The Power of Adaptivity in Representation Learning: From Meta-Learning to Federated Learning

Aryan Mokhtari
ECE Department, UT Austin

TILOS - OPTML Seminar
MIT
October 26th, 2022

- MAML and ANIL Provably Learn Representations, ICML 2022.

Liam Collins
UT Austin

Sewoong Oh
UW

Sanjay Shakkottai
UT Austin

- FedAvg with Fine Tuning: Local Updates Lead to Representation Learning, NeurIPS 2022

Liam Collins
UT Austin

Hamed Hassani
UPenn

Sanjay Shakkottai
UT Austin

- Consider the Multi-Task Learning (MTL) setup
  - $\Rightarrow$ We are given a set of tasks
  - $\Rightarrow$ For each task, we have access to some (small number of) samples
  - $\Rightarrow$ The tasks often share some similarity (but are not identical)

- Main Goal: Train a model that generalizes well to new tasks/domains
  - $\Rightarrow$ With **limited data and computation**!

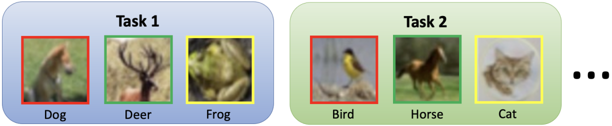
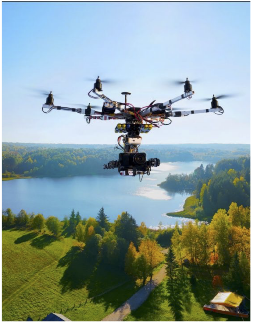


Image credits: Alex Krizhevsky, Learning Multiple Layers of Features from Tiny Images, 2009.

Training Data



Task in new Scenario

Image Credits: bit.ly/3i5m8ay, bit.ly/3w723ZY, bit.ly/3KHMQ5E, bit.ly/3i7pREJ, bit.ly/34I1ytT

- Training data in various scenarios; goal is to quickly adapt to the task in a new scenario

How to do Multi-Task Learning

- Suppose that we have access to

  $\Rightarrow$ large amounts of data from different training environments

  $\Rightarrow$ 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 environments
  - (b) Fine-tune the model using the small amount of deployment data

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

Obvious Approach: Build a model to minimize average training loss, and then fine tune for deployment

- Set of tasks: $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^{i=n}$ coming from distribution p
- Select a model $\boldsymbol{\theta}^*_{train}$

Average Risk (Loss) Minimization

$\boldsymbol{\theta}^*_{train} \in \text{argmin}_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta})$

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

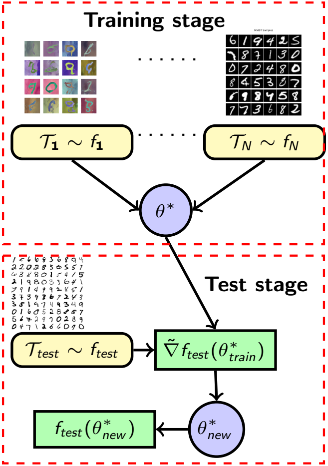


Image Credits: https://bit.ly/392pda9, https://bit.ly/3EEIElq

Does (ARM + Fine Tuning) work well?

- 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
  - Many different subsets == Many tasks

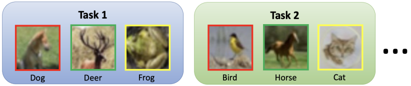
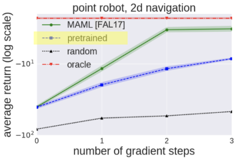


Image credits: Alex Krizhevsky, Learning Multiple Layers of Features from Tiny Images, 2009.

- ARM + Fine Tuning has mixed performance [FAL17]

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. (Finn et al., 2017)	$\mathbf{48.07 \pm 1.75\%}$	$\mathbf{63.15 \pm 0.91\%}$
MAML (Finn et al., 2017)	$\mathbf{48.70 \pm 1.84\%}$	$\mathbf{63.11 \pm 0.92\%}$

"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])

Does (ARM + Fine Tuning) work well?

- These experiments manifest that ARM + Fine-tuning does not work well!

  $\Rightarrow$ The solution trained by ARM often does not generalize well

- For multi-task linear regression, we can formally justify this!
- We show that with adaptivity, we can capture the common structure/representation among the tasks

Takeaway: The adaptivity idea can be done in two ways:

- Change the loss function $\Rightarrow$ as done in MAML
- Change the training procedure $\Rightarrow$ as done in Federated Learning

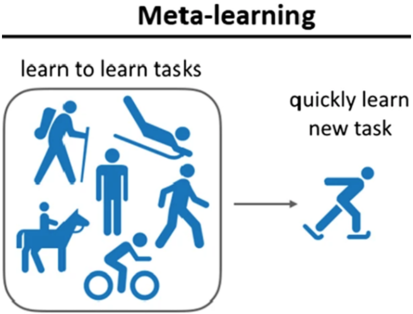


Image credit: https://meta-world.github.io, [HRJ21]

Model-Agnostic Meta Learning [FAL17]

- Set of tasks: $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^{i=n}$ coming from distribution p
- Select a model $\boldsymbol{\theta}^*_{train}$

Change the Loss Function

$\boldsymbol{\theta}^*_{train} \in$
$\text{argmin}_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta} - \alpha \nabla f_i(\boldsymbol{\theta}; (X_i, y_i)))$

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

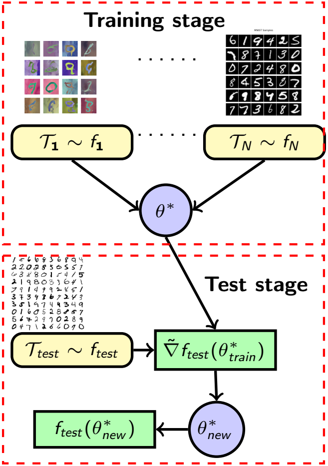


Image Credits: https://bit.ly/392pda9, https://bit.ly/3EEIElq

Seems like finding the right initialization for adaptation!

