Meta-Learning by the Baldwin Effect

Chrisantha Fernando, Jakub Sygnowski, Simon Osindero, Jane Wang, Tom Schaul, Denis Teplyashin, Pablo Sprechmann, Alexander Pritzel, Andrei Rusu

Google DeepMind

London, UK

chrisantha@google.com

ABSTRACT

We show that the Baldwin effect is capable of evolving few-shot supervised and reinforcement learning mechanisms, by shaping the hyperparameters and the initial parameters of deep learning algorithms. This method rivals a recent meta-learning algorithm called MAML "Model Agnostic Meta-Learning," which uses second-order gradients instead of evolution to learn a set of reference parameters that can allow rapid adaptation to tasks sampled from a distribution. The Baldwin effect does not require gradients to be backpropagated to the reference parameters or hyperparameters, and permits effectively any number of gradient updates in the inner loop, learning strong learning dependent biases.

ACM Reference Format:

Chrisantha Fernando, Jakub Sygnowski, Simon Osindero, Jane Wang, Tom Schaul, Denis Teplyashin, Pablo Sprechmann, Alexander Pritzel, Andrei Rusu. 2018. Meta-Learning by the Baldwin Effect. In *Proceedings of the Genetic and Evolutionary Computation Conference 2018 (GECCO '18)*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205763

1 INTRODUCTION

Of several approaches to learning to learn, the model-agnostic meta-learning (MAML) algorithm [2] is powerful but requires a differentiable learning procedure. Here we show that the Baldwin effect is able to achieve similar results.

As in earlier work we evolve inductive bias in the form of the initial parameters P and hyperparameters h of a learning algorithm [1, 6] but in addition provide a way to evolve agents for few-shot data-efficient learning on a task distribution. First we show that the Baldwin effect and MAML are comparable on a supervised learning task. Secondly we demonstrate that the Baldwin effect can be used in cases where MAML cannot be used, for instance in cases where the genotype is non-differentiable, e.g. where we evolve the macro-actions used by a discrete action RL algorithm, or the algorithms' discrete hyperparameters themselves. Thirdly we examine how genetic accommodation takes place in real deep neural networks undergoing the Baldwin effect.

2 ALGORITHMS

Model-Agnostic Meta-Learning can be summarized as learning (using gradient descent) a set of reference parameters θ^* such that a small number of gradient descent steps using a small amount of data leads to effective generalization on a task.

Algorithm 1 Baldwinian Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: $\mathcal{P}$: initial population-representation of individuals
Require: $\mathcal{S}$: procedure to obtain a batch of individuals (i.e. parameters and hyper-parameters) given a population-representation
Require: $\mathcal{U}$: procedure to update a population-representation given a batch of fitness-scored individuals
Require: $\mathcal{F}$: fitness scoring function
Require: N: number of gradient steps to take during per-task gradient training

1: **while** not done **do**
2: Generate batch of individuals from population: $\theta^{g,\emptyset}, \alpha^g \sim \mathcal{S}(\mathcal{P})$
3: **for all** $\theta^{g,\emptyset}$ **do**
4: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
5: **for all** $\mathcal{T}_i$ **do**
6: $\theta^{g,i} \leftarrow \theta^{g,\emptyset}$
7: **for** `k=1...N` **do**
8: Evaluate $\nabla_{\theta^{g,i}} \mathcal{L}_{\mathcal{T}_i}(\theta^{g,i})$
9: Update adapted parameters with gradient descent: $\theta^{g,i} \leftarrow \theta^{g,i} - \alpha^g \nabla_{\theta^{g,i}} \mathcal{L}_{\mathcal{T}_i}(\theta^g, i)$
10: **end for**
11: Compute fitness-score for current task: $f_i^g = \mathcal{F}(\theta^{g,i})$
12: **end for**
13: Compute overall fitness estimate: $f^g = \sum_i f_i^g$
14: **end for**
15: Update population based on fitness of individuals: $\mathcal{P} \leftarrow \mathcal{U}(\mathcal{P}, \{(1, \theta^1, \alpha^1, f^1), ..., (g, \theta^g, \alpha^g, f^g)\})$
16: **end while**

Here, we use the Baldwin effect to find such a set of parameters (and hyperparameters) via evolution in a similar, meta-learning setup. The pseudocode, shown in Algorithm 1, can be further modified to adapt to the continual learning setting with the use of Lamarckian inheritance (e.g. changing line 15 to update the population using the previous parameters) or to use Darwinian evolution by disabling gradient descent-based learning altogether.

3 TASKS

3.1 Sinusoid fitting

We compare the performance of MAML and the Baldwin effect in the form of Natural Evolution Strategies [7] and Generational Genetic Algorithm [3] on the task of fitting sinusoids. We follow the same testing procedure and use the same network architectures

GECCO '18, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205763

as described in [2]. The results, presented in Figure 1a, show that we achieve performance comparable to MAML.

