Parallel Learning: a Perspective and a Framework

Li Li, *Fellow, IEEE*, Yilun Lin, Nanning Zheng, *Fellow, IEEE*, Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—The development of machine learning in complex system is hindered by two problems nowadays. The first problem is the inefficiency of exploration in state and action space, which leads to the data-hungry of some state-of-art data-driven algorithm. The second problem is the lack of a general theory which can be used to analyze and implement a complex learning system. In this paper, we proposed a general methods that can address both two issues. We combine the concepts of descriptive learning, predictive learning, and prescriptive learning into a uniform framework, so as to build a parallel system allowing learning system improved by self-boosting. Formulating a new perspective of data, knowledge and action, we provide a new methodology called parallel learning to design machine learning system for real-world problems.**

***Index Terms*—Descriptive learning, machine learning, parallel learning, parallel systems, predictive learning, prescriptive learning.**

I. Introduction

MOTIVATED by the rapid development of machine learning and especially the fulminic success of deep learning, data-driven learning has become the state of art not only in many conventional pattern recognition and nature language processing areas [1], [2] but also in some novel areas (e.g., Atari game [3] and Go game [4]). Now, the growth of data-driven learning has impressed both academic and industrial fields. However, there still exist several difficulties that hinder us in deploying machine learning dealing complex real world problems.

First, it is expensive or even impossible to collect all the needed data in a well labeled manner for a data-driven model in a complex system. Even crowdsourcing may not be able to get all the data labeled. We need to let the machine to self-label the data via learning.

Second, for a challenge in the real world, the space of action can be too large to explore without any guide. We need to find an acceptable solution based on the knowledge built on limited available data. A rational constraint on the action space is desirable as well as an efficient explore policy.

Third, lacking a general mathematical framework, it is extremely difficult to assemble different components into a practical system for real-world application, or to analyze and improve the existing works. Three basic elementals, data, knowledge and actions, of a machine learning system should be considered in an integrated way rather than in a separated way.

To address such difficulties, we have to rethink the relationship between data, actions, and knowledge. Conventionally, it is popular to assume that data was sampled independently from a distribution [5], and action was taken by a policy given certain data [6]. Under such assumptions, one might argue that knowledge is just an abstraction of data [7] and have little to do with actions.

However, increasing researchers had realized that data are meaningless without a purpose in many applications. Some researchers consider knowledge as an expectation of actions' consequence. However, without a higher-level understanding the nature of data, such expectations can hardly be generalized from past experience. As shown in [3], though a deep reinforcement learning based controller can play simple Atari games in human level, it cannot deal with complex environments such as Montezuma's revenge.

In this paper, we tackle the data and action by their mutual dependency and consider the knowledge as the way to rebuild this mutual dependency in an artificial system parallel to the real system from observations. The basic idea of parallel system was initialized by Fei-Yue Wang in [8], [9], where he proposed an approach called ACP: artificial societies for modeling, computational experiments for analysis, and parallel execution for control. He proposed the first new paradigm to combine both data and action with an artificial system expressing the knowledge. The exploration of both the state and action space happened independently in the real system and the artificial system, letting the learning process more efficiency and less data-hungry. Such approach has already been applied to solve both theoretical and practical problems [10]−[15].

Combining the core idea of ACP with some cutting-edge techniques including descriptive learning, predictive learning and prescriptive learning, we aim to build a framework named parallel learning [16], [17] that can extend current machine learning methods to deal with data collecting and policy exploring difficulties, and most importantly, guiding the implementation of complex system capable to handle real-world problem.

To present our ideas, the rest of paper organized as follow. The framework and elementary components will be introduced in Section II, then we discuss its role as analyzer and guide line of machine learning system by examples in Section III. Finally, we conclude the paper in Section IV.

Manuscript received May 1, 2017; accepted June 22, 2017. This work was supported in part by the National Natural Science Foundation of China (91520301). Recommended by Associate Editor Derong Liu. *(Corresponding author: Fei-Yue Wang.)*

Citation: L. Li, Y. L. Lin, N. N. Zheng, and F. -Y. Wang, "Parallel Learning: A Perspective and A Framework", *IEEE/CAA J. of Autom. Sinica*, vol. 4, no. 3, pp. 389−395, Jul. 2017.

L. Li is with the Department of Automation, TNList, Tsinghua University, Beijing 100084, China (e-mail: li-li@mail.tsinghua.edu.cn).

Y. L. Lin is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China; and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: linyilun2014@ia.ac.cn).

N. N. Zheng is with the Institute of Artificial Intelligence and Robotics (IAIR), Xi'an Jiaotong University, Xi'an 710049, China (e-mail: nnzheng@mail.xjtu.edu.cn).

F. -Y. Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Research Center for Computational Experiments and Parallel Systems Technology, National University of Defense Technology, Changsha 410073, China (e-mail: feiyue.wang@ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2017.7510493

II. Parallel Learning: a General Machine Learning Framework

A. The Framework, Descriptive Learning and the Parallel Systems

Parallel learning framework aims to build a methodology to determine which operation should be performed based on online-updated knowledge gained from the consequence of actions that we took. The core architecture of the parallel learning is an abstract loop of data, knowledge and action. The loop started from data, refined as knowledge and guide the action, which in turn yields new data to update the knowledge and restart/halt the loop; see Fig. 1 for an illustration.

