The power of adaptivity in representation learning:
From meta-learning to federated learning

Sewoong Oh

Paul G. Allen School of Computer Science and Engineering
University of Washington

joint work with

Liam Collins (UT Austin)
Aryan Mokhtari (UT Austin)
Sanjay Shakkottai (UT Austin)
Meta-learning for few-shot learning [FAL17]

Training stage
Meta model $\theta^*_\text{train}$

$$\{(x_{1,j}, y_{1,j})\}_{j=1}^{t_1} \quad \{(x_{2,j}, y_{2,j})\}_{j=1}^{t_2} \quad \ldots \quad \{(x_{n,j}, y_{n,j})\}_{j=1}^{t_n}$$
Meta-learning for few-shot learning [FAL17]

Training stage

Meta model $\theta^{*}_{\text{train}}$

$\{(x_{1,j}, y_{1,j})\}_{j=1}^{t_1}$

$\{(x_{2,j}, y_{2,j})\}_{j=1}^{t_2}$

$\quad \cdots \quad$

$\{(x_{n,j}, y_{n,j})\}_{j=1}^{t_n}$

Test stage

Fine-tuned model $\theta^{*}_{\text{new}}$

$\{(x_j, y_j)\}_{j=1}^t$

New task
Meta-learning for few-shot learning [FAL17]

Training stage
Meta model $\theta^{*}_{\text{train}}$

$\{(x_{1,j}, y_{1,j})\}_{j=1}^{t_1}$ $\{(x_{2,j}, y_{2,j})\}_{j=1}^{t_2}$ $\cdots$ $\{(x_{n,j}, y_{n,j})\}_{j=1}^{t_n}$

Test stage
Fine-tuned model $\theta^{*}_{\text{new}}$

$\{(x_{j}, y_{j})\}_{j=1}^{t}$

New task
Central goal: generalize to new but similar tasks

How to do meta-learning

- Suppose that we have access to
  - (a) Training phase: large number of similar but distinct tasks each with small data
  - (b) Test phase: a small amount of data available just prior to deployment from the deployment environment

- Given this setup how should we train our model?

- Possible Approach:
  - (a) Build a model using data from the training phase
  - (b) Fine-tune the model using the small amount of deployment data

How can we build a model that is easily fine-tunable?

First attempt: Build a model to minimize average training loss, and then fine-tune for deployment
Pooling all the data together

Average Risk Minimization (ARM) + Fine-tuning

- Set of tasks: \( \mathcal{T} = \{ \mathcal{T}_i \}_{i=1}^{n} \) coming from distribution \( p \)
- Select a model \( \theta_{\text{train}}^* \)

Average Risk (Loss) Minimization

\[
\theta_{\text{train}}^* \in \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)
\]

- A new task \( \mathcal{T}_{\text{test}} \) is revealed, drawn according to dist. \( p \)
- Fine-tune the model: \( \theta_{\text{train}}^* \rightarrow \theta_{\text{new}}^* \)
- Performance goal: \( f_{\text{test}}(\theta_{\text{new}}^*) \)
Pooling data has lost the structural information

- Suppose we have images from a large number of classes (e.g., Imagenet)
  - Task = classifying images among a $K$-subset of these classes, small $K$

- ARM + Fine-tuning has mixed performance [FAL17]

<table>
<thead>
<tr>
<th>Model</th>
<th>1-shot Accuracy</th>
<th>5-shot Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>fine-tuning baseline</td>
<td>28.86 ± 0.54%</td>
<td>49.79 ± 0.79%</td>
</tr>
<tr>
<td>nearest neighbor baseline</td>
<td>41.08 ± 0.70%</td>
<td>51.04 ± 0.65%</td>
</tr>
<tr>
<td>matching nets [Vinyals et al., 2016]</td>
<td>43.56 ± 0.84%</td>
<td>55.31 ± 0.73%</td>
</tr>
<tr>
<td>meta-learner LSTM [Ravi &amp; Larochelle, 2017]</td>
<td>43.44 ± 0.77%</td>
<td>60.60 ± 0.71%</td>
</tr>
<tr>
<td>MAML, first order approx. [Finn et al., 2017]</td>
<td>48.07 ± 1.75%</td>
<td>63.15 ± 0.91%</td>
</tr>
<tr>
<td>MAML [Finn et al., 2017]</td>
<td>48.70 ± 1.84%</td>
<td>63.11 ± 0.92%</td>
</tr>
</tbody>
</table>

