

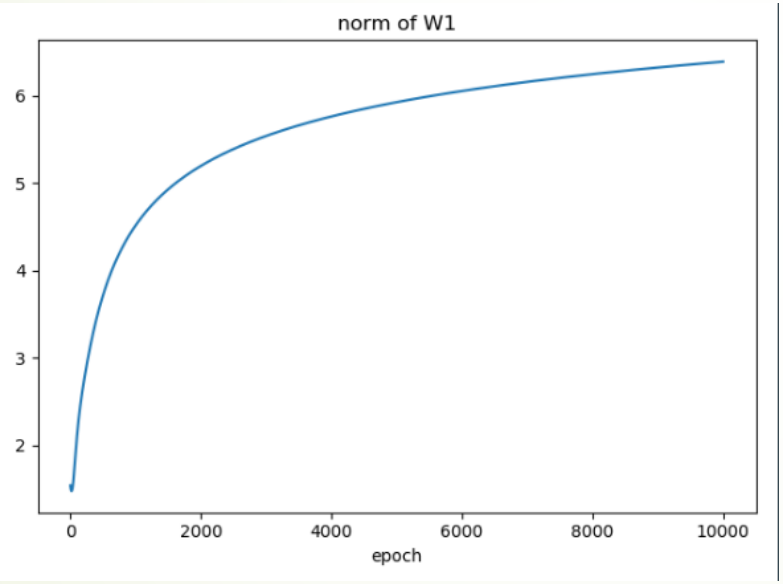
For large q, e.g. q=5, it becomes separable: logistic loss drops down at ~1/k

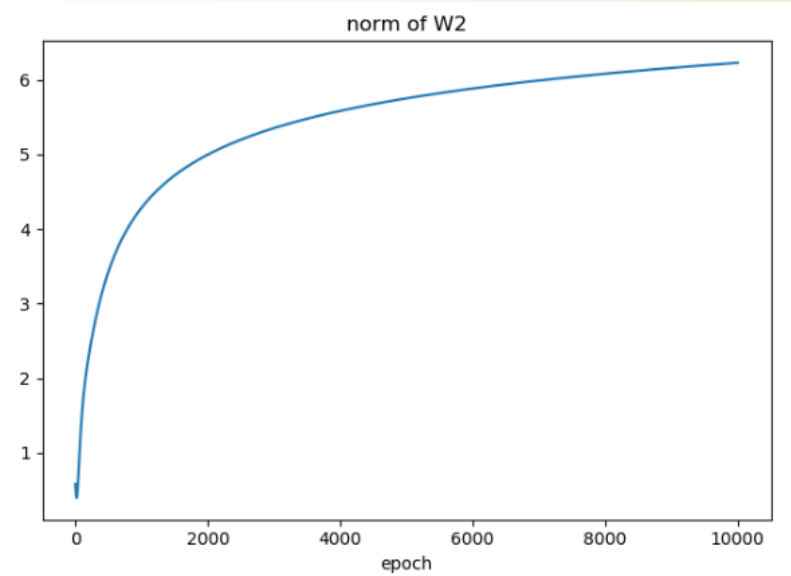




By *Yifei HUANG*, HKUST

Both W_1 and W_2 grows to infinity (log k)!





By *Yifei HUANG*

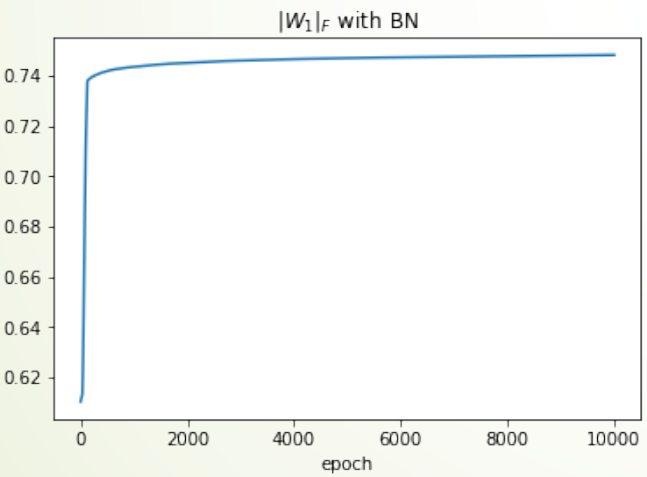
Batch Normalization

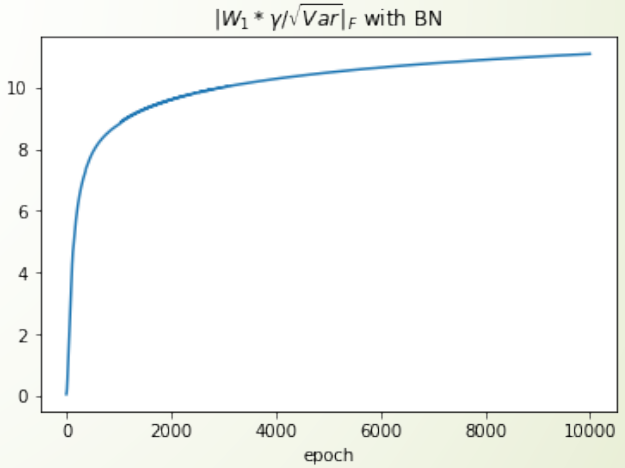
Batch Normalization:

$$\tilde{x}_p = \frac{x_p - \mu_t(x_p)}{\sqrt{\sigma_t^2(x_p) + \epsilon}}\gamma + \beta, \quad \epsilon = 10^{-5}.$$

So a linear layer $x_{l+1} = W_l x_l + b_l$ followed by Batch Normalization has its weight matrix rescaled by

