

Automated Machine Learning and Meta-Learning for Multimedia

Wenwu Zhu • Xin Wang

Automated Machine Learning and Meta-Learning for Multimedia

 Springer

Wenwu Zhu
Department of Computer Science
and Technology
Tsinghua University
Beijing, China

Xin Wang
Department of Computer Science
and Technology
Tsinghua University
Beijing, China

ISBN 978-3-030-88131-3 ISBN 978-3-030-88132-0 (eBook)
<https://doi.org/10.1007/978-3-030-88132-0>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword by Martin Ester

Machine learning (ML) has achieved spectacular breakthroughs in many real-life applications, including image classification, machine translation, and robotics. Deep neural networks have been particularly successful, due to their great capacity to approximate extremely complex, non-linear functions. Neural networks achieve this capacity through increasingly diverse and complex network architectures. However, the great potential comes at a hefty price: the development of machine learning models does not only require suitable training datasets and knowledge of the application domain but also deep knowledge of the machine learning methods to actually leverage their full potential.

Automated machine learning (AutoML) is a research direction that has recently emerged in response to the ever-increasing complexity of ML models and their development, aiming to automate the development process as much as possible. Goals are, in particular, to automatically tune the many hyperparameters, e.g., the number and size of layers, and to determine the most appropriate architecture of a neural network, e.g., a convolutional network for feature extraction combined with a fully connected network for classification. A related goal is meta-learning, i.e., learning to learn, which promises to reduce the effort of model development by transferring a model from a source domain to a target domain. While AutoML is still a fairly young research area, neural networks developed through AutoML have already achieved performance comparable to that of neural networks handcrafted by data scientists in some applications. Since 2015, the research community has organized the AutoML Challenge, which has provided a benchmark and much stimulation to the field. Several ML development tools, including RapidMiner and Microsoft Azure, have already implemented the features of AutoML. In conclusion, AutoML is a promising direction in ML that is expected to mature in the years to come.

This timely book by two experts in the field introduces the state-of-the-art in AutoML with a focus on it for multimedia data. Wenwu Zhu is a Professor at the Department of Computer Science at Tsinghua University and is widely recognized for his research in the areas of multimedia networking and computing as well as multimedia big data. Xin Wang is an Assistant Professor at the Department of

Computer Science at Tsinghua University. Multimedia data, including image, video, audio, and text data, is much more complex in nature than structured data such as records stored in a relational database, and multimedia has been the domain where deep neural networks have had the greatest impact. In addition to being unstructured, multimedia data is typically very large and multimodal, i.e., combines various types of multimedia data, e.g., images and text. Finally, multimedia data is often served in a streaming fashion, i.e., large amounts of data arrive rapidly and have to be processed online. I highly recommend this book to anyone who wants to understand the state-of-the-art in AutoML, in particular the special challenges and methods for multimedia data.

Professor, Simon Fraser University
Burnaby, BC, Canada

Martin Ester

Foreword by Steven C.H. Hoi

As a fundamental subset of AI techniques, machine learning has drawn popular attention from both academia and industry and made significant impact in real-world applications. This book covers two important and closely related subfields of machine learning, automated machine learning (AutoML) and meta-learning, which have been actively studied in recent years.

AutoML aims to automate the task of applying machine learning to solve a real-world problem. For example, one popular technique in AutoML is hyper-parameter optimization (HPO), which aims to automatically choose optimal hyperparameters for a learning algorithm. Another well-known AutoML technique is neural architecture search (NAS) for deep learning or deep neural network (DNN), which aims to automate the design of deep learning architectures.

Meta-learning, also known as learning to learn, aims to design a model that can learn new skills or adapt to new environments rapidly with limited training data. Meta-learning can be applied to tackle the AutoML tasks, such as HPO and NAS. In addition, meta-learning can be used for several other kinds of machine learning tasks and real-world applications, such as cold-start recommendation in multimedia and few-shot learning in computer vision and NLP.

This book provides a comprehensive understanding of AutoML and meta-learning methods and their applications. It is organized into two parts. Part I covers the subjects on the fundamentals of AutoML and meta-learning methodologies, including basics of some popular algorithms and recent advances in machine learning. Part II covers the subjects on applying AutoML and meta-learning techniques for a range of application domains, such as computer vision, natural language processing, multimedia, data mining, and recommender systems.

The authors are established AI experts and researchers with extensive experiences in investigating machine learning techniques for real-world applications. This book is strongly recommended for AI researchers, engineers, graduate students, and

any readers who are interested in learning advanced machine learning subjects in AutoML and meta-learning.

