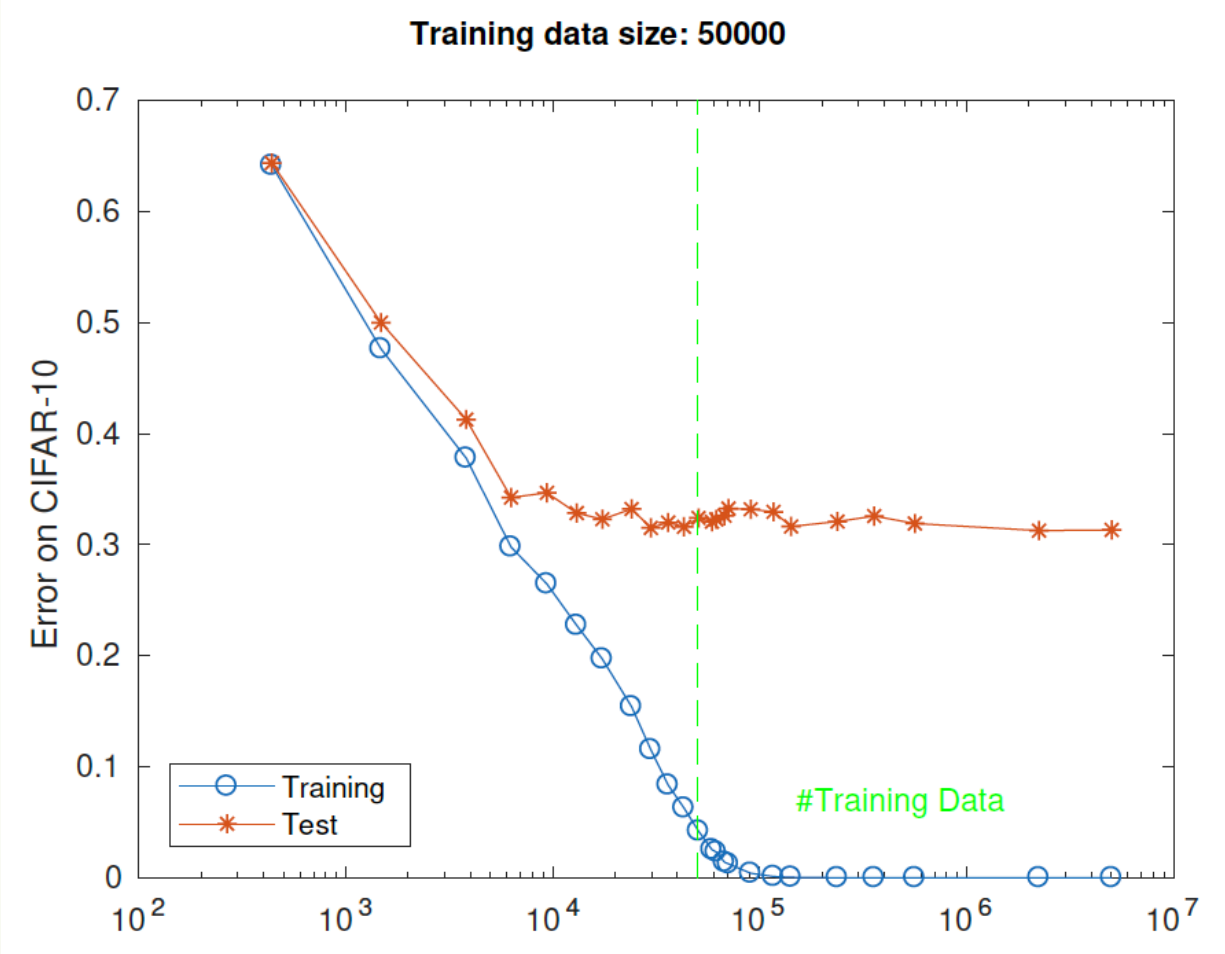
Implicit Regularization in Gradient Descent

Yuan YAO

HKUST

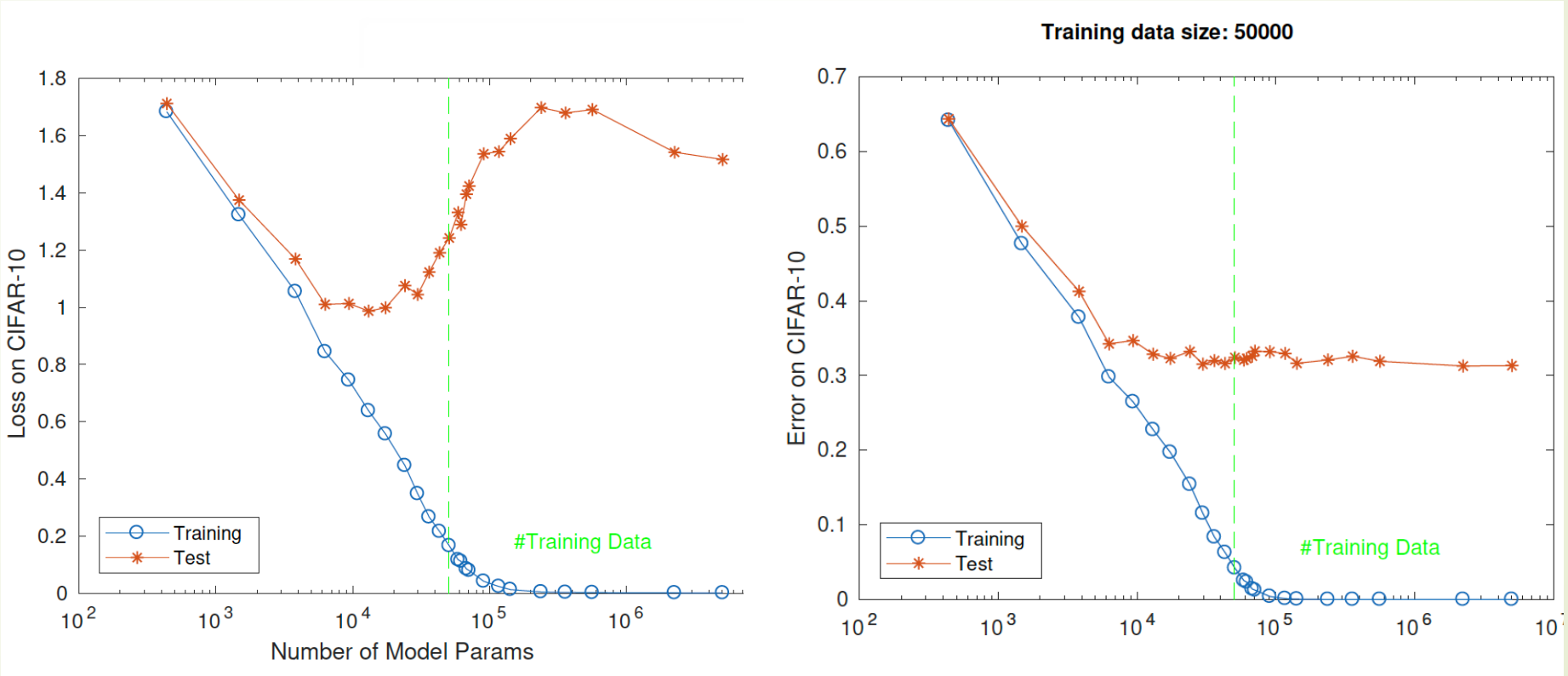
Based on Tomaso Poggio, Peter Bartlett, Leo Breiman, Nati Srebro, et al.

You might have reproduced this:



As iterations go, training error goes down to zero, but test error does not increase. Why overparametric models do not overfit here? -- Tommy Poggio, 2018

Overfit of test **loss**, Nonoverfit of test **error!**



Tommy Poggio, 2018

New challenges to understanding

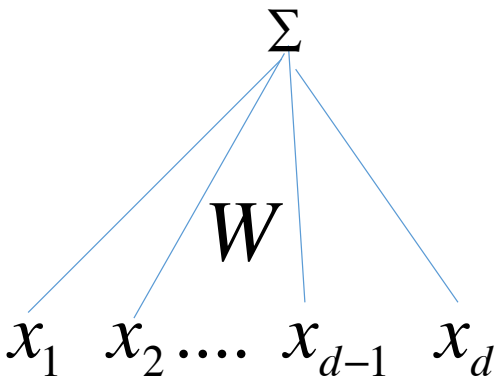
- Why do big models generalize well without overfitting?
- Big (**overparametric**) models may have **simple landscape** of empirical risks, easy to find global optima (zero-training error), *Joan Bruna et al.*
- *Big (**overparametric**) models may **generalize** well: gradient based algorithms tend to find **max margin** models which generalize well, Srebro, Poggio et al.*

Linear Regression Case

A. Gradient Descent converges to minimal 2-norm solution

B. Early stopping plays an equivalent (yet more general) role as l2-regularization

Implicit Regularization by GD/SGD in Linear Regression (1-layer Linear Network with Square Loss)



$$W = YX^{\dagger}$$

Corollary 1. *When initialized with zero, both GD and SGD converges to the minimum-norm solution.*

Min norm solution is the limit for $\lambda \to 0$ *of regularized solution*

Minimal Norm Least Square solution

Let us look in more detail at the gradient descent solution (from (8)). Consider the following setup: $X = (x_1, \ldots, x_n)^\top \in \mathbb{R}^{n,d}$ are the data points, with $d > n$. We further assume that the data matrix is of full row rank: $\text{rank}(X) = n$. Let $y \in \mathbb{R}^n$ be the labels, and consider the following linear system:

$$Xw = y \tag{9}$$

where $w \in \mathbb{R}^d$ is the weights to find. This linear system has infinite many solutions because X is of full row rank and we have more parameters than the number of equations. Now suppose we solve the linear system via a least square formulation

$$L(w) = \frac{1}{2n}\|Xw - y\|^2$$

by using gradient descent (GD) or stochastic gradient descent (SGD).

$$w_+ \triangleq X^\top (XX^\top)^{-1} y$$

is the minimum norm solution.

Proof Note since X is of full row rank, so $XX^\top$ is invertible. By definition,

$$Xw_+ = XX^\top (XX^\top)^{-1} y = y$$

Therefore, w_+ is a solution. Now assume $\hat{w}$ is another solution to (9), we show that $\|\hat{w}\| \geq \|w_+\|$. Consider the inner product

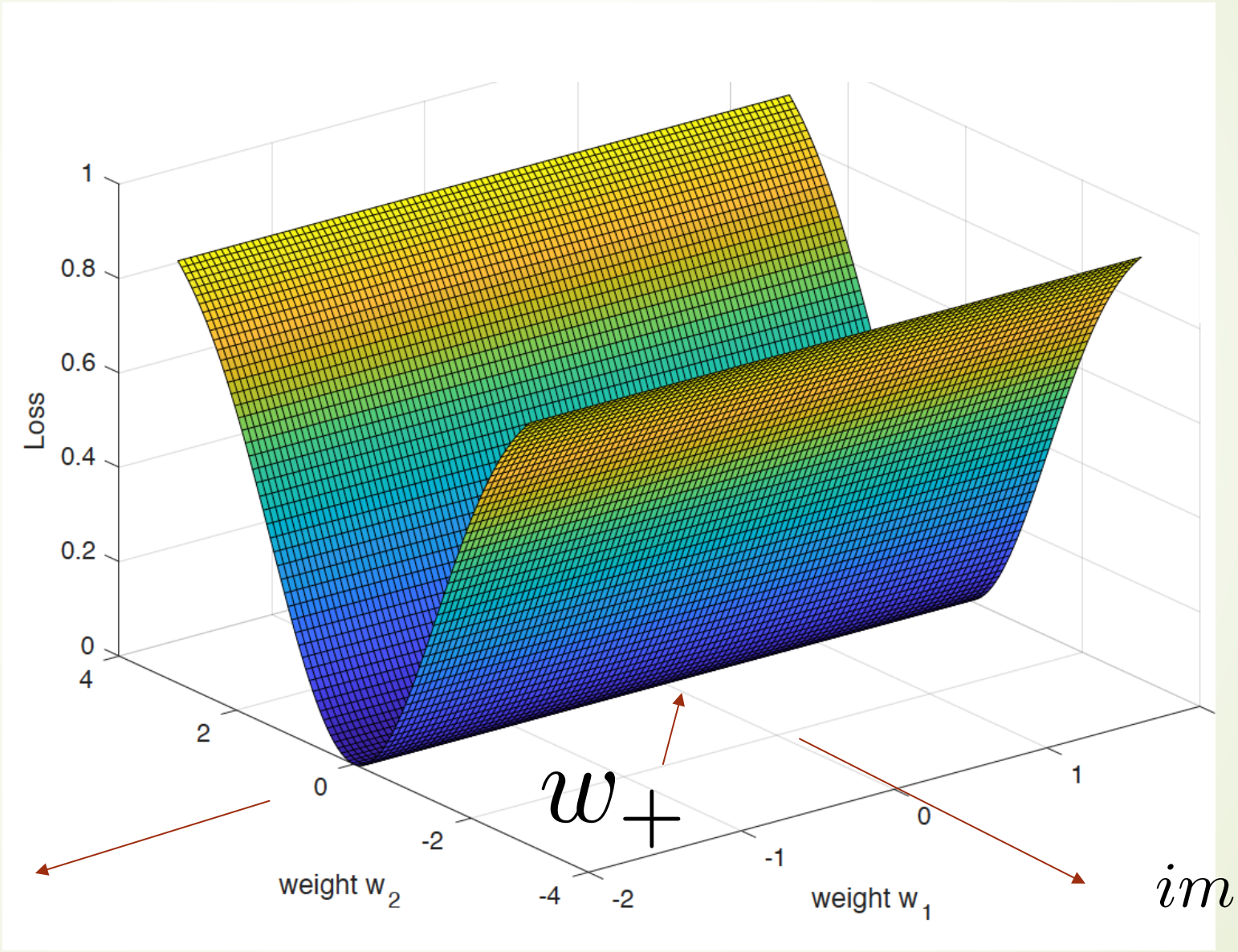
$$\begin{aligned}
\langle w_+, \hat{w} - w_+ \rangle &= \langle X^\top (XX^\top)^{-1} y, \hat{w} - w_+ \rangle \\
&= \langle (XX^\top)^{-1} y, X\hat{w} - X_+ \rangle \\
&= \langle (XX^\top)^{-1} y, y - y \rangle \\
&= 0
\end{aligned}$$

