Ouvrons la boîte noire : Reconstruction de réseaux de neurones parcimonieux par échantillonnage

Valérie Castin[1], Rémi Gribonval[2]

[1]ENS PSL, Paris [2]Inria et ENS de Lyon, Lyon

Palais des Congrès de Strasbourg, 28 août 2025

Parcimonie et distillation réduisent les coûts des modèles

- Modèles actuels : 10^{9-12} paramètres, données d'entraînement massives
  $\rightarrow$ **coûteux** à entraîner, stocker, évaluer

Parcimonie et distillation réduisent les coûts des modèles

- Modèles actuels : 10^{9-12} paramètres, données d'entraînement massives
  $\rightarrow$ **coûteux** à entraîner, stocker, évaluer

- Alternative : modèles **parcimonieux**, i.e. peu de poids non nuls

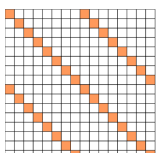
$$\begin{pmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 4 & 0 & 3 & 0 \end{pmatrix}$$

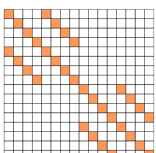
Parcimonie et distillation réduisent les coûts des modèles

- Modèles actuels : 10^{9-12} paramètres, données d'entraînement massives
  $\rightarrow$ **coûteux** à entraîner, stocker, évaluer

- Alternative : modèles **parcimonieux**, i.e. peu de poids non nuls, **structurés**

$$\begin{pmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 4 & 0 & 3 & 0 \end{pmatrix}$$



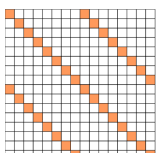


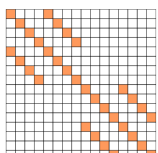
Parcimonie et distillation réduisent les coûts des modèles

- Modèles actuels : 10^{9-12} paramètres, données d'entraînement massives
  $\rightarrow$ **coûteux** à entraîner, stocker, évaluer

- Alternative : modèles **parcimonieux**, i.e. peu de poids non nuls, **structurés**

$$\begin{pmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 4 & 0 & 3 & 0 \end{pmatrix}$$





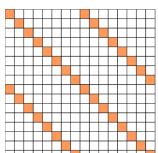
- Souvent entraînés par **distillation** [1], i.e. sur la sortie d'un grand modèle "enseignant"

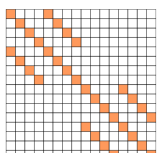
Parcimonie et distillation réduisent les coûts des modèles

- Modèles actuels : 10^{9-12} paramètres, données d'entraînement massives
  $\rightarrow$ **coûteux** à entraîner, stocker, évaluer

- Alternative : modèles **parcimonieux**, i.e. peu de poids non nuls, **structurés**

$$\begin{pmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 4 & 0 & 3 & 0 \end{pmatrix}$$





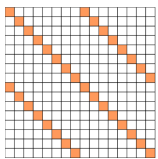
- Souvent entraînés par **distillation** [1], i.e. sur la sortie d'un grand modèle "enseignant"
  - accès aux probabilités prédites pour chaque classe $\rightarrow$ meilleure généralisation, avec moins de données

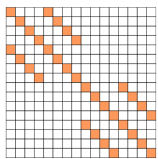
Parcimonie et distillation réduisent les coûts des modèles

- Modèles actuels : 10^{9-12} paramètres, données d'entraînement massives
  $\rightarrow$ **coûteux** à entraîner, stocker, évaluer

- Alternative : modèles **parcimonieux**, i.e. peu de poids non nuls, **structurés**

$$\begin{pmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 4 & 0 & 3 & 0 \end{pmatrix}$$





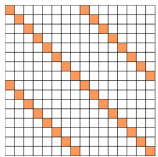
- Souvent entraînés par **distillation** [1], i.e. sur la sortie d'un grand modèle "enseignant"
  - accès aux probabilités prédites pour chaque classe $\rightarrow$ meilleure généralisation, avec moins de données
  - modèle distillé = résumé du modèle enseignant à moindre coût

