ISSN: 2664-9640

A Meta Learning Algorithm based on Randomized Block

Xingang Xu [a], Qingtao Wu [b]

School of Information Engineering, Henan, University of Science and Technology, Luoyang 471023, China;

[a] xuxingang@stu.haust.edu.cn, [b] wqt8921@haust.edu.cn

Abstract

Computing efficiency is an important criterion for measuring an algorithm. How to accelerate the convergence rate is a major research direction in reinforcement learning. Meta-learning, as a representative of fast learning, avoids the problem of excessive sample training in deep reinforcement learning. In the paper, we propose a meta learning algorithm based on randomized block. On the basis of meta-learning, the algorithm adds random coordinate blocks to speed up the gradient calculation process of the algorithm, so as to achieve the goal of increasing the speed of convergence and reducing the computational cost of the algorithm. In the experiments, we verified the convergence performance of the algorithm in the image dataset Omniglot.

Keywords

Meta learning, randomized block coordinate, convergence rate.

1. Introduction

The increasing amount of data in the information age provides a prerequisite for the development of deep learning. At the same time, because of its powerful function approximation ability, combined with reinforcement learning, it has brought new technological revolutions in some areas, such as UAV control [1], 5G Network [2], Blockchain [3], Robots [4] , etc. However, the research on deep learning mostly exists in the experimental stage, and there is no corresponding theoretical proof. However, due to its complex network structure and a large number of parameter representations, its experimental effects are difficult to reproduce in practical applications. And at the same time in different scenarios, it is difficult to provide enough time and training sample size to ensure the convergence performance of the deep network, deep reinforcement learning has exerted great advantages in more and more fields. However, in the continuous research process of deep reinforcement learning, some difficult problems have appeared, such as large training samples, difficult to reproduce experimental results and high hardware requirements.

Meta-learning is a way of learn to learn. Meta-learning hopes to learn the ability to learn through prior knowledge, rather than just the ability to learn a certain task. It has also shown its advantages in many fields, such as image processing [5], cloud computing [6], text categorization [7], etc. Nowadays, the main research direction of meta-learning is the continuous improvement of its strategy parameters and the application of meta-learning. Meanwhile, meta-learning has achieved good results in training with few samples.

For the development of meta-learning, it is mainly divided into the following aspects. Firstly, the methods based on memory. Santoro [8] proposes a meta-learning method based on the memory mechanism, it stores the last input to combine the next output, which enables the subsequent input to obtain relevant images through external memory for comparison to achieve better predictions. But this method has the problem of excessive calculation. Secondly, the methods based on predictive gradient. Andrychowicz [9] proposes a meta-reinforcement

learning method based on prediction gradient, which trains a deep network to predict gradients. The object of this method is to achieve fast learning, it starts from the perspective of speeding up the update of forecasting strategy parameters, which can accelerate the learning rate. However, in the process of calculating the gradient, a large amount of calculation is still required, and the calculation cost is still high. Thirdly, the methods based on attentional mechanism. Vinyals [9] combines the attentional mechanism with meta-learning, which means that it will focus on the most important part in the data training, the methods gains the final output with the overlay of attention mechanism training. However, the training of the attention mechanism requires a lot of training process, which increases the convergence time of the algorithm. Finally, the method combined with reinforcement learning is a new approach to research meta-learning. Yan [10] adds the additional input in the training to achieve the fast learning, it forces the neural network to learn some task level information, the agent use these information to further the study. But the task-level information need a extra time and training samples to attain, which increases the convergence time in the training.

