Nonconvex optimization landscapes

François Malgouyres[1]

[1] Institut de Mathématiques de Toulouse, Université Paul Sabatier

Octobre 2024

Plan

Outline

Classification et régression

Il existe un couple de variables aléatoires (X, Y)

- On connaît $\mathscr{X}$ et $\mathscr{Y}$ tels que $\mathbb{P}(X \in \mathscr{X}) = \mathbb{P}(Y \in \mathscr{Y}) = 1$
- On sait que l'on va observer x d'une réalisation (x, y) de (X, Y)
- On veut construire une fonction $g : \mathscr{X} \longrightarrow \mathscr{Y}$ prédisant y

Plus précisément, pour une fonction coût $L : \mathscr{Y} \times \mathscr{Y} \longrightarrow \mathbb{R}$, on veut trouver

$$g^* \in \operatorname{argmin} \ R(g)$$

On appelle $R(g) = \mathbb{E}\left(L(g(X), Y)\right)$ le risque de g.

Car : La personnes utilisant g observe le coût

$$R_{test}(g) \simeq \frac{1}{n'} \sum_{i=1}^{n'} L(g(x_i'), y_i')$$

pour un échantillon i.i.d $(x_i', y_i')_{i=1..n'}$ suivant la loi (X, Y), **indépendant de l'échantillon d'apprentissage**. La loi des grands nombres conduit à considérer R.

Classification et régression : les principes

- On observe $(x_i, y_i)_{i=1..n}$ un échantillon i.i.d. de même loi que (X, Y)
- On considère une famille $\mathscr{F}$ de fonctions g_w paramétrés par des paramètres w d'un espace Euclidien.
  - Pour nous : elle est basée sur les **réseaux de neurones**
- On voudrait résoudre
$$\mathbf{w}^* \in \operatorname{argmin}_{\mathbf{w}} \; R(g_{\mathbf{w}}).$$
- La **minimisation du risque empirique** consiste à résoudre
$$\widehat{\mathbf{w}} \in \operatorname{argmin}_{\mathbf{w}} \; \widehat{R}(g_{\mathbf{w}})$$
où
$$\widehat{R}(g_{\mathbf{w}}) = \frac{1}{n}\sum_{i=1}^{n} L(g_{\mathbf{w}}(x_i), y_i)$$

Ci-dessous, pour simplifier, on note $R(\mathbf{w})$ au lieu de $R(g_{\mathbf{w}})$ (idem $\widehat{R}(\mathbf{w})$).

Outline

Le contrôle du risque

- Pour toute fonction g, on appelle **risque en population** et **risque empirique** de g :

$$R(g) = \mathbb{E}\left(L(g(X),Y)\right) \qquad \text{et} \qquad \widehat{R}(g) = \frac{1}{n}\sum_{i=1}^{n} L(g(x_i),y_i)$$

- $R^* = \inf_g R(g)$ le risque optimal
- Pour $\varepsilon > 0$, on fixe $\mathbf{w}^*$ et $\widehat{\mathbf{w}}$ tels que :

$$R(g_{\mathbf{w}^*}) \leq \inf_{\mathbf{w}} R(g_{\mathbf{w}}) + \varepsilon \qquad \text{et} \qquad \widehat{R}(g_{\widehat{\mathbf{w}}}) \leq \inf_{\mathbf{w}} \widehat{R}(g_{\mathbf{w}}) + \varepsilon$$

- Pour $\mathbf{w}$ retourné par un algorithme

On **décompose**

$$\begin{array}{rlcll}
0 \leq & R(g_{\mathbf{w}}) & - & R^* & \text{(l'excès de risque)} \\
= & R(g_{\mathbf{w}}) & - & \widehat{R}(g_{\mathbf{w}}) & \text{(erreur de généralisation)} \\
+ & \widehat{R}(g_{\mathbf{w}}) & - & \widehat{R}(g_{\widehat{\mathbf{w}}}) & \text{(erreur d'optimisation)} \\
+ & \widehat{R}(g_{\widehat{\mathbf{w}}}) & - & \widehat{R}(g_{\mathbf{w}^*}) & \leq \varepsilon \\
+ & \widehat{R}(g_{\mathbf{w}^*}) & - & R(g_{\mathbf{w}^*}) & \text{(erreur de généralisation)} \\
+ & R(g_{\mathbf{w}^*}) & - & R^* & \text{(erreur d'approximation)}
\end{array}$$

Puis on **majore** indépendamment chaque terme.

Le contrôle du risque

- **Erreur d'optimisation** : $|\widehat{R}(g_{\mathbf{w}}) - \widehat{R}(g_{\widehat{\mathbf{w}}})|$
  - Est-ce-que l'optimisation a réussi : Optimisation non-convexe, paysage de la fonction objectif...
  - **En deep-learning :** Elle est petite pour les **"gros" réseaux**.
- **Erreur de généralisation** : $|R(g_{\mathbf{w}}) - \widehat{R}(g_{\mathbf{w}})|$
  - Quelles conditions sur le réseau et l'échantillon pour avoir la **convergence uniforme**

$$\sup_{\mathbf{w}} |\widehat{R}(g_{\mathbf{w}}) - R(g_{\mathbf{w}})| \qquad \text{soit petit?}$$

  - **En théorie :** Dimension de Vapnik-Chervonenkis $\simeq$ nombre de paramètres
  - **En pratique :** Reste faible même pour les **"gros" réseaux**

$\rightsquigarrow$ **régularisation implicite**.

- **Erreur d'approximation** : $|R(g_{\mathbf{w}^*}) - R(g_b)|$, où $R(g_b) \simeq \inf_g R(g)$
  - Il suffit que $\|g_{\mathbf{w}^*} - g_b\|$ soit petit (le choix de la norme compte).
  - Quelles fonctions peut-on approximer avec notre classe de fonctions? ("Expressive power", "Expressivité",...)
  - **En deep-learning :** Elle est petite pour les **"gros" réseaux**.

Outline

Les réseaux de neurones

- H couches (on dit aussi $H-1$ couches cachées)
- les tailles $n_0, n_1, \cdots, n_H \in \mathbb{N}$ des différentes couches
- On note $f_h(x)$: le résultat obtenu en calculant le contenu de la couche h pour l'entrée $x \in \mathbb{R}^{n_0}$
- On note $W_h \in \mathbb{R}^{n_h \times n_{h-1}}$: la matrice contenant les poids sur les arcs entre la couche $h-1$ et la couche h
- On note $b_h \in \mathbb{R}^{n_h}$: le biais ajouté à la couche h
- On note σ_h: la fonction d'activation appliquée à chaque couche
  - Typiquement, elle applique la même fonction à chaque entrée d'un vecteur

La prédiction/l'inférence : la fonction $f_H : \mathbb{R}^{n_0} \longrightarrow \mathbb{R}^{n_H}$

$$\begin{cases} f_0(x) = x \\ f_h(x) = \sigma_h\left(W_h f_{h-1}(x) + b_h\right) \qquad , \forall h = 1, \cdots, H \end{cases}$$

Le plus souvent: $\sigma(t) = \max(0, t)$ (ReLU) et

$$f_H(x) = W_H\Big(\cdots\big(\sigma\, W_2 \sigma\left(W_1 x + b_1\right) + b_2\big) \cdots\Big) + b_H$$

Les réseaux de neurones

- **Fully-connected layer** : La matrice est “pleine”
- Fully connected **linear** network: On a $\sigma(t) = t$, $\forall t \in \mathbb{R}$, sans biais $b_h = 0$, pour tout $h = 1..H$.
- **Fonctions d’activation** :
  - Une zoologie abondante : tanh, heavyside, identité, sigmoïde, etc
  - Des non-locales : group-sort, max_pooling
  - La plus utilisée est "Rectified Linear Unit" (ReLU) : $\sigma(t) = \max(0, t)$
  - Souvent, celle associée à la dernière couche est l’identité ou softmax (pour la classification).
- **L’architecture du réseau:** La donnée des hyperparamètres.
  Ex : Un réseau ReLU fully-connected de largeur 100 et de profondeur 10.
- Une **zoologie d’architectures abondante** :
  - Res-Net
  - En vision : U-Net (segmentation), YOLO (detection), transformer...
  - Pour les séries temporelles et modèles de langages : RNN, LSTM, GRU, SSM, transformer...
  - etc

Les réseaux de neurones

Proposition : Propriétés de f_H, cas ReLU

Pour tout réseau de neurones, avec la fonction d'activation ReLU et l'identité sur la dernière couche

1. La fonction f_H est continue
2. La fonction f_H est affine par morceaux
3. Il existe une partition compatible ayant moins de $2^{n_1+\cdots+n_{H-1}}$ morceaux dont les morceaux sont des polyèdres ayant au plus $n_1 + \cdots + n_{H-1}$ faces.