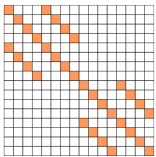
Parcimonie et distillation réduisent les coûts des modèles

- Modèles actuels : 10^{9-12} paramètres, données d'entraînement massives
  $\rightarrow$ **coûteux** à entraîner, stocker, évaluer

- Alternative : modèles **parcimonieux**, i.e. peu de poids non nuls, **structurés**

$$\begin{pmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 4 & 0 & 3 & 0 \end{pmatrix}$$





- Souvent entraînés par **distillation** [1], i.e. sur la sortie d'un grand modèle "enseignant"
  - accès aux probabilités prédites pour chaque classe $\rightarrow$ meilleure généralisation, avec moins de données
  - modèle distillé = résumé du modèle enseignant à moindre coût

[1] Distilling the Knowledge in a Neural Network, G. Hinton et al., 2015

La reconstruction parcimonieuse nécessite moins d'échantillons

Distillation

- f réseau enseignant
- R_θ réseau élève, $\theta \in \Theta$ à apprendre à partir de $(x_i, f(x_i))_{1 \leq i \leq n}$

Reconstruction

- R_{θ^*} réseau à reconstruire
- Objectif : trouver θ tq $R_\theta = R_{\theta^*}$ en échantillonnant R_{θ^*}

La reconstruction parcimonieuse nécessite moins d'échantillons

Distillation

- f réseau enseignant
- R_θ réseau élève, $\theta \in \Theta$ à apprendre à partir de $(x_i, f(x_i))_{1 \leq i \leq n}$

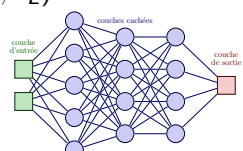
Reconstruction

- R_{θ^*} réseau à reconstruire
- Objectif : trouver θ tq $R_\theta = R_{\theta^*}$ en échantillonnant R_{θ^*}

Prenons R_θ perceptron (MLP) de params $\theta = (W_1, \ldots, W_L, b_1, \ldots, b_L)$:

$$y_0(x) = x, \quad \begin{cases} z_\ell(x) = W_\ell y_{\ell-1}(x) + b_\ell \\ y_\ell(x) = \mathrm{ReLU}(z_\ell(x)) \end{cases}, \quad R_\theta(x) \coloneqq z_L(x)$$

où $\mathrm{ReLU}(t) \coloneqq \max(t, 0)$



La reconstruction parcimonieuse nécessite moins d'échantillons

Distillation

- f réseau enseignant
- R_θ réseau élève, $\theta \in \Theta$ à apprendre à partir de $(x_i, f(x_i))_{1 \leq i \leq n}$

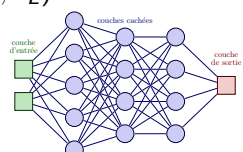
Reconstruction

- R_{θ^*} réseau à reconstruire
- Objectif : trouver θ tq $R_\theta = R_{\theta^*}$ en échantillonnant R_{θ^*}

Prenons R_θ perceptron (MLP) de params $\theta = (W_1, \ldots, W_L, b_1, \ldots, b_L)$:

$$y_0(x) = x, \quad \begin{cases} z_\ell(x) = W_\ell y_{\ell-1}(x) + b_\ell \\ y_\ell(x) = \mathrm{ReLU}(z_\ell(x)) \end{cases}, \quad R_\theta(x) \coloneqq z_L(x)$$

où $\mathrm{ReLU}(t) \coloneqq \max(t, 0)$



Si Θ assez parcimonieux et il existe θ^* tq $f = R_{\theta^*}$, alors n petit suffit à trouver θ tq $R_\theta = R_{\theta^*}$ (invariances de R)

