Introduction au Meta-Learning

Séminaire TALia

Jean Vassoyan

ONEPOINT & TELECOM PARIS

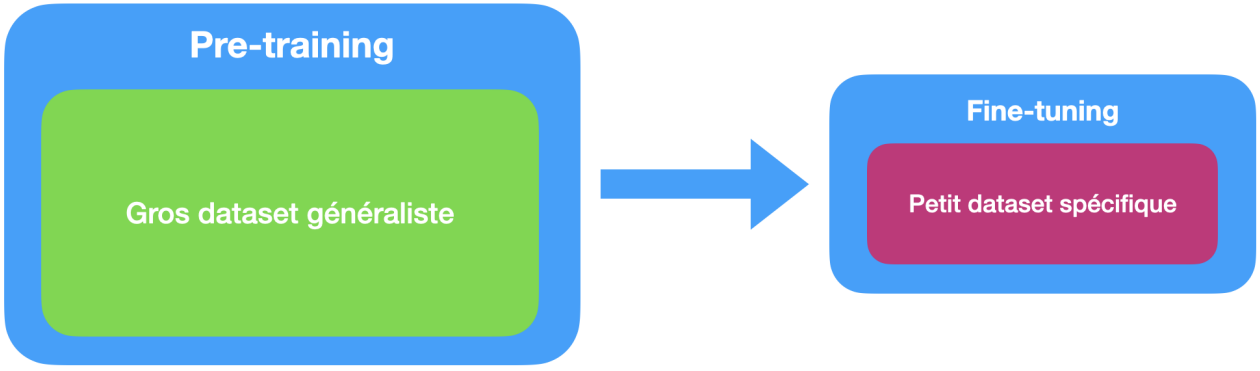
September 16, 2022

Table of Contents

Table of Contents

Pourquoi le Meta-Learning ?

- On veut construire un modèle **généraliste**
- Approche classique : Transfer Learning



- Défauts :
  - Nécessite dataset de grande taille pour chaque *downstream task*
  - Faible capacité de généralisation "out-of-distribution"
  - Les humains apprennent avec peu d'exemples

Objectif du Meta-Learning

Objectif : créer un modèle capable de s'adapter **rapidement** à de nouvelles taches (pas vues à l'entrainement)

rapidement = peu d'exemples par classe
(few-shot learning, one-shot learning, zero-shot learning)

Objectif du Meta-Learning

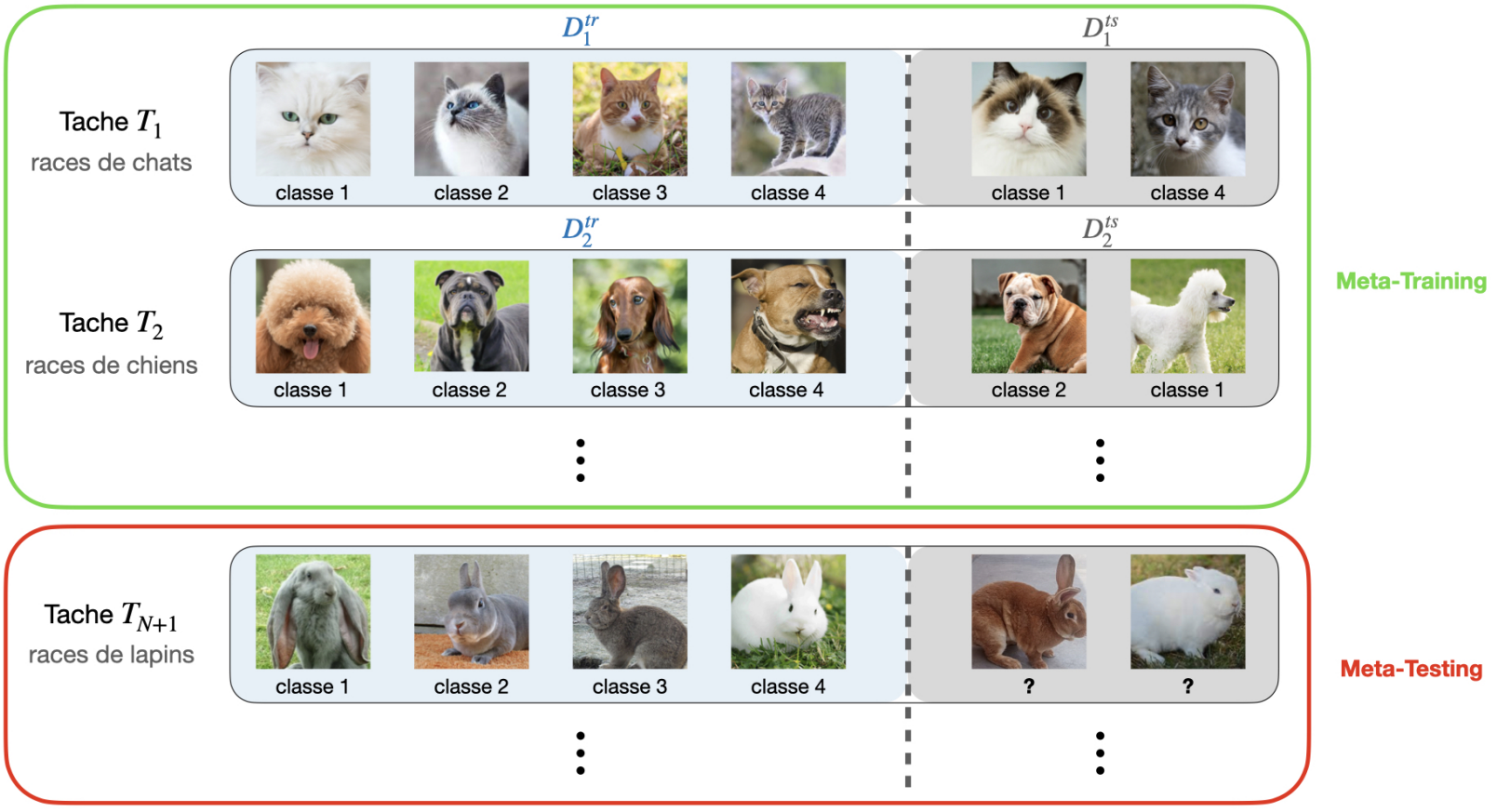
Objectif : créer un modèle capable de s'adapter **rapidement** à de nouvelles taches (pas vues à l'entrainement)

