AML A2 - Few-Shot Learning

Irina-Mona Epure

1 Introduction

A problem that is often encountered when training neural networks is having very little labeled data available for the training process. Two popular solutions for this issue are transfer learning and meta-learning. In this assignment, I experimented with different techniques for solving the sine wave regression problem proposed by [1] with very limited data available. I used pretraining and finetuning, MAML [1], and Reptile [2] to obtain good initial parameters before training on the new task. Meta-learning sometimes outperforms transfer learning, which is a simpler strategy. However, depending on the tasks, the opposite may occur. It is therefore difficult to know what to expect from this experiment.

2 Background

Pretraining and **finetuning** are the two main steps in transfer learning. During pretraining, a neural network is trained on a task with sufficient data to achieve good results. After that, the same neural network can be finetuned for a different, but similar task, for which limited data is available. Generally the weights of the neural network are frozen after pretraining, and only the weights in the final Fully Connected layer are updated during finetuning.Transfer learning is just a way to initialize a neural network to give it a head start for a new task.
In contrast, meta-learning sets out explicitly to to train the initial neural network so that it is easily generalizable. That means finding a set of weights that we can train very quickly with Stochastic Gradient Descent on a new similar task. During training, **MAML** copies the initial weights of the network, takes a training task and backpropagates the loss on its query set to update the initial weights. **Reptile**, on the other hand, performs Stochastic Gradient Descent after training on several tasks. Reptile is a first-order algorithm and is thus, more efficient than MAML, but there is often not a big difference in performance between the two. Because both of these meta-learning techniques use backpropagation, they are slower than transfer learning.

3 Experiments

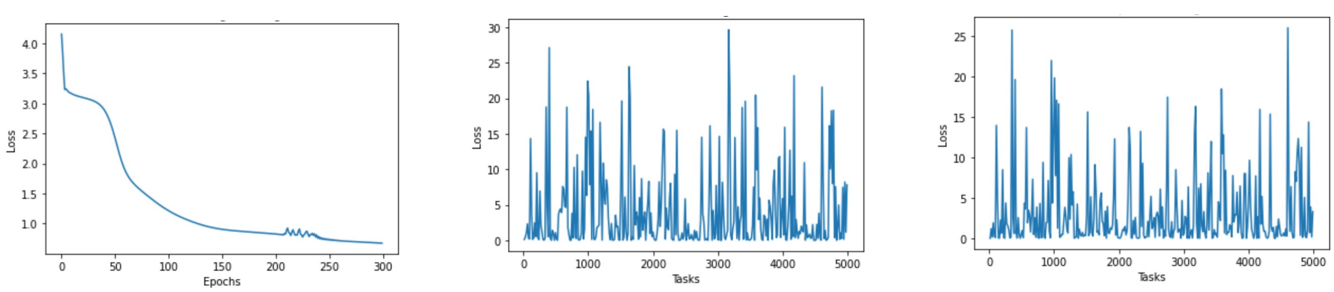
The entirety of the code used for the implementation of this experiment is linked in the footnotes [1].
For the experiments, I used the provided SineLoader class to obtain the following:

- a non-episodic training data loader for pretraining during transfer learning;

[1] https://drive.google.com/drive/folders/1mlZ6Cuq7IJxIgNfmos2ALWFbRtVUAvBu?usp=sharing

- an episodic training data loader for obtaining the initial weights for Reptile and MAML;
- an episodic validation data loader and another testing one to be used for all three strategies.

For transfer learning, a neural network was trained for 300 epochs on a batch obtained from the non-episodic train loader. It was optimized using Stochastic Gradient Descent. After 300 epochs, the network converged to have the training loss of 0.676.
For meta-learning, training was done on 5000 tasks referred to as episodes, which had a support set and a query set. Stochastic Gradient Descent was used internally by both algorithms, but for Reptile weight update was performed only at the end of batches containing 16 episodes. The hyperparameters used are: $\beta = 0.1$ and $\nabla_\theta = 0.1$ for MAML, and $\epsilon = 0.5$ for Reptile. The training performance of the three techniques was tracked with the help of the plots in Figure **??**. It is very obvious from this figure that the network pretrained for transfer learning converged, while the others did not.



(a) Pretraining (b) MAML (c) Reptile

Figure 1: Performance of the three methods throughout training.

For validation and testing, the three networks obtained during training were trained again on the support sets and evaluated on the validation sets of the episodes obtained from the validation and test loaders. The results are reproduced in Table 1. Reptile achieved the lowest loss, while finetuning performed slightly better than MAML. The time needed for pretraining the networks differs a lot. Transfer learning was very fast at 2.176 seconds, while MAML and Reptile took a much longer time: 48.564 seconds, and 85.189 seconds, respectively.

Table 1: Performance of transfer learning, MAML, and Reptile.

Algorithm	**Training time**	**Average Validation Loss**	**Average Test Loss**
Finetuning	2.178 seconds	4.295	4.301
MAML	48.564 seconds	4.499	4.482
Reptile	85.189 seconds	3.204	3.210

4 Discussion and conclusions

Among the three techniques included in these experiments, Reptile achieves the lowest loss for the sine wave regression problem, but also requires the most time. It is followed by transfer learning,

which is by far the fastest method, and then closely by MAML, with a slightly bigger loss, but once again, a lot of time required for finding good initialization parameters. It appears that, indeed, it cannot be stated with certainty that meta-learning is better than transfer learning in all cases or vice versa.
However, there does appear to be a trade-off involved between training time and accuracy, which is shown by the two best alternatives analysed: Reptile and finetuning.

References

[1] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[2] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.