Deep Reinforcement Learning

Hao Dong • Zihan Ding • Shanghang Zhang
Editors

Deep Reinforcement Learning

Fundamentals, Research and Applications

Springer

Editors
Hao Dong
EECS
Peking University
Beijing, China

Zihan Ding
CS
Imperial College London
London, UK

Shanghang Zhang
EECS
University of California, Berkeley
Berkeley, USA

ISBN 978-981-15-4094-3 ISBN 978-981-15-4095-0 (eBook)
https://doi.org/10.1007/978-981-15-4095-0

Translation from the English language edition: Deep Reinforcement Learning by Hao Dong, Zihan Ding and Shanghang Zhang

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Foreword

I am impressed by the breadth of topics covered by this book. From fundamental underlying theory of deep reinforcement learning to technical implementation with elaborated code details, the authors devoted significant efforts to provide a comprehensive description. Such a style makes the book an ideal study material for novices and scholars. Embracing the open-source community is an indispensable reason for deep learning to have such a rapid development. I am glad that this book is accompanied by the open-source code. I believe that this book will be very useful for researchers who can learn from such a comprehensive overview of the field, as well as the engineers who can learn from scratch with hands-on practice using the open source code examples.

FREng MAE Director of Data Science Institute Yike Guo
Imperial College London
London, UK

This book provides the most reliable entry to deep reinforcement learning, bridging the gap between fundamentals and practices, featuring detailed explanation and demonstration of algorithmic implementation, offering tips and cheat sheet. The authors are researchers and practitioners from leading universities and open source community who conduct research on deep reinforcement learning or apply its new techniques in various applications. The book serves as an extremely useful resource for readers of diverse background and objectives.

Director of the Center on Frontiers of Computing Studies Baoquan Chen
Peking University
Beijing, China

This is a timely book in an important area—deep reinforcement learning (RL). The book presents a comprehensive set of tools in a clear and succinct fashion: covering the foundations and popular algorithms of deep RL, practical implementation details, as well as forward-looking research directions. It is ideally suited for anyone who would like to learn deep RL, to implement deep RL algorithms for their applications, or to begin fundamental research in the area of deep RL.

Princeton University Chi Jin
Princeton, NJ, USA

This is a book for pure fans of reinforcement learning, in particular deep reinforcement learning.

Deep reinforcement learning (DRL) has been changing our lives and the world since 2013 in many ways (e.g. autonomous cars, AlphaGo). It has showed the capability to comprehend the ‘beauty of Go’ better than professionals. The same idea is currently being implemented in technology, healthcare and finance. DRL explores the ultimate answer to one of the most fundamental questions: how do human beings learn from interaction with environment? This mechanism could be a silver bullet of avoiding the ‘big data’ trap, a necessary path towards ‘Strong AI’, as well as a virgin land that no human intelligence has touched before.

This book, written by a group of young researchers with full passion in machine learning, will show you the world of DRL and enhance your understanding by means of practical examples and experiences. Recommend to all learners who want to keep the key to future intelligence in their own pocket.

University College London Kezhi Li
London, UK

Preface

Deep reinforcement learning (DRL) combines deep learning (DL) with a reinforcement learning (RL) architecture. It has been able to perform a wide range of complex decision-making tasks that were previously intractable for a machine. Moreover, DRL has contributed to the recent great successes in artificial intelligence (AI) like AlphaGo and OpenAI Five. Indeed, DRL has opened up many exciting avenues to explore in a variety of domains such as healthcare, robotics, smart grids, and finance.

Divided into three main parts, this book provides a comprehensive and self-contained introduction to DRL. The first part introduces the foundations of DL, RL and widely used DRL methods and then discusses their implementations, which includes Chaps. 1–6. The second part covers selected DRL research topics in Chaps. 7–12, which are useful for those would like to specialize in DRL research. To help readers gain a deep understanding of DRL and quickly apply the techniques in practice, the third part including Chaps. 13–17 presents a rich set of applications, such as the AlphaZero and learning to run, with detailed descriptions.

The book is intended for computer science students, both undergraduate and postgraduate, who would like to learn DRL from scratch, practice its implementation, and explore the research topics. This book might also appeal to engineers and practitioners who do not have strong machine learning background but want to quickly understand how DRL works and use these techniques in their practical applications.

