Meta Learning and Meta RL

Di Wu
McGill & MILA

4th April 2019
Outline

1. Introduction to Meta Learning
2. Overview of Meta Learning Approach
3. Meta Reinforcement Learning
4. General Discussion: Related Problems & Research Opportunities
Motivation for Meta Learning

- Deep learning models can perform well with huge amount of data
Motivation for Meta Learning

- Deep learning models struggle when
  - Training data is limited
  - Need to adapt fast to changes in the task

- Humans can generalize well with very small amount of data
Objective of Meta Learning

• Human can learn fast since they never learn from scratch
  ◦ Samples
  ◦ Models
  ◦ Representation
  ◦ Learning to learn

• Learning to learn/Meta Learning: Train a model on several learning tasks to solve new learning tasks with a small number of training samples.
Definition of Meta Learning

- Generalization across **tasks** instead of **data points**. Task Level!

- What is Meta Learning / Learning to Learn?
  - Go beyond train from samples from a single distribution.
  - Distribution over tasks, so model has to “learn to learn” when a new task is presented.

“... a system that improves or discovers a learning algorithm”
Hochreiter et al, ‘01
What is Meta Knowledge

Anything that is determined before learning process.

- Parameter initialization
- Loss function
- Network structure
- Hyper parameters: batch size, learning rate..
- Optimization algorithm
- ...

Di Wu  McGill & MILA
Meta Learning  Comp767
4th April 2019  7 / 60
Formalization of Meta Learning

- We have a distribution of tasks $\mathcal{T} = P(\mathcal{T}_i)$
- $\mathcal{T}_i$ is episodic and defined by: input $x_t$, output $a_t$, loss function $\mathcal{L}_i(x_t, a_t)$, and an episodic length $H_i$
- Meta learner models distribution $\pi(a_t | x_1, ..., x_t; \theta)$ and the objective is to minimize the **Meta Loss** with respect to $\theta$:

$$\min_{\theta} \mathbb{E}_{\mathcal{T}_i \sim T} \left[ \sum_{t=0}^{H_t} \mathcal{L}_i(x_t, a_t) \right]$$
Important Concepts for Meta Learning

- **Meta Training:** Optimizing the meta loss on sampled tasks.
  - Sample set, query set.

- **Meta Testing:** Evaluated on unseen tasks from the same task distributions.
  - Support set, test set.
Overview of Meta Learning Approach

Outline

1. Introduction to Meta Learning
2. Overview of Meta Learning Approach
3. Meta Reinforcement Learning
4. General Discussion: Related Problems & Research Opportunities
Early Works on Meta Learning

- Evolutionary principles in self-referential learning [Schmidhuber, 1987], applying generic programming to itself to evolve better genetic programming algorithm.

- Learning to learn [Thrun and Pratt, 1998], a discussion on learning to learn applications.

- ...

Not a new learning paradigm. Paid more attention since 2016.
Approaches for Recent Meta Learning Works

- **Gradient Based**: MAML [Finn et al., 2017], Reptile [Nichol and Schulman, 2018].
  - Learn a model initialization, easy to fine tune
- **Metric Based**: Matching Networks [Vinyals et al., 2016], Prototypical Networks [Snell et al., 2017].
  - Learn an embedding function for non-parametric method
- **RNN Memory Based**: MANN [Santoro et al., 2016], Learning to reinforcement learn [Wang et al., 2016], RL$^2$ [Duan et al., 2016].
  - Store meta knowledge in RNN hidden states or external memory.
Meta Supervised Learning

Classification Data Set: Omniglot and MinImagenet.
- Omniglot (Transpose of MNIST) consists of 20 instances of 1623 characters.
- MinImagenet: a subset of ImageNet dataset, 100 classes, each with 600 instances.

N-way, K-shot Classification:
- K samples for each class of N classes.
Meta Learning and One-shot (Few-shot) Learning

- Firstly appeared in computer vision [Fe-Fei et al., 2003].

- One-shot (Few-shot) learning problem: Estimate models of categories from very few, one in the limit, training examples.

- Suitable for meta learning algorithms.
One Shot Learning Approaches

Approaches for One Shot Learning

- Directly supervised learning-based approaches, e.g., Non-parametric Methods
- Data Augmentation Methods
- Transfer Learning
- Meta Learning
Motivation for Metric-based Methods

Matching Networks [2016 NeurIPS]
- Deep learning model does not work well with small amount of data
  - Huge amount of parameters, overfitting
- Non-parametric methods: Allow novel examples be rapidly assimilated.
- Metric based methods: Combine merits of both.
  - Easy to recognize with non-parametric methods (nearest neighborhood) in the embedding space.
Model Architecture for Matching Network
Proposed Model of Matching Network

- $x_i$ and $y_i$ are the inputs and corresponding labels from the support set $S$

- Attention mechanism/nearest neighbor: softmax over cosine distance $c$: $a(\hat{x}, x_i) = \frac{e^{c(f(\hat{x}), g(x_i))}}{\sum_{j=1}^{K} e^{c(f(\hat{x}), g(x_j))}}$
Contributions

- Matching Network model: Non parametric method with parametric neural networks
- Training philosophy: Training and testing should match (N-way, K-shot)
Motivation for Prototypical Networks

Prototypical Networks [2017 NeurIPS]

- There exists one single prototype representation for each class in the embedding space.

