

Medium Story Link: <https://adityatushar-wadnerkar.medium.com/design-neural-networks-with-meta-learning-9be9a4e12ded>

Slide Share Link: <https://www.slideshare.net/AdityaWadnerkar1/design-neural-networks-with-meta-learning-239803054>

Video Link:

Paper Link: <https://arxiv.org/pdf/2004.05439.pdf>

Short Story

Design Neural Networks with Meta-Learning

We have seen meta-learning technology emerging in recent times. The meta-learning approach aims to learn and improve learning algorithm unlike the traditional approach where it a fixed algorithm is used for solving a task in AI.

We will discuss...

The aim of this article is to provide a short survey of different meta-learning techniques. Fundamental issues of generalization along with contemporary landscape of meta-learning are discussed in the survey. We will discuss the functioning of meta-learning techniques and how they differ from other fields like transfer learning, hyperparameter optimization and multi-task learning.

So, what is Meta-Learning?

In simple words, “Learning to Learn”



According to the author of the paper, “Meta Learning in Neural Networks: A Survey”, meta-learning methodologies learn from the experience of series of multiple tasks or episodes of learning. It tackles many traditional deep learning challenges including computation bottlenecks and data bottlenecks.

Have you heard of these fields...?

Transfer Learning extracts features from one task and learns from experience of the source task to transfer features to destination task by improving learning.

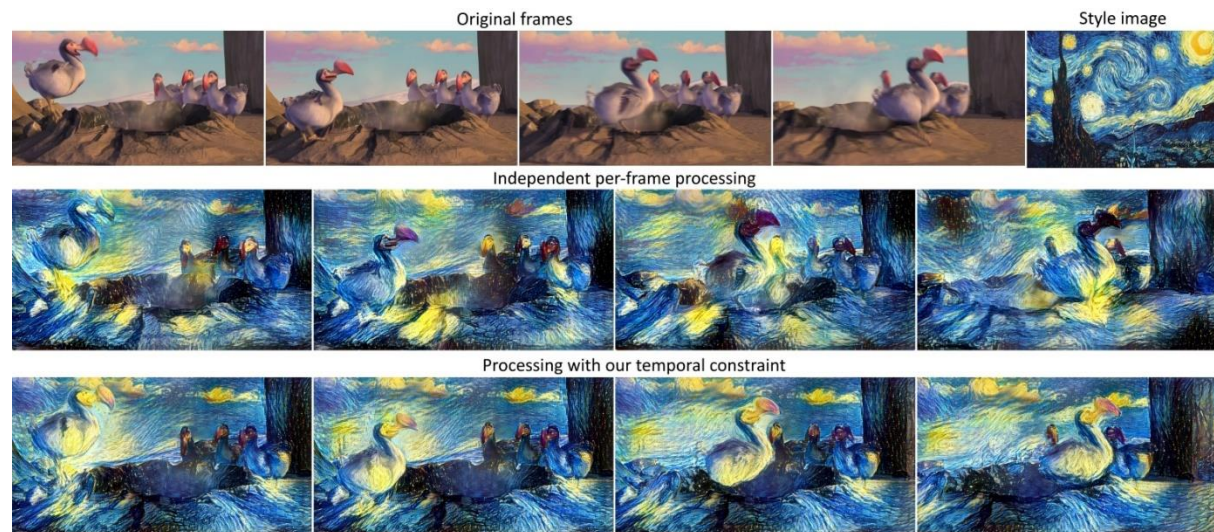


Fig. Transfer Learning for imposing style transfer from one image to other

Domain Adaptation and Domain Generalisation is a different variant attempting to address the problem. It uses sparse data from target and adapts to the model which is source trained.

Continual Learning learns a long sequence of tasks and accelerates learning of new tasks and while saving the old tasks which are drawn from non-stationary distribution.

Multi-Task Learning has a goal to learn several tasks at the same time. It differs from meta-learning as it has fixed number of different tasks to solve while meta-learning provides solutions for unforeseen tasks.

Hyperparameter Optimization includes gradient-based hyperparameter learning but excludes approaches like random search.

Hierarchical Bayesian Models provide modelling instead of algorithmic framework for getting into the meta-learning process.

AutoML automates learning process like selecting features, hyperparameter tuning data cleaning and preparation.

New Taxonomies?

The common breakdown of meta-learning methods: Black box, Non-parametric and Optimization. These make it difficult to form connections in understanding the various latest meta-learning frameworks. So, let's break it down in a new way.

Based on three independent axes...

Meta-Representation("What?") – representation of meta-knowledge (w)

Meta-Optimizer("How?") – optimizer for outer level (gradient-decent, etc)

Meta-Objective("Why?") – determine the goal of meta-learning (L^{meta})

Each axis has taxonomy that represents latest meta-learning scenario.

