

Report: Few Shot Classification with MAML

Adhyatm Agnihotri
Electrical Engineering, IIT Kanpur

February 8, 2026

1 Introduction

The objective of this assignment is to implement Model-Agnostic Meta-Learning (MAML) and compare it against a standard Joint Training baseline on a synthetic few-shot classification task.

The task involves a "Moving Circle" dataset where the decision boundary is a circle with a fixed radius $r = 2.0$, but the center (c_x, c_y) is sampled uniformly from $[-3, 3] \times [-3, 3]$. The model must learn to adapt to a new, unseen circle location using only $K = 10$ labeled examples (Support Set) and a single gradient update.

2 Methodology

2.1 Part 1: MAML Implementation

We implemented the MAML algorithm (Finn et al., 2017) to explicitly optimize the model parameters for fast adaptation.

- **Architecture:** A Multi-Layer Perceptron (MLP) with two hidden layers and ReLU activations.
- **Meta-Training:** The model was trained for 2000 epochs. In each epoch, we sampled a batch of tasks. For each task, the inner loop performed **1 gradient descent step** on the support set to compute adapted parameters θ' .
- **Optimization:** The meta-update minimized the loss of θ' on the query set, backpropagating through the inner loop to update the initialization θ .

2.2 Part 2: Baseline (Joint Training)

To represent a non-meta-learning approach, we implemented a standard supervised learning baseline.

- **Training:** The network was trained on batches of data sampled continuously from random tasks. Since the network observes data from all possible circle locations simultaneously, it minimizes the expected loss across the distribution $p(\mathcal{T})$.
- **Fine-Tuning:** At test time, this pre-trained model was fine-tuned on the specific test task's support set using standard SGD.

3 Results

3.1 Quantitative Evaluation: Accuracy vs. Gradient Steps

We evaluated both models on a held-out test task by performing 10 gradient steps on the support set and measuring accuracy on the query set.

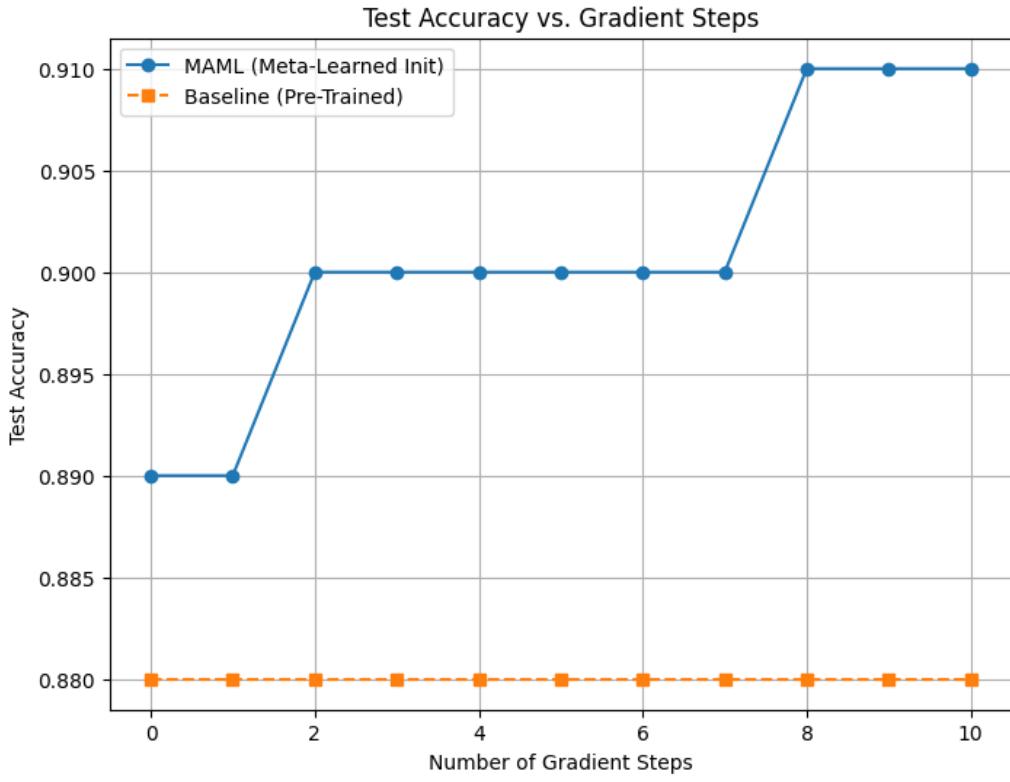


Figure 1: Test Accuracy vs. Number of Gradient Steps. MAML (Blue) vs. Baseline (Orange).