La reconstruction parcimonieuse nécessite moins d'échantillons

Distillation

- f réseau enseignant
- R_θ réseau élève, $\theta \in \Theta$ à apprendre à partir de $(x_i, f(x_i))_{1 \le i \le n}$

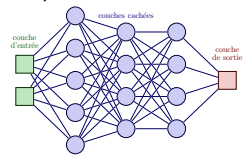
Reconstruction

- R_{θ^*} réseau à reconstruire
- Objectif : trouver θ tq $R_\theta = R_{\theta^*}$ en échantillonnant R_{θ^*}

Prenons R_θ perceptron (MLP) de params $\theta = (W_1, \dots, W_L, b_1, \dots, b_L)$:

$$y_0(x) = x, \quad \begin{cases} z_\ell(x) = W_\ell y_{\ell-1}(x) + b_\ell \\ y_\ell(x) = \mathrm{ReLU}(z_\ell(x)) \end{cases}, \quad R_\theta(x) \coloneqq z_L(x)$$

où $\mathrm{ReLU}(t) \coloneqq \max(t, 0)$

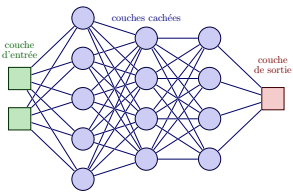


Si Θ assez parcimonieux et il existe θ^* tq $f = R_{\theta^*}$, alors n petit suffit à trouver θ tq $R_\theta = R_{\theta^*}$ (invariances de R)

$$W_\ell = D_\ell W_\ell^* D_{\ell-1}^{-1} \quad \text{et} \quad b_\ell = D_\ell b_\ell^* \quad \text{pour} \quad D_1, \dots, D_{L-1} > 0 \quad \Rightarrow R_\theta = R_{\theta^*}$$

Le problème de reconstruction d'un MLP parcimonieux

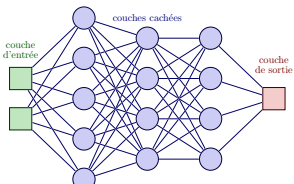
f MLP ReLU de paramètres inconnus $(W_1, \dots, W_L, b_1, \dots, b_L)$



$$y_\ell(x) = \mathrm{ReLU}(z_\ell(x))$$

Le problème de reconstruction d'un MLP parcimonieux

f MLP ReLU de paramètres inconnus $(W_1, \dots, W_L, b_1, \dots, b_L)$

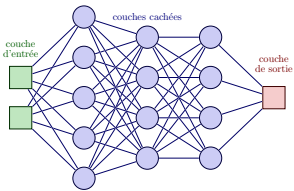


$$y_\ell(x) = \mathrm{ReLU}(z_\ell(x))$$

Jacobienne : $Df(x) = W_L A_{L-1} \dots A_1 W_1$ où $(A_\ell)_{ii} = \begin{cases} 1 \text{ si } z_\ell(x)_i > 0 \\ 0 \text{ sinon} \end{cases}$

Le problème de reconstruction d'un MLP parcimonieux

f MLP ReLU de paramètres inconnus $(W_1, \dots, W_L, b_1, \dots, b_L)$



$$y_\ell(x) = \mathrm{ReLU}(z_\ell(x))$$

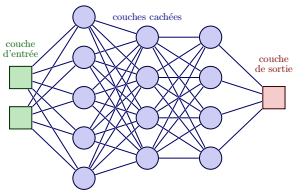
Jacobienne : $Df(x) = W_L A_{L-1} \dots A_1 W_1$ où $(A_\ell)_{ii} = \begin{cases} 1 \text{ si } z_\ell(x)_i > 0 \\ 0 \text{ sinon} \end{cases}$

Hypothèses de travail :