Beijing, China Hao Dong

Acknowledgements

The authors would like to thank the people who provided feedback and suggestions on the contents of the book, including: Jie Fu from Mila, Jianhong Wang and Shikun Liu from Imperial College London, Kun Chen from Peking University, Meng Song from University of California, San Diego, Chen Ma, Chenjun Xiao and Jingcheng Mei from University of Alberta, Tong Yu from Samsung Research, Xu Luo from Fudan University, Dian Shi from University of Houston, Weipeng Zhang from Shanghai Jiaotong University, Yashu Kang from Georgia Institute of Technology, Chenxiao Zhao from East China Normal University, Tianlin Liu from Friedrich Miescher Institute, Gavin Ding from Borealis AI, Ruilong Su from Xiaohongshu Technology Co., Ltd., and Yingjun Pei from Chinese Academy of Sciences. We also want to thank Jared Sharp for the language proofread of most chapters in the book.

Many other people have contributed to this and the code base of the book—open-source contributors, such as Ruihai Wu, Luo Mai, Rundi Wu, Guo Li, Cheng Lai, and Jonathan Dekhtiar, who develop and maintain TensorLayer and the reinforcement learning examples, and colleagues who have provided important insights into the book design. To all these, we offer our thanks and gratitude. Hao Dong would especially like to thank the Center on Frontiers of Computing Studies of the Department of Computer Science at Peking University and Peng Cheng Laboratory for the strong support of developing and maintaining TensorLayer. Zihan Ding would like to thank Dr. Edward Johns for sharing his understandings and useful discussions.

Contents

Editors and Contributors

About the Editors

Hao Dong is currently an Assistant Professor at Peking University. He received his Ph.D. in Computing from Imperial College London in 2019, supervised by Prof. Yike Guo. Hao's research chiefly involves Deep Learning and Computer Vision, with the goal of reducing the amount of data required for learning intelligent systems. He is passionate about popularizing artificial intelligence technologies and established TensorLayer, a deep learning and reinforcement learning library for scientists and engineers, which won the Best Open Source Software Award at ACM Multimedia 2017.

Zihan Ding received his M.Sc. degree in Machine Learning with distinction from the Department of Computing, Imperial College London, supervised by Dr. Edward Johns. He holds double Bachelor degrees from the University of Science and Technology of China: in Photoelectric Information Science and Engineering (Physics) and in Computer Science and Technology. His research interests include deep reinforcement learning, robotics, computer vision, quantum computation and machine learning. He has published papers in ICRA, AAAI, NIPS, IJCAI, and Physical Review. He also contributed to the open-source projects TensorLayer RLzoo, TensorLet and Arena.

Shanghang Zhang is a postdoctoral research fellow in the Berkeley AI Research (BAIR) Lab, the Department of Electrical Engineering and Computer Sciences, UC Berkeley, USA. She received her Ph.D. from Carnegie Mellon University in 2018. Her research interests cover deep learning, computer vision, and reinforcement learning, as reflected in her numerous publications in top-tier journals and conference proceedings, including NeurIPS, CVPR, ICCV, and AAAI. Her research mainly focuses on machine learning with limited training data, including low-shot learning, domain adaptation, and meta-learning, which enables the learning system to automatically adapt to real-world variations and new environments. She was one of the “2018 Rising Stars in EECS”İ (a highly selective program launched at MIT in 2012, which has since been hosted at UC Berkeley, Carnegie Mellon, and Stanford annually). She has also been selected for the Adobe Academic Collaboration Fund, Qualcomm Innovation Fellowship (QInF) Finalist Award, and Chiang Chen Overseas Graduate Fellowship.

About the Authors

Hang Yuan is currently a Ph.D. candidate of Computer Science at the University of Oxford, specializing in AI Safety for Deep Learning and its applications in Healthcare AI. He conducted his master thesis at Swiss Federal Institute of Technology Lausanne (EPFL) with the Computer Vision Lab under Dr. Mathieu Salzmann and Dr. François Fleuret on the topic of delayed adversarial attack using recurrent neural networks for Deep Reinforcement Learning. Previously, he has also researched and studied at Carnegie Mellon University, Max Planck Institute for Intelligent Systems Empirical Inference Group and Imperial College London. He obtained his MSc degree at EPFL in Neuroscience and BSc at Jacobs University in Computer Science under the supervision of Prof. Herbert Jaeger.

Hongming Zhang is currently an engineer at the Institute of Automation, Chinese Academy of Sciences (CASIA). His research focuses on Reinforcement Learning and Game Theory. Before CASIA, he received his MSc degree in Statistics from Peking University, Bachelor degree in Mathematics from Beijing Normal University.