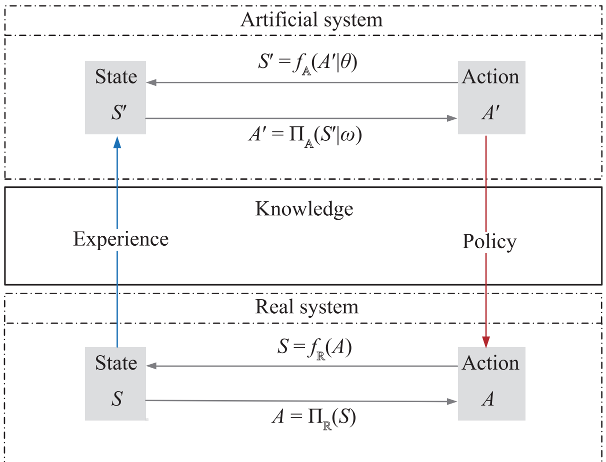


Fig. 1. Parallel learning framework.

The real system $\mathbb{R}$ and artificial system $\mathbb{A}$ run by their own laws. The knowledge, which is model takes the data of one system as input and parameters of the other as output, form and connect them together. Presenting the data as records of mapping from actions to state, the pattern of real system can be distilled as an experience according to a certain purpose, and then be used to rectify the artificial system. The right part of knowledge is the inverse model of experience, which using the updated policy coming from the artificial system to control the actor and gain feedback from the real environment.

The blue lines represent a process from the observation to the imagination, we call it predictive learning, where the system learn to predict the future base on the past and prior knowledge. On the other side, the red line shows a process called prescriptive learning, which is a mapping from imagination to reality by taking control in real worlds. The experience and policy interlaces with each other by applying some constraint upon the update process, as we will discuss later. The bridge between these two processes is a concretization of knowledge and imagination with respect to the purpose of the designer, whose law is self-consistent and do not violate the observation of real system.

Drawn by the prior knowledge and rectified by the further observation, the artificial system exists as both the start and end point of the complete loop of parallel learning. Consider a system containing interactions between actions $a_i \in A, i = 0, 1, \ldots, n$ and state $s_i \in S, i = 0, 1, \ldots, n$. The space of action A is consist of all the possible actions, which can be finite of infinite sequences of operations. The size of A depends only on the capability of the controller. The space of state S consist of all the possible actions, which can be sequences of situations as well. The size of S depends only on the capability of the observer. The interaction in a system can then be defined as

$$\begin{aligned} s_i &= f(a_i), \quad i = 0, 1, \ldots \\ a_j &= \Pi(s_j), \quad j = 0, 1, \ldots \end{aligned} \tag{1}$$

where f describes how actions induce system states and Π denotes the policy that guide our actions based on system state. More precisely, if we aim to gain reward and meanwhile push the system toward a certain state, we can define f and Π as

$$\begin{aligned} f(a_i) &\triangleq \arg\max_{s_j \in S} L(a_i, s_j) \\ \Pi(s_i) &\triangleq \arg\max_{a_j \in A} L(s_i, a_j) \end{aligned} \tag{2}$$

where $L(a_i, s_j)$ refers to the real likelihood that action a_i and state s_j happened sequentially, $Q(s_i, a_j)$ is the real long term reward brought by state s_i and action a_j. In other words, the inductive function f depends on the possibility that some states can be observed by given certain actions, and the policy function Π depends on reward of taken action under given situation. The same structure can be used to describe an artificial system, except that the likelihood and reward are estimated functions; see Fig. 2 for an illustration.

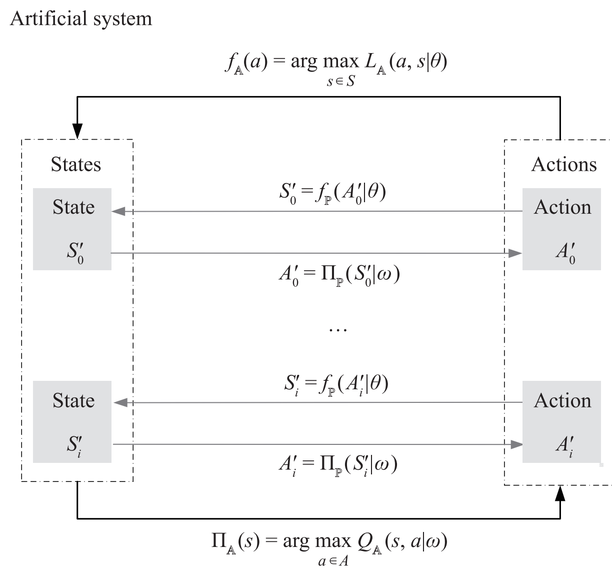


Fig. 2. An artificial system.

