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

Naresh Kumar Devulapally

CSE 701 SEM

March 15, 2023

Outline

Need for Few-Shot Learning

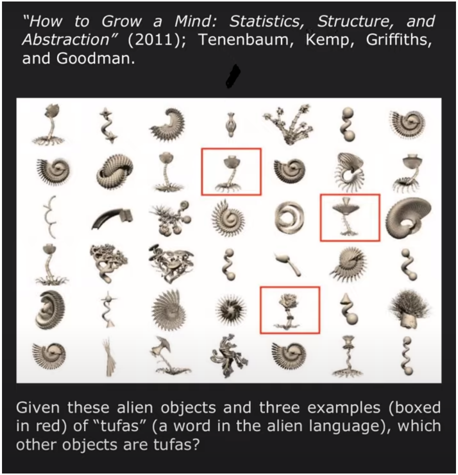

Figure: source

Need for Few-Shot Learning

"How to Grow a Mind: Statistics, Structure, and Abstraction" (2011); Tenenbaum, Kemp, Griffiths, and Goodman.

Given these alien objects and three examples (boxed in red) of "tufas" (a word in the alien language), which other objects are tufas?

Almost everyone selects just the objects boxed in gray.

Figure: source

Introduction to Few-Shot Learning

- Traditional supervised learning methods use large quantities of labeled data for training.
- Moreover, the test set comprises data samples that belong not only to the same categories as the training set but also must come from a similar statistical distribution. For example, a dataset created by images taken on a mobile phone is statistically different from that created by images taken on an advanced DSLR camera.
- This is popularly known as domain shift.

Introduction to Few-Shot Learning

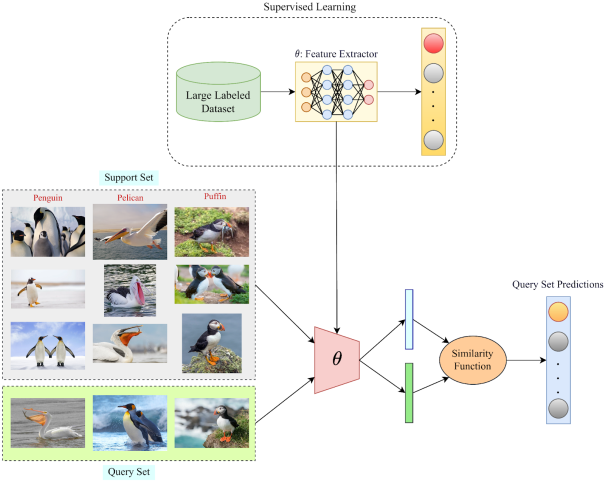

Figure: source and datasets

Methods to solve Few-Shot Learning

- Transfer Learning
- Metric Learning (Siasmese Networks)
- Prototypical Networks (Sort of clustering solution).
- Data Augmentation techniques (DAGAN).
- **Meta-Learning (Learning to Learn).**

Model-Agnostic Meta-Learning

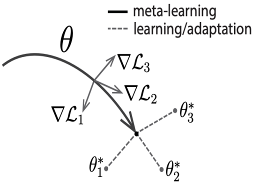

Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

Algorithm 2 MAML for Few-Shot Supervised 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: Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
6: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
7: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
8: Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
9: **end for**
10: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 2 or 3
11: **end while**

Source: MAML paper

Outer and Inner Loops in MAML

The MAML algorithm finds an initialization for a neural network so that new tasks can be learnt with very few examples (k examples from each class for k-shot learning) via two optimization loops:

- **Outer Loop:** Updates the initialization of the neural network parameters (often called the meta-initialization) to a setting that enables fast adaptation to new tasks.
- **Inner Loop:** Performs adaptation: takes the outer loop initialization, and, separately for each task, performs a few gradient updates over the k labelled examples (the support set) provided for adaptation.