Jingqing Zhang is currently a Ph.D. candidate at Data Science Institute, Imperial College London under the supervision of Prof. Yike Guo. His research interest includes Deep Learning, Machine Learning, Text Mining, Data Mining and their applications. He received his BEng degree in Computer Science and Technology from Tsinghua University, 2016, and MRes degree with distinction in Computing from Imperial College London, 2017.

Yanhua Huang is currently a software engineer at Xiaohongshu Technology Co., Ltd., working on large-scale machine learning and reinforcement learning in recommender systems. He received his B.S. degree from the Department of Mathematics, East China Normal University in July 2016. Yanhua also contributed to some open-source projects, such as PyTorch, TensorFlow, and Ray.

Tianyang Yu is currently a MSc candidate of Computer Science at Nanchang University. Previously, he interned at the Institute of Automation, Chinese Academy of Sciences. Tianyang is interested in Reinforcement Learning and has strong experiences on applying Reinforcement Learning techniques into real-world applications.

Huaqing Zhang is currently a software engineer at Google LLC, exploring on the areas of multi-agent reinforcement learning and hierarchical game theory. He received the B.S. degree in Huazhong University of Science and Technology, Wuhan, China, in June 2013, and the Ph.D. degree in the department of electronic and computer engineering at University of Houston, Houston, TX, USA, in December 2017.

Ruitong Huang is currently a researcher at Borealis AI. His research interests broadly include topics such as online learning, convex optimization, adversarial learning, and reinforcement learning. Ruitong obtained his PhD in Statistical Machine Learning from the computing science department of University of Alberta. Before that, Ruitong spent four years at the University of Science and Technology of China for his Bachelor degree in Math and two years in the David R. Cheriton School of Computer Science at University of Waterloo for his Master's in Symbolic Computation.

Acronyms

AC	Actor-critic
ACKTR	Actor-critic using Kronecker-factored trust region
AGAIL	Action-guided adversarial imitation learning
AI	Artificial intelligence
AIRL	Adversarial inverse reinforcement learning
ANN	Artificial neural network
A2C	Advantage actor-critic
A3C	Asynchronous advantage actor-critic
BC	Behavioral cloning
BCO	Behavioral cloning from observation
BO	Bayesian optimization
BPTT	Backpropagation through time
CE	Cross entropy
CFD	Contrastive forward dynamics
CMA	Covariance matrix adaptation
CMA-ES	Covariance matrix adaptation evolution strategy
CNN	Convolutional neural network
CPU	Central processing unit
C51	Categorical 51
DAgger	Dataset aggreation
DDPG	Deep deterministic policy gradient
DDPGfD	Deep deterministic policy gradient from demonstration
DL	Deep learning
DMP	Dynamic movement primitives
DNN	Deep neural network
DP	Dynamic programming
DPG	Deterministic policy gradient
DQN	Deep Q-network
DQfD	Deep Q-learning from demonstrations
DRL	Deep reinforcement learning
EM	Expectation maximization

FAIL	Forward adversarial imitation learning
FC	Fully connected
FRL	Feudal reinforcement learning
FuN	Feudal network
GAN	Generative adversarial network
GAN-GCL	Generative adversarial network guided cost learning
GAIL	Generative adversarial imitation learning
GCL	Guided cost learning
GMM	Gaussian misture model
GMR	Gaussian mixture regression
GP	Gaussian process
GPU	Graphics processing unit
GPI	Generalized policy iteration
GPR	Gaussian process regression
HAM	Hierarchical abstract machine
HIRO	Hierarchical reinforcement learning with off-policy correction
HRL	Hierarchical reinforcement learning
IfO	Imitation learning from observation
IL	Imitation learning
ILPO	Imitating latent policies from observation
IMPALA	Importance weighted actor-learner architecture
InRL	Independent reinforcement learning
IRL	Inverse reinforcement learning
KL	Kullback-Leibler
KMP	Kernelized movement primitives
LQR	Linear quadratic regulators
LSTM	Long short-term memory
MARL	Multi-agent reinforcement learning
MaxEnt	Maximum entropy
MC	Monte Carlo
MCTS	Monte Carlo tree search
MDP	Markov decision process
ML	Machine learning
MLP	Multi-layer perceptron
MPO	Maximum a posteriori policy optimization
MRP	Markov reward process
MSE	Mean square error
NAC	Normalized actor-critic
OU	Ornstein-Uhlenbeck
PBT	Population based training
PER	Prioritized experience replay
PG	Policy gradient
POMDP	Partially observed Markov decision process
PPO	Proximal policy optimization
ProMP	Probabilistic movement primitives