$$\tilde{W}_l = \frac{\gamma}{\sigma_t} W_l.$$

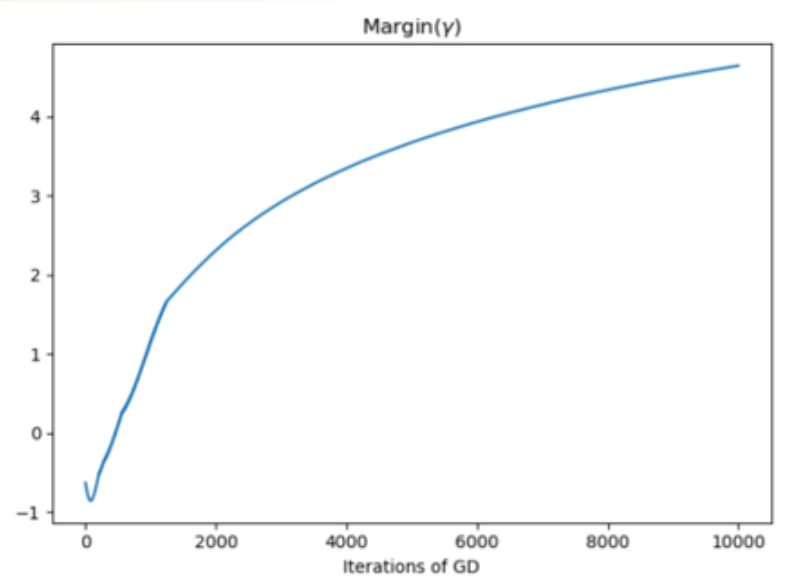


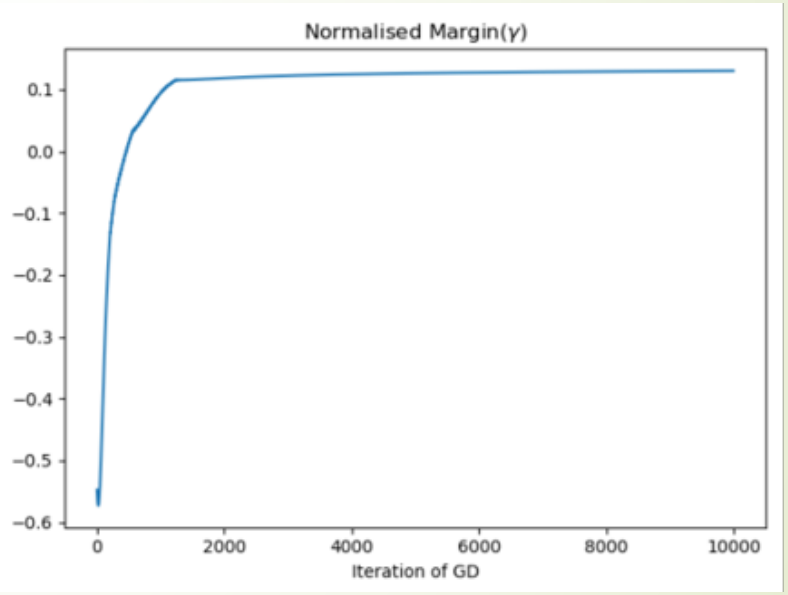


Normalized Margin is scale invariant and keeps on improving in training

$$\gamma := \min_i y_i f(x_i)$$

$$\gamma_n := \frac{\gamma}{\prod_{i=1}^{2} \|W_i\|}$$





After about 1000 epochs, it correctly classifies all training examples and continues to improve the margin.

By *Yifei HUANG*.

Spectrally-Normalized Margin Bounds
[Bartlett-Foster-Telgarsky'2017]

$$F_{\mathcal{A}}(x) := \sigma_L(A_L\sigma_{L-1}(A_{L-1}\cdots\sigma_1(A_1x)\cdots)). \tag{1.1}$$

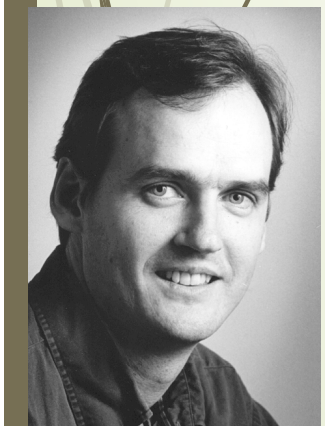
$\mathcal{A} = (A_1, \ldots, A_L)$ *reference matrices* $(M_1, \ldots, M_L)$ with the same dimensions as $A_1, \ldots, A_L$

$$R_{\mathcal{A}} := \left(\prod_{i=1}^{L} \rho_i \|A_i\|_\sigma\right)\left(\sum_{i=1}^{L} \frac{\|A_i^\top - M_i^\top\|_{2,1}^{2/3}}{\|A_i\|_\sigma^{2/3}}\right)^{3/2}. \tag{1.2}$$

Theorem 1.1. *Let nonlinearities* $(\sigma_1, \ldots, \sigma_L)$ *and reference matrices* $(M_1, \ldots, M_L)$ *be given as above (i.e.,* σ_i *is* ρ_i*-Lipschitz and* $\sigma_i(0) = 0$*). Then for* $(x, y), (x_1, y_1), \ldots, (x_n, y_n)$ *drawn iid from any probability distribution over* $\mathbb{R}^d \times \{1, \ldots, k\}$*, with probability at least* $1 - \delta$ *over* $((x_i, y_i))_{i=1}^n$*, every margin* $\gamma > 0$ *and network* $F_{\mathcal{A}} : \mathbb{R}^d \to \mathbb{R}^k$ *with weight matrices* $\mathcal{A} = (A_1, \ldots, A_L)$ *satisfy*

$$\Pr\left[\arg\max_j F_{\mathcal{A}}(x)_j \neq y\right] \leq \widehat{\mathcal{R}}_\gamma(F_{\mathcal{A}}) + \widetilde{\mathcal{O}}\left(\frac{\|X\|_2 R_{\mathcal{A}}}{\gamma n}\ln(W) + \sqrt{\frac{\ln(1/\delta)}{n}}\right),$$

where $\widehat{\mathcal{R}}_\gamma(f) \leq n^{-1}\sum_i \mathbb{1}\left[f(x_i)_{y_i} \leq \gamma + \max_{j \neq y_i} f(x_i)_j\right]$ *and* $\|X\|_2 = \sqrt{\sum_i \|x_i\|_2^2}$.