Therefore, w_+ is orthogonal to $\hat{w} - w_+$. As a result, by Pythagorean theorem,

$$\|\hat{w}\|^2 = \|(\hat{w} - w_+) + w_+\|^2 = \|\hat{w} - w_+\|^2 + \|w_+\|^2 \geq \|w_+\|^2$$

$\hat{w} - w_+$

$ker(X)$



$im(X^T)$

Lemma 3. *When initializing at zero, the solutions found by both GD and SGD for problem* (10) *live in the span of rows of* X. *In other words, the solutions are of the following parametric form*

$$w = X^\top \alpha \tag{12}$$

for some $\alpha \in \mathbb{R}^n$.

Proof.

Proof

The gradient for (10) is

$$\nabla_w L(w) = \frac{1}{n} X^\top (Xw - y) = X^\top e$$

where we define $e = (1/n)(Xw - y)$ to be the error vector. GD use the following update rule:

$$w_{t+1} = w_t - \eta_t \nabla_w L(w_t) = w_t - \eta_t X^\top e_t$$

Expanding recursively, and assume $w_0 = 0$. we get

$$w_t = \sum_{\tau=0}^{t-1} -\eta_\tau X^\top e_\tau = X^\top \left(-\sum_{\tau=0}^{t-1} \eta_\tau e_\tau \right)$$

The same conclusion holds for SGD, where the update rule could be explicitly written as

$$w_{t+1} = w_t - \eta_t (x_{i_t}^\top w - y_{i_t}) x_{i_t}$$

where (x_{i_t}, y_{i_t}) is the pair of sample chosen at the t-th iteration. The same conclusion follows with $w_0 = 0$.

Why Early Stopping?

- GD asymptotically converges to minimal norm least square solution
- The minimal norm least square may overfit the training data if it is noisy
- Can we use early stopping as a regularization?
  - Yes, early stopping plays the same role as Ridge regression (Tikhonov regularization) [Yao-Rosasco-Caponetto'2005]

Let

$$f_\lambda = \arg\min_{f \in \mathscr{H}_K} \mathscr{E}(f) + \lambda \|f\|_K^2,$$

be the solution of problem (1) with Tikhonov regularization. It is known [e.g. Cucker and Smale 2002] that

$$f_\lambda = (L_K + \lambda I)^{-1} L_K f_\rho.$$

$$f_t = \sum_{i=0}^{t-1} (I - L_K)^i L_K f_\rho = \sum_{i=0}^{t-1} (I - L_K)^i (I - (I - L_K)) f_\rho = (I - (I - L_K)^t) f_\rho.$$

Those are *regularization polynomials*.

Now let $(\mu_i, \phi_i)_{i\in\mathbb{N}}$ be an eigen-system of the compact operator $L_K : \mathscr{L}^2_{\rho_X} \to \mathscr{L}^2_{\rho_X}$. Then we have decompositions

$$f_\lambda = \sum_i \frac{\mu_i}{\mu_i + \lambda} \langle f_\rho, \phi_i \rangle \phi_i,$$

and

$$f_t = \sum_i (1 - (1 - \mu_i)^t) \langle f_\rho, \phi_i \rangle \phi_i.$$

Compactness of L_K implies that $\lim_{i\to\infty} \mu_i = 0$. Therefore for most μ_i which are sufficiently small, $\mu_i/(\mu_i + \lambda) \approx 0$ and $1 - (1 - \mu_i)^t$ converges to 0 with gap dropping at least exponentially with t. Therefore both early stopping and Tikhonov regularization can be regarded as low pass

GD with Early stopping:

$$\lambda \sim 1/t$$

seems to have better upper bounds than Tikhonov regularization. In fact, it can be shown [Minh 2005; or Appendix by Minh, in Smale and Zhou 2005] that if $f_\rho \in L_K^r(B_R)$ for some $r > 0$,

$$\|f_\lambda - f_\rho\|_\rho \leq O(\lambda^{\min(r,1)}),$$

and for $r > 1/2$,

$$\|f_\lambda - f_\rho\|_K \leq O(\lambda^{\min(r-1/2,1)}).$$

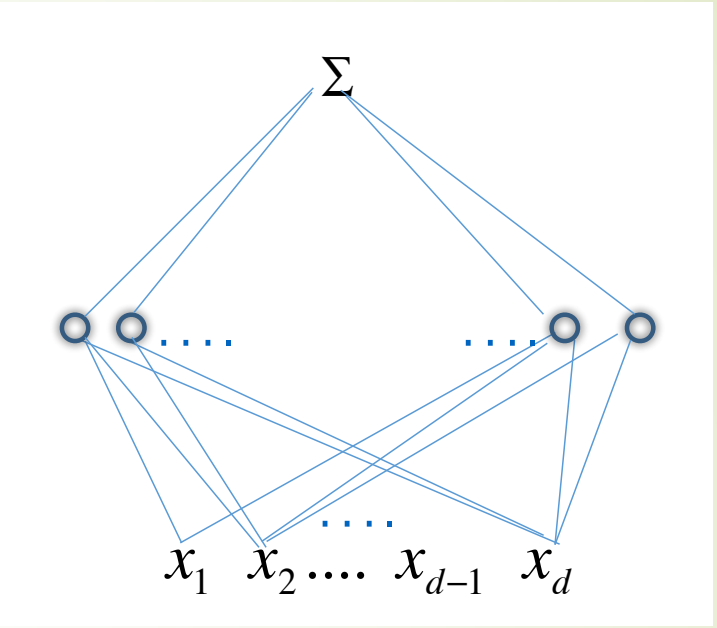
We can see for large r, the upper bound can not go faster than t^{-1} in Tikhonov regularization. On the other hand in early stopping regularization, taking $\theta = 0$ in Theorem 2.9 we have that for $r > 0$,

$$\|f_t - f_\rho\|_\rho \leq O(t^{-r}),$$

and for $r > 1/2$,

$$\|f_t - f_\rho\|_K \leq O(t^{-(r-1/2)}).$$

2-Layer Linear Networks



Model Description and notation Consider a network, d inputs, N hidden *linear* units and d' outputs. We denote the loss with $L(w) = \frac{1}{2}||W_2W_1X - Y||^2$, where $X \in \mathbb{R}^{d,n}, Y \in \mathbb{R}^{d',n}$, $W_2 \in \mathbb{R}^{d',N}$ and $W_1 \in \mathbb{R}^{N,d}$. Let $E = W_2W_1X - Y \in \mathbb{R}^{d',n}$. Let $w = \text{vec}(W_1^\top, W_2^\top) \in \mathbb{R}^{Nd+d'N}$.

GD and SGD converge to the Minimum Norm Solution Assume that $N, d \geq n \geq d'$ (overparametrization). Let $A = W_2 W_1$. For any matrix M, let $\mathrm{Col}(M)$ and $\mathrm{Null}(M)$ be the column space and null space of M.

$$\frac{\partial L}{\partial A^*} = (A^* X - Y) X^\top = 0$$
$$\Leftrightarrow A^* \in \{A = Y X^\top (X X^\top)^+ + B_X : B_X X = 0\}$$

The minimum norm solution $A^*_{\min}$ is the global minimum A^* with its rows not in $\mathrm{Null}(X^\top)$, which is

$$A^*_{\min} = Y X^\top (X X^\top)^+.$$

Due to the over-parameterization with w, for any entries of A, there exist entries of W_1 and W_2 such that $A = W_2 W_1$. Thus, the global minimum solutions in terms of A and w are the same.

Gradient dynamics and Hessian We can write the dynamical system corresponding to its gradient as

$$\dot{W}_1 = -\nabla_{W_1} L(w) = -W_2^\top E X^\top = W_2^\top Y X^\top - W_2^\top W_2 W_1 X X^\top$$

$$\dot{W}_2 = -Y X^\top W_1^\top + W_2 W_1 X X^\top W_1^\top$$

$$\nabla^2 L(w) = \begin{bmatrix} W_2^\top W_2 \otimes X X^\top & C \\ C^\top & I_{d'} \otimes X X^\top W_1 X X^\top W_1 \end{bmatrix} \in \mathbb{R}^{(Nd+d'N),(Nd+d'N)}$$

where

$$C = [W_2^\top \otimes X X^\top W_1^\top] + [I_N \otimes X(E^\top)_{\bullet 1}, \ldots, I_N \otimes X(E^\top)_{\bullet d'}].$$

Here, $(E^\top)_{\bullet i}$ denote the i-th column of $E^\top$.

Lemma 4. *For gradient descent and stochastic gradient descent with any mini-batch size,*

- *any number of the iterations adds no element in* $\mathrm{Null}(X^\top)$ *to the rows of* W_1*, and hence*
- *if the rows of* W_1 *has no element in* $\mathrm{Null}(X^\top)$ *at anytime (including the initialization), the sequence converges to a minimum norm solution if it converges to a solution.*

Proof. From $\frac{\partial L}{\partial \mathrm{vec}(W_1)} = [X \otimes W_2^\top]\mathrm{vec}(E) = \mathrm{vec}(W_2^\top E X^\top)$, we obtain that

$$\frac{\partial L}{\partial W_1} = W_2^\top E X^\top.$$

For SGD with any mini-batch size, let $\bar{X}_t$ be the input matrix corresponding to a mini-batch used at t-th iteration of SGD. Let $\bar{L}_t$ and $\bar{E}_t$ be the corresponding loss and the error matrix. Then, by the same token,

$$\frac{\partial \bar{L}_t}{\partial W_1} = W_2^\top \bar{E}_t \bar{X}_t^\top.$$

From these gradient formulas, the first statement follows by noticing that for any t, $\mathrm{Col}(\bar{X}_t) \subseteq \mathrm{Col}(X) \perp \mathrm{Null}(X^\top)$. The second statement follows the fact that if the rows of W_1 has no element in $\mathrm{Null}(X^\top)$ at anytime t, then for anytime after that time t, $\mathrm{Col}(W_1^\top W_2^\top) \subseteq \mathrm{Col}(W_1^\top) \subseteq \mathrm{Col}(X) \perp \mathrm{Null}(X^\top)$.

