Outline

1 Motivation and Introduction

2 Main Results
Motivation: “learn” to automatically classify images

Machine Learning:
- given $m = 2M$ images of cats $x_{1}^{cat}, x_{2}^{cat}, \ldots, x_{m/2}^{cat}$ 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 $?=dog$. 

![Images of cats and dogs]
How to “learn” to classify?

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

- **Input:** training set $(X,Y)$, images $X = [x_{1}^{\text{cat}}, \ldots, x_{1}^{\text{dog}}, \ldots] \in \mathbb{R}^{d_x \times m}$ with associated labels $Y = [y_{\text{cat}}, \ldots, y_{\text{dog}}, \ldots] \in \mathbb{R}^{d_y \times m}$.
- **Output:** $W$ that minimizes the difference $\|Y - W \cdot X\|^2$

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

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

- $x_{\text{new}}^{?}$ to be a cat ($y_{\text{new}}^{?} = y_{\text{cat}}$) if $\|y_{\text{cat}} - W \cdot x_{\text{new}}^{?}\| < \|y_{\text{dog}} - W \cdot x_{\text{new}}^{?}\|$
- $x_{\text{new}}^{?}$ to be a dog ($y_{\text{new}}^{?} = y_{\text{dog}}$) otherwise
Objective: given \((X, Y)\), find \(W\) that minimizes the difference \(\|Y - W \cdot X\|^2\).

⇒ “Best” solution: if \(W \cdot X = WX\), linear regression \(W_{LR} = YX^T(XX^T)^{-1}\) 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 \(Y\)
⇒ (Brain-inspired) **LINEAR** neural network models (back to [Rosenblatt, 1958])

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

Linear deep learning (LDL):

$W_{LDL} = W_{H+1}W_H \cdots W_1$

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

\[ h_H := W_H \sigma(h_{H-1}) \]

\[ W_{H+1} \in \mathbb{R}^{d_y \times d_H}, W_1 \in \mathbb{R}^{d_1 \times d_x} \]

\[ W_{H+1} \sigma(h_H), \quad h_1 = W_1 X, \quad X \in \mathbb{R}^{d_x \times m} \]

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

with (nonlinear) activation function \( \sigma(z) \): ReLU \( (z) = \max(z, 0) \), Leaky ReLU \( \max(z, az) \) \((a > 0)\) or sigmoid \( \sigma(z) = \frac{1}{1+e^{-z}} \), arctan, tanh, ...
Why we need to be “deep”?

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

![Samples from the MNIST dataset](image)

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

<table>
<thead>
<tr>
<th>Network</th>
<th>Classification error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H = 0$ (linear regression)</td>
<td>12.0%</td>
</tr>
<tr>
<td>$H = 2$ [LeCun et al. 1998]</td>
<td>2.5%</td>
</tr>
<tr>
<td>$H = 4$ [LeCun et al. 1998]</td>
<td>0.8%</td>
</tr>
</tbody>
</table>

**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

   $$F(W_1, W_2) = \|Y - W_2W_1X\|_F^2$$

   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

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

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

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 $d_0 = d_x$)

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

and Gradient Descent associated with $F$

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

Conjecture ($\iff$ Overfitting Problem = OVF)

$$\text{(OVF): For a.e. } (X, Y) \text{ and } W_0 \in \mathcal{W}, \text{ traj. of } (GD)_{(X,Y)} \text{ starting at } W_0 \text{ CV to a GLOBAL minimum of } F.$$
Gradient Descent for Linear Neural Networks - First reductions

(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

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

**Notation**

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

Gradient dynamics, \(1 \leq j \leq H + 1\)

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

**Definition** Critical points \(\nabla F(W) = 0\)

\[\text{Crit}(F) = \{W = (W_{H+1}, \cdots, W_1) \in \mathcal{W}, \ (\Pi W)^{H+1}_{j+1} M (\Pi W)^{j-1}_1 = 0\}.\]

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 $\text{Crit}(F)$.***

**PROOF**

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

**Proposition (Lojasiewicz 50’s')**

*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 $(GD)_Y$*

\[
W_{j+1}^TW_{j+1} - W_jW_j^T = (W_{j+1}^TW_{j+1} - W_jW_j^T)_{|t=0}.
\]

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

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

\[
\frac{dg}{dt} \leq -C_0g^{H+1}(t) + C_1\left(1 + g^H(t)\right), \quad \forall t \geq 0.
\]
Gradient Descent for Linear Neural Networks - Study of $\text{Crit}(F)$

**Definition**

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

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

$$Z(W) = (\Pi W)^H_2, \quad r_Z(W) = \text{rank } Z(W) \geq R(W).$$

Then

$$\text{Crit}(F) = \bigcup_{r=0}^{d_y} \text{Crit}_r(F), \quad \text{Crit}_r(F) = \{W \in \text{Crit}(F), r(W) = r\}.$$  

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

**Proposition (Landscape of Deep Linear Networks)**

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

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

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

- $iii$) For $0 \leq r \leq d_y - 1$, $\text{Crit}_r(F)$ algebraic variety of dim. $> 0$ made of saddle points. If $r_Z > r \geq 0$, $\text{Hessian}(F)(W)$ 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) $CrV(F) = \left\{ \frac{1}{2} \sum_{s \in I} s^2 \mid I \subset S_Y \right\}$, finite.

ii) $Crit_y(L) =$ set of local (and global) minima with $F = 0$ and $M = 0$.

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