Managing Director, Salesforce Research Asia
Professor, Singapore Management University
Singapore

Steven C.H. Hoi

Foreword by Tong Zhang

I have known professor Wenwu Zhu for many years, and we have collaborated on a number of problems. Professor Zhu is a highly regarded scientist in multimedia and big data research. He has not only published many influential scientific papers but also worked on practical problems in the industrial setting.

Both automated machine learning and meta-learning are emerging topics in machine learning, which have drawn significant attention in recent years due to their many practical applications. This book presents a comprehensive overview of the recent advances in these subjects as well as their applications. The book contains two parts. Part I presents a unified view of basic concepts and many recently proposed algorithms that are scattered in the literature. This helps the readers to quickly grasp the basic concepts and algorithmic foundations. Part II contains case studies and applications that help the readers to understand how these methods can be applied to real-world problems. Although examples in this book focused on multimedia applications, the material should greatly benefit all researchers and practitioners who want to learn these advanced machine learning methods through practical examples.

Chair Professor
The Hong Kong University of Science and Technology
Hong Kong, China

Tong Zhang

Preface

This book is to disseminate and promote the recent research progress and frontier development in AutoML and meta-learning as well as their applications in computer vision, natural language processing, multimedia, and data mining-related fields, which are exciting and fast-growing research directions in the general field of machine learning. We will advocate novel, high-quality research findings and innovative solutions to the challenging problems in AutoML and meta-learning. This topic is at the core of the scope of artificial intelligence and is attractive to audience from both academia and industry.

Our efforts in writing this book is motivated by the following reasons. First of all, the topics on meta-learning and AutoML are very new emerging topics, which urgently requires a well-organized monograph on these topics. Second, several current viewpoints may treat neural architecture search (NAS) and Bayesian optimization (BO), two important techniques in AutoML, as components in meta-learning. Our book differs from them by regarding AutoML and meta-learning as two parallel tools that can enhance each other. Third, in this book, we will discuss more recent advances in AutoML and meta-learning, such as continual learning, hardware-aware architecture search, and automated graph learning. Last but not the least, this book also focuses on the applications of AutoML and meta-learning in many research fields, such as computer vision, natural language processing, and multimedia etc.

Therefore, we deeply hope that this book can benefit interested readers from both academy and industry, covering the needs from junior starters in research to senior practitioners in IT companies.

Beijing, China

Wenwu Zhu

Beijing, China
June, 2021

Xin Wang

Acknowledgments

We thank Wenpeng Zhang for participating in the NeurIPS contest on AutoML in 2018, which initiated our research on AutoML and meta-learning. Many thanks to Jie Cai, Yudong Chen, Kun Cheng, Chaoyu Guan, Jiyan Jiang, Yuhang Jiang, Xiaohan Lan, Haoyang Li, Yue Liu, Yijian Qin, Guangyao Shen, Zheng Xiong, Yitian Yuan, Zeyang Zhang, Ziwei Zhang, and Shiji Zhou from the Tsinghua University for their contributions during the writing of this book.

We would also like to express our sincere thanks to Martin Ester from the Simon Fraser University, Steven C.H. Hoi from the Salesforce Asia/Singapore Management University, and Tong Zhang from The Hong Kong University of Science and Technology for writing forewords for this book.

Xin Wang is the corresponding author.

Contents

**Part I Basics and Advances for Automated Machine Learning
and Meta-Learning**

- 1 Automated Machine Learning** 3
 - 1.1 Hyper-Parameter Optimization 4
 - 1.1.1 Model-Free Methods 6
 - 1.1.2 Bayesian Optimization 8
 - 1.1.3 Bandit Variants 12
 - 1.2 Neural Architecture Search 14
 - 1.2.1 Search Space 16
 - 1.2.2 Search Strategy 22
 - 1.2.3 Performance Estimation Strategy 47
 - 1.3 Advances on AutoML 52
 - 1.3.1 Grey-Box AutoML 52
 - 1.3.2 Hardware-Aware AutoML 54
 - 1.3.3 AutoML for Reinforcement Learning 56
 - 1.3.4 AutoML for Graph 58
 - References 61
- 2 Meta-Learning** 71
 - 2.1 Model-Based Methods 73
 - 2.1.1 Meta-Learning with Memory-Augmented Neural Networks (MANN) 74
 - 2.1.2 Extensions and Variants of MANN 75
 - 2.2 Metric-Based Methods 76
 - 2.2.1 Matching Networks 76
 - 2.2.2 Prototypical Networks 78
 - 2.2.3 Extensions and Variants of Prototypical Networks 78
 - 2.3 Optimization-Based Methods 79
 - 2.3.1 The Model Agnostic Meta-Learning (MAML) Algorithm ... 79
 - 2.3.2 Extensions and Variants of MAML 80
 - 2.4 Advances on Meta-Learning 81