Initialized by prior knowledge and limited observations, $L_{\mathbb{A}}$, $Q_{\mathbb{A}}$ of the artificial system can be set to a proper form with reasonable values of θ and ω in an autonomous way. Such process was named as descriptive learning in our methods. The core of descriptive learning is to form a self-consistent system that do not violate the conclusion drawn from observation. We can formalize this process as

$$\begin{aligned} &\arg\min_{\theta,\omega} d_{f,\prod}(|s'_i - f_{\mathbb{A}}(\prod_A(s'_i))|, |a'_i - \prod_A(f_{\mathbb{A}}(a'_i))|) \\ &\qquad i = 0, 1, \ldots, m \\ &\text{s.t.} \quad g_f(|s_j - f_{\mathbb{A}}(a_j)|) \le 0 \\ &\qquad g_{\Pi}(|a_j - \Pi_{\mathbb{A}}(s_j)|) \le 0, j = 0, 1, \ldots, n. \end{aligned} \tag{3}$$

The objective function $d_{f,\Pi}$ is a metric describing how well the mutual-dependency of state and action holds consistent under current setting of parameter θ and ω. By optimizing the measurement with respect to θ and ω, the consistency of the inductive function $f_{\mathbb{A}}$ and policy function $\Pi_{\mathbb{A}}$ increase. For the inequality constraints, g_f is a constraint upon the inductive function, holding it from inducting state s'_j too far from the real sample s_j. The constraint g_Π plays similar role upon the policy function. Building a self-consistent artificial system by descriptive learning, we can then update the likelihood and reward functions by predictive learning and prescriptive learning described in following sections.

The integration of the real and artificial system as a whole is called parallel system. In a first glance, the idea of introducing the parallel system is similar to some previous works (e.g., neural turing machine (NTM) [18] or its advanced version differentiable neural computer (DNC) [19]) that concretize knowledge in the learning process. Like the external memory in NTM and DNC architecture, the parallel system gives controller more space to store and organize the intermediate results of computing, make learning machines that can store knowledge and reason about it flexible, and therefore stabilize the learning process and extend its learning capability. Such knowledge can also be shared among different tasks, which leads to a general intelligent agent. Combining external memories, the learning system can handle several tasks that require rational reasoning, such as planning a multi-stage journey using public transport.

The parallel system goes further than being merely a random-accessed memory, but providing a complete system which is parallel to the real system and runs asynchronously. By executing the artificial system to the real system, the parallel system provides a playground to explore the space of state and action in an efficient way. The efficiency comes from both the nature of simulation and the prior knowledge we introduced in initialization stage. The likelihood and reward functions can update in an adversarial way and converge to an optimal status.

B. Predictive Learning

Started from a prior knowledge coded in parallel system, the learning system explore the state and action space in a simplified ideal environment, then guided by the understanding grasped in this process, it mimics the behavior observed and processes several operations in the real world, which in turn brings more records that reveal the consequences of certain policy. A better understanding of the real system can then be drawn and used to update the artificial system from these new experiences. This is somewhat similar to what Prof. Richard Feynman had stated, "What I cannot create, I do not understand." A self-labeling process can be performed by rectifying the mapping relationship from data to state in the artificial system, letting same actions in both real and artificial system generate similar outputs.

The process to draw knowledge from experiences will be constrained by the knowledge gained from the policy; see Fig. 3 for an illustration. We can define such process as an optimization problem:

$$\begin{aligned}&\min d_\Pi(f_\mathbb{R}, f_\mathbb{A})\\ &\text{s.t.}\quad g_f(|s'_i - f_\mathbb{R}(\Pi_\mathbb{A}(s'_i))|) \le 0, \quad i = 1, \ldots, m\end{aligned} \tag{4}$$

where the objective function d_Π is a metric describing the differences between $f_\mathbb{P}$ and $f_\mathbb{R}$ conditional to the policy Π. By minimizing the differences of occurrence frequency in different systems, we extend the state space of parallel system to the area that has not been discovered yet. On the other side, the inequality in (4), similar to the one in (3), indicating that the predictive learning process will be constrained by policy gained in the parallel system and take those rare situation in the real system into consideration with lower confidence.

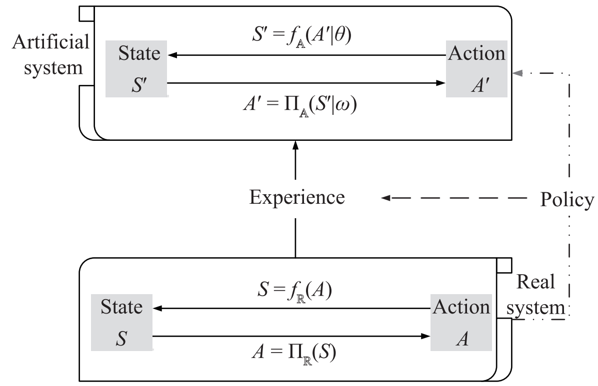


Fig. 3. Predictive learning.

