Dynamical aspects of Deep Learning

Zhenyu Liao, **Yacine Chitour**
Joint work with R. Couillet

L2S, CentraleSupélec
Université Paris-Saclay
Paris, France

February 28, 2019

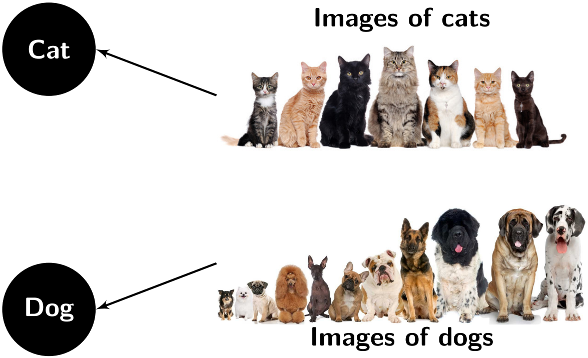
Outline

Motivation: "learn" to automatically classify images

Machine Learning:

- given $m = 2M$ images of cats $x_1^{cat}, x_2^{cat}, \ldots, x_{m/2}^{dog}$ and dogs $x_1^{dog}, x_2^{dog}, \ldots, x_{m/2}^{dog}$ of labels y_{cat} and y_{dog} ($y_{cat} \neq y_{dog}$), respectively.

(**Labels are chosen by user!!!**)



- **Goal**: for a new image $x_{new}^{?}$ with $? \in \{cat, dog\}$, predict ?=cat or ?=dog.

How to "learn" to classify?

Learning phase: find $\mathbf{W}$ that minimizes $\sum_{i,j} \|y_{cat} - \mathbf{W} \cdot x_i^{cat}\|^2 + \|y_{dog} - \mathbf{W} \cdot x_j^{dog}\|^2$, where $x_i^{cat}, x_j^{dog} \in \mathbb{R}^{d_x}$ and $y_{cat}, y_{dog} \in \mathbb{R}^{d_y}$ **chosen by user**.

- **Input**: training set (X, Y), images $X = [x_1^{cat}, \ldots, x_1^{dog}, \ldots] \in \mathbb{R}^{d_x \times m}$ with associated labels $Y = [y_{cat}, \ldots, y_{dog}, \ldots] \in \mathbb{R}^{d_y \times m}$.
- **Output**: $\mathbf{W}$ that minimizes the difference $\|Y - \mathbf{W} \cdot X\|^2$

Here $\mathbf{W} \cdot X$ is a **PROCEDURE**, e.g., $\mathbf{W} \cdot X = WX$, with $W \in \mathbb{R}^{d_y \times d_x}$.

Prediction phase: for a new image $x_{new}^?$ (of unknown true label $y_{new}^?$), predicts:

- $x_{new}^?$ to be a cat ($y_{new}^? = y_{cat}$) if $\|y_{cat} - \mathbf{W} \cdot x_{new}^?\| < \|y_{dog} - \mathbf{W} \cdot x_{new}^?\|$
- $x_{new}^?$ to be a dog ($y_{new}^? = y_{dog}$) otherwise

Objective: given (X,Y), find $\mathbf{W}$ that minimizes the difference $\|Y - \mathbf{W}\cdot X\|^2$.

$\Rightarrow$"Best" solution: if $\mathbf{W}\cdot X = WX$, linear regression $W_{LR} = YX^\mathsf{T}(XX^\mathsf{T})^{-1}$ if XX^T invertible. However,

- linear regression may easily **overfit**: "learned" W **too "adapted"** to the given pair (X,Y) and $\|y^?_{new} - W_{LR}x^?_{new}\|$ large if $x^?_{new} \notin X$, i.e.,

$$\|y^?_{new} - W_{LR}x^?_{new}\|^2 \gg \frac{1}{m}\sum_{i=1}^{m}\|y_i - W_{LR}x_i\|^2 = \frac{1}{m}\|Y - W_{LR}X\|^2$$

- does not work well for difficult problems (e.g., cat & dogs classification, face recognition, etc): describe solely a linear transformation between X and Y

From Linear Regression to Deep Neural Networks

⇒(Brain-inspired) LINEAR neural network models (back to [Rosenblatt, 1958])

$\Longrightarrow$

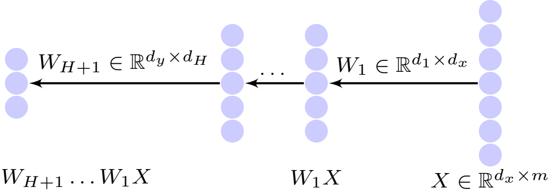


Figure: Illustration of H-hidden-layer linear neural network

Linear deep learning (LDL) : $\boxed{W_{LDL} = W_{H+1} W_H \cdots W_1}$

Numerical tests show that linear deep learning also overfits.
Reason: algorithms based on linear deep learning essentially provide $\mathbf{W_{LDL}} = \mathbf{W_{LR}}$.