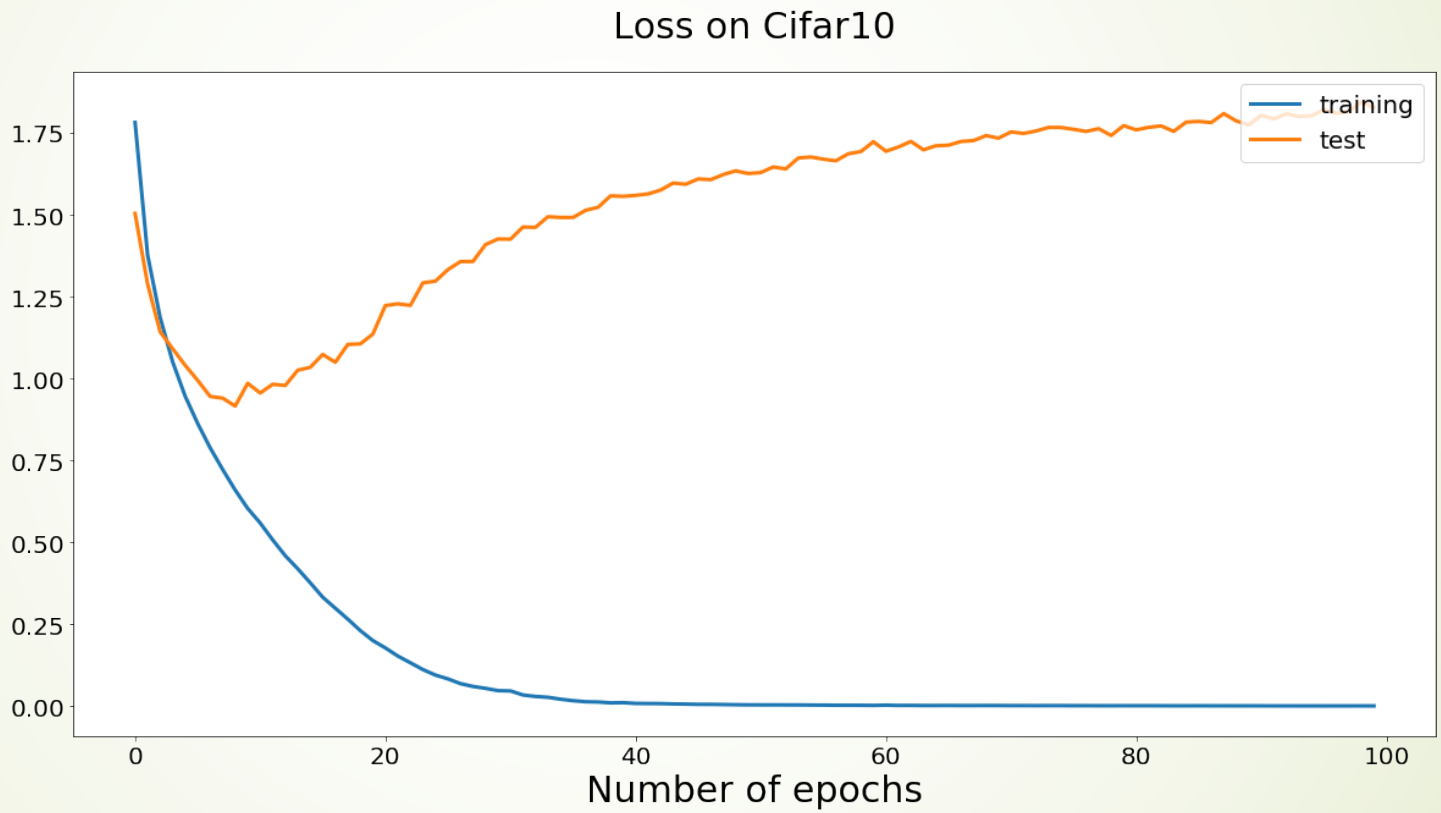


N. Srebro et al. ICLR 2018 has another PAC-Bayes generalization error bound.

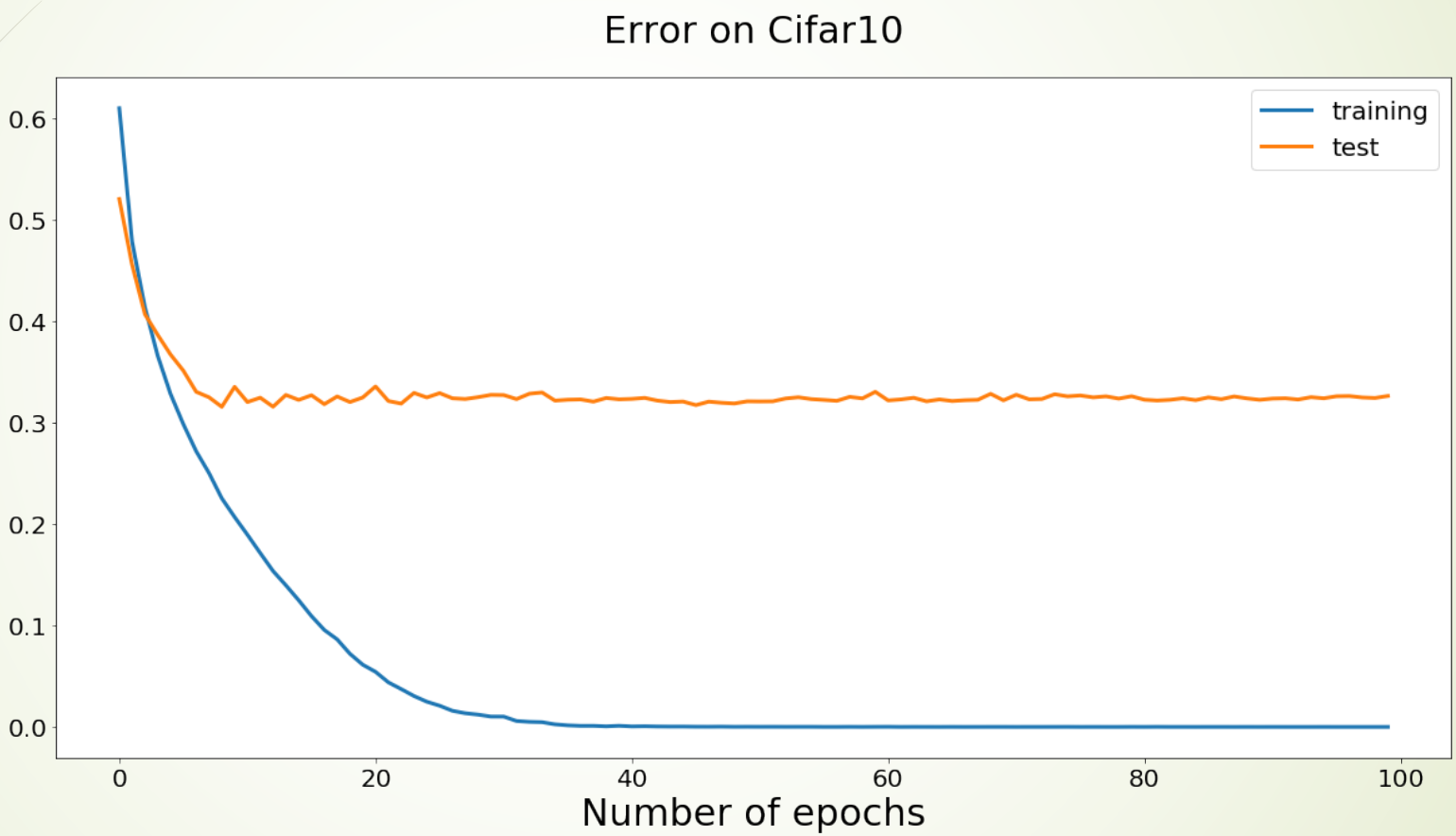
Train 5-layer CNN on Cifar10

- 5 convolutional layers:
  - 100 channels,
  - kernel 3x3,
  - stride 2,
  - padding 1
- ReLU, and/or
- Batch-normalization (batch_size=100)
- Last layer is fully-connected classifier with Cross-Entropy loss