MAML Algorithm: GD on MAML Loss

- **Average Risk Minimization (ARM)**: $\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta})$
- GD update for ARM.: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{\theta}_t)$
- Gradient evaluated at same $\boldsymbol{\theta}_t$ for all tasks $\implies$ **not adaptive**

MAML Algorithm: GD on MAML Loss

- **Average Risk Minimization (ARM)**: $\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta})$
- GD update for ARM.: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{\theta}_t)$
- Gradient evaluated at same $\boldsymbol{\theta}_t$ for all tasks $\implies$ **not adaptive**

- **Model-Agnostic Meta-Learning (MAML)**: $\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta} - \alpha\nabla f_i(\boldsymbol{\theta}))$
- GD update on MAML loss can be implemented as follows

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n}\sum_{i=1}^{n}(\boldsymbol{I} - \alpha\nabla^2 f_i(\boldsymbol{\theta}_t))\nabla f_i(\boldsymbol{\theta}_t - \alpha\nabla f_i(\boldsymbol{\theta}_t))$$

TEXAS The University of Texas at Austin

- **Average Risk Minimization (ARM)**: $\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta})$
- GD update for ARM.: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{\theta}_t)$
- Gradient evaluated at same $\boldsymbol{\theta}_t$ for all tasks $\implies$ **not adaptive**

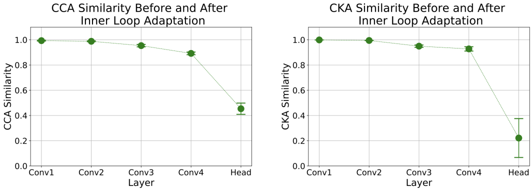
- **Model-Agnostic Meta-Learning (MAML)**: $\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta} - \alpha\nabla f_i(\boldsymbol{\theta}))$
- GD update on MAML loss can be implemented as follows

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n}\sum_{i=1}^{n}(\boldsymbol{I} - \alpha\nabla^2 f_i(\boldsymbol{\theta}_t))\nabla f_i(\boldsymbol{\theta}_t - \alpha\nabla f_i(\boldsymbol{\theta}_t))$$

which can be implemented via inner and outer loops

  - **Inner loop**: Compute $\boldsymbol{\theta}_{t,i} = \boldsymbol{\theta}_t - \alpha\nabla f_i(\boldsymbol{\theta}_t)$ for $i = 1, \ldots, n$
  - **Outer loop**: Compute $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n}\sum_{i=1}^{n}(\boldsymbol{I} - \alpha\nabla^2 f_i(\boldsymbol{\theta}_t))\nabla f_i(\boldsymbol{\theta}_{t,i})$
- $\boldsymbol{\theta}_{t,i}$ adapted to each task $\implies$ **adaptive**

1. MAML learns models that can quickly solve new tasks [FAL17, AES19]
   - In image classification, sinusoid regression, reinforcement learning.
2. MAML seems to be learning a representation shared across tasks [RRBV20]
   - Even though it is not designed for representation learning!



The representation learned by MAML does not change significantly when adapted to each task. Figure credit: [RRBV20]

- Can we formally prove this claim for some multi-task learning setting?

Multi-task linear regression:

- Task i has ground-truth solution $\boldsymbol{\theta}_{*,i} \in \mathbb{R}^d$:

$$y_i \sim \boldsymbol{\theta}_{*,i}^\top \boldsymbol{x}_i + z_i$$

$\Rightarrow$ $\boldsymbol{x}_i$ is a random feature vector and $z_i \in \mathbb{R}$ is random, mean-zero noise.

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

Representation learning in multi-task linear regression

Multi-task linear regression:

- Task i has ground-truth solution $\boldsymbol{\theta}_{*,i} \in \mathbb{R}^d$:

$$y_i \sim \boldsymbol{\theta}_{*,i}^\top \boldsymbol{x}_i + z_i$$

$\Rightarrow$ $\boldsymbol{x}_i$ is a random feature vector and $z_i \in \mathbb{R}$ is random, mean-zero noise.

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

- Now suppose the $\boldsymbol{\theta}_{*,i}$ lie in a shared k-dimensional subspace, $k \ll d$
- Let the columns of $\boldsymbol{B}_* \in \mathbb{R}^{d \times k}$ span this subspace, that is, for all tasks there exists $\boldsymbol{w}_{*,i} \in \mathbb{R}^k$ such that

$$\boldsymbol{\theta}_{*,i} = \boldsymbol{B}_* \boldsymbol{w}_{*,i}$$

- **Task Diversity:** The concatenation of $\boldsymbol{w}_{*,1}, \ldots, \boldsymbol{w}_{*,n}$ spans $\mathbb{R}^k$

- If we know $\mathrm{col}(\boldsymbol{B}_*)$, we can solve new tasks with only $O(k)$ samples

Multi-task linear regression:

- Task i has ground-truth solution $\boldsymbol{\theta}_{*,i} \in \mathbb{R}^d$:

$$y_i \sim \boldsymbol{\theta}_{*,i}^\top \boldsymbol{x}_i + z_i$$

$\Rightarrow$ $\boldsymbol{x}_i$ is a random feature vector and $z_i \in \mathbb{R}$ is random, mean-zero noise.

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

- Now suppose the $\boldsymbol{\theta}_{*,i}$ lie in a shared k-dimensional subspace, $k \ll d$
- Let the columns of $\boldsymbol{B}_* \in \mathbb{R}^{d \times k}$ span this subspace, that is, for all tasks there exists $\boldsymbol{w}_{*,i} \in \mathbb{R}^k$ such that

$$\boldsymbol{\theta}_{*,i} = \boldsymbol{B}_* \boldsymbol{w}_{*,i}$$

- **Task Diversity:** The concatenation of $\boldsymbol{w}_{*,1}, \ldots, \boldsymbol{w}_{*,n}$ spans $\mathbb{R}^k$
- If we know $\mathrm{col}(\boldsymbol{B}_*)$, we can solve new tasks with only $O(k)$ samples
- Does GD on ARM learn $\mathbf{B}^*$? Does GD on MAML learn $\mathbf{B}^*$?

- Loss function for task i:

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

.