rapidement = peu d'exemples par classe
(few-shot learning, one-shot learning, zero-shot learning)

$\rightarrow$ Comment prépare-t-on un modèle à effectuer des taches auxquelles il n'a jamais été confronté ?

Le problème de Meta-Learning

Le problème de Meta-Learning



Le problème de Meta-Learning

Objectif :
Etant donné un ensemble de taches $\mathcal{T}_1, \ldots, \mathcal{T}_n$, résoudre rapidement la nouvelle tache $\mathcal{T}_{\text{test}}$.

Les taches $\mathcal{T}_i = \{\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{test}}, \mathscr{L}_i\}$ sont i.i.d. et tirées suivant une même distribution $p(\mathcal{T})$

$\rightarrow$ Ne fonctionne que si toutes ces tâches partagent une structure géométrique commune !

Le problème de Meta-Learning

Supervised Learning:

Inputs: $\mathbf{x}$ Outputs: $\mathbf{y}$ Data: $\{(\mathbf{x}, \mathbf{y})_i\}$

$$\mathbf{y} = g_\phi(\mathbf{x})$$

Meta Supervised Learning:

Inputs: $\mathcal{D}^{\text{tr}}$ $\mathbf{x}^{\text{ts}}$ Outputs: $\mathbf{y}^{\text{ts}}$ Data: $\{\mathcal{D}_i\}$

$$\{(\mathbf{x}, \mathbf{y})_{1:K}\} \quad \mathbf{y}^{\text{ts}} = f_\theta(\mathcal{D}^{\text{tr}}, \mathbf{x}^{\text{ts}})$$

$$\mathcal{D}_i : \{(\mathbf{x}, \mathbf{y})_j\}$$

Finn. *Learning to Learn with Gradients*. PhD Thesis. 2018

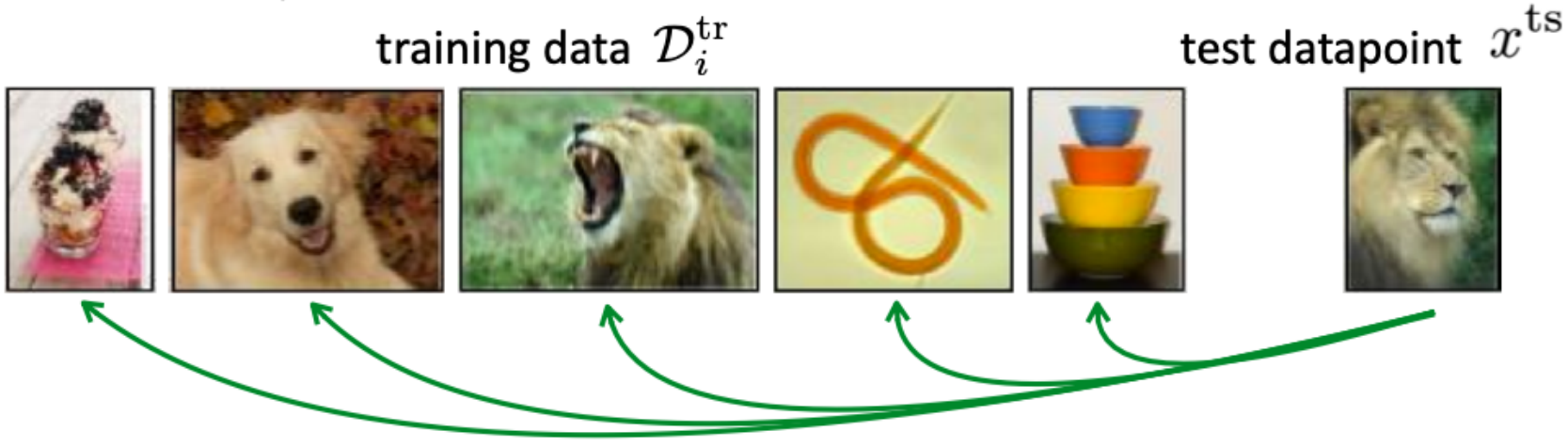
Fonction objectif :

