Mid-Term Evaluation Report: From Classical Machine Learning to Model-Agnostic Meta-Learning

Adhyatm Agnihotri
B.Tech-EE (Roll No: 240045)
Indian Institute of Technology, Kanpur
adhyatmag24@iitk.ac.in

Supervisor: Prof. Rohit Buddhiraja

Abstract

This mid-term report documents the progress of a Winter Project focused on Model-Agnostic Meta-Learning (MAML) and few-shot adaptation. The work intentionally builds upward: starting from core ideas in supervised learning (linear/logistic regression, SVMs, and optimization with stochastic gradient descent), moving to a concise deep-learning primer (backpropagation and CNN inductive bias), and finally transitioning into meta-learning. The central contribution so far is a structured theoretical review of MAML (Finn et al., 2017) supported by preliminary experiments on transfer learning using MNIST as a motivating bridge. We interpret MAML as learning an initialization that is not merely good*, but* rapidly improvable *with one or a few gradient steps. We also contrast this optimization-based view with metric-based approaches such as Siamese networks.*

1. Introduction

A persistent limitation of standard deep learning is data hunger: models often require large labeled datasets to generalize well. In contrast, humans routinely learn new concepts from a handful of examples. *Few-Shot Learning* (FSL) studies how to close this gap by exploiting prior experience across tasks.

This project explores one influential strategy: *meta-learning*, or "learning to learn." The motivating hypothesis is that rapid adaptation can be framed as an optimization problem: rather than training a model to solve one task well, train it so that it can *quickly* specialize to a new task after a small number of gradient updates.

Figure 1 summarizes the progression of topics covered so far: from classical ML foundations and optimization dynamics, to deep learning essentials, to transfer learning experiments, and finally to meta-learning (with MAML as the current focus).

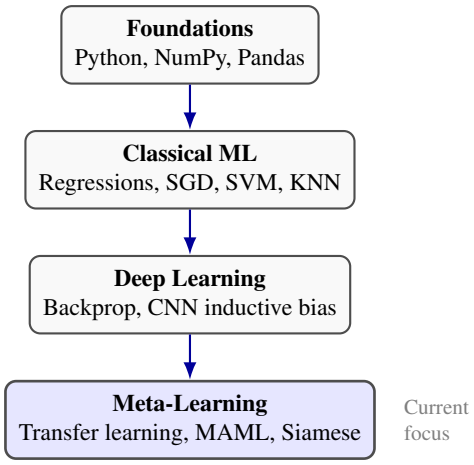

Figure 1. Roadmap of work completed so far, culminating in meta-learning and MAML.

What has been completed so far.

- Implemented core supervised learning models to build intuition about losses, gradients, and generalization.
- Studied SGD behavior (noise, stability, learning-rate sensitivity) and its relevance to deep networks.
- Reviewed CNN architectures as the standard backbone for vision-based few-shot benchmarks.
- Ran a small transfer-learning style experiment on MNIST to motivate the importance of initialization.
- Conducted a detailed reading of the MAML paper and organized its algorithmic and mathematical structure.

2. Foundations & Classical Machine Learning

Before meta-learning, it is important to be comfortable with the basics: what a model optimizes, how it generalizes, and how optimization choices change outcomes. This stage was less about "getting a high accuracy" and more about understanding *why* training behaves the way it does.

2.1. Optimization Dynamics (GD vs. SGD)

Implementations of linear and logistic regression were used as a controlled setting to study gradient-based optimization.

- **Batch vs. Stochastic updates:** Batch gradient descent yields smooth loss curves but can be slow; SGD introduces gradient noise that often helps exploration and speeds up iteration.
- **Learning rate sensitivity:** The learning rate η is the main stability knob. Large η can cause oscillation/divergence; small η can make training painfully slow.

A generic SGD update (single sample or mini-batch) for parameters w is:

$$w_{t+1} = w_t - \eta \nabla_w \mathcal{L}(w_t; \mathcal{B}_t), \tag{1}$$

where $\mathcal{B}_t$ is the sample/mini-batch used at iteration t.