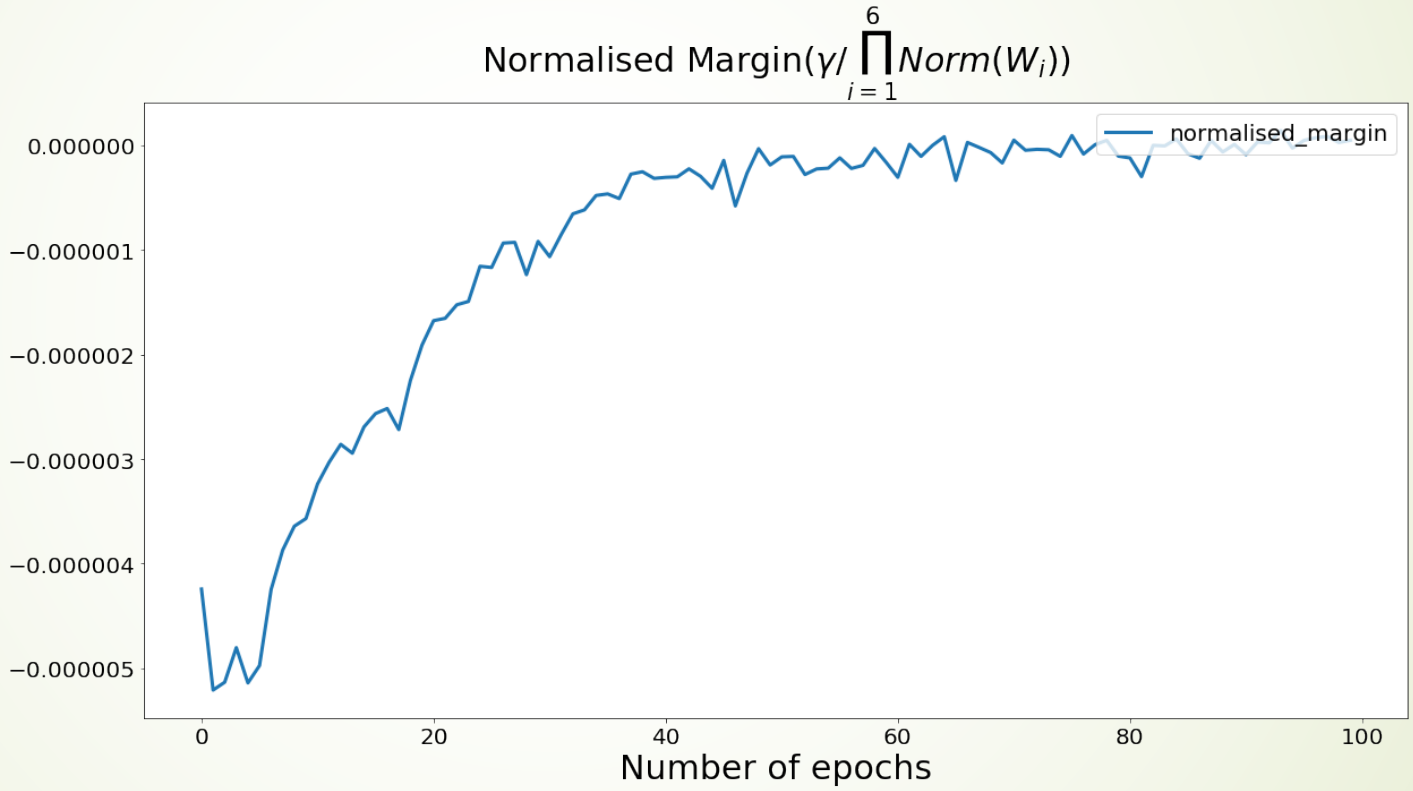
Cross-Entropy Loss overfits!



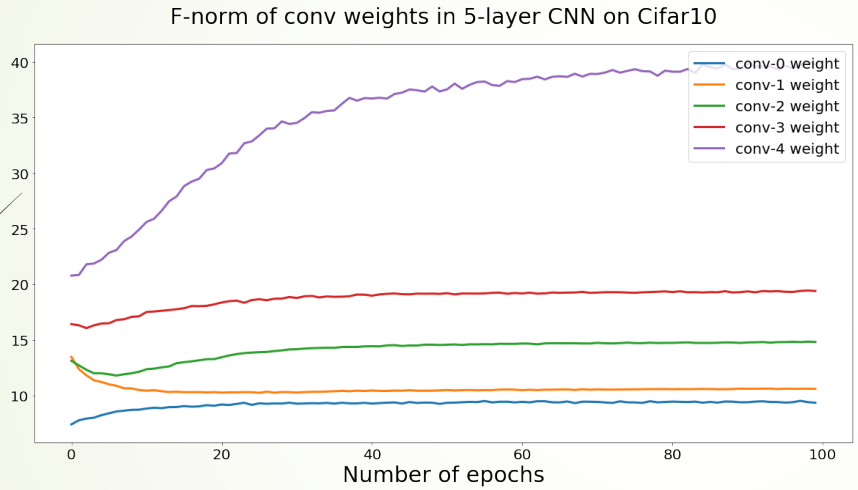
Yet, Test Error does not overfit!



After rescaling, Normalized Margin keeps on improving!

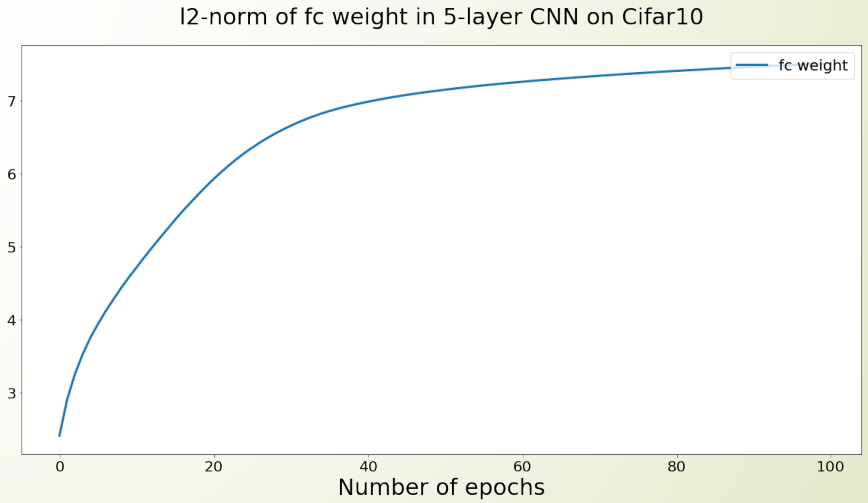


Rescaled weights are growing…



Late layer weights change fast while early ones change slowly.