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \sum_{(x,y) \sim \mathcal{D}_i^{\text{ts}}} \mathscr{L}_i \left(f_\theta \left(\mathcal{D}_i^{\text{tr}}, x \right), y \right)$$

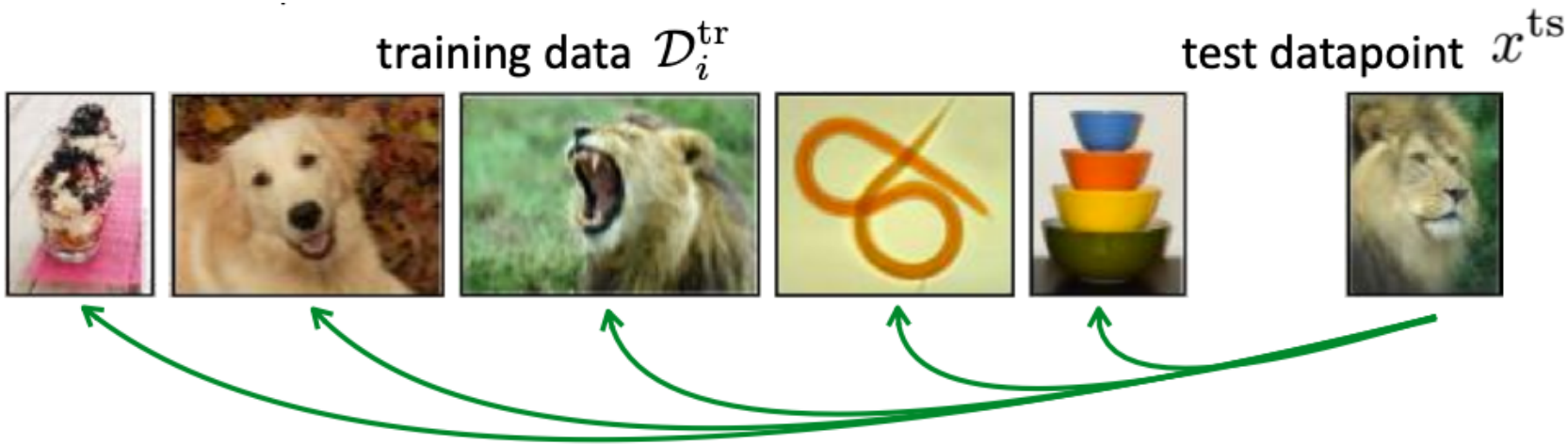
Table of Contents

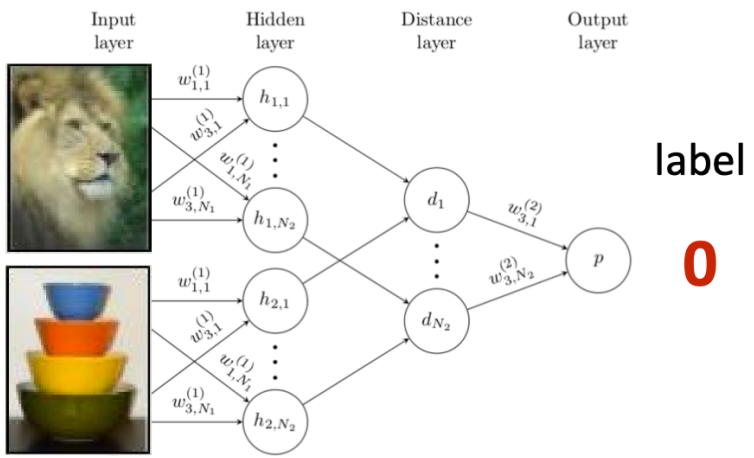
Non-Parametric Meta-Learning

Idée : Apprendre à comparer les éléments de $\mathcal{D}_i^{\text{ts}}$ à ceux de $\mathcal{D}_i^{\text{tr}}$