Lemma 5. *If $W_2 \neq 0$, every stationary point w.r.t. W_1 is a global minimum.*

Proof. For any global minimum solution A^*, the transpose of the model output is

$$(A^*X)^\top = X^\top (XX^\top)^+ XY^\top$$

which is the projection of $Y^\top$ onto $\mathrm{Col}(X^\top)$. Let $D = [W_2 \otimes X^\top]$. Then, with the transpose of the model output, the loss can be rewritten as

$$L(w) = \frac{1}{2}\|X^\top W_1^\top W_2^\top - Y^\top\|^2 = \frac{1}{2}\|D\mathrm{vec}(W_1^\top) - Y^\top\|^2.$$

The condition for a stationary point yields

$$0 = \frac{\partial L}{\partial \mathrm{vec}(W_1^\top)} = D^T(D\mathrm{vec}(W_1^\top) - Y^\top)$$
$$\Rightarrow D\mathrm{vec}(W_1^\top) = D(D^TD)^+D^TY^\top = \text{Projection of } Y^\top \text{ onto } \mathrm{Col}(D).$$

If $W_2 \neq 0$, we obtain $\mathrm{Col}(D) = \mathrm{Col}(X^\top)$. Hence any stationary point w.r.t. W_1 is a global minimum. □

Theorem 2. *If*

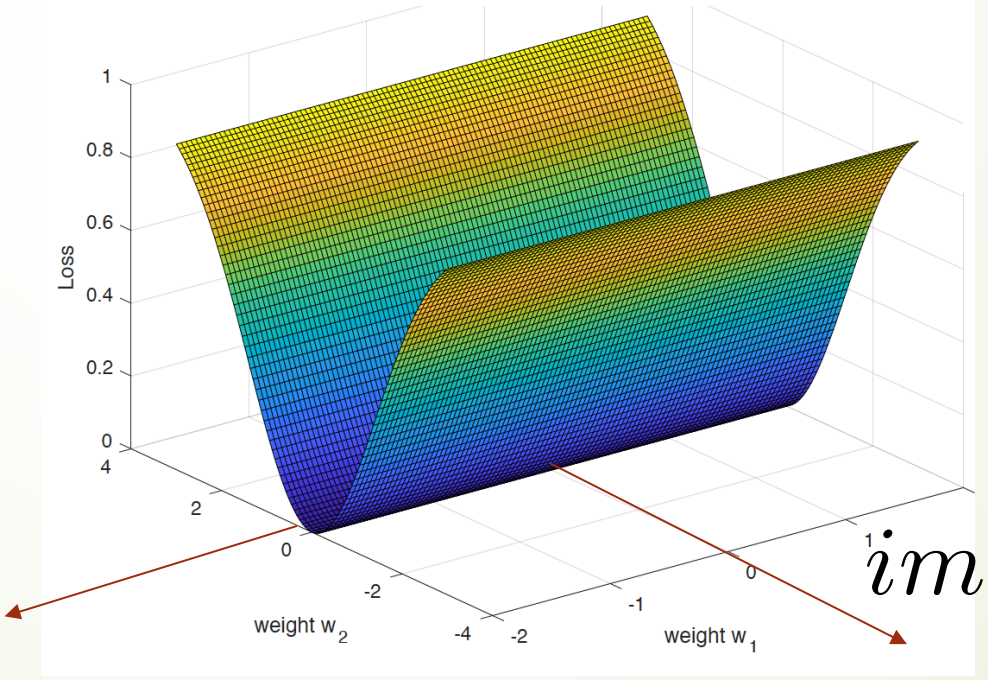
- *the rows of* W_1 *are initialized with no element in* $\mathrm{Null}(X^\top)$,
- *and if* $W_2 \neq 0$,

then gradient descent (and stochastic gradient descent) converges to a minimum norm solution.

It follows from Lemmas 4 and 5 that, if we initialize the rows of W_1 with no element in $\mathrm{Null}(X^\top)$, and if $W_2 \neq 0$, GD and SGD find a minimum norm solution.

As a final remark, the analysis above holds for a broad range of loss function and not just the square loss. Asymptotically $\dot{W} = -\nabla_W L(W^\top X)$, in which we makes explicit the dependence of the loss L on the linear function $W^\top X$. Since $\nabla_W L(W^\top X) = X^\top \nabla_Z L(Z)$ the update to W is in the span of the data X, that is, it does not change the projection of W in the null space of X. Thus if the norm of the components of W in the null space of X was small at the beginning of the iterations, it will remain small at the end. This means that among all the solutions W with zero error, gradient descent selects the minimum norm one.

$ker(X)$



$im(X^T)$

1-hidden layer polynomial network

Consider a polynomial activation for the hidden units. The case of interest here is $n > d$. Consider the loss $L(w) = \frac{1}{2}||P_m(X) - Y||_F^2$ where

$$P_m(X) = W_2(W_1X)^m. \tag{15}$$

where the power m is elementwise.

We obtain, denoting $E = P_m(X) - Y$, with $E \in \mathbb{R}^{d',n}$, $W_2 \in \mathbb{R}^{d',N}$, $W_1 \in \mathbb{R}^{N,d}$, $E' \in \mathbb{R}^{d',d}$

$$\nabla_{W_1} L(w) = m(W_1X)^{m-1} \circ (W_2^T E)]X^\top = E'X^\top \tag{16}$$

where the symbol $\circ$ denotes Hadamard (or entry-wise) product. In a similar way, we have

$$\nabla_{W_2} L(w) = E(((W_1X)^m)^\top). \tag{17}$$

Open: under what conditions, GD/SGD may find minimal norm global optima?

Summary

- For overparametric multilinear networks (linear models) with square loss, for appropriate initial conditions, GD provides implicit regularization by evolving in a restricted strongly convex subspace (the column space of X'). Thus the asymptotic solution is the *minimum norm* global minima.

- This is expected to be extended to multilayer neural networks with nonlinear activations, but precise characterization is open.

- What about classification? For loss functions such as cross-entropy, GD on linear networks with separable data converges asymptotically to the max-margin solution with any starting point, while the norm diverges (Srebro et al., 2017). The convergence is very slow and only logarithmic in the convergence of the loss itself.

Binary Classifications

1. For separable case, GD search logistic/exponential loss toward the infinite global optimizer convergent to max-margin classifier
2. For non-separable case, over-parametric multi-layer neural networks may make it separable

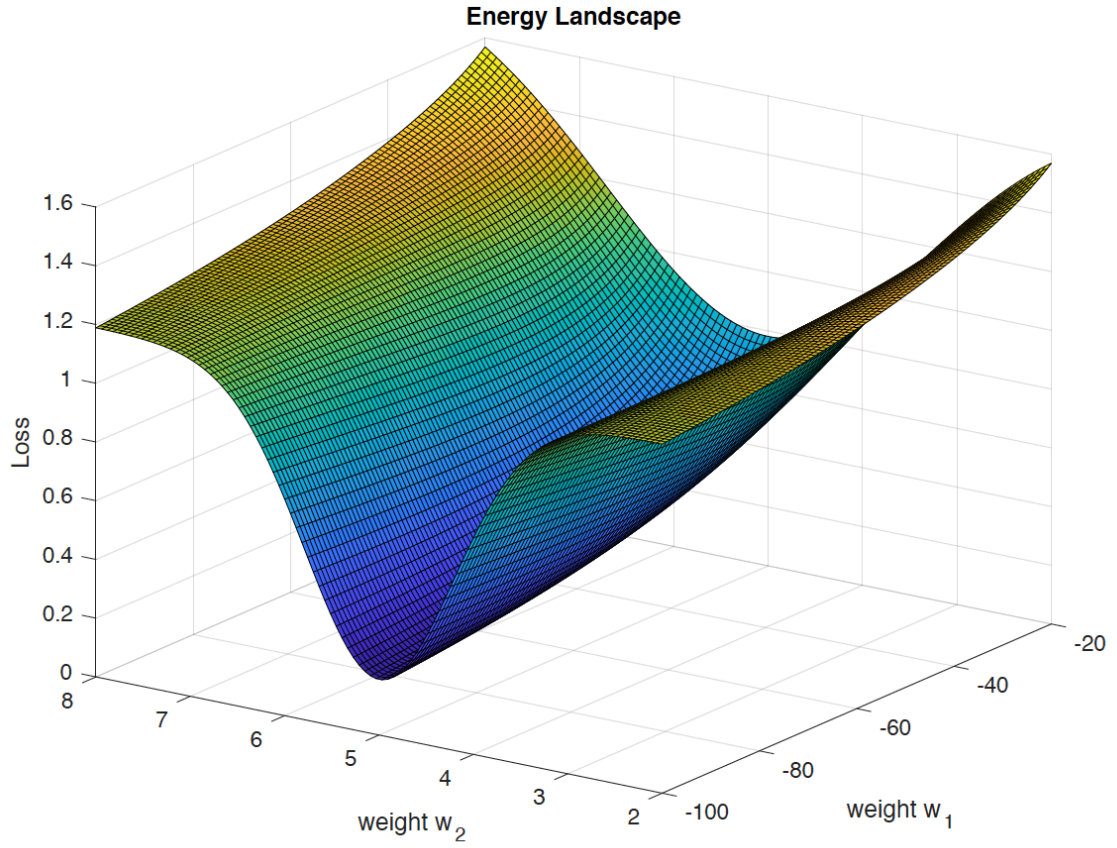
Binary Classification Problem

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

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

Cross-entropy Loss Landscape in Binary Classifications

Cross entropy loss



The Perceptron Algorithm

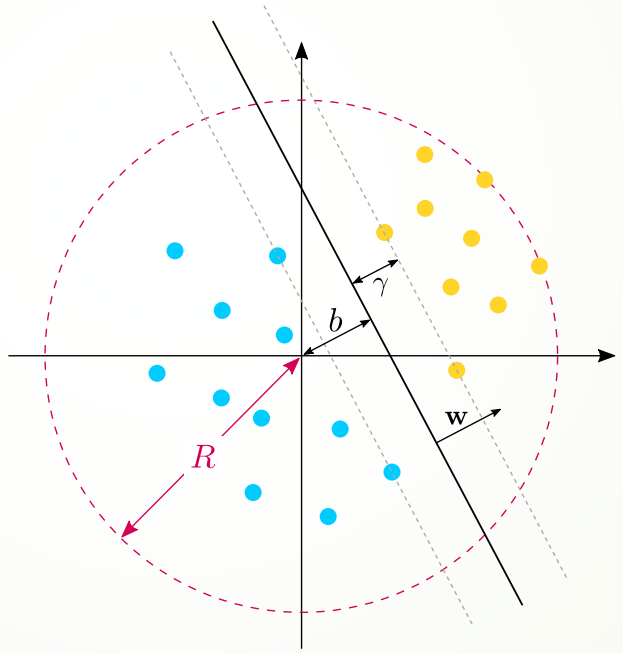
$$\ell(w) = -\sum_{i \in \mathcal{M}_w} y_i \langle w, \mathbf{x}_i \rangle, \quad \mathcal{M}_w = \{i : y_i \langle \mathbf{x}_i, w \rangle < 0, y_i \in \{-1, 1\}\}.$$

The Perceptron Algorithm is a *Stochastic Gradient Descent* method:

$$\begin{aligned} w_{t+1} &= w_t - \eta_t \nabla_i \ell(w) \\ &= \begin{cases} w_t - \eta_t y_i \mathbf{x}_i, & \text{if } y_i w_t^T \mathbf{x}_i < 0, \\ w_t, & \text{otherwise.} \end{cases} \end{aligned}$$

Input ball: $R = \max_i \|\mathbf{x}_i\|.$

Margin: $\gamma = \min_i t_i \mathbf{w}^{\mathsf{T}} \mathbf{x}_i .$ $t_i = y_i$



The perceptron convergence theorem was proved by Block (1962) and Novikoff (1962). The following version is based on that in Cristianini and Shawe-Taylor (2000).

Finiteness of Stopping Time

Theorem 1 (Block, Novikoff). *Let the training set* $S = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_n, t_n)\}$ *be contained in a sphere of radius* R *about the origin. Assume the dataset to be linearly separable, and let* $\mathbf{w}_{\text{opt}}$, $\|\mathbf{w}_{\text{opt}}\| = 1$, *define the hyperplane separating the samples, having functional margin* $\gamma > 0$. *We initialise the normal vector as* $\mathbf{w}_0 = \mathbf{0}$. *The number of updates,* k, *of the perceptron algorithms is then bounded by*

$$k \leq \left(\frac{2R}{\gamma}\right)^2 . \tag{10}$$

Proof.

Proof. Though the proof can be done using the augmented normal vector and samples defined in the beginning, the notation will be a lot easier if we introduce a different augmentation: $\hat{\mathbf{w}} = (\mathbf{w}^\mathsf{T}, b/R)^\mathsf{T} = (w_1, \ldots, w_D, b/R)^\mathsf{T}$ and $\hat{\mathbf{x}} = (\mathbf{x}^\mathsf{T}, R)^\mathsf{T} = (x_1, \ldots, x_D, R)^\mathsf{T}$.

