Meta-Reinforcement Learning for Adaptive Autonomous Driving

Anonymous Authors[1]

Abstract

Reinforcement learning (RL) methods achieved major advances in multiple tasks surpassing human performance. However, most of RL strategies show a certain degree of weakness and may become computationally intractable when dealing with high-dimensional and non-stationary environments. In this paper, we build a meta-reinforcement learning (MRL) method embedding an adaptive neural network (NN) controller for efficient policy iteration in changing task conditions. Our main goal is to extend RL application to the challenging task of urban autonomous driving in CARLA simulator.

1. Introduction

"Every living organism interacts with its environment and uses those interactions to improve its own actions in order to survive and increase" (Lewis & Vrabie, 2009). Inspired from animal behaviorist psychology, reinforcement learning (RL) is widely used in artificial intelligence research and refers to goal-oriented optimization driven by an impact response or signal (Sutton & Barto, 2018). Properly formalized and converted into practical approaches (Khan et al., 2012), RL algorithms have recently achieved major progress in many fields as games (Mnih et al., 2015; Silver et al., 2016) and advanced robotic manipulations (Levine et al., 2016; Lillicrap et al., 2016) beating human performance. However, and despite several years of research and evolution, most of RL strategies show a certain degree of weakness and may become computationally intractable when dealing with high-dimensional and non-stationary environments (Wahlström et al., 2015). More specifically, the industrial application of autonomous driving in which we are interested in this work, remains a highly challenging "unsolved problem" more than one decade after the promising 2007 DARPA Urban Challenge (Buehler et al., 2009). The origin of its complexity lies in the large variability inherent to driving task arising from the uncertainty of human behavior, diversity of driving styles and complexity of scene perception.

An interpretation of the observed vulnerability due to learning environment changes has been provided in context-aware (dependence) research assuming that "concepts in the real world are not eternally fixed entities or structures, but can have a different appearance or definition or meaning in different contexts" (Widmer, 1997). There are several tasks that require context-aware adaptation like weather forecast with season or geography, speech recognition with speaker origins and control processes of industrial installations with climate conditions. One solution to cope with this variability is to imitate the behavior of human who are more comfortable with learning from little experience and adapting to unexpected perturbations. These natural differences compared to machine learning and specifically RL methods are shaping the current research intending to eschew the problem of data inefficiency and improve artificial agents generalization capabilities (Lake et al., 2017). Tackling this issue as a multi-task learning problem (Caruana, 1997), meta-learning has shown promising results and stands as one of the preferred frames to design fast adapting strategies (Santoro et al., 2016; Ravi & Larochelle, 2017). It refers to learn-to-learn approaches that aim at training a model on a set of different but linked tasks and subsequently generalize to new cases using few additional examples (Finn et al., 2017).

In this paper we aim at extending RL application to the challenging task of urban autonomous driving in CARLA simulator. We build a meta-reinforcement learning (MRL) method where agent policies behave efficiently and flexibly in changing task conditions. We consolidate the approach robustness by integrating a neural network (NN) controller that performs a continuous iteration of policy evaluation and improvement. The latter allows reducing the variance of the policy-based RL and accelerating its convergence. Before embarking with a theoretical modeling of the proposed approach in section 3, we introduce in the next section meta-learning background and related work in order to better understand the current issues accompanying its application to RL settings. In the last section, we evaluate our method using CARLA simulator and discuss experimental results.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Background and Related Work