- f "parcimonieux" (à définir)
- on peut échantillonner librement f et Df

Le problème de reconstruction d'un MLP parcimonieux

f MLP ReLU de paramètres inconnus $(W_1, \dots, W_L, b_1, \dots, b_L)$



$$y_\ell(x) = \mathrm{ReLU}(z_\ell(x))$$

Jacobienne : $Df(x) = W_L A_{L-1} \dots A_1 W_1$ où $(A_\ell)_{ii} = \begin{cases} 1 \text{ si } z_\ell(x)_i > 0 \\ 0 \text{ sinon} \end{cases}$

Hypothèses de travail :

- f "parcimonieux" (à définir)
- on peut échantillonner librement f et Df

Objectif : déterminer θ tq $R_\theta = f$ avec garanties, i.e. **sans descente de gradient**

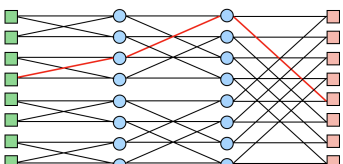
I - La structure de parcimonie : les perceptrons multi-arbres (*multitree*)

Perceptron multi-arbres (MTP) $\Leftrightarrow$ unicité des chemins

Définition

Un perceptron est multi-arbres (MTP) s'il y a au plus 1 chemin reliant chaque couple de neurones de son graphe.

Exemple :

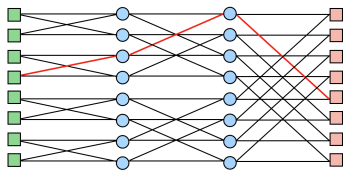


Perceptron multi-arbres (MTP) $\Leftrightarrow$ unicité des chemins

Définition

Un perceptron est multi-arbres (MTP) s'il y a au plus 1 chemin reliant chaque couple de neurones de son graphe.

Exemple :



Caractérisation utile via la Jacobienne

Si f MTP ReLU de params $(W_1, \dots, W_L, b_1, \dots, b_L)$:

$$Df(x)_{i_L i_0} = \mathbf{1}_{\{x \text{ active } i_0 \to i_L\}} (W_L \dots W_1)_{i_L i_0}$$

Factorisation de produits multi-arbres

Proposition [Le, Zheng, Riccietti, Gribonval, *Fast learning of fast transforms, with guarantees*]

(W_1, W_2) poids inconnus d'un MTP à 1 couche cachée. Si :

- supports de (W_1, W_2) connus
- produit $W_2 W_1$ connu

Factorisation de produits multi-arbres

Proposition [Le, Zheng, Riccietti, Gribonval, *Fast learning of fast transforms, with guarantees*]

(W_1, W_2) poids inconnus d'un MTP à 1 couche cachée. Si :

- supports de (W_1, W_2) connus
- produit $W_2 W_1$ connu

il existe un algorithme qui permet de construire $(\hat{W}_1, \hat{W}_2)$ tq

$$\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$$

avec D diagonale inversible

Factorisation de produits multi-arbres

Proposition [Le, Zheng, Riccietti, Gribonval, *Fast learning of fast transforms, with guarantees*]

(W_1, W_2) poids inconnus d'un MTP à 1 couche cachée. Si :

- supports de (W_1, W_2) connus
- produit $W_2 W_1$ connu

il existe un algorithme qui permet de construire $(\hat{W}_1, \hat{W}_2)$ tq

$$\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$$

avec D diagonale inversible

Application itérative : permet de factoriser un produit MTP $W_L \dots W_1$ en $(\hat{W}_1, \dots, \hat{W}_L)$ tq

$$\hat{W}_\ell = D_\ell W_\ell D_{\ell-1}^{-1}, \quad D_\ell \text{ diagonale inversible pour } 1 \leq \ell \leq L-1$$

II - Cas d'un MTP à une couche cachée

Algorithme de reconstruction : cas à une couche cachée

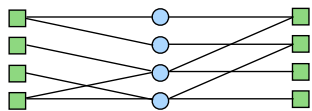
À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$, $\quad (W_1, W_2, b)$ MTP de supports connus

Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$, (W_1, W_2, b) MTP de supports connus

