Few-Shot Learning on Non-Linear Decision Boundaries: A Comparative Study of MAML vs. Joint Training

Utkarsh Singh

February 8, 2026

Abstract

This report investigates the efficacy of Model-Agnostic Meta-Learning (MAML) for few-shot adaptation on a synthetic 2D "Moving Circle" dataset. We compare MAML against a standard baseline trained via joint optimization (Joint Training). The task requires a neural network to learn a circular decision boundary from only $K = 10$ labeled examples within a single gradient update. Our experiments demonstrate that while the baseline model learns an aggregate representation of the task distribution, it fails to adapt rapidly to specific instances. In contrast, MAML explicitly optimizes for sensitivity to new data, allowing it to recover the correct decision boundary with minimal data and computation.

1 Introduction

Deep learning models typically require large datasets to generalize effectively. However, in many real-world scenarios, data is scarce, and models must adapt to new tasks rapidly—a paradigm known as Few-Shot Learning. This assignment explores this challenge using a synthetic "Moving Circle" dataset, where the underlying concept (a circle of fixed radius) remains constant, but the location (center) shifts between tasks.

We implement and compare two approaches:

1. **Model-Agnostic Meta-Learning (MAML):** An optimization-based meta-learning algorithm that learns an initialization θ capable of fast adaptation [1].
2. **Joint Training (Baseline):** A standard supervised learning approach that trains a single model on data mixed from all tasks, followed by fine-tuning at test time.

2 Problem Formulation

2.1 The "Moving Circle" Dataset

We define a distribution of tasks $p(\mathcal{T})$. Each task $\mathcal{T}_i$ corresponds to a binary classification problem characterized by a circle with a fixed radius $r = 2.0$ and a random center $\mathbf{c}_i = (c_x, c_y)$.

The center is sampled uniformly:

$$ithc_x, c_y \sim \mathcal{U}[-3, 3] \tag{1}$$

The input space is $\mathbf{x} \in \mathbb{R}^2$ with $x_1, x_2 \in [-5, 5]$. The ground truth label y for a point $\mathbf{x}$ is given by the indicator function:

$$y = \mathbb{I}\left(\sqrt{(x_1 - c_x)^2 + (x_2 - c_y)^2} < r\right) \tag{2}$$

For each task $\mathcal{T}_i$, we are given a support set $\mathcal{D}_{\text{sup}}$ containing $K = 10$ labeled examples. The goal is to find model parameters ϕ that minimize the loss on a query set $\mathcal{D}_{\text{qry}}$ after seeing $\mathcal{D}_{\text{sup}}$.

3 Methodology

3.1 Model Architecture

Both the MAML and Baseline approaches utilize the same Multi-Layer Perceptron (MLP) architecture to ensure a fair comparison:

- **Input:** 2 units (x_1, x_2)
- **Hidden Layers:** Two layers of 64 units with ReLU activation.
- **Output:** 1 unit with Sigmoid activation (probability of class 1).
- **Loss Function:** Binary Cross-Entropy (BCE).

3.2 Algorithm 1: Model-Agnostic Meta-Learning (MAML)

MAML aims to find an initial set of parameters θ such that one step of gradient descent on a specific task $\mathcal{T}_i$ results in significant performance improvement.

3.2.1 Inner Loop (Adaptation)

For a sampled task $\mathcal{T}_i$, we compute the adapted parameters θ_i' using the support set $\mathcal{D}_{\text{sup}}^{(i)}$:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta; \mathcal{D}_{\text{sup}}^{(i)}) \tag{3}$$

where α is the inner learning rate. In our experiments, we perform exactly **1 gradient step**.

3.2.2 Outer Loop (Meta-Optimization)

The meta-objective minimizes the loss of the *adapted* parameters θ_i' on the query set $\mathcal{D}_{\text{qry}}^{(i)}$:

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}; \mathcal{D}_{\text{qry}}^{(i)}) \tag{4}$$

The meta-update for the initialization parameters θ is performed using stochastic gradient descent (SGD):

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}; \mathcal{D}_{\text{qry}}^{(i)}) \tag{5}$$

where β is the meta-learning rate. Crucially, this requires computing second-order derivatives (Hessian-vector products) since θ_i' is itself a function of θ.

3.3 Algorithm 2: Baseline (Joint Training)

The baseline approach treats the problem as standard supervised learning. We train a single network f_ϕ to minimize the expected loss across all tasks simultaneously.

3.3.1 Pre-Training

We sample batches of tasks and mix their data, effectively training the model to learn the "average" decision boundary across the distribution $p(\mathcal{T})$:

$$\phi^* = \arg\min_\phi \mathbb{E}_{\mathcal{T} \sim p(\mathcal{T})} \left[\mathcal{L}_{\mathcal{T}}(f_\phi) \right] \tag{6}$$

Since the centers are uniformly distributed in $[-3, 3]$, the optimal pre-trained model ϕ^* likely learns a diffuse probability distribution centered at $(0, 0)$.