- ARM: Uses GD to solve $\min_{\boldsymbol{B},\boldsymbol{w}} \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{B}, \boldsymbol{w})$

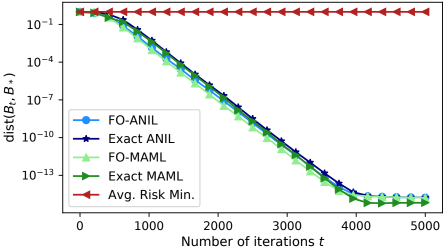
Algorithm

- (Outer loop) For $t = 1, \ldots, T$:
  - Select n tasks satisfying diversity condition $(\text{span}(\{\boldsymbol{w}_{*,i}\}_{i\in[n]}) = \mathbb{R}^k)$
  - (Inner loop) For $i = 1, \ldots, n$:
    - \- Adapt head: $\boldsymbol{w}_{t,i} = \boldsymbol{w}_t - \alpha\nabla_{\boldsymbol{w}} f_i(\boldsymbol{B}_t, \boldsymbol{w}_t)$.
    - \- If MAML: Adapt rep: $\boldsymbol{B}_{t,i} = \boldsymbol{B}_t - \alpha\nabla_{\boldsymbol{B}} f_i(\boldsymbol{B}_t, \boldsymbol{w}_t)$

      If ANIL: **Do not** adapt rep: $\boldsymbol{B}_{t,i} = \boldsymbol{B}_t$
  - $\begin{bmatrix} \boldsymbol{w}_{t+1} \\ \bar{\boldsymbol{B}}_{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_t \\ \bar{\boldsymbol{B}}_t \end{bmatrix} - \frac{\beta}{n}\sum_{i=1}^n \boldsymbol{H}_{t,i,\text{Alg}}(\boldsymbol{B}_t, \boldsymbol{w}_t) \begin{bmatrix} \nabla_{\boldsymbol{w}} f_i(\boldsymbol{B}_{t,i}, \boldsymbol{w}_{t,i}) \\ \nabla_{\boldsymbol{B}} f_i(\boldsymbol{B}_{t,i}, \boldsymbol{w}_{t,i}) \end{bmatrix}$ where $\boldsymbol{H}_{t,i,\text{Alg}}(\boldsymbol{B}_t, \boldsymbol{w}_t)$ is a Hessian that differs between MAML and ANIL.

- We consider four meta-learning algorithms:
  - MAML,
  - ANIL [RRBV19], a close relative of MAML, and
  - their first-order approximations (FO-MAML and FO-ANIL).

(2).pdf



For multi-task linear regression with population losses, the meta-learning approaches learn col($\boldsymbol{B}_*$), while ARM does not.

- How can we explain this? Can we formalize this observation?

Main Results for MAML (Informal)

- In the multi-task linear representation learning setting:

Theorem (informal)

- *Under standard assumptions, MAML, ANIL and their first-order analogues recover* col$(\boldsymbol{B}_*)$ *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 learn the ground-truth subspace.*
- *The key is that MAML and ANIL's adaptation of the head harnesses* ***task diversity*** *to improve the representation in all directions.*

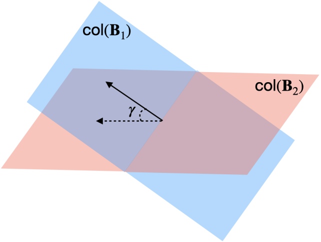
- First results showing that MAML and ANIL provably learn effective representations!

Theorem (informal)

There exist problems for which the model trained by ARM fails to learn col$(\boldsymbol{B}_*)$.

Principal Angle Distance

- We use the **principal angle distance** to measure the distance between representations.



- Formally,

$$\text{dist}(\boldsymbol{B}_1, \boldsymbol{B}_2) := \|\hat{\boldsymbol{B}}_{1,\perp}^\top \hat{\boldsymbol{B}}_2\|_2,$$

where $\hat{\boldsymbol{B}}_{1,\perp}$ and $\hat{\boldsymbol{B}}_2$ are orthonormal matrices s.t. $\text{col}(\hat{\boldsymbol{B}}_{1,\perp}) = \text{col}(\boldsymbol{B}_1)^\perp$ and $\text{col}(\hat{\boldsymbol{B}}_2) = \text{col}(\boldsymbol{B}_2)$.

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

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

- In this case we can show:

$$\boldsymbol{B}_{t+1} = \boldsymbol{B}_t \left(\underbrace{\boldsymbol{I}_k - \beta \boldsymbol{w}_t \boldsymbol{w}_t^\top}_{\text{prior weight}} \right) + \boldsymbol{B}_* \underbrace{\frac{\beta}{n} \sum_{i=1}^{n} \boldsymbol{w}_{*,i} \boldsymbol{w}_t^\top}_{\text{signal weight}}$$

- The prior weight is rank $k-1$, while the signal weight is only rank 1.

Key observation

The representation can only move closer to $\mathrm{col}(\boldsymbol{B}_*)$ in **one** direction on each iteration $\rightarrow$ not clear that it can eventually reach $\mathrm{col}(\boldsymbol{B}_*)$.

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

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

- In this case we can show:

$$\mathbf{B}_{t+1} = \mathbf{B}_t\left(\underbrace{\mathbf{I}_k - \beta \mathbf{w}_t\mathbf{w}_t^\top}_{\text{prior weight}}\right) + \mathbf{B}_* \underbrace{\frac{\beta}{n}\sum_{i=1}^{n} \mathbf{w}_{*,i}\mathbf{w}_t^\top}_{\text{signal weight}}$$

- The prior weight is rank $k-1$, while the signal weight is only rank 1.

Key observation

The representation can only move closer to $\text{col}(\mathbf{B}_*)$ in **one** direction on each iteration $\rightarrow$ not clear that it can eventually reach $\text{col}(\mathbf{B}_*)$.

Theorem

For any $\delta \in (0., 0.5], \alpha, T, \{w_{*,i}\}$ and full rank $\mathbf{B}_0$, there exists a $\mathbf{B}_*$ whose column space is δ-close to col$(\mathbf{B}_0)$, i.e., dist$(\mathbf{B}_0, \mathbf{B}_*) = \delta$, while its distance from the representation learned by ARM is at least 0.7δ, i.e., dist$(\mathbf{B}_T^{ARM}, \mathbf{B}_*) > 0.7\delta$.