- Consider euclidean distance
Demo for Prototypical Networks

Prototype: Mean of all the samples of the same class.
Model of Prototypical Networks

- Class distribution computed over prototypes in embedding spaces

\[
p_\Phi(y = k \mid x) = \frac{\exp(-d(f_\Phi(x), c_k))}{\sum_{k'} \exp(-d(f_\Phi(x), c_{k'}))}
\]
Relationship with Matching Network

- Equivalent when $k = 1$
- Weighted Nearest Neighbor Method
- Euclidean distance performs better than cosine distance
Motivation for Relation Network

Relation Networks [2018 CVPR]

- Whether we should keep the metric fixed?
- Whether we can jointly learn the metric as well as the embedding?
Overview for Relation Network

- Embedding Module
  - Feature maps concatenation
  - Relation module
    - Relation score
    - One-hot vector

Di Wu McGill & MILA
Meta Learning Comp767
4th April 2019 25 / 60
Relation score $r_{i,j}$ for the relation between query input $x_i$ and training sample set example $x_j$

$$r_{i,j} = g_{\phi}(C(f_{\psi}(x_i), f_{\psi}(x_j))), \quad i = 1, 2, \ldots, C$$
Objective function

- One-shot vs K-shot: concatenating all the features of k samples.
- Treated as a regression problem.

\[
\varphi, \phi \leftarrow \arg\min_{\varphi, \phi} \sum_{i=1}^{m} \sum_{j=1}^{n} (r_{i,j} - 1(y_i == y_j))^2
\]
Why It Works

- Nonlinear in embedding space is not enough
- Jointly learning provide better performance
MAML: Model Agnostic Meta Learning

- Basic ideas: To learn a good model initialization.
- Agnostic: be applied to any models trained with gradient descent
- Learn a transferable internal representation to make the models to be easy and fast to fine-tune (maximize sensitivity).
MAML: Model Agnostic Meta Learning

Optimizing for a representation $\theta$ that can quickly adapt to new tasks.
MAML: Model Agnostic Meta Learning

Supervised learning: \( f_\theta \)
- Squared error for regression
- Cross entropy for classification

Algorithm 1 Model-Agnostic Meta-Learning

1. randomly initialize \( \theta \)
2. while not done do
3. Sample batch of tasks \( \mathcal{T}_i \sim p(\mathcal{T}) \)
4. for all \( \mathcal{T}_i \) do
5. Evaluate \( \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \) with respect to \( K \) examples
6. Compute adapted parameters with gradient descent: \( \theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \)
7. end for
8. Update \( \theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i}) \)
9. end while
MAML: Model Agnostic Meta Learning

Gradient over Meta Loss

$$\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'}^i) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})})$$
MAML: Model Agnostic Meta Learning

MAML for supervised learning

---

**Algorithm 2 MAML for Few-Shot Supervised Learning**

1. **Require**: \( p(T) \): distribution over tasks
2. **Require**: \( \alpha, \beta \): step size hyperparameters
3. randomly initialize \( \theta \)
4. while not done do
   5. Sample batch of tasks \( T_i \sim p(T) \)
   6. for all \( T_i \) do
      7. Sample \( K \) datapoints \( D = \{x^{(j)}, y^{(j)}\} \) from \( T_i \)
      8. Evaluate \( \nabla_\theta \mathcal{L}_{T_i}(f_\theta) \) using \( D \) and \( \mathcal{L}_{T_i} \) in Equation (2) or (3)
      9. Compute adapted parameters with gradient descent: \( \theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{T_i}(f_\theta) \)
      10. Sample datapoints \( D'_i = \{x^{(j)}, y^{(j)}\} \) from \( T_i \) for the meta-update
   11. end for
12. Update \( \theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta'_i}) \) using each \( D'_i \) and \( \mathcal{L}_{T_i} \) in Equation 2 or 3
13. end while
## Results on Omniglot