Generally, in order to acquire new skills, it is more useful to rely on previous experience than starting from scratch. Indeed, we learn how to learn across tasks requiring, each time, less data and trial-and-error effort to conquer further skills (Lake et al., 2017). The term meta-learning that refers to learning awareness on the basis of prior experience was first cited by (Biggs, 1985) in the field of educational psychology. It consists in taking control of a learning process and guiding it in accordance with the context of a specific task. In machine learning research, meta-learning is not a new concept and displays many similarities with the above definition (Thrun & Pratt, 1998; Schmidhuber & Huber, 1991; Naik & Mammone, 1992). It assumes that rather than building a learning strategy on the basis of a single task, it will be more effective to train over a series of tasks sharing a set of similarities then generalize to new situations. By acquiring prior biases, meta-learning addresses models inaccuracies achieving fast adaptation from few additional data (Clavera et al., 2018). At an architectural level, the learning is operated at two scales: a base-level system is assigned to rapid learning within each task, and a meta (higher) level system uses previous one feedback for gradual learning across tasks (Wang et al., 2017).

One of the first contribution to meta-learning is the classical Algorithm Selection Problem (ASP) proposed by (Rice, 1976) considering the relationship between problem characteristics and the algorithm suitable to solve it. Then based on the concept of ASP, the No Free Lunch (NFL) theorem (Wolpert & Macready, 1997) demonstrated that the generalization performance of any learner across all tasks is equal to 0. The universal learner is consequently a myth and each algorithm performs well only on a set of tasks delimiting its area of expertise. ASP and NFL theorem triggered a large amount of research assigned to parameter and algorithm recommendation (Jankowski & Grabczewski, 2011; Brazdil et al., 2008; Soares et al., 2004; Pfahringer et al., 2000). In this type of meta-learning, a meta-learner apprehend the relationship between data characteristics called meta-features and base-learners performance in order to predict the best model to solve a specific task. Various meta-learners have been used and generally consist of shallow algorithms like decision trees, k-Nearest Neighbors and Support Vector Machines (Vanschoren, 2018). Regarding meta-features, the most commonly used ones included statistical and information-theoretic parameters as well as land-marking and model-based extractors (Vilalta et al., 2009).

The recent regain of interest in neural network models and more specifically deep learning resulting from the advent of large training datasets and computational resources allowed the resurgence of neural network Meta-learning (Li et al., 2018). Instead of requiring explicit task characteristics, the meta-level learns from the structure of base-models themselves. Neural networks are particularly suitable to this kind of transfer learning given their inner capabilities of data features abstraction and rule inductions reflected in their connection weights and biases. The typology of meta-learners developed so far includes recurrent models, metrics and optimizers with several areas of application in classification, regression and RL (Li et al., 2017).

Meta-learning algorithms extended recently to the context of RL can be classified in two broad categories. A first set of methods implement a recurrent neural network (RNN) or its memory-augmented variant (LSTM) as the meta-learner. (Duan et al., 2016) study RL optimization in the frame of a reinforcement learning problem (RL2) where policies are represented with RNNs that receive past rewards and actions, in addition to the usual inputs. The approach is evaluated on multi-armed bandits (MAB) and tabular Markov Decision Processes (MDPs). In (Wang et al., 2017), Advantage Actor-Critic (A2C) algorithms with recurrence are trained using different architectures of LSTM (simple, convolutional and stacked). The experiments are conducted on bandits problems with increasing level of complexity (dependent/independent arms and restless).

In the second category, the learner gradients are used for meta-learning. Such methods are task-agnostic and adaptable to any model trained with gradient-descent. The gradient-based strategy has been originally introduced by (Finn et al., 2017) with their Model-Agnostic Meta-Learning (MAML) algorithm. It has been demonstrated efficient for different problem settings including gradient RL with neural network policies. MAML mainly aims at generating a model initialization sensitive to changes and reaching optimal results on a new scenario after just few gradient updates. Meta-SGD (Li et al., 2017) uses stochastic gradient descent to meta-learn, besides a model initialization, the inner loop learning rate and the direction of weights update. In Reptile (Nichol et al., 2018), the authors design a first order approximation of MAML computationally less expensive than the original method which includes second order derivative of gradient. (Shedivat et al., 2018) propose a probabilistic view of MAML for continuous adaptation in RL settings. A competitive multi-agent environment (RoboSumo) was designed to run iterated adaptation games for the approach testing.