However, the above algorithms can only be used in one of supervised learning or reinforcement learning. The model agnostic meta-learning method [11] is a new approach to research meta-learning. MAML breaks the restriction that the above algorithm can only be limited to either supervised learning or enhanced learning, which trains multiple tasks at the same time, and then updates gradient with the composite gradient direction. But the algorithm has a large amount of calculation, and the calculation requires a large amount of memory. Jaesik [12] used MAML to generate many candidate solutions, which can be used for reinforcement learning. But did not learn the distribution of the solution, only learned possible solutions; the memory requirement is large, and the sets of parameters are saved. In order to decrease the calculation, this paper proposed a meta-learning algorithm named R-MAML, which combines the model-agnostic meta-learning method with the randomized block method. MAML updates the gradient parameters twice in the training set and the test set, and part of the training set data is used as the prediction set data to reduce the training samples of the algorithm and achieve the goal of quickly adapting to new tasks. In order to improve the convergence speed, we introduce a random coordinate block method in the first gradient update process, which randomly calculates the gradient variable of one dimension to achieve the purpose of reducing the amount of calculation. The paper makes the following contributions.

In this paper, we combines the model-agnostic Meta-learning method with the randomized block method, which decreases the amount of calculation in the gradient process and reduce the training cost of MAML.We use the image dataset omniglot to verify our algorithm, it can be seen from the experimental results that the algorithm can improve its convergence speed and reduce its training time.

The paper is organized as follows. We review the related works in Section 2. And the preliminaries of these paper is described in Section \ref{preliminaries}. The algorithm R-MAML is proposed in Section 3. The specifics of the experiment is described in Section 4. Finally, we conclude this article in Section 5.

2. Related Work

In the case of meta- learning, it is a kind of process-oriented training, rather than the result of learning. It's not learning a mathematical model to predict directly, but learning how to learn a mathematical model faster and better than before. Nikhil [13] proposed a meta learning method based on sequence, which mainly used the whole sequence to the network, which eliminate the problem of information loss due to long memory, the method has a better effect than the algorithms based on deep network, meanwhile, the method decreased the training time in the sample processing. Rein [14] used the coding technology instead of neural network

to process prior knowledge. It encoded historical information into a loss function, which could use the prior knowledge to achieve the faster leaning. However, the loss function is difficult to learn in the tasks with a large state space, it could only work in small experiments, and its performance would decrease in the complex training tasks. Abhishek [15] added a Gaussian latent state in the input of the existing policy, the method could accelerate the training in the training set, and it used prior knowledge and latent state to achieve the purpose of fast training through adaptive learning.

Bradly [16] proposed a meta-learning method named EMAML, which is based on MAML [11]. It changed the gradient update method of MAML and added a exploration in the data training, which enhanced the exploration capacity in training, and the method had a good performance in the experiment. But the training time is longer than other meta-learning methods. Xu [17] constructed an exploration policy to exploration, which mainly changed the gradient updating in deep reinforcement learning, it constructed a stochastic exploration policy, it used the teacher-student model to explore the training. The experiment showed that the method has a better performance than deep reinforcement learning, and the method decreased the training samples in exploration. Ignasi [18] considered the model adaptation in meta learning, facing a new environment, it could quickly adapt to a new model based on historical information, and it specially learned a meta-model for the data training, which could use the model-based approach to achieve better results. The work of this paper is quite enlightening. it really focus on learning a Meta Model, so that it could use model-based method to achieve better effect and more sample efficiency. The previous method could only say that the model information is hidden in the meta reinforcement algorithm. Zhongwen [19] used the meta-learning to learn the discount factor, which skewed the parameters in favor of the outcome. The method achieved an adaptive discount factor in data training, which improved the computational efficiency and training efficiency of the algorithm, this algorithm performed better in experiments than the meta-learning algorithm that manually selects parameters. Kelvin [20] used the meta-learning to learn the reward function, which learned a reward function with a small sample, it used the MAML to learn a prior reward and used it to train new task, it could update quickly to get the relay reward, and the adapted reward is better than training, the method has a good result in image classification. But the experimental results are difficult to reproduce. Abhishek [21] proposed a algorithm Diayn based on mutual information, it could learn without a reward at all, of course, a reward is required, but a reward function is constructed based on mutual information. Then, a lot of skills could be learned through this learning. After that, specific task rewards could be designated to accelerate learning. Li [22] proposed a meta-learning with stochastic gradient descent. In the process of gradient descent, the algorithm used the random coordinate gradient descent method to learn a learning rate for each layer of MAML parameters, which could make the algorithm achieve better training effects during the training process, and at the same time used the idea of random coordinate descent. The training samples in the gradient descent process of the algorithm could be reduced. Alex [23] proposed a meta learning algorithm named Reptile, this algorithm was an extremely simplified version of the classic algorithm MAML, which solved the problem of computation and memory consumption. But the algorithm lacks theoretical proof, and the actual effect in the experiment is not as good as MAML. Erin [24] introduced MAML into the probability framework, so that the task specific parameter is no longer a definite value, but a probability distribution, thereby describing the confidence of the model for the prediction result. But the calculation amount is increased. The Laplace approximation of the probabilistic inference approximation method used has approximate error. Sung [25] used a neural network as a trainable similarity measure, rather than a predefined measure, it could this mechanism to let the network learn faster, and this method used the relation networks to attain the faster training.