Proof (continued)

We first derive an upper bound on how fast the normal vector grows. As the hyperplane is unchanged if we multiply $\hat{\mathbf{w}}$ by a constant, we can set $\eta = 1$ without loss of generality. Let $\hat{\mathbf{w}}_{k+1}$ be the updated (augmented) normal vector after the kth error has been observed.

$$\|\hat{\mathbf{w}}_{k+1}\|^2 = (\hat{\mathbf{w}}_k + t_i\hat{\mathbf{x}}_i)^\mathsf{T}(\hat{\mathbf{w}}_k + t_i\hat{\mathbf{x}}_i) \tag{11}$$

$$= \hat{\mathbf{w}}_k^\mathsf{T}\hat{\mathbf{w}}_k + \hat{\mathbf{x}}_i^\mathsf{T}\hat{\mathbf{x}}_i + 2t_i\hat{\mathbf{w}}_k^\mathsf{T}\hat{\mathbf{x}}_i \tag{12}$$

$$= \|\hat{\mathbf{w}}_k\|^2 + \|\hat{\mathbf{x}}_i\|^2 + 2t_i\hat{\mathbf{w}}_k^\mathsf{T}\hat{\mathbf{x}}_i\,. \tag{13}$$

Since an update was triggered, we know that $t_i\hat{\mathbf{w}}_k^\mathsf{T}\hat{\mathbf{x}}_i \leq 0$, thus

$$\|\hat{\mathbf{w}}_k\|^2 + \|\hat{\mathbf{x}}_i\|^2 + 2t_i\hat{\mathbf{w}}_k^\mathsf{T}\hat{\mathbf{x}}_i \leq \|\hat{\mathbf{w}}_k\|^2 + \|\hat{\mathbf{x}}_i\|^2 \tag{14}$$

$$= \|\hat{\mathbf{w}}_k\|^2 + (\|\mathbf{x}_i\|^2 + R^2) \tag{15}$$

$$\leq \|\hat{\mathbf{w}}_k\|^2 + 2R^2\,. \tag{16}$$

This implies that $\|\hat{\mathbf{w}}_k\|^2 \leq 2kR^2$, thus

$$\|\hat{\mathbf{w}}_{k+1}\|^2 \leq 2(k+1)R^2\,. \tag{17}$$

Proof (continued)

We then proceed to show how the inner product between an update of the normal vector and $\hat{\mathbf{w}}_{\text{opt}}$ increase with each update:

$$\hat{\mathbf{w}}_{\text{opt}}^{\mathsf{T}}\hat{\mathbf{w}}_{k+1} = \hat{\mathbf{w}}_{\text{opt}}^{\mathsf{T}}\hat{\mathbf{w}}_k + t_i\hat{\mathbf{w}}_{\text{opt}}^{\mathsf{T}}\hat{\mathbf{x}}_i \tag{18}$$

$$\geq \hat{\mathbf{w}}_{\text{opt}}^{\mathsf{T}}\hat{\mathbf{w}}_k + \gamma \tag{19}$$

$$\geq (k+1)\gamma\,, \tag{20}$$

since $\hat{\mathbf{w}}_{\text{opt}}^{\mathsf{T}}\hat{\mathbf{w}}_k \geq k\gamma$. We therefore have

$$k^2\gamma^2 \leq (\hat{\mathbf{w}}_{\text{opt}}^{\mathsf{T}}\hat{\mathbf{w}}_k)^2 \leq \|\hat{\mathbf{w}}_{\text{opt}}\|^2\|\hat{\mathbf{w}}_k\|^2 \leq 2kR^2\|\hat{\mathbf{w}}_{\text{opt}}\|^2\,, \tag{21}$$

where we have made use of the Cauchy-Schwarz inequality. As $k^2\gamma^2$ grows faster than $2kR^2$, Eq. (21) can hold if and only if

$$k \leq 2\|\hat{\mathbf{w}}_{\text{opt}}\|^2\frac{R^2}{\gamma^2}\,. \tag{22}$$

Proof (continued)

As $b \leq R$, we can rewrite the norm of the normal vector:

$$\|\hat{\mathbf{w}}_{\text{opt}}\|^2 = \|\mathbf{w}_{\text{opt}}\|^2 + \frac{b^2}{R^2} \leq \|\mathbf{w}_{\text{opt}}\|^2 + 1 = 2. \tag{23}$$

The bound on k now becomes

$$k \leq 4\frac{R^2}{\gamma^2} = \left(\frac{2R}{\gamma}\right)^2, \tag{24}$$

which therefore bounds the number of updates necessary to find the separating hyperplane. □

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 $\mathbf{X} \in \mathbb{R}^{d\times N}$.

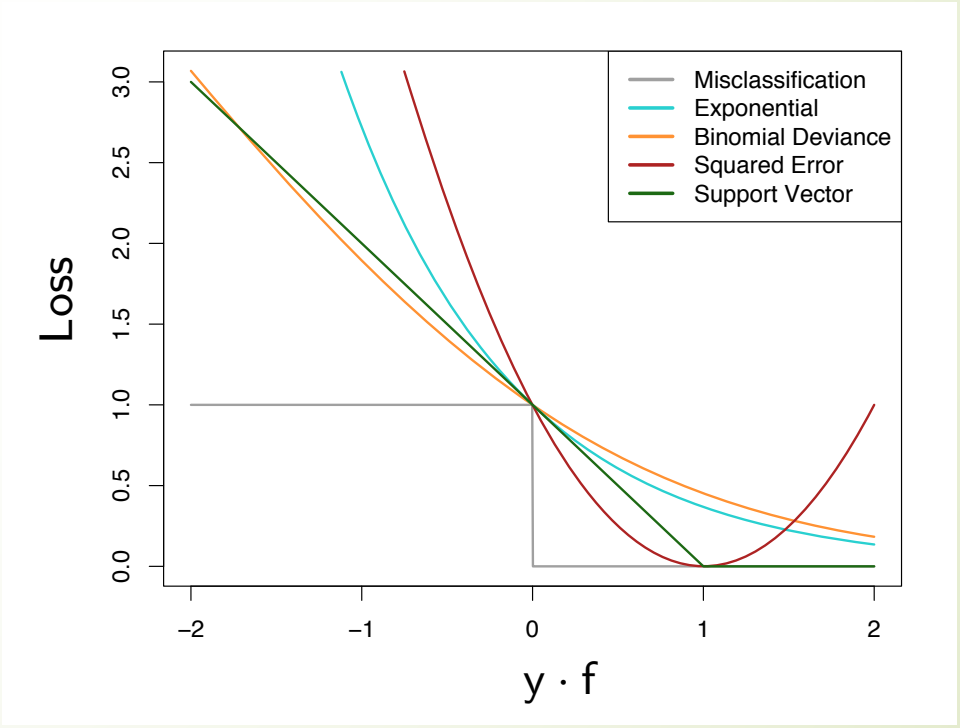
Binomial Deviance/Cross-Entropy/Logistic loss:

$$H(p,q) = -\sum_i p_i \log q_i = -y \log \hat{y} - (1-y)\log(1-\hat{y})$$

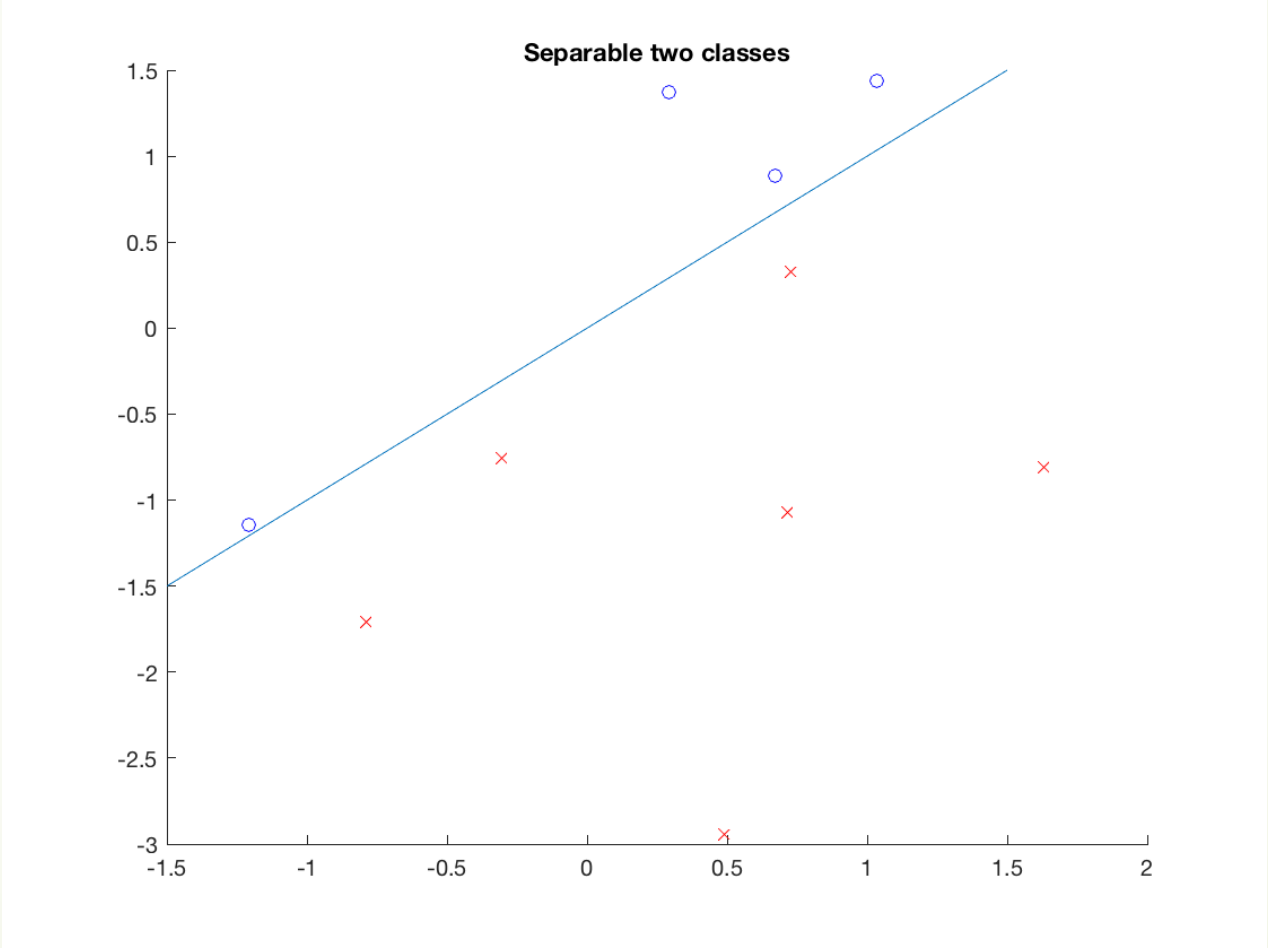
$$\Leftrightarrow \ell(u) = \log\left(1 + e^{-u}\right)$$

Exponential loss:

$$\ell(u) = e^{-u}$$



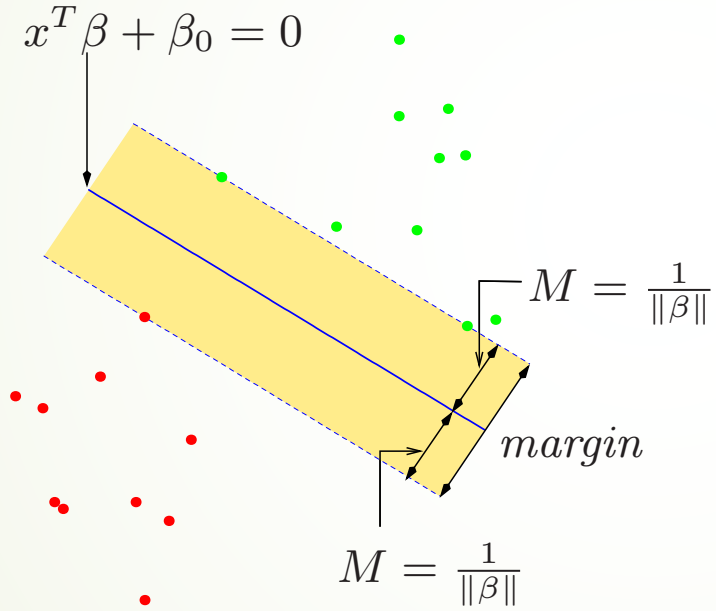
Separable Classification

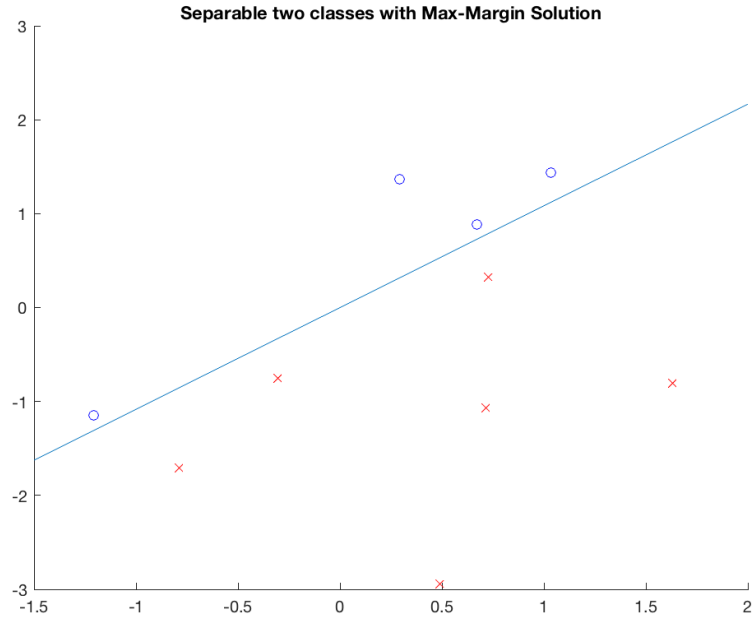


Max-Margin Classifier (SVM)

