Optimisation des hyperparamètres des réseaux de neurones profonds

Dounia Lakhmiri

Directeur de recherches: Sébastien Le Digabel

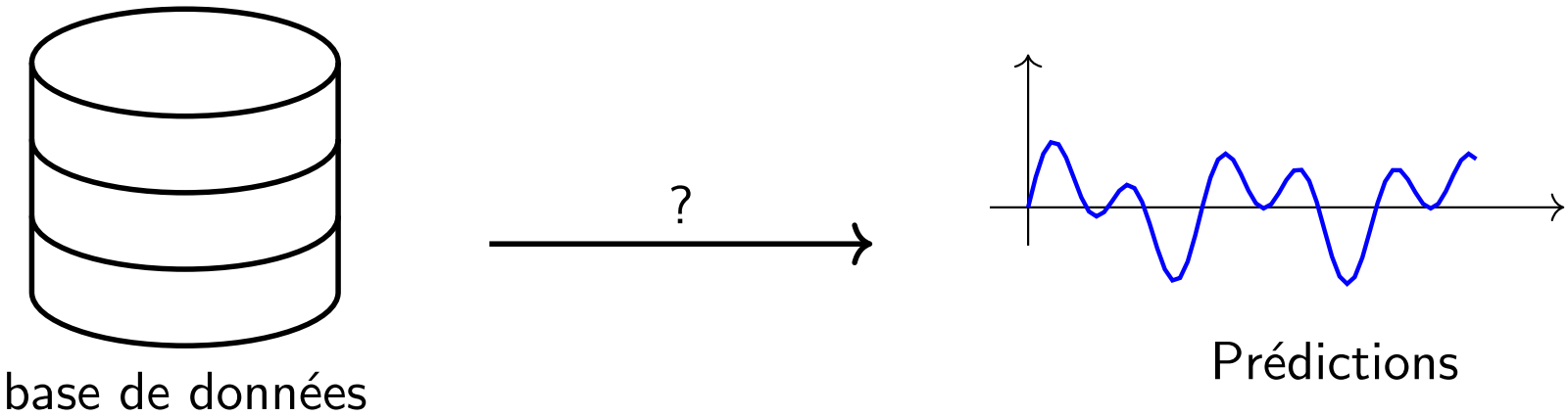
Polytechnique Montréal & GERAD

22 Mars 2021

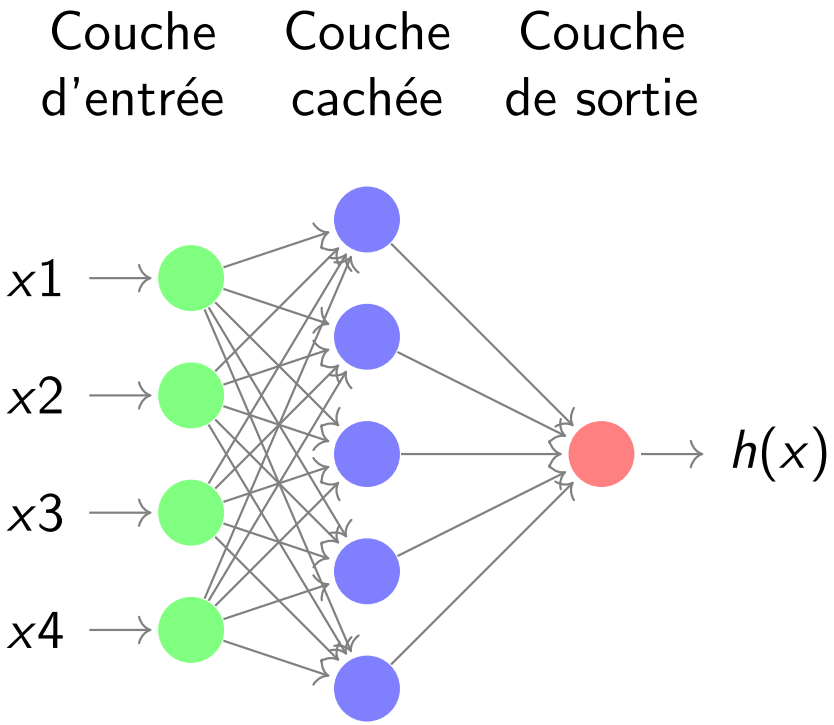
Table des matières

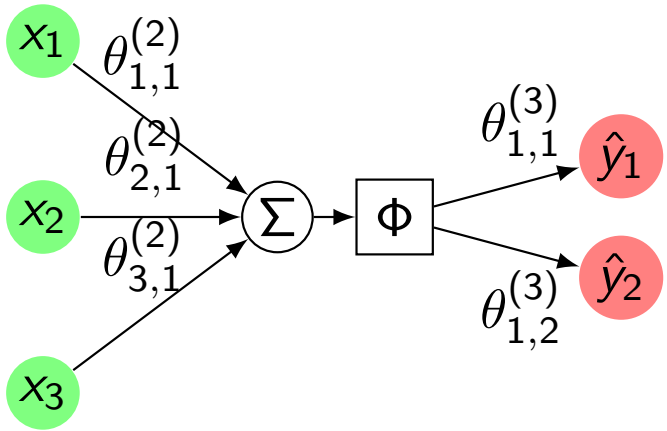
Introduction

On cherche un réseau de neurone pour apprendre au mieux d'une base de données pour produire des prédictions.



Architecture du réseau





Entraînement

- Apprentissage supervisé: $(x_i, y_i)_{1 \leq i \leq N}$

Entraînement

- Apprentissage supervisé: $(x_i, y_i)_{1 \leq i \leq N}$
- Ensemble d'entraînement: 80%.

Entraînement

- Apprentissage supervisé: $(x_i, y_i)_{1 \leq i \leq N}$
- Ensemble d'entraînement: 80%.
- Ensemble de validation: 20%.

Entraînement

- Apprentissage supervisé: $(x_i, y_i)_{1 \leq i \leq N}$
- Ensemble d'entraînement: 80%.
- Ensemble de validation: 20%.
- Ensemble de test: 20%.

Optimiser les poids du réseau

$$J(\Theta) = Err(h(x), y) + \lambda ||\Theta||^2$$

Hyperparamètres

Architecture

- Nombre de couches
- Types
- Caractéristiques
- Activation

Entraînement

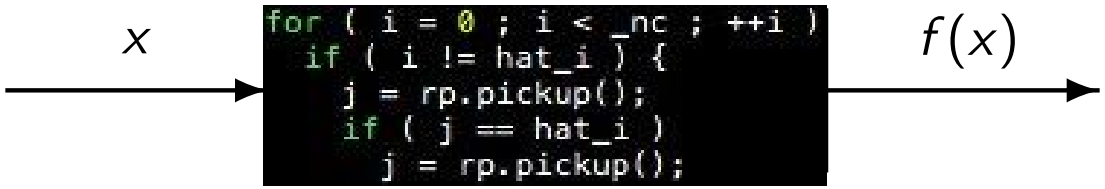
- Optimiseur
- Longueur du pas
- Régularisation
- ...

Le problème d'optimisation

Objectif: Minimiser l'erreur sur les prédictions.

$$\min_{x \in \Omega} f(x)$$

Un vecteur d'HPs x composé d'éléments continus, entiers et de catégories.