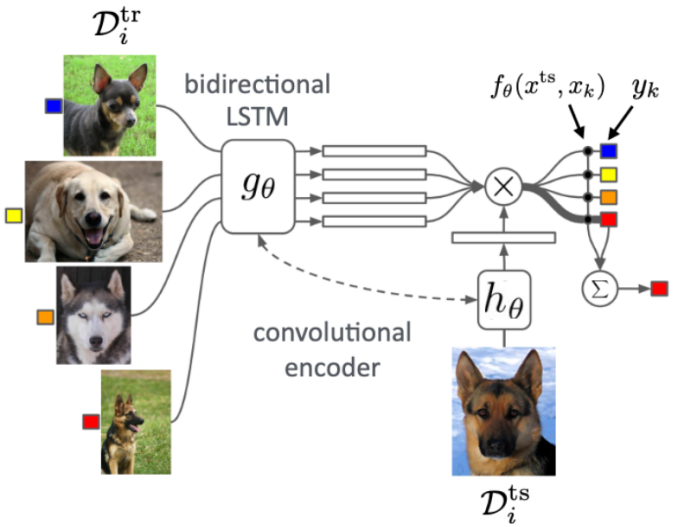


Non-Parametric Meta-Learning



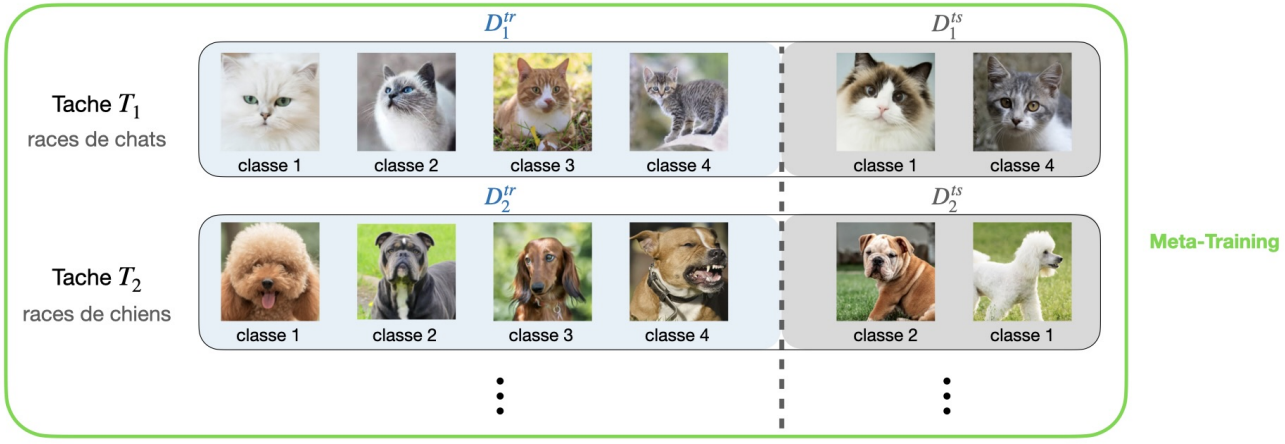


(a) Koch et al., ICML 2015



(b) Vinyals et al., NeurIPS 2016

Non-Parametric Meta-Learning : *Matching Networks*



1. Sample task $\mathcal{T}_i$
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}$, $\mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$
3. Compute $\hat{y}^{\text{ts}} = \sum_{x_k, y_k \in \mathcal{D}_i^{\text{tr}}} f_\theta \left(x^{\text{ts}}, x_k\right) y_k$
4. Update θ using $\nabla_\theta \mathcal{L} \left(\hat{y}^{\text{ts}}, y^{\text{ts}}\right)$

Non-Parametric Meta-Learning : Avantages / Inconvénients

Points forts :

- Généralement très expressive
- Rapide et facile à optimiser

Points faibles :

- Difficile à généraliser sur des tailles de $\mathcal{D}_i^{\text{tr}}$ variables
- Peu scalable sur les $\mathcal{D}_i^{\text{tr}}$ de grandes tailles
- Approche limitée à la classification

Table of Contents

Black-Box Meta-Learning (BBML)

Objectif

Entrainer un réseau de neurones f_θ à prédire les paramètres ϕ_i d'un autre réseau de neurones g_{ϕ_i} selon la tâche $\mathcal{T}_i$.

Black-Box Meta-Learning (BBML)

Objectif

Entrainer un réseau de neurones f_θ à prédire les paramètres ϕ_i d'un autre réseau de neurones g_{ϕ_i} selon la tâche $\mathcal{T}_i$.

$$f_\theta : \mathcal{D}_i^{\text{tr}} \mapsto \phi_i$$

$$g_{\phi_i} : \mathbf{x}^{\text{ts}} \mapsto \mathbf{y}^{\text{ts}}$$

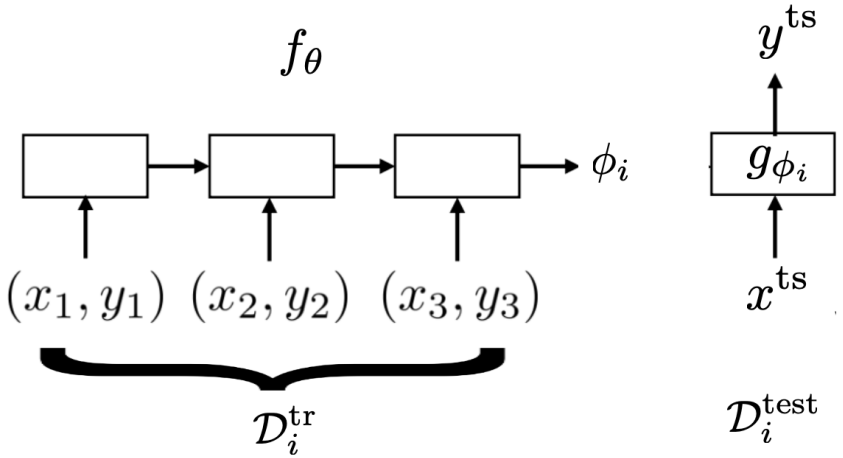
Black-Box Meta-Learning (BBML)