“Fine-tuning baseline”: Few-shot image classification accuracy of ARM after fine-tuning (image taken from [FAL17])

“Pretrained”: Fine-tuning reward for ARM on robot 2d navigation task (image taken from [FAL17])
Model-Agnostic Meta Learning (MAML) [FAL17]

- Set of tasks: $\mathcal{T} = \{T_i\}_{i=1}^{i=n}$ coming from distribution $p$
- Select a model $\theta^\ast_{train}$ such that

\[
\theta^\ast_{train} \in \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta - \alpha \nabla f_i(\theta))
\]

- A new task $\mathcal{T}_{test}$ is revealed, drawn according to dist. $p$
- Fine-tune the model: $\theta^\ast_{train} \rightarrow \theta^\ast_{new}$
- Performance goal: $f_{test}(\theta^\ast_{new})$

Original motivation: finding the right initialization for adaptation.
Average Risk Minimization (ARM): \( \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) \)

GD update for ARM: \( \theta_{t+1} = \theta_t - \frac{\beta}{n} \sum_{i=1}^{n} \nabla f_i(\theta_t) \)

Gradient evaluated at same \( \theta_t \) for all tasks \( \Rightarrow \) not adaptive
MAML Algorithm: GD on MAML Loss

- **Average Risk Minimization (ARM):** \( \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) \)

- GD update for ARM: \( \theta_{t+1} = \theta_t - \frac{\beta}{n} \sum_{i=1}^{n} \nabla f_i(\theta_t) \)

- Gradient evaluated at same \( \theta_t \) for all tasks \( \implies \) not adaptive

- **Model-Agnostic Meta-Learning (MAML):**

  \[
  \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta - \alpha \nabla f_i(\theta))
  \]

- GD update on MAML loss can be implemented as follows

  \[
  \theta_{t+1} = \theta_t - \frac{\beta}{n} \sum_{i=1}^{n} (I - \alpha \nabla^2 f_i(\theta_t)) \nabla f_i(\theta_t - \alpha \nabla f_i(\theta_t))
  \]
MAML Algorithm: GD on MAML Loss

- **Average Risk Minimization (ARM):** \( \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) \)

- **GD update for ARM:** 
  \[
  \theta_{t+1} = \theta_t - \frac{\beta}{n} \sum_{i=1}^{n} \nabla f_i(\theta_t)
  \]
  Gradient evaluated at same \( \theta_t \) for all tasks \( \implies \) not adaptive

- **Model-Agnostic Meta-Learning (MAML):** 
  \[
  \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta - \alpha \nabla f_i(\theta))
  \]

- **GD update on MAML loss can be implemented as follows**
  \[
  \theta_{t+1} = \theta_t - \frac{\beta}{n} \sum_{i=1}^{n} (I - \alpha \nabla^2 f_i(\theta_t)) \nabla f_i(\theta_t - \alpha \nabla f_i(\theta_t))
  \]
  which can be implemented via inner and outer loops

  - **Inner loop:** Compute \( \theta_{t,i} = \theta_t - \alpha \nabla f_i(\theta_t) \) for \( i = 1, \ldots, n \)
  - **Outer loop:** Compute \( \theta_{t+1} = \theta_t - \frac{\beta}{n} \sum_{i=1}^{n} (I - \alpha \nabla^2 f_i(\theta_t)) \nabla f_i(\theta_{t,i}) \)

- \( \theta_{t,i} \) adapted to each task \( \implies \) adaptive
Empirical observations of MAML

- Original motivation: MAML learns models that **quickly adapt to new tasks** [FAL17, AES19]

- New empirical evidence suggests: MAML learns a good representation **shared across tasks** [RRBV20]
  - Even though it is not designed for representation learning!

