TRANSFERRING KNOWLEDGE ACROSS LEARNING PROCESSES

Sebastian Flennerhag[a], Pablo G. Moreno[b], Neil D. Lawrence[b], Andreas Damianou[b]
[a]The Alan Turing Institute, [b]Amazon

SUMMARY

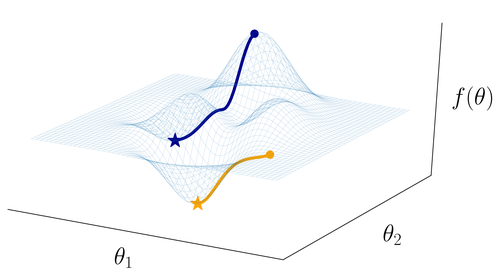
- We propose Leap: a light-weight meta-learner designed for long training processes–even millions of gradient descent steps.
- In complex transfer learning scenarios, current approaches in transfer learning and meta-learning fail as they don't take the *process* of learning into account.
- Leap does this by learning an initialization that minimises the distance a task learner has to travel on the task loss surface.
- We show that Leap outperforms alternative methods across a variety complex transfer learning scenarios.

MEASURING THE LENGTH OF A LEARNING PROCESS

Given a learning objective f that maps parameters θ to a scalar loss value, we focus on gradient based training:

$$\theta^{i+1} = \theta^i - \alpha \nabla f(\theta^i). \tag{1}$$

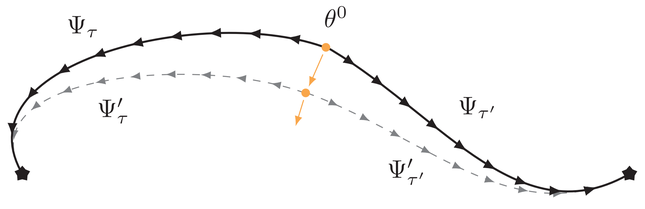
This process approximates a smooth *curve* γ on the loss surface:



We approximate its length by the cumulative chordal distance:

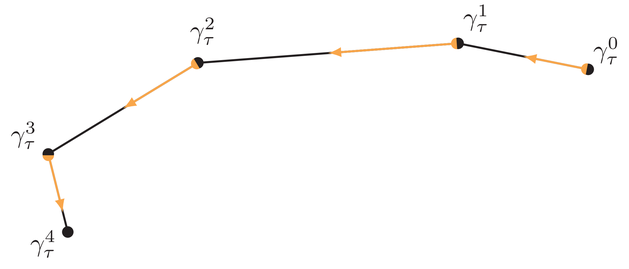
$$\text{Length}(\gamma) \approx \sum_{i=0}^{K-1} \|\gamma^{i+1} - \gamma^i\| = d(\theta^0), \tag{2}$$

where $\gamma^i = (\theta^i, f(\theta^i))$. We transfer knowledge by minimizing the *expected gradient path length*, $\mathbb{E}_\tau\left[d(\theta^0)\right]$, across a task gradient paths Ψ_τ:



LEAP MINIMIZES THE EXPECTED DISTANCE TO TRAVEL

To minimize $\mathbb{E}_\tau\left[d(\theta^0)\right]$ we pull the initialization forward along the observed gradient paths $(\gamma^0, \dots \gamma^K)$, where $\gamma^i = (\theta^i, f(\theta^i))$.



Pulling the initialization forward is accomplished by freezing γ^{i+1} in each norm term, denoted $\bar{\gamma}^{i+1}$. This pulls γ^i forward which in turns pulls the initialization forward. Leap's meta-objective is

$$\min_{\theta^0} \bar{F}(\theta^0) = \mathbb{E}_\tau\left[\sum_{i=0}^{K-1} \left\|\bar{\gamma}_\tau^{i+1} - \gamma_\tau^i\right\|\right], \qquad \gamma_\tau^0 = (\theta^0, f(\theta^0)). \tag{3}$$

FINDS SHORTEST PATHS WITH BEST-CASE PERFORMANCE

Gradient descent on eq. 3 yields initializations with incrementally shorter expected gradient paths that retains or improves performance:

Theorem 1 (Pull-forward). *Define a sequence of initializations* $\{\theta_s^0\}_{s\in\mathbb{N}}$ *by*

$$\theta_{s+1}^0 = \theta_s^0 - \beta_s \nabla \bar{F}(\theta_s^0). \tag{4}$$

For $\beta_s > 0$ *sufficiently small, there exist learning rates for all tasks such that* $\theta_{k\to\infty}^0$ *is a limit point with* $f(\theta_{k\to\infty}^0) \leq f(\theta_0^0)$.