<table>
<thead>
<tr>
<th>Model</th>
<th>Fine Tune</th>
<th>5-way Acc.</th>
<th>20-way Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
</tr>
<tr>
<td><strong>MANN [32]</strong></td>
<td>N</td>
<td>82.8%</td>
<td>94.9%</td>
</tr>
<tr>
<td><strong>Convolutional Siamese Nets [20]</strong></td>
<td>N</td>
<td>96.7%</td>
<td>98.4%</td>
</tr>
<tr>
<td><strong>Convolutional Siamese Nets [20]</strong></td>
<td>Y</td>
<td>97.3%</td>
<td>98.4%</td>
</tr>
<tr>
<td><strong>Matching Nets [39]</strong></td>
<td>N</td>
<td>98.1%</td>
<td>98.9%</td>
</tr>
<tr>
<td><strong>Matching Nets [39]</strong></td>
<td>Y</td>
<td>97.9%</td>
<td>98.7%</td>
</tr>
<tr>
<td><strong>Siamese Nets with Memory [18]</strong></td>
<td>N</td>
<td>98.4%</td>
<td>99.6%</td>
</tr>
<tr>
<td><strong>Neural Statistician [8]</strong></td>
<td>N</td>
<td>98.1%</td>
<td>99.5%</td>
</tr>
<tr>
<td><strong>Meta Nets [27]</strong></td>
<td>N</td>
<td>99.0%</td>
<td>-</td>
</tr>
<tr>
<td><strong>Prototypical Nets [36]</strong></td>
<td>N</td>
<td>98.8%</td>
<td>99.7%</td>
</tr>
<tr>
<td><strong>MAML [10]</strong></td>
<td>Y</td>
<td>98.7 ± 0.4%</td>
<td>99.9 ± 0.1%</td>
</tr>
<tr>
<td><strong>Relation Net</strong></td>
<td>N</td>
<td>99.6 ± 0.2%</td>
<td>99.8 ± 0.1%</td>
</tr>
</tbody>
</table>
## Experimental Results on MiniImagenet

### Results on MiniImagenet

<table>
<thead>
<tr>
<th>Model</th>
<th>FT</th>
<th>5-way Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1-shot</td>
</tr>
<tr>
<td>Matching Nets [39]</td>
<td>N</td>
<td>43.56 ± 0.84%</td>
</tr>
<tr>
<td>Meta Nets [27]</td>
<td>N</td>
<td>49.21 ± 0.96%</td>
</tr>
<tr>
<td>Meta-Learn LSTM [29]</td>
<td>N</td>
<td>43.44 ± 0.77%</td>
</tr>
<tr>
<td>MAML [10]</td>
<td>Y</td>
<td>48.70 ± 1.84%</td>
</tr>
<tr>
<td>Prototypical Nets [36]</td>
<td>N</td>
<td>49.42 ± 0.78%</td>
</tr>
<tr>
<td>Relation Net</td>
<td>N</td>
<td><strong>50.44 ± 0.82%</strong></td>
</tr>
</tbody>
</table>
Outline

1. Introduction to Meta Learning
2. Overview of Meta Learning Approach
3. Meta Reinforcement Learning
4. General Discussion: Related Problems & Research Opportunities
Objective:
Enable an agent to quickly learn a policy for a new task with a small amount of experiences/interactions.
Reinforcement Learning Tasks

Differences compared with meta supervised learning
- Loss function
- How data is gathered and presented

Changes for RL tasks
- Achieve a new goal in the same environment.
- Achieve the same goal in a different environment.
Meta Knowledge for Reinforcement Learning

RL meta knowledge
- Reward function: $\gamma$, $\lambda$
- Exploration strategy
- Model initialization...

Suitable approaches
- Gradient based methods
- RNN memory based methods
MAML for Reinforcement Learning

Loss function: Negative of the reward function $R$

```
Algorithm 3 MAML for Reinforcement Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: $\alpha, \beta$: step size hyperparameters

1: randomly initialize $\theta$
2: while not done do
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: for all $\mathcal{T}_i$ do
5: Sample $K$ trajectories $\mathcal{D} = \{(x_1, a_1, \ldots x_H)\}$ using $f_\theta$ in $\mathcal{T}_i$
6: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
7: Compute adapted parameters with gradient descent: $\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
8: Sample trajectories $\mathcal{D}'_i = \{(x_1, a_1, \ldots x_H)\}$ using $f_{\theta'_i}$ in $\mathcal{T}_i$
9: end for
10: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$ using each $\mathcal{D}'_i$
11: end while
```
MAML for Reinforcement Learning

Results for 2d navigation tasks.

- Goal position is randomly chosen within a unit square.
Experimental Results

Results for half-cheetah and ant locomotion tasks.

- The goal of velocity or direction is chosen randomly for each task.
Representative Works

- **RNN Based**
- *Learning to reinforcement learn* [Wang et al., 2016]
- *RL2: Fast Reinforcement Learning via Slow Reinforcement Learning* [Duan et al., 2016]
- *A simple neural attentive meta-learner* [Mishra et al., 2017]
- **Initialization**
- *Learn the model initialization for fast adaptation* [Finn et al., 2017].
- **Hierarchial RL**
- *Meta learning shared hierarchies* [Frans et al., 2017]
Representative Works

**Exploration**
- *Meta-reinforcement learning of structured exploration strategies* [Gupta et al., 2018]
- *Some considerations on learning to explore via meta-reinforcement learning* [Stadie et al., 2018]
- *Learning to explore with meta-policy gradient* [Xu et al., 2018a]

**Reward Function**
- *Meta-gradient reinforcement learning* [Xu et al., 2018b]
- *Evolved policy gradients* [Houthooft et al., 2018]

**Credit Assignment**
- *ProMP: Proximal Meta-Policy Search* [Rothfuss et al., 2018]
Representative Works

- **Model Based RL**
  - *Learning to adapt: Meta-learning for model-based control* [Clavera et al., 2018]
  - *Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning* [Nagabandi et al., 2018a]
  - *Deep