ISSN: 2664-9640

3. R-MAML Design

The artificial intelligence has developed rapidly recent years, reinforcement learning is an important method among them. Meanwhile, combining with neural network (e.g., Feedforward Neural Network and Recurrent Neural Network), which is named Deep Reinforcement Learning, is a new way to study. Compared to Reinforcement Learning, it can dispose the high dimensional data and have a better performance in a complex environment. However, the number of data samples for training is large, which makes a long convergence time in the real applications. The meta learning is a new study filed to improve the learning efficiency. Meta-learning is gradually improving the disadvantages of deep reinforcement learning training set collection, which is difficult and the amount of training is huge. Based on the integration of multiple methods such as reinforcement learning and predictive gradients with meta-learning ideas, and modeling for different application scenarios, it can solve practical problems in fixed scenarios.

In this paper, we focus on decreasing the amount of calculation of MAML, we combine the randomized block method to reduce the training cost. In the training, each task is defined as:

$$\mathcal{G}=\{\mathcal{L}(\mathbf{s}_1,\mathbf{a}_1,\ldots,\mathbf{s}_T^M,\mathbf{a}_T^K,q\left(\mathbf{s}_1^1\right),q\left(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t\right),T\},$$

Where $q_i(\mathbf{s}_1)$ is the original state distribution of each task, $q_i(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t)$ is the transition distribution. The state of each flows is took as the input of the Meta-Reinforcement learning. In the process of its first gradient calculation, we randomly select the h-th dimension of block coordinates to update, speeding up the first update process of the algorithm:

$$\sigma'_{i,h}=\sigma-\alpha\nabla_\sigma\mathcal{L}_{\mathcal{G}_{i,h}}\left(f_\sigma\right),$$

where α is the fixed step size. Then the meta-objective is to find a σ to make the good performance of

$$\min_\sigma\sum_{\mathcal{G}_{i,h}\sim(\mathcal{G})}\mathcal{L}_{\mathcal{G}_{i,h}}\left(f_{\sigma'_{i,h}}\right)=\sum_{\mathcal{G}_{i,h}\sim(\mathcal{G})}\mathcal{L}_{\mathcal{G}_{i,h}}\left(f_{\sigma-\alpha\nabla_\sigma\mathcal{L}_{\mathcal{G}_{i,h}}(f_\sigma)}\right),$$

The negative reward is defined as the loss function for task $\mathcal{G}_{i,h}$, the loss function f_ϕ takes the form:

$$\mathcal{L}_{\mathcal{G}_{i,h}}\left(f_\phi\right)=-\mathbb{E}_{\mathbf{x}_t,\mathbf{a}_t\sim f_\phi,q_{\mathcal{G}_{i,h}}}\left[\sum_{t=1}^{N}R_{i,h}\left(\mathbf{x}_t,\mathbf{a}_t\right)\right].$$

Then it updates the parameters σ once more, which takes the form:

$$\sigma\leftarrow\sigma-\beta\nabla_\sigma\sum_{\mathcal{G}_{i,h}\sim p(\mathcal{G})}\mathcal{L}_{\mathcal{G}_{i,h}}\left(f_{\sigma'_{i,h}}\right).$$

where β is the meta step size. The algorithm is summarized in Algorithm 1.