- Réciproque dans : Arora, Raman, et al. "Understanding Deep Neural Networks with Rectified Linear Units." ICLR, 2018:
  Toute fonction continue, affine sur un nombre fini de polyhèdres (dont l'union est $\mathbb{R}^{n_0}$) peut être représenté par un réseau de neurone ReLU.

Plan

Outline

L'optimisation : Premières propriétés

- On dispose d'un échantillon $(x_i, y_i)_{i=1..n}$.
- On dispose d'un réseau de profondeur H, d'architecture fixée,
  - de paramètre
    $$\theta = (W_H, \ldots, W_1, b_H, \ldots, b_1) \in \Theta$$
  - les fonctions d'activation sont notées σ_h

  La prédiction est la fonction

$$f_\theta(x) = \sigma_H\Big(W_H \cdots \sigma_2\big(W_2 \sigma_1(W_1 x + b_1) + b_2\big) \cdots + b_H\Big)$$

- Les réseaux ReLU:
  - $\sigma_h = ReLU$, pour $h = 1, \ldots, H-1$
  - Cas de la **régression** : $\sigma_H = Id$
  - Cas de la **classification** : σ_H est softmax.
- On considère une fonction coût définie par

$$\begin{array}{rcl} E : \Theta & \longrightarrow & \mathbb{R} \\ \theta & \longmapsto & E(\theta) = \sum_{i=1}^n L(f_\theta(x_i), y_i) \end{array}$$

  pour une fonction coût $L : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$.

L'optimisation : Premières propriétés

Proposition: Non-coercivité dans le cas ReLU

Pour tout réseau ReLU, pour tout échantillon d'apprentissage, la fonction E **n'est pas coercive**.

Preuve utilisant l'homogénéité:
On considère θ avec des biais nuls et, pour tout $\lambda > 0$, on définit θ^λ par

$$W_1^\lambda = \lambda^{H-1} W_1 \qquad \text{et} \qquad W_h^\lambda = \lambda^{-1} W_h, \forall h = 2, \ldots, H$$

et des biais nuls.
Comme pour tout $\lambda > 0$, $\sigma(\lambda t) = \lambda \sigma(t)$, on a pour tout x et tout $\lambda > 0$,

$$\begin{aligned} f_{\theta^\lambda}(x) &= \sigma_H\Big(\lambda^{-1} W_H \cdots \sigma_2\big(\lambda^{-1} W_2 \sigma_1\big(\lambda^{H-1} W_1 x\big)\big) \cdots\Big) \\ &= f_\theta(x) \end{aligned}$$

Donc $E(\theta) = E(\theta^\lambda)$, pour tout $\lambda > 0$.
Donc E n'est pas coercive.

□

L'optimisation : Premières propriétés

Proposition: Non-convexité

Pour la loss quadratique $L(y',y) = (y'-y)^2$, le réseaux ReLU fully connnected d'architecture $(1,1,1)$ et l'échantillon constitué de l'exemple $(x,y) = (1,0)$, le risque empirique est non convexe.

Preuve: On a pour tout $(w_1,w_2) \in \mathbb{R}^2$ avec $w_1 \geq 0$, et pour $(b_1,b_2) = (0,0)$

$$E(w_1,w_2,b_1,b_2) = (w_2\sigma(w_1 x + b_1) + b_2 \; - \; y)^2 = (w_1 w_2)^2.$$

Cette fonction n'est pas convexe car:

- En $(w_1,w_2) = (0,1)$, $\qquad E(0,1,0,0) = 0$
- En $(w_1,w_2) = (1,0)$, $\qquad E(1,0,0,0) = 0$
- En $(w_1,w_2) = (0.5,0.5) = 0.5 \times (1,0) + 0.5 \times (0,1)$,

$$E(0.5,0.5,0,0) = 0.25^2 \;\; > \;\; 0 = 0.5 \times E(1,0,0,0) + 0.5 \times E(0,1,0,0).$$

$\square$

Outline

La Rétropropagation

On suppose:

- La fonction coût est de la forme :

$$\theta \longmapsto E(\theta) = \sum_{i=1}^{n} L\big(f_\theta(x_i), y_i\big)$$

  où L est C^1.
- On suppose que, pour tout $h \in \{1, \ldots, H\}$, σ_h est C^1
  - On peut régulariser σ_h
  - Il existe un cadre formel pour faire du calcul différentiel et de l'optimisation avec des activations comme ReLU
- Sous ces hypothèses: E **est différentiable**.
- Les frameworks de deep learning utilisent la **différentiation automatique**.
- Ci-dessous, on ne considère qu'un unique exemple (x, y):
  - On somme si besoin plusieurs gradients $\nabla L\big(f_\theta(x_i), y_i\big)$

La Rétropropagation

- $\mathbf{W}_h$ la matrice pour passer de la couche $h-1$ à h,
- $f_\theta^h(x)$, le contenu de la couche h,
- $d_\theta^h(x) = \sigma'_h\big(\mathbf{W}_h f_\theta^{h-1}(x) + \mathbf{b}_h\big)$.

Proposition: Gradient pour un réseau fully-connected

On a pour $h = 1..H$, $i = 1..n_h$, $j = 1..n_{h-1}$

$$\frac{\partial E}{\partial (\mathbf{W}_h)_{i,j}}(\theta) = \left[f_\theta^{h-1}(x)\right]_j \left[d_\theta^h(x)\right]_i \Delta_i^h(x)$$

$$\frac{\partial E}{\partial (\mathbf{b}_h)_i}(\theta) = \left[d_\theta^h(x)\right]_i \Delta_i^h(x)$$

où $\Delta^h(x) \in \mathbb{R}^{n_h}$ est défini par

$$\Delta^h(x) = \begin{cases} \nabla_{y_1} L(f_\theta(x), y) & \text{, si } h = H \\ \mathbf{W}_{h+1}^T.\mathrm{diag}\left(d_\theta^{h+1}(x)\right).\Delta^{h+1}(x) & \text{, sinon} \end{cases}$$

Rq: $\mathbf{W}_{h+1}^T$ remonte d'un niveau dans le réseau: **Rétropropagation**.

Rétropropagation: Problèmes connus

- **"Vanishing/Exploding gradient"** :
  - Si $\mathbf{W}_{h+1}^T.\mathrm{diag}\left(d_\theta^{h+1}(x)\right)$ est systématiquement une contraction $\Longrightarrow$ "Vanishing gradient"
  - Si $\mathbf{W}_{h+1}^T.\mathrm{diag}\left(d_\theta^{h+1}(x)\right)$ augmente systématiquement la norme $\Longrightarrow$ "Exploding gradient"

Rétropropagation: Problèmes connus

Dans certaines régions de l'espace, la fonction objectif est très irrégulière :
Pour une fonction d'activation homogène[1] (notamment ReLU):
Pour $\lambda > 0$, $\lambda \sim 0$

$$\mathbf{W}_1^\lambda = \lambda^{H-1}\mathbf{W}_1 \qquad \text{et} \qquad \mathbf{W}_h^\lambda = \lambda^{-1}\mathbf{W}_h, \forall h = 2..H$$

$$\mathbf{b}_1^\lambda = \lambda^{H-1}\mathbf{b}_1 \qquad \text{et} \qquad \mathbf{b}_h^\lambda = \lambda^{H-h}\mathbf{b}_h, \forall h = 2..H$$

