Mid-Term Project Evaluation

Model-Agnostic Meta-Learning (MAML)

Project Report

Rohith Rajeev Nambiar

January 9, 2026

Project Professor

Prof. Rohit Budhiraja

Project Mentors

Chaitanya Nitawe
Vashi Diwakar
Hrushikesh Roop Avvari

1 Introduction

Traditional machine learning systems are typically designed for fixed tasks and require large amounts of labeled data to perform effectively. While such approaches have achieved remarkable success across a wide range of applications, they often struggle in settings where data is scarce and rapid adaptation is required. In contrast, real world learning scenarios frequently demand the ability to generalize quickly to new tasks using only a small number of examples.

This limitation motivates the study of *meta-learning*, commonly described as "learning to learn," which focuses on training models that are explicitly optimized for adaptability rather than performance on a single task.

This project centers on the study of Model Agnostic Meta Learning (MAML), a gradient based meta learning framework that enables fast adaptation by learning parameter initializations well suited for Few Shot Learning(FSL).

The learning trajectory underlying this project follows a deliberate and structured progression:

- **Foundations of Machine Learning:** Learning is framed as an optimization problem, with classical supervised learning algorithms studied to develop intuition for loss functions, gradients, decision boundaries, and the bias–variance tradeoff.
- **Deep Learning:** The transition from shallow models to deep neural networks is explored, emphasizing hierarchical representation learning, neural network architectures, and backpropagation as the mechanism enabling end-to-end optimization.
- **Transfer Learning:** Transfer learning is introduced as a conceptual bridge to meta-learning, examining the role of initialization and pre-training through controlled experiments and highlighting the limitations of standard fine-tuning approaches.
- **Meta-Learning with MAML:** The trajectory culminates in an in-depth study of Model-Agnostic Meta-Learning, focusing on its task-based formulation, bi-level optimization structure, and gradient-based adaptation mechanism for few-shot learning.

2 Foundations of Machine Learning

Before engaging with meta learning frameworks, it was necessary to build a strong conceptual and practical foundation in classical machine learning. This stage of the project emphasized understanding the mechanisms underlying learning algorithms, rather than focusing exclusively on predictive performance. Developing this intuition was essential for appreciating how more advanced methods, such as meta-learning, extend classical ideas.

At a fundamental level, machine learning can be interpreted as an optimization driven process. Given a set of input and output pairs, the goal is to identify model parameters that minimize a predefined loss function measuring prediction error.

In this formulation, the model learns directly from data by adjusting its parameters, rather than relying on explicitly defined rules. Viewing learning as an optimization process provided a unified perspective across the different algorithms studied in the project.

Parameter updates are typically performed using gradient based optimization methods such as gradient descent and stochastic gradient descent (SGD). Through implementation and experimentation, it became clear that these methods exhibit different behaviors. While batch gradient descent offers stable and smooth convergence, stochastic variants introduce randomness that can help the optimization process escape poor local minima. This insight later became relevant when analyzing optimization dynamics within meta learning inner loops.

2.1 Supervised Learning Framework

Most of the methods examined in this project fall within the supervised learning setting, where models are trained using datasets that pair inputs with their correct outputs. Learning in this framework is driven by feedback from labeled data, allowing the model to gradually improve its predictions by correcting observed errors.

In this setting, a model acts as a mapping from inputs to outputs, producing predictions based on its current parameter values. The quality of

these predictions is evaluated using a loss function, which provides a numerical measure of how closely the model's outputs align with the ground truth. Training proceeds by iteratively adjusting the model parameters in a direction that reduces this loss, typically using gradient based optimization methods. Through repeated updates, the model refines its behavior, improving performance on the training data and ideally on unseen examples as well.

This learning framework applies broadly to both regression and classification problems. While the nature of the outputs and loss functions differs between these tasks, the underlying learning mechanism remains the same, reinforcing the view of supervised learning as a unified optimization-driven process rather than a collection of unrelated algorithms.

2.2 Regression and the Bias–Variance Tradeoff

Regression tasks were used to study how model complexity affects generalization. Experiments with linear and polynomial regression highlighted the bias–variance tradeoff, a recurring theme throughout the project. Simpler models often failed to capture underlying data structure, leading to underfitting, whereas overly flexible models adapted too closely to training data and performed poorly on unseen samples.

This tradeoff is particularly critical in Few Shot Learning(FSL) scenarios, where limited data worsens variance. Understanding this behavior provided valuable intuition for later sections, where models are expected to adapt rapidly without overfitting to small support sets.

2.3 Classification and Decision Boundaries

Classification algorithms further reinforced these ideas by illustrating how different learning strategies shape decision boundaries. Logistic regression introduced probabilistic classification through the sigmoid function and cross-entropy loss. Instance based methods such as K Nearest Neighbors (KNN) demonstrated how decision boundaries depend strongly on data distribution and hyperparameter selection, with small neighborhood sizes leading to highly irregular boundaries.