Propriété MTP

$$Df(x) = \sum_{h=1}^{H} \mathbf{1}_{\{x \text{ active } h\}} \underbrace{W_2[:,h] W_1^\top[h,:]}_{\text{à supports disjoints}}$$

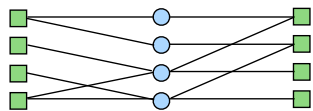


Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$, (W_1, W_2, b) MTP de supports connus

Propriété MTP

$$Df(x) = \sum_{h=1}^{H} \mathbf{1}_{\{x \text{ active } h\}} \underbrace{W_2[:,h] W_1^\top[h,:]}_{\text{à supports disjoints}}$$



Reconstruction des poids :

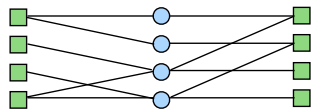
1. Échantillonner $x \sim \mathcal{N}(0, I_d)$

Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$, (W_1, W_2, b) MTP de supports connus

Propriété MTP

$$Df(x) = \sum_{h=1}^{H} \mathbf{1}_{\{x \text{ active } h\}} \underbrace{W_2[:,h] W_1^\top[h,:]}_{\text{à supports disjoints}}$$



Reconstruction des poids :

1. Échantillonner $x \sim \mathcal{N}(0, I_d)$

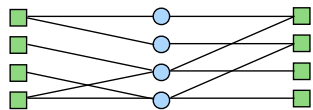
2. Répéter $x \leftarrow 2x$ jusqu'à ce que tout neurone caché soit activé par x ou $-x$, i.e. $|\langle L_h(W_1), x\rangle| > |b_h|$ pour tout $1 \le h \le H$. Alors
$$Df(x) + Df(-x) = W_2 W_1$$

Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$, $\quad (W_1, W_2, b)$ MTP de supports connus

Propriété MTP

$$Df(x) = \sum_{h=1}^{H} \mathbf{1}_{\{x \text{ active } h\}} \underbrace{W_2[:,h] W_1^\top[h,:]}_{\text{à supports disjoints}}$$



Reconstruction des poids :

3. Factoriser $Df(x) + Df(-x) = \hat{W}_2 \hat{W}_1$ tq

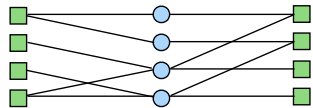
$$\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1 \quad \text{pour } D \text{ diagonale inversible}$$

Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$, $\quad (W_1, W_2, b)$ MTP de supports connus

Propriété MTP

$$Df(x) = \sum_{h=1}^{H} \mathbf{1}_{\{x \text{ active } h\}} \underbrace{W_2[:,h] W_1^\top[h,:]}_{\text{à supports disjoints}}$$



Reconstruction des poids :

3. Factoriser $Df(x) + Df(-x) = \hat{W}_2 \hat{W}_1$ tq

$$\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1 \quad \text{pour } D \text{ diagonale inversible}$$

4. Si $\langle L_h(\hat{W}_1), x \rangle$ a le mauvais signe, faire $\begin{cases} L_h(\hat{W}_1) \leftarrow -L_h(\hat{W}_1) \\ C_h(\hat{W}_2) \leftarrow -C_h(\hat{W}_2) \end{cases}$

Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$

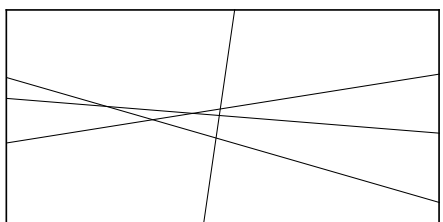
On a obtenu $(\hat{W}_1, \hat{W}_2)$ tq : $\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$ pour D diagonale > 0

Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$

On a obtenu $(\hat{W}_1, \hat{W}_2)$ tq : $\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$ pour D diagonale > 0

Reconstruction des biais :