On a, pour tout x, $f_{\theta^\lambda}(x) = f_\theta(x)$, pour $h > 1$

$$\begin{aligned}
\frac{\partial E}{\partial \mathbf{W}_{h,i,j}}(\theta^\lambda) &= \left[f_{\theta^\lambda}^{h-1}(x)\right]_j \left[d_{\theta^\lambda}^h(x)\right]_i (\Delta_\lambda^h)_i(x) \\
&= \lambda^{H-(h-1)}\left[f_\theta^{h-1}(x)\right]_j \left[d_\theta^h(x)\right]_i \lambda^{-(H-h)}\Delta_i^h(x) \\
&= \lambda \frac{\partial E}{\partial \mathbf{W}_{h,i,j}}(\theta)
\end{aligned}$$

mais $\frac{\partial E}{\partial \mathbf{b}_{h,i}}(\theta^\lambda) = \lambda^{-(H-h)}\frac{\partial E}{\partial \mathbf{b}_{h,i}}(\theta)$ et $\frac{\partial E}{\partial \mathbf{W}_{1,i,j}}(\theta^\lambda) = \lambda^{-(H-1)}\frac{\partial E}{\partial \mathbf{W}_{1,i,j}}(\theta)$.
Les petits b_h^λ ont de forts gradients et seront multipliés par des forts W_h^λ.

[1] Attention à la non-différentiabilité en 0

Pour gagner en profondeur, faciliter l'apprentissage

- Couches ResNet

$$f_h(x) = \sigma\Big(W_h \, \sigma\big(W_{h-1} f_{h-2}(x) + b_{h-1}\big) + b_h + f_{h-2}(x)\Big)$$

- Batch normalisation : On centre (au mieux) les données après chaque couche, à l'aide d'un opérateur diagonal.
- Contraindre les matrices $\mathbf{W}_h$ à être orthogonales ou presque. Bonus: Gain de robustesse.
- Augmenter la largeur. Pb: Complexité = largeur^2.

Plan

Outline

Paysage de la fonction objectif : Introduction

Rappel, pour tout[2] **w**:

$$
\begin{array}{rlcll}
0 \leq & R(f_{\mathbf{w}}) & - & R^* & \text{(l'excès de risque)} \\
= & R(f_{\mathbf{w}}) & - & \widehat{R}(f_{\mathbf{w}}) & \text{(erreur de généralisation)} \\
+ & \widehat{R}(f_{\mathbf{w}}) & - & \widehat{R}(f_{\widehat{\mathbf{w}}}) & \text{(erreur d'optimisation)} \\
+ & \widehat{R}(f_{\widehat{\mathbf{w}}}) & - & \widehat{R}(f_{\mathbf{w}^*}) & \leq \varepsilon \\
+ & \widehat{R}(f_{\mathbf{w}^*}) & - & R(f_{\mathbf{w}^*}) & \text{(erreur de généralisation)} \\
+ & R(f_{\mathbf{w}^*}) & - & R^* & \text{(erreur d'approximation)}
\end{array}
$$

On utilise un algorithme d'optimisation pour trouver un **w**, on veut que

$$\widehat{R}(f_{\mathbf{w}}) - \inf_{\mathbf{w}} \widehat{R}(f_{\mathbf{w}})$$

soit le plus faible possible.

Idéalement, on voudrait que cette quantité soit nulle ou au moins on voudrait la borner supérieurement.

[2] Pour alléger les notations, on oublie **b**.

Paysage de la fonction objectif : Introduction

On suppose que $\mathbf{w} \longmapsto \widehat{R}(f_{\mathbf{w}})$ est C^2 partout. On a

$$\widehat{R}(f_{\mathbf{w}}) = \widehat{R}(f_{\mathbf{w}^*}) + \langle \nabla_{\mathbf{w}} \widehat{R}(f_{\mathbf{w}^*}), \mathbf{w} - \mathbf{w}^* \rangle + \frac{1}{2} \langle \nabla^2_{\mathbf{w}} \widehat{R}(f_{\mathbf{w}^*})(\mathbf{w} - \mathbf{w}^*), \mathbf{w} - \mathbf{w}^* \rangle + o(\|\mathbf{w} - \mathbf{w}^*\|^2)$$

On distingue:

- **$\mathbf{w}^*$ est un minimiseur global:**
  - $\forall \mathbf{w}, \qquad \widehat{R}(f_{\mathbf{w}^*}) \leq \widehat{R}(f_{\mathbf{w}})$
  - $\widehat{R}(f_{\mathbf{w}^*}) = \min_{\mathbf{w}} \widehat{R}(f_{\mathbf{w}})$
- **$\mathbf{w}^*$ est un minimiseur local:**
  - Il existe un voisinage ouvert $\mathcal{O}$ de $\mathbf{w}^*$ tel que

$$\forall \mathbf{w} \in \mathcal{O}, \qquad \widehat{R}(f_{\mathbf{w}^*}) \leq \widehat{R}(f_{\mathbf{w}})$$

- **$\mathbf{w}^*$ est un point critique du second ordre:**
  - On a

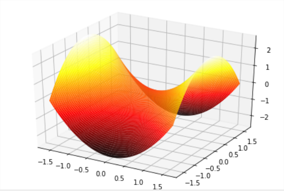
$$\nabla \widehat{R}(f_{\mathbf{w}^*}) = 0 \qquad \text{et} \qquad \nabla^2 \widehat{R}(f_{\mathbf{w}^*}) \succeq 0$$

- **$\mathbf{w}^*$ est un point critique du premier ordre:**
  - On a

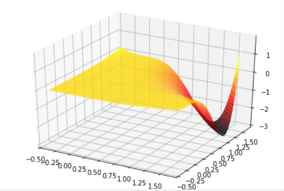
$$\nabla \widehat{R}(f_{\mathbf{w}^*}) = 0$$

Paysage de la fonction objectif : Introduction

- ***w*** * **est un point selle** si c'est un point critique qui n'est ni un minimiseur local, ni un maximiseur local
  - **Un point selle** $\boldsymbol{w^*}$ **est strict :** si ce n'est pas un point critique du second ordre (i.e., le Hessian a une v.p. négative).
  - **Un point selle** $\boldsymbol{w^*}$ **est non-strict:** si c'est un point critique du second ordre (i.e. le Hessian est semi-defini positif et a une v.p. égale à 0. Typiquement, un terme d'ordre supérieur en fait un point selle.).



(a) Point selle strict



(b) Point selle non-strict

Paysage de la fonction objectif : Introduction

Optimisation non convexe avec les mains

Pour des fonctions non-convexes, on sait montrer que

- **dans un cadre assez vaste**, l'algorithme du gradient (ou gradient stochastique) **converge vers un point critique du premier ordre**
- **dans un cadre plus restreint**, l'algorithme du gradient **converge vers un point critique du second ordre**
- Pour le gradient stochastic, **sans vitesse de convergence**, que les itérés convergent vers un minimiseur local.

- S. Gadat, F. Panloup, S. Saadane. "Stochastic heavy ball." Electronic Journal of Statistics 12.1 (2018): 461-529.
- J. Lee, M. Simchowitz, M. Jordan, B. Recht." Gradient Descent Converges to Minimizers." COLT 2016.

Outline

Paysage pour les réseaux larges

Différents énoncés décrits dans

- Gori, Tesi, "On the problem of local minima in backpropagation", IEEE PAMI, 1992
- Yu, Chen, "On the local minima free condition of backpropagation learning", IEEE Trans. Neural Networks, 1995
- Nguyen, Hein, "The loss surface of deep and wide neural networks", ICML, 2017
- ....

On considère:

- un problème de régression
- un réseau fully-connected de paramètre $\theta = (\mathbf{W}_1, \ldots, \mathbf{W}_H, \mathbf{b}_1, \ldots, \mathbf{b}_H)$
- des observations $(x^i, y^i)_{i=1..n}$:

$$E(\theta) = \widehat{R}(f_\theta) = \sum_{i=1}^{n} L(f_\theta(x^i) - y^i)$$

Paysage pour les réseaux larges

Théoreme (Le Paysage pour les réseaux larges)

On suppose que σ est C^1 et que, pour tout $t \in \mathbb{R}$, $\sigma'(t) \neq 0$. On suppose que le coût L est C^1, à valeur dans $\mathbb{R}^+$ et tel que $L(0) = 0$. On suppose aussi que $\nabla L(y) = 0$ si et seulement si $y = 0$. On note $X = [x^1 \cdots x^n] \in \mathbb{R}^{n_0 \times n}$ et $A = \begin{pmatrix} X \\ \mathbb{1}_n^T \end{pmatrix}$.