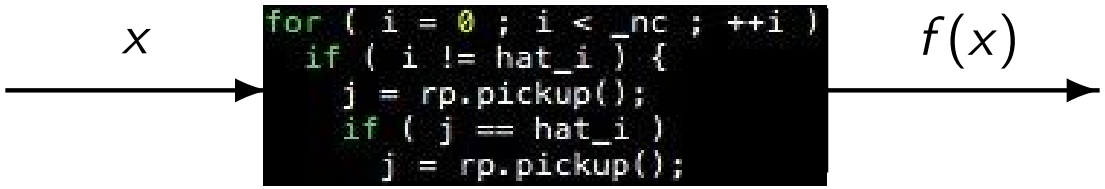
- Long à évaluer: plusieurs heures.

Le problème d'optimisation

Objectif: Minimiser l'erreur sur les prédictions.

$$\min_{x \in \Omega} f(x)$$

Un vecteur d'HPs x composé d'éléments continus, entiers et de catégories.



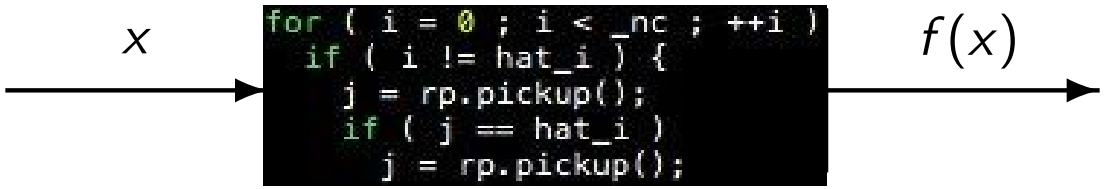
- Long à évaluer: plusieurs heures.
- Peut échouer pour certains x.

Le problème d'optimisation

Objectif: Minimiser l'erreur sur les prédictions.

$$\min_{x \in \Omega} f(x)$$

Un vecteur d'HPs x composé d'éléments continus, entiers et de catégories.



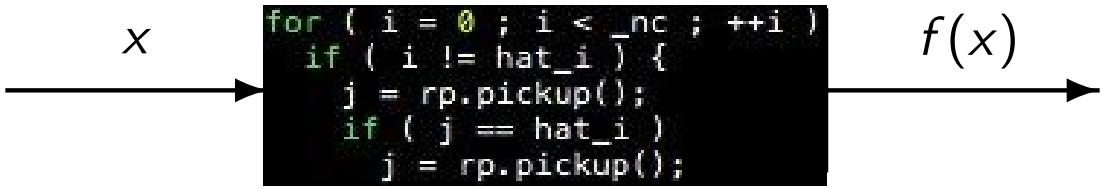
- Long à évaluer: plusieurs heures.
- Peut échouer pour certains x.
- Stochastique $f(x) \neq f(x)$.

Le problème d'optimisation

Objectif: Minimiser l'erreur sur les prédictions.

$$\min_{x \in \Omega} f(x)$$

Un vecteur d'HPs x composé d'éléments continus, entiers et de catégories.



- Long à évaluer: plusieurs heures.
- Peut échouer pour certains x.
- Stochastique $f(x) \neq f(x)$.
- Pas de dérivées disponibles.

MADS

Algorithm 1: Mesh adaptive direct search.

$k \leftarrow 0$, $\Delta_0^p \geq \Delta_0^m > 0, x_0$

[1] **Recherche (optionnelle)**

Utiliser une stratégie pour trouver un ensemble fini de points sur le maillage $S = \{s_1, s_2, \ldots, s_l\}$

Si c'est un *succès*, aller à l'étape **[4]**

[2] **Sonde**

Définir l'ensemble P_k

Évaluer les points de P_k tant qu'aucune amélioration n'est trouvée.

Aller à l'étape **[4]**

[3] **Mise à jour**

Mettre à jour $\Delta_k^p, \Delta_k^m, x_k, M_k$ selon si l'itération est un *succès* ou un *échec*