3.2 Reinforcement learning tasks

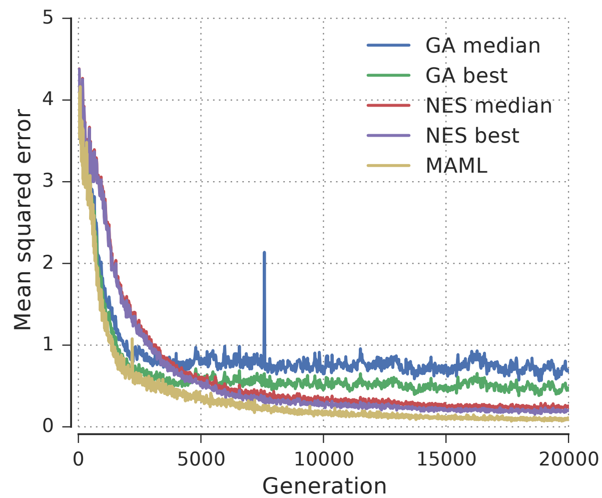
We use the Planar Cheetah model from a physics simulator MuJoCo [5] to define two reinforcement learning tasks to test the Baldwin effect:

- Goal Velocity, where the reward is a negative absolute value between the current velocity of the agent and a target velocity,
- Goal Direction, where a target direction (backward or forward) is chosen in an alternating manner from episode to episode, and the reward is the magnitude of the velocity in that direction.

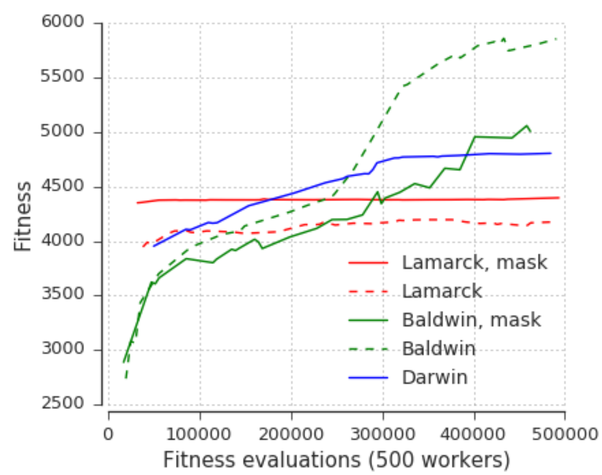
Goal Velocity. We observed that Lamarckian evolution outperformed Baldwinian evolution, which in turn outperformed Darwinian evolution.

Goal Direction. Baldwinian evolution evolved a model capable of quickly adapting its direction to the target direction within a single episode lasting only 30 simulated seconds. Figure 1b shows the best agent fitness recorded over five independent evolutionary runs; two that use Baldwinian evolution (green), two that use Lamarckian evolution (red) and one that uses Darwinian evolution (blue), in the goal direction task. Best performance is obtained by Baldwinian evolution without an explicit plasticity mask, and second best with Baldwinian evolution with an explicit plasticity mask, followed by Darwinian evolution, with Lamarckian evolution a very clear loser in this task.

In the supervised learning task, we observed genetic accommodation of the initial function prior to learning, i.e. the regression network's prior was initially sinusoidal. Rapid learning continued to be selected for throughout evolution. For RL tasks, we observed that Baldwinian evolution (compared to Lamarckian) tended to result in relatively high learning rates and low discount factors, with the initial behaviour 'at birth' providing strong biases to the learning algorithm which continued to show rapid learning throughout evolution. The Baldwin effect is superior to Lamarckian learning when the distribution of tasks is broad or quickly changing (Goal Direction), whereas Lamarckian learning is superior when the task distribution is narrow (Goal Velocity).



(a) Sinusoid fitting task for Baldwinian evolution vs. MAML



(b) Performance of evolution variants on the Cheetah Goal Direction task.

Figure 1: Baldwinian evolution (GA and NES) compared to MAML in sinusoid fitting, and evolution variants in Goal Direction task. "Mask" refers to an evolved binary vector that determines whether or not each parameter can be learned, emulating the setup in Hinton and Nowlan [4].

4 DISCUSSION AND CONCLUSION

We have demonstrated that the Baldwin effect is capable of producing learning algorithms and models capable of few shot learning when combined with deep learning in supervised and reinforcement learning tasks.

Remarkably, meta-learning through evolution enables the use of non-differentiable fitness functions, in contrast to popular meta-learning approaches. For example, the fitness function can be defined on different, potentially multi-modal data distributions, making it a prime candidate for multi-objective optimization, even when data from one or several objectives is not always available to the low level optimization process.

REFERENCES

[1] PA Castillo, MG Arenas, JG Castellano, JJ Merelo, A Prieto, V Rivas, , and G Romero. 2006. Lamarckian evolution and the Baldwin effect in evolutionary neural networks. *arXiv preprint arXiv:cs/0603004* (2006).

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400* (2017).

[3] David E Goldberg and Kalyanmoy Deb. 1991. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of genetic algorithms*. Vol. 1. Elsevier, 69–93.

[4] Geoffrey E Hinton and Steven J Nowlan. 1987. How learning can guide evolution. *Complex systems* 1, 3 (1987), 495–502.

[5] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. (10 2012), 5026-5033 pages.

[6] Peter D Turney. 2002. Myths and legends of the Baldwin effect. *arXiv preprint cs/0212036* (2002).

[7] Daan Wierstra, Tom Schaul, Jan Peters, and Jürgen Schmidhuber. 2008. Natural Evolution Strategies. In *Proceedings of the Congress on Evolutionary Computation (CEC08), Hongkong*. IEEE Press.