On considère un point critique du premier ordre $\theta = (\mathbf{W}_1, \ldots, \mathbf{W}_H, \mathbf{b}_1, \ldots, \mathbf{b}_H)$ de E. On suppose que $\operatorname{rang}(A) = n$ et que, pour tout $h \in \{1, \ldots, H\}$, $\operatorname{rang}(\mathbf{W}_h) = n_h$. Alors, on a

$$\widehat{R}(f_\theta) = 0$$

et θ est un minimiseur global.

Les hypothèses fortes sont celles sur le rang. Elles impliquent notamment

$$n \leq n_0 + 1 \qquad \text{et} \qquad n_H \leq n_{H-1} \leq \cdots \leq n_0$$

Nb: Ci-dessus l'indice 0 est arbitraire car on pourrait supposer que les x_i sont le résultat des premières couches d'un réseau.

Plan

Géométrie locale des réseaux ReLU fully-connected

En collaboration avec

Joachim Bona-Pellissier
Postdoc Malga, Genova, Italie

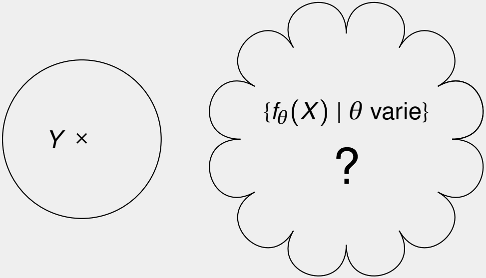
François Bachoc
Mcf, IUF, ANITI, IMT, UPS

L'objet étudié

- Étant donné un réseau ReLU
- $X \in \mathbb{R}^{n_0 \times n}$ (resp $Y \in \mathbb{R}^{n_L \times n}$) contiennent n exemples d'entrées (resp sorties) dans $\mathbb{R}^{n_0}$ (resp dans $\mathbb{R}^{n_L}$)
- On note $f_\theta(X) \in \mathbb{R}^{n_L \times n}$ la prédiction de X par le réseau de paramètre $\theta \in \mathbb{R}^p$
- Pour apprendre, on résoud (par exemple)

$$\mathrm{argmin}_\theta \|f_\theta(X) - Y\|_F^2$$

- On étudie les ensembles
  - l'*image* $\{f_\theta(X) \mid \theta \text{ varies}\}$
  - la *pre-image* $\{\theta' \mid f_{\theta'}(X) = f_\theta(X)\}$

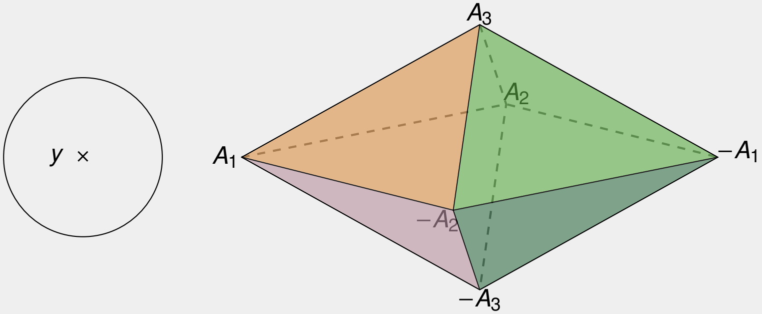


Analogie avec la régularisation ℓ^1

- Soient $A \in \mathbb{R}^{n\times p}$, $y \in \mathbb{R}^n$
- On écrit la régularisation ℓ^1 sous la forme

$$\left| \begin{array}{l} \mathrm{argmin}_x \; \|Ax - y\|^2 \\ \|x\|_1 \leq \tau \end{array} \right.$$

$$\{Ax \mid \|x\|_1 \leq \tau\} = \mathrm{conv}\big(A_1, -A_1, \cdots, A_p, -A_p\big)$$

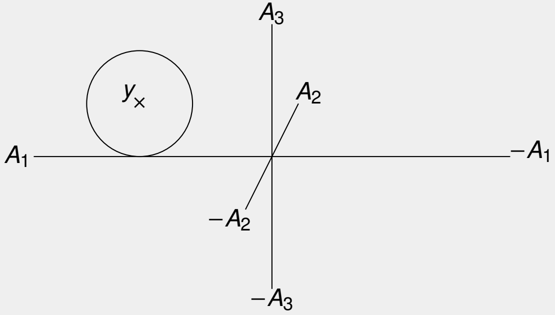


Analogie avec la régularisation ℓ^0

- Soient $A \in \mathbb{R}^{n\times p}$, $y \in \mathbb{R}^n$
- On écrit la régularisationℓ^1 sous la forme

$$\left| \begin{array}{l} \operatorname{argmin}_x \ \|Ax - y\|^2 \\ \|x\|_0 \leq k \end{array} \right.$$

$$\{Ax \mid \|x\|_0 \leq 1\}$$

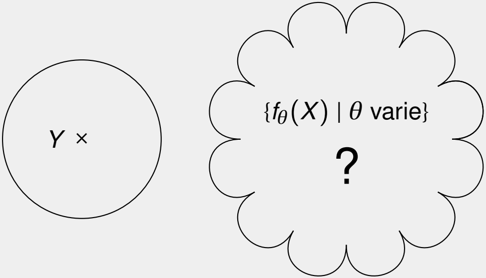


L'objet étudié

- Étant donné un réseau ReLU
- $X \in \mathbb{R}^{n_0 \times n}$ (resp $Y \in \mathbb{R}^{n_L \times n}$) contiennent n exemples d'entrées (resp sorties) dans $\mathbb{R}^{n_0}$
- On note $f_\theta(X) \in \mathbb{R}^{n_L \times n}$ la prédiction de X par le réseau de paramètre $\theta \in \mathbb{R}^p$
- Pour apprendre, on résoud (par exemple)

$$\mathrm{argmin}_\theta \|f_\theta(X) - Y\|_F^2$$

- On étudie les ensembles
  - l'*image* $\{f_\theta(X) \mid \theta \text{ varies}\}$
  - la *pre-image* $\{\theta' \mid f_{\theta'}(X) = f_\theta(X)\}$



L'objet étudié

- On étudie les ensembles
  - l'*image* $\{f_\theta(X) \mid \theta \text{ varies}\}$;
  - la *pre-image* $\{\theta' \mid f_{\theta'}(X) = f_\theta(X)\}$.

$$\mathbb{R}^p \longrightarrow \mathbb{R}^{n_L \times n}$$

$\times\, \theta$ $\qquad\qquad$ $\times\, f_\theta(X)$

$\{\theta' \mid f_{\theta'}(X) = f_\theta(X)\}$

Géométrie locale des réseaux ReLU fully-connected.

On note E les arcs du réseaux et B les neurones des couches 1 jusqu'à H.
On note $\theta \in \mathbb{R}^E \times \mathbb{R}^B$ que l'on identifie, si besoin, à
$(W_1, \ldots, W_H, b^1, \ldots, b^H) \in (\mathbb{R}^{n_1 \times n_0} \times \cdots \times \mathbb{R}^{n_H \times n_{H-1}}) \times (\mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_H})$.
On rappelle

$$\begin{cases} f_0(x) = x \\ f_h(x) = \sigma_h(W_h f_{h-1}(x) + b_h) & , \forall h = 1, \cdots, H \end{cases}$$

Pour $n \in \mathbb{N}^*$, on définit la fonction **pattern d'activation**

$$\begin{array}{rcl} a : \mathbb{R}^{n_0 \times n} \times (\mathbb{R}^E \times \mathbb{R}^B) & \longrightarrow & \{0,1\}^{(n_1 + \cdots + n_{H-1}) \times n} \\ (X, \theta) & \longmapsto & a(X, \theta) \end{array}$$

où

$$a(X,\theta)_{i,j} = \begin{cases} 1 & \text{si } [W_h f_{h-1}(x^j) + b_h]_v \geq 0 \\ 0 & \text{sinon,} \end{cases}$$

pour v le neurone correspondant à i, $h \in \{1, \ldots, H-1\}$ la couche de v, et la j^{ieme} colonne x^j de X.