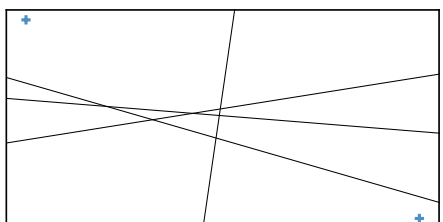


Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$

On a obtenu $(\hat{W}_1, \hat{W}_2)$ tq : $\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$ pour D diagonale > 0

Reconstruction des biais :

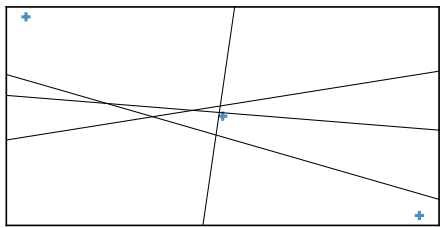


Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$

On a obtenu $(\hat{W}_1, \hat{W}_2)$ tq : $\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$ pour D diagonale > 0

Reconstruction des biais :

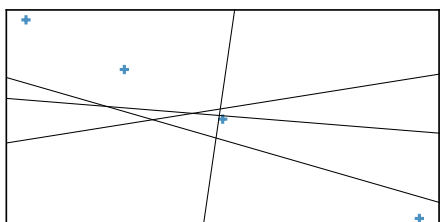


Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$

On a obtenu $(\hat{W}_1, \hat{W}_2)$ tq : $\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$ pour D diagonale > 0

Reconstruction des biais :

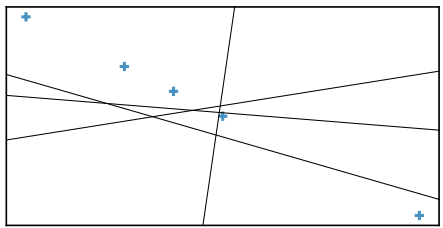


Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$

On a obtenu $(\hat{W}_1, \hat{W}_2)$ tq : $\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$ pour D diagonale > 0

Reconstruction des biais :

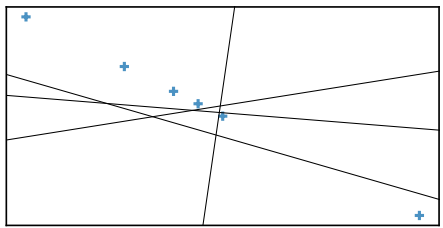


Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$

On a obtenu $(\hat{W}_1, \hat{W}_2)$ tq : $\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$ pour D diagonale > 0

Reconstruction des biais :

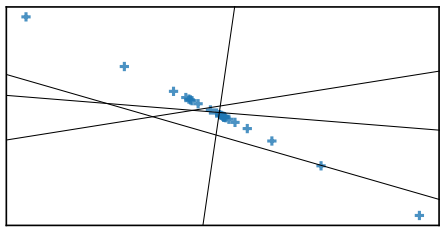


Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$

On a obtenu $(\hat{W}_1, \hat{W}_2)$ tq : $\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$ pour D diagonale > 0

Reconstruction des biais :

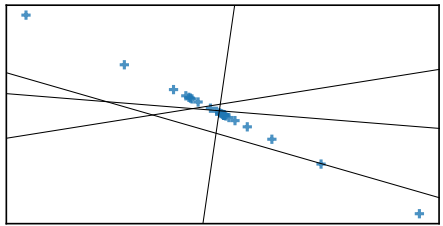


Algorithme de reconstruction : cas à une couche cachée

À reconstruire : $f(x) = W_2 \,\mathrm{ReLU}(W_1 x + b)$

On a obtenu $(\hat{W}_1, \hat{W}_2)$ tq : $\hat{W}_2 = W_2 D^{-1}, \quad \hat{W}_1 = D W_1$ pour D diagonale > 0

Reconstruction des biais :



Proposition

L'algorithme retourne $(\hat{W}_1, \hat{W}_2, \hat{b})$ induisant la même réalisation que (W_1, W_2, b)