- Can we formally prove this conjecture?
Meta-learning from linear regression tasks

Setting from multi-task learning and **linear representation learning**:

- Each task $i$ is linear regression with ground truth parameter $\theta_{*,i} \in \mathbb{R}^d$:

  $$y_i \sim \theta_{*,i}^\top x_i + z_i,$$

  where $x_i$ is a random input vector and $z_i \in \mathbb{R}$ is random zero-mean noise.

- Solving each task individually requires $\Omega(d)$ samples per task.
Meta-learning from linear regression tasks

Setting from multi-task learning and **linear representation learning**:  
- Each task $i$ is linear regression with ground truth parameter $\theta_{\star,i} \in \mathbb{R}^d$:
  \[ y_i \sim \theta_{\star,i}^\top x_i + z_i , \]
- $x_i$ is a random input vector and $z_i \in \mathbb{R}$ is random zero-mean noise.
- Solving each task individually requires $\Omega(d)$ samples per task.

Questions in representation-based meta-learning
- When does solving other tasks help you solve a new task?
- What notion of similarities make meta-learning efficient for linear tasks?
Meta-learning from linear regression tasks

Setting from multi-task learning and **linear representation learning**:

- Each task $i$ is linear regression with ground truth parameter $\theta_{*,i} \in \mathbb{R}^d$:
  \[
  y_i \sim \theta_{*,i}^\top x_i + z_i,
  \]
  
  $x_i$ is a random input vector and $z_i \in \mathbb{R}$ is random zero-mean noise.

- Solving each task individually requires $\Omega(d)$ samples per task.

- Now suppose the $\theta_{*,i}$ lie in a shared $k$-dimensional subspace, $k \ll d$

- Let the columns of $B_* \in \mathbb{R}^{d \times k}$ span this subspace, that is, for each task there exists a corresponding low-dimensional $w_{*,i} \in \mathbb{R}^k$ such that
  \[
  \theta_{*,i} = B_* w_{*,i}
  \]

- If we know $\text{col}(B_*)$, we can solve new tasks with only $O(k)$ samples.
Meta-learning from linear regression tasks

Setting from multi-task learning and **linear representation learning**: 

- Each task $i$ is linear regression with ground truth parameter $\theta_{*,i} \in \mathbb{R}^d$:
  $$y_i \sim \theta_{*,i}^\top x_i + z_i,$$
  $x_i$ is a random input vector and $z_i \in \mathbb{R}$ is random zero-mean noise.

- Solving each task individually requires $\Omega(d)$ samples per task.

- Now suppose the $\theta_{*,i}$ lie in a shared $k$-dimensional subspace, $k \ll d$

- Let the columns of $B_* \in \mathbb{R}^{d \times k}$ span this subspace, that is, for each task there exists a corresponding low-dimensional $w_{*,i} \in \mathbb{R}^k$ such that
  $$\theta_{*,i} = \underbrace{B_*}_{\text{Representation}} \underbrace{w_{*,i}}_{\text{Head}}$$

- If we know $\text{col}(B_*)$, we can solve new tasks with only $O(k)$ samples.

Does GD on ARM learn $B_*$? Does GD on MAML learn $B_*$?
Prior work use matrix completion/sensing techniques

\[ y_i \sim x_i^T B^* w_{*,i} \]

known measurement matrix

unknown low-rank parameter
Prior work use matrix completion/sensing techniques

\[ y_i \sim x_i^T B^* w_{*,i} = \langle \begin{bmatrix} x_i e_i^T \end{bmatrix}, \begin{bmatrix} B^* W^T w_{*,i} \end{bmatrix} \rangle \]

known measurement matrix  unknown low-rank parameter

- [TJJ21, CHMS21, TJNO21] show that although the standard assumptions are not satisfied, i.e.,

Restricted Isometry Property for matrix sensing

Incoherence property for matrix completion

sample efficient learning is possible as long as we have task diversity = small condition number of \( W_* \).