Si aucune condition d'arrêt n'est satisfaite, aller à l'étape **[1]**

MADS

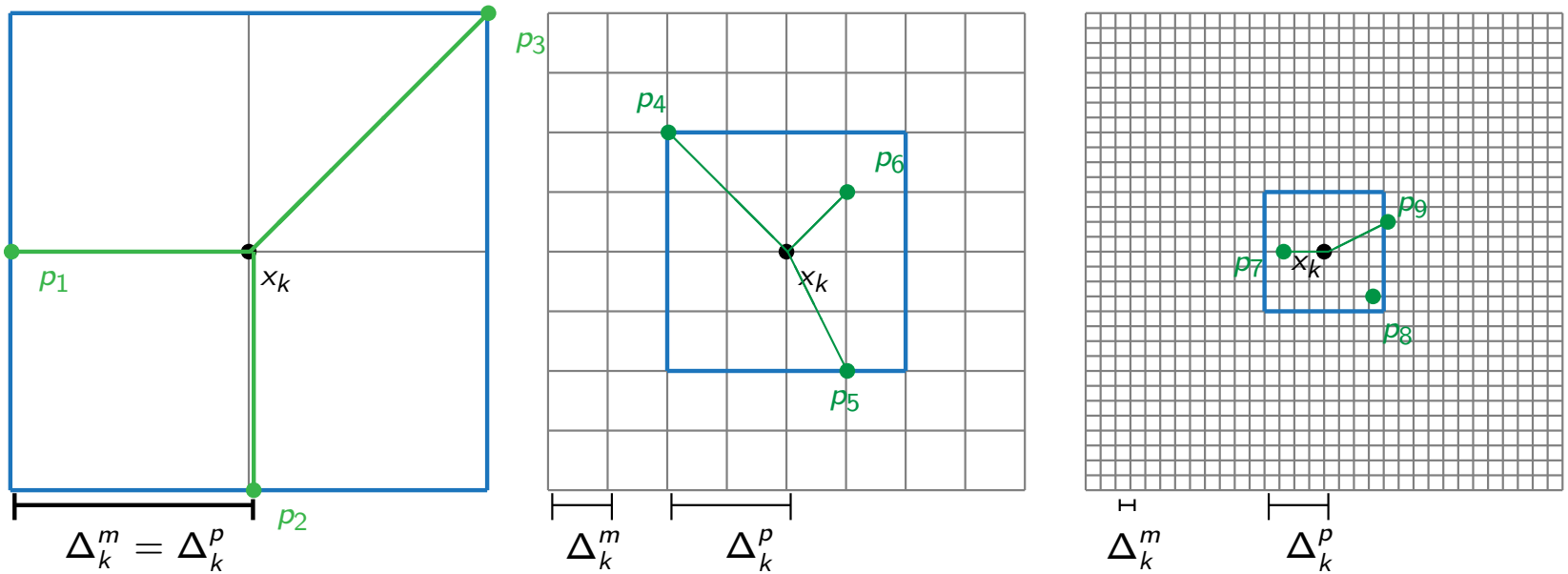
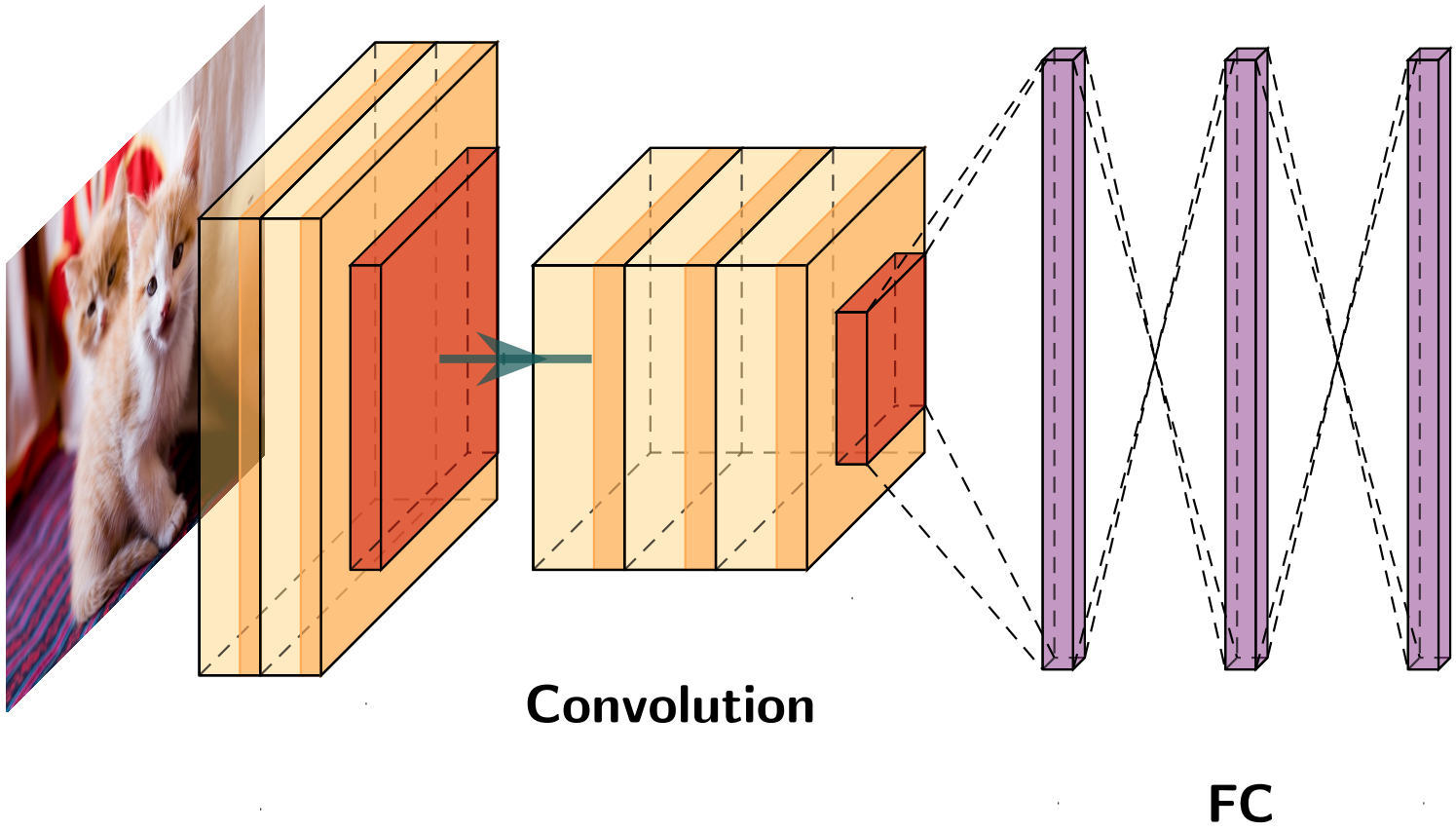
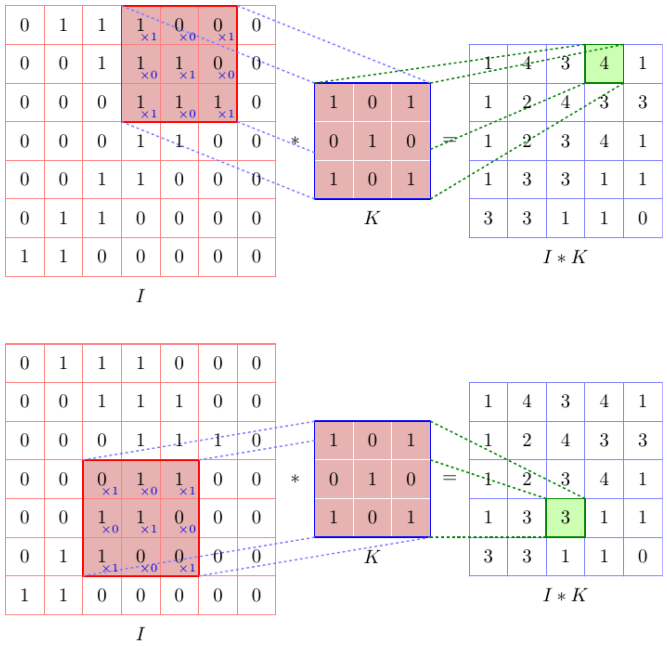


Figure: Exemple d'échecs successifs dans MADS. La taille du treillis Δ_k^m est plus agressivement rétrécie que celle de la sonde Δ_k^p. La figure est adaptée à partir de [2].

Réseaux de convolution



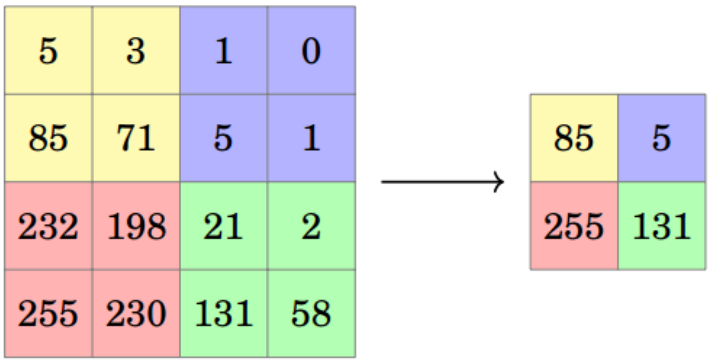
Convolution



Pour chaque couche de convolution

- Nombre de canaux de sortie.
- Taille du noyaux.
- Pas du noyau.
- Rembourrage.

Convolution



Pour chaque couche de convolution

- Nombre de canaux de sortie.
- Taille du noyaux.
- Pas du noyau.
- Rembourrage.
- Taille du pool

Architecture du CNN

#	**Hyperparameter**	**Type**	**Domaine**
1	Nb couches de convolutions (n_1)	De catégorie	$\{0,\ldots,20\}$
2	Nb de canaux de sortie	Entier	$\{0,\ldots,50\}$
3	Taille noyau	Entier	$\{0,\ldots,10\}$
4	Pas du noyau	Entier	$\{1, 2, 3\}$
5	Rembourrage	Entier	$\{0, 1, 2\}$
6	Taille pool	Entier	$\{0,\ldots,5\}$
7	Nb couches connectées (n_2)	De catégorie	$\{0, 1, \ldots, 30\}$
8	Nb de neurones	Entier	$\{0, 1, \ldots, 500\}$
9	*Dropout*	Réel	[0;1]
10	Activation	De catégorie/Entier	$ReLU(1)$, Sigmoid (2), Tanh (3)

Entraînement du CNN

Optimiseur	**Hyperparameter**	**Type**	**Range**
Stochastic Gradient Descent (SGD)	Pas d'apprentissage initial	Réel	[0;1]
	Moment	Réel	[0;1]
	Dampening	Réel	[0;1]
	Dégradation des pondérations	Réel	[0;1]
Adam	Pas d'apprentissage initial	Réel	[0;1]
	β_1	Réel	[0;1]
	β_2	Réel	[0;1]
	Dégradation des pondérations	Réel	[0;1]
Adagrad	Pas d'apprentissage initial	Réel	[0;1]
	Dégradation du pas d'apprentissage	Réel	[0;1]
	Initial accumulator	Réel	[0;1]
	Dégradation des pondérations	Réel	[0;1]
RMSProp	Pas d'apprentissage initial	Réel	[0;1]
	Moment	Réel	[0;1]
	Constante de lissage	Réel	[0;1]
	Dégradation des pondérations	Réel	[0;1]

Hyperparamètres

- n_1 couches de convolutions.