- 2.4.1 Dynamic Meta-Learning 81
- 2.4.2 Multimodal Meta-Learning 85
- 2.4.3 Meta Reinforcement Learning 86
- 2.4.4 Meta-Learning for Graph..... 89
- References 91

**Part II Applications of Automated Machine Learning and
Meta-Learning to Multimedia**

- 3 Automated Machine Learning for Multimedia 97**
 - 3.1 HPO for Multimedia Search and Recommendation 98
 - 3.1.1 Multimedia Search and Recommendation 98
 - 3.1.2 Applying HPO on Multimedia Search and Recommendation 99
 - 3.2 HPO for Computer Vision 106
 - 3.2.1 HPO for Image Applications 106
 - 3.2.2 HPO for Video Applications 111
 - 3.2.3 HPO for Audio Applications 116
 - 3.3 HPO for Natural Language Processing 118
 - 3.3.1 Bayesian Optimization (BO) for Text..... 119
 - 3.3.2 Sequential Model-Based Optimization (SMBO) for Text 121
 - 3.4 NAS for Multimedia Search and Recommendation 123
 - 3.4.1 Multimedia Search with Deep Architectures 124
 - 3.4.2 Multimedia Recommendation with Deep Architectures 125
 - 3.4.3 Applying NAS on Multimedia Search and Recommendation 126
 - 3.4.4 Multimodal NAS in Search and Recommendation..... 128
 - 3.5 NAS for Computer Vision 134
 - 3.5.1 Image Classification 134
 - 3.5.2 Dense Image Prediction 140
 - 3.5.3 Image Restoration 151
 - 3.5.4 Image Generation 151
 - 3.5.5 Other Applications..... 153
 - 3.6 NAS for Natural Language Processing 155
 - 3.6.1 Language Modeling 155
 - 3.6.2 Text Classification 157
 - 3.6.3 Other NLP Tasks..... 159
 - 3.7 Other Applications of NAS 159
 - 3.7.1 NAS for Network Compression 163
 - 3.7.2 NAS for Quick Adaption on New Tasks..... 164
 - 3.7.3 NAS for Out-of-Distribution Detection..... 166
 - References 167
- 4 Meta-Learning for Multimedia 179**
 - 4.1 Meta-Learning for Multimedia Search and Recommendation 180
 - 4.1.1 Meta-Learning for Cold-Start Recommendation 181
 - 4.1.2 Meta-Learning for Algorithm Selection 187
 - 4.1.3 Meta-Learning for Incremental Product Search 189

4.2	Meta-Learning for Computer Vision	191
4.2.1	Meta-Learning for Classification in Images and Videos	191
4.2.2	Meta-Learning for Detection in Images and Videos	192
4.2.3	Meta-Learning for Image Captioning.....	193
4.2.4	Meta-Learning for Tracking in Images and Videos	193
4.2.5	Meta-Learning for Visual Question Answering	194
4.3	Meta-Learning for Natural Language Processing	205
4.3.1	Meta-Learning for NLP Classification Tasks.....	205
4.3.2	Multi-Domain Low-Resource Natural Language Generation	206
4.3.3	Knowledge Graph (KG) Completion	207
4.3.4	Other Tasks.....	208
4.3.5	Datasets for Few-Shot NLP Tasks	208
	References	209
5	Future Research Directions	217
5.1	On Hyper-Parameter Optimization	217
5.2	On Neural Architecture Search.....	218
5.3	On Meta-Learning	219
	References	220
Index		223