What are the methodologies?

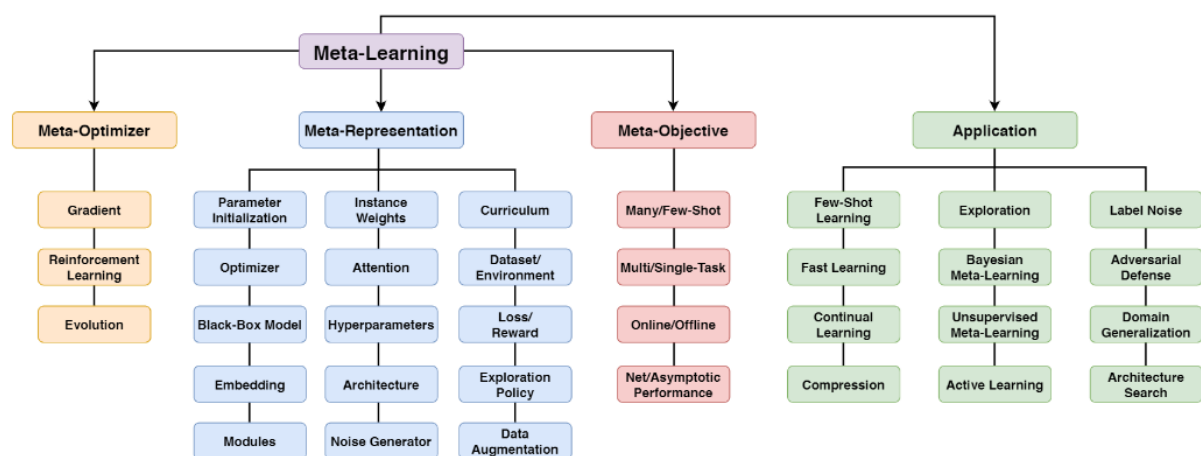


Fig. 1. Overview of the meta-learning landscape including algorithm design (meta-optimizer, meta-representation, meta-objective), and applications.

Meta-Optimizer

Gradient descent needs computing derivatives connected through model parameter. The methods are highly efficient but carry challenges such as cases with more than one steps are used for inner optimization have challenge differentiating via long computational graphs.

Reinforcement Learning methods are used in cases where we have non-differentiable meta-objective or steps for optimizing outer objective.

Evolution optimize meta-objective similar to Reinforcement Learning, but the performance isn't dependent on reward sparsity or length.

Meta-Representation	Meta-Optimizer		
	Gradient	RL	Evolution
Initial Condition	[19], [75], [84], [99], [152], [152]–[154]	[155]–[157] [19], [61], [62]	[158], [159]
Optimizer	[78], [91] [41], [75], [103], [160], [161]	[77], [89]	
Hyperparam	[25]	[162], [163]	[159] [164]
Black-box model	[40], [82], [165], [166]	[22], [106], [109]	
Metric	[20], [86], [87]		
Loss/Reward	[44], [92] [118]	[117] [113], [167]	[115] [23] [164]
Architecture	[26] [125]	[28]	[27]
Exploration Policy		[24], [168]–[172]	
Dataset/Environment	[144] [146]	[149]	[150]
Instance Weights	[142]		
Feature/Metric	[20], [86]–[88]		
Data Augmentation/Noise Generator	[134] [173] [174]	[133]	[135]
Modules	[130], [131]		
Annotation Policy	[175], [176]	[177]	

TABLE 1

Fig: The colour code for the data representation follows red for sample efficiency, green for learning speed, purple for asymptotic performance, blue for cross-domain.

Meta-Representation

Parameter Initialization is used for problems involving few-short learning which can be learned using examples without over-fitting them. The issue with the method is that outer optimization must solve for all the parameters equal to that of inner optimization.

Optimizer approach is focussed to learn inner optimizer where a function is trained by taking input optimization states and producing an optimization step for each iteration of base learning.

Black-Box Models don't rely on gradient iterative optimization. Instead, they train the learners providing mapping of feed-forward directly to parameters for classifying test cases from the support set.

Embedding functions compare the similarity between support and query instances for learning embedding network transforming into representation that suits for recognition from the raw input data.

Losses and Auxiliary Tasks build a narrow neural network takes input to losses and give scalars which are treated by inner optimizer as loss. These learn through unlabelled instances as well.

Attention modules improved the performance of generalization and interpretability for them. They used as comparators and feature extractors for preventing catastrophic forgetting that we witness in few-short learning. They are recently made capable summarizing textual data.