Hyperparamètres

- n_1 couches de convolutions.
- n_2 couches entièrement connectées.

Dimension

$\Rightarrow 5n_1 + n_2 + 4$ HPs pour définir l'architecture.

Hyperparamètres

- n_1 couches de convolutions.
- n_2 couches entièrement connectées.
- Choix de l'optimiseur + 4 HPs.

Hyperparamètres

- n_1 couches de convolutions.
- n_2 couches entièrement connectées.
- Choix de l'optimiseur + 4 HPs.
- Taille du batch

Dimension

$\Rightarrow 5n_1 + n_2 + 10$ HPs au total.

HyperNOMAD

Algorithm 2: MADS avec voisinage pour variables de catégories.

$k \leftarrow 0$, $\Delta_0^p \geq \Delta_0^m > 0, x_0$

[1] **Recherche (optionnelle)**

Utiliser une stratégie pour trouver un ensemble fini de points sur le maillage $S = \{s_1, s_2, \ldots, s_l\}$

Si c'est un *succès*, aller à l'étape **[4]**

[2] **Sonde**

Définir l'ensemble P_k

Évaluer les points de P_k tant qu'aucune amélioration n'est trouvée.

[2′] **Sonde élargie**

Produire et évaluer les voisins de x_k.

Si aucune amélioration, faire une descente pour chaque voisin suffisamment proche d'un succès.

Aller à l'étape **[4]**

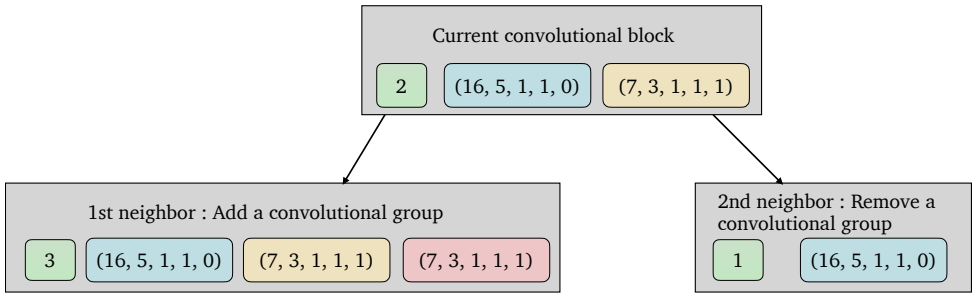
[3] **Mise à jour**

Mettre à jour $\Delta_k^p, \Delta_k^m, x_k, M_k$ selon si l'itération est un *succès* ou un *échec*

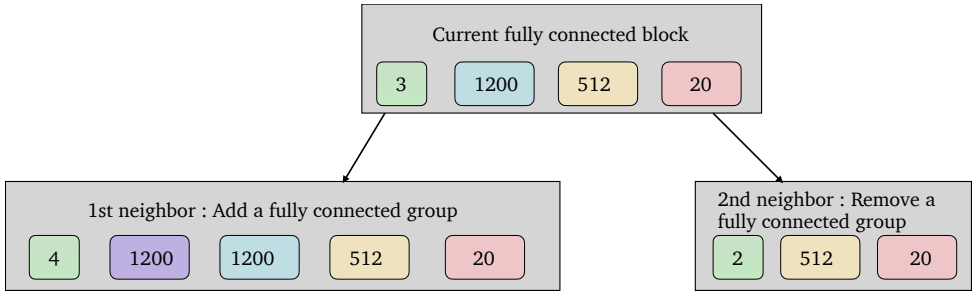
Si aucune condition d'arrêt n'est satisfaite, aller à l'étape **[1]**

Voisinages

Voisinage du bloc des couches de convolution:

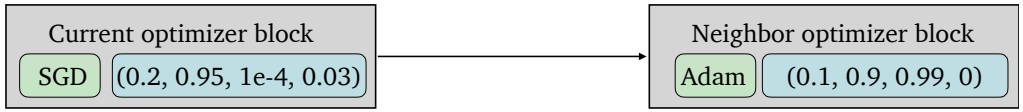


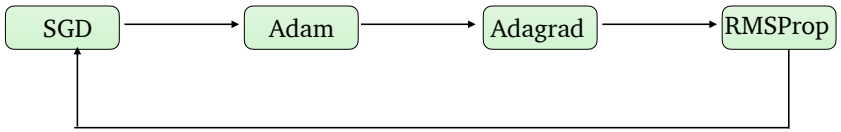
Voisinage du bloc des couches entièrement connectées:



Voisinages

Voisinage du bloc de l'optimiseur:



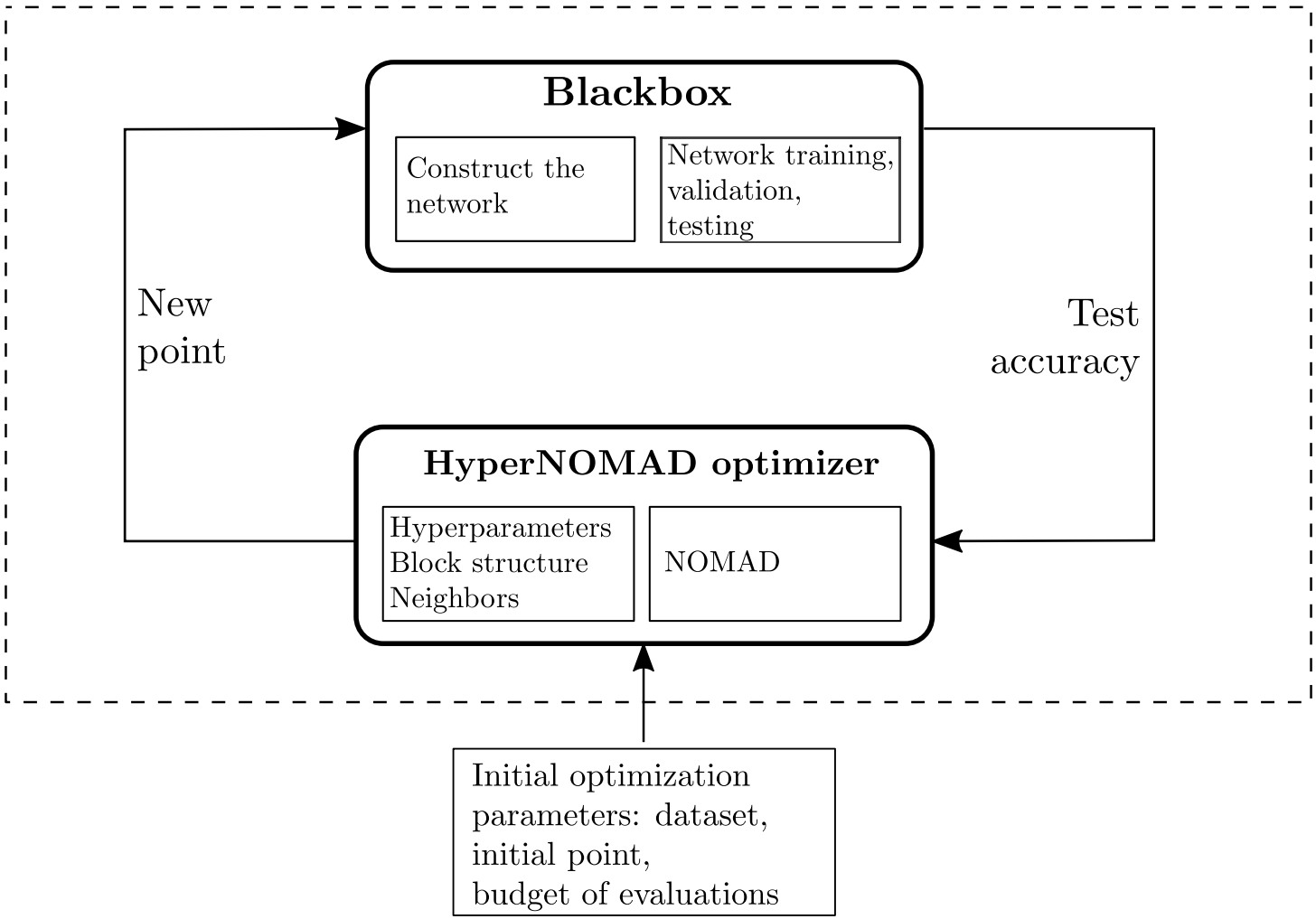


Structure de voisinage

Chaque point x_k possède 5 voisins.