- For FO-ANIL, we have

$$\mathbf{B}_{t+1} = \mathbf{B}_t \underbrace{\left(\mathbf{I}_k - \frac{\beta}{n} \sum_{i=1}^{n} \mathbf{w}_{t,i} \mathbf{w}_{t,i}^\top \right)}_{\text{prior weight}} + \mathbf{B}_* \underbrace{\frac{\beta}{n} \sum_{i=1}^{n} \mathbf{w}_{*,i} \mathbf{w}_{t,i}^\top}_{\text{signal weight}}$$

- Suppose $\frac{1}{n}\sum_{i=1}^{n} \mathbf{w}_{t,i}\mathbf{w}_{t,i}^\top$ is full rank (i.e., the $\mathbf{w}_{t,i}$'s are diverse), then:

Key observation

Prior weight reduces energy from $\mathbf{B}_t$, and signal weight boosts energy from $\mathbf{B}_*$ in all directions.
$\implies$ **Head adaptation** and **task diversity** are critical!

- The more diverse the adapted $\mathbf{w}_{t,i}$'s (i.e. smaller condition number of $\frac{1}{n}\sum_{i=1}^{n} \mathbf{w}_{t,i}\mathbf{w}_{t,i}^\top$), the faster convergence rate.

- Need to show that the $\boldsymbol{w}_{t,i}$'s are sufficiently diverse, i.e. evenly spread across $\mathbb{R}^k$.
- We can show:

$$\begin{aligned}\boldsymbol{w}_{t,i} &= \boldsymbol{w}_t - \alpha \nabla f_i(\boldsymbol{B}_t, \boldsymbol{w}_t) \\ &= \underbrace{(\boldsymbol{I}_k - \alpha \boldsymbol{B}_t^\top \boldsymbol{B}_t)\boldsymbol{w}_t}_{\text{shared for all } i} + \underbrace{\alpha \boldsymbol{B}_t^\top \boldsymbol{B}_* \boldsymbol{w}_{*,i}}_{\text{unique for each } i}\end{aligned}$$

- We must show the unique part of $\boldsymbol{w}_{t,i}$ dominates the shared part.

$$\mathbf{w}_{t,i} = \underbrace{(\mathbf{I}_k - \alpha \mathbf{B}_t^\top \mathbf{B}_t)\mathbf{w}_t}_{\text{shared for all } i} + \underbrace{\alpha \mathbf{B}_t^\top \mathbf{B}_* \mathbf{w}_{*,i}}_{\text{unique for each } i}$$

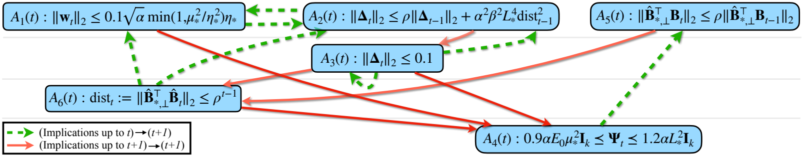
- In order to show the unique part dominates, we must show three things hold for all t:
  1. $\|\mathbf{I}_k - \alpha \mathbf{B}_t^\top \mathbf{B}_t\|_2$ is small
  2. $\|\mathbf{w}_t\|_2$ is small
  3. $\sigma_{\min}(\mathbf{B}_t^\top \mathbf{B}_*)$ is lower bounded $\Longleftarrow$ $\text{dist}(\mathbf{B}_t, \mathbf{B}_*) < 1 - c$ for a positive constant c
- Difficult because the algorithms lack explicit regularization and a normalization step.
- Leads to an intricate 6-way induction....

- Define $\mathbf{\Delta}_t := \mathbf{I}_k - \alpha \mathbf{B}_t^\top \mathbf{B}_t$, $\mathbf{\Psi}_t := \frac{1}{n}\sum_{i=1}^n \mathbf{w}_{t,i}\mathbf{w}_{t,i}^\top$, and $\rho := 1 - \Omega(\beta\alpha)$.
- Inductive hypotheses:
  1. $A_1(t) := \{\|\mathbf{w}_t\|_2 = O(\sqrt{\alpha})\}$ - *global head stays small*
  2. $A_2(t) := \{\|\mathbf{\Delta}_t\|_2 = \rho\|\mathbf{\Delta}_{t-1}\|_2 + O(\beta^2\alpha^2\,\mathrm{dist}_{t-1}^2)\}$ - *$\mathbf{B}_t$ gets closer to orthogonal as* dist_t *decreases*
  3. $A_3(t) := \{\|\mathbf{\Delta}_t\|_2 = O(1)\}$ - *$\mathbf{B}_t$ stays uniformly close to orthogonal*
  4. $A_4(t) := \{\kappa(\mathbf{\Psi}_t) = O(1)\}$ - *adapted heads $\mathbf{w}_{t,i}$ are diverse*
  5. $A_5(t) := \{\|\mathbf{B}_{*,\perp}^\top \mathbf{B}_t\|_2 = \rho\|\mathbf{B}_{*,\perp}^\top \mathbf{B}_{t-1}\|_2\}$ - *energy of $\mathbf{B}_t$ in perpendicular space to* $\mathrm{col}(\mathbf{B}_*)$ *is contracting*
  6. $A_6(t) := \{\mathrm{dist}_t \leq \rho^{t-1}\}$ - *PA distance of $\mathbf{B}_t$ to $\mathbf{B}_*$ is linearly decreasing*

TEXAS
The University of Texas at Austin

$$\boldsymbol{B}_{t+1} = \boldsymbol{B}_t \underbrace{\left(\boldsymbol{I}_k - \frac{\beta}{n} \sum_{i=1}^{n} \boldsymbol{w}_{t,i} \boldsymbol{w}_{t,i}^\top \right)}_{\text{prior weight}} + \boldsymbol{B}_* \underbrace{\frac{\beta}{n} \sum_{i=1}^{n} \boldsymbol{w}_{*,i} \boldsymbol{w}_{t,i}^\top}_{\text{signal weight}} \qquad \boldsymbol{w}_{t,i} = \underbrace{(\boldsymbol{I}_k - \alpha \boldsymbol{B}_t^\top \boldsymbol{B}_t)\boldsymbol{w}_t}_{\text{shared for all } i} + \underbrace{\alpha \boldsymbol{B}_t^\top \boldsymbol{B}_* \boldsymbol{w}_{*,t,i}}_{\text{unique for each } i}$$