Curriculum learning, minibatch selection and sample weights are used for automation of learning of a curriculum, which is sequence and concepts of data for producing better performance.

Meta-Objective and Episode Design

Many and Few-Shot Episode Design is selected based on the goal if it is for improving few or many shot performance. The episodes of inner loop learning may be defined accordingly as few or many examples per task.

Multi and Single Task depending on the goal whether it is to tune learner for solving a task belonging to a family of tasks or simply solve one specific task in a better way.

Online and Offline need to be compared when there is not a single set of learning operations to amortize over. In some cases, they have been able to perform meta-optimization online.

Fast Adaptation and Asymptotic Performance is when it is calculated for the sum of validation loss. The training boosts learning in base task after each inner optimization step.

Applications

The technology is trending for making cutting-edge innovations disrupting the industry. We list some of them:

Computer Vision and Graphics: used for classification, object detection, landmark prediction, object segmentation, image generation, video synthesis and density estimation tasks.

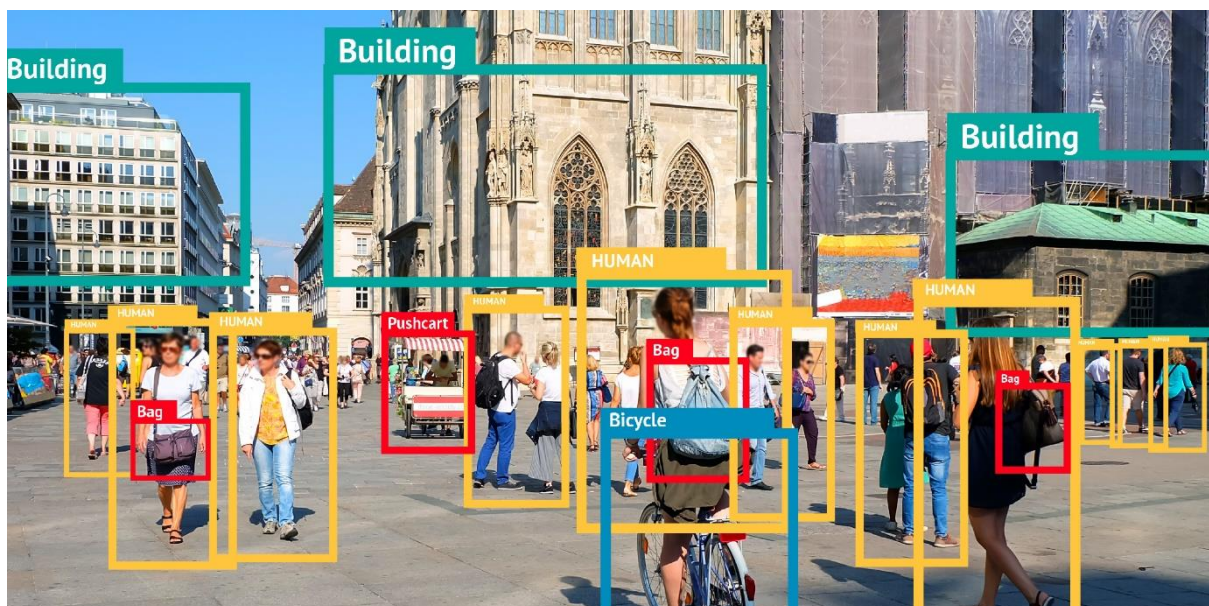


Fig. Computer vision for object detection

Meta Reinforcement Learning and Robotics: used for exploration, optimization, online Meta-RL, on and off-policy Meta-RL tasks.

Environment Learning and Sim2Real: it generates simulation matching the real-world for training a model.

Neural Architecture Search: performs outer optimization search for best validation performance while inner optimization trains the network as per the defined architecture.

Active Learning: finds the best data subset for annotating which maximizes the downstream learning performance with least number of annotations.

Language Modelling: use few-shot learning form modelling language and finding relationship between words and extracting context.

Speech Recognition: meta learning is using few-shot for automatic speech recognition and quickly learning to train for optimizing the model for each speaker individually.

Social Good: used in drug discovery, medical classification and addressing major problems like cancer detection.