- NONLINEAR neural networks:

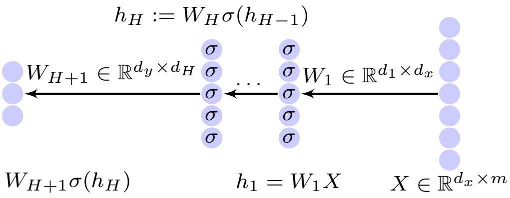
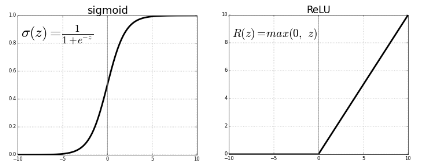


Figure: Illustration of H-hidden-layer nonlinear neural network

with (nonlinear) *activation function* $\sigma(z)$: $\mathrm{ReLU}(z) = \max(z, 0)$, Leaky ReLU $\max(z, az)$ $(a > 0)$ or sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$, $\arctan$, $\tanh, \dots$.

$$\boxed{\mathbf{W} \cdot X = W_{H+1}\sigma(W_H \sigma(W_{H-1}\sigma(\cdots W_1 X))).}$$

$\Longrightarrow$



Practitioners find “deeper” structures brings better performance, e.g., for (simple) handwritten digits classification:

Figure: Samples from the MNIST dataset [LeCun et al. 1998].

Network	Classification error rate
$H = 0$ (linear regression)	12.0%
$H = 2$ [LeCun et al. 1998]	2.5%
$H = 4$ [LeCun et al. 1998]	0.8%

Table: Evolution of state of the art on MNIST dataset.

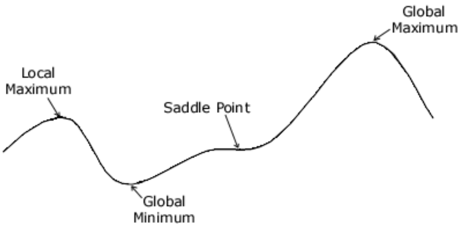
However, deep networks are computationally more challenging!

So, what is the difficulty?

1. huge demand of computational resources: [LeCun et al. 1998] 5-layer of $60K$ parameters to [He et al. 2015] 152-layer of $60M$ parameters
2. ONLY possible to use first-order optimization methods due to complexity constraints, typically with (stochastic) gradient decent
3. unfortunately non-convex optimization problem: for example in a single-layer linear network, use gradient descent to find (W_1, W_2) that minimizes

$$F(W_1, W_2) = \|Y - W_2 W_1 X\|_F^2$$

   clearly, $F(W_1^*, W_2^*) = F(\alpha W_1^*, \frac{1}{\alpha} W_2^*)$ so $(\alpha W_1^*, \frac{1}{\alpha} W_2^*)$ is as "good" as (W_1^*, W_2^*)!
4. even worse, there may be local minima, saddle points and even maxima! All depend on (X, Y) and the design of network.



Non-convexity in Deep Neural Networks

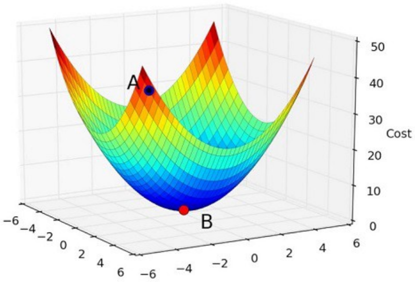


Figure: Convex landscape

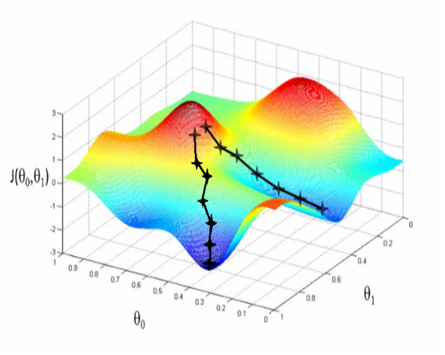


Figure: Non-convex landscape

In non-convex case, the performance of gradient descent can be very sensitive to initialization!

So, can we still obtain some general results in this difficult case?