- Inductive logic:

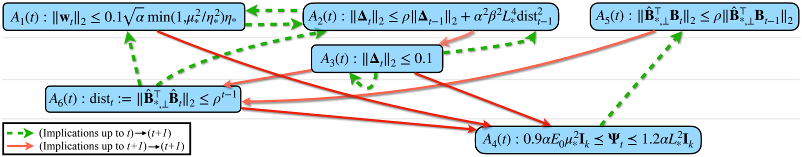


Notable implications (1/3):

- $A_4(t) \implies A_5(t+1) \overset{A_3(t+1)}{\implies} A_6(t+1)$: Adapted head diversity and well-conditioned $\boldsymbol{B}_{t+1}$ implies dist_{t+1} is linearly converging.
  - *Proof.* Use expression for $\boldsymbol{B}_{t+1}$ discussed earlier.

$$\boldsymbol{B}_{t+1} = \boldsymbol{B}_t \underbrace{\left(\boldsymbol{I}_k - \frac{\beta}{n} \sum_{i=1}^{n} \boldsymbol{w}_{t,i} \boldsymbol{w}_{t,i}^\top \right)}_{\text{prior weight}} + \boldsymbol{B}_* \underbrace{\frac{\beta}{n} \sum_{i=1}^{n} \boldsymbol{w}_{*,i} \boldsymbol{w}_{t,i}^\top}_{\text{signal weight}} \qquad \boldsymbol{w}_{t,i} = \underbrace{(\boldsymbol{I}_k - \alpha \boldsymbol{B}_t^\top \boldsymbol{B}_t) \boldsymbol{w}_t}_{\text{shared for all } i} + \underbrace{\alpha \boldsymbol{B}_t^\top \boldsymbol{B}_* \boldsymbol{w}_{*,t,i}}_{\text{unique for each } i}$$

- Inductive logic:

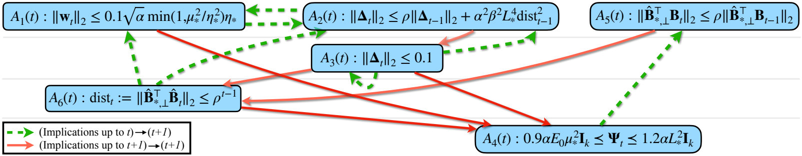


Notable implications (2/3):

- $A_1(t+1), A_3(t+1), A_6(t+1) \implies A_4(t+1)$: Small $\|\boldsymbol{w}_t\|_2$, $\|\boldsymbol{\Delta}_t\|_2$, and dist_t implies adapted heads are diverse.
  - *Proof.* Use expression for $\boldsymbol{w}_{t,i}$ discussed earlier.

TEXAS The University of Texas at Austin

- Inductive logic:



Notable implications (3/3):

- $A_2(t) + A_6(t) \implies A_1(t+1)$: Bounds on $\|\mathbf{\Delta}_t\|_2$ and dist_t imply $\|\mathbf{w}_{t+1}\|_2$ stays small.
    - *Proof.* $A_2(t)$ and the linear convergence of dist_t implies $\|\mathbf{\Delta}_t\|_2$ eventually linearly converges to zero. We can show $\|\mathbf{w}_{t+1}\|_2 \le \|\mathbf{w}_t\|_2 + O(\|\mathbf{\Delta}_t\|_2)$, which implies $\left|\|\mathbf{w}_{t+1}\|_2 - \|\mathbf{w}_t\|_2\right|$ eventually linearly converges to zero.

Main Theorem [Collins-M-Oh-Shakkottai, ICML 2022]

Suppose there are $m = \infty$ *samples/task, the ground-truth heads satisfy* $\mu_*^2 \boldsymbol{I}_k \preceq \frac{1}{n}\sum_{i=1}^n \boldsymbol{w}_{*,t,i}\boldsymbol{w}_{*,t,i}^\top \preceq L_*^2 \boldsymbol{I}_k$*, and the step sizes* α*,* β *are sufficiently small. Then after* T *iterations, ANIL, FO-ANIL, MAML, and FO-MAML learn a representation* $\boldsymbol{B}_T$ *that satisfies:*

$$\mathrm{dist}(\boldsymbol{B}_T, \boldsymbol{B}_*) \leq \left(1 - \Omega(\beta\alpha\mu_*^2)\right)^{T-1}$$

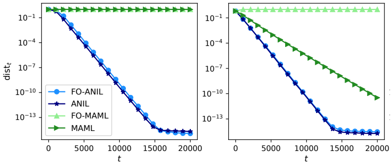
as long as:

- *ANIL, FO-ANIL:* $\mathrm{dist}(\boldsymbol{B}_0, \boldsymbol{B}_*) \leq c$ *for a constant* c*.*
- *MAML:* $\mathrm{dist}(\boldsymbol{B}_0, \boldsymbol{B}_*) = O((L_*/\mu_*)^{-0.75})$*.*
- *FO-MAML:* $\mathrm{dist}(\boldsymbol{B}_0, \boldsymbol{B}_*) = O((L_*/\mu_*)^{-1})$ *and* $\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{w}_{*,t,i}\|_2 = O((L_*/\mu_*)^{-1.5})$*.*

- 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, the mean of $\boldsymbol{w}_{*,i}$'s must be ≈ 0.
- Here we show these conditions are tight:



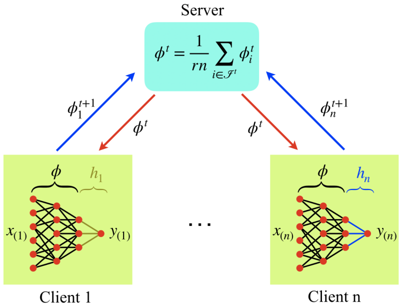
(Left) Random initialization and (Right) structured initialization. In both cases, the mean of the ground-truth heads is far from zero, explaining why FO-MAML fails. MAML succeeds only with structured initialization.

$\implies$ MAML/FO-MAML's inner loop update of the representation can inhibit representation learning.