Algorithm 1：R-MAML Algorithm

1. Randomly initialize a distribution over tasks $p(\mathcal{G})$
2. Randomly initialize σ
3. Randomly initialize α： step size
4. Randomly initialize β： meta step size
5. **while** (TRUE) do
6. Sample batch of tasks $\mathcal{G}_i\sim p(\mathcal{G})$

7. **for all** $\mathcal{G}_i$ **do**

8. /*****Training set*****/

9. Sample H trajectories $\tau' = \{(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T^M, \mathbf{a}_T^K)\}$ using f_σ in $\mathcal{G}_{i,\ h}$

10. Evaluate $\nabla_\sigma \mathcal{L}_{\mathcal{G}_{i,h}}\left(f_\sigma\right)$ using τ and $\mathcal{L}_{\mathcal{G}_i}$

12. Randomly select the h-th dimension gradient to update: $\sigma'_{i,h} = \sigma - \alpha \nabla_\sigma \mathcal{L}_{\mathcal{G}_{i,h}}\left(f_\sigma\right)$

14. Sample H trajectories $\tau' = \{(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T^M, \mathbf{a}_T^K)\}$ using $f_{\sigma'}$ in $\mathcal{G}_{i,\ h}$

15. **end for**

16. /*****Test set*****/

17. The second gradient update using τ' and $\mathcal{L}_{\mathcal{G}_{i,\ h}}$ by (4-11):

18. $\sigma \leftarrow \sigma - \beta \nabla_\sigma \sum_{\mathcal{G}_{i,h} \sim p(\mathcal{G})} \mathcal{L}_{\mathcal{G}_{i,h}}\left(f_{\sigma'_{i,h}}\right)$

19. **end while**

14. end for

15. Output: $a_t = f_\sigma\left(s_t\right)$

In short, the algorithm is based on the model-agnostic meta-learning, the algorithm uses two gradient descents in the training set and the test set to update the parameters. Our algorithm use the random coordinate block method in the first training process for a larger sample of the training set, and randomly selects a dimension to update the policy gradient. And the second strategy update for the test set still uses the original parameter update algorithm. In the training process of processing complex samples, the algorithm based on random coordinate blocks can reflect the advantages of the algorithm. In the gradient update process, the training cost of the algorithm is reduced, and the convergence performance of the algorithm is effectively improved.

4. Experiments

In this part, we verify the convergence performance of the algorithm through experiments. The algorithm is mainly based on the image data set Omniglot. We also deploy the classic MAML, and compare the algorithm with its training effect and training time on the image data set Omniglot.

4.1. Experiment Setup

In the experiments, we implement our algorithm R-MAML and MAML in a computer with Inter(R) Xeon(R) Sliver 4114 CPU and 64 RAM. The experiments environment is mainly based on pytorch 1.6.0 [26], Gym [27], and the environment is mainly implemented on a VMware Workstation 15 Pro 15.5.0. In order to express fairness, during the experiment, the results obtained are the average results obtained after multiple training.

ISSN: 2664-9640

The image data set Omniglot [28] is a updated version of MiniImagenet, The dataset contains 16231623 different handwritten characters from 5050 different letters. Each character is drawn online by 2,020 different people via Amazon's Mechanical Turk. The Omniglot dataset contains a total of 5050 letters. We usually divide these into a set of 3030-letter Background sets and a set of 2020-letter Evaluation sets. The more challenging presentation learning task is to use the smaller background sets "Background Small 1" and "Background Small 2". Each contains only 55 letters, more similar to the experience an adult might encounter when learning regular characters.