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

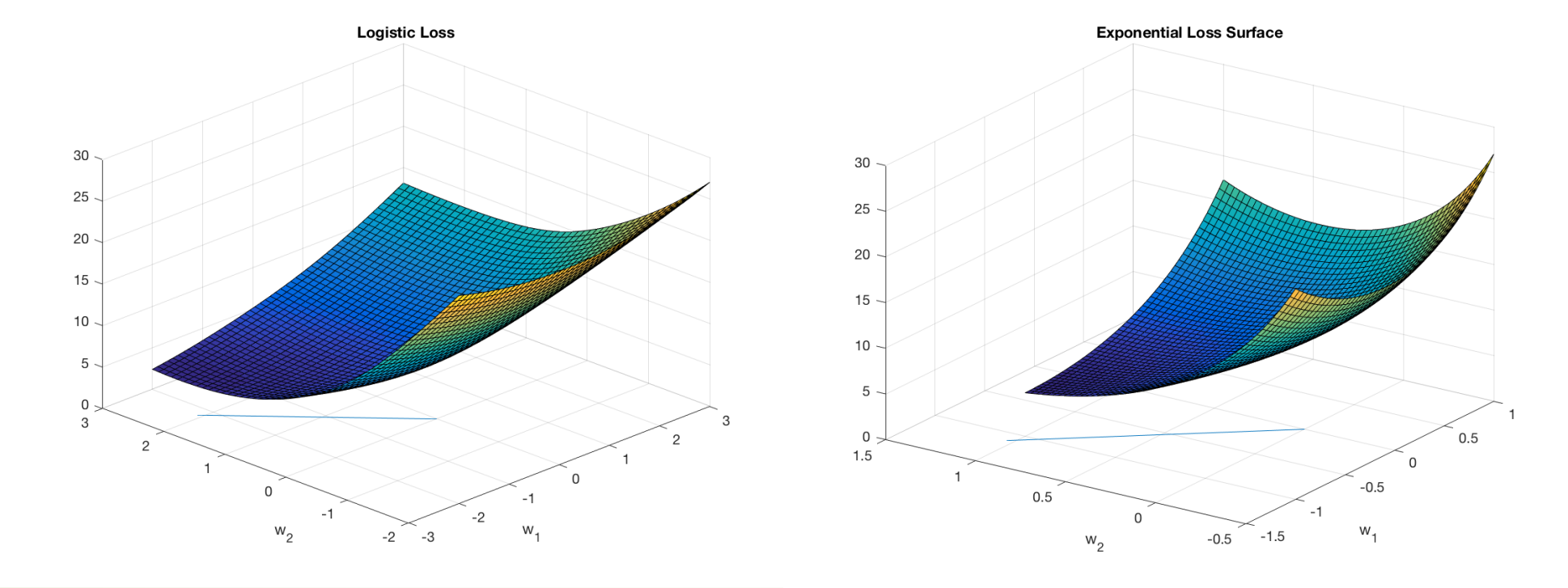
$$\text{subject to } y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq 1 \text{ for all } i$$





Logistic Regression converges at infinity to max-margin classifier for separable problems

Landscape of Logistic/Exponential Loss



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

Lemma 1 *Let* $\mathbf{w}(t)$ *be the iterates of gradient descent (eq. 2) with* $\eta < 2\beta^{-1}\sigma_{\max}^{-2}(\mathbf{X})$ *and any starting point* $\mathbf{w}(0)$. *Under Assumptions 1 and 2, we have: (1)* $\lim_{t\to\infty}\mathcal{L}(\mathbf{w}(t)) = 0$, *(2)* $\lim_{t\to\infty}\|\mathbf{w}(t)\| = \infty$, *and (3)* $\forall n : \lim_{t\to\infty}\mathbf{w}(t)^\top \mathbf{x}_n = \infty$.

Proof Since the data is linearly separable, $\exists \mathbf{w}_*$ which linearly separates the data, and therefore

$$\mathbf{w}_*^\top \nabla\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N} \ell'\left(\mathbf{w}^\top \mathbf{x}_n\right) \mathbf{w}_*^\top \mathbf{x}_n.$$

For any finite $\mathbf{w}$, this sum cannot be equal to zero, as a sum of negative terms, since $\forall n : \mathbf{w}_*^\top \mathbf{x}_n > 0$ and $\forall u : \ell'(u) < 0$. Therefore, there are no finite critical points $\mathbf{w}$, for which $\nabla\mathcal{L}(\mathbf{w}) = \mathbf{0}$. But gradient descent on a smooth loss with an appropriate stepsize is always guaranteed to converge to a critical point: $\nabla\mathcal{L}(\mathbf{w}(t)) \to \mathbf{0}$ (see, *e.g.* Lemma 10 in Appendix A.4, slightly adapted from Ganti (2015), Theorem 2). This necessarily implies that $\|\mathbf{w}(t)\| \to \infty$ while $\forall n : \mathbf{w}(t)^\top \mathbf{x}_n > 0$ for large enough t—since only then $\ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right) \to 0$. Therefore, $\mathcal{L}(\mathbf{w}) \to 0$, so GD converges to the global minimum. ■

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*

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

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.

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.

Proof Sketch We first understand intuitively why an exponential tail of the loss entail asymptotic convergence to the max margin vector: Assume for simplicity that $\ell(u) = e^{-u}$ exactly, and examine the asymptotic regime of gradient descent in which $\forall n : \mathbf{w}(t)^\top \mathbf{x}_n \to \infty$, as is guaranteed by Lemma 1. If $\mathbf{w}(t) / \|\mathbf{w}(t)\|$ converges to some limit $\mathbf{w}_\infty$, then we can write $\mathbf{w}(t) = g(t)\,\mathbf{w}_\infty + \boldsymbol{\rho}(t)$ such that $g(t) \to \infty$, $\forall n : \mathbf{x}_n^\top \mathbf{w}_\infty > 0$, and $\lim_{t\to\infty} \boldsymbol{\rho}(t)/g(t) = 0$. The gradient can then be written as:

$$-\nabla \mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N} \exp\left(-\mathbf{w}(t)^\top \mathbf{x}_n\right) \mathbf{x}_n = \sum_{n=1}^{N} \exp\left(-g(t)\,\mathbf{w}_\infty^\top \mathbf{x}_n\right) \exp\left(-\boldsymbol{\rho}(t)^\top \mathbf{x}_n\right) \mathbf{x}_n\,. \tag{5}$$

As $g(t) \to \infty$ and the exponents become more negative, only those samples with the largest (*i.e.*, least negative) exponents will contribute to the gradient. These are precisely the samples with the smallest margin $\operatorname{argmin}_n \mathbf{w}_\infty^\top \mathbf{x}_n$, aka the "support vectors". The negative gradient (eq. 5) would then asymptotically become a non-negative linear combination of support vectors. The limit $\mathbf{w}_\infty$ will then be dominated by these gradients, since any initial conditions become negligible as $\|\mathbf{w}(t)\| \to \infty$ (from Lemma 1). Therefore, $\mathbf{w}_\infty$ will also be non-negative linear combination of support vectors, and so will its scaling $\hat{\mathbf{w}} = \mathbf{w}_\infty / \left(\min_n \mathbf{w}_\infty^\top \mathbf{x}_n\right)$. We therefore have:

$$\hat{\mathbf{w}} = \sum_{n=1}^{N} \alpha_n \mathbf{x}_n \qquad \forall n \ \left(\alpha_n \geq 0 \text{ and } \hat{\mathbf{w}}^\top \mathbf{x}_n = 1\right) \ \text{OR} \ \left(\alpha_n = 0 \text{ and } \hat{\mathbf{w}}^\top \mathbf{x}_n > 1\right) \tag{6}$$

These are precisely the KKT condition for the SVM problem (eq. 4) and we can conclude that $\hat{\mathbf{w}}$ is indeed its solution and $\mathbf{w}_\infty$ is thus proportional to it.

Corollary 8 *We examine a multilayer neural network with component-wise ReLU functions* $f(z) = \max[z, 0]$, *and weights* $\{\mathbf{W}_l\}_{l=1}^{L}$. *Given input* $\mathbf{x}_n$ *and target* $y_n \in \{-1, 1\}$, *the DNN produces a scalar output*

$$u_n = \mathbf{W}_L f\left(\mathbf{W}_{L-1} f\left(\cdots \mathbf{W}_2 f\left(\mathbf{W}_1 \mathbf{x}_n\right)\right)\right)$$

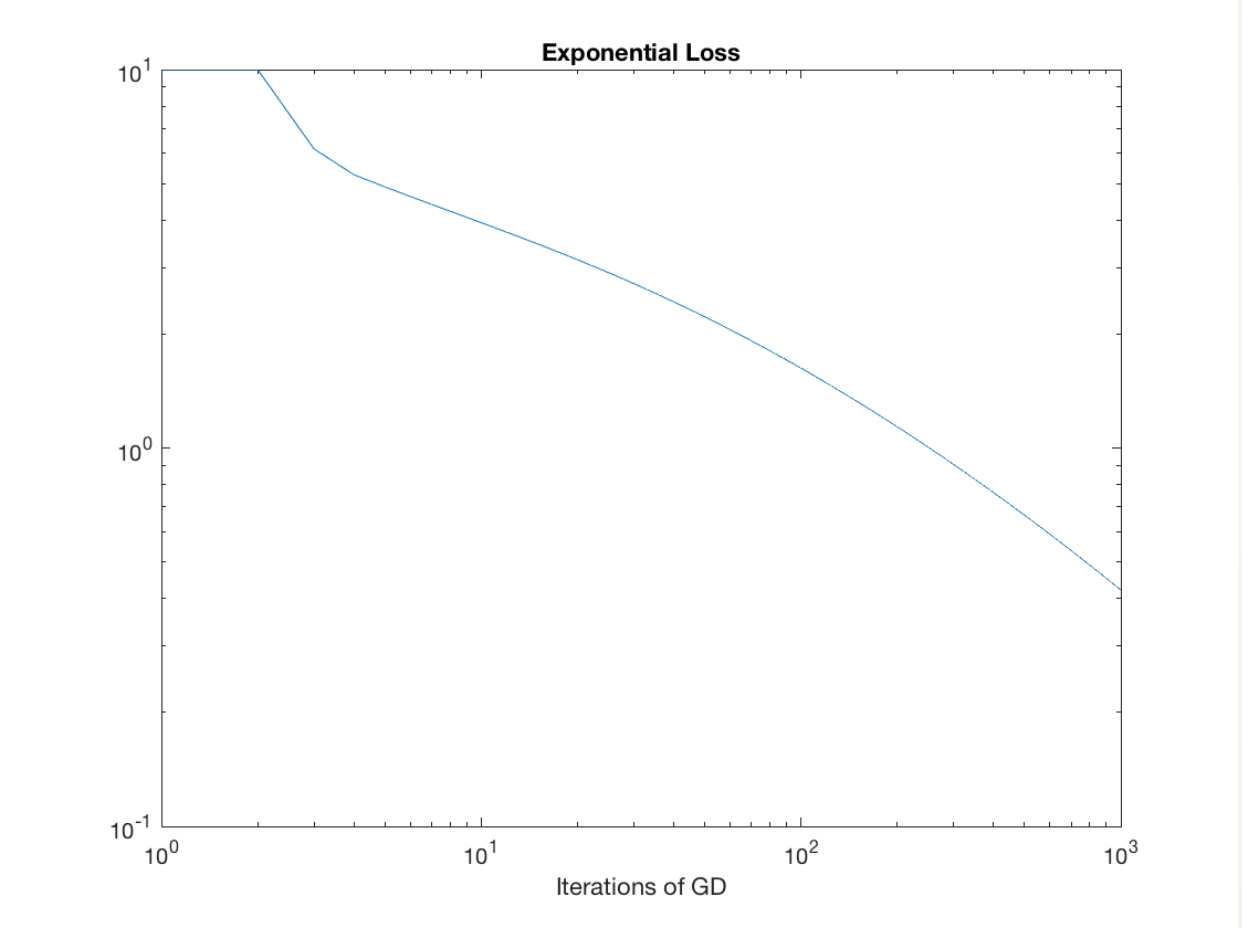
and has loss $\ell(y_n u_n)$, *where* ℓ *obeys assumptions 2 and 3.*

If we optimize a single weight layer $\mathbf{w}_l = \mathrm{vec}\left(\mathbf{W}_l^\top\right)$ *using gradient descent, so that* $\mathcal{L}(\mathbf{w}_l) = \sum_{n=1}^{N} \ell\left(y_n u_n(\mathbf{w}_l)\right)$ *converges to zero, and* $\exists t_0$ *such that* $\forall t > t_0$ *the ReLU inputs do not switch signs, then* $\mathbf{w}_l(t) / \|\mathbf{w}_l(t)\|$ *converges to*

$$\hat{\mathbf{w}}_l = \underset{\mathbf{w}_l}{\operatorname{argmin}} \|\mathbf{w}_l\|^2 \ \text{ s.t. } y_n u_n(\mathbf{w}_l) \geq 1.$$