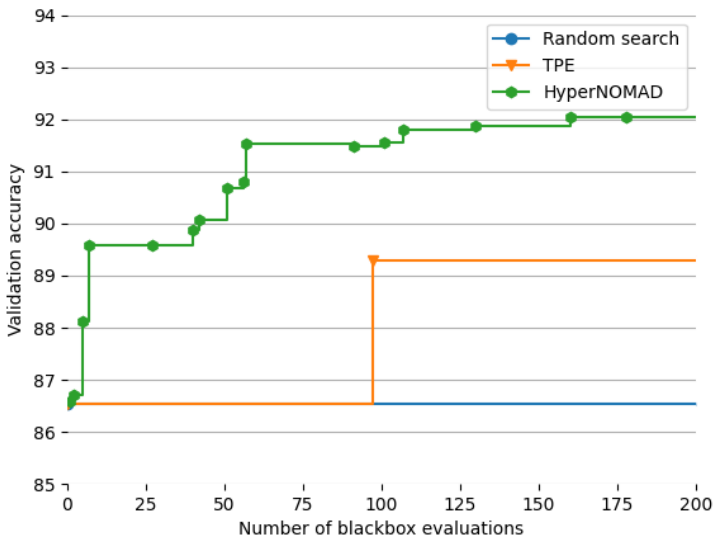
- $\pm$ 1 couche de convolution.
- $\pm$ 1 couche entièrement connectées.
- Changement de l'optimiseur.

HyperNOMAD

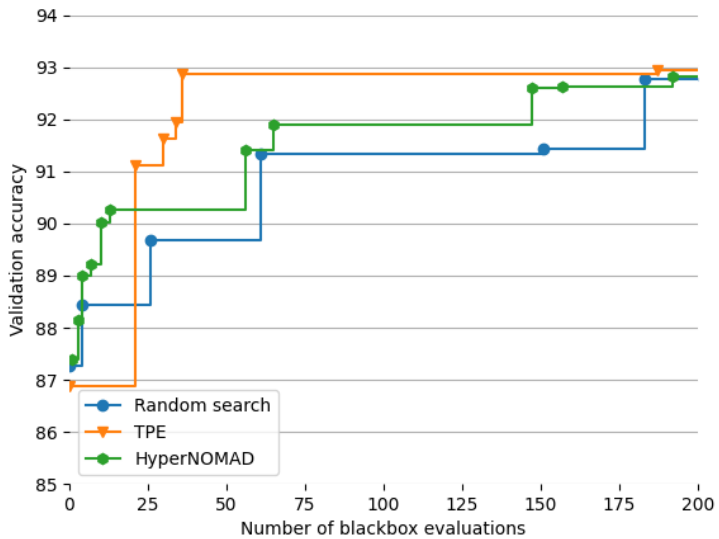

Blackbox
Construct the network
Network training, validation, testing
New point
Test accuracy
HyperNOMAD optimizer
Hyperparameters
Block structure
Neighbors
NOMAD
Initial optimization parameters: dataset, initial point, budget of evaluations

Fashion-MNIST

- Configuration par défaut: 17 hyperparamètres.
- 200 évaluations de la boîte noire.



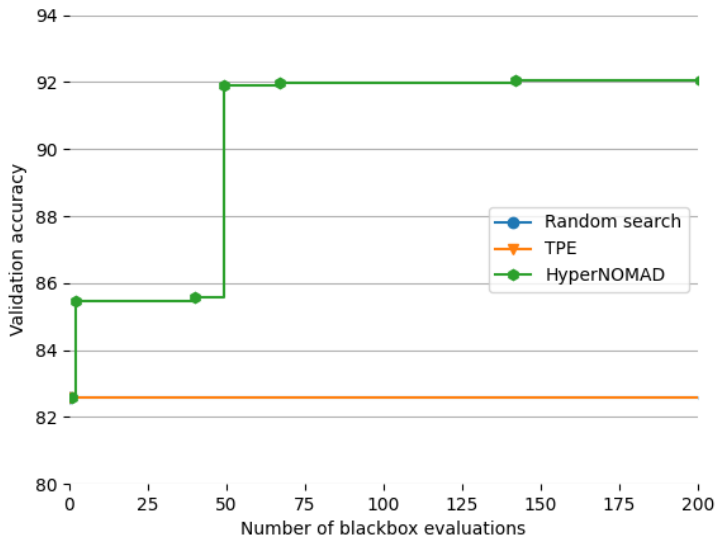
(a) Graphe de convergence avec dimension variable.



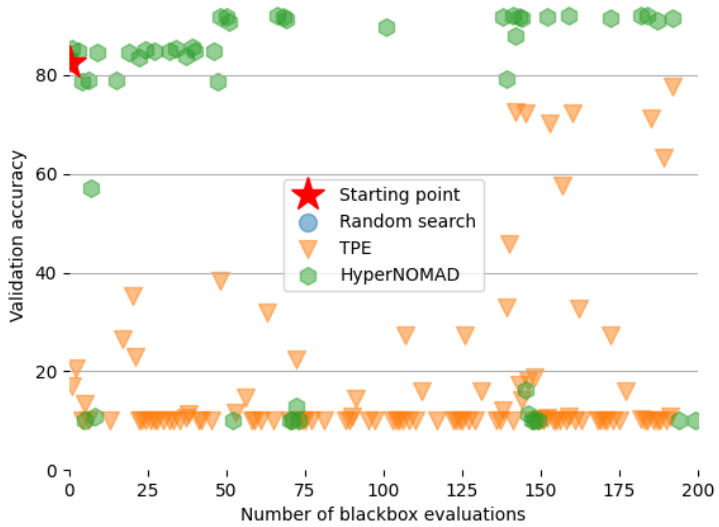
(b) Graphe de convergence avec dimension fixe.

CIFAR-10

- Configuration VGG-13: 75 hyperparamètres.
- 200 évaluations de la boîte noire.



(a) Graphe de convergence de chaque méthode.