- Can a single parameter algorithm, such as ARM and MAML, learn the ground truth (linear) representation?
MAML for linear representation learning

- Loss function for task $i$ at round $t$:

$$f_i(B, w) := \frac{1}{2} \mathbb{E}_{x_i, y_i} [\langle Bw, x_i \rangle - y_i]^2]$$

- MAML is called a gradient-based meta-learning algorithm (as opposed to representation-based meta-learning)

Algorithm (MAML)

- **(Outer loop)** For $t = 1, \ldots, T$:
  - Sample $n$ linear tasks
  - **(Inner loop)** For each task $i \in \{1, \ldots, n\}$:
    - Adapt: $\begin{bmatrix} w_{t,i} \\ B_{t,i} \end{bmatrix} = \begin{bmatrix} w_t \\ B_t \end{bmatrix} - \alpha \begin{bmatrix} \nabla_w f_i(B_t, w_t) \\ \nabla_B f_i(B_t, w_t) \end{bmatrix}$
    - $\begin{bmatrix} w_{t+1} \\ B_{t+1} \end{bmatrix} = \begin{bmatrix} w_t \\ B_t \end{bmatrix} - \frac{\beta}{n} \sum_{i=1}^{n} (I - \alpha \nabla^2_{w, B} f_i(B_t, w_t)) \begin{bmatrix} \nabla_w f_i(B_{t,i}, w_{t,i}) \\ \nabla_B f_i(B_{t,i}, w_{t,i}) \end{bmatrix}$
MAML vs. ANIL (Almost No Inner Loop)

- Loss function for task $i$ at round $t$:
  $$f_i(B,w) := \frac{1}{2} \mathbb{E}_{x_i,y_i}[(\langle Bw, x_i \rangle - y_i)^2]$$

- MAML is a **gradient-based meta-learning** algorithm
- ANIL is a **representation-based meta-learning** algorithm

**Algorithm (MAML and ANIL)**

- **(Outer loop)** For $t = 1, \ldots, T$:
  - Sample $n$ linear tasks
  - **(Inner loop)** For each task $i \in \{1, \ldots, n\}$:
    - MAML adapts both:
      $$\begin{bmatrix} w_{t,i} \\ B_{t,i} \end{bmatrix} = \begin{bmatrix} w_t \\ B_t \end{bmatrix} - \alpha \begin{bmatrix} \nabla_w f_i(B_t, w_t) \\ \nabla_B f_i(B_t, w_t) \end{bmatrix}$$
    - ANIL adapts only head:
      $$\begin{bmatrix} w_{t,i} \\ B_{t,i} \end{bmatrix} = \begin{bmatrix} w_t \\ B_t \end{bmatrix} - \alpha \begin{bmatrix} \nabla_w f_i(B_t, w_t) \\ 0 \end{bmatrix}$$

- $$\begin{bmatrix} w_{t+1} \\ B_{t+1} \end{bmatrix} = \begin{bmatrix} w_t \\ B_t \end{bmatrix} - \frac{\beta}{n} \sum_{i=1}^{n} H_{t,i,\text{Alg}}(B_t, w_t) \begin{bmatrix} \nabla_w f_i(B_{t,i}, w_{t,i}) \\ \nabla_B f_i(B_{t,i}, w_{t,i}) \end{bmatrix}$$

where $H_{t,i,\text{Alg}}(\cdot)$ is a Hessian that differs between MAML and ANIL
MAML: Evidence of representation learning

- We consider four meta-learning algorithms:
  - ANIL (representation-based meta-learning),
  - MAML (gradient-based meta-learning),
  - their first-order approximations (FO-MAML and FO-ANIL).

**Figure:** MAML learns the true (linear) representation, \( \text{col}(B_*) \), while ARM does not.

- We only evaluate the training phase, assuming that failure to learn the representation leads to failure in few-shot fine-tuning.
Main Results (informal)

- Under the linear representation learning setting

**Informal theorem**

- Under standard assumptions, MAML, ANIL and their first-order analogues recover $\text{col}(B_*)$ exponentially fast when run on the task population losses.

- ANIL and FO-ANIL require $m = \Omega((\frac{d}{n} + 1)k^3) \ll d$ samples per task to recover $\text{col}(B_*)$.

- The key is that MAML and ANIL’s adaptation of the head harnesses task diversity to improve the representation in all directions.

- First results showing that MAML and ANIL provably learn effective representations!

**Informal negative result from [CHMS22]**

There exist problems for which ARM fails to learn $\text{col}(B_*)$. 
We use the **principal angle distance** to measure the distance between representations.