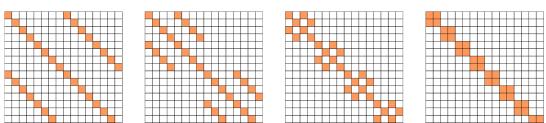
III - Cas d'un MTP *papillon* multi-couches

Le cas papillon dyadique

Définition

$(W_1, \dots, W_L)$ a une structure papillon dyadique si

$$\forall 1 \leq \ell \leq L, \quad \operatorname{supp} W_\ell \subseteq I_{2^{L-\ell}} \otimes \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes I_{2^{\ell-1}}$$



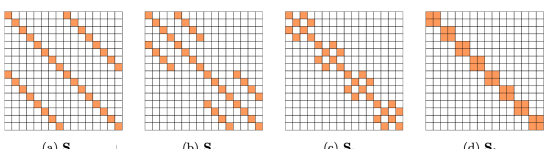
(a) $\mathbf{S}_4$ (b) $\mathbf{S}_3$ (c) $\mathbf{S}_2$ (d) $\mathbf{S}_1$

Le cas papillon dyadique

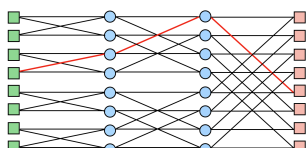
Définition

$(W_1, \ldots, W_L)$ a une structure papillon dyadique si

$$\forall 1 \leq \ell \leq L, \quad \operatorname{supp} W_\ell \subseteq I_{2^{L-\ell}} \otimes \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes I_{2^{\ell-1}}$$



(a) $\mathbf{S}_4$ (b) $\mathbf{S}_3$ (c) $\mathbf{S}_2$ (d) $\mathbf{S}_1$

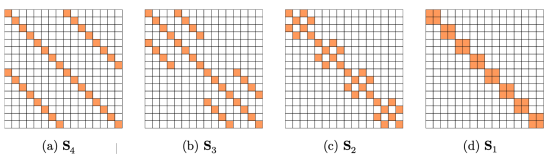


Le cas papillon dyadique

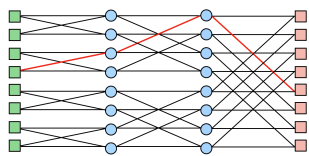
Définition

$(W_1, \dots, W_L)$ a une structure papillon dyadique si

$$\forall 1 \leq \ell \leq L, \quad \operatorname{supp} W_\ell \subseteq I_{2^{L-\ell}} \otimes \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes I_{2^{\ell-1}}$$



(a) $\mathbf{S}_4$ (b) $\mathbf{S}_3$ (c) $\mathbf{S}_2$ (d) $\mathbf{S}_1$



Remarque importante

Si $L \geq 3$, tous les chemins ne sont plus forcément activables $\rightarrow$ impossible de reconstruire le produit $W_L \dots W_1$

Algorithme de reconstruction : cas papillon multi-couches

À reconstruire : $f(x) = W_L \,\mathrm{ReLU}(W_{L-1} \dots \mathrm{ReLU}(W_1 x))$ (sans biais)

Algorithme de reconstruction : cas papillon multi-couches

À reconstruire : $f(x) = W_L \,\mathrm{ReLU}(W_{L-1} \dots \mathrm{ReLU}(W_1 x))$ (sans biais)

Reconstruction des poids :

1. Échantillonnage : activer le plus de chemins possible $\rightarrow$ échantillons x i.i.d. gaussiens

Algorithme de reconstruction : cas papillon multi-couches

À reconstruire : $f(x) = W_L \,\mathrm{ReLU}(W_{L-1} \dots \mathrm{ReLU}(W_1 x))$ (sans biais)

Reconstruction des poids :

1. Échantillonnage : activer le plus de chemins possible $\rightarrow$ échantillons x i.i.d. gaussiens
2. Reconstruire une version masquée $M \odot W_L \dots W_1$ du produit des poids