Model-Agnostic Meta-Learning-2

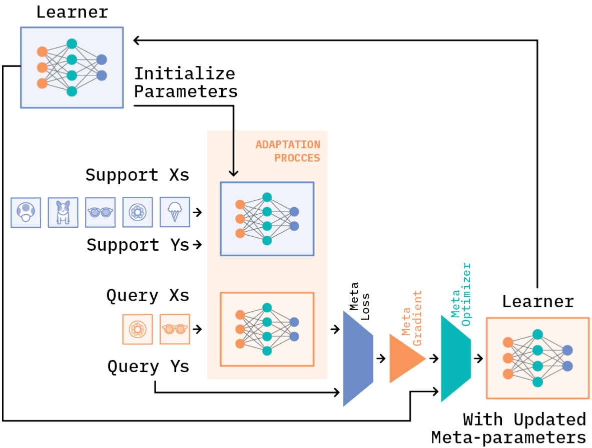

Figure: source

Math behind MAML

$$\theta_m^{(b)} = \theta_{m-1}^{(b)} - \alpha \nabla_{\theta_{m-1}^{(b)}} \mathcal{L}_{S_{T_b}}(f_{\theta_{m-1}^{(b)}(\theta)}) \tag{1}$$

for m fixed across all tasks, where $\mathcal{L}_{S_{T_b}}(f_{\theta_{m-1}^{(b)}(\theta)})$ is the loss on the support set of T_b after $m-1$ inner loop updates.

We then define the meta-loss as

$$\mathcal{L}_{meta}(\theta) = \sum_{b=1}^{B} \mathcal{L}_{Z_{T_b}}(f_{\theta_m^{(b)}(\theta)})$$

where $\mathcal{L}_{Z_{T_b}}(f_{\theta_m^{(b)}(\theta)})$ is the loss on the target set of T_b after m inner loop updates, making clear the dependence of $f_{\theta_m^{(b)}}$ on θ. The outer optimization loop then updates θ as

$$\theta = \theta - \eta \nabla_\theta \mathcal{L}_{meta}(\theta)$$

Figure: source

Rapid Learning or Feature Reuse?

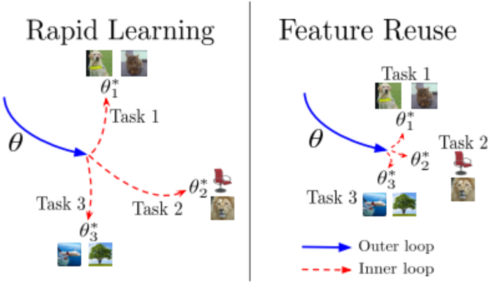

Figure: Rapid learning and feature reuse paradigms. In Rapid Learning, outer loop training leads to a parameter setting that is well-conditioned for fast learning, and inner loop updates result in significant task specialization. In Feature Reuse, the outer loop leads to parameter values corresponding to reusable features, from which the parameters do not move significantly in the inner loop

MAML vs ANIL

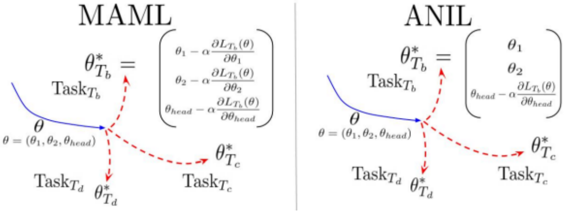

Figure: Schematic of MAML and ANIL algorithms. The difference between the MAML and ANIL algorithms: in MAML (left), the inner loop (task-specific) gradient updates are applied to all parameters θ, which are initialized with the meta-initialization from the outer loop. In ANIL (right), only the parameters corresponding to the network head θ_{head} are updated by the inner loop, during training and testing.

Results and Discussion

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

Table 2: **ANIL matches the performance of MAML on few-shot image classification and RL.** On benchmark few-shot classification tasks MAML and ANIL have comparable accuracy, and also comparable average return (the higher the better) on standard RL tasks (Finn et al., 2017).

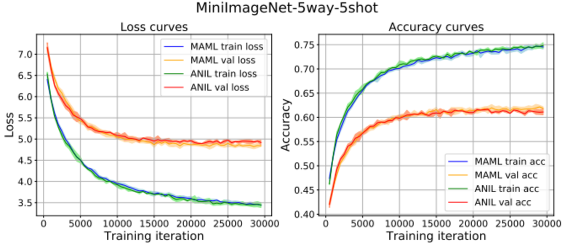

Figure 5: **MAML and ANIL learn very similarly**. Loss and accuracy curves for MAML and ANIL on MiniImageNet-5way-5shot, illustrating how MAML and ANIL behave similarly through the training process.

References

- Rapid Learning or Feature Reuse? ArXiV
- MAML Implementation
- Few Shot Learning blog
- MAML ArXiV