META-GRADIENT DESCENT FOR FREE (ALMOST)

If $\partial\theta^k/\partial\theta^0 \approx I_n$, the meta gradient can be approximated on the fly at almost no cost during task training:

$$\nabla \bar{F}(\theta^0) \approx -\frac{1}{T}\sum_{\tau=0}^{T}\sum_{i=0}^{K} \frac{\Delta f_\tau^i \, \nabla f_\tau(\theta_\tau^i) + \Delta\theta_\tau^i}{\|\Delta\gamma_\tau^i\|} \tag{5}$$

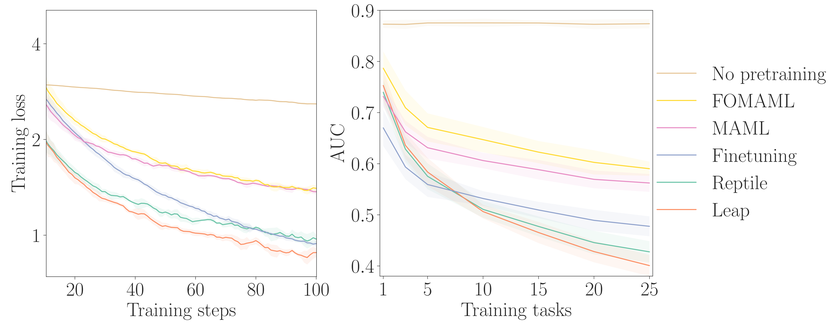
Leap is also trivial to parallelize, only the initialization is shared.

ALGORITHM 1: LEAP

Require: β_s, $p(\tau)$, $\tau = (f_\tau, u_\tau, p_\tau)$: distribution over tasks
randomly initialize θ^0
while not done **do**
  $\nabla \leftarrow 0$: initialize meta gradient
  sample task batch $\mathcal{B}$ from $p(\tau)$
  for all $\tau \in \mathcal{B}$ **do**
    $\theta_\tau^0 \leftarrow \theta^0$: initialize task baseline
    for all $i \in \{0, \dots, K_\tau - 1\}$ **do**
      $\theta_\tau^{i+1} \leftarrow u_\tau(\theta_\tau^i)$: update baseline
      increment ∇ using the pull-forward gradient (Eq. 5)
    end for
  end for
  $\theta^0 \leftarrow \theta^0 - \frac{\beta}{|\mathcal{B}|}\nabla$: update initialization
end while

EXPERIMENTS

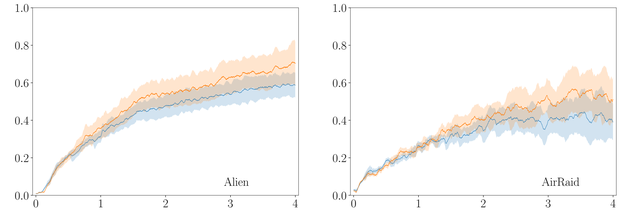
- Multi-shot Omniglot (each alphabet is a task). Mean training curves and error AUC on held-out tasks:



- Multi-CV: a dataset is a task; mean normalized improvement:

	Leap	Finetuning	Progressive Nets [3]	HAT [4]
AUC	**0.74**	0.90	1.06	1.09
Test Error	**0.89**	1.20	0.97	1.15

- Atari: each game is a task; examples of Leap vs. random init:



References

[1] Finn et. al.. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ICML, 2017.
[2] Nichol et. all.. On First-Order Meta-Learning Algorithms. arxiv:1803:02999, 2018.
[3] Rusu et. al.. Progressive Neural Networks. arxiv:1606:04671, 2016.
[4] Serra et. al.. Overcoming Catastrophic Forgetting with Hard Attention to the Task. ICML, 2018.