Formally,

$$\text{dist}(B_1, B_2) := \| \hat{B}_{1,\perp} \hat{B}_2 \|_2,$$

where $\hat{B}_{1,\perp}$ and $\hat{B}_2$ are orthonormal matrices s.t. $\text{col}(\hat{B}_{1,\perp}) = \text{col}(B_1)_{\perp}$ and $\text{col}(\hat{B}_2) = \text{col}(B_2)$. 

Average Risk Minimization (ARM) fails to recover $\text{col}(B_*)$

- Let’s focus on the population case to simplify the expressions

\[
\text{ARM: } \min_{B, w} \frac{1}{n} \sum_{i=1}^{n} f_i(B, w)
\]

- Dynamics of GD on ARM:

\[
B_{t+1} \leftarrow B_t \left( I_k - \beta w_t w_t^T \right) + \beta B_* \left( \frac{1}{n} \sum_{i=1}^{n} w_{*,i} \right) w_t^T
\]

- Two issues:
  1. Prior weight only reduces $B_t$ in one direction $\Rightarrow$ slow in forgetting $B_0$
  2. Column space of signal weight is rank one and does not change over time $\Rightarrow$ we only improve in one fixed direction of the true signal $B_*$. 
Average Risk Minimization (ARM) fails to recover $\text{col}(B_*)$

- Let's focus on the population case to simplify the expressions

\[
\text{ARM: } \min_{B,w} \frac{1}{n} \sum_{i=1}^{n} f_i(B, w)
\]

- Dynamics of GD on ARM:

\[
B_{t+1} \leftarrow B_t \left( I_k - \beta w_t w_t^\top \right) + \beta B_* \left( \frac{1}{n} \sum_{i=1}^{n} w_{*,i} \right) w_t^\top
\]

Formal Theorem from [CHMS22]

*For any $\delta \in (0, 0.5]$, $\alpha, T, \{w_{*,i}\}$ and full rank $B_0$, there exists a $B_*$ whose column space is $\delta$-close to $\text{col}(B_0)$, i.e., $\text{dist}(B_0, B_*) = \delta$, while its distance from the representation learned by ARM is at least $0.7\delta$, i.e., $\text{dist}(B_T^{\text{ARM}}, B_*) > 0.7\delta$. 
Dynamics of ANIL, MAML, and FO variations

- For FO-ANIL under population loss, we have

\[
B_{t+1} \leftarrow B_t \left( I_k - \frac{\beta}{n} \sum_{i=1}^{n} w_{t,i} w_{t,i}^\top \right) + \beta B_* \left( \frac{1}{n} \sum_{i=1}^{n} w_{*,i} w_{t,i}^\top \right)
\]

\[
\text{prior weight}
\]

\[
\text{signal weight}
\]

- Suppose

\begin{itemize}
  \item \( \frac{1}{n} \sum_{i=1}^{n} w_{*,i} w_{*,i}^\top \) has small condition number (task diversity), and
  \item \( w_{t,i} \)'s are close to \( w_{*,i} \)'s (head adaptation), then:
\end{itemize}

**Key observation**

Prior weight reduces energy from \( B_t \), and signal weight boosts energy from \( B_* \) in all directions.

\[ \implies \text{Head adaptation and task diversity are critical!} \]
Challenges in proving representation learning

- Need to show **head adaptation**, that the $w_{t,i}$'s are close to the true heads $w_{*,i}$'s

- From the inner loop of ANIL/MAML:

$$w_{t,i} \leftarrow (I_k - \alpha B_t^T B_t)w_t + \alpha B_t^T B_* w_{*,i}$$

  - prior weight shared for all tasks
  - signal weight unique for each task $i$

- In order to show the unique part dominates, we must show three things hold for all $t$:
  1. $\|I_k - \alpha B_t^T B_t\|_2$ is small
  2. $\|w_t\|_2$ is small
  3. $\sigma_{\min}(B_t^T B_*)$ is lower bounded

- Difficult because the algorithms lack explicit regularization and a normalization step.