Discussion

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

- So far we have seen that by adding adaptivity via changes in the function we can learn the representation
- Next, we show that by changing algorithms in the training process that leads to adaptivity we can also learn the representation


Server
$\phi^t = \frac{1}{rn} \sum_{i \in \mathcal{I}^t} \phi_i^t$
ϕ_1^{t+1}
ϕ^t
ϕ^t
ϕ_n^{t+1}
ϕ
h_1
$x_{(1)}$
$y_{(1)}$
ϕ
h_n
$x_{(n)}$
$y_{(n)}$
Client 1
Client n

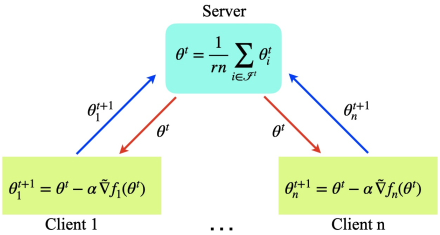
Recap: Average Risk Minimization (ARM)

- **Step 1**: Train a single model by minimizing the average loss

$$\boldsymbol{\theta}^* \in \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta}; \mathcal{D}_i)$$

- **Step 2**: Fine tune the obtained model $\boldsymbol{\theta}^*$ to new task / deployment environment

- Popular approach: Distributed Stochastic Gradient Descent (DSGD)



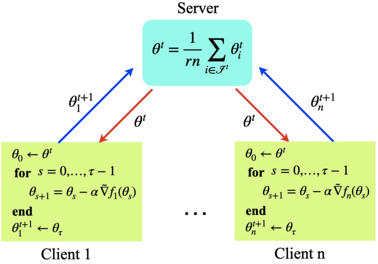
- Effective update $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \frac{\alpha}{rn} \sum_{i \in \mathcal{I}^t} \tilde{\nabla} f_i(\boldsymbol{\theta}^t)$

  $\Rightarrow$ running SGD on the global loss

Same Loss as ARM – Only change is a Distributed Implementation

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \ \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta}; \mathcal{D}_i)$$

- Federated learning is similar to DSGD
  - Except it allows for multiple local updates
- FedAvg: the most common approach!



- Advantage: Number of communication rounds $\ll$ number of updates
  - More communication-efficient than distributed SGD!
- Issue: Drifting effect with data heterogeneity (aka multiple tasks)
  - The local grads ∇f_i are not aligned with the global direction $\sum_{i=1}^{n} \nabla f_i$

From an optimization point of view:

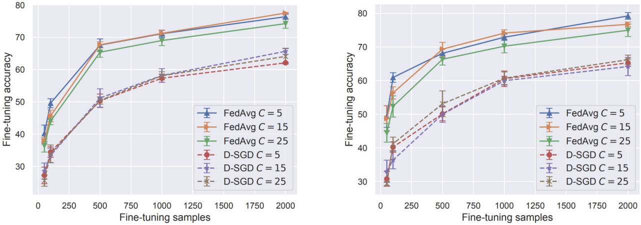
- Due to local drifting, FedAvg may not solve global objective in the data heterogeneous setting
  - [WJ19], [CK21], ...
- There are several other sophisticated methods to control/remove the local drift issue in FedAvg
  - SCAFFOLD: [KKMRSS20]
  - VRL-SGD: [LLZSM20]
  - FEDGATE: [HKMM21]
  - FedNova: [WLLJP20]
  - .....

Our Takeaway

Implicitly, the new implementation has changed the objective

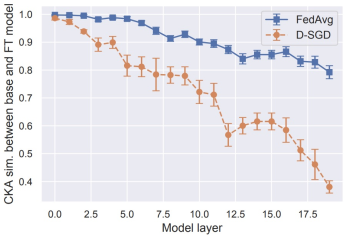
Local updates in FedAvg are useful!

- The local updates are in fact good for generalization!



- 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 $\tau = 1$, $T = 125000$)

- 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 among the tasks!

Open Problems

- *Does FedAvg provably learn representations of heterogeneous tasks?*
- *Are the local updates are essential for learning representations?*

- Multi-task linear regression.
- Task i has ground-truth solution $\boldsymbol{\theta}_{*,i} \in \mathbb{R}^d$:

$$y_i \sim \boldsymbol{\theta}_{*,i}^\top \boldsymbol{x}_i + z_i$$

  - $\boldsymbol{x}_i$ is a random feature vector, $z_i \in \mathbb{R}$ is random, mean-zero noise

- Now suppose the $\boldsymbol{\theta}_{*,i}$ lie in a shared k-dimensional subspace, $k \ll d$
- Let the columns of $\boldsymbol{B}_* \in \mathbb{R}^{d \times k}$ span this subspace, that is, for all tasks there exists $\boldsymbol{w}_{*,i} \in \mathbb{R}^k$ such that

$$\boldsymbol{\theta}_{*,i} = \boldsymbol{B}_* \boldsymbol{w}_{*,i}$$

Task Diversity: The concatenation of $\boldsymbol{w}_{*,1}, \ldots, \boldsymbol{w}_{*,n}$ spans $\mathbb{R}^k$

- Now suppose $\tau = 1$ and we have the update of DGD

$$\boldsymbol{B}_{t+1} = \boldsymbol{B}_t \underbrace{\left(\boldsymbol{I}_k - \alpha \boldsymbol{w}_t \boldsymbol{w}_t^\top\right)}_{\text{prior weight}} + \alpha \boldsymbol{B}_* \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}_{*,i}\right) \boldsymbol{w}_t^\top}_{\text{signal weight}}$$

- The prior weight is rank $k-1$, while the signal weight is only rank 1.
- The representation can only move closer to $col(\boldsymbol{B}_*)$ in **one** direction on each iteration

Theorem

For any $\delta \in (0., 0.5], \alpha, T, \{w_{*,i}\}$ and full rank B_0, there exists a $\boldsymbol{B}_*$ whose column space is δ-close to $col(\boldsymbol{B}_0)$, i.e., $dist(\boldsymbol{B}_0, \boldsymbol{B}_*) = \delta$, while its distance from the representation learned by DGD is at least 0.7δ, i.e., $dist(\boldsymbol{B}_T^{DGD}, \boldsymbol{B}_*) > 0.7\delta$.