Pour $X \in \mathbb{R}^{n_0 \times n}$ fixé, elle atteint un nombre fini de valeur que l'on note $\Delta_1^X, \ldots, \Delta_{q_X}^X$.

Géométrie locale des réseaux ReLU fully-connected.

On note, pour $X \in \mathbb{R}^{n_0 \times n}$ fixé et pour $j \in \{1, \ldots, q_X\}$,

$$\widetilde{\mathcal{U}}_j^X = \mathrm{Int}\left\{\theta \in \mathbb{R}^E \times \mathbb{R}^B \mid a(X,\theta) = \Delta_j^X\right\},$$

et $m_X = \left|\left\{j \in \{1, \ldots, q\} \mid \widetilde{\mathcal{U}}_j^X \neq \emptyset\right\}\right|$.
Quitte à changer l'ordre, on suppose que les $\widetilde{\mathcal{U}}_1^X, \ldots, \widetilde{\mathcal{U}}_{m_X}^X$ sont non-vides.

Lemme (Prédiction polynomiale par morceaux en θ)

Pour tout $n \in \mathbb{N}^*$, $X \in \mathbb{R}^{n_0 \times n}$, les ensembles $\widetilde{\mathcal{U}}_1^X, \ldots, \widetilde{\mathcal{U}}_{m_X}^X$ sont non-vides, ouverts et disjoints

- *Pour tout $j = 1, \ldots, m_X$, $\theta \longmapsto f_\theta(X)$ est une fonction polynomiale de degré inférieur à H sur $\widetilde{\mathcal{U}}_j^X$.*
- *Le complémentaire $\left(\cup_{j=1}^{m_X} \widetilde{\mathcal{U}}_j^X\right)^c$ est fermé et de mesure de Lebesgue nulle.*

Rq : La fonction $\theta \longmapsto f_\theta(X)$ est une composition de fonctions continues, elle est continue.

Géométrie locale des réseaux ReLU fully-connected.

Pour $n \in \mathbb{N}^*$, $X \in \mathbb{R}^{n_0 \times n}$, et $j \in \{1, \ldots, m_X\}$, on note

$$r_j^X = \max_{\theta \in \widetilde{\mathcal{U}}_j^X} \mathrm{rk}\left(\mathbf{D} f_\theta(X)\right) \qquad \text{et} \qquad \mathcal{U}_j^X = \{\theta \in \widetilde{\mathcal{U}}_j^X \mid \mathrm{rk}\left(\mathbf{D} f_\theta(X)\right) = r_j^X\}.$$

Théoreme (J. Bona-Pellissier, F. Bachoc, F. Malgouyres 2023)

Pour tout réseau ReLU, $n \in \mathbb{N}^*$, $X \in \mathbb{R}^{n_0 \times n}$, par définition

- $\mathcal{U}_1^X, \ldots, \mathcal{U}_{m_X}^X$ *sont non-vides et disjoints;*
- *Pour tout $j \in \{1, \ldots, m_X\}$, la fonction $\theta \mapsto a(X, \theta)$ est constante sur $\mathcal{U}_j^X$ et prend des valeurs distinctes pour $j' \neq j$;*
- *Pour tout $j \in \{1, \ldots, m_X\}$, $\theta \mapsto \mathrm{rk}\left(\mathbf{D} f_\theta(X)\right)$ est constante sur $\mathcal{U}_j^n$ et vaut r_j^X.*

De plus,

- *Les ensembles $\mathcal{U}_1^X, \ldots, \mathcal{U}_{m_X}^X$ sont ouverts;*
- $\left(\cup_{j=1}^{m_X} \mathcal{U}_j^X\right)^c$ *est fermé et de mesure de Lebesgue nulle;*
- *Pour tout $j \in \{1, \ldots, m_X\}$, $\theta \mapsto f_\theta(X)$ est une fonction polynomiale de degré inférieur à H sur $\mathcal{U}_j^X$.*

Géométrie locale des réseaux ReLU fully-connected.

Corollaire

Pour tout réseaux ReLU fully-connected, profond.
Pour tout $n \in \mathbb{N}^*$, $X \in \mathbb{R}^{n_0 \times n}$, $j \in \{1, \ldots, m_X\}$ *et* $\theta \in \mathscr{U}_j^X$, *il existe* $\varepsilon_{X,\theta} > 0$ *tel que*

- *L'image locale*
$$\left\{ f_{\theta'}(X) \in \mathbb{R}^{n_L \times n} \mid \|\theta' - \theta\| < \varepsilon_{X,\theta} \right\}$$
est une variété régulière de dimension $\mathrm{rk}\left(\mathbf{D} f_\theta(X)\right)$;
- *la pre-image*
$$\left\{ \theta' \in \mathbb{R}^E \times \mathbb{R}^B \mid f_{\theta'}(X) = f_\theta(X) \quad \textit{and} \quad \|\theta' - \theta\| < \varepsilon_{X,\theta} \right\}$$
est une variété régulière de dimension $|E| + |B| - \mathrm{rk}\left(\mathbf{D} f_\theta(X)\right)$.

Quelles sont les positions relatives des zones ? Interprétation des expériences

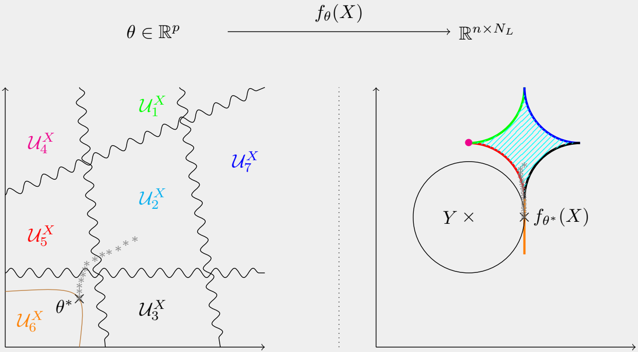
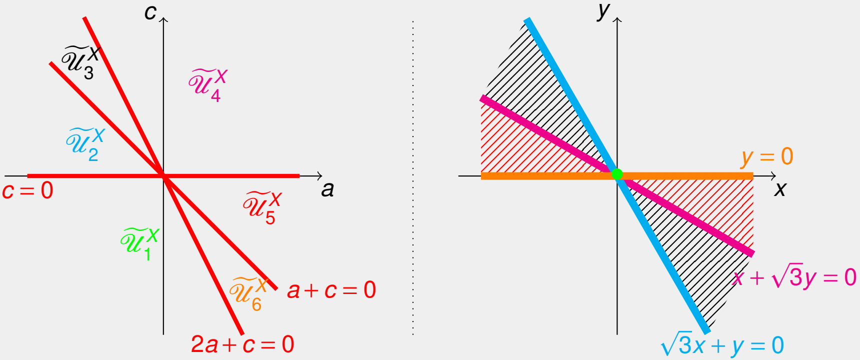


Figure: (Gauche) Représentation schématiques des $\mathscr{U}_j^X$ et (droite) de leurs images $\{f_\theta(X) \mid \theta \in \mathscr{U}_j^X\}$, $j \in \{1, \ldots, 7\}$.