- Leads to an intricate 6-way induction....
Population case result

Main Theorem [Collins-Mokhtari-O-Shakkottai, ICML 2022]

Suppose there are \( m = \infty \) samples/task, the ground-truth heads satisfy
\[
\mu_\ast^2 I_k \preceq \frac{1}{n} \sum_{i=1}^{n} w_{\ast,i} w_{\ast,i}^\top \preceq L_\ast^2 I_k \quad \text{(Task Diversity)},
\]
and the step sizes \( \alpha, \beta \) are sufficiently small. Then after \( T \) iterations, ANIL, FO-ANIL, MAML, and FO-MAML learn a representation \( B_T \) that satisfies:

\[
\text{dist}(B_T, B_\ast) \leq \left(1 - \Omega(\beta \alpha \mu_\ast^2)\right)^{T-1}
\]
as long as:

- **ANIL, FO-ANIL**: \( \text{dist}(B_0, B_\ast) \leq c \) for a constant \( c \).
- **MAML**: \( \text{dist}(B_0, B_\ast) = O((L_\ast/\mu_\ast)^{-0.75}) \).
- **FO-MAML**: \( \text{dist}(B_0, B_\ast) = O((L_\ast/\mu_\ast)^{-1}) \) and
  \[
  \| \frac{1}{n} \sum_{i=1}^{n} w_{\ast,t,i} \|_2 = O((L_\ast/\mu_\ast)^{-1.5})
  \]

- We also show finite-sample results in the paper.
MAML vs. ANIL

- Recall that our result requires
  1. stronger initialization for MAML and FO-MAML than for ANIL and FO-ANIL, and
  2. for FO-MAML, \( \frac{1}{n} \sum_{i=1}^{n} w_{*,i} \approx 0 \).

- We empirically show these conditions are tight:

  (Left) Random initialization leads MAML and FO-MAML to fail

  (Right) Even with good initialization, \( \frac{1}{n} \sum_{i=1}^{n} w_{*,i} \) far from zero leads FO-MAML to fail

  \( \implies \) MAML/FO-MAML’s updates \( B_t \) in the inner loop, which can inhibit representation learning.
Proof sketch - FO-ANIL (1/4)

\[ B_{t+1} = B_t \left( I_k - \beta \Psi_t \right) + \beta B \* \left( \frac{1}{n} \sum_{i=1}^{n} w_*, i w_{t,i}^\top \right), \]

\[ w_{t,i} = \Delta_t \ w_t + \alpha B_t^\top B_\* w_*, i \]

- Inductive hypotheses:
  - Bounded Head Weight
    \[ A_1(t) := \{ \| w_t \|_2 = O(\sqrt{\alpha}) \} \]
  - Small Head Prior Weight
    \[ A_2(t) := \{ \| \Delta_t \|_2 = \rho \| \Delta_{t-1} \|_2 + O(\beta^2 \alpha^2 \text{dist}_{t-1}^2) \} \]
    \[ A_3(t) := \{ \| \Delta_t \|_2 = O(1) \} \]
  - Small Representation Prior Weight
    \[ A_4(t) := \{ \kappa(\Psi_t) = O(1) \} \]
  - Progress
    \[ A_5(t) := \{ \| B_\* \perp B_t \|_2 = \rho \| B_\* \perp B_{t-1} \|_2 \} \]
    \[ A_6(t) := \{ \text{dist}_t \leq \rho^{t-1} \} \]

where, \( \Delta_t := I_k - \alpha B_t^\top B_t \), \( \Psi_t := \frac{1}{n} \sum_{i=1}^{n} w_t, i w_{t,i}^\top \), and \( \rho := 1 - \Omega(\beta \alpha) \)
Proof sketch - FO-ANIL (2/4)