Objectif

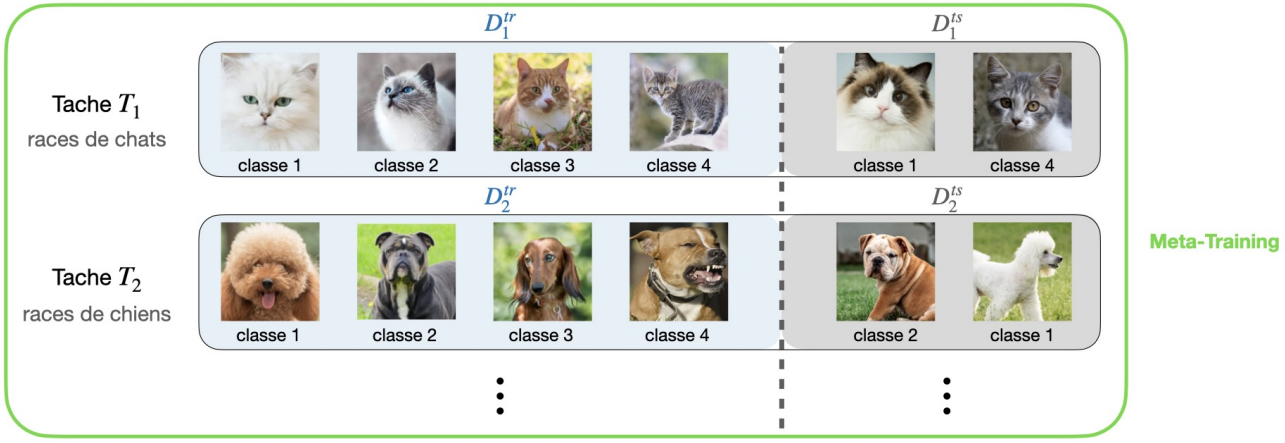
Entrainer un réseau de neurones f_θ à prédire les paramètres ϕ_i d'un autre réseau de neurones g_{ϕ_i} selon la tâche $\mathcal{T}_i$.

$$f_\theta : \mathcal{D}_i^{\text{tr}} \mapsto \phi_i$$

$$g_{\phi_i} : \mathbf{x}^{\text{ts}} \mapsto \mathbf{y}^{\text{ts}}$$



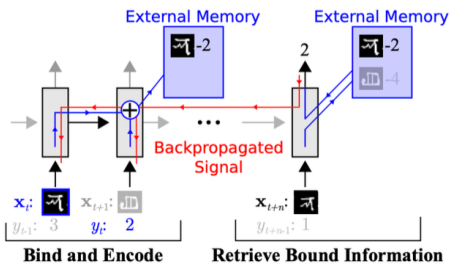
BBML : Algorithme Général



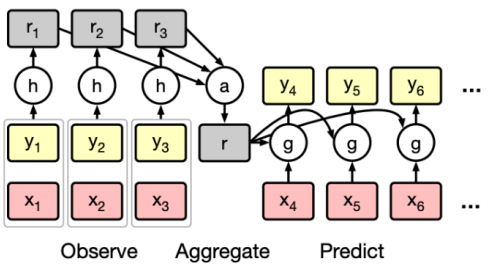
1. Sample task $\mathcal{T}_i$ (or mini batch of tasks)
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}$, $\mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$
3. Compute $\phi_i \leftarrow f_\theta\left(\mathcal{D}_i^{\text{tr}}\right)$
4. Update θ using $\nabla_\theta \mathcal{L}\left(g_{\phi_i}, \mathcal{D}_i^{\text{test}}\right)$

En pratique

- Prédire tous les paramètres d'un réseau de neurones ne serait pas très scalable !
- En pratique : prédire seulement une représentation suffisante de $\mathcal{T}_i$



Meta-Learning with Memory-Augmented Neural Networks Santoro, Bartunov, Botvinick, Wierstra, Lillicrap. ICML '16



Conditional Neural Processes. Garnelo, Rosenbaum, Maddison, Ramalho, Saxton, Shanahan, Teh, Rezende, Eslami. ICML '18

Exemple : GPT-3

Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***

Jared Kaplan† **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**

Amanda Askell **Sandhini Agarwal** **Ariel Herbert-Voss** **Gretchen Krueger** **Tom Henighan**

Rewon Child **Aditya Ramesh** **Daniel M. Ziegler** **Jeffrey Wu** **Clemens Winter**

Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**

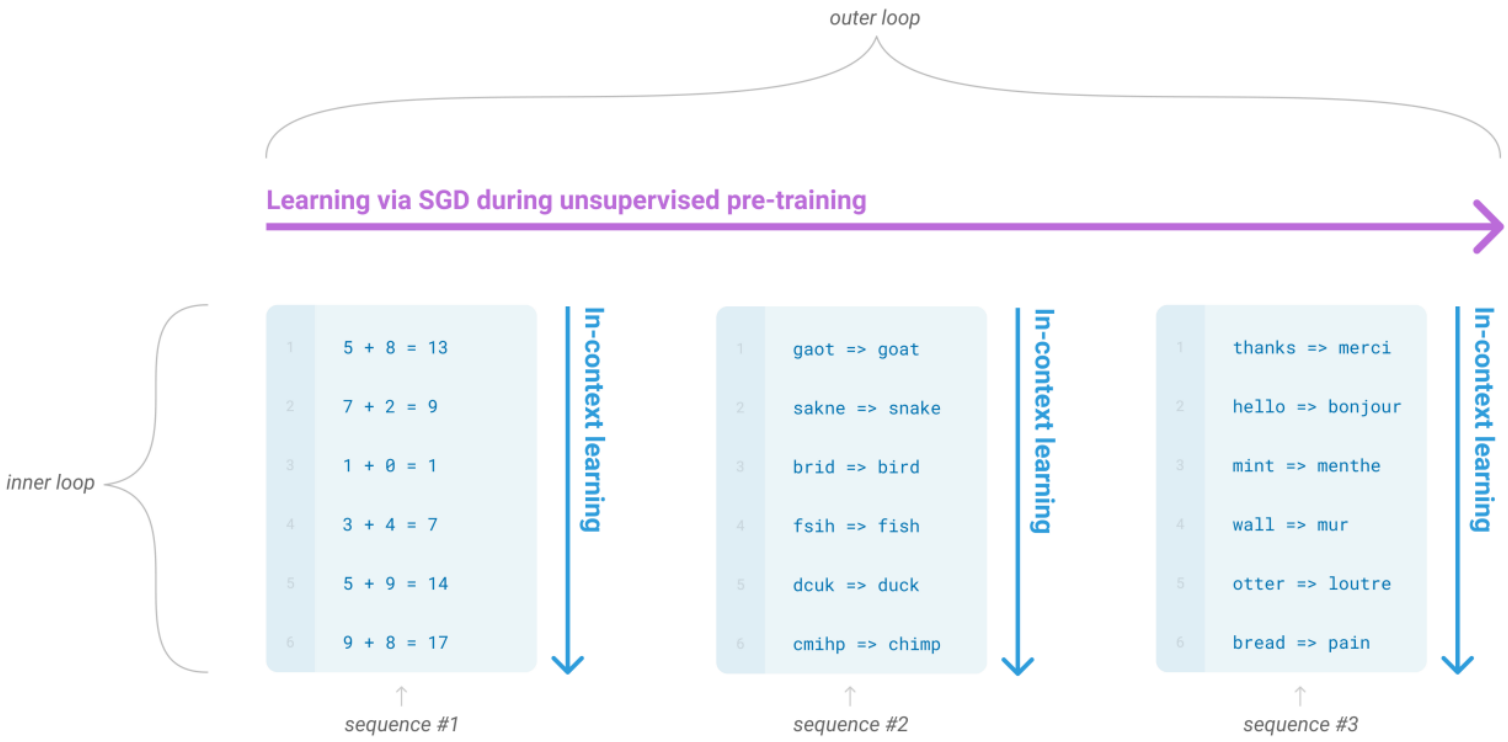
Benjamin Chess **Jack Clark** **Christopher Berner**

Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**

OpenAI

Exemple : GPT-3

- Grande diversité de tâches :
  - corrections orthographiques
  - traduction
  - problèmes de maths simples
  - ...
- $\mathscr{D}_i^{\mathrm{tr}}$: séquence de caractères (servant d'exemples)
- $\mathscr{D}_i^{\mathrm{ts}}$: suite de la séquence



Black-Box Meta-Learning : avantages / inconvénients

Points forts :

- Très expressif
- Compatible avec une grande variété de problèmes (régression, classification, RL)

Points faibles :

- Taches complexes $\Rightarrow$ modèles complexes : problème d'optimisation difficile
- Souvent *data-inefficient*

Table of Contents

Optimization-Based Meta-Learning

- Idée : intégrer le fine-tuning au processus de meta-learning

  $\rightarrow$ on optimise les paramètres du modèle de manière à anticiper les futurs fine-tuning !

- Principal représentant : MAML

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

---

Chelsea Finn [1] **Pieter Abbeel** [1 2] **Sergey Levine** [1]

Proceedings of the 34th International Conference on Machine Learning, PMLR 70:1126-1135, 2017

Objectif du MAML

Fine-tuning pour la tache $\mathcal{T}_i$:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$$

Objectif du MAML

Fine-tuning pour la tache $\mathcal{T}_i$:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$$

Objectif : trouver le θ tel que :

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)}\right)$$

Objectif du MAML

Fine-tuning pour la tache $\mathcal{T}_i$:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$$

Objectif : trouver le θ tel que :