In the experiments, we setup some experiment parameters as follows： the neural network size is 3*3*32, the meta learning rate is 0.0001, the base learning rate is 0.1, the training epoches is 40000. And we test our algorithm and MAML in a 5 way 1 shot. In order to easy read, the specific parameter settings are shown in Table 1.

Table 1. Summary of experimental parameters

Parameters	R-MAML	MAML
Neural Network size	3*3*32	3*3*32
Meta Learning Rate	0.0001	0.0001
Base Learning Rate	0.1	0.1
Traing step	40000	40000

4.2. Experimental Result

In this part, we verify the effect of the algorithm in actual image processing by comparing the performance of the R-MAML and MAML in the image data set Omniglot. We verified the effect of the algorithm in the data set and test set, and compared the running time of the two algorithms. During the experiment, due to the randomness of the results of the algorithm in the actual classification training, we use the average of multiple classification results as the final result.

As shown in Fig. 1, in the training set of the image data set Omniglot, we can see that the classification accuracy of our proposed R-MAML algorithm and MAML can reach close to 1. It can be seen that in the process of processing data, the effects of the two algorithms in the training set can meet the basic requirements.

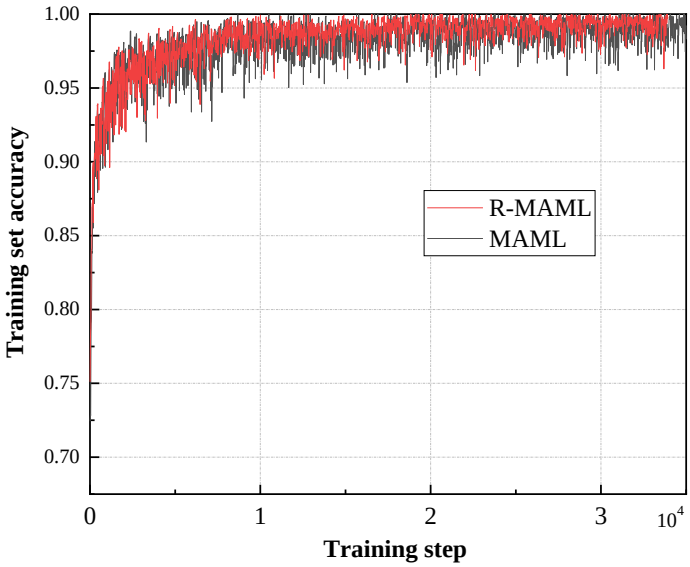

Figure 1. The accuracy of train set

ISSN: 2664-9640

As can be seen from the Fig. 2, our proposed algorithm R-MAML is used in the classification task of the image data set. It can be seen that compared with MAML, our algorithm has similar performance and is doing the same classification. During the task, the accuracy of the algorithm in the test set can reach about 94\%. During the training process, due to the randomness of the results, we use the average of ten training results as the classification accuracy of the two algorithms in the test set.