Classification layer weight grows



Summary

- For separable classification, GD for logistic regression, cross entropy loss, and exponential loss, etc., converges at infinity to the maximal margin solution in direction
- For non-separable classification, over-parameterized deep networks may make it separable and GD converges toward improving margin at infinity
- Spectrally-normalized margin provides a data-dependent measure of generalization error

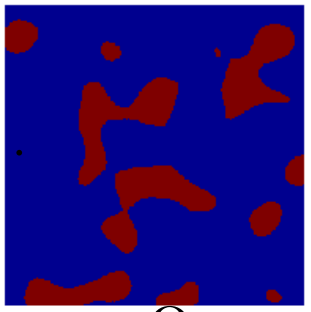
What's the Landscape of Empirical Risks and How to optimize them efficiently?

Over-parameterized models lead to simple landscapes while SGD finds flat minina.

Sublevel sets and topology

- Given loss $E(\theta)$, $\theta \in \mathbb{R}^d$, we consider its representation in terms of level sets:

$$E(\theta) = \int_0^\infty \mathbf{1}(\theta \in \Omega_u) du \ , \ \Omega_u = \{y \in \mathbb{R}^d \ ; \ E(y) \leq u\}$$



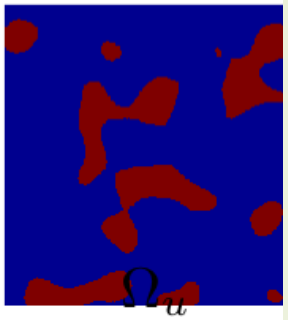
Ω_u

- A first notion we address is about the topology of the level sets .

- In particular, we ask how connected they are, i.e. how many connected components N_u at each energy level u?

Topology of Non-convex Risk Landscape

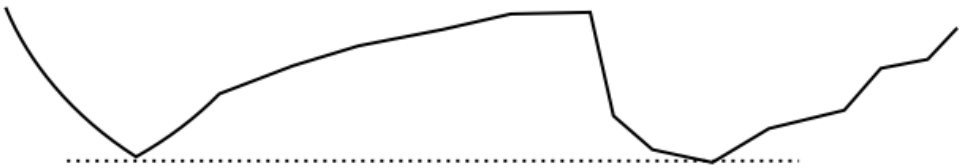
- A first notion we address is about the topology of the level sets .
  - In particular, we ask how connected they are, i.e. how many connected components N_u at each energy level u?
- This is directly related to the question of global minima:



Proposition: If $N_u = 1$ for all u then E has no poor local minima.

$$(\text{i.e. no local minima } y^* \text{ s.t. } E(y^*) > \min_y E(y))$$

- We say *E* is *simple* in that case.
- The converse is clearly not true.



Weaker: P.1, no spurious local valleys

Given a parameter space Θ and a loss function $L(\theta)$ as in (2), for all $c \in \mathbb{R}$ we define the sub-level set of L as

$$\Omega_L(c) = \{\theta \in \Theta : L(\theta) \leq c\}.$$

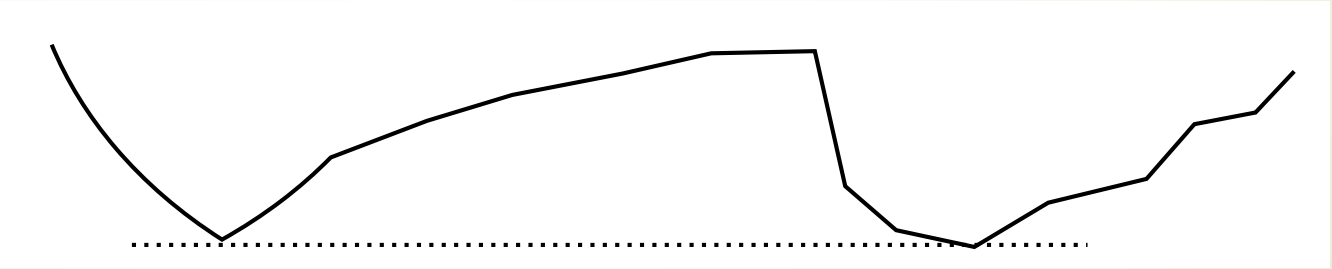
We consider two (related) properties of the optimization landscape. The first one is the following:

P.1 Given any *initial* parameter $\theta_0 \in \Theta$, there exists a continuous path $\theta : t \in [0,1] \mapsto \theta(t) \in \Theta$ such that:

(a) $\theta(0) = \theta_0$

(b) $\theta(1) \in \arg\min_{\theta \in \Theta} L(\theta)$

(c) The function $t \in [0,1] \mapsto L(\theta(t))$ is non-increasing.



The landscape has no spurious local valleys.

Overparameterized LN -> Single Basin

$$E(W_1, \ldots, W_K) = \mathbb{E}_{(X,Y)\sim P} \|W_K \ldots W_1 X - Y\|^2 \ .$$

Proposition: [BF'16]

1. If $n_k > \min(n, m)$, $0 < k < K$, then $N_u = 1$ for all u.

2. (2-layer case, ridge regression)
   $E(W_1, W_2) = \mathbb{E}_{(X,Y)\sim P} \|W_2 W_1 X - Y\|^2 + \lambda(\|W_1\|^2 + \|W_2\|^2)$
   satisfies $N_u = 1 \ \forall \ u$ if $n_1 > \min(n, m)$.

- We pay extra redundancy price to get simple topology.

Bruna, Freeman, 2016

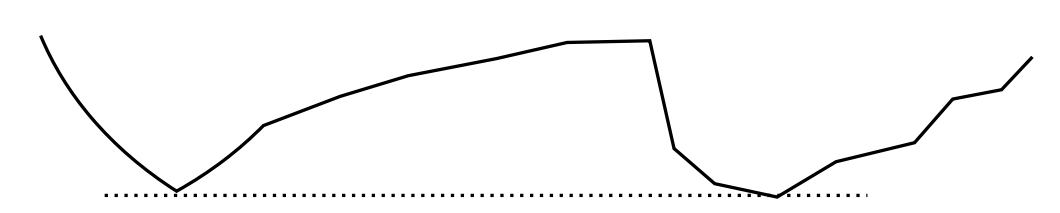
Venturi-Bandeira-Bruna'18

$$\Phi(x;\theta) = W_{K+1}\cdots W_1 x \ , \tag{13}$$

where $\theta = (W_{K+1}, W_K, \ldots, W_2, W_1) \in \mathbb{R}^{n\times p_{K+1}} \times \mathbb{R}^{p_{K+1}\times p_K} \times \cdots \mathbb{R}^{p_2\times p_1} \times \mathbb{R}^{p_1\times n}$.

Theorem 8 *For linear networks* (13) *of any depth* $K \geq 1$ *and of any layer widths* $p_k \geq 1$, $k \in [1, K+1]$, *and input-output dimensions* n, m, *the square loss function* (2) *admits no spurious valleys.*

Symmetry $f(W_i) = f(QW_i)$ $(Q \in GL(R^{n_l}))$ helps remove the network width constraint.