2.2. Decision Boundaries and Kernels

KNN and SVMs were studied to connect geometry (decision boundaries) with generalization (bias–variance tradeoff). In particular, SVMs introduced a clean example of how feature mappings can make classification easier:

$$K(x_i, x_j) = \phi(x_i)^\top \phi(x_j), \tag{2}$$

where the kernel trick computes inner products in a (possibly very high-dimensional) feature space without explicitly evaluating $\phi(\cdot)$. Conceptually, this foreshadows deep learning: representation matters, and the right representation can simplify the downstream task.

3. Deep Learning Primer

Classical ML offered a clean view of optimization and geometry; deep learning adds scale and representation learning. Since standard few-shot benchmarks (e.g., Omniglot) are vision-based, this section focused on architectures commonly used in practice.

3.1. Architectures Studied

- **Multilayer Perceptrons (MLPs):** Used to understand backpropagation as repeated application of the chain rule.
- **Convolutional Neural Networks (CNNs):** Studied for their inductive biases (local connectivity, weight sharing) and why they are the default backbone for images.

A compact backprop view: for layer l with pre-activation z^l and activation $a^l = \sigma(z^l)$, the error term δ^l is

$$\delta^l = \frac{\partial \mathcal{L}}{\partial z^l} = \left(W^{l+1}\right)^\top \delta^{l+1} \odot \sigma'(z^l), \tag{3}$$

and gradients follow as $\frac{\partial \mathcal{L}}{\partial W^l} = \delta^l (a^{l-1})^\top$. This viewpoint becomes useful later because MAML differentiates *through* gradient updates.

4. Transfer Learning Experiments (Motivating Bridge)

A small transfer-learning experiment was performed on MNIST to build intuition around *initialization*. Although MNIST is not a few-shot benchmark in the strict sense, it provides a simple sandbox to observe the benefit of starting from a representation that already encodes useful structure.

Setup: Split MNIST into Task A (digits 0 vs. 1) and Task B (digits 2 vs. 3).

1. **Random Init:** Train on Task B from scratch.
2. **Transfer (A $\rightarrow$ B):** Pre-train on Task A, then fine-tune on Task B.
3. **Generalist:** Pre-train on all digits, then fine-tune on binary tasks.

Observation: The more general pre-training typically converged faster during fine-tuning, supporting the idea that *where you start* strongly affects how quickly you learn.

Limitation (important): Standard transfer learning optimizes for performance on the source task; it does not explicitly optimize for *adaptability*. This is precisely the gap MAML attempts to fill.

5. Model-Agnostic Meta-Learning (MAML)

This section is the core of the mid-term report and summarizes the main ideas from Finn et al. [1].

5.1. Problem Formulation

Let $p(\mathcal{T})$ be a distribution over tasks. Each task $\mathcal{T}_i$ provides:

- a **support set** $\mathcal{D}_S$ used for adaptation, and
- a **query set** $\mathcal{D}_Q$ used for evaluation.

The goal is to learn global parameters θ such that, after a small number of gradient steps on $\mathcal{D}_S$, the adapted parameters θ_i' perform well on $\mathcal{D}_Q$.

5.2. Intuition: "Good" vs. "Quickly Improves"

Transfer learning finds parameters that are already good at the source task. MAML instead learns an initialization that is *easy to fine-tune*: after one or a few gradient steps, the parameters move toward what that specific task needs.

Figure 2 provides a geometric visualization: MAML tries to place θ so that short gradient trajectories (inner-loop updates) can reach task-specific optima.

5.3. Algorithm (Bi-level Optimization)

MAML can be written as a nested optimization process: an *inner loop* adapts to each task, while an *outer loop* updates the shared initialization so that post-adaptation performance improves.

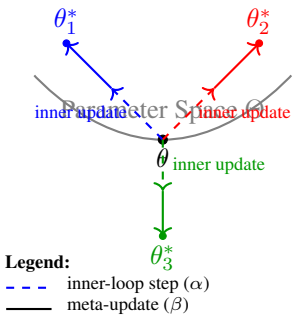

Figure 2. Conceptual view of MAML: learn an initialization θ that adapts quickly to task-specific optima θ_i^* via a small number of gradient steps.