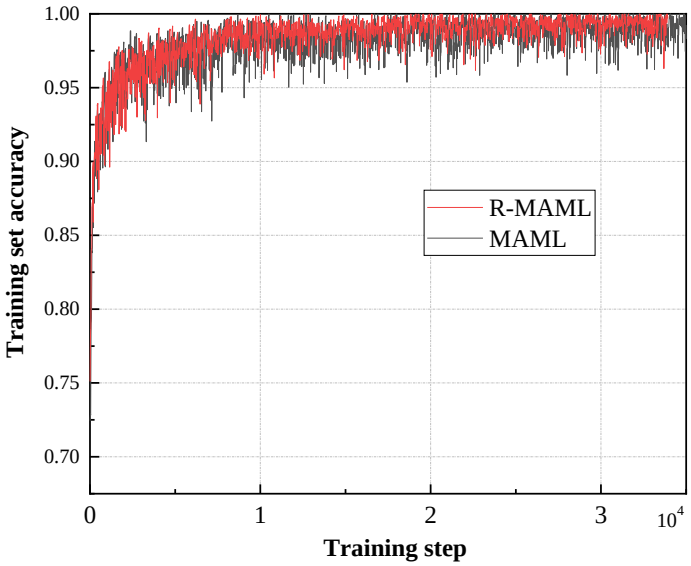

Figure 2. The accuracy of test set

Next, we will compare the training time of the two algorithms, as shown in Fig. 3, We can see that the training time of the R-MAML algorithm is less than that of MAML. From the figure, it can be seen that the actual MAML required in the image training set is close to four hours, while the training time of our proposed algorithm R-MAML is nearly half an hour shorter than it. More precisely, on the same experiment environment, the time required for MAML to process the image data set is 238 minutes, while our proposed algorithm takes 206 minutes. It can be seen that our algorithm effectively reduces the convergence time of the algorithm. In combination with Figure 1, it can be seen that the training results of our proposed algorithm in the test set is close to MAML.

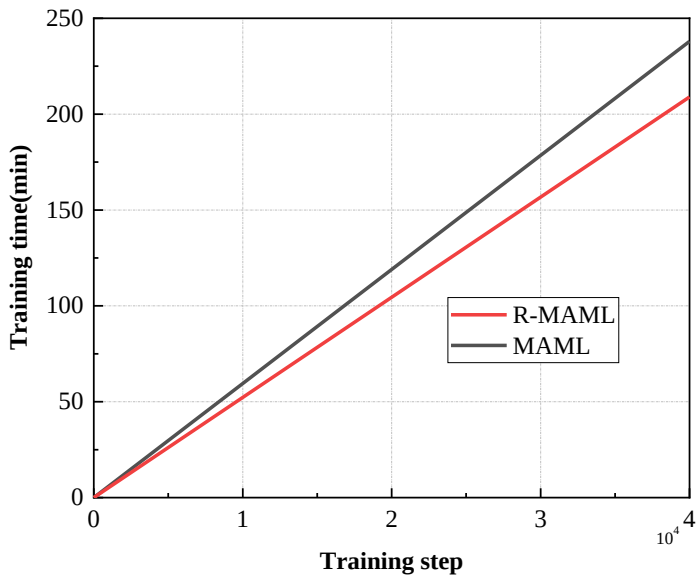

Figure 3. The Training time

Finally, through the above experimental results, we can conclude that our proposed algorithm R-MAML uses random coordinate blocks to improve the convergence performance of MAML. On the basis of ensuring good experimental performance, the algorithm effectively improves the convergence speed of the algorithm, which is very important in dealing with complex tasks.

5. Conclusion

In this paper, based on the MAML meta-learning framework, we use the random coordinate block method to improve the first policy gradient parameters of MAML. We randomly select a dimension coordinate block to update, so that the gradient of the algorithm computational cost in the update process is reduced, and its convergence speed is accelerated at the same time. On the basis of fast learning, in the process of complex task processing, it can have better processing capabilities for high-dimensional data samples and achieve faster learning. We have verified the convergence performance advantage of the algorithm in the image data set through experiments. Computational efficiency is an important criterion for measuring an algorithm. In future work, we should continue to research from the perspective of algorithm stability and computational efficiency improvement, while verifying the performance of the algorithm in different data sets and more complex application scenarios.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants no. 61871430.

References

[1] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "UAV path planning for wireless data harvesting: A deep reinforcement learning approach," in IEEE Global Communications Conference, GLOBECOM 2020, Virtual Event, Taiwan, December 7-11, 2020. IEEE, 2020, pp.1–6.

[2] J. Kim, J. Lee, T. Kim, and S. Pack, "Deep reinforcement learning based cloud-native network function placement in private 5g networks," in IEEE Globecom Workshops, GLOBECOM Workshops 2020, Virtual Event, Taiwan, December 7-11, 2020. IEEE, 2020, pp. 1–6

[3] H. Chen, Z. Chen, F. Lin, and P. Zhuang, "Effective management for blockchain-based agri-food supply chains using deep reinforcement learning," IEEE Access, vol. 9, pp. 36008–36018, 2021.