(b) Scores des configurations testées

HyperNOMAD

Points forts

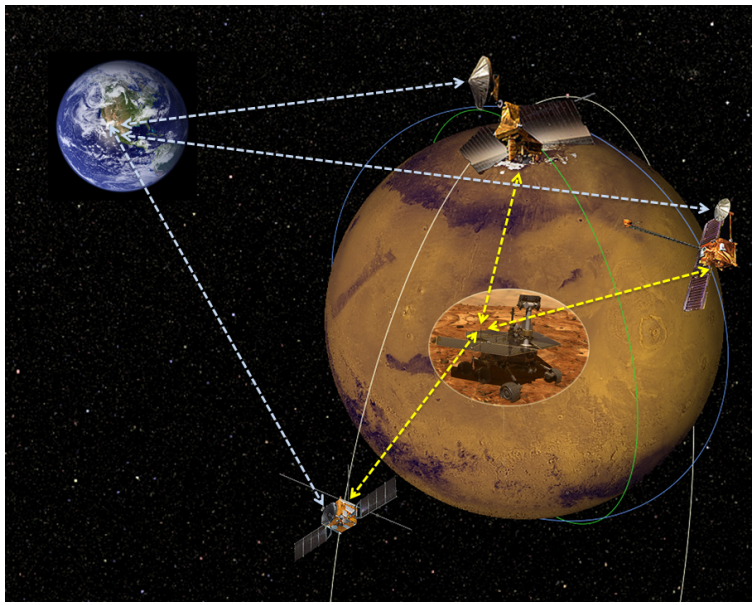
- Résultats compétitifs
- Dimension variable
- Architecture et entraînement
- Exploration adaptée à l'espace de recherche.

Points faibles

- Architectures limitées
- Dimension élevée
- Temps d'exécution important
- Consommation des ressources
- Limité aux réseaux de convolution.

Contexte

Une perte ou une corruption de données peut se produire lors de leur transfert du robot Curiosity au Mars Science Lab lorsqu'elles transitent par le pipeline suivant:



Base de données

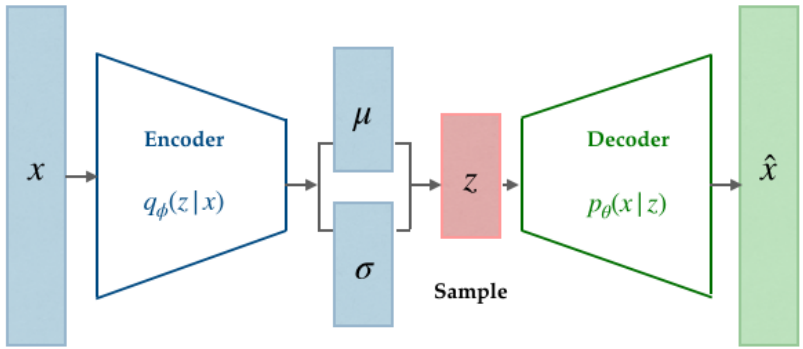
Aspects pertinents sur la base de données

- On a 6 satellites: MRO, ODY, MEX, MVN, TGO et la possibilité d'une transmission directe: DTE.
- Chaque vecteur d'entrée comprend 43 éléments: réels, entiers and booléens.
- Empiriquement on constate un taux de 13% de passes incomplètes.

Remarque:

Ce problème est équivalent à une classification non supervisée sur des données disproportionnées.

auto-encodeur variationnel (VAE)



Function de coût

$$L = ||\hat{x} - x|| + D_{KL}(q_\phi(z|x)||p(z)). \quad (1)$$

où $p(z) \sim N(0, I)$ et D_{KL} est la divergence de Kullback-Liebler qui mesure la dissimilarité entre deux distributions de probabilités.

Hyperparamètres

Hyperparameter	**Type**	**Range**
Nombre de couche d'encodage	Entier	$[1, 50]$
Dimension de la couche centrale	Entier	$[1, n_0[$
Taille du batch	Entier	$[10, 512]$
Fonction d'activation	De catégorie	1 : ReLU, 2 : Sigmoid, 3 : Tanh.
Dropout rate	Réel	$[0, 1]$
Choix de l'optimiseur	De catégorie	1 : SGD, 2 : Adam. 3 : Adagrad. 4 : RMSProp.
4 HPs de l'optimiseur	Réel	$[0, 1]$
Seuil α	Réel	$[0.50, 1]$

Δ-DOGS

Méthode d'optimisation sans dérivées avec modèles basée sur la triangularisation de Delaunay.

Soit

- f la fonction objectif,
- p une fonction d'interpolation,
- e la fonction d'incertitude calculée sur la base de la triangulation de Delaunay,
- f_0 la valeur cible,
- et la fonction de recherche $s = \begin{cases} \frac{p(x)-f_0}{e(x)} & \text{si } p(x) \geq f_0 \\ p(x) - f_0 & \text{sinon.} \end{cases}$

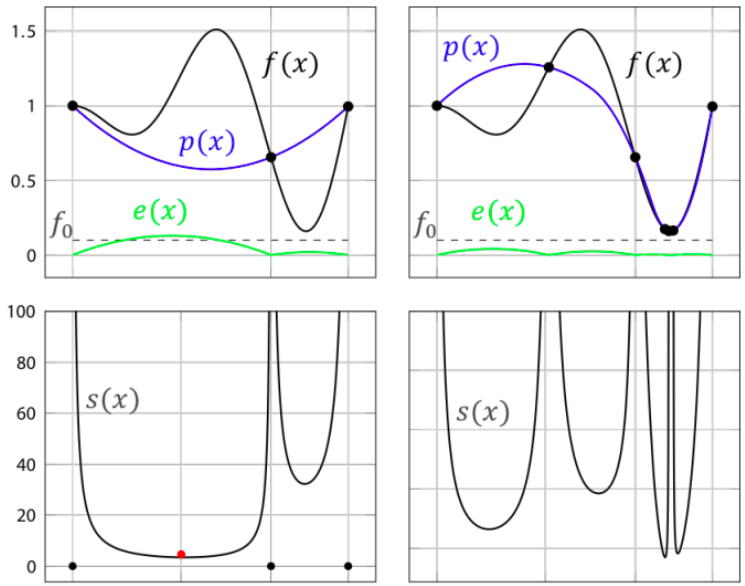


Figure: Mécanisme de Δ-DOGS. Image from [1]

Δ-MADS

Algorithm 3: MADS + Δ-DOGS.

$k \leftarrow 0$, $\Delta_0^p \geq \Delta_0^m > 0, x_0$ $y_0, \epsilon \in]0,1[$

[1] **Recherche**

Fixer les valeurs entières et catégoriques du point courant x_k^N et utiliser Δ-DOGS sur le sous problème aux variables réelles x_k^R avec pour valeur cible y_k

retourner le nouveau point $\hat{x_k} = x_k^N \cup x_k^R$

Si c'est un *succès*, aller à l'étape **[4]**

[2] **Sonde**

Définir l'ensemble P_k autour de $\hat{x_k}$

Évaluer les points de P_k tant qu'aucune amélioration n'est trouvée.

Aller à l'étape **[4]**

[3] **Mise à jour**

Soit f_k la meilleure valeur de l'objectif

Si $f_k < y_k$ **Alors** $y_{k+1} = y_k - \epsilon$

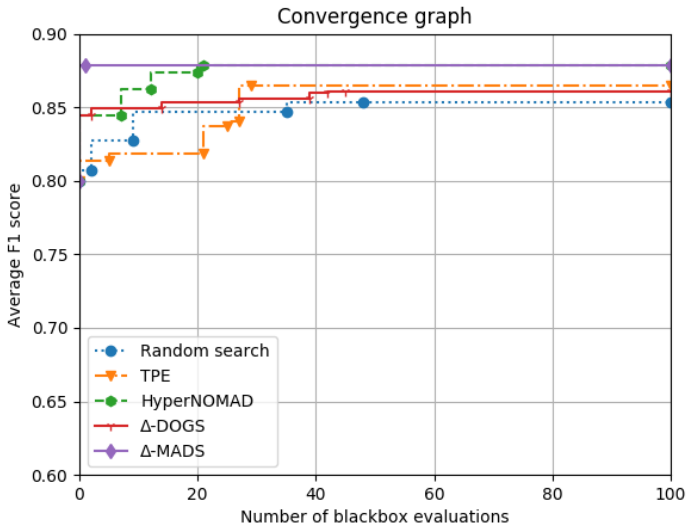
Sinon $y_{k+1} = y_k + \epsilon$

Mettre à jour $\Delta_k^p, \Delta_k^m, x_k, M_k$ selon si l'itération est un *succès* ou un *échec*