List of Figures

Fig. 1.1	Global search space: chain-structured search space [54]	17
Fig. 1.2	Global search space: chain-structured space with skip-connection [123]	18
Fig. 1.3	Global search space: search space with segments [108]	18
Fig. 1.4	Cell-based search space in NASNet [124]: whole architecture constructed by cells	20
Fig. 1.5	Cell-based search space in NASNet [124]: a sample of a reduction cell	21
Fig. 1.6	Cell-based search space in BlockQNN [121]: whole architecture constructed by cells	22
Fig. 1.7	Cell-based search space in BlockQNN [121]: a sample of a cell	23
Fig. 1.8	An example of super network	42
Fig. 2.1	A taxonomy of meta-learning methods	73
Fig. 2.2	The process of interaction between an agent and the environment of Meta-RL [53] and RL^2 [48]	88
Fig. 3.1	Diagram of predicting whether a news is popular	103
Fig. 3.2	The HyperNOMAD workflow	108
Fig. 3.3	Bayesian optimization used in Borgli et al.'s work [53]	109
Fig. 3.4	The system flow adopted in [53]	110
Fig. 3.5	A general biometric system for face recognition where both classification module and biometric database are replaced by the adaptive classification system	112

Fig. 3.6	The evolution of a new adaptive classification system (ACS) according to generic incremental learning scenario. New blocks of data are used by the ACS to update the classifier over time. Let D_1, D_2, \dots be blocks of learning data available at different instants in time. The ACS starts with an initial hypothesis hyp_0 which constitutes the prior knowledge of the domain. Each hypothesis hyp_{t-1} are updated to hyp_t by the ACS on the basis of the new data block D_t	112
Fig. 3.7	Complete pipeline describing the final strategy used for ConvNet 1 model	113
Fig. 3.8	Schematic diagram of the proposed CNN for object classification	114
Fig. 3.9	A general framework for multimedia search	124
Fig. 3.10	A general framework for multimedia recommendation	125
Fig. 3.11	General structure of a bi-modal fusion network. Top: a neural network with several hidden layers (grey boxes) with input \mathbf{x} , and output $\hat{\mathbf{z}}_{\mathbf{x}}$. Bottom: A second network with input \mathbf{y} , and output $\hat{\mathbf{z}}_{\mathbf{y}}$. In MFAS, finding efficient fusion schemes (yellow box and dotted lines) is the focus	129
Fig. 3.12	Two realizations of MFAS search space on a small bi-modal network. Left: network defined by $[(\gamma_1^m = 1, \gamma_1^n = 2, \gamma_1^p = 1), (\gamma_2^m = 3, \gamma_2^n = 4, \gamma_2^p = 2)]$. Right: network defined by $[(\gamma_1^m = 3, \gamma_1^n = 3, \gamma_1^p = 2)]$	130
Fig. 3.13	The flowchart of the MMnas framework, which consists of (a) unified encoder-decoder backbone and (b) task-specific heads on top the backbone for visual question answer (VQA), image-text matching (ITM), and visual grounding (VG). Note that the searched architecture shown in (a) is only a schematic example	132
Fig. 3.14	Best cell and network structure found by RandomNet. From left to right: (a) image cell, (b) audio cell, (c) fusion network connectivity. 3×3 and 5×5 indicate regular convolution with the specific kernel dimension, sep indicates depthwiseseperable convolution. Colors describe the activation type: orange: identity, yellow: Tanh, green: sigmoid, pink: ReLU	133
Fig. 3.15	Auto-generated networks on CIFAR-10(left) and ImageNet(right). The searched blocks are repeated in a predefined structure	136

Fig. 3.16	An example of a three-level hierarchical architecture representation. The bottom row shows how level-1 primitive operations $o_1^{(1)}, o_2^{(1)}, o_3^{(1)}$ are assembled into a level-2 motif $o_1^{(2)}$. The top row shows how level-2 motifs $o_1^{(2)}, o_2^{(2)}, o_3^{(2)}$ are assembled into a level-3 motif $o_1^{(3)}$	137
Fig. 3.17	An illustration of transforming a single layer to a tree-structured motif via path-level transformation operations, where they apply Net2DeeperNet operation to replace an identity mapping with a 3×3 depthwise-separable convolution in (c)	138
Fig. 3.18	Schematic diagram of top ranked DPC	141
Fig. 3.19	A general network architecture in the search space	142
Fig. 3.20	Detailed structure of the fusion node	142
Fig. 3.21	Example of the encoder-decoder auxiliary search layout	143
Fig. 3.22	left: the network level search space with $L = 12$. Gray nodes represent the fixed “stem” layers, and a path along the blue nodes represents a candidate network level architecture. Right: during the search, each cell is a densely connected structure	144
Fig. 3.23	An overview of the network structure for semantic segmentation in [155]. Each cell employs the previous two cells’ outputs as its inputs	144
Fig. 3.24	Illustration of the Graph-Guided Network Architecture Search [116]. In reduction cells, all the operations adjacent to the input nodes are of stride two. (a) The backbone network, it’s stacked by a series of independent cells. (b) The GCN-Guided Module (GGM), it propagates information between adjacent cells. α_k and α_{k-1} represent the architecture parameters for cell k and cell $k - 1$, respectively, and α'_k is the updated architecture parameters by GGM for cell k . The dotted lines indicate GGM is only utilized in the search progress	145
Fig. 3.25	The framework proposed in [102]. (a) SOARS stratifies OAR segmentation across two dimensions: distinct processing frameworks and distinct architectures where the later is executed using differentiable NAS. (b) Depicts illustrates the backbone network (P-HNN) with NAS, which allows for an automatic selection across 2D, 3D, P3D convolutions. (c) Demonstrates the NAS search space setting	145