On LINEAR Deep Neural Networks

Set $d_{H+1} := d_y$, $d_0 := d_x$ and consider

$$X \in \mathbb{R}^{d_0 \times m} \quad Y \in \mathbb{R}^{d_{H+1} \times m}.$$

Goal: find $\mathbf{W} = (W_{H+1}, \cdots, W_1)$ that minimizes the function (depending on (X, Y) !!)

$$F(\mathbf{W}) := \|Y - WX\|^2, \quad W = W_{H+1} W_H \cdots W_1,$$

where

$$W_j \in \mathbb{R}^{d_j \times d_{j-1}}, \quad 1 \leq j \leq H+1.$$

Define state space $\mathcal{W}$ (recall $d_y = d_{H+1}$ and $d_0 = d_x$)

$$\mathcal{W} = \mathbb{R}^{d_{H+1} \times d_H} \times \cdots \mathbb{R}^{d_1 \times d_0}.$$

and **Gradient Descent associated with** F

$$(GD)_{(X,Y)} \qquad \frac{d\mathbf{W}}{dt} = -\nabla F(\mathbf{W}), \quad \mathbf{W} \in \mathcal{W}.$$

Conjecture ($\Longleftrightarrow$ Overfitting Problem = OVF)

(OVF): *For a.e.* (X, Y) *and* $\mathbf{W}_0 \in \mathcal{W}$, *traj. of* $(GD)_{(X,Y)}$ *starting at* $\mathbf{W}_0$ *CV to a* **GLOBAL** *minimum of* F.

(Usual) working assumptions

$$X, Y \text{ full rank } , m \geq \max(d_i) \geq \min(d_i) = d_y.$$

Up to SVD and computations, can assume

$$X = Id_{d_x} \text{ (i.e. } m = d_x), \quad Y = \begin{pmatrix} D_Y & 0 \end{pmatrix}, \quad D_Y \in \mathbb{R}^{d_y \times d_y} \text{ diagonal } > 0.$$

$\boxed{Notation}$

$$(\Pi W)_i^j = W_j \cdots W_i, \quad 1 \leq i \leq j \leq H+1, \quad M = Y - (\Pi W)_1^{H+1}.$$

Gradient dynamics, $1 \leq j \leq H+1$

$$(GD)_Y \qquad \frac{dW_j}{dt} = (\Pi W)_{j+1}^{H+1} M (\Pi W)_1^{j-1}.$$

$\boxed{Definition}$ Critical points $\nabla F(\mathbf{W}) = 0$

$$\mathrm{Crit}(F) = \{\mathbf{W} = (W_{H+1}, \cdots, W_1) \in \mathcal{W}, \;\; (\Pi W)_{j+1}^{H+1} M (\Pi W)_1^{j-1} = 0\}.$$

Candidates for limit points of trajectories.

Gradient Descent for Linear Neural Networks - Convergence

Theorem (C., Liao, Couillet '18)

Every traj. of $(GD)_Y$ *converges to an element of* $\mathrm{Crit}(F)$.

PROOF

(Obvious but) Key remark: $(GD)_Y$ analytic $\Longrightarrow$ Lojasiewicz's theorem can be used

Proposition (Lojasiewicz 50s')

Every **BOUNDED** *traj. of* **ANALYTIC** *gradient system converges to critical point.*

Proof reduces to show that trajectories are bounded.

Proposition (Invariants)

For $1 \leq j \leq H$, *following quantities are conserved along traj. of* $(GD)_Y$

$$W_{j+1}^{\mathsf{T}} W_{j+1} - W_j W_j^{\mathsf{T}} = \left(W_{j+1}^{\mathsf{T}} W_{j+1} - W_j W_j^{\mathsf{T}}\right)\Big|_{t=0}.$$

$\Longrightarrow \|W_j(t)\|_F^2 = \|W_{H+1}\|_F^2 + C_j \quad t \geq 0,\ 1 \leq j \leq H.$

Set $g(t) = \|W_{H+1}\|_F^2$. Given a traj. of $(GD)_Y$, one proves that there exists $C_0, C_1 > 0$

$$\frac{dg}{dt} \leq -C_0 g^{H+1}(t) + C_1 \Big(1 + g^H(t)\Big), \quad \forall t \geq 0.$$

Definition

For $\mathbf{W} \in \mathrm{Crit}(F)$ define

$$R(\mathbf{W}) = (\Pi W)_2^{H+1}, \quad r(\mathbf{W}) = \mathrm{rank}\, R(\mathbf{W}) \in [0, d_y],$$

$$Z(\mathbf{W}) = (\Pi W)_2^{H} \quad r_Z(\mathbf{W}) = \mathrm{rank}\, Z(\mathbf{W}) \geq R(\mathbf{W}).$$

Then

$$\mathrm{Crit}(F) = \cup_{r=0}^{d_y} \mathrm{Crit}_r(F), \quad \mathrm{Crit}_r(F) = \{\mathbf{W} \in \mathrm{Crit}(F),\ r(\mathbf{W}) = r\}.$$

$CrV(F) =$ Set of critical values of $F = \{F(\mathbf{W}), \mathbf{W} \in \mathrm{Crit}(F)\}$.