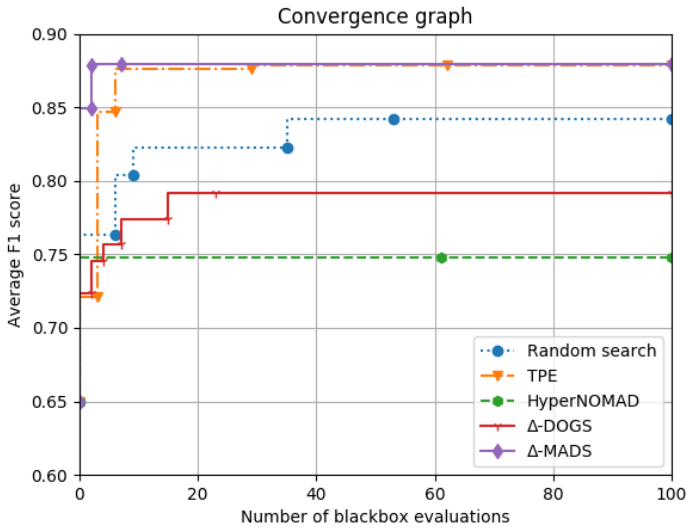
Si aucune condition d'arrêt n'est satisfaite, aller à l'étape **[1]**

Résultats

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F1 = 2\frac{R \times P}{R + P} \tag{2}$$



(a) Convergence graphs for each HPO algorithm on the advantageous initialization.



(b) Convergence graphs for each HPO algorithm on the disadvantageous initialization.

Δ-MADS

Points forts

- Détection d'anomalies non supervisée,
- Δ-MADS obtient de meilleurs résultats plus rapidement.

Points faibles

- Les scores sur les passes complètes et incomplètes dépendent fortement de la valeur du seuil α. Il est donc difficile de dépasser la barre des 90% sur les deux simultanément.

Retour sur HyperNOMAD

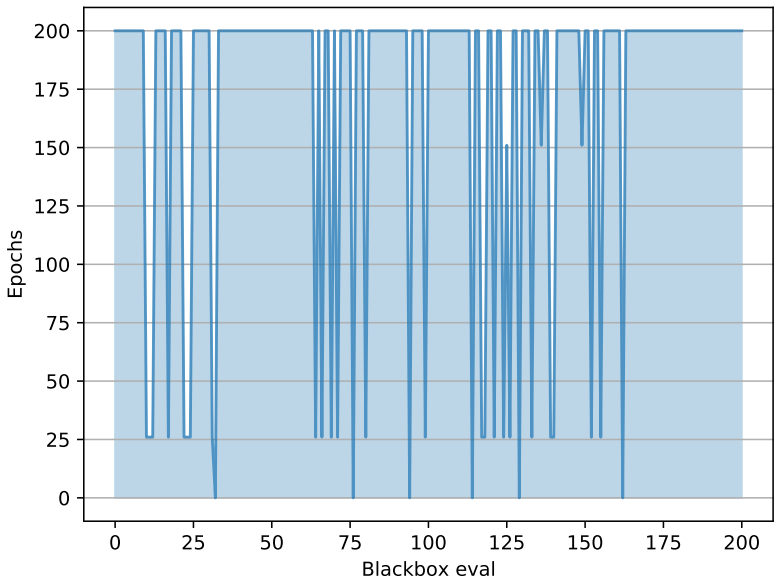
Temps d'exécution

Pour 200 évaluations de boîte noire:

- MNIST: 4 jours.
- CIFAR-10: 10 – 14 jours.

Ressources

Presque toutes les configurations sont entraînées avec le budget complet.



MADS

Algorithm 4: MADS avec substituts statiques.

$k \leftarrow 0$, $\Delta_0^p \geq \Delta_0^m > 0, x_0$

Substitut de classement S_1 et substitut d'interruption S_2

[1] **Recherche (optionnelle)**

Utiliser une stratégie pour trouver un ensemble fini de points sur le maillage $S = \{s_1, s_2, \ldots, s_l\}$

Si c'est un *succès*, aller à l'étape **[4]**

[2] **Sonde**

Définir l'ensemble P_k

Ordonner les points à évaluer suivant S_1

Évaluer suivant S_2 les points de P_k tant qu'aucune amélioration n'est trouvée.

Aller à l'étape **[4]**

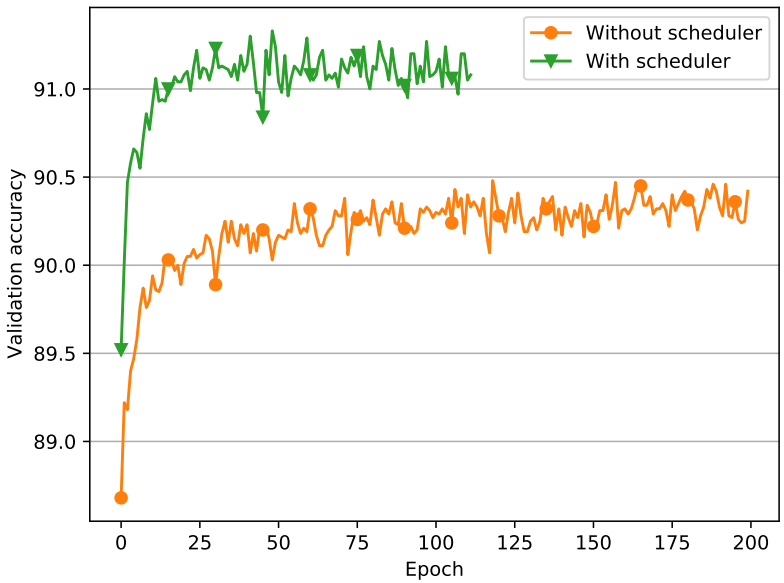
[3] **Mise à jour**

Mettre à jour $\Delta_k^p, \Delta_k^m, x_k, M_k$ selon si l'itération est un *succès* ou un *échec*

Si aucune condition d'arrêt n'est satisfaite, aller à l'étape **[1]**

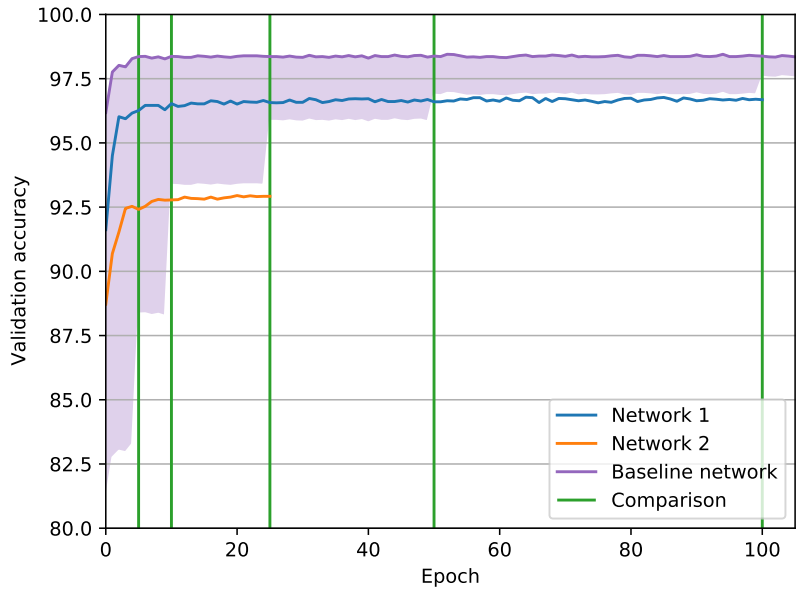
Interruptions

- Plus d'amélioration sur les scores de validations.
- Amélioration trop lente et minime.