Fig. 3.26	An illustration of the proposed C2FNAS. Each path from the left-most node to the right-most node is a candidate architecture. Each color represents one category of operations, e.g., depthwise conv, dilated conv, or 2D/3D/P3D conv which are more common in medical image area. The dotted line indicates skip-connections from encoder to decoder. The macro-level topology is determined by coarse stage search, while the micro-level operations are further selected in fine stage search	146
Fig. 3.27	An overview of the architecture search framework for detection, which can be stacked on any backbone and focuses on searching better architectures of the neck and the RCNN head of a detection system. The proposed Auto-fusion module aims at finding a better architecture for utilizing information from all feature hierarchy and conducting feature fusion for better prediction. The Auto-head module is constructed by different auto-searched cells in order to perform a better classification and bounding-box (bbox) prediction	147
Fig. 3.28	A conceptual example of the NAS-FCOS decoder. It consists of two sub networks, an FPN f and a set of prediction heads h which have shared structures. One notable difference with other FPN-based one-stage detectors is that the heads have partially shared weights. Only the last several layers of the predictions heads are tied by their weights. The number of layers to share is decided automatically by the search algorithm. Note that both FPN and head are in the actual search space, and have more layers than shown in this figure which is for the purpose of illustration only	148
Fig. 3.29	A searchable MnasFPN block. MnasFPN re-introduces the Inverted Residual Block (IRB) into the NAS-FPN head. Any path connecting an input and a new feature, as highlighted in blue dashed rectangle, resembles an IRB. MnasFPN also employs size dependent ordering (SDO) shown in black rectangle to re-order the resizing operation and the 1×1 convolution prior to feature merging. The searchable components are highlighted in red	148
Fig. 3.30	An overview of SM-NAS for detection pipeline. A two-stage coarse-to-fine searching strategy directly on detection dataset is proposed: S1: structural-level searching stage first aims to finding an efficient combination of different modules; S2: modular-level search stage then evolves each specific module and pushes forward to a faster task-specific network	149

Fig. 3.31	The framework of the FNA. Firstly, the authors select an artificially designed network as the seed network \mathcal{N}_s and expand \mathcal{N}_s to a super network \mathcal{N}_{sup} which is the representation of the search space. Then parameters of \mathcal{N}_s are remapped to \mathcal{N}_{sup} . They utilize the NAS method to start the architecture adaptation with the super network and obtain the target architecture $Arch_t$. Before parameter adaptation, the parameters of \mathcal{N}_s to $Arch_t$ are remapped. Finally, they adapt the parameters of $Arch_t$ to get the target network \mathcal{N}_t	150
Fig. 3.32	The search space of Up-Cell and Down-Cell. The architectures of both Up-Cell and Down-Cell will continuously promote each other in an adversarial manner	152
Fig. 3.33	AssembleNet with multiple intermediate streams. Darker colors of connections indicate stronger connections. At each convolutional block, multiple 2D and (2+1)D residual modules are repeated alternatingly. The network has 4 block levels (+ the stem level connected to raw data). Each convolutional block has its own output channel size (i.e., the number of filters) C and the temporal resolution r controlling the 1D temporal convolutional layers in it	153
Fig. 3.34	Illustration of the search space in [130]. Here, \otimes denotes matrix multiplication. \oplus is the element-wise summation. There are eight function modules for generating graphs. The top part is a implementation of Chebyshev polynomial. The authors also add its separate components to the graphs and let the network choose the final ones. The bottom part contains three dynamic graph modules. All the graphs are added together. The contribution of each module works as the architecture parameters. Note that there is a softmax function before the summation operation for dynamic graphs	154
Fig. 3.35	An example of a recurrent cell constructed from a tree that has two leaf nodes (base 2) and one internal node in [158]. Left: the tree that defines the computation steps to be predicted by controller. Center: an example set of predictions made by the controller for each computation step in the tree. Right: the computation graph of the recurrent cell constructed from example predictions of the controller	156