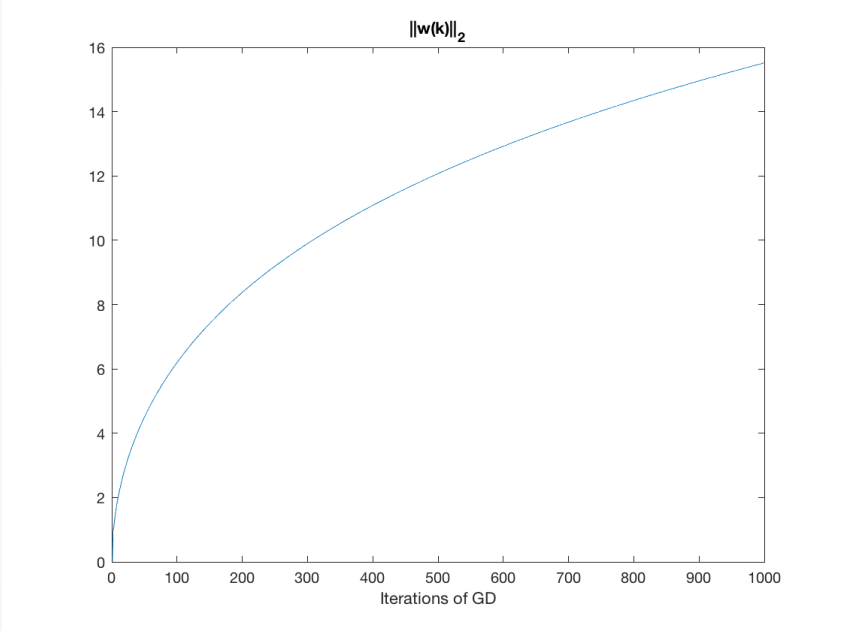
Soudry et al. 2018

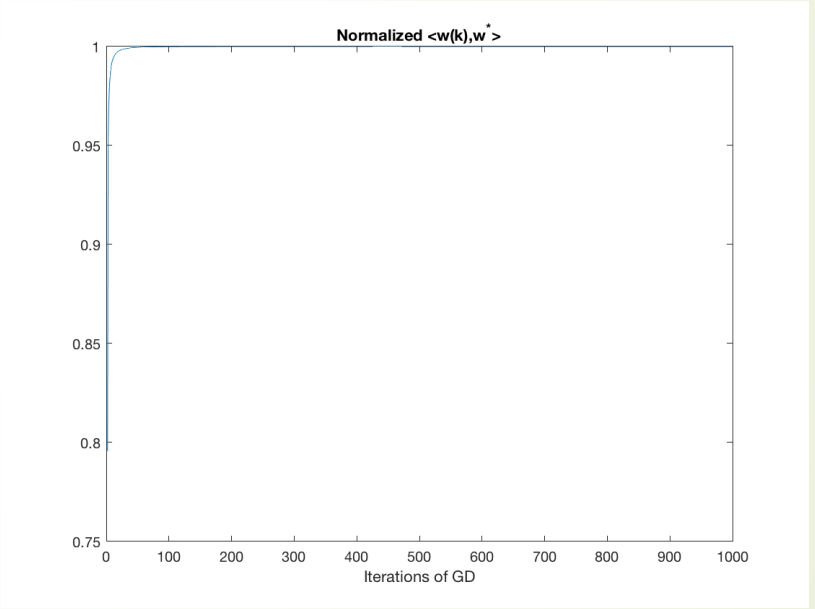
Exponential loss drops at ~1/*k* in GD



Separable Logistic Regression

```
#Max-Margin
cvx_begin
    variables w(2,1);
    minimize(w'*w);
    subject to
        y.*(x'*w)>=1;
cvx_end
```





Left: 2-norm grows ~ log *k*; Right: angle converges to max-margin classifier

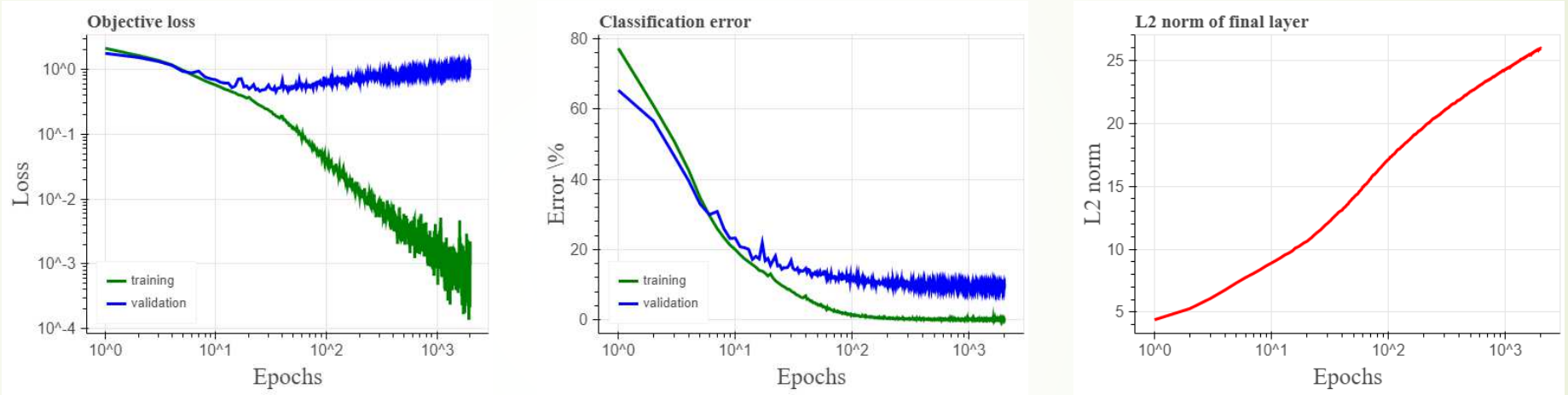
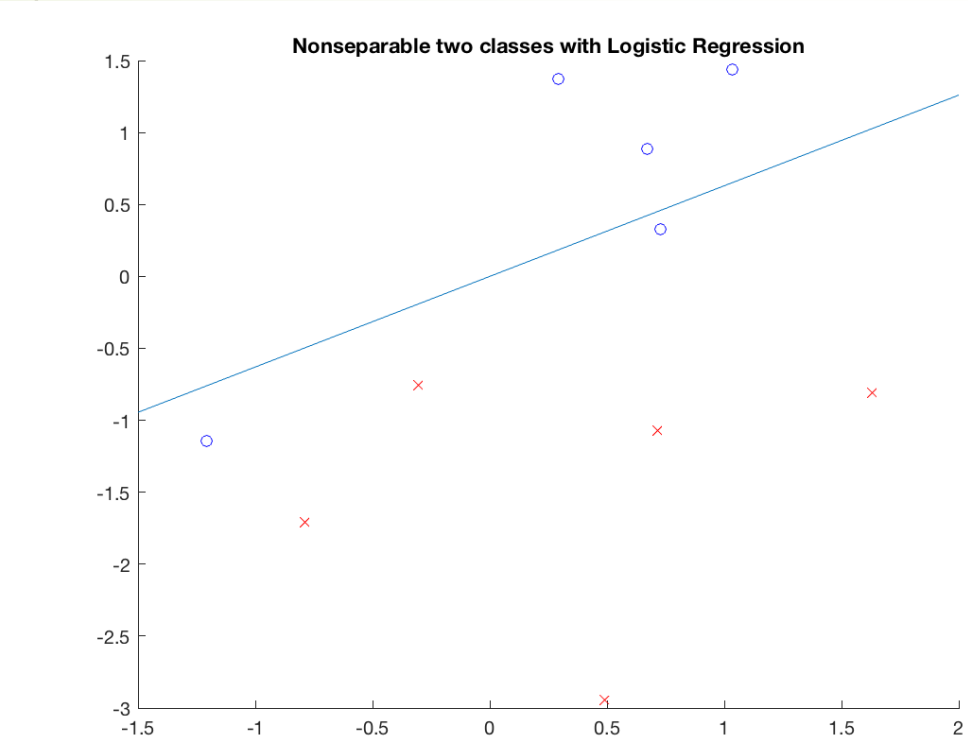


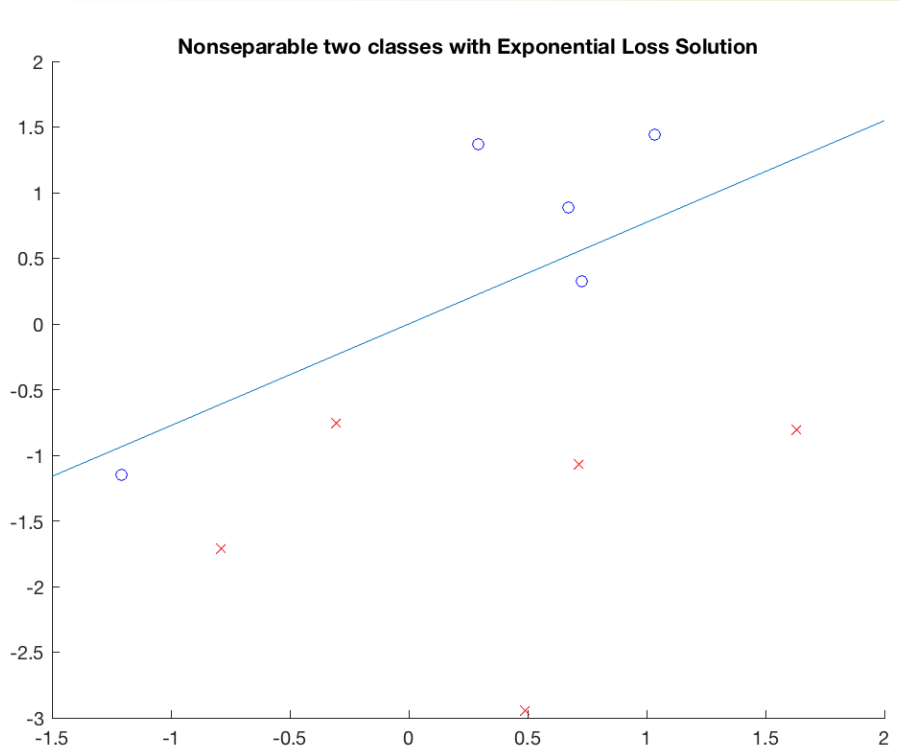
Figure 2: Training of a convolutional neural network on CIFAR10 using stochastic gradient descent with constant learning rate and momentum, softmax output and a cross entropy loss, where we achieve 8.3% final validation error. We observe that, approximately: (1) The training loss decays as a t^{-1}, (2) the L_2 norm of last weight layer increases logarithmically, (3) after a while, the validation loss starts to increase, and (4) in contrast, the validation (classification) error slowly improves.

Soudry et al. 2018

Nonseparable?
Deep networks make it separable in "feature space"

Nonseparable classification?