2-layer Neural Networks

[Venturi, Bandeira, Bruna, 2018]

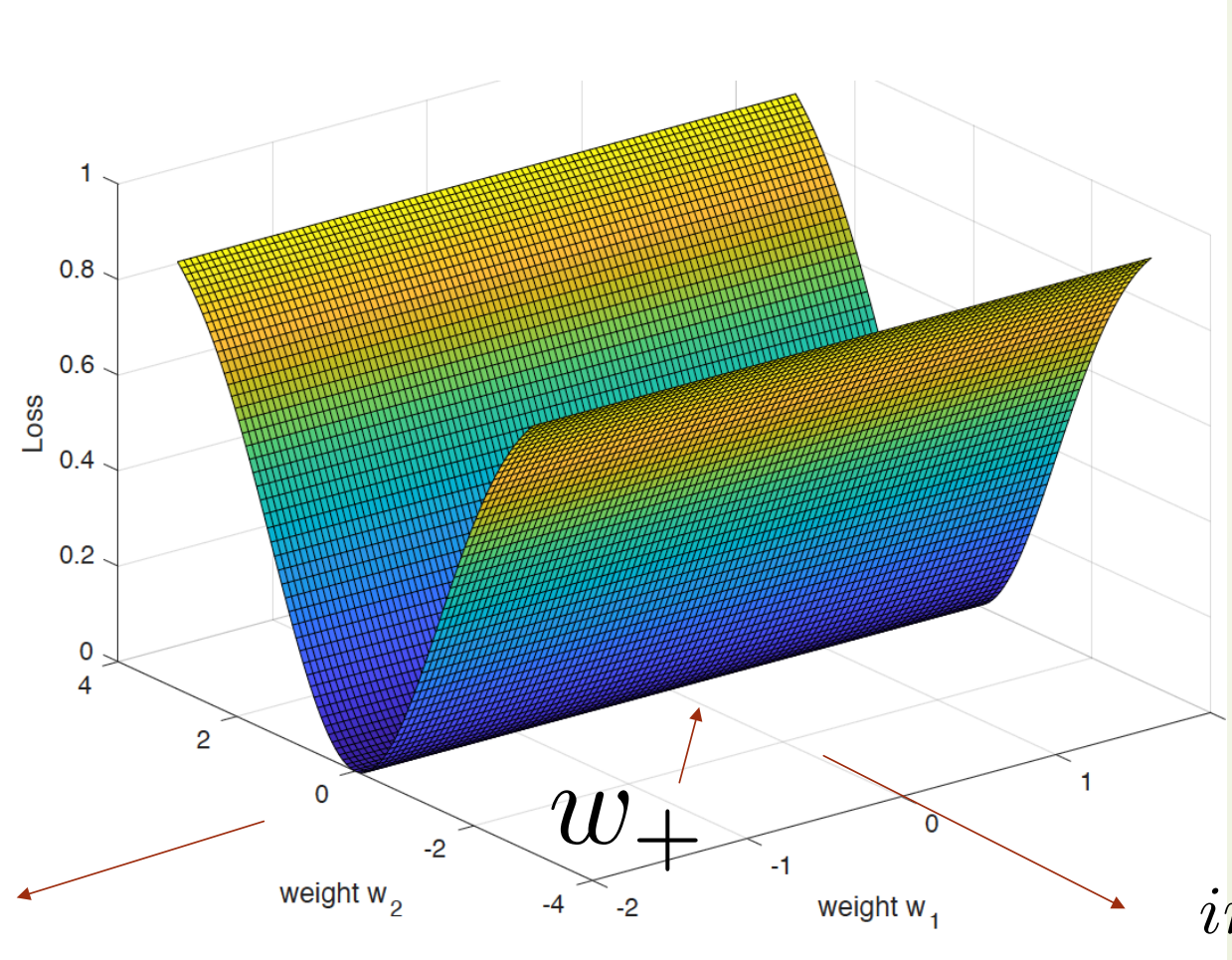
Theorem 5 *The loss function*

$$L(\theta) = \mathbb{E}\|\Phi(X;\theta) - Y\|^2$$

of any network $\Phi(x;\theta) = U\rho W x$ *with effective intrinsic dimension* $q < \infty$ *admits no spurious valleys, in the over-parametrized regime* $p \geq q$. *Moreover, in the over-parametrized regime* $p \geq 2q$ *there is only one global valley.*

- Reproducing Kernel Hilbert Spaces (RKHS) are exploited in the proof!
- Matrix factorizations are of similar ideas.

Over-parameterized Landscapes: as p>n, equilibria are all degenerate



$ker(X)$

$im(X^T)$

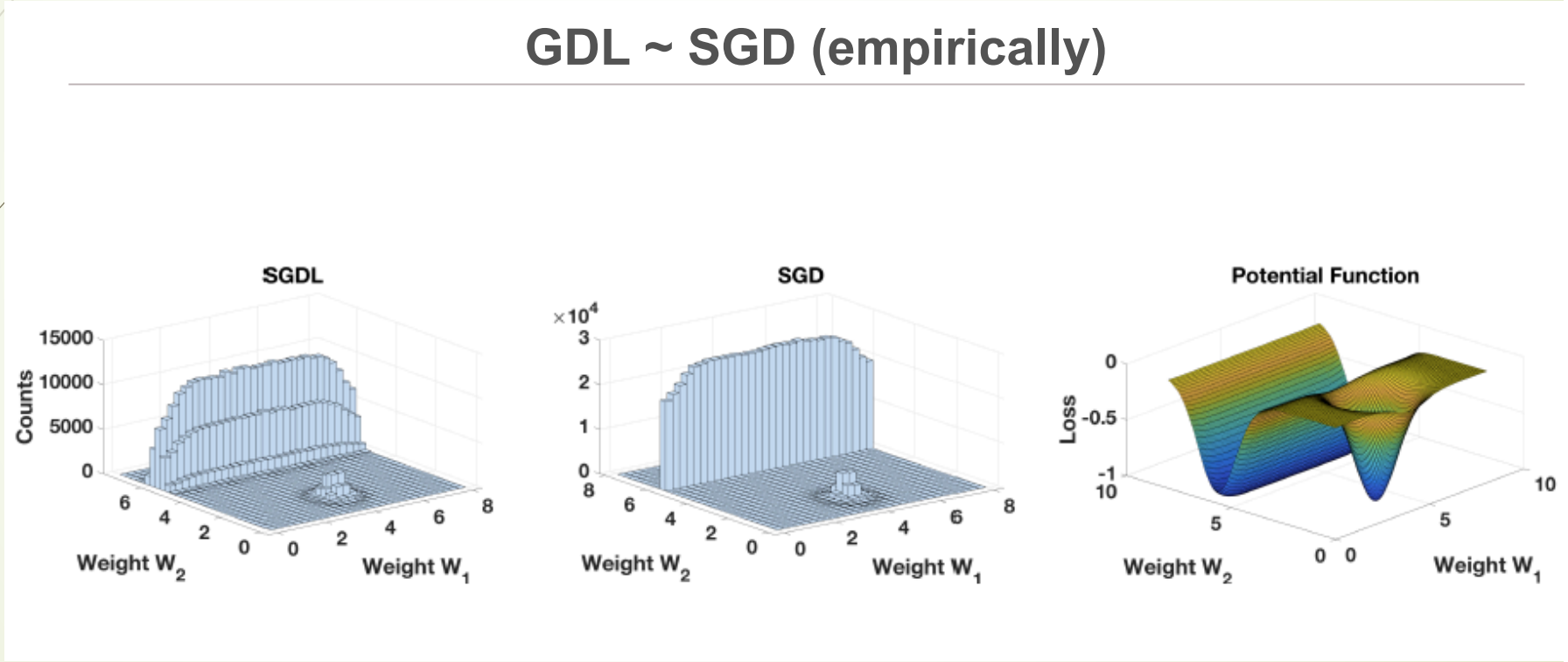
Recall: SGD behaves like Gradient Descent Langevin dynamics (GDL)