A major part of MRL papers have been evaluated either at a preliminary level of experimentation or on elementary tasks (2D navigation, simulated muJoCo robots and bandit problems). In this work we consider an application of gradient-based MRL in a more challenging dynamic environment involving realistic and complex sides of real world tasks, which is CARLA simulator for autonomous driving (Dosovitskiy et al., 2017).

3. MRL with Policy Evaluation

The proposed model consists of a MRL framework embedding an adaptive NN controller to tackle both the non-stationarity and high dimensionality issues inherent to autonomous driving environments in CARLA simulator.

3.1. Preliminaries

The RL task considered in this work is a Markov Decision Process (MDP) defined according to the tuple $(S, A, p, r, \gamma, \rho_0, H)$ where S is the set of states, A is the set of actions, $p(s_{t+1}|s_t, a_t)$ is the state transition distribution predicting the probability to reach a state s_{t+1} in the next time step given current state and action, r is a reward function, γ is the discount factor, ρ_0 is the initial state distribution and H the horizon. Consider the sum of expected rewards (return) from a trajectory $\tau_{(0,H-1)} = (s_0, a_0, ..., s_{H-1}, a_{H-1}, s_H)$. A RL setting aims at learning a policy π of parameters θ (either deterministic or stochastic) that maps each state s to an optimal action a maximizing the return R of the trajectory.

$$R_t = \sum_{j=t}^{t+H-1} \gamma^{j-t} r_{j+1} \tag{1}$$

Following the discounted return expressed above, we can define a state value function $V(s) : S \rightarrow R$ to measure the current state return estimated under policy π:

$$V(s_t) = \mathbb{E}[R_t|s_t = s] \tag{2}$$

In order to optimize the parameterized policy π_θ, we use gradient descents like in the family of REINFORCE algorithms (Williams, 1992) updating the policy parameters θ in the direction:

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s_t|a_t) R_t \tag{3}$$

3.2. Method

We build an approach of NN meta-learning compatible with RL setting. Our contribution consists in combining (1) a gradient-based meta-learner like in MAML (Finn et al., 2017) to learn a generalizable model initialization and (2) a NN controller for more robust and continuous adaptation. The agent policy π_θ approximated by a convolutional neural network (CNN) is trained to quickly adapt to a new task through few standard gradient descents. Explicitly, this consists in finding an optimal initialization of parameters θ^* allowing a few-shot generalization of the learned model. Given a batch of tasks T_i sampled from $p(T)$, the meta-objective is formulated as follows:

$$\max_\theta \mathbb{E}_{T_i \sim p(T)} R_{T_i}(\theta'_i) \; where \; \theta'_i = \theta + \alpha \nabla_\theta R_{T_i}(\theta) \tag{4}$$

The MRL approach includes two levels of processing: the inner and the outer loops associated respectively to the base and meta-learning.

In the inner loop, we start by reducing the disturbances characterizing policy based methods and induced by the score function R_t. Indeed, complex domains with conflicting dynamics and high dimensional observations like autonomous driving yield a large amount of uncertainty. One flexible solution to reduce disturbances and accelerate learning convergence is policy iteration. Subsequently, we modify the RL scheme by integrating a step of policy evaluation and improvement that generates added bonuses to guide the agent towards new states.

The policy evaluation is performed with temporal difference (TD) learning combining Monte Carlo method and dynamic programming (Sutton & Barto, 2018) to learn, with step size ω, the value function approximated by a CNN:

$$V(s_t) = V(s_t) + \omega \delta_t \tag{5}$$

Where δ_t is the multi-step TD error that consists in bootstrapping the sampled returns from the value function estimate:

$$\delta_t = [\sum_{j=t}^{t+H-1} \gamma^{j-t} r_j] + \gamma^H V(s_{t+H}) - V(s_t) \tag{6}$$