[4] Y. Xiao, J. Hoffman, T. Xia, and C. Amato, "Multi-agent/robot deep reinforcement learning with macro-actions (student abstract)," in The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020.

[5] P. Zhang, J. Li, Y. Wang, and J. Pan, "Domain adaptation for medical image segmentation: A meta-learning method," J. Imaging, vol. 7, no. 2, p.31, 2021

[6] J. Wang, J. Hu, G. Min, A. Y. Zomaya, and N. Georgalas, "Fast adaptive task offloading in edge computing based on meta reinforcement learning," IEEE Trans. Parallel Distributed Syst., vol. 32, no. 1, pp. 242– 253, 2021.

[7] N. van der Heijden, H. Yannakoudakis, P. Mishra, and E. Shutova, "Multilingual and cross-lingual document classification: A meta-learning approach," CoRR, vol. abs/2101.11302, 2021.

[8] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. P. Lillicrap, "Meta-learning with memory-augmented neural networks," in Proceed-ings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, ser. JMLR Workshop and Conference Proceedings, vol. 48. JMLR.org, 2016, pp. 1842–1850.

[9] M. Andrychowicz, M. Denil, S. G. Colmenarejo, M. W. Hoffman, D. P-fau, T. Schaul, and N. de Freitas, "Learning to learn by gradient descent by gradient descent," in Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016, pp. 3981–3989.

[10] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rl^2: Fast reinforcement learning via slow reinforcement learning," CoRR, vol. abs/1611.02779, 2016.
[11] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 1126–1135.
[12] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, "Bayesian model-agnostic meta-learning," in Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal,Canada, 2018, pp. 7343–7353.
[13] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 – May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.
[14] R. Houthooft, Y. Chen, P. Isola, B. C. Stadie, F. Wolski, J. Ho, and P. Abbeel, "Evolved policy gradients," in Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 5405–5414.
[15] A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine, "Meta-reinforcement learning of structured exploration strategies," in Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 5307–5316.
[16] B. C. Stadie, G. Yang, R. Houthooft, X. Chen, Y. Duan, Y. Wu,P. Abbeel, and I. Sutskever, "Some considerations on learning to explore via meta-reinforcement learning," CoRR, vol. abs/1803.01118, 2018.
[17] T. Xu, Q. Liu, L. Zhao, and J. Peng, "Learning to explore with meta-policy gradient," CoRR, vol. abs/1803.05044, 2018.
[18] I. Clavera, A. Nagabandi, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, "Learning to adapt: Meta-learning for model-based control," CoRR, vol. abs/1803.11347, 2018.
[19] Z. Xu, H. van Hasselt, and D. Silver, "Meta-gradient reinforcement learning," in Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 2402–2413.
[20] K. Xu, E. Ratner, A. D. Dragan, S. Levine, and C. Finn, "Learning a prior over intent via meta-inverse reinforcement learning," in Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 6952–6962.
[21] A. Gupta, B. Eysenbach, C. Finn, and S. Levine, "Unsupervised meta-learning for reinforcement learning," CoRR, vol. abs/1806.04640, 2018.
[22] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few shot learning," CoRR, vol. abs/1707.09835, 2017.
[23] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," CoRR, vol. abs/1803.02999, 2018.
[24] E. Grant, C. Finn, S. Levine, T. Darrell, and T. L. Griffiths, "Recasting gradient-based meta-learning as hierarchical bayes," in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.
[25] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society, 2018, pp. 1199–1208.
[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada,

ISSN: 2664-9640

H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 8024–8035.

[27] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," CoRR, vol. abs/1606.01540,2016.

[28] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," vol. 350, no.6266, pp. 1332–1338, 2015.