The goal of predictive learning is to minimize the reconstruction error of a generative model. This topic has been studied in unsupervised learning field for a long time and several important paradigms has been established, such as predictability minimization [20], adversarial learning [21], etc. Take the recent achievement of adversarial learning, the generative adversarial nets (GAN) as an example. The objective of GAN is to minimize the difference between real samples and generative samples just like predictive learning [22]. The discriminator is equivalent to that map a sample s_i into an embedding representation a_i, and the generator is equivalent to f that remap representation a_i into s_i. In such perspective, we can consider GAN as a special case of parallel learning. The success of GAN has proved the capability of predictive learning idea while handling with self-labeling missions and therefore can be used to address data-collection problems.

C. Prescriptive Learning

After self-boosting process happened in the parallel system, the learning system grasps an idea about how states transformed after taken certain actions, and therefore it develops an optimal policy. Since the space of action is explored efficiently in parallel learning, we aim to keep the learning process of prescriptive learning as stable as possible and makes it mainly focus on the consequence of actions that likely to happen according to the experience; see Fig. 4 for an illustration.

The prescriptive learning process is concerned about whether the policy learned in the parallel system can actually be adopted in the real world, which means given a state $s_i \in S$, the action $a_i \in A$ given by $\Pi_\mathbb{R}$ should be close to the action given by $\Pi_\mathbb{A}$. According to the goal of prescriptive learning, we can define this learning process as an optimization problem:

$$\begin{aligned}&\min d_f(\Pi_\mathbb{R}, \Pi_\mathbb{A})\\ &\text{s.t.}\quad g_\Pi(|a'_i - \Pi_\mathbb{R}(f_\mathbb{A}(a'_i))|) \le 0, \quad i = 1, \ldots, m\end{aligned} \tag{5}$$

where the objective function d_f is a metric describing the differences between $\Pi_\mathbb{A}$ and $\Pi_\mathbb{R}$ conditional to the generative model f, which can be varied under different circumstances.

In general, d_Π should be correlated to the confidences of different policies under given situation. Minimizing the differences of confidence about their reactions, we can extend the knowledge of policy upon those action space that has not been explored yet in the parallel system. In the meanwhile, the inequality constraint in (5), similar to the one in (3) indicates that, the prescriptive learning process will be constrained by experience gained in the parallel system and it will take those very unlikely taken actions given by real system policy into consideration very carefully.

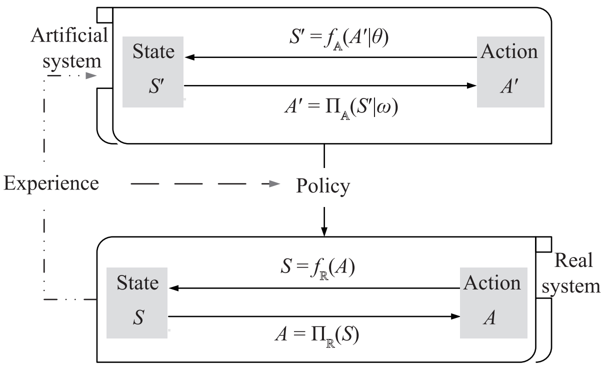


Fig. 4. Prescriptive learning.

Analog to the Inverse reinforcement learning (IRL) [23] and imitation learning (IL) [24], prescriptive learning is about extracting a reward function from given observed behavior. Unlike IRL/IL, we do not rely on an oracle to give an optimal policy. Instead, we will estimate the optimal policy under some prior assumption about reward functions in the parallel system, and we then make this policy more executable by trying to mimic the behaviors of the real system. Such setting compresses the scale of action space into a reasonable size and makes the learning process stable and smooth.

D. The Algorithm of Parallel Learning

Combining the idea of parallel system, prescriptive learning, and predictive learning, we can form a complete loop of data, knowledge, and action. The algorithm of parallel learning consists of three stage, including a self-boosting stage in the parallel system, a self-adaptive stage by prescriptive learning, and a self-labeling stage by predictive learning; see Fig. 5. The sketch of this algorithm can be summarized as the following pseudo code snippet:

Algorithm 1 Parallel Learning Algorithm

1. $L, \theta, Q, \omega, f_{\mathbb{P}}, \Pi_{\mathbb{P}} \leftarrow$ Initialization according to prior knowledge;
2. **repeat**
3.   **repeat**
4. $O^T_{\mathbb{A}} \leftarrow$ Minibatch of T observations consist of $(a'_i,\ s'_i)$, generated in parallel system according to (1);
5.   Update L, Q by $O^T_{\mathbb{A}}$;
6.   Set up $f_{\mathbb{A}}$, $\prod_{\mathbb{A}}$ according to (2);
7.   Perform the descriptive learning according to (3);
8.   **until** convergence of parameters θ, ω;
9. $O^T_{\mathbb{R}} \leftarrow$Minibatch of T observations consist of (a_i, s_i), generated in real system by performing $\Pi_{\mathbb{R}}$;
10.   Perform the prescriptive learning according to (5) and $O^T_{\mathbb{R}}$;
11.   Perform the predictive learning according to (4) and $O^T_{\mathbb{R}}$;
12. **until** max iteration;
13. **return** $f_{\mathbb{A}}, \Pi_{\mathbb{A}}$

III. Application of Parallel Learning

A. Analyze Existing Machine Learning System in Perspective of Parallel Learning Framework

Parallel learning serves as a suitable framework to analyze complex machine learning system. In general, considering actions as reactions to a certain state, we can evaluate our confidence of actions according to a trainable policy. From this viewpoint, we can put a large class of machine learning methods into parallel learning framework.