Multi-step returns allow the agent to gather more information on the environment before calculating the error in the value function estimates. Subsequently, the improvement of the policy is performed through the replacement of the score function R_t by the TD error δ_t in the policy gradient:

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s_t|a_t) \delta_t \tag{7}$$

For each sampled task T_i, the policy parameters θ'_i are computed using the updated gradient descent:

$$\theta'_i = \theta + \alpha \nabla_\theta \log \pi_\theta(s_t|a_t) \delta_t \tag{8}$$

Once the models and related evaluations are generated for all batch tasks, the outer loop is activated. It consists in operating a meta-gradient update of the initial model parameters with a meta-step size β on the basis of the previous level rewards $R_{T_i}(\theta'_i)$:

$$\theta \leftarrow \theta + \beta \nabla_\theta \sum_{T_i \sim p(T)} R_{T_i}(\theta'_i) \tag{9}$$

The steps detailed above are iterated until an accepted performance is reached. The resulting model initialization θ^* should be able to achieve fast driving adaptation after only a few gradient steps.

4. Experiments

In this section we evaluate the performance of the continuous-adapting MRL model on the challenging task of urban autonomous driving. The goal of our experiment is to demonstrate the effectiveness of meta-level learning combined with a NN controller to optimize the RL policy and achieve a more robust learning of high-dimensional and complex environments. At this stage of work, we present

the preliminary results of our study assessing 2 basic assumptions. The MRL agent is (1) adapting faster at training time and (2) displaying better generalization capabilities in unseen environments.

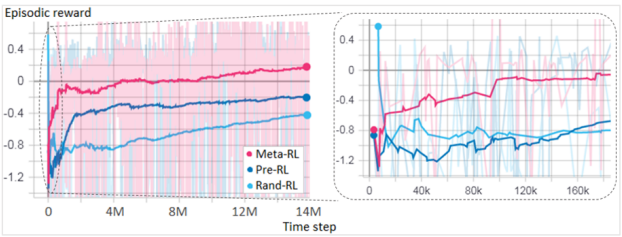

Figure 1. Comparison of our approach (Meta-RL), the pre-trained (Pre-RL) and the randomly initialized (Rand-RL) algorithms in test-time adaptation to unseen environments.

Environment settings. We conduct our experiments using CARLA simulator for autonomous driving (Dosovitskiy et al., 2017; Palanisamy, 2018) designed as a server-client system. Carla 3D environment consists of static objects and dynamic characters. As we consider the problem of autonomous driving in changing conditions, we induce non-stationary environments across training episodes by varying several server settings. (1) The task complexity: select one of the available towns as well as different start and end positions for the vehicle tasks (straight or with-turn driving). (2) The traffic density: control the number of dynamic objects such as pedestrians and vehicles. (3) Weather and lightening: select a combination of weather and illumination conditions to diversify visual effects controlling sun position, radiation intensity, cloudiness and precipitation. Hence we can exclusively use a subset of environments for meta-training ("seen") and a second subset for test-time adaptation ("unseen"). The reward is shaped as a weighted sum of the distance traveled to target, speed in km/h, collisions damage and overlaps with sidewalk and opposite lane.

Results. Given the preliminary level of experiments and the absence of various state-of-the-art work on the recent CARLA simulator, we adopt (Finn et al., 2017) methodology consisting in comparing the continuous-adapting MRL initialization with conventionally pre-trained and randomly initialized RL algorithms. In all experiments the average episodic reward is used to describe the methods global performance. An episode is terminated when the target destination is reached or after a collision with a dynamic character.

Figure 1 depicts the test-time adaptation performance of the 3 models. During this phase, the RL agent initialized with meta-learning still uses the NN controller for continuous adaptation. The results confirm that our approach generates models adapting faster in "unseen" environments comparatively to the standard RL strategies. Zooming the initial driving steps (figure 1), we notice that our method has distinctly surpassed the standard RL versions only after 10000 steps (500 gradient descents). Subsequently we should lead further tests to identify a specific threshold for few shot learning when evolving from low to high-dimensional settings like autonomous driving task.