Support Vector Machines (SVMs) extended this analysis through margin maximization and kernel methods. The kernel trick illustrated how nonlinear decision boundaries in the original feature space can be represented as linear separators in higher-dimensional spaces, foreshadowing the representational capabilities of neural networks.

2.4 Data Preprocessing and Evaluation

Equally important to model design was the role of data preprocessing. Cleaning datasets, handling missing values, and applying feature scaling were essential steps prior to training. Algorithms such as KNN and SVM are particularly sensitive to feature magnitudes, making normalization a critical component of effective learning.

Model evaluation practices were also emphasized throughout this phase. Separating data into training and test sets ensured that performance metrics reflected genuine generalization rather than memorization.

The theoretical concepts discussed in this section were further reinforced through an assignment focused on implementing regression and classification algorithms using the `scikit-learn` library.

The assignment followed a standard machine learning workflow that included dataset exploration, train–test splitting, feature scaling, model training, and evaluation. Multiple algorithms were implemented, including linear and polynomial regression, logistic regression, K-Nearest Neighbors, decision trees, and Support Vector Machines, allowing for direct comparison of their behavior under varying conditions.

A key outcome of this exercise was the observation of the bias–variance tradeoff in practice. Simpler models, such as low-degree polynomial regressors or KNN models with large neighborhood sizes, consistently underfit the data, exhibiting poor performance on both training and test sets. In contrast, highly flexible models, including high-degree polynomials and KNN with very small neighborhood sizes, demonstrated overfitting by achieving low training error but poor generalization. Observing these effects in implementation provided a deeper and more intuitive understanding than theoretical analysis

alone.

The assignment also highlighted the impact of data preprocessing on algorithm performance. Feature scaling was found to significantly improve the performance of distance-based and margin-based methods such as KNN and SVM, while tree-based models remained largely unaffected. These observations emphasized that algorithmic performance is influenced not only by model choice, but also by preprocessing decisions and evaluation methodology, reinforcing the importance of a disciplined and systematic machine learning pipeline.

3 Deep Learning and Representation Learning

Deep learning emerged as a response to the limitations of classical machine learning models when applied to high-dimensional and complex data such as images, audio, and text. While traditional algorithms rely heavily on manually engineered features, deep neural networks learn representations directly from data, enabling models to capture increasingly abstract patterns through multiple layers of transformation.

A neural network is composed of multiple layers, where each layer applies a linear transformation to its input followed by a non-linear activation function.

The operation performed at layer l can be expressed as:

$$h^{(l)} = \sigma(W^{(l)}h^{(l-1)} + b^{(l)}),$$

where $W^{(l)}$ and $b^{(l)}$ are the learnable weights and biases, and σ denotes a non-linear activation function.

Stacking several such layers allows the network to learn hierarchical representations, with earlier layers capturing simpler patterns and deeper layers representing more abstract features.

Non-linear activation functions play a critical role in this process. Without non-linearities, stacking linear transformations would collapse into a single linear mapping, eliminating the benefit of depth. Activation functions

such as the sigmoid and Rectified Linear Unit (ReLU) introduce the non-linearity necessary for modeling complex relationships. In practice, ReLU has become a standard choice in deep networks due to its simplicity and favorable gradient propagation properties.

Training deep networks follows the same optimization principles introduced in classical machine learning. A loss function quantifies the discrepancy between predicted outputs and targets, and learning proceeds by minimizing this loss. The key algorithm enabling this process is backpropagation, which efficiently computes gradients of the loss with respect to all parameters by applying the chain rule through successive layers. These gradients are then used by optimization methods such as gradient descent or adaptive variants to update parameters.

An important conceptual insight emphasized during this transition is the continuity between classical models and neural networks. Logistic regression can be viewed as a single-layer neural network with a sigmoid activation and cross-entropy loss, while linear regression corresponds to a network without hidden layers and with squared error loss. From this perspective, deep learning does not replace classical machine learning, but rather generalizes it by introducing depth and learned feature hierarchies.

This connection becomes especially relevant when progressing toward meta-learning. Methods such as Model-Agnostic Meta-Learning (MAML) rely on the ability to differentiate through learning updates themselves, which is only possible due to the differentiable structure of neural networks and the backpropagation algorithm. Thus, understanding deep learning as an extension of optimization-driven learning provides the necessary foundation for appreciating gradient-based meta-learning frameworks.

4 Transfer Learning

To further examine the role of transfer learning as a precursor to meta-learning, an experiment was conducted using the MNIST digit classification dataset.