$$\frac{dw}{dt} = -\gamma_t \nabla V(w(t), z(t)) + \gamma_t' dB(t)$$

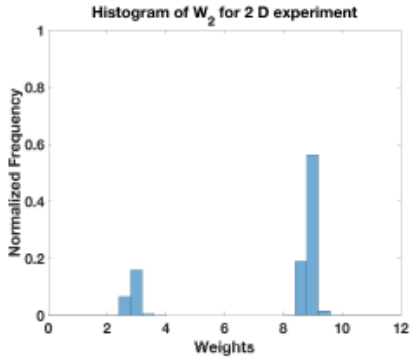
with the Boltzmann equation as asymptotic "solution"

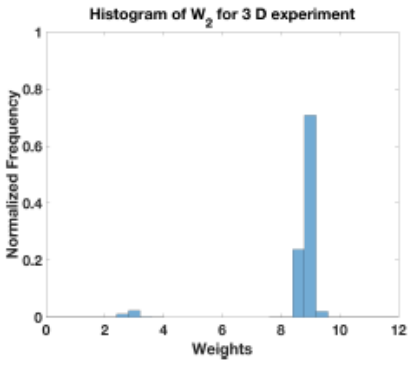
$$p(w) \sim \frac{1}{Z} = e^{-\frac{V(w)}{T}}$$

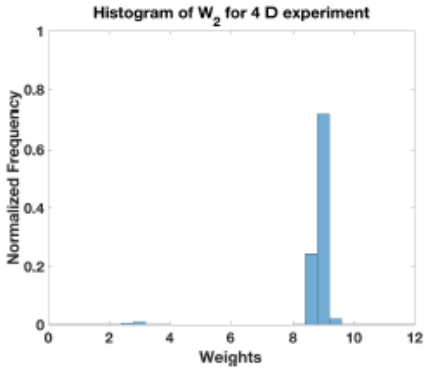
SGD/GDL selects larger volume minima e.g. degenerate

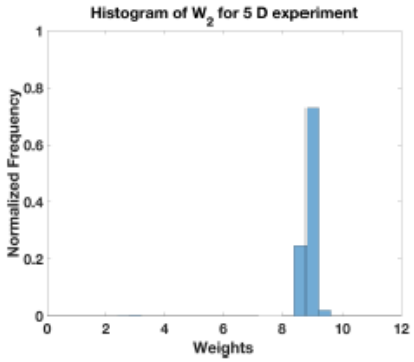


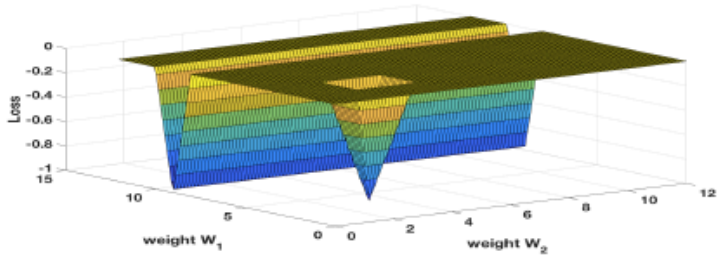
Concentration because of high dimensionality











Poggio, Rakhlin, Golovitc, Zhang, Liao, 2017

Brains Minds + Machines

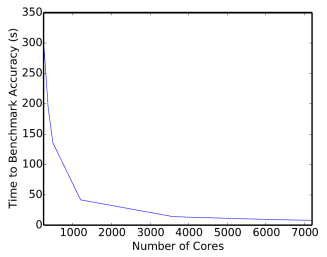
Summary

- Over-parameterization may lead to simple risk landscapes with **flat** (degenerate) global minima
- SGD tends to find **flat** global minima like Langevin Dynamics
- But SGD for multilayer neural networks suffer from vanishing gradient issue (architecture revision: ReLU, ResNet, LSTM)…

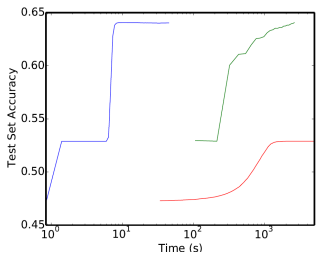
Alternative: Variable Splitting with Block Coordinate Descent

- ADMM-type: **Taylor** et al. ICML 2016
- **Proximal Propagation**, ICLR 2018
- BCD with zero Lagrangian multiplier: **Zhang** et al. NIPS 2017
- Discrete EMSA of PMP: **Qianxiao LI** et al 2017, talk on Monday in IAS workshop
  - No-vanshing gradients and parallelizable
- Global convergence can be established via Kurdyka-Lojasiewicz inequality: with **Jinshan Zeng and Tsz Kit Lau et al.**

Experiment results on Higgs dataset from Taylor et al'16

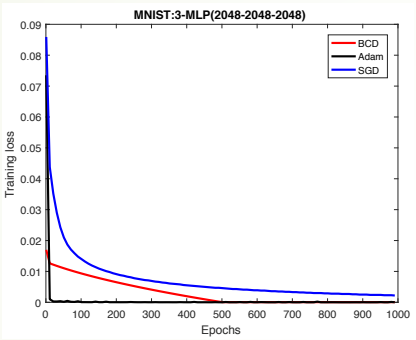


(a) **Time required for ADMM to reach 64% test accuracy when parallelized over varying levels of cores**. L-BFGS on a GPU required 181 seconds, and conjugate gradients required 44 minutes. SGD never reached 64% accuracy.