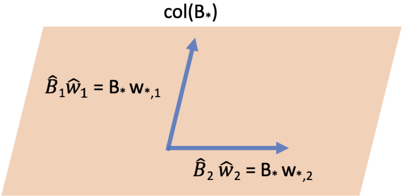
- DGD cannot guarantee to recover the ground-truth representation.

- Recall that the local updates are with respect to the local loss:

$$f_i(\boldsymbol{B}, \boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{B}\boldsymbol{w} - \boldsymbol{B}_*\boldsymbol{w}_{*,i}\|_2^2$$

- Let $\hat{\boldsymbol{B}}_i, \hat{\boldsymbol{w}}_i$ be the result of τ local updates for client i
- (Naive) idea: Use fact that if $\hat{\boldsymbol{B}}_i\hat{\boldsymbol{w}}_i = \boldsymbol{B}_*\boldsymbol{w}_{*,i}$ for all i, then the local products $\{\hat{\boldsymbol{B}}_i\hat{\boldsymbol{w}}_i\}_i$ all lie in the correct subspace

$\Rightarrow$ and $\text{span}(\{\hat{\boldsymbol{B}}_i\hat{\boldsymbol{w}}_i\}_i) = \text{col}(\boldsymbol{B}_*)$

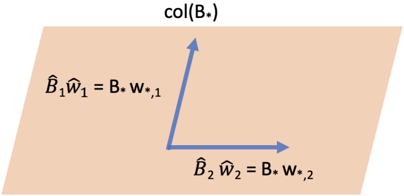


- Recall that the local updates are with respect to the local loss:

$$f_i(\boldsymbol{B}, \boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{B}\boldsymbol{w} - \boldsymbol{B}_*\boldsymbol{w}_{*,i}\|_2^2$$

- Let $\hat{\boldsymbol{B}}_i, \hat{\boldsymbol{w}}_i$ be the result of τ local updates for client i
- (Naive) idea: Use fact that if $\hat{\boldsymbol{B}}_i\hat{\boldsymbol{w}}_i = \boldsymbol{B}_*\boldsymbol{w}_{*,i}$ for all i, then the local products $\{\hat{\boldsymbol{B}}_i\hat{\boldsymbol{w}}_i\}_i$ all lie in the correct subspace

$\Rightarrow$ and $\text{span}(\{\hat{\boldsymbol{B}}_i\hat{\boldsymbol{w}}_i\}_i) = \text{col}(\boldsymbol{B}_*)$



- However, this does not imply anything meaningful about $\boldsymbol{B}_{t+1} := \frac{1}{n}\sum_{i=1}^{n}\hat{\boldsymbol{B}}_i$

- Let $k = n = 2$, and suppose $\boldsymbol{\theta}_{*,1} = \boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, $\boldsymbol{\theta}_{*,2} = \boldsymbol{e}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$

$$\implies \mathrm{col}(\boldsymbol{B}_*) = \mathrm{span}(\boldsymbol{e}_1, \boldsymbol{e}_2)$$

- Let $k = n = 2$, and suppose $\boldsymbol{\theta}_{*,1} = \boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, $\boldsymbol{\theta}_{*,2} = \boldsymbol{e}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$

  $\implies \mathrm{col}(\boldsymbol{B}_*) = \mathrm{span}(\boldsymbol{e}_1, \boldsymbol{e}_2)$

- Let $\hat{\boldsymbol{B}}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$, $\hat{\boldsymbol{w}}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\hat{\boldsymbol{B}}_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$, $\hat{\boldsymbol{w}}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

- Then $\hat{\boldsymbol{B}}_1 \hat{\boldsymbol{w}}_1 = \boldsymbol{B}_* \boldsymbol{w}_{*,1}$ and $\hat{\boldsymbol{B}}_2 \hat{\boldsymbol{w}}_2 = \boldsymbol{B}_* \boldsymbol{w}_{*,2}$, yet

$$\boldsymbol{B}_{t+1} = \tfrac{1}{2}\hat{\boldsymbol{B}}_1 + \tfrac{1}{2}\hat{\boldsymbol{B}}_2 = [\tfrac{1}{2}(\boldsymbol{e}_1 + \boldsymbol{e}_2), \mathbf{0}]$$

  has $\mathrm{dist}(\boldsymbol{B}_{t+1}, \boldsymbol{B}_*) = 1\ldots.$

  $\Rightarrow$ Can't rely only on local convergence!

Theorem (informal) [Collins-Hassani-M-Shakkottai, NeurIPS 2022]

If the number of local updates satisfies $\tau \geq 2$, FedAvg recovers col$(\boldsymbol{B}^*)$ exponentially fast when run on the task population losses.

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

$$\boldsymbol{B}_{t+1} = \boldsymbol{B}_t \underbrace{\left(\frac{1}{n} \sum_{i=1}^{n} \prod_{s=0}^{\tau-1} \left(\boldsymbol{I}_k - \alpha \boldsymbol{w}_{t,i,s} \boldsymbol{w}_{t,i,s}^\top \right) \right)}_{\text{prior weight}}$$

$$+ \boldsymbol{B}_* \underbrace{\left(\frac{\alpha}{n} \sum_{i=1}^{n} \boldsymbol{w}_{*,i} \sum_{s=0}^{\tau-1} \boldsymbol{w}_{t,i,s}^\top \prod_{r=s+1}^{\tau-1} \left(\boldsymbol{I}_k - \alpha \sum_{i=1}^{n} \boldsymbol{w}_{t,i,r} \boldsymbol{w}_{t,i,r}^\top \right) \right)}_{\text{signal weight}}$$

- Prior weight reduces energy from $\boldsymbol{B}_t$, and signal weight boosts energy from $\boldsymbol{B}_*$ in all directions

Main Results (FedAvg Learns Representation)

Theorem (informal) [Collins-Hassani-M-Shakkottai]

If the number of local updates satisfies $\tau \geq 2$, FedAvg recovers col$(\boldsymbol{B}^*)$ exponentially fast when run on the task population losses.