Quelles sont les positions relatives des zones ?
Un exemple

- $n_0 = n_1 = n_2 = 1$, $\theta = (a,b,c,d) \in \mathbb{R}^4$, $n = 3$, $X = (0,1,2) \in \mathbb{R}^{1\times 3}$,
$$f_\theta(X) = \big(b\sigma(c)+d \ \ , \ \ b\sigma(a+c)+d \ \ , \ \ b\sigma(2a+c)+d\big) \in \mathbb{R}^{1\times 3}.$$
- $\mathscr{P}$ espace vectoriel orthogonal à $(1,1,1)$.

$$\theta = (a,b,c,d) \in \mathbb{R}^4 \xrightarrow{f_\theta(X)} \mathbb{R}^{1\times 3} \xrightarrow{\text{restrict to } \mathscr{P}} \mathbb{R}^2$$

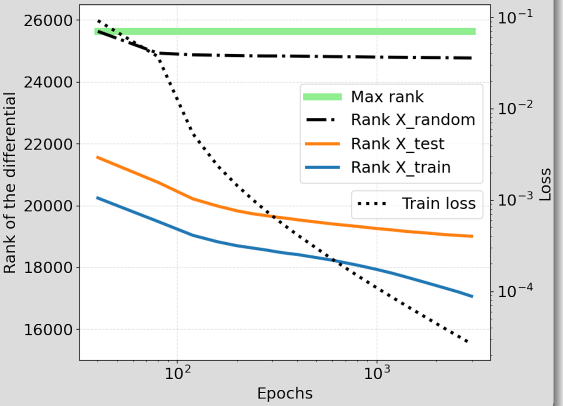


Experience 1: Comportement pendant l'apprentissage

Description de l'expérience

- architecture $(784, 30, 30, 30, 10)$;
- nombre of parametres: 25720;
- Données d'apprentissage MNIST X_{train} de taille 4000;
- Données de test MNIST X_{test} de taille 20000;
- Données aléatoires X_{random} (Bruit Gaussien) de taille 20000.

Dimension locale pendant l'apprentissage

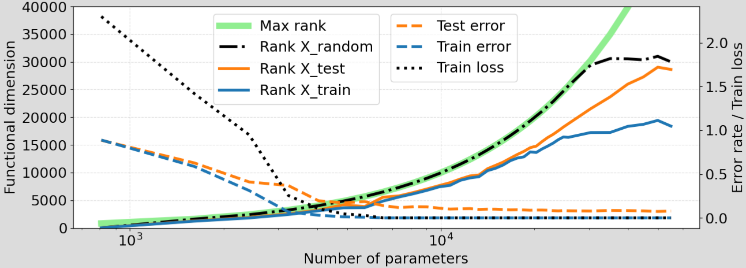


Experience 2: Comportement quand la largeur varie

Description de l'expérience

- architecture $(784, w, w, w, 10)$, pour plusieurs w;
- Données d'apprentissage MNIST X_{train} de taille 4000;
- Données de test MNIST X_{test} de taille 10000;
- Données aléatoires X_{random} (Bruit Gaussien) de taille 40000.

Functional dimensions as the width increases

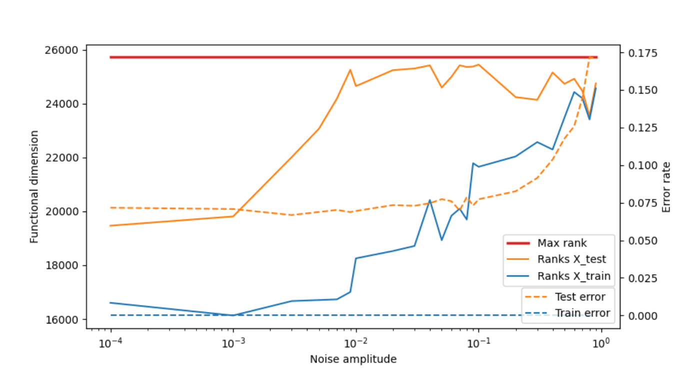


Experience 3: Comportement quand X est bruité

Description de l'expérience

- Architecture $(784, 30, 30, 30, 10)$;
- Bruit Gaussien additif sur X_{train}.
- Même données.

Functional dimensions with noisy inputs.



Experience 4: Comportement quand Y est bruité

Description de l'expérience

- Architecture $(784, 30, 30, 30, 10)$;
- Ajout de données avec Y aléatoire;
- Données d'apprentissage MNIST X_{train} propre de taille 4000.

Dimension locale pour des sortie bruitées.

Conclusion

- La dimension locale, appelé **functional dimensions**, varie
- Existence d'une **régularisation implicite induite par la géométrie** sur MNIST
- La dimension locale est (presque sûrement) induite par le pattern d'activation.
- Il est invariant par les changements d'échelles positifs et les permutations de neurones.
- Il est lié à la distribution des X
- Le lien avec la distribution des Y n'est pas mis en évidence

Bien plus dans l'article: https://arxiv.org/pdf/2402.08269

Plan

Outline

Paysage de la fonction objectif : Introduction

Rappel, pour tout[3] **w**:

$$
\begin{array}{rlcll}
0 \leq & R(f_{\mathbf{w}}) & - & R^* & \text{(l'excès de risque)} \\
= & R(f_{\mathbf{w}}) & - & \widehat{R}(f_{\mathbf{w}}) & \text{(erreur de généralisation)} \\
+ & \widehat{R}(f_{\mathbf{w}}) & - & \widehat{R}(f_{\widehat{\mathbf{w}}}) & \text{(erreur d'optimisation)} \\
+ & \widehat{R}(f_{\widehat{\mathbf{w}}}) & - & \widehat{R}(f_{\mathbf{w}^*}) & \leq \varepsilon \\
+ & \widehat{R}(f_{\mathbf{w}^*}) & - & R(f_{\mathbf{w}^*}) & \text{(erreur de généralisation)} \\
+ & R(f_{\mathbf{w}^*}) & - & R^* & \text{(erreur d'approximation)}
\end{array}
$$

On utilise un algorithme d'optimisation pour trouver un **w**, on veut que

$$\widehat{R}(f_{\mathbf{w}}) - \inf_{\mathbf{w}} \widehat{R}(f_{\mathbf{w}})$$

soit le plus faible possible.

Idéalement, on voudrait que cette quantité soit nulle ou au moins on voudrait la borner supérieurement.

[3] Pour alléger les notations, on oublie **b**.

Paysage de la fonction objectif : Introduction

On suppose que $\mathbf{w} \longmapsto \widehat{R}(f_{\mathbf{w}})$ est C^2 partout. On a

$$\widehat{R}(f_{\mathbf{w}}) = \widehat{R}(f_{\mathbf{w}^*}) + \langle \nabla_{\mathbf{w}} \widehat{R}(f_{\mathbf{w}^*}), \mathbf{w} - \mathbf{w}^* \rangle + \frac{1}{2} \langle \nabla^2_{\mathbf{w}} \widehat{R}(f_{\mathbf{w}^*})(\mathbf{w} - \mathbf{w}^*), \mathbf{w} - \mathbf{w}^* \rangle + o(\|\mathbf{w} - \mathbf{w}^*\|^2)$$

On distingue:

- **$\mathbf{w}^*$ est un minimiseur global:**
  - $\forall \mathbf{w}, \qquad \widehat{R}(f_{\mathbf{w}^*}) \leq \widehat{R}(f_{\mathbf{w}})$
  - $\widehat{R}(f_{\mathbf{w}^*}) = \min_{\mathbf{w}} \widehat{R}(f_{\mathbf{w}})$
- **$\mathbf{w}^*$ est un minimiseur local:**
  - Il existe un voisinage ouvert $\mathcal{O}$ de $\mathbf{w}^*$ tel que

$$\forall \mathbf{w} \in \mathcal{O}, \qquad \widehat{R}(f_{\mathbf{w}^*}) \leq \widehat{R}(f_{\mathbf{w}})$$

- **$\mathbf{w}^*$ est un point critique du second ordre:**
  - On a

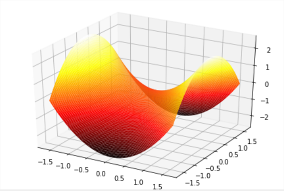
$$\nabla \widehat{R}(f_{\mathbf{w}^*}) = 0 \qquad \text{et} \qquad \nabla^2 \widehat{R}(f_{\mathbf{w}^*}) \succeq 0$$

- **$\mathbf{w}^*$ est un point critique du premier ordre:**
  - On a

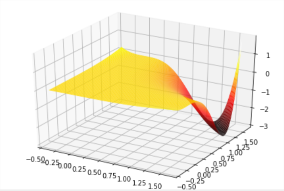
$$\nabla \widehat{R}(f_{\mathbf{w}^*}) = 0$$

Paysage de la fonction objectif : Introduction

- ***w*** * **est un point selle** si c'est un point critique qui n'est ni un minimiseur local, ni un maximiseur local
  - **Un point selle *w*** * **est strict :** si ce n'est pas un point critique du second ordre (i.e., le Hessian a une v.p. négative).
  - **Un point selle *w*** * **est non-strict:** si c'est un point critique du second ordre (i.e. le Hessian est semi-defini positif et a une v.p. égale à 0. Typiquement, un terme d'ordre supérieur en fait un point selle.).