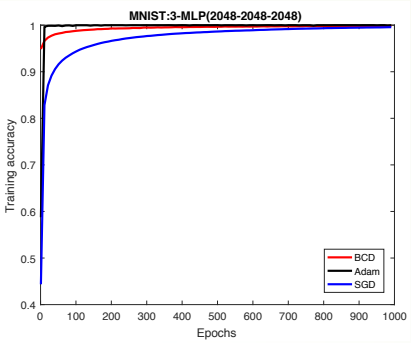


(b) **Test set predictive accuracy as a function of time** for ADMM on 7200 cores (blue), conjugate gradients (green), and SGD (red). Note the x-axis is scaled logarithmically.

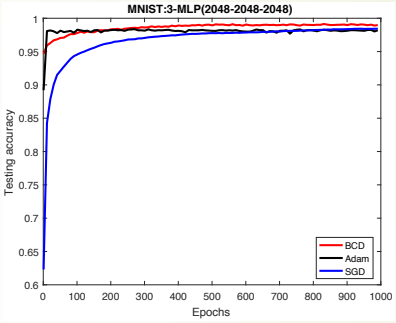
Adam, BCD, vs. SGD



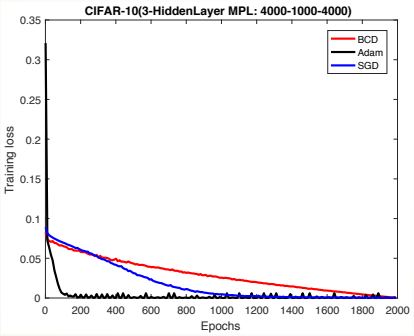
(a) MNIST (Training loss)



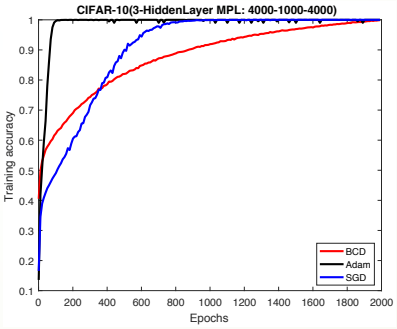
(b) MNIST (Training Acc.)



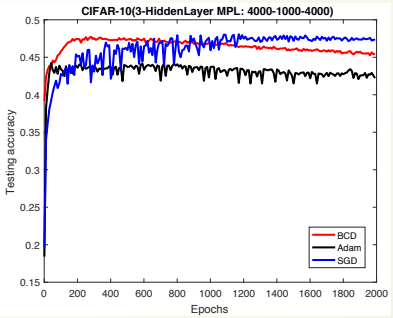
(c) MNIST (Testing Acc.)



(d) CIFAR-10 (Training loss)



(e) CIFAR-10 (Training Acc.)

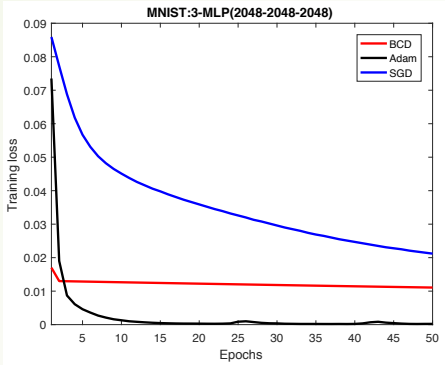


(f) CIFAR-10 (Testing Acc.)

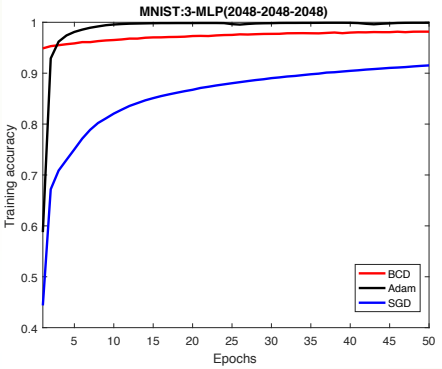
Figure 1: Comparisons on MNIST and CIFAR-10 datasets. The first row consists of the figures including the curves of training loss, training accuracy and testing accuracy on MNIST dataset. The second row gives the associated figures on CIFAR-10 dataset.

by **Jinshan Zeng** et al.

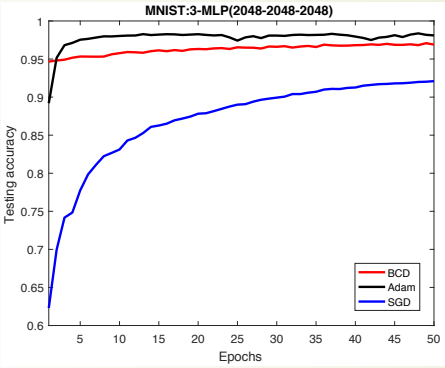
BCD may be good initializers



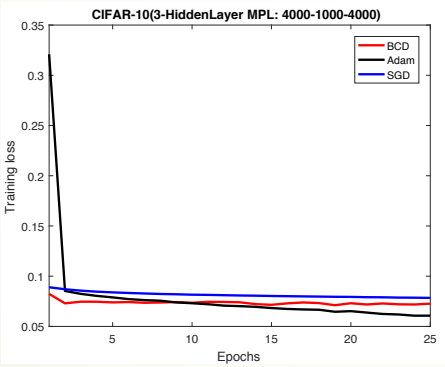
(a) MNIST (Training loss)



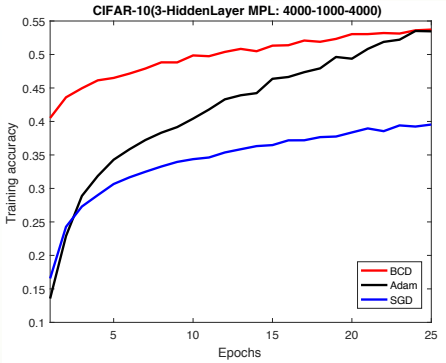
(b) MNIST (Training Acc)



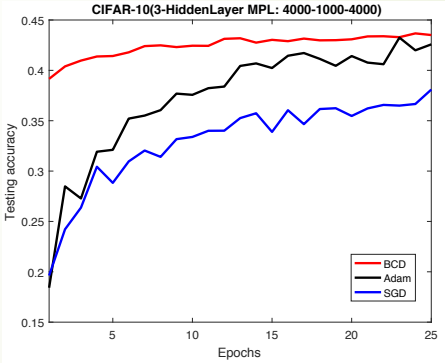
(c) MNIST (Testing Acc)



(d) CIFAR-10 (Training loss)



(e) CIFAR-10 (Training Acc)



(f) CIFAR-10 (Testing Acc)

Figure 2: Comparisons of the performance of three methods at the early stage.

by **Jinshan Zeng** et al.

Thank you!