QR-DQN	Quantile regression deep Q-network
RBF	Radial basis function
RCANs	Randomized-to-canonical adaptation networks
ReLU	Rectified linear unit
RIDM	Reinforced inverse dynamics modeling
RL	Reinforcement learning
RNN	Recurrent neural network
R2D2	Recurrent replay distributed DQN
SAC	Soft actor-critic
SEED	Scalable and efficient deep-RL
Sim2Real	Simulation to reality
SMDP	Semi-Markov decision process
SPG	Stochastic policy gradient
SRL	State representation learning
SVG	Stochastic value gradients
TCN	Time-contrastive networks
TD	Temporal difference
TD3	Twin delayed deep deterministic policy gradient
TRPO	Trust region policy optimization
UCB	Upper confidence bound
UCT	Upper confidence bounds applied to trees
VIME	Variational information maximizing exploration

Mathematical Notation

Jingqing Zhang, jingqing.zhang15@imperial.ac.uk.
We have tried to minimize the mathematical content of this book so as to minimize the requirements for understanding this field.

Fundamentals

x	A scalar
$\boldsymbol{x}$	A vector
$\boldsymbol{X}$	A matrix
$\mathbb{R}$	The set of real numbers
$\frac{\mathrm{d}y}{\mathrm{d}x}$	Derivative of y with respect to x
$\frac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\boldsymbol{x}} y$	Gradient of y with respect to $\boldsymbol{x}$
$\nabla_{\boldsymbol{X}} y$	Matrix derivatives of y with respect to $\boldsymbol{X}$
$P(X)$	A probability distribution over a discrete variable
$p(X)$	A probability distribution over a continuous variable, or over a variable whose type has not been specified
$X \sim p$	The random variable X has distribution p
$\mathbb{E}[X]$	Expectation of a random variable
$\mathrm{Var}[X]$	Variance of a random variable
$\mathrm{Cov}(X, Y)$	Covariance of two random variables
$D_{\mathrm{KL}}(P \Vert Q)$	Kullback-Leibler divergence of P and Q
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Deep Reinforcement Learning

s, s'	States
a	Action
r	Reward
R	Reward function
$\mathcal{S}$	Set of all non-terminal states
$\mathcal{S}^+$	Set of all states, including the terminal state
$\mathcal{A}$	Set of actions
$\mathcal{R}$	Set of all possible rewards
$\boldsymbol{P}$	Transition matrix
t	Discrete time step
T	Final time step of an episode
S_t	State at time t
A_t	Action at time t
R_t	Reward at time t, typically due, stochastically, to A_t and S_t
G_t	Return following time t
$G_t^{(n)}$	n-step return following time t
G_t^{λ}	λ-return following time t
π	Policy, decision-making rule
$\pi(s)$	Action taken in state s under *deterministic* policy π
$\pi(a\|s)$	Probability of taking action a in state s under *stochastic* policy π
$p(s', r\|s, a)$	Probability of transitioning to state s', with reward r, from state s and action a
$p(s'\|s, a)$	Probability of transitioning to state s', from state s taking action a
$v_\pi(s)$	Value of state s under policy π (expected return)
$v_*(s)$	Value of state s under the optimal policy
$q_\pi(s, a)$	Value of taking action a in state s under policy π
$q_*(s, a)$	Value of taking action a in state s under the optimal policy
V, V_t	Estimates of state-value function $v_\pi(s)$ or $v_*(s)$
Q, Q_t	Estimates of action-value function $q_\pi(s, a)$ or $q_*(s, a)$
τ	Trajectory, which is a sequence of states, actions and rewards, $\tau = (S_0, A_0, R_0, S_1, A_1, R_1, \dots)$
γ	Reward discount factor, $\gamma \in [0, 1]$
ϵ	Probability of taking a random action in ϵ-greedy policy
α, β	Step-size parameters
λ	Decay-rate parameter for eligibility traces

Introduction

Ever since the advent of the first computer in 1946, people have been striving to create more intelligent computers. Artificial Intelligence (AI) has benefited so much from the rapid development in the computing power and data volume that it can already outperform humans on many tasks, which were once considered intractable for machines such as board games like chess and Go, disease diagnosis, and video gaming. AI technology is also widely incorporated into other applications like drug discovery, weather prediction, advanced materials, recommended system, robotics perception and control, autonomous driving, human face recognition, speech recognition and dialog.

