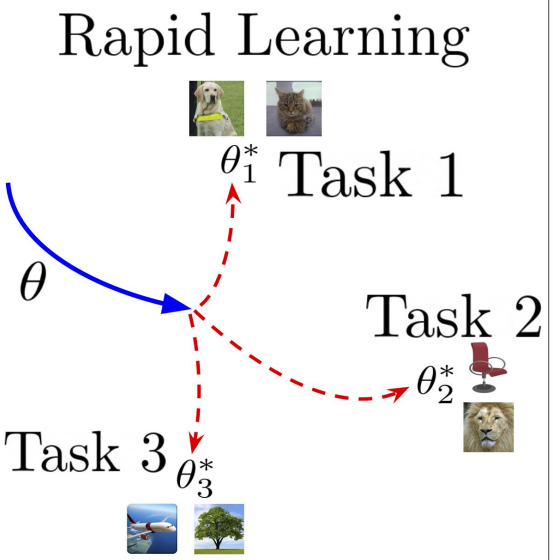
Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML

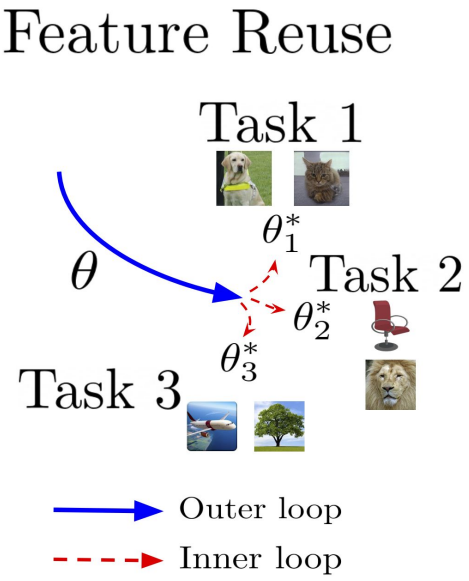
Aniruddh Raghu, Maithra Raghu,
Samy Bengio, Oriol Vinyals

In this paper

- Where does the superior of MAML comes frome?
  - Rapid learning
  - Feature reuse
- Almost No Inner Loop (ANIL) and No Inner Loop (NIL)
-