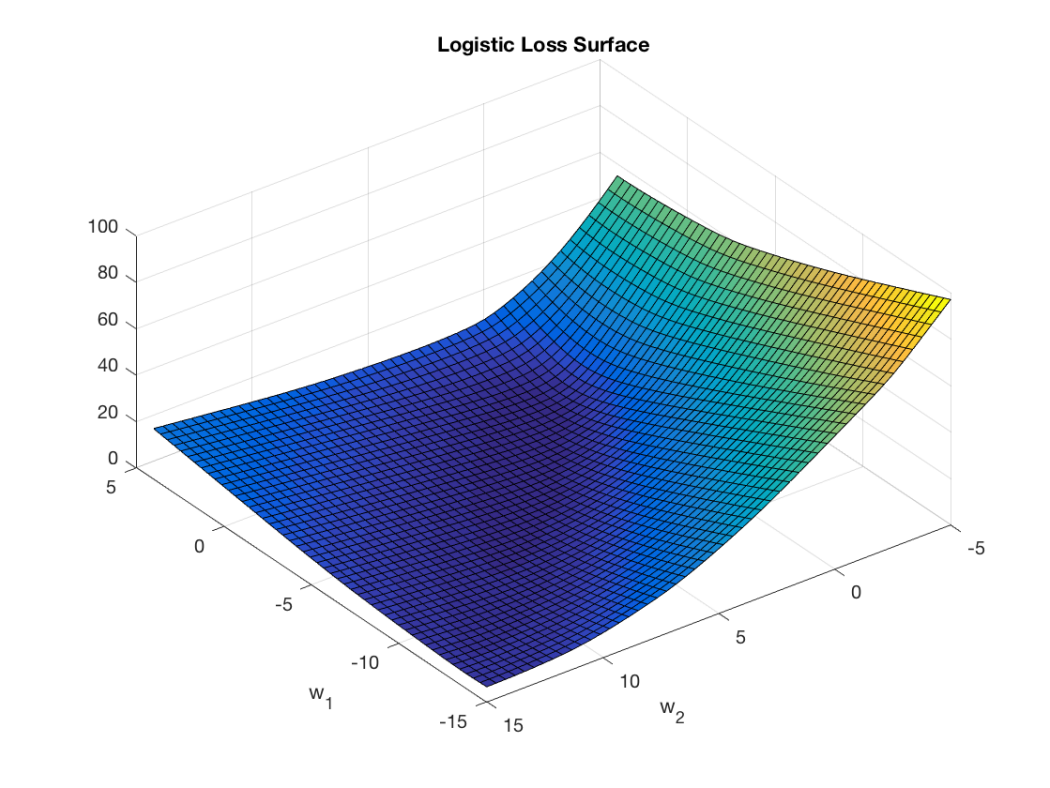


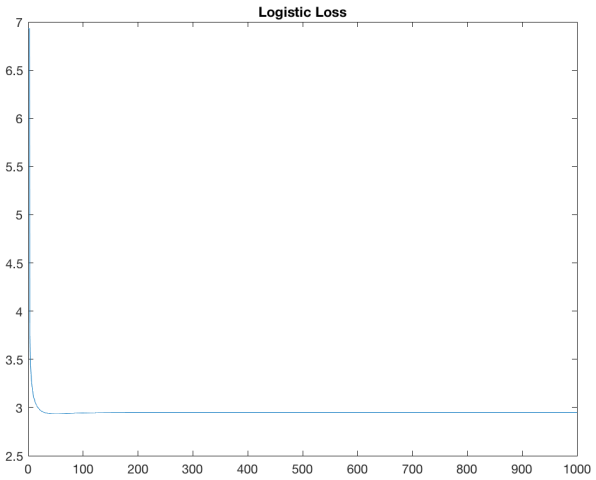


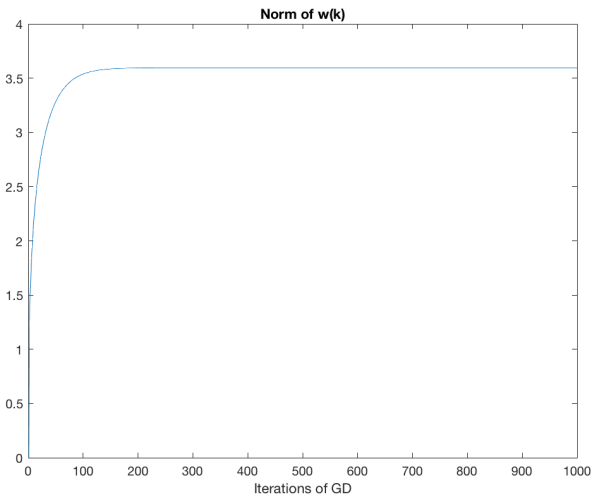
Left: GD solution for logistic loss

Right: GD solution for exponential loss

GD for nonseparable logistic regression may converge to a finite solution

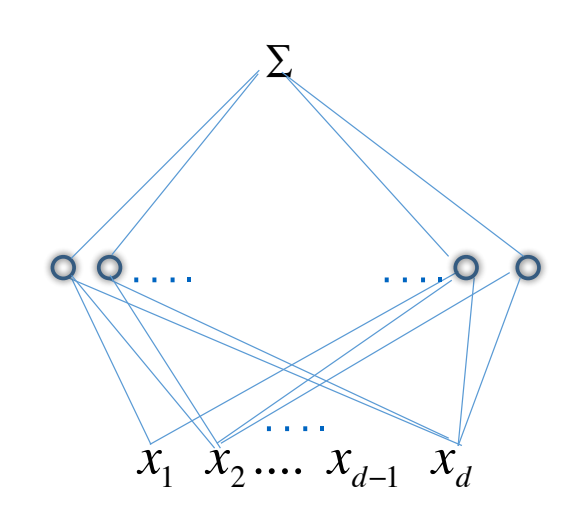






The global optima is finite.

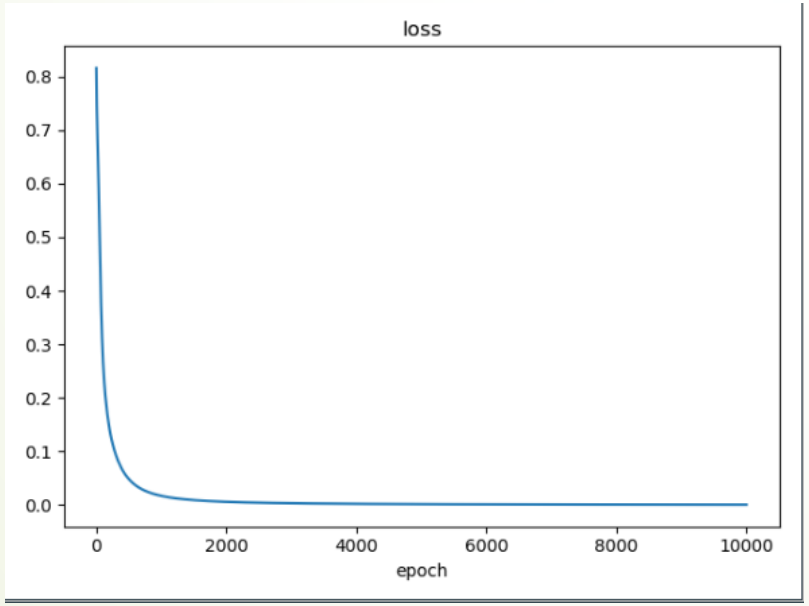
2-Layer Neural Networks

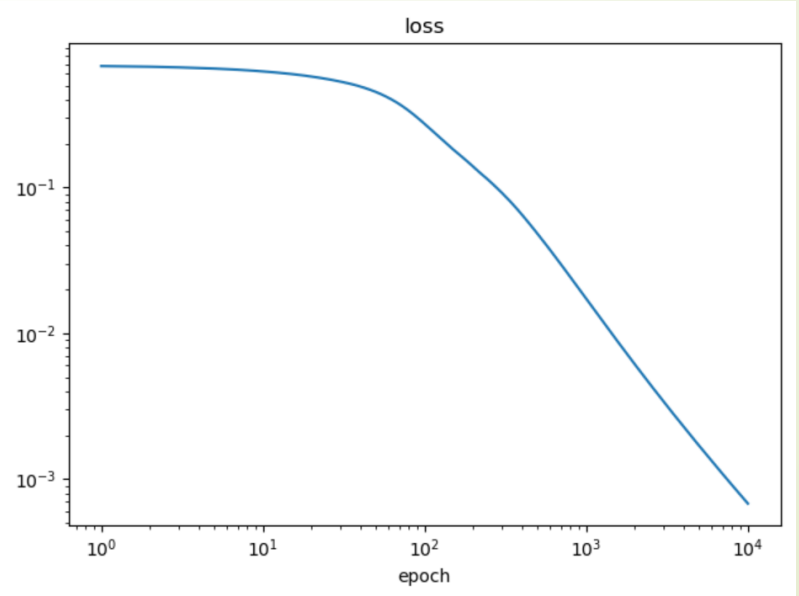


$$f(x) = W_2\sigma(W_1 x)$$

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

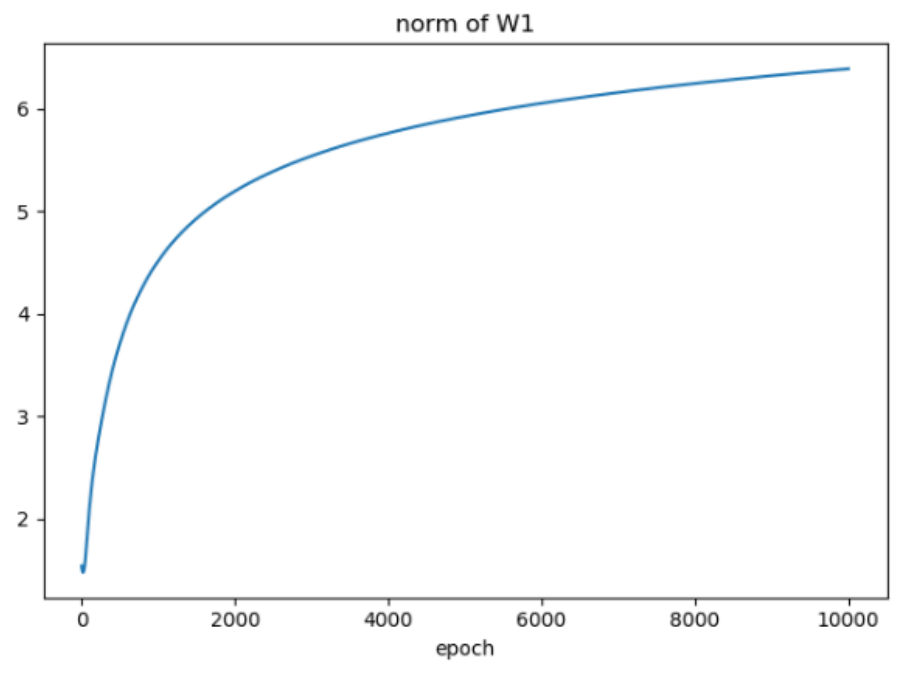
For large q, e.g. q=5, it becomes separable: logistic loss drops down at ~1/k

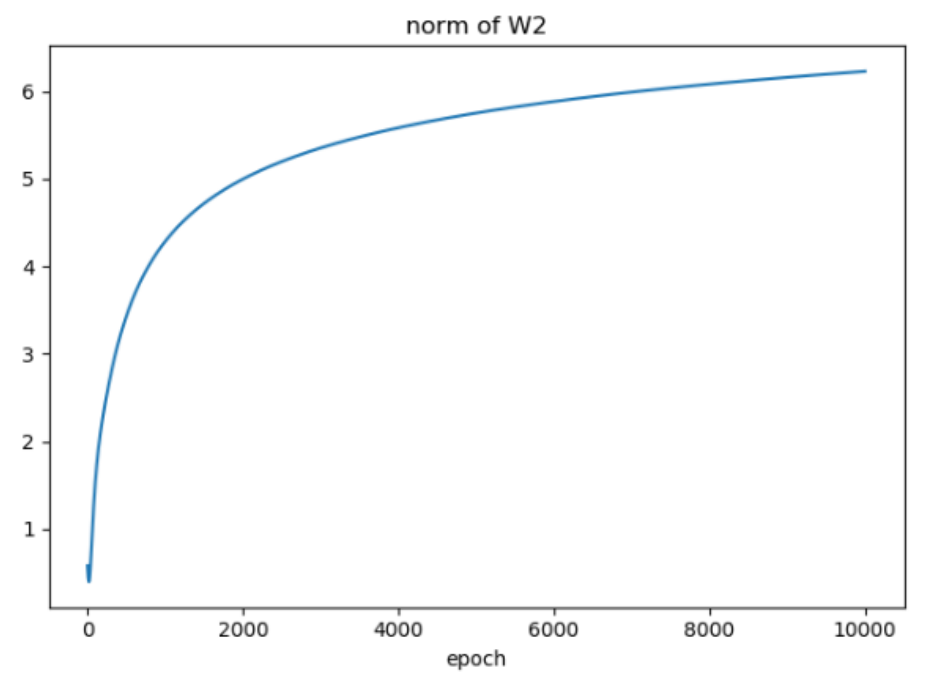




By *Yifei HUANG*

Both W1 and W2 grows to infinity!



