Reptile: a Scalable Meta-learning Algorithm

Alex Nichol and John Schulman

2018

1 What

A meta-learning algorithm which can quickly learn from previously unseen task sampled from the same distribution as training tasks belong to. It is easy to implement and does not require differentiating through the optimisation process.

2 Why

Humans do not learn from scratch, but we want the machines to do that. What if we just pre-train the model to adapt to new tasks quickly? Meta-learning is the field that is interested in this.

One approach of meta-learning is to encode the learning algorithm in the weights of an RNN, but do not perform gradient descent at test time [Hochreiter et al., 2001, Santoro et al., 2016]. A second approach is to learn the model initialisation and then fine-tune it at test time on the new task [Deng et al., 2009, Finn et al., 2017].

The good thing about MAML is that it uses a gradient-based learning algorithm which can generalise better even when getting an out-of-sample data. On the other hand, it's not the best choice for problems which require a lot of grad steps at test time, since MAML needs to differentiate through the optimisation process. (*Don't get this one. Is it about computational complexity?*)

3 How

Algorithm 1: Reptile, serial version

Initialise ϕ, initial params of the model;
for $iteration = 1$ **to** n_{epochs} **do**
    Sample task τ with loss L_τ;
    $W \leftarrow \text{SGD}(L_\tau, \phi, k)$, where k is the number of SGD steps;
    Do the update $\phi \leftarrow \phi + \epsilon(W - \phi)$;
end

If you want, you can also use Adam or another algorithm to do the grad update for ϕ.

$$\mathbb{E}_\tau[SGD(L_\tau, \phi, k)] \stackrel{?}{=} SGD(\mathbb{E}_\tau[L_\tau], \phi, k) \quad (1)$$

Equation 1 holds only when $k = 1$. This explains why Reptile converges to a solution that's very different from the minimizer of $\mathbb{E}_\tau L_\tau$.

Algorithm 2: Reptile, batched version

Initialise ϕ, initial params of the model;
Sample tasks $\tau_1, \tau_2, ..., \tau_n$;
for *iteration*$\leftarrow 1$ **to** n_{epochs} **do**
  for $i \leftarrow 1$ **to** *batch_size* **do**
    $W \leftarrow \text{SGD}(L_{\tau_i}, \phi, k)$, where k is the number of SGD steps;
  end
  Do the update $\phi \leftarrow \phi + \frac{\epsilon}{k} \sum_{i=1}^{n} (W_i - \phi)$;
end

I really like the fact, that apart from just showing the plots, the authors take a step forward and analyse behaviour of the algorithm in the **Analysis** section. First, they factor and compare the grads of MAML, First-Order MAML and Reptile. Second, they show (informally) that Reptile converges to a solution close (in Euclidean distance) to each task manifold of optimal solutions.

Apart from that, the authors experiment with different combinations of inner loop gradients. The conclusions of these experiments are:

- the more mini-batches is better
- more inner loop iterations also help

4 Evaluation

4.1 Sine Wave Regression as in [Finn et al., 2017]

The task here is to fit a sine wave $f_\tau = a \sin(x + b)$. Amplitude $a \sim U([0.1, 5.0])$ and phase $b \sim U([0, 2\pi])$ define the task. They sample p points $x_1, x_2, ..., x_p \sim U([-5, 5])$ and try to fit the curve by minimising the loss $L_\tau(f) = \int_{-5}^{5} \|f(x) - f_\tau(x)\|_2^2 \, dx$

MAML and Reptile behave similarly on this toy-task.

4.2 Few-shot classification on Omniglot and Mini-ImageNet

Lot's of tables with lot's of figures to show that Reptile is pretty close to MAML (still a bit worse in terms of performance).

5 Comments

- Weird, that they show no comparison with other methods when $k = 1$.
- Weird, that their k parameter is different for each of the experiments. They do not compare different behaviour using the same k.

References

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

[Finn et al., 2017] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*.

[Hochreiter et al., 2001] Hochreiter, S., Younger, A. S., and Conwell, P. R. (2001). Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer.

[Santoro et al., 2016] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850.