(a) Point selle strict



(b) Point selle non-strict

Paysage de la fonction objectif : Introduction

Optimisation non convexe avec les mains

Pour des fonctions non-convexes, on sait montrer que

- **dans un cadre assez vaste**, l'algorithme du gradient (ou gradient stochastique) **converge vers un point critique du premier ordre**
- **dans un cadre plus restreint**, l'algorithme du gradient **converge vers un point critique du second ordre**
- Pour le gradient stochastic, **sans vitesse de convergence**, que les itérés convergent vers un minimiseur local.

- S. Gadat, F. Panloup, S. Saadane. "Stochastic heavy ball." Electronic Journal of Statistics 12.1 (2018): 461-529.
- J. Lee, M. Simchowitz, M. Jordan, B. Recht." Gradient Descent Converges to Minimizers." COLT 2016.

Outline

Paysage pour les réseaux linéaires

Pointeurs bibliographiques

- Baldi, Hornik, "Neural networks and principal component analysis: Learning from examples without local minima", Neural networks, 1989.
- Baldi, Hornik, "Learning in linear neural networks: A survey", IEEE Transactions on neural networks, 1995.
- Kawaguchi, "Deep learning without poor local minima", NeurIPS 2016
- .... (Notamment dans les groupes de S. Arora à Princeton, de F. Bach à l'ENS)

Paysage pour les réseaux linéaires

En collaboration avec

El-Mehdi Achour
Postdoc, RWTH, Aachen university, Allemagne

Sébastien Gerchinovitz
Chercheur DEEL, IR Saint-Exupéry

Paysage pour les réseaux linéaires

On note

- $X \in \mathbb{R}^{d_x \times n}$ et $Y \in \mathbb{R}^{d_y \times n}$ les matrices contenant les données.
- $\widehat{R}(\mathbf{W}) = \sum_{i=1}^m \| W_H W_{H-1} \cdots W_2 W_1 x_i - y_i \|_2^2 = \| W_H \cdots W_1 X - Y \|^2$
- $$\Sigma_{XX} = \sum_{i=1}^n x_i x_i^T = XX^T \in \mathbb{R}^{d_x \times d_x} \quad , \quad \Sigma_{YY} = \sum_{i=1}^n y_i y_i^T = YY^T \in \mathbb{R}^{d_y \times d_y},$$
  $$\Sigma_{XY} = \sum_{i=1}^n x_i y_i^T = XY^T \in \mathbb{R}^{d_x \times d_y} \quad , \quad \Sigma_{YX} = \sum_{i=1}^n y_i x_i^T = YX^T \in \mathbb{R}^{d_y \times d_x},$$
- $$\Sigma^{1/2} = \Sigma_{YX} \Sigma_{XX}^{-1} X \in \mathbb{R}^{d_y \times n},$$
  sa SVD
  $$\Sigma^{1/2} = U \Delta V^T,$$
  avec $U \in \mathbb{R}^{d_y \times d_y}$, $\Delta = \mathrm{diag}((\delta_i)_{i=1..d_y}) \in \mathbb{R}^{d_y \times n}$, $V \in \mathbb{R}^{n \times n}$.
- $r_{max} = \min(d_x, n_1, \ldots, n_{H-1}, d_y)$

Paysage pour les réseaux linéaires

On suppose

- $d_y \leq d_x \leq n$
- Σ_{XX} est inversible et Σ_{XY} est de rang plein
- les valeurs singulières de $\Sigma^{1/2}$ sont distinctes

Lemme (Inférence et valeurs critiques)

Soit $\mathbf{W} = (W_1, \ldots, W_H)$ *un point critique du premier ordre de* $\widehat{R}$ *et* $r = rk(W_H \cdots W_1)$.
Il existe un unique sous-ensemble $\mathscr{S} \subset [\![1, d_y]\!]$ *de taille* r *tel que:*

$$W_H \cdots W_1 = U_{\mathscr{S}} U_{\mathscr{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}.$$

La valeur critique correspondante vaut $\widehat{R}(\mathbf{W}) = \mathrm{tr}(\Sigma_{YY}) - \sum_{i \in \mathscr{S}} \delta_i^2$.
On dit que le point critique $\mathbf{W}$ *est associé à* $\mathscr{S}$.

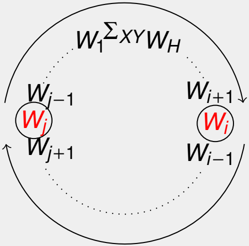
Proposition

Pour tout $\mathscr{S} \subset [\![1, d_y]\!]$ *de taille* $r \in [\![0, r_{max}]\!]$, *il existe un point critique* $\mathbf{W}$ *associé à* $\mathscr{S}$.

Paysage pour les réseaux linéaires

- **Pivot** $(i,j) \in [\![1,H]\!]^2$, avec $i > j$
- **Blocs complémentaires du pivot** (i,j):

Premier bloc complémentaire : $W_{j-1} \cdots W_1 \Sigma_{XY} W_H \cdots W_{i+1}$



Second bloc complémentaire : $W_{i-1} \cdots W_{j+1}$

- **Pivot tenu pour W** : L'un des blocs complémentaires est de rang $\mathrm{rk}(W_H \ldots W_1)$
- **W est point critique tenu**: **W** est un point critique et tous les pivots sont tenus pour **W**

Paysage pour les réseaux linéaires

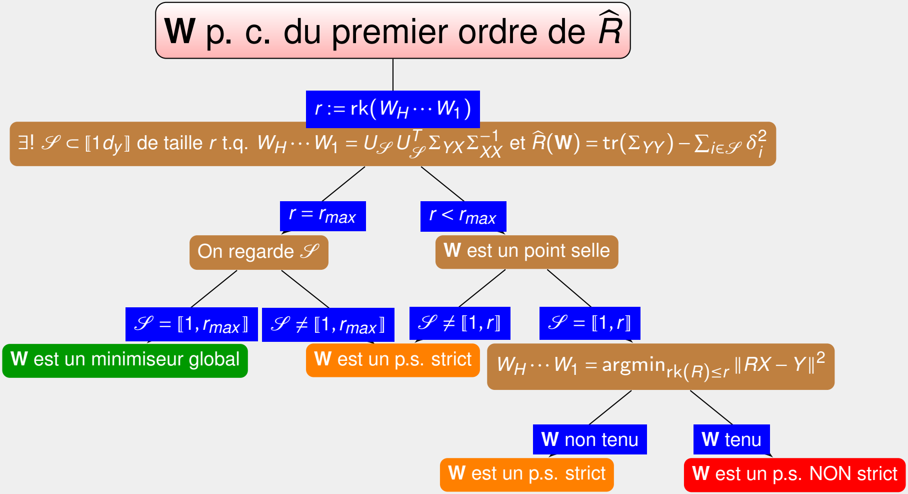


Figure: Theorem [E. Achour et al.]

Paysage pour les réseaux linéaires

Proposition (E. Achour et al.)

- *Pour $H = 2$, il n'existe pas de point selle non-strict.*
- *Pour $H \geq 3$, pour tout $r < r_{max}$, il existe des points critiques associés à $[\![1, r]\!]$ tenus et non-tenus.*

Proposition (E. Achour et al.)