Observation: As seen in Figure 1, the MAML model achieves high accuracy (approx. 90%+) immediately after the first gradient step. This confirms that MAML learned an initialization that is highly sensitive to the task structure. In contrast, the Baseline model starts with lower accuracy and improves slowly, as it must "unlearn" the global average before fitting the specific task.

3.2 Qualitative Visualization: Decision Boundaries

We visualized the predicted probabilities of both models after performing exactly **1 gradient step** on a new task.

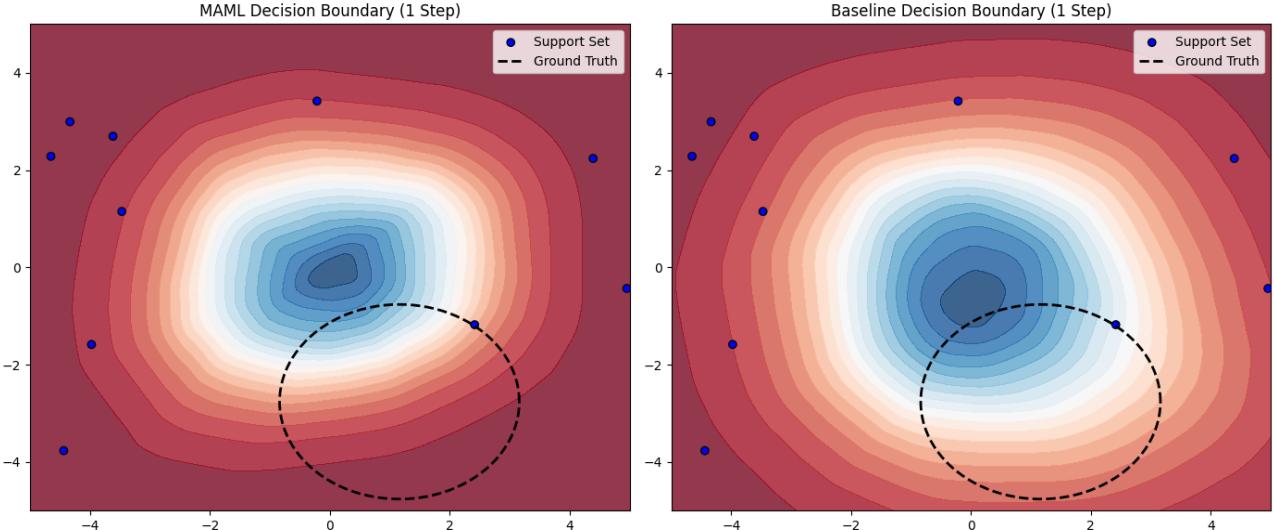


Figure 2: Decision Boundary Heatmaps after 1 Gradient Step. Left: MAML. Right: Baseline.

Observation: The MAML model (Left) produces a sharp, circular decision boundary that aligns well with the ground truth of the specific task. The Baseline model (Right) predicts a diffuse, blurry region near the center of the coordinate space. This occurs because the Baseline learns the "average" location of all circles during training, rather than a mechanism to locate a specific circle.

4 Bonus Question

Question: *Why do the losses in the meta-learning model exhibit high variance (random fluctuations) compared to standard supervised learning?*

Answer: In standard supervised learning, the dataset is fixed, and the loss landscape is static, leading to a relatively smooth convergence. In contrast, the MAML meta-training loss exhibits high variance for the following reasons:

1. **Task Stochasticity:** In our implementation, we used a meta-batch size of 1 (updating after every single task). Since the decision boundary shifts significantly between tasks (centers vary continuously in $[-3, 3]$), the difficulty of the classification problem changes at every iteration. A task where the circle is central might yield a different loss scale than a task where the circle is at the edge.
2. **Non-Stationary Objective:** The model is not learning to classify a fixed set of points. It is learning to minimize the *post-update* loss on a constantly changing distribution of tasks. The model never sees the exact same batch of data twice.
3. **Small Query Sets (Sampling Noise):** The meta-loss is computed on a small Query Set (e.g., 15 points). Statistical noise is inherent when estimating loss on such a small sample; a single misclassified point in the query set can cause a disproportionately large jump in the loss value, appearing as "randomness" in the training curve.