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

$$\mathbf{B}_{t+1} \approx \mathbf{B}_t \Bigg(\underbrace{\boldsymbol{I}_k - \frac{\alpha}{n} \sum_{i=1}^{n} \sum_{s=0}^{\tau-1} \mathbf{w}_{t,i,s} \mathbf{w}_{t,i,s}^\top}_{\text{prior weight}} \Bigg) + \mathbf{B}_* \Bigg(\underbrace{\frac{\alpha}{n} \sum_{i=1}^{n} \sum_{s=0}^{\tau-1} \mathbf{w}_{*,i} \mathbf{w}_{t,i,s}^\top}_{\text{signal weight}} \Bigg)$$

- Prior weight reduces energy from $\boldsymbol{B}_t$, and signal weight boosts energy from $\boldsymbol{B}_*$ in all directions
  - Local updates and task diversity are critical!

- **Client Diversity Assumption:** Let $\mathbf{W}_* := [\mathbf{w}_{*,1}, \ldots, \mathbf{w}_{*,n}]$. Then $\sigma_{\min,*} := \sigma_{\min}(\mathbf{W}_*) > 0$.

Theorem [Collins-Hassani-M-Shakkottai, NeurIPS '22]

If the Client Diversity Assumption holds, the number of local updates between rounds satisfies $\tau \geq 2$, the step size $\alpha = O(\frac{1}{\sqrt{\tau}})$, and the initialization satisfies $\text{dist}(\mathbf{B}_0, \mathbf{B}_*) \leq 1 - c$ *for some $c \in (0, 1]$, then for any $\epsilon \in (0, 1)$, FedAvg learns a representation $\mathbf{B}_T$ satisfying* $\text{dist}(\mathbf{B}_T, \mathbf{B}_*) \leq \epsilon$ *after at most*

$$T = O\left(\frac{\log(1/\epsilon)}{\alpha^2 \tau \sigma_{\min,*}^2}\right)$$

communication rounds.

- **Client Diversity Assumption:** Let $\mathbf{W}_* := [\mathbf{w}_{*,1}, \ldots, \mathbf{w}_{*,n}]$. Then $\sigma_{\min,*} := \sigma_{\min}(\mathbf{W}_*) > 0$.

Theorem [Collins-Hassani-M-Shakkottai, NeurIPS '22]

If the Client Diversity Assumption holds, the number of local updates between rounds satisfies $\tau \geq 2$, the step size $\alpha = O(\frac{1}{\sqrt{\tau}})$, and the initialization satisfies $\mathrm{dist}(\mathbf{B}_0, \mathbf{B}_*) \leq 1 - c$ for some $c \in (0, 1]$, then for any $\epsilon \in (0, 1)$, FedAvg learns a representation $\mathbf{B}_T$ satisfying $\mathrm{dist}(\mathbf{B}_T, \mathbf{B}_*) \leq \epsilon$ after at most

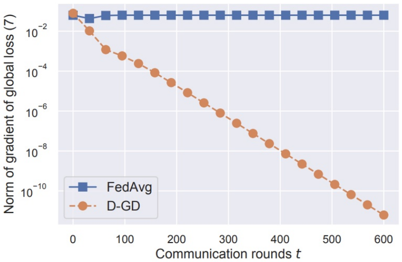
$$T = O\left(\frac{\log(1/\epsilon)}{\alpha^2 \tau \sigma_{\min,*}^2}\right)$$

communication rounds.

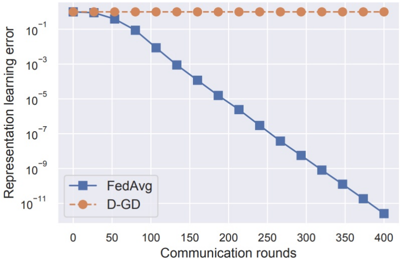
- First result showing that FedAvg learns an expressive representation in any setting!

Important Observations and Conclusion (1/2)

- **Optimization point of view**
  - DGD finds a stationary point of the average loss, while FedAvg does not

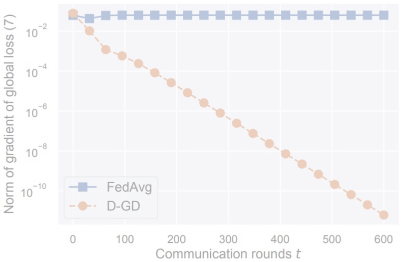


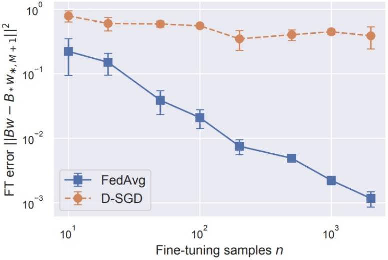
- **Generalization point of view**
  - FedAvg learns the correct representation (principal angle distance)



- Optimization point of view
  - DGD finds a stationary point of the average loss, while FedAvg does not

- Generalization point of view
  - FedAvg generalizes better (loss with new task, after fine tuning)!





[FAL17] C. Finn, P. Abbeel, S. Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Neural Networks, *ICML*, 2017.

[AES19] A. Antoniou, H. Edwards, A. Storkey. How to Train Your MAML, *ICLR*, 2019.

[RRBV19] A. Raghu, M. Raghu, S. Bengio, O. Vinyals. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML, *ICLR*, 2020.

[HRJ21] M. Huisman, J. N. van Rijn, A. Plaat. A Survey of deep Meta-Learning, *Artificial Intelligence Review* Volume 54, pages 4483-4541, 2021.

[LLZSM20] Liang, Liu, Ziyin, Salakhutdinov, and Morency. Think locally, act globally: Federated learning with local and global representations, 2020.

[WLLJP20] Wang, Liu, Liang, Joshi, and Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization, *NeurIPS* 2020.

[CK21] Charles, Konecny. Convergence and accuracy trade-offs in federated learning and meta-learning, *AISTATS* 2021.

[WJ18] Wang, Joshi. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms, 2018.

[KKMRSS20] Karimireddy, Kale, Mohri, Reddi, Stich, and Suresh. Scaffold: Stochastic controlled averaging for federated learning, *ICML* 2020.

[HKMM21] Haddadpour, Kamani, Mokhtari, Mahdavi. Federated Learning with Compression: Unified Analysis and Sharp Guarantees, *AISTATS* 2021.