Algorithm 1 Model-Agnostic Meta-Learning (MAML)

Require: Task distribution $p(\mathcal{T})$, step sizes α (inner), β (outer)

Initialize parameters θ
while not done **do**
  Sample tasks $\{\mathcal{T}_i\}_{i=1}^{B} \sim p(\mathcal{T})$
  for each task $\mathcal{T}_i$ **do**
    Sample support set $\mathcal{D}_S^{(i)}$ and query set $\mathcal{D}_Q^{(i)}$
    Compute adapted parameters (one inner step shown):
    $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta; \mathcal{D}_S^{(i)})$
  end for
  Meta-update using query losses:
  $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{i=1}^{B} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}; \mathcal{D}_Q^{(i)})$
end while

5.4. Mathematics of the Meta-Update (Why "Second-Order" Appears)

The meta-objective can be written as

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta_i'}; \mathcal{D}_Q^{(i)}\right), \quad \text{where } \theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta; \mathcal{D}_S^{(i)}). \tag{4}$$

Differentiating through $\theta_i'(\theta)$ yields

$$\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) = \nabla_{\theta_i'} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \cdot \left(I - \alpha \nabla_\theta^2 \mathcal{L}_{\mathcal{T}_i}(f_\theta)\right). \tag{5}$$

The Hessian term appears because the outer update must account for how the inner-loop gradient itself changes with θ. Practically, this is where computational cost and instability can enter for deep networks.

5.5. Canonical Example: Sinusoid Regression

A defining experiment in the original MAML paper is few-shot regression for $y = A\sin(x + \phi)$ with only $K = 5$ data points per task.

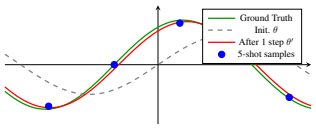

Figure 3. Conceptual sinusoid regression: the initialization captures useful structure, enabling fast adaptation with a few samples.

Optimization-based (MAML)	**Metric-based (Siamese)**
Learns an initialization θ optimized for fast fine-tuning.	Learns an embedding space for similarity-based inference.
Requires task-time adaptation (gradient steps).	Often no gradient updates at test time (nearest neighbor).
Model-agnostic (works across SL/RL with suitable loss).	Typically tailored to classification/retrieval settings.
Higher compute due to bi-level gradients (often second-order).	Inference can be lightweight once embeddings are computed.

Table 1. High-level comparison: MAML vs. Siamese-style metric learning.

- **Pre-trained baseline:** often collapses toward an "average" function and struggles to represent task-to-task variation in phase/amplitude.
- **MAML:** learns an initialization that already encodes the right *structure* (periodicity), so a small update can fit the specific A, ϕ quickly.

6. Metric-Based Few-Shot Methods: A Comparison

To avoid viewing meta-learning through a single lens, we also reviewed metric-based methods, where the goal is to learn an embedding space in which simple nearest-neighbor classification works.

6.1. Siamese Networks

Siamese networks learn an embedding $g(\cdot)$ such that examples of the same class are close and different classes are far. A common contrastive loss is

$$L = \sum \frac{1}{2}(1 - Y)D^2 + \frac{1}{2}Y\{\max(0, m - D)\}^2, \tag{6}$$

where $D = \|g(x_1) - g(x_2)\|_2$, $Y \in \{0, 1\}$ indicates whether the pair is different/same (depending on convention), and m is a margin.

7. Challenges Encountered and Practical Notes

7.1. Computational Cost

Full MAML involves differentiating through inner-loop updates, which can require second-order information (Hessian-vector products). For deep networks, this increases both memory and runtime.

A widely used approximation is **First-Order MAML (FOMAML)**, which ignores the second-order term and treats

$$\left(I - \alpha\nabla^2\right) \approx I. \tag{7}$$

This often provides a strong speed–accuracy tradeoff and is a natural baseline for reproduction work.

7.2. Training Instability

Meta-learning can be sensitive to:

- inner-loop learning rate choice (α),
- number of inner steps,
- task batch composition, and
- mismatch between support/query distributions.

A practical takeaway so far is that debugging meta-learning requires extra discipline: separate checks for inner-loop correctness, outer-loop gradients, and task sampling.

8. Conclusion

At mid-term, the project has established a clear conceptual pipeline: classical optimization $\rightarrow$ deep representation learning $\rightarrow$ the need for "adaptability-aware" training. The MAML framework provides a clean formalization of this idea via bi-level optimization, and preliminary transfer experiments further reinforce the importance of learning good initializations.

References

[1] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.