In the recent decade, not only do countries like China, the UK, the US, Japan and Germany have enacted concrete AI policies to support the development of AI but also tech giants like Google, Facebook, MicroSoft, Apple, Baidu, Huawei and Tencent have spent billions on AI research. AI is becoming almost omnipresent in our daily life, a few examples of which can be self-driving car, face ID, and chatbots. Without a doubt, AI is of paramount importance for the development of human society.

Before we dive into this book, we should first understand the relationships between various subdomains of AI, namely, machine learning (ML), deep learning (DL), reinforcement learning (RL), and the topic of this book—deep reinforcement learning (DRL). Figure 1 illustrates their relationships in a Venn diagram, and we will start to briefly introduce each of them in the following.

Artificial Intelligence

Since computers were first invented, scientists have endeavored to make the machines become more intelligent. However, the definition of intelligence even till today is still in an ongoing debate. So, without defining what intelligence is, Sir Alan Turing first introduced the Turing Test in his paper "Computing Machinery and Intelligence" at University of Manchester in 1950. The Turing test measures a

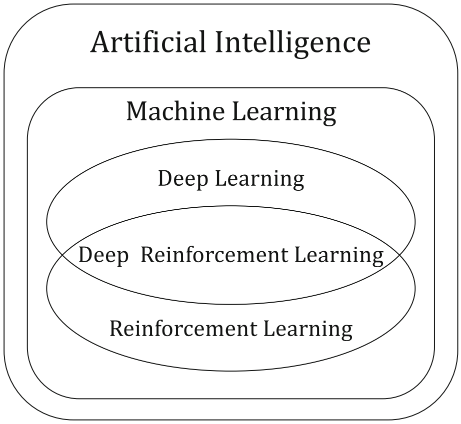

Fig. 1 Relationship of artificial intelligent, machine learning, deep learning, reinforcement learning, and deep reinforcement learning

machine's capability to imitate intelligent human behavior. Specifically, it describes an "imitation game", during which an interrogator asks a man and a computer in another room a series of questions, to determine which of the other two players is man, and which one is computer. The test is passed, if the computer can fool the interrogator.

AI was coined by John McCarthy in the famous Dartmouth conference in summer of 1956. This conference was seen as the starting point of AI being a field of computer science. In the early days of AI, the AI algorithms were mainly designed to solve problems that can be formulated by mathematical rules and logic rules.

Machine Learning

ML was coined in 1959 by Arthur Samuel (Bell Labs, IBM, Stanford). An AI system needs to has the ability to learn its own knowledge from the raw data. This capacity is known as ML. Many AI problems can be solved by designing a pattern recognition algorithm to extract features from raw data for that problem, and then providing these features to the ML algorithm.

For example, in the early days, to perform face recognition with a computer, we need specific facial feature extraction algorithms. The simplest way is to use Principal Component Analysis (PCA) to reduce the data dimension, and the feed these features into a classifier. Handcrafted feature engineering specific for face recognition is often required to improve the recognition performance. Nonetheless, it is fairly time-consuming to design the task-specific handcrafted feature extraction algorithms for different tasks, and let alone in many cases, it is extremely difficult to design a feature extraction algorithm. For example, the feature extraction of language translation requires the knowledge of grammar, which may require many language experts. A general algorithm is desired to extract features for different tasks, so as to reduce the reliance on prior knowledge from human.

Academics have invested lots of efforts in making ML learn the data representation automatically. Learning representation automatically is able to not only improve the performance but also rapidly reduce the cost to solve the AI problems.

Deep Learning

Deep Learning (DL) is a subset of ML algorithms based on artificial neural networks (ANN) Goodfellow et al. (2016). We call it neural network because it is inspired by biological neural networks. In 1943, Warren Sturgis McCulloch and Walter Pitts published "A Logical Calculus of the Ideas Immanent in Nervous Activity," McCulloch and Pitts (1943) which are deemed as the foundations for ANN. Since then, ANN shows the potential of automatic feature learning in which we do not need to design a specific feature learning algorithm for difficult input data, saving the development time of algorithms.

Deep Neural Network (DNN) is the "deep version" of ANN that consists of more neural network layers and can have greater data representation capacity as compared with the "shadow" neural networks. The difference between DL and non-DL methods is illustrated in Fig. 2, in which the DL methods free developers from hand-craft feature engineering to extracting and selecting useful features from input data for the final tasks. We also sometimes call this end-to-end learning as we only care about the input and the output and less on the feature. It is worth noting that this layer of abstraction is not always better as many people have spotted that DL methods tend to offer less transparency and interpretability.