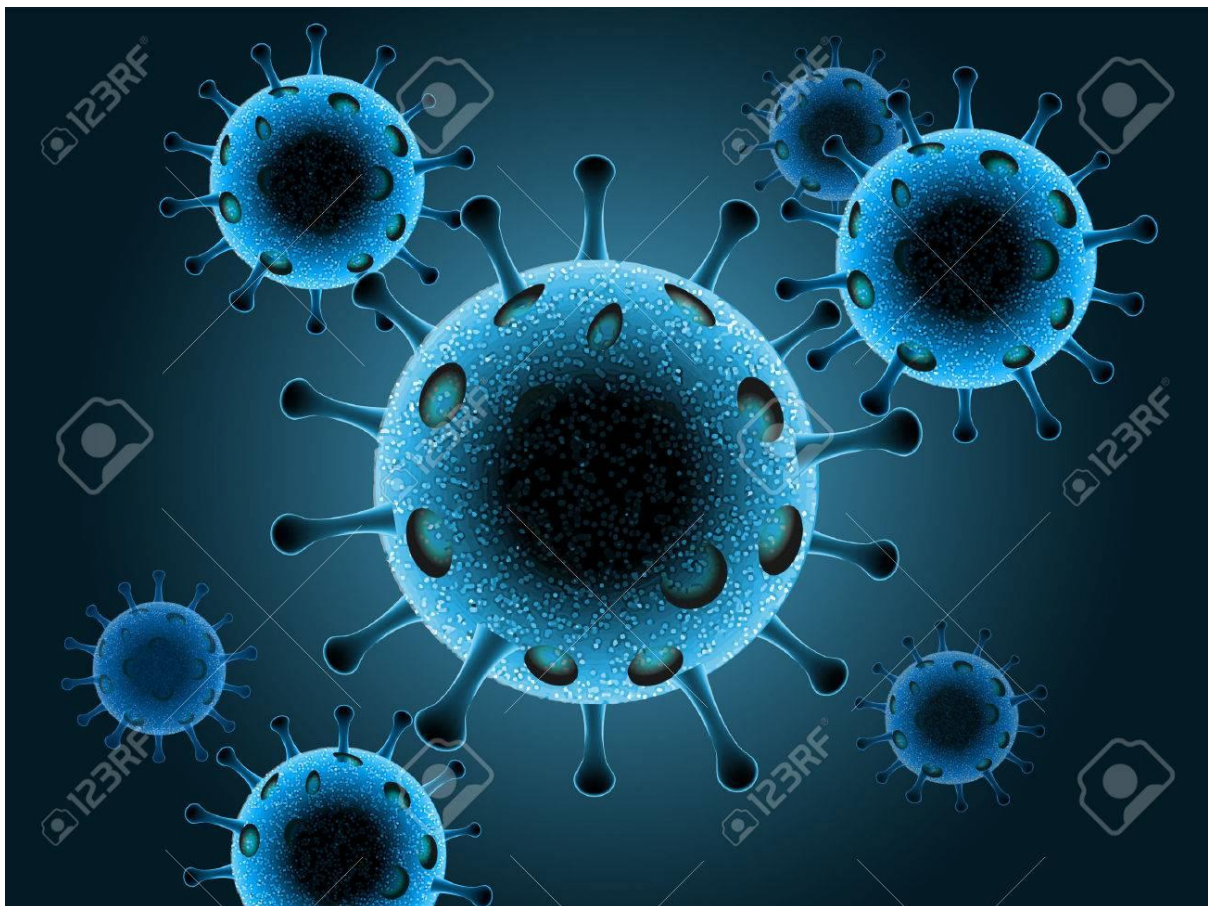


Fig. Meta-Learning for drug discovery

Inferences and Conclusions

Meta learning has picked up a rapid pace for addressing the real-world problems. The latest advancements show that it is going to be one of the most revolutionary technologies of the decade. The article summarized the insights from the paper, “Meta-Learning in Neural Networks: A Survey”, published by Timothy Hospedales, Antreas Antoniou, Paul Micaelli and Amos Storkey. There are some challenges which relate to meta-generalization,

multimodality of task distribution, task families, computational costs, cross-modal transfer and heterogenous tasks to be taken into consideration for future development.

At last, there are some of the comments on the paper submission:

Paper quality: the paper described all the methods in detail, discussed current use of each and the challenges associated.

Critique/ Suggestions on improvement: they could have explained with each while discussing the pro and cons accurately in each instance with examples and more figures which could have made it more appealing to comprehend.

References:

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, Amos Storkey, “Meta-Learning in Neural Networks: A Survey”, 7 Nov 2020, version, v2, <https://arxiv.org/abs/2004.05439>

Isha Salian, “How Deep Learning Can Paint Videos in the Style of Art’s Great Masters”, May 25, 2016, <https://blogs.nvidia.com/blog/2016/05/25/deep-learning-paints-videos/>

J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pretraining Of Deep Bidirectional Transformers For Language Understanding,” in ACL, 2019.

Wikipedia, “Meta Learning (computer science)”, [https://en.wikipedia.org/wiki/Meta_learning_\(computer_science\)](https://en.wikipedia.org/wiki/Meta_learning_(computer_science))