Let us take the Alpha Go system [4] as an example. Considering the moves taken by Alpha Go as action a, the final results given current board situation as a state s, games played according to the rule of Go as areal system, a Monte Carlo tree was set as a parallel system. The tree with nodes containing the V-value of the board situation corresponding to a given move, which can be viewed as the initial $f_{\mathbb{A}}$. The policy to pick the move leading to then ode with max V-value can be viewed as the initial $f_{\mathbb{A}}$. By training supervised learning (SL) policy networks p_σ, p_π, directly from expert human moves and p_ρ from reinforcement learning (RL), the AlphaGo system performed a prescriptive learning letting $f_{\mathbb{A}}$ more similar to $\Pi_{\mathbb{A}}$. Then AlphaGo played against itself in real system guided by policy networks. Using both the human moves and self-played data, a predictive learning can then be performed. A value network v_θ was trained by mix data, letting it predict more accurate about what state will most possibly happen when it takes certain moves. Finally, the AlphaGo system can then update the value of each state in search tree by using policy and value networks to get a better $f_{\mathbb{A}}$, $f_{\mathbb{A}}$. Such process can loop over and over again to achieve a human-level performance on playing the game of Go. Fig. 6 illustrates the complete framework. A further discussion about the relationship of AlphaGo system and parallel learning can be found in [25].

B. Build New Complex Machine Learning System by Using Parallel Learning Methods

Building a machine learning system upon a parallel system enable us to combine domain knowledge with the data-driven method and thus makes the learning process more efficient and continuous. Such framework has already been adapted to build intelligent transportation systems [26]−[29], vision systems [30], [31] and other social systems [32], [33].

Recently, we design a trajectory planning system for automated vehicles. There exist two major difficulties. First, it is hard to appropriately consider the dynamic constraints of vehicles. Second, it is hard to determine the immediate reward of vehicles. Most existing approaches are indirect trajectory planning in two steps: First, design a reference parking trajectory. Second, design a controller to make vehicle track it. However, such approaches have three problems. First, the dynamic constraints are implicit and inaccurate. Second, it is hard to design a proper controller. Third, if the first step gives a wrong solution, we have no chance to make up in the second step. To solve these problems, we propose direct trajectory planning method whose core idea is to learn the mapping relation between the final state and the corresponding trajectory via deep learning networks [34]. More precisely, we obtain the data by randomly simulating actions and obtaining the resulting trajectories. Then, we use deep neural network

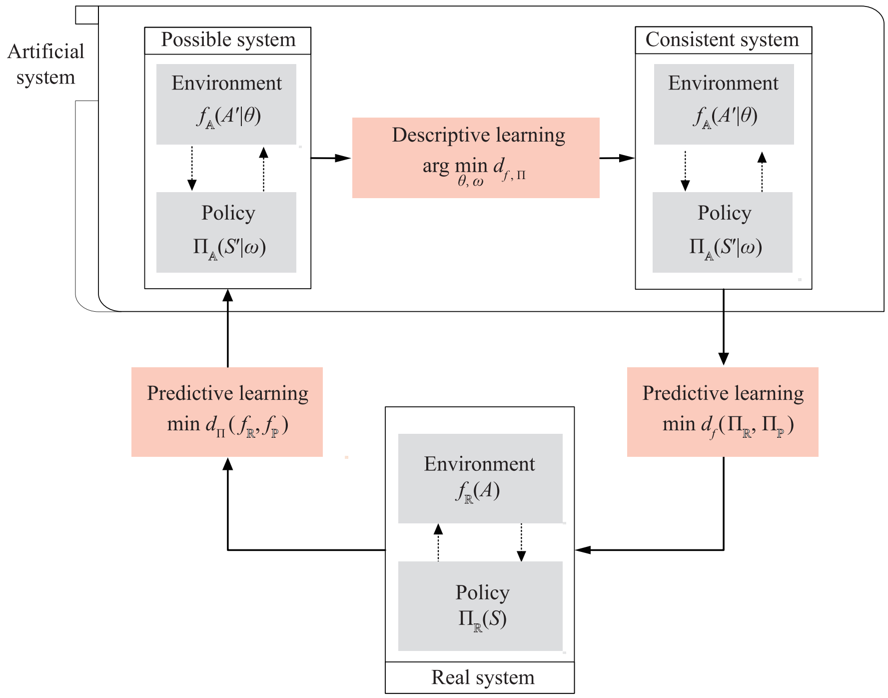


Fig. 5. Parallel learning loop.