Algorithme de reconstruction : cas papillon multi-couches

À reconstruire : $f(x) = W_L \,\mathrm{ReLU}(W_{L-1} \dots \mathrm{ReLU}(W_1 x))$ (sans biais)

Reconstruction des poids :

1. Échantillonnage : activer le plus de chemins possible $\rightarrow$ échantillons x i.i.d. gaussiens
2. Reconstruire une version masquée $M \odot W_L \dots W_1$ du produit des poids
3. Factorisation adaptée aux valeurs manquantes : donne $(\hat{W}_1, \dots, \hat{W}_L)$ papillon tq

$$M \odot (\hat{W}_L \dots \hat{W}_1 - W_L \dots W_1) = 0$$

Algorithme de reconstruction : cas papillon multi-couches

À reconstruire : $f(x) = W_L \,\mathrm{ReLU}(W_{L-1} \dots \mathrm{ReLU}(W_1 x))$ (sans biais)

Reconstruction des poids :

1. Échantillonnage : activer le plus de chemins possible $\rightarrow$ échantillons x i.i.d. gaussiens
2. Reconstruire une version masquée $M \odot W_L \dots W_1$ du produit des poids
3. Factorisation adaptée aux valeurs manquantes : donne $(\hat{W}_1, \dots, \hat{W}_L)$ papillon tq

$$M \odot (\hat{W}_L \dots \hat{W}_1 - W_L \dots W_1) = 0$$

4. Changement du signe de certains $L_h(\hat{W}_\ell)$ et $C_h(\hat{W}_{\ell+1})$

Algorithme de reconstruction : cas papillon multi-couches

À reconstruire : $f(x) = W_L \,\mathrm{ReLU}(W_{L-1} \dots \mathrm{ReLU}(W_1 x))$ (sans biais)

Reconstruction des poids :

1. Échantillonnage : activer le plus de chemins possible $\rightarrow$ échantillons x i.i.d. gaussiens
2. Reconstruire une version masquée $M \odot W_L \dots W_1$ du produit des poids
3. Factorisation adaptée aux valeurs manquantes : donne $(\hat{W}_1, \dots, \hat{W}_L)$ papillon tq

$$M \odot (\hat{W}_L \dots \hat{W}_1 - W_L \dots W_1) = 0$$

4. Changement du signe de certains $L_h(\hat{W}_\ell)$ et $C_h(\hat{W}_{\ell+1})$

Proposition

Si tous les chemins activables sont activés à l'étape 1, alors $(\hat{W}_1, \dots, \hat{W}_L)$ induit la même réalisation que $(W_1, \dots, W_L)$

Conclusion et perspectives

Problème de reconstruction parcimonieuse :

- 1 couche cachée : faible complexité d'échantillonnage, reconstruction garantie
- papillon multi-couches : algorithme satisfaisant pour $L \approx 10$ mais complexité d'échantillonnage encore non contrôlée

Conclusion et perspectives

Problème de reconstruction parcimonieuse :

- 1 couche cachée : faible complexité d'échantillonnage, reconstruction garantie
- papillon multi-couches : algorithme satisfaisant pour $L \approx 10$ mais complexité d'échantillonnage encore non contrôlée

Vers la distillation :

- robustesse au bruit des algos présentés ?
- expressivité des architectures MTP ?
- extension au-delà des MTP ?

Conclusion et perspectives

Problème de reconstruction parcimonieuse :

- 1 couche cachée : faible complexité d'échantillonnage, reconstruction garantie
- papillon multi-couches : algorithme satisfaisant pour $L \approx 10$ mais complexité d'échantillonnage encore non contrôlée

Vers la distillation :

- robustesse au bruit des algos présentés ?
- expressivité des architectures MTP ?
- extension au-delà des MTP ?

Merci pour votre écoute !

V. Castin, R. Gribonval, *Ouvrons la boîte noire : reconstruction de réseaux de neurones parcimonieux par échantillonnage*, GRETSI 2025