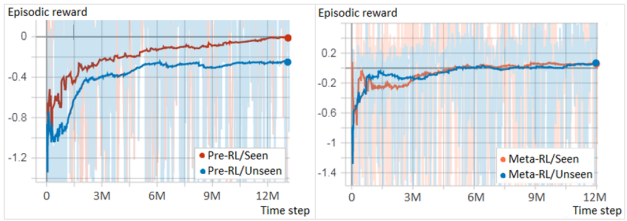

Figure 2. Generalization capabilities of our approach (right) and the pre-trained RL (left): Comparison of adaptation results in "seen" and "unseen" environments.

In order to evaluate the generalization assumption, we compare the models behavior on "seen" and "unseen" environments. Figure 2 does not reveal a significant "shortfall" of our approach performance between the 2 scenarios reflecting its robustness in non-stationary conditions. In the contrary, the performance of the pre-trained standard RL decreased notably in "unseen" environments due to the lack of generalization capabilities.

Although all results indicate a certain robustness of the continuous-adapting MRL, it is too early to draw firm conclusions at this preliminary stage of evaluation. First, the episodic reward indicator should be completed with the percentage of successfully ended episodes in order to demonstrate the effective learning of the agent and allow the comparison with state-of-the-art work (Dosovitskiy et al., 2017; Liang et al., 2018). Second, further consideration should be addressed to the pertinence of few shot learning regimes in very complex and high dimensional environments like autonomous driving since the meta-learned strategy may acquires a particular bias at training time "that allows it to perform better from limited experience but also limits its capacity of utilizing more data" (Shedivat et al., 2018).

5. Conclusion

In this paper we addressed the limits of RL algorithms in solving high-dimensional and complex tasks. Built on gradient-based meta-learning, the proposed approach implements a continuous process of policy assessment and improvement using a NN controller. Evaluated on the challenging problem of autonomous driving using CARLA simulator, our approach showed higher performance and faster learning capabilities than conventionally pre-trained and randomly initialized RL algorithms. Considering this paper as a preliminary attempt to scale up RL approaches to high-dimensional real world applications like autonomous driving, we plan in future work to bring deeper focus on several sides of the approach such as the reward function, CNN architecture and including vehicle characteristics in the tasks complexity setup.

References

Biggs, J. The role of meta-learning in study process. In *The British Psychological Society*, volume 55, pp. 185–212, 1985.

Brazdil, P., Carrier, C., Soares, C., and Vilalta, R. *Metalearning: Applications to Data Mining*. Cognitive Technologies. Springer Berlin Heidelberg, 2008. ISBN 9783540732624.

Buehler, M., Iagnemma, K., and Singh, S. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Springer Publishing Company, Incorporated, 1st edition, 2009.

Caruana, R. Multitask learning. *Mach. Learn.*, 28(1): 41–75, July 1997. ISSN 0885-6125. doi: 10.1023/A:1007379606734. URL https://doi.org/10.1023/A:1007379606734.

Clavera, I., Rothfuss, J., Schulman, J., Fujita, Y., Asfour, T., and Abbeel, P. Model-based reinforcement learning via meta-policy optimization. In *CoRL*, volume 87 of *Proceedings of Machine Learning Research*, pp. 617–629. PMLR, 2018.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. CARLA: An open urban driving simulator. In Levine, S., Vanhoucke, V., and Goldberg, K. (eds.), *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16. PMLR, 2017.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. Rl^2: Fast reinforcement learning via slow reinforcement learning. *CoRR*, abs/1611.02779, 2016.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Jankowski, N. and Grabczewski, K. Universal meta-learning architecture and algorithms. In *Meta-Learning in Computational Intelligence*, 2011.