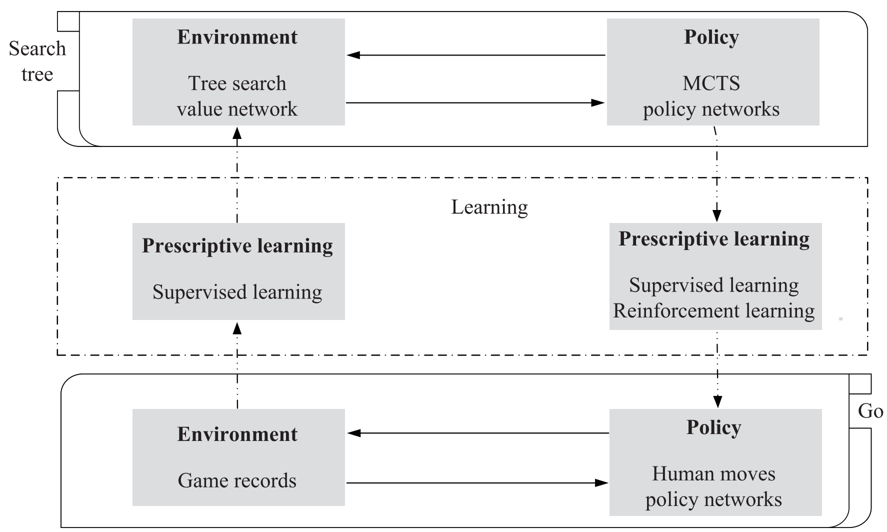


Fig. 6. AlphaGo as a parallel learning system.

to distill and remember the knowledge as the dependence between actions and the resulting trajectories. This is indeed an inverse mapping between actions and the resulting trajectories. Each time when we aim to determine a desired trajectory that links the start point and the destination point, we can directly find the answer from the trained deep neural network. In short, we apply Descriptive Learning to sample all the possible solutions (trajectories), apply Predictive Learning to link the destination with the corresponding actions, and apply Prescriptive Learning to generate trajectory via deep neural network. The benefits of such a new approach include: 1) the dynamic constraints are naturally satisfied; 2) there is no need of complex controller; and 3) we get an integrated solution in just one step. One major difficulty of this system is to master general parking skills so that the obtained knowledge can be used to handle various vehicle dynamics under different environments. A favorable way to solve this problem is using a parallel learning system that automatically adjusts our knowledge according to new observation. The parallel system can be built by combining simulation upon simplified vehicle dynamic model and a data-driven generative model. For example, we first build a trajectory generative model as artificial system. Using the trajectories collected from real parking scenario and generated by simplified bicycle model, we can adjust the parallel system to predict the corresponding trajectories given control sequence in an acceptable level. A planning model can then be learned in this parallel system and used for handling parking problems in real scenario, in turns to generate more raw data to improve the parallel system; see Fig. 7 for a visualization of such idea. Tests reported in [35] has shown that, using deep learning technique to build a trainable model as a parallel system, the parallel learning

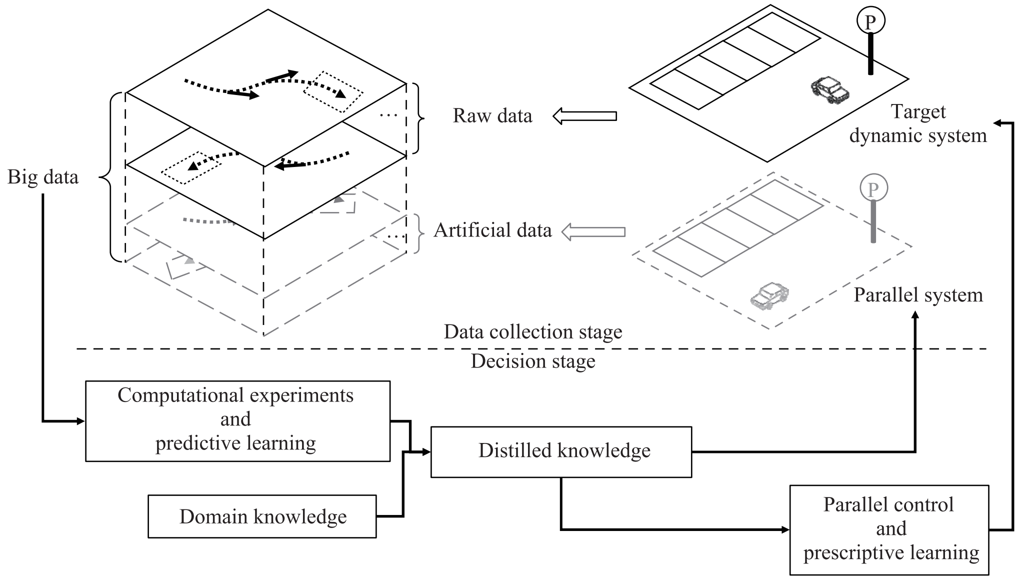


Fig. 7. Parallel learning system for trajectory planning.

system can achieve an excellent performance as well as high transferability for short-term trajectory planning, by using just a small fraction of data comparing to traditional data-driven methods.

IV. Conclusion

In this paper, we introduce the general concept and framework of parallel learning. The core idea behind parallel learning is a new paradigm that takes three basic elements of machine learning: data, knowledge, and action into consideration as a whole system. Particularly, we apply 1) Descriptive learning to distill knowledge from data and learn from data by creating the same (kind of) data. 2) Predictive learning to label data by letting the system evolve in an unsupervised manner. 3) Prescriptive learning to guide the system with growing knowledge and make the system evolve appropriately by special trying-and-testing. Just like what Peter F. Drucker said: "The best way to predict the future is to create it."