- Inductive logic:

\[ A_1(t) : \|w_t\|_2 \leq 0.1\sqrt{\alpha \min(1,\mu_t^2/\eta_t^2)\eta_t} \]
\[ A_2(t) : \|\Delta_t\|_2 \leq \rho\|\Delta_{t-1}\|_2 + \alpha^2\beta^2L_*^4\text{dist}_{t-1}^2 \]
\[ A_3(t) : \|\Delta_t\|_2 \leq 0.1 \]
\[ A_4(t) : \|\Delta_t\|_2 \leq \rho\|\Delta_{t-1}\|_2 + \alpha^2\beta^2L_*^4\text{dist}_{t-1} \]
\[ A_5(t) : \|\tilde{B}_t^\top \ast,\perp \tilde{B}_{t-1}\|_2 \leq \rho\|\tilde{B}_t^\top \ast,\perp B_{t-1}\|_2 \]
\[ A_6(t) : \text{dist}_t := \|\tilde{B}_t^\top \ast,\perp B_t\|_2 \leq \rho^{t-1} \]

\[ B_{t+1} = B_t \left( I_k - \beta \Psi_t \right) + \beta B_* \left( \frac{1}{n} \sum_{i=1}^{n} w_{*,i} w_{t,i}^\top \right), \quad \Delta_t := I_k - \alpha B_t^\top B_t \]

Notable implications (1/3):

- \( A_4(t) \implies A_5(t+1) \implies A_6(t+1) \)
  - well-conditioned \( \Psi_t \) implies small prior weight and hence per-step improvement
  - per-step improvements imply geometric convergence
Proof sketch - FO-ANIL (3/4)

- Inductive logic:

\[ A_1(t) : \|w_t\|_2 \leq 0.1\sqrt{\alpha \min(1, \mu_{\mu_2}^2 / \eta^2)} \eta \]

\[ A_2(t) : \|\Delta_t\|_2 \leq \rho \|\Delta_{t-1}\|_2 + \alpha^2 \beta^2 L_s^4 \text{dist}_{t-1}^2 \]

\[ A_3(t) : \|\Delta_t\|_2 \leq 0.1 \]

\[ A_4(t) : 0.9 \alpha \mu_{\mu_2}^2 I_k \leq \Psi_t \leq 1.2 \alpha L_s^2 I_k \]

\[ A_5(t) : \|\hat{B}^\top \otimes B_t\|_2 \leq \rho \|\hat{B}^\top \otimes B_{t-1}\|_2 \]

\[ A_6(t) : \text{dist} := \|\hat{B}^\top \otimes \hat{B}^\top\|_2 \leq \rho^{t-1} \]

\[ w_{t,i} = \Delta_t w_t \quad \text{shared for all } i \]

\[ \alpha B_t^\top B_* w_{*,i} \quad \text{unique for each } i \]

\[ \Psi_t := \frac{1}{n} \sum_{i=1}^{n} w_{t,i} w_{t,i}^\top \]

Notable implications (2/3):

- \( A_1(t + 1), A_3(t + 1), A_6(t + 1) \implies A_4(t + 1) \)

  - Small \( \|\Delta_t\|_2, \|w_t\|_2, \) and \( \text{dist}_t(B_t, B_*) \) implies adapted heads are diverse
Proof sketch - FO-ANIL (4/4)

- Inductive logic:

\[ A_1(t) : \|w_t\|_2 \leq 0.1 \sqrt{\alpha \min(1, \mu_\sigma^2/\eta_\sigma^2)} \eta_\sigma \]
\[ A_2(t) : \|\Delta_t\|_2 \leq \rho \|\Delta_{t-1}\|_2 + \alpha^2 \beta^2 L_4 \text{dist}^2_{t-1} \]
\[ A_3(t) : \|\Delta_t\|_2 \leq 0.1 \]
\[ A_4(t) : 0.9 \alpha E_0 \mu_\sigma^2 I_k \leq \Psi_t \leq 1.2 \alpha L_4 \text{dist}^2_t \]
\[ A_5(t) : \|\hat{B}^\top_{\perp} B_t\|_2 \leq \rho \|\hat{B}^\top_{\perp} B_{t-1}\|_2 \]
\[ A_6(t) : \text{dist}_t := \|\hat{B}^\top_{\perp} B_t\|_2 \leq \rho^{t-1} \]

Notable implications (3/3):

- \( A_2(t) + A_6(t) \implies A_1(t + 1) \)
  - This is tricky as it relies on showing that \( \|\Delta_t\|_2 \) and \( \text{dist}_t \) are summable to show that \( \|w_t\| \) is bounded
Discussion

- We have obtained the first results showing that ANIL and MAML learn effective representations.*

- Inner loop adaptation of the head is key to MAML and ANIL’s ability to learn representations.

- Inner loop adaptation of the representation restricts representation learning for MAML.

---

*L. Collins, A. Mokhtari, S. Oh, S. Shakkottai. MAML and ANIL Provably Learn Representations, ICML 2022*
Connections to federated learning
Connections to federated learning†

- **Distributed Stochastic Gradient Descent (D-SGD)**

![Diagram showing the flow of data and updates between the server and clients.]

- Federated implementation of Average Risk Minimization (ARM):
  \[
  \theta^{t+1} = \theta^t - \alpha \beta \frac{1}{n} \sum_{i=1}^{n} \nabla \theta f_i(\theta^t)
  \]

- **Major difference:** Data never leaves the client device for privacy

Connections to federated learning

- **Federated Average (FedAvg)**‡ performs multiple local updates similar to MAML

\[ \theta^{t+1} \leftarrow \theta^t + \beta \frac{1}{n} \sum_{i \in [n]} \Delta_i^t \]

\[ \theta_1^t \leftarrow \theta^t \]
For \( s = 1, \ldots, \tau \)
\[ \theta_1^t \leftarrow \theta_1^t - \alpha \nabla f_1(\theta_1^t) \]
\[ \Delta_1^t \leftarrow \theta_1^t - \theta^t \]

\[ \theta_n^t \leftarrow \theta^t \]
For \( s = 1, \ldots, \tau \)
\[ \theta_n^t \leftarrow \theta_n^t - \alpha \nabla f_n(\theta_n^t) \]
\[ \Delta_n^t \leftarrow \theta_n^t - \theta^t \]

- Original motivation: communication rounds \( \ll \) number of gradient updates
- New observation: effective representation learner

‡ introduced in [MMRHA17]
Local updates help in personalization [CHMS22]

- **Left plot:** Models trained on 80 classes from CIFAR-100 (with C classes/client) and fine-tuned on new clients from 20 new classes from CIFAR-100
- **Right plot:** Models trained on CIFAR-100 (with C classes/client) and fine-tuned on new clients from CIFAR-10

* $T \tau = 125000$ for both.
  * (FedAvg $\tau = 50$, $T = 2500$, DSGD $\tau = 1$, $T = 125000$)
Representation learned by FedAvg changes less in fine-tuning [CHMS22]

- The early layers of FedAvg’s pre-trained model (corresponding to the representation) change much less than those of D-SGD

- Local updates enable learning the common representation across the clients.
Local updates help in personalization [JKRK19]

- Personalization in FL: Federated trained model is further fine-tuned on client data and evaluated on client data
- FedAvg (left) achieves higher personalization accuracy compared to D-SGD (right)
FedAvg provably learns representations

**Theorem (informal) [CHMS22]**

*Under the linear representation learning setting, if the number of local updates is more than one, i.e., $\tau \geq 2$, FedAvg recovers $\text{col}(B^*)$ exponentially fast when run on the task population losses.*

- The key insight is that FedAvg local updates harness **task diversity** to improve the representation in all directions.

\[
B_{t+1} \approx B_t \left( I_k - \frac{\alpha}{n} \sum_{i=1}^n \sum_{s=0}^{\tau-1} w_{t,i,s} w_{t,i,s}^\top \right) + B_* \left( \frac{\alpha}{n} \sum_{i=1}^n \sum_{s=0}^{\tau-1} w_{*,i} w_{t,i,s}^\top \right)
\]

- **Prior weight** reduces energy from $B_t$, and **signal weight** boosts energy from $B_*$ in all directions.

- **Local updates** and **task diversity** are critical.
Discussion

- We have obtained the first results showing that ANIL and MAML learn effective representations.\(^\S\)

- Inner loop **adaptation of the head** is key to MAML and ANIL’s ability to learn representations.

- Inner loop adaptation of the representation restricts representation learning for MAML.

- Follow-up work by [CMHS22]\(^\¶\) shows that Federated Averaging also learns effective representations.

---

\(^\S\) L. Collins, A. Mokhtari, S. Oh, S. Shakkottai. MAML and ANIL Provably Learn Representations, ICML 2022

References


References