3.3.2 Test-Time Fine-Tuning

At test time, given a new task $\mathcal{T}_{\text{test}}$, we initialize the network with ϕ^* and perform standard gradient descent using the support set:

$$\phi_{\text{new}} = \phi^* - \alpha \nabla_\phi \mathcal{L}_{\mathcal{T}_{\text{test}}}(f_{\phi^*}; \mathcal{D}_{\text{sup}}) \tag{7}$$

4 Experimental Setup

- **Outer Epochs:** 2000
- **Meta Batch Size:** 4 tasks
- **Inner/Fine-tuning LR (α):** 0.1
- **Outer LR (β):** 0.001
- **Shots (K):** 10 examples

5 Results and Discussion

5.1 Quantitative Evaluation: Adaptation Speed

We evaluated both models on a held-out test task by performing 10 steps of gradient descent (fine-tuning) and measuring the accuracy on a query set at each step.

- **Step 0 (Initialization):** Both models start with low accuracy ($\approx 50 - 60\%$). The Baseline performs slightly better initially as it learns the "average" circle, whereas MAML's initialization is not optimized for direct performance but for *adaptability*.
- **Step 1 (The Jump):** MAML exhibits a massive spike in accuracy (typically $> 90\%$) after a single gradient update. This validates that MAML successfully learned an initialization on the manifold of task parameters.
- **Baseline Performance:** The Baseline improves slowly. Since its weights were optimized for the average case, the gradients from just 10 examples are insufficient to drastically reshape the decision boundary to the specific new location immediately.

5.2 Qualitative Evaluation: Decision Boundaries

We visualize the predicted decision boundaries after exactly **one gradient step**.

- **MAML:** The model "hallucinates" a circular boundary that fits the support points well. It has effectively learned the geometric concept of a circle and only needs the support points to determine the center.
- **Baseline:** The decision boundary appears unstructured or linear in the locality of the support points. Lacking the meta-learned prior, it attempts to fit the 10 points using generic features, resulting in a poor approximation of the circle.

Bonus Question

Question: Why does the MAML meta-training loss curve often appear noisy or random compared to the smooth decreasing curve of standard training?

Answer: The instability and high variance (noise) observed in MAML training curves compared to standard supervised learning are inherent to the meta-learning objective. The three primary mathematical and structural reasons for this phenomenon are:

1. **The "Double Sampling" Stochasticity**
   In standard training, variance arises from sampling a batch of data points $x \sim \mathcal{D}$. In MAML, we introduce two layers of stochasticity:

   - First, we sample a batch of Tasks $\mathcal{T}_i \sim p(\mathcal{T})$.
   - Second, for each task, we sample a Support Set ($K = 10$) and a Query Set.

   Because K is very small (10-shot), the "difficulty" of the batch fluctuates wildly.

   - *Scenario A:* The 10 support points are perfectly spaced around the circle. The inner gradient update is effective, resulting in low query loss.
   - *Scenario B:* The 10 support points are clustered in one corner or are ambiguous. The inner update is poor, resulting in high query loss.

   This creates a high-variance signal that standard SGD averaging cannot smooth out as easily as it does in large-batch supervised learning.

2. **Sensitivity Optimization (The "Edge of Chaos")**
   The explicit goal of MAML is to find a set of parameters θ that are highly sensitive to new data. Mathematically, MAML relies on the term $\nabla_\theta \mathcal{L}_{\text{inner}}$ to drive adaptation. It requires the weights to change significantly in response to a small gradient.

   Consequence: A small update to the meta-parameters θ can drastically change the trajectory of the inner loop θ_i'. This implies that the meta-loss landscape is extremely sharp and rugged, leading to "spiky" loss curves where the model jumps in and out of basins of attraction.

3. **Second-Order Instability**
   MAML optimization involves the Hessian (second derivative matrix) of the loss function. The meta-gradient is computed as:

   $$\nabla_\theta \mathcal{L}_{\text{meta}} = \nabla_{\theta'} \mathcal{L}_{\text{outer}} \cdot \left(I - \alpha \nabla_\theta^2 \mathcal{L}_{\text{inner}}\right)$$

   Estimating curvature (the Hessian ∇_θ^2) is notoriously noisy, especially with small batch sizes. The "noise" observed in the curve often reflects the optimizer's struggle to navigate this complex curvature without the stability typically found in first-order supervised learning.

6 Conclusion

This study confirms that for the "Moving Circle" dataset, MAML significantly outperforms standard Joint Training in the few-shot regime. By explicitly optimizing the model's sensitivity to gradient updates, MAML acquires a strong inductive bias (the circular shape) that allows it to solve the task with minimal data. In contrast, standard pre-training learns a global average that is robust but lacks the plasticity required for rapid adaptation.

References

[1] Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *International Conference on Machine Learning (ICML)*.