As Whitehead once pointed out, "Every categorical type of existence in the world presupposes the other types in terms of which it is explained" [36]. Parallel learning shift the paradigm from considering these elements separately to considering mutual co-constitution. In this new framework, the existence of data and action can only be understood by considering their dependency of each other, as well as their dependency of observer and controller.

By looking at the mutual dependency between data and action, we constrain the exploration of state and action space into a reasonable scale and thus speed up the learning process and address the problem of observation-insufficiency. In addition, the data-driven model, which had long been viewed as an component irrelevant to the data in learning process, should be now considered as an important role in data generation and interpretation process as well. The resulting parallel system, which is a combination of both real and artificial systems, can lead to an efficient observation and a desirable action in the parallel learning process.

Nowadays, while the arsenal of data-driven methods has been filled with guns of model,the data ammunition tends to be inadequate, and the general principle to organize these firepower remains absence. We believe that parallel learning, as one of the first theory to address both problems in one framework, will be vital for the further development of machine learning.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436−444, May 2015.

[2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85−117, Jan. 2015.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529−533, Feb. 2015.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D, Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484−489, Jan. 2016.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, USA: Springer, 2006.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, USA: The MIT Press, 1998.

[7] C. Y. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," ArXiv: 1611.03530, 2016.

[8] F. -Y. Wang, "Artificial societies, computational experiments, and parallel systems: A discussion on computational theory of complex social-economic systems," *Complex Syst. Complex. Sci.*, vol. 1, no. 4, pp. 25−35, Oct. 2004.

[9] F. -Y. Wang, "Toward a paradigm shift in social computing: The ACP approach," *IEEE Intell. Syst.*, vol. 22, no. 5, pp. 65−67, Sept.−Oct. 2007.

[10] F. -Y. Wang, "Toward a revolution in transportation operations: AI for complex systems," *IEEE Intell. Syst.*, vol. 23, no. 6, pp. 8−613, Nov.−Dec. 2008.

[11] F. -Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630−638, Sep. 2010.

[12] F. -Y. Wang, J. Zhang, Q. L. Wei, X. H. Zheng, and L. Li, "PDP: Parallel dynamic programming," *IEEE/CAA J. Autom. Sin.*, vol. 4, no. 1, pp. 1−5, Jan. 2017.

[13] F. -Y. Wang, K. M. Carley, D. Zeng, and W. J. Mao, "Social computing: From social informatics to social intelligence," *IEEE Intell. Syst.*, vol. 22, no. 2, pp. 79−83, Mar. 2007.

[14] F. -Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. magazine*, vol. 4, no. 2, 2009.

[15] F. -Y. Wang and S. N. Tang, "Artificial societies for integrated and sustainable development of metropolitan systems," *IEEE Intell. Syst.*, vol. 19, no. 4, pp. 82−87, Jul.−Aug. 2004.

[16] F. -Y. Wang, X. Wang, L. X. Li, and L. Li, "Steps toward parallel intelligence," *IEEE/CAA J. Automat. Sin.*, vol. 3, no. 4, pp. 345−348, Oct. 2016.

[17] L. Li, Y. L. Lin, D. P. Cao, N. N. Zheng, and F. -Y. Wang, "Parallel learning-a new framework for machine learning," *Acta Autom. Sin.*, vol. 43, no. 1, pp. 1−8, Jan. 2017.

[18] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing machines," ArXiv: 1410.5401, 2014.

[19] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwiska, S. Gmez Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A. P. Badia, K. M. Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, P. Blunsom, K. Kavukcuoglu, and D. Hassabis, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471−476, Oct. 2016.

[20] J. Schmidhuber, "Learning factorial codes by predictability minimization," *Neural Comput.*, vol. 4, no. 6, pp. 863−879, Nov. 1992.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 27th International Conference on Neural Information Processing Systems*, Montreal, Canada, 2014, pp. 2672−2680.

[22] K. F. Wang, C. Gou, Y. J. Duan, Y. L. Lin, X. H. Zheng, and F. -Y. Wang. "Generative adversarial networks: the state of the art and beyond," *Acta Automat. Sin.*, vol. 43, no. 3, pp. 321−332, Apr. 2017.

[23] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning.," in *Proc. 17th International Conference on Machine Learning*, San Francisco, CA, USA, 2000, pp. 663−670.

[24] H. He, H. Daumé III, and J. Eisner, "Imitation learning by coaching," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 3149−3157.

[25] F. -Y. Wang, J. J. Zhang, X, H. Zheng, X. Wang, Y. Yuan, X. X. Dai, J. Zhang, and L. Q. Yang, "Where does AlphaGo go: From church-Turing thesis to AlphaGo thesis and beyond," *IEEE/CAA J. Autom. Sin.*, vol. 3, no. 2, pp. 113−120, Apr. 2016.