Rapid Learning vs Feature Reuse

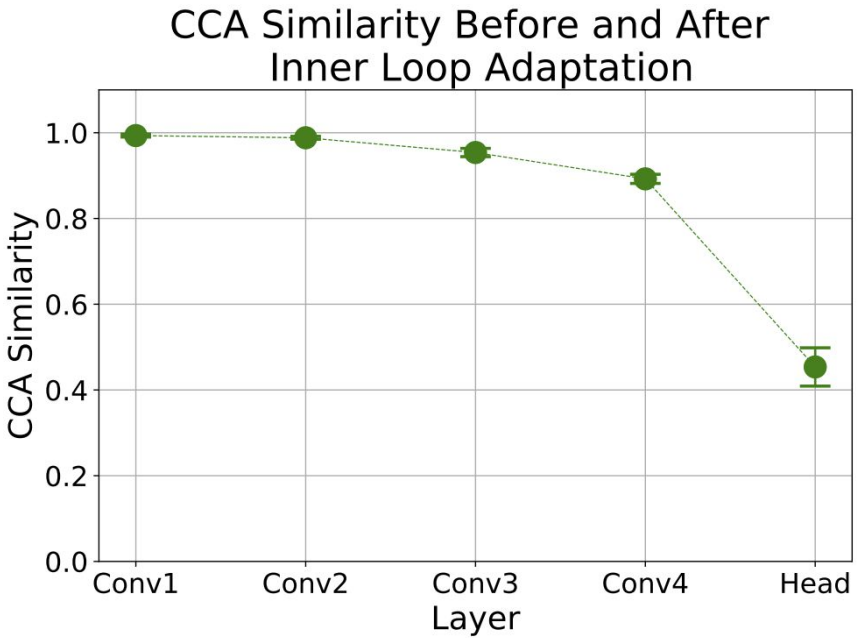


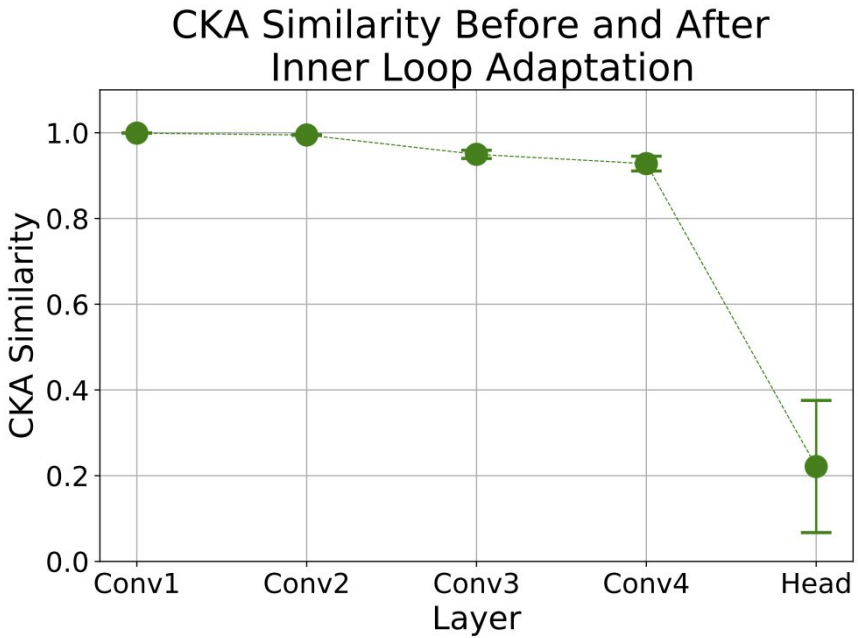


Outer loop performs meta-initialization
Feature reuse

Inner loop performs task-adaptation
Rapid learning

Examine Feature Update



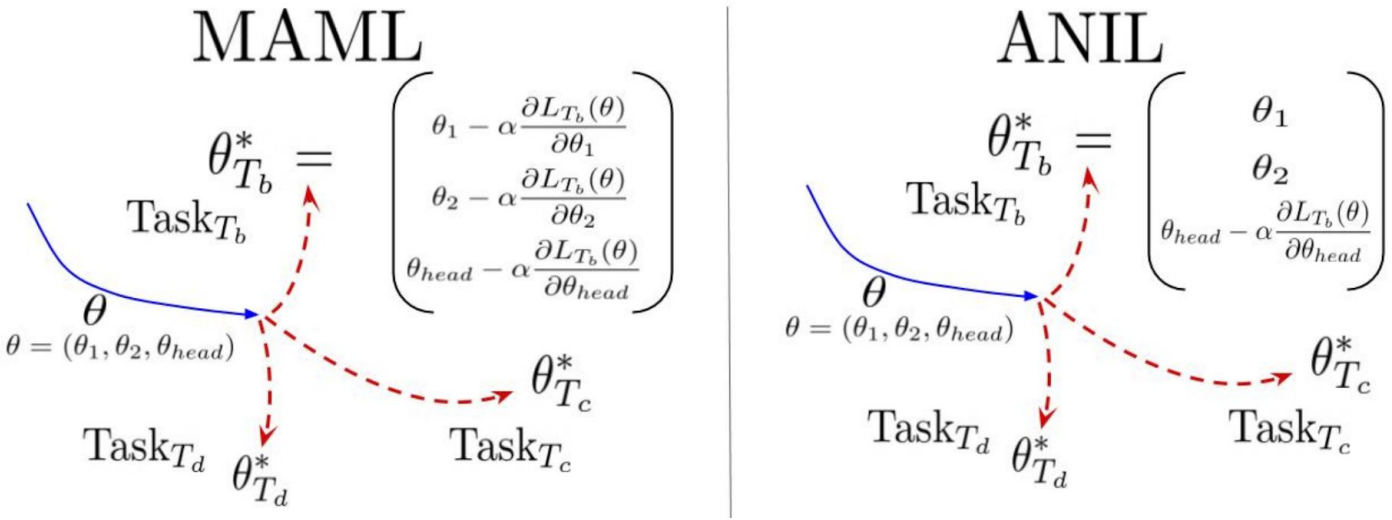


The inner loop mostly change the top layer.

? Any coincidence with vanishing gradient

ANIL

- Inner loop update:
  - Remove params update of the lower layers
  - Remain params update of the top layer
-



ANIL

Method	Omniglot-20way-1shot	Omniglot-20way-5shot	MiniImageNet-5way-1shot	MiniImageNet-5way-5shot
MAML	93.7 ± 0.7	96.4 ± 0.1	46.9 ± 0.2	63.1 ± 0.4
ANIL	96.2 ± 0.5	98.0 ± 0.3	46.7 ± 0.4	61.5 ± 0.5

Method	HalfCheetah-Direction	HalfCheetah-Velocity	2D-Navigation
MAML	170.4 ± 21.0	-139.0 ± 18.9	-20.3 ± 3.2
ANIL	363.2 ± 14.8	-120.9 ± 6.3	-20.1 ± 2.3

- The performance of MAML and ANIL are comparable
- Inner loop updates for lower layer is not necessary

NIL

- Top layer at inference
  - Train ANIL/MAML as usual
  - Testing: Replace the top layer by cosine similarity
- Conclusion:
  - With no task-specific head, no task specific adaptation, the model is comparable to MAML/NIL
  - The feature learned by MAML/ANIL is good enough

Method	Omniglot-20way-1shot	Omniglot-20way-5shot	MiniImageNet-5way-1shot	MiniImageNet-5way-5shot
MAML	93.7 ± 0.7	96.4 ± 0.1	46.9 ± 0.2	63.1 ± 0.4
ANIL	96.2 ± 0.5	98.0 ± 0.3	46.7 ± 0.4	61.5 ± 0.5
NIL	96.7 ± 0.3	98.0 ± 0.04	48.0 ± 0.7	62.2 ± 0.5

Reviewer comments

All these datasets are artificially created from the same dataset and hence it might be very easy to reuse features to get good performance.

I am not sure if the same analysis will hold if we consider a dataset where tasks are not this similar (like Meta-dataset, Triantafillou et al 2019)