Interruptions

- Le score de validation s'améliore.
- Score assez loin du meilleur candidat.



Interruptions

- MNIST
- Maximum de 200 évaluations de boîte noire.

Early stopping strategy	Top-1 val. acc.	Wall-clock time (s)	Total epochs
HyperNOMAD	99.38%	305856	35503
Dernier succès	99.40%	129600	17534
Scheduler	99.29%	170208	21987
Scheduler and baseline	99.41%	126144	9681

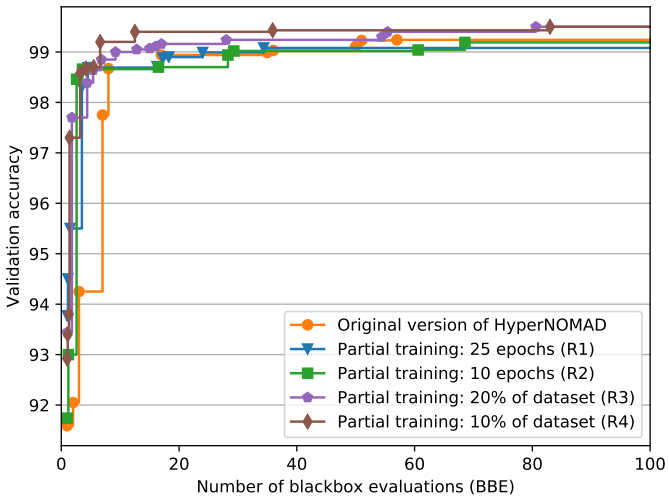
Classement

- Estimation fiable de la précision après entraînement.
- Estimation rapide.

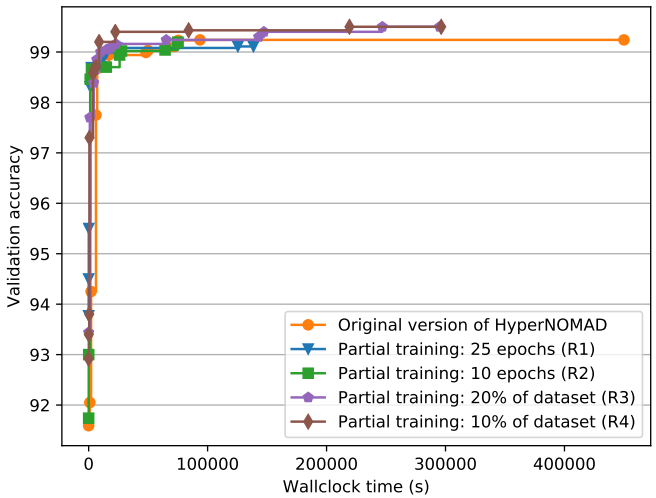
Function	Training budget (epochs)	Portion of dataset	Cost ratio to full BBE
Objective function	200	100%	100%
Substitut R_1	25	100%	12.5%
Substitut R_2	10	100%	5%
Substitut R_3	200	20%	20%
Substitut R_4	200	10%	10%

Classement

- MNIST
- Maximum de 200 évaluations de boîte noire.



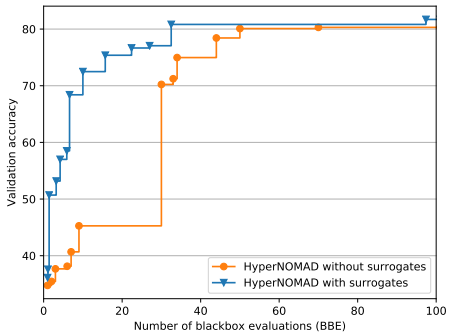
(a) Convergence of each variant in terms of validation accuracy per number of BBE.



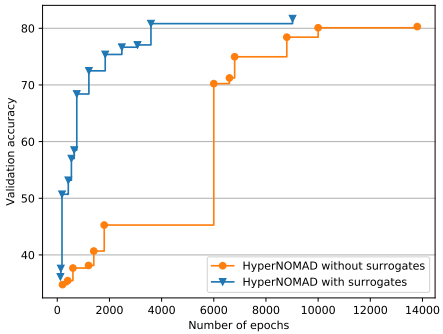
(b) Convergence of each variant in terms of validation accuracy per overall execution time.

Interruptions et classement

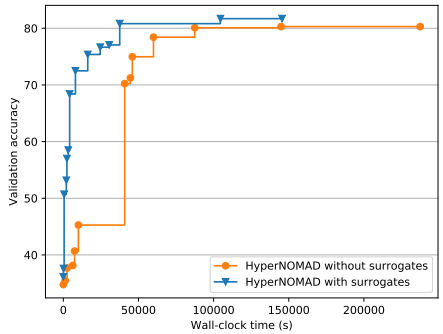
- CIFAR-10
- 17 HPs.



(a) Convergence en terme de précision de validation par évaluations de la boîte noire.



(b) Convergence en terme de précision de validation par nombre d'epochs.



(c) Convergence en terme de précision de validation par temps d'exécution.

HyperNOMAD + Substituts statiques

Points forts

- 30% de moins de budget d'entraînement.
- 25% plus rapide.
- pas de perte considérable sur la précision.

Points faibles

- Pas encore intégré à l'implémentation officielle de HyperNOMAD.
- Espace de recherches limité.

Syntèse

- Adaptation de MADS pour l'optimisation des hyperparamètres des réseaux de neurones profonds.

Syntèse

- Adaptation de MADS pour l'optimisation des hyperparamètres des réseaux de neurones profonds.
- Résultats compétitifs sur des problèmes académiques ou réel.

Syntèse

- Adaptation de MADS pour l'optimisation des hyperparamètres des réseaux de neurones profonds.
- Résultats compétitifs sur des problèmes académiques ou réel.
- Exploiter la recherche et les substituts statiques pour accélérer l'optimisation et économiser les ressources.

Améliorations futures

- Architectures par blocs.

Améliorations futures

- Architectures par blocs.
- Compression d'architectures.

Améliorations futures

- Architectures par blocs.
- Compression d'architectures.
- Ajouter des hyperparamètres pour l'augmentation et les transformations sur les données.

Améliorations futures

- Architectures par blocs.
- Compression d'architectures.
- Ajouter des hyperparamètres pour l'augmentation et les transformations sur les données.
- Étendre HyperNOMAD pour d'autres tâches, autres types de réseaux.

Améliorations futures

- Architectures par blocs.
- Compression d'architectures.
- Ajouter des hyperparamètres pour l'augmentation et les transformations sur les données.
- Étendre HyperNOMAD pour d'autres tâches, autres types de réseaux.
- Avoir une interface plus conviviale, avec des outils visuels pour suivre l'exécution de HyperNOMAD.

Questions?

Références I

[1] S. R. Alimo, P. Beyhaghi, and T. Bewley. Optimization combining derivative-free global exploration with derivative-based local refinement. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2531–2538. IEEE, 2017.

[2] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm. *ACM Transactions on Mathematical Software*, 37(4):44:1–44:15, 2011. doi: 10.1145/1916461.1916468. URL http://dx.doi.org/10.1145/1916461.1916468.