[26] J. P. Zhang, F. -Y. Wang, K. F. Wang, W. H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transport. Syst.*, vol. 12, no.4, pp. 1624−1639, Dec. 2011.

[27] F. -Y. Wang and S. M. Tang, "Concepts and frameworks of artificial transportation systems," *Complex Syst. Complex. Sci.*, vol. 1, no. 2, pp. 52−59, Aug. 2004.

[28] Y. S. Lv, Y. J. Duan, W. W. Kang, Z. X. Li, and F. -Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no.2, pp. 865−873, Apr. 2015.

[29] N. Zhang, F. -Y. Wang, F. H. Zhu, D. B. Zhao, and S. M. Tang, "DynaCAS: Computational experiments and decision support for ITS," *IEEE Intell. Syst.*, vol. 23, no. 6, Article no. 10390800, Nov.−Dec. 2008.

[30] K. F. Wang, C. Gou, and F. -Y. Wang, "Parallel Vision: an ACP-based Approach to Intelligent Vision Computing," *Acta Automat. Sin.*, vol. 42, no. 10, pp. 1490−1500, Oct. 2016.

[31] K. F. Wang, C. Gou, N. N. Zheng, J. M. Rehg, and F. -Y. Wang, "Parallel vision for perception and understanding of complex scenes: methods, framework, and perspectives," *Artif. Intell. Rev.*, to be published.

[32] F. -Y. Wang, "Computational experiments for behavior analysis and decision evaluation of complex systems," *J. Syst. Simul.*, vol. 16, no. 5, pp. 893−897, May 2004.

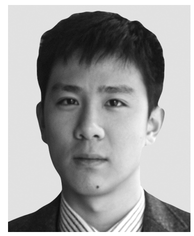
[33] F. -Y. Wang and J. S. Lansing, "From artificial life to artificial societies: New methods for studies of complex social systems," *Complex Syst. Complex. Sci.*, vol. 1, no. 1, pp. 33−41, 2004.

[34] Y. L. Lin, L. Li, X. Y. Dai, N. N. Zheng, and F. -Y. Wang, "Master general parking skill via deep learning," in *Proc. 2017 IEEE Intelligent Vehicles Symposium (IV)*, Crown Plaza, Redondo Beach, CA, USA, 2017.

[35] W. Liu, Z. H. Li, L. Li, and F. -Y. Wang, "Parking like a human: a direct trajectory planning solution," *IEEE Trans. Intell. Transp. Syst.*, no. 99, pp. 1−10, Apr. 2017.

[36] A. N. Whitehead, D. R. Griffin, D. W. Sherburne, *Process and reality: An essay in cosmology*, Cambridge, USA: University Press, 1929.

Li Li (S'05-M'06-SM'10-F'17) is currently an Aassociate Professor with Department of Automation, Tsinghua University, China. His research interests include complex and networked systems, intelligent control and sensing, intelligent transportation systems and intelligent vehicles. He had published over 50 SCI indexed international journal papers and over 50 international conference papers as a first/corresponding author. He serves as an Associate Editor for *IEEE Transactions on Intelligent Transportation Systems.*



YiLun Lin is a Ph.D. candidate at The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include social computing, intelligent transportation systems, deep learning and reinforcement learning.

Nanning Zheng (SM'93-F'06) is a Professor at Institute of Artificial Intelligence and Robotics (IAIR), Xi′an Jiaotong University. Member of Chinese Academy of Engineering. His research interests include pattern recognition and intelligent systems, computer vision and image processing.

Fei-Yue Wang (S'87-M'89-SM'94-F'03) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York in 1990. He joined the University of Arizona in 1990 and became a professor and Director of the Robotics and Automation Laboratory (RAL) and Program in Advanced Research for Complex Systems (PARCS). In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Oversea Chinese Talents Program from the State Planning Council and "100 Talent Program" from CAS, and in 2002, was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory of Management and Control for Complex Systems. Dr. Wang's current research focuses on methods and applications for parallel systems, social computing, and knowledge automation. He was the Founding Editor-in-Chief of the *International Journal of Intelligent Control and Systems* (1995−2000), Founding EiC of *IEEE ITS Magazine* (2006−2007), EiC of *IEEE Intelligent Systems* (2009−2012), and EiC of *IEEE Transactions on ITS* (2009−2016). Currently he is EiC of *China's Journal of Command and Control*. Since 1997, he has served as General or Program Chair of more than 20 IEEE, INFORMS, ACM, ASME conferences. He was the President of IEEE ITS Society (2005−2007), Chinese Association for Science and Technology (CAST, USA) in 2005, the American Zhu Kezhen Education Foundation (2007−2008), and the Vice President of the ACM China Council (2010−2011). Since 2008, he is the Vice President and Secretary General of Chinese Association of Automation. Dr. Wang is elected Fellow of IEEE, INCOSE, IFAC, ASME, and AAAS. In 2007, he received the 2nd Class National Prize in Natural Sciences of China and awarded the Outstanding Scientist by ACM for his work in intelligent control and social computing. He received IEEE ITS Outstanding Application and Research Awards in 2009 and 2011, and IEEE SMC Norbert Wiener Award in 2014.