Fig. 3.36	DAG design of recurrent cells in [133]. The graph represents the entire search space while the red arrows define a model in the search space, which is decided by a controller. Here, node 1 is the input to the model whereas nodes 3 and 6 are the model's outputs	156
Fig. 3.37	An example of inter-cell and intra-cell network in [113]. h_t, x_t stands for hidden state and input at timestamp t . \hat{h}_t, \hat{x}_t stands for the intermediate feature maps	157
Fig. 3.38	Architecture composition from encoding in [138]. Each block produces a new hidden state that is added to the pool of hidden states subsequent blocks can select as branch inputs. Each encoder has 6 unique blocks per cell and each decoder has 8 unique blocks per cell. Each cell is repeated <i>number of cells</i> times	158
Fig. 3.39	General structure of a bi-modal fusion network in [131]. Top: a neural network with several hidden layers (grey boxes) with input \mathbf{x} , and output $\hat{\mathbf{z}}_{\mathbf{x}}$. Bottom: a second network with input \mathbf{y} , and output $\hat{\mathbf{z}}_{\mathbf{y}}$	160
Fig. 3.40	Continual architecture search (CAS): blue edges (weight parameters) are shared, newly learned edges are represented with red edges	160
Fig. 3.41	Multi-task cell structure learning using joint rewards from n datasets	161
Fig. 3.42	An overview of the joint design methodology in [145]. The serial number represents the order of the steps. The model first trains an accuracy predictor for the full precision NN, then incrementally trains an accuracy predictor for the quantized NN (predictor-transfer). Finally, evolutionary search is performed to find the specialized NN architecture with quantization policy that fits hardware constraints	162
Fig. 3.43	Macro architectures for image translation (top) and super resolution (bottom) designed in [96]	164
Fig. 3.44	Architecture fine-tuning with NETTAILOR [125]. Pre-trained blocks shown in gray, task-specific in yellow. Left: pre-trained CNN augmented with low-complexity blocks that introduce skip connections. Center: optimization prunes blocks of poor trade-off between complexity and impact on recognition. Right: the final network is a combination of pre-trained and task-specific blocks	166
Fig. 3.45	Augmentation of pre-trained block at one layer G_l with multiple proxy layers A_p^l in [125], \mathbf{x}_i represents the network activation after layer i	166

Fig. 3.46	Search space in [73] of a single block design following [109]	167
Fig. 4.1	A general framework of bilevel optimization for cold-start user setting	183
Fig. 4.2	The SEATTLE framework [12], which applies metric-based meta-learning to cold-start item recommendation	187
Fig. 4.3	The framework of traditional meta-learning for recommendation algorithm selection [3]	188
Fig. 4.4	Examples from MS-COCO dataset	196
Fig. 4.5	Examples from visual genome dataset	196
Fig. 4.6	Examples from VQA dataset	197
Fig. 4.7	Examples from TGIF-QA dataset	198
Fig. 4.8	Examples from CLEVR-Humans dataset, with unrestricted words and phrases annotations	199
Fig. 4.9	Examples from VCR dataset	200
Fig. 4.10	Training process of MetaNMT	207

List of Tables

Table 1.1	Decomposition of GNN layer	60
Table 3.1	Hyper-parameters of Bias SGD Algorithm (Chan et al. [4])	100
Table 3.2	Hyper-parameters of ALS Algorithm (Chan et al. [4])	100
Table 3.3	Comparison of RMSE on the hold out (test) ratings after tuning ALS algorithm [11]	104
Table 3.4	Choices of the optimizers and the corresponding hyper-parameters [43]	108
Table 3.5	Comparison of algorithms in terms of time and score on the training set, taken from [50]	116
Table 3.6	List of hyper-parameters used in SVM and boosted regression trees [22]	119
Table 3.7	Hyper-parameters and the corresponding values used in SVM and boosted regression trees [22]	120
Table 3.8	Summary of three current papers in multimodal NAS	129
Table 3.9	Meaning of 5 dimensions in a description vector	135
Table 4.1	Literature overview of meta-learning for cold-start recommendation	182
Table 4.2	Bilevel optimization meta-learning for cold start recommendation	185