Proposition (Landscape of Deep Linear Networks)

Assume Y has **two by two distinct singular values** $S_Y = \{s_1, \cdots, s_{d_Y}\}$.

i) $CrV(F) = \{\frac{1}{2}\sum_{s \in I} s^2 \mid I \subset S_Y\}$, *finite.*

ii) $\mathrm{Crit}_{d_y}(L) =$ *set of local (and global) minima with* $F = 0$ *and* $M = 0$.

iii) *For* $0 \leq r \leq d_y - 1$, $\mathrm{Crit}_r(F)$ *algebraic variety of dim.* > 0 *made of saddle points.* *If* $r_Z > r \geq 0$, $Hessian(F)(\mathbf{W})$ *has at least one negative eigenvalue.*

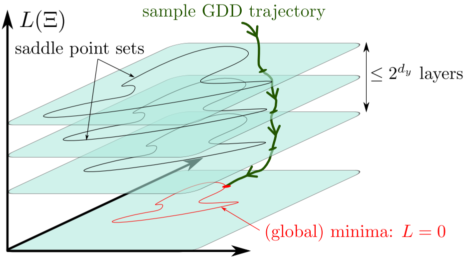
Proposition (Landscape of Deep Linear Networks)

Assume Y *has* **two by two distinct singular values** $S_Y = \{s_1, \cdots, s_{d_Y}\}$.

$i)$ $CrV(F) = \{\frac{1}{2}\sum_{s\in I} s^2 \mid I \subset S_Y\}$, *finite.*

$ii)$ $\mathrm{Crit}_{d_y}(L)$ = *set of local (and global) minima with* $F = 0$ *and* $M = 0$.

$iii)$ *For* $0 \le r \le d_y - 1$, $\mathrm{Crit}_r(F)$ *algebraic variety of dim.* > 0 *made of saddle points. If* $r_Z > r \ge 0$, $Hessian(F)(\mathbf{W})$ *has at least one negative eigenvalue.*



Gradient Descent for Linear Neural Networks - Case $H = 1$

Reformulation of Conjecture (OVF)

Conjecture (New formulation of (OVF))

For a.e. (X, Y), the union of the basins of attraction of saddles points is of measure zero.

Proposition (C., Liao, Couillet '18)

Conjecture (OVF) true if $H = 1$.

Argument relies on concept of **Normal Hyperbolicity** (due to Fenichel 1972).

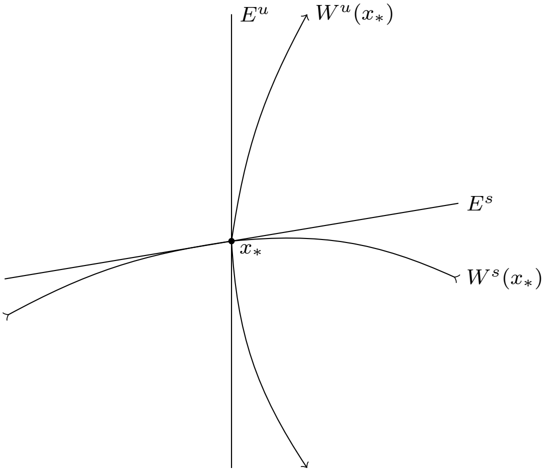


Figure: Illustration of Hyperbolic Equilibrium Point

Figure 3D

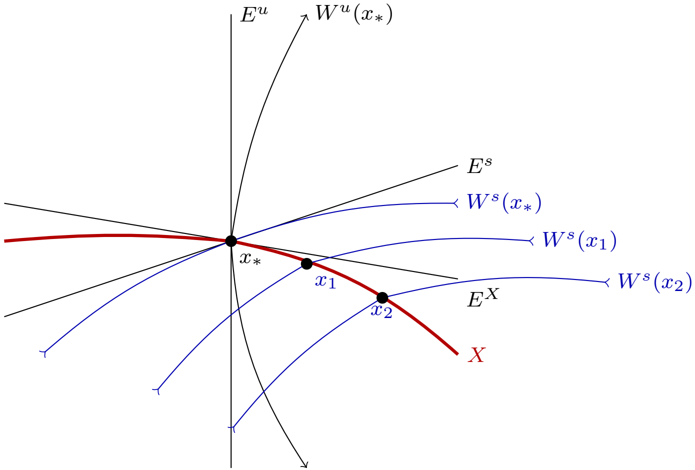


Figure: Illustration of a single-hidden-layer linear neural network