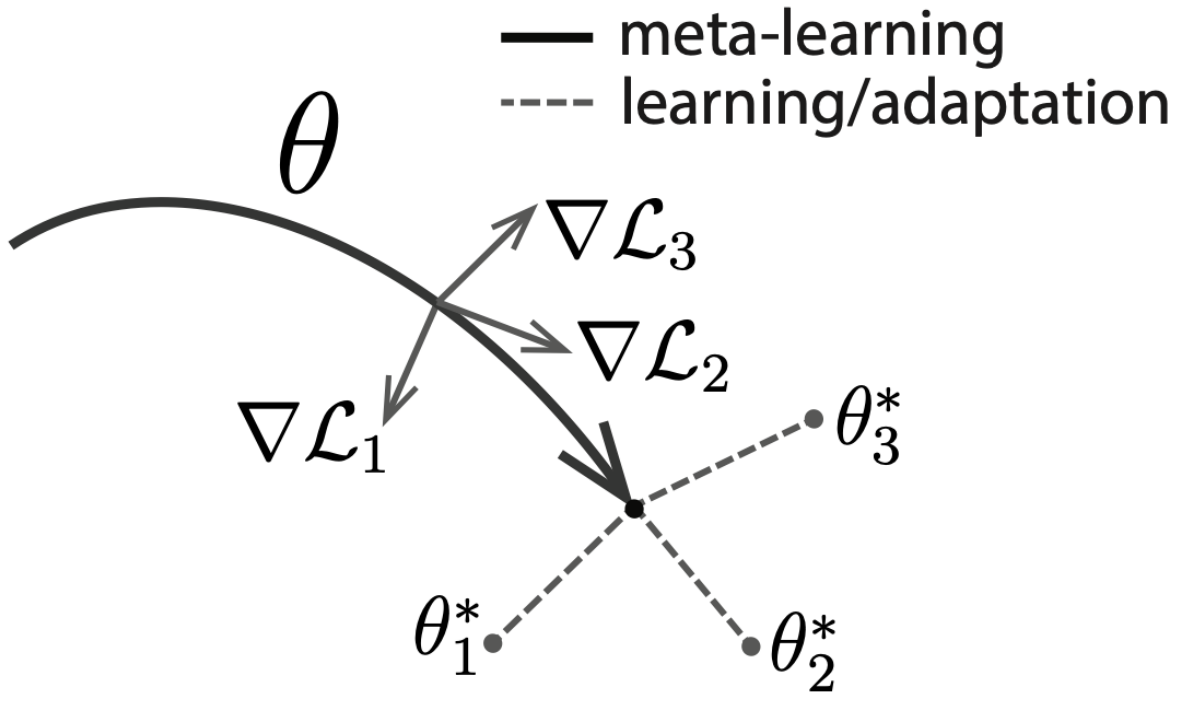
$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)}\right)$$

Meta-optimisation sur toutes les tâches $\mathcal{T}_i \sim p(\mathcal{T})$:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)}\right)$$

$\rightarrow$ Implique un gradient à travers un gradient ! (produits vectoriels hessiens)

MAML : intuition



Algorithme MAML

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
6: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
7: **end for**
8: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
9: **end while**

MAML : Avantages / Inconvénients

Points forts :

- Agnostique du point de vue de la tâche et du modèle

  $\rightarrow$ compatible avec n'importe quel modèle entrainé par descente de gradient
- Pas de paramètres supplémentaires à apprendre

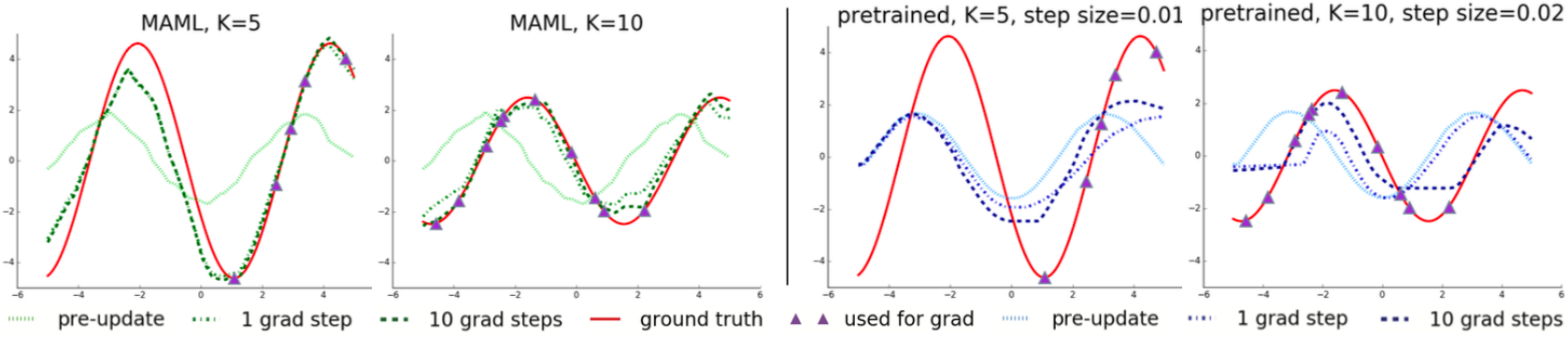
Points faibles :

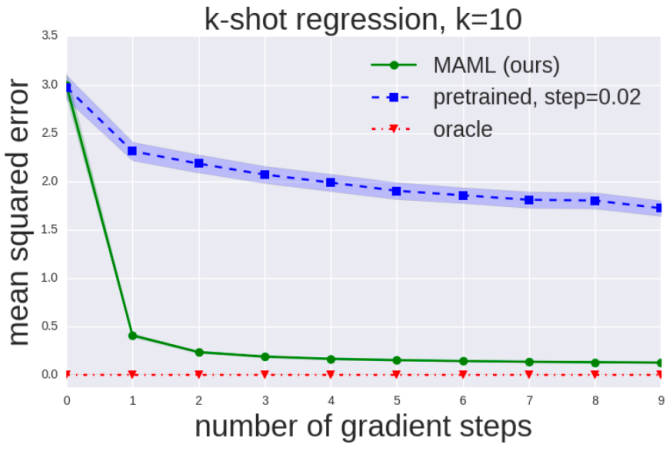
- Dérivée du deuxième ordre : coûteux en calcul et en mémoire