Despite the promises DL has shown today, in the early step of DL history, due to the high computational cost of ANN, the hardware limitation of computers, and the black-box problem (we cannot explain what features the neural networks learned), DL was limited to use in practice and did not get much attention in academia.

This situation changed in 2012, mainly due to a neural network architecture called Alexnet Krizhevsky et al. (2012) which outperformed previous non-DL algorithms by more than 10% in image classification challenge event, ImageNet Russakovsky et al. (2015). DL starts to receive more attention and DL-based methods start to outperform many non-deep learning methods in different fields, such as computer vision Girshick (2015); Johnson et al. (2016); Ledig et al. (2017); Pathak et al. (2016); Vinyals et al. (2016) and natural language processing Bahdanau et al. (2015).

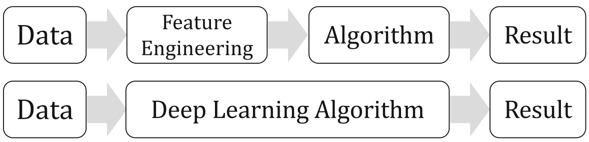

Fig. 2 Non-deep learning vs. deep learning algorithms

Reinforcement Learning

Even though, DL has a powerful data representation ability but it is not enough to build a smart AI system. This is because an AI system should not only able to learn from the provided data but also able to learn from interactions with the real world environment like a human. RL is a subset of ML that enables computers to learn by interacting with the real world environment.

In brief, RL separates the real world into two components—an environment and an agent. The agent interacts with the environment by performing specific actions and receives feedback from the environment. The feedback is usually termed as the “reward” in RL. The agent learns to perform “better” by trying to get more positive rewards from the environment. This learning process forms a feedback loop between the environment and agent, guiding the improvement of the agent with RL algorithms.

Deep Reinforcement Learning

DRL is to combine the advantages of DL and RL for building AI systems. The main reason to use DL in RL is to leverage the scalability of DNN in high-dimensional space, for example, the value function approximation utilizes the data representation of DNN to represent the highly compositional data distribution through end-to-end gradient-based optimization.

DeepMind, a research-oriented AI company established in London, plays an important role in the DRL history. In 2013, just one year after Alexnet, they published “Playing Atari with Deep Reinforcement Learning” which is the first successful DL model that learned how to play seven different Atari games using the raw pixels as the input without any adjustment of the model and learning algorithm. Different from the previous methods that relied on handcrafted features, DeepMind’s method frees developer from feature engineering and outperforms all previous methods on six of the games and even surpasses a human expert on three of them.

In 2017, DeepMind’s AlphaGo defeated the No.1 GO player Jie Ke in China, this event indicates that AI has the ability to perform better than human in a predefined environment via DRL algorithms. DRL is recognized as a subfield of ML that has the potential to achieve Artificial General Intelligence (AGI). However, there are still many challenges need to be addressed before we reach that point.

TensorLayer

Often, understanding the concepts is one thing and having to implement the mathematical formulae is a whole other thing. Therefore, at the end of many chapters of this book, we will also include a practical section in which we implement some of the key concepts in the corresponding chapter to better illustrate how different concepts are used in practice. Since DL is becoming increasingly popular, there exist many open-source frameworks, such as TensorFlow, Chainer, Theano, and Pytorch, to support automatic optimization for neural networks. In this book, we choose to adopt TensorLayer, a DL and DRL library designed specifically for researchers and engineers, which won the Best Open Source Software Award issued by ACM Multimedia in 2017. By the time we publish this book, TensorLayer supports TensorFlow as the computational backend, but with the continuous developing, TensorLayer may support more backends and the usage may be changed. Please refer to Github for more information https://github.com/tensorlayer/tensorlayer.

Beijing, China — Hao Dong
Berkeley, USA — Shanghang Zhang

References

Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: Proceedings of the international conference on learning representations (ICLR)

Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1440–1448

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge. http://www.deeplearningbook.org

Johnson J, Alahi A, Fei-Fei L (2016) Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of the European conference on computer vision (ECCV)

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of the neural information processing systems. Advances in neural information processing systems, pp 1097–1105

Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)

McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5(4):115–133

Pathak D, Krahenbuhl P, Donahue J, Darrell T, Efros AA (2016) Context encoders: feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2536–2544

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252

Vinyals O, Toshev A, Bengio S, Erhan D (2016) Show and tell: lessons learned from the 2015 mscoco image captioning challenge. IEEE Trans Pattern Anal Mach Intell. arXiv:1609.06647v1