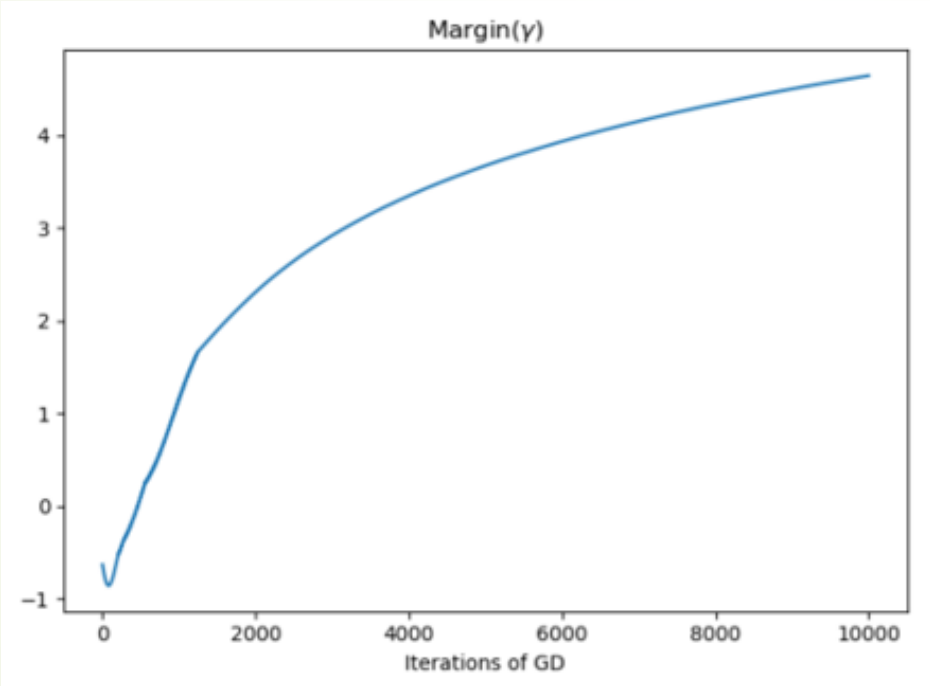


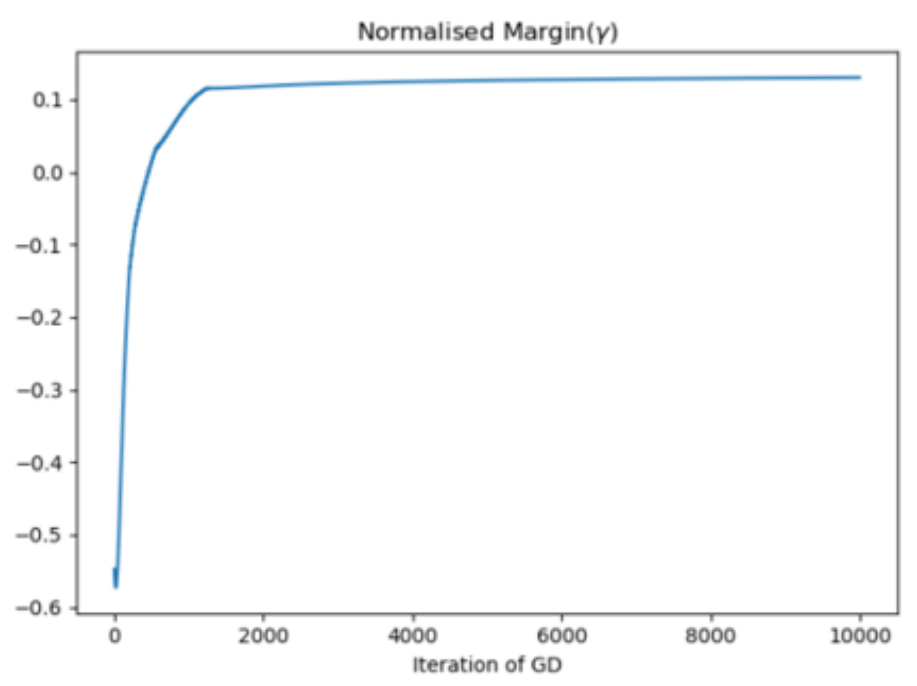
By *Yifei HUANG*

Normalized Margin stabilizes!

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

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





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

By *Yifei HUANG*.

Spectral Complexity of Networks [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 $A_1, \ldots, A_L$

for ResNet (He et al., 2016), it is sensible to set $M_i := I$

for MLP, the simple choice $M_i = 0$ suffices.

The *spectral complexity* $R_{F_{\mathcal{A}}} = R_{\mathcal{A}}$ of a network $F_{\mathcal{A}}$

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

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

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*

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

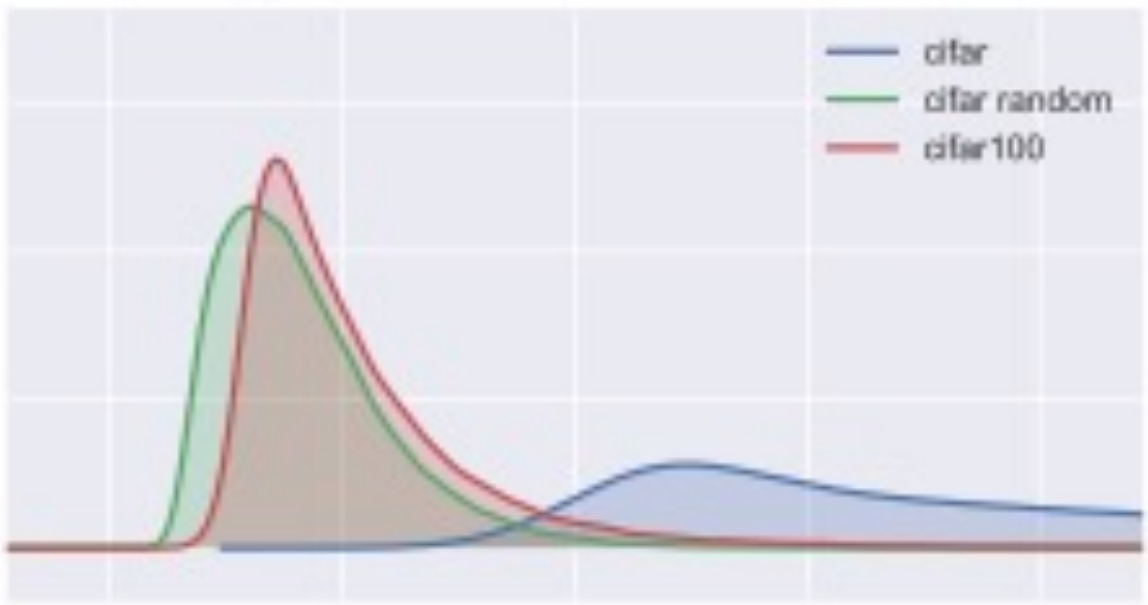
where $\widehat{\mathcal{R}}_\gamma(f) \leq n^{-1}\sum_i \mathbb{1}\left[f(x_i)_{y_i} \leq \gamma + \max_{j \neq y_i} f(x_i)_j\right]$ *and* $\|X\|_2 = \sqrt{\sum_i \|x_i\|_2^2}$.

Bartlett (1997) showed that the generalization error can be bounded by a margin-sensitive fat-shattering dimension, which is in turn bounded by the l1-norm of weights.
"For valid generalization the size of the weights is more important than the size of the network." NIPS 1996.

Normalized Margin Distribution

$$(x, y) \mapsto \frac{F_{\mathcal{A}}(x)_y - \max_{i \neq y} F_{\mathcal{A}}(x)_i}{R_{\mathcal{A}} \|X\|_2 / n},$$

(a) **mnist** is easier than **cifar10**.



(c) **cifar100** is as hard as **cifar10** with random labels!

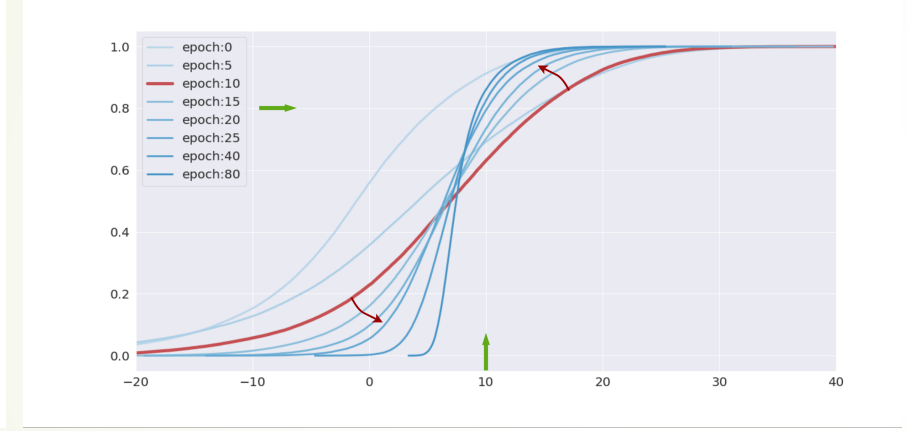
Train 5-layer basic CNN(n) on Cifar10

- 5 convolutional layers:
  - n 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

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 50, 16, 16]           1,400
       BatchNorm2d-2          [-1, 50, 16, 16]             100
              ReLU-3          [-1, 50, 16, 16]               0
            Conv2d-4            [-1, 50, 8, 8]          22,550
       BatchNorm2d-5            [-1, 50, 8, 8]             100
              ReLU-6            [-1, 50, 8, 8]               0
            Conv2d-7            [-1, 50, 4, 4]          22,550
       BatchNorm2d-8            [-1, 50, 4, 4]             100
              ReLU-9            [-1, 50, 4, 4]               0
           Conv2d-10            [-1, 50, 2, 2]          22,550
      BatchNorm2d-11            [-1, 50, 2, 2]             100
             ReLU-12            [-1, 50, 2, 2]               0
           Conv2d-13            [-1, 50, 1, 1]          22,550
      BatchNorm2d-14            [-1, 50, 1, 1]             100
             ReLU-15            [-1, 50, 1, 1]               0
           Linear-16                  [-1, 10]             510
================================================================
Total params: 92,610
Trainable params: 92,610
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 0.39
Params size (MB): 0.35
Estimated Total Size (MB): 0.76
----------------------------------------------------------------
```

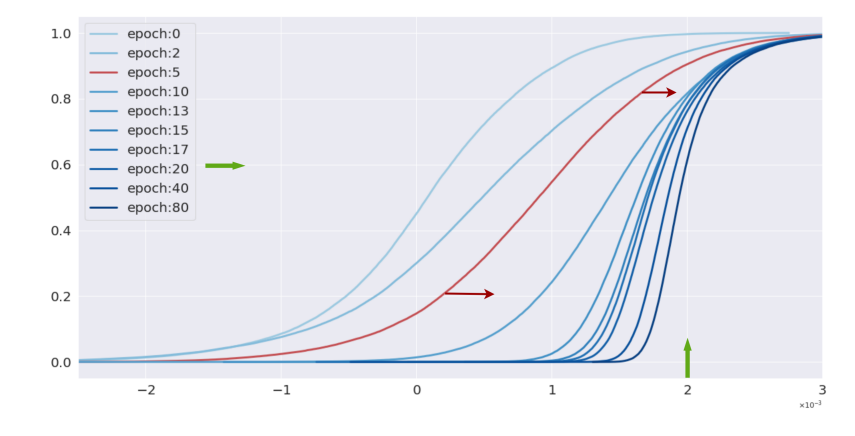
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 400, 16, 16]          11,200
       BatchNorm2d-2          [-1, 400, 16, 16]             800
              ReLU-3          [-1, 400, 16, 16]               0
            Conv2d-4            [-1, 400, 8, 8]       1,440,400
       BatchNorm2d-5            [-1, 400, 8, 8]             800
              ReLU-6            [-1, 400, 8, 8]               0
            Conv2d-7            [-1, 400, 4, 4]       1,440,400
       BatchNorm2d-8            [-1, 400, 4, 4]             800
              ReLU-9            [-1, 400, 4, 4]               0
           Conv2d-10            [-1, 400, 2, 2]       1,440,400
      BatchNorm2d-11            [-1, 400, 2, 2]             800
             ReLU-12            [-1, 400, 2, 2]               0
           Conv2d-13            [-1, 400, 1, 1]       1,440,400
      BatchNorm2d-14            [-1, 400, 1, 1]             800
             ReLU-15            [-1, 400, 1, 1]               0
           Linear-16                  [-1, 10]           4,010
================================================================
Total params: 5,780,810
Trainable params: 5,780,810
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 3.12
Params size (MB): 22.05
Estimated Total Size (MB): 25.19
----------------------------------------------------------------
```

Breiman's Dillema: Dynamics of Normalized Margin distributions [Zhu-Huang-Y.'18]



Basic CNN(50) (#param=92,610) trained on Cifar10

Basic CNN(400) (#param=5.8M) trained on Cifar10



Breiman's Dilemma (1999): "The evidence is that if we try too hard to make the margins larger, then overfitting sets in. "

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-parametric deep networks may make it separable in ``feature spaces'' and GD converges toward some max-margin solution in infinity
- Spectrally-normalized margin may give a stable measure of generalization ability
- Open problem: Breiman's dilemma shows the limitation of margin-maximization theory, and early stopping regularization is needed in deep neural networks.

Reference

- Poggio, T, Liao, Q, Miranda, B, Rosasco, L, Boix, X, Hidary, J, Mhaskar, H. Theory of Deep Learning III: explaining the non-overfitting puzzle. [MIT CBMM Memo v3, 1/30/2018].
- Yuan Yao, Lorenzo Rosasco and Andrea Caponnetto, On Early Stopping in Gradient Descent Learning, Constructive Approximation, 2007, 26 (2): 289-315.
- Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, Nathan Srebro. The Implicit Bias of Gradient Descent on Separable Data. arXiv:1710.10345
- Peter L. Bartlett, Dylan J. Foster, Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. https://arxiv.org/abs/1706.08498
- Weizhi Zhu, Yifei Huang, Yuan Yao. **On Breiman's Dilemma in Neural Networks: Phase Transitions of Margin Dynamics.** [arXiv:1810.03389]

Thank you!