  $\rightarrow$ approximer le premier gradient par l'identité (donne de bons résultats sur les problèmes simples !)
- L'optimisation à deux niveaux rend l'entrainement plus instable

  $\rightarrow$ apprendre le *learning rate* interne, optimiser sous-ensemble des paramètres, introduire des variables de contexte

Résultats: Régression

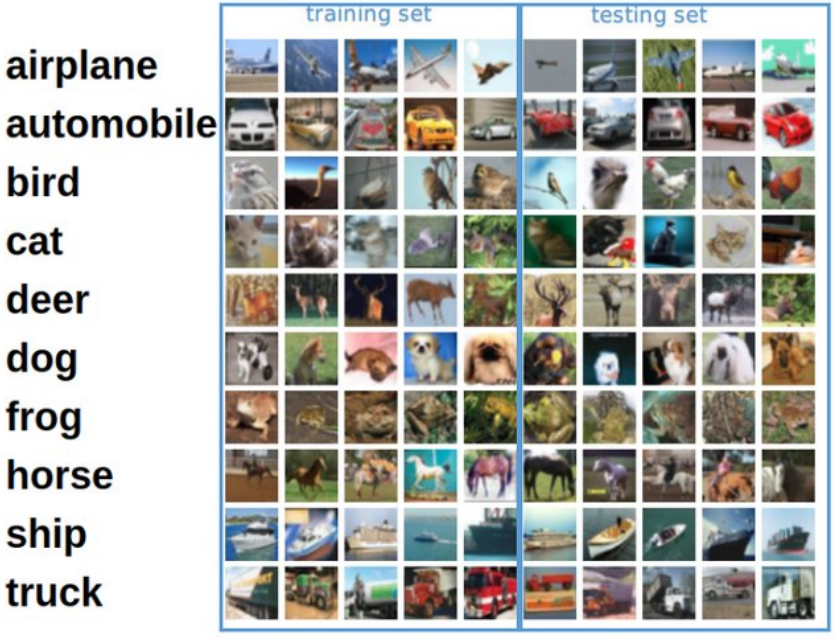




Résultats: Classification - Omniglot

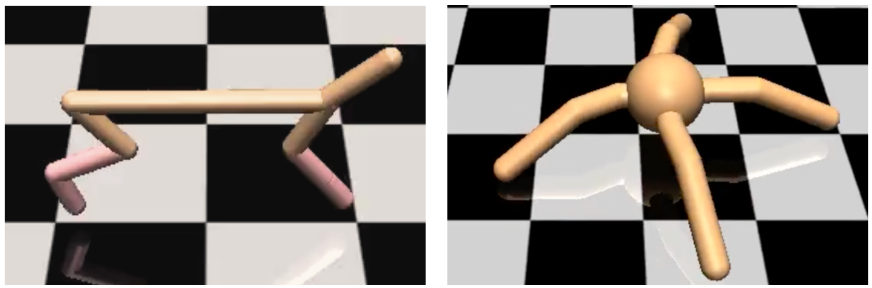
Omniglot (Lake et al., 2011)	5-way Accuracy 1-shot	5-way Accuracy 5-shot	20-way Accuracy 1-shot	20-way Accuracy 5-shot
MANN, no conv (Santoro et al., 2016)	82.8%	94.9%	–	–
MAML, no conv (ours)	**89.7 ± 1.1%**	**97.5 ± 0.6%**	–	–
Siamese nets (Koch, 2015)	97.3%	98.4%	88.2%	97.0%
matching nets (Vinyals et al., 2016)	98.1%	98.9%	93.8%	98.5%
neural statistician (Edwards & Storkey, 2017)	98.1%	99.5%	93.2%	98.1%
memory mod. (Kaiser et al., 2017)	98.4%	99.6%	95.0%	98.6%
MAML (ours)	**98.7 ± 0.4%**	**99.9 ± 0.1%**	**95.8 ± 0.3%**	**98.9 ± 0.2%**

Résultats: Classification - MiniImagenet



MiniImagenet (Ravi & Larochelle, 2017)	5-way Accuracy 1-shot	5-shot
fine-tuning baseline	$28.86 \pm 0.54\%$	$49.79 \pm 0.79\%$
nearest neighbor baseline	$41.08 \pm 0.70\%$	$51.04 \pm 0.65\%$
matching nets (Vinyals et al., 2016)	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
meta-learner LSTM (Ravi & Larochelle, 2017)	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
MAML, first order approx. (ours)	$\mathbf{48.07 \pm 1.75\%}$	$\mathbf{63.15 \pm 0.91\%}$
MAML (ours)	$\mathbf{48.70 \pm 1.84\%}$	$\mathbf{63.11 \pm 0.92\%}$

Résultats: Reinforcement Learning



(a) Half-Cheetah (b) Ant