The problem was framed as a multi task setting, where knowledge learned

from one classification task was transferred to another related task. Models were trained under different initialization strategies, including training from scratch, pre training on a related source task followed by fine tuning on the target task, and more general pre training across multiple digit classes.

The experimental results demonstrated that initializing a model with pre trained representations significantly accelerated convergence on the target task compared to random initialization. However, a key limitation also became apparent.

While transfer learning improved initial performance, it mainly optimized features for the source task rather than for adaptability. In some cases, these transferred features were too specialized, reducing how efficiently the model could adapt to a new task. This limitation motivated the use of meta learning approaches such as Model Agnostic Meta Learning, which explicitly optimize models for rapid adaptation instead of source-task performance alone.

5 Model-Agnostic Meta-Learning (MAML)

Model-Agnostic Meta-Learning (MAML) is a gradient based meta learning framework designed to enable rapid adaptation to new tasks using only a small number of training examples. Unlike traditional machine learning approaches, which assume a fixed task and optimize performance on a single dataset, MAML operates over a distribution of tasks. Each task is defined by a small support set used for adaptation and a corresponding query set used for evaluation.

The central idea behind MAML is to learn model parameters that can be efficiently adapted to new tasks. Instead of learning task specific solutions, MAML optimizes an initialization that can be quickly fine tuned to new tasks through a few gradient updates. This formulation directly addresses the limitations observed in transfer learning, where representations optimized for source tasks may not adapt efficiently to new ones.

5.1 Task Based Learning and Bi-Level Optimization

MAML reframes learning as a task based optimization problem. During meta training, tasks are repeatedly sampled from a task distribution, and the model learns how to adapt efficiently across these tasks. This process is formalized through a bi-level optimization structure.

In the inner loop, the model performs task-specific adaptation by updating its parameters using the support set of a given task:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta),$$

where θ represents the shared initialization, α is the inner-loop learning rate, and $\mathcal{L}_{\mathcal{T}_i}$ denotes the loss for task $\mathcal{T}_i$.

In the outer loop, the adapted parameters are evaluated on the corresponding query sets, and the original parameters are updated to improve post-adaptation performance across tasks:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}),$$

where β is the meta-learning rate. This outer-loop update optimizes the initialization such that a small number of inner-loop gradient steps leads to strong generalization on new tasks.

This nested optimization structure differs from standard training procedures, as it explicitly optimizes model performance after adaptation rather than performance on a single task.

5.2 Second-Order and First-Order MAML

Because the adapted parameters θ_i' depend on the initial parameters θ, computing the meta-gradient involves second-order derivatives that capture how inner-loop updates influence outer-loop performance. While incorporating these second-order terms leads to more accurate optimization, it also increases computational cost.

To address this, First-Order MAML (FOMAML) approximates the meta-gradient by ignoring second order derivatives. This simplification reduces

computational overhead while retaining much of the performance benefit observed in full MAML.

These concepts were further reinforced through an assignment that required applying transfer learning and meta learning principles in practice. The assignment involved experimenting with different initialization strategies and observing how models adapted to new tasks under limited data conditions. By comparing training from scratch, pre-trained initialization, and rapid fine-tuning, the exercise highlighted the limitations of standard transfer learning and demonstrated the importance of learning parameters that are explicitly optimized for fast adaptation. Implementing and evaluating these approaches provided practical insight into the bi-level optimization structure of MAML and clarified how meta-learning differs from conventional training by focusing on post-adaptation performance rather than single-task optimization.

6 Conclusion and Future Work

At this stage of the project, a clear conceptual and practical foundation for gradient based meta learning has been established. Beginning from classical machine learning principles, the work progressed through deep learning and transfer learning, ultimately leading to a structured understanding of Model-Agnostic Meta-Learning (MAML). Rather than focusing on isolated algorithms, the project emphasized viewing learning as an optimization process, understanding the role of initialization, and recognizing how adaptability can be explicitly optimized across tasks.

The current status of the project reflects a strong level of preparedness for further exploration. Core machine learning models, loss functions, and optimization techniques have been implemented and analyzed, providing the necessary prerequisites for meta learning. Transfer learning experiments demonstrated the importance of initialization and highlighted the limitations of standard fine tuning approaches. Most importantly, the MAML framework has been studied and understood as a bi-level optimization problem that trades immediate task performance for rapid adaptability, rather than as a black box algorithm.

Looking ahead, the next phase of the project will focus on extending this understanding through deeper experimentation and refinement. This includes exploring additional task distributions, analyzing the effects of different inner loop update strategies, and further examining the tradeoffs between computational efficiency and adaptation performance. These directions aim to build on the current foundation and move toward a more comprehensive evaluation of meta learning methods in few shot settings.

References

[1] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.