On note $\mathscr{S}_{max} = [\![1, r_{max}]\!]$ et $Q_{max} = [\![1, d_y]\!] \setminus \mathscr{S}_{max} = [\![r_{max} + 1, d_y]\!]$.
W *est un* **minimiseur global** *de $\widehat{R}$* **si et seulement si** *il existe des matrices inversibles $D_{H-1} \in \mathbb{R}^{d_{H-1} \times d_{H-1}}, \ldots, D_1 \in \mathbb{R}^{d_1 \times d_1}$, et des matrices $A_R \in \mathbb{R}^{(d_y - r_{max}) \times (d_{H-1} - r_{max})}$, $(W_h)_{DR} \in \mathbb{R}^{(d_h - r_{max}) \times (d_{h-1} - r_{max})}$ pour $h \in [\![2, H-1]\!]$, et $M_D \in \mathbb{R}^{(d_1 - r_{max}) \times d_x}$ tels que:*

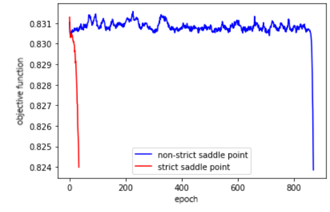
$$W_H = [U_{\mathscr{S}_{max}}, U_{Q_{max}} A_R] D_{H-1}^{-1}$$

$$W_h = D_h \begin{bmatrix} I_{r_{max}} & 0 \\ 0 & (W_h)_{DR} \end{bmatrix} D_{h-1}^{-1} \qquad \forall h \in [\![2, H-1]\!]$$

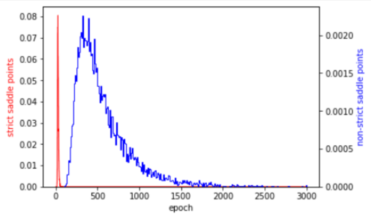
$$W_1 = D_1 \begin{bmatrix} U_{\mathscr{S}_{max}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ M_D \end{bmatrix}.$$

Paysage pour les réseaux linéaires

Empiriquement, on trouve:



(a) Initialisation au voisinage d'un point selle strict vs non-strict



(b) Histogramme des epoch d'échappement

Paysage pour les réseaux linéaires

Conclusion [E. Achour et al.]

- **Classification** de l'ensemble des points critiques en: minimiseurs globaux; points selles stricts; point selles non-stricts.
- Tout **point critique du second ordre** qui n'est pas un minimiseur global conduit à une solution de la régression linéaire sous **contrainte de rang**.
- Les points selles non-stricts sont associés avec $\mathbf{r_{max}}$ **valeurs plateau** pour le risque empirique
- **Paramétrisation des minimiseurs globaux**

Outline

Paysage pour les réseaux linéaires (cas à 1 couche cachée)

On simplifie

- Cas $H = 2$
- On note pour $A \in \mathbb{R}^{n_2 \times n_1}$ et $B \in \mathbb{R}^{n_1 \times n_0}$

$$E(A,B) = \sum_{i=1}^{L} \|y_i - ABx_i\|^2$$

- On note pour $D \in \mathbb{R}^{n_2 \times n_0}$

$$F(D) = \sum_{i=1}^{L} \|y_i - Dx_i\|^2$$

- On note

$$\Sigma_{XX} = \sum_{i=1}^{L} x_i x_i^T \in \mathbb{R}^{n_0 \times n_0} \qquad , \qquad \Sigma_{XY} = \sum_{i=1}^{L} x_i y_i^T \in \mathbb{R}^{n_0 \times n_2}$$

$$\Sigma_{YX} = \sum_{i=1}^{L} y_i x_i^T \in \mathbb{R}^{n_2 \times n_0} \qquad , \qquad \Sigma_{YY} = \sum_{i=1}^{L} y_i y_i^T \in \mathbb{R}^{n_2 \times n_2}$$

Paysage pour les réseaux linéaires (cas à 1 couche cachée)

Remarques

On a

1. Pour tout $C \in \mathbb{R}^{n_1 \times n_1}$ inversible, $AB = (AC)(C^{-1}B) = A'B'$
2. Si Σ_{XX} est inversible alors

$$\Sigma_{YX}\Sigma_{XX}^{-1} \in \mathrm{argmin}_D F(D)$$

3. Soit $M \in \mathbb{R}^{n \times p}$ avec $p \leq n$ de rang p. Pour tout $x \in \mathbb{R}^n$, la projection $P_M(x)$ de x sur l'espace vectoriel généré par les colonnes de M vaut

$$P_M(x) = M(M^T M)^{-1} M^T x$$

Paysage pour les réseaux linéaires (cas à 1 couche cachée)

Lemme 1

Si Σ_{XX} est inversible. Soient $A \in \mathbb{R}^{n_2 \times n_1}$ telle que $\mathrm{rang}(A) = n_1$ et $B \in \mathbb{R}^{n_1 \times n_0}$. Alors, (A, B) est un point critique du premier ordre de E si et seulement si

$$AB\Sigma_{XX}B^T = \Sigma_{YX}B^T \qquad \text{et} \qquad B = (A^T A)^{-1} A^T \Sigma_{YX}\Sigma_{XX}^{-1}$$

Lemme 2

Si Σ_{XX} est inversible. Soient $A \in \mathbb{R}^{n_2 \times n_1}$ telle que $\mathrm{rang}(A) = n_1$ et $B \in \mathbb{R}^{n_1 \times n_0}$.
Les deux assertions suivantes sont équivalentes:

1. (A, B) est un point critique du premier ordre de E
2. $$AB = P_A \Sigma_{YX}\Sigma_{XX}^{-1}$$
   et
   $$P_A\Sigma = P_A \Sigma P_A = \Sigma P_A$$
   pour
   $$\Sigma = \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY} \quad \in \mathbb{R}^{n_2 \times n_2}$$

Paysage pour les réseaux linéaires (cas à 1 couche cachée)

On diagonalise (Σ est symétrique)

$$\Sigma = U \Lambda U^T$$

avec $\Lambda \in \mathbb{R}^{n_2 \times n_2}$ diagonale et $U \in \mathbb{R}^{n_2 \times n_2}$ unitaire.
Pour $\mathscr{S} \subset \{1, \cdots, n_2\}$, on note $U_{\mathscr{S}}$ la matrice extraite de U en prenant les colonnes d'indice dans $\mathscr{S}$.

Proposition 1: Paramétrisation des points critiques

Si Σ_{XX} est inversible. Soient $A \in \mathbb{R}^{n_2 \times n_1}$ et $B \in \mathbb{R}^{n_1 \times n_0}$ avec A telle que $\mathrm{rang}(A) = n_1$.

On suppose que les valeurs propres de Σ sont distinctes.

Alors, (A, B) est un point critique du premier ordre de E si et seulement si il existe $C \in \mathbb{R}^{n_1 \times n_1}$ inversible et $\mathscr{S} \subset \{1, \cdots, n_2\}$ de taille n_1 tels que

$$A = U_{\mathscr{S}} C \qquad \text{et} \qquad B = C^{-1} U_{\mathscr{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$$

Paysage pour les réseaux linéaires (cas à 1 couche cachée)

Pour simplifier les notations, on suppose les valeurs propres de Σ ordonnées:

$$\lambda_1 > \lambda_2 > \cdots > \lambda_{n_2}$$

Proposition 2: minimiseur global ⇔ minimiseur local

Si Σ_{XX} est inversible. Soient $A \in \mathbb{R}^{n_2 \times n_1}$ et $B \in \mathbb{R}^{n_1 \times n_0}$ avec A telle que $\operatorname{rang}(A) = n_1$.

On suppose que les valeurs propres de Σ sont distinctes.

1. pour tout point critique du premier ordre (A, B) de E et pour $\mathscr{S}$ définissant A et B (voir la proposition précédente), on a
   1. $$AB = P_{U_{\mathscr{S}}} \Sigma_{YX} \Sigma_{XX}^{-1}$$
   2. $$E(A, B) = \operatorname{trace}(\Sigma_{YY}) - \sum_{i \in \mathscr{S}} \lambda_i$$
2. Si (A, B) est un minimiseur global alors c'est un point critique du premier ordre associé à $\mathscr{S} = \{1, \cdots, n_1\}$.
3. (A, B) **est minimiseur local si et seulement si c'est un minimiseur global**.

Merci pour votre attention !