Khan, S. G., Herrmann, G., Lewis, F. L., Pipe, T., and Melhuish, C. Reinforcement learning and optimal adaptive control: An overview and implementation examples. *Annual Reviews in Control*, 36(1):42 – 59, 4 2012. ISSN 1367-5788. doi: 10.1016/j.arcontrol.2012.03.004.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *The Behavioral and brain sciences*, 40:e253, 2017.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17(1):1334–1373, January 2016.

Lewis, F. L. and Vrabie, D. Reinforcement learning and adaptive dynamic programming for feedback control. *Cir. and Sys. Mag.*, 09(3):32–50, September 2009. ISSN 1531-636X. doi: 10.1109/MCAS.2009.933854. URL http://dx.doi.org/10.1109/MCAS.2009.933854.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-sgd: Learning to learn quickly for few shot learning. *CoRR*, abs/1707.09835, 2017.

Liang, X., Wang, T., Yang, L., and Xing, E. CIRL: controllable imitative reinforcement learning for vision-based self-driving. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211, pp. 604–620. Springer, 2018.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *ICLR*, 2016.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, February 2015.

Naik, D. K. and Mammone, R. J. Meta-neural networks that learn by learning. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pp. 437–442 vol.1, June 1992. doi: 10.1109/IJCNN.1992.287172.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018.

Palanisamy, P. *Hands-On Intelligent Agents with OpenAI Gym: Your Guide to Developing AI Agents Using Deep Reinforcement Learning*. Packt Publishing, 2018.

Pfahringer, B., Bensusan, H., and Giraud-Carrier, C. G. Meta-learning by landmarking various learning algorithms. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 743–750, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Rice, J. R. The algorithm selection problem. *Advances in Computers*, 15:65–118, 1976.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. Meta-learning with memory-augmented neural networks. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1842–1850, New York, New York, USA, 2016. PMLR.

Schmidhuber, J. and Huber, R. Learning to generate artificial fovea trajectories for target detection. *Int. J. Neural Syst.*, 2(1-2):125–134, 1991.

Shedivat, M., Bansal, T., Burda, Y., Sutskever, I., Mordatch, I., and Abbeel, P. Continuous adaptation via meta-learning in nonstationary and competitive environments. In *ICLR*. OpenReview.net, 2018.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, January 2016.

Soares, C., Brazdil, P. B., and Kuba, P. A meta-learning method to select the kernel width in support vector regression. *Machine Learning*, 54(3):195–209, Mar 2004.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Thrun, S. and Pratt, L. Learning to learn. chapter Learning to Learn: Introduction and Overview, pp. 3–17. Kluwer Academic Publishers, Norwell, MA, USA, 1998. URL http://dl.acm.org/citation.cfm?id=296635.296639.

Vanschoren, J. Meta-learning: A survey. *CoRR*, abs/1810.03548, 2018. URL http://arxiv.org/abs/1810.03548.

Vilalta, R., Giraud-carrier, C., Sa, E. I., and Brazdil, P. *META-LEARNING Concepts and Techniques*. Data Mining and Knowledge Discovery Handbook. Springer, 2009.

Wahlström, N., Schon, T. B., and Deisenroth, M. P. From pixels to torques: Policy learning with deep dynamical models. *Deep Learning Workshop at the 32nd International Conference on Machine Learning*, 2015.

Wang, J. X., Kurth-Nelson, Z., Soyer, H., Leibo, J. Z., Tirumala, D., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. V. Learning to reinforcement learn. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society, CogSci 2017, London, UK, 16-29 July 2017*, 2017. URL https://mindmodeling.org/cogsci2017/papers/0252/index.html.

Widmer, G. Tracking context changes through meta-learning. *Mach. Learn.*, 27(3):259–286, June 1997. ISSN 0885-6125. doi: 10.1023/A:1007365809034. URL https://doi.org/10.1023/A:1007365809034.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, pp. 229–256, 1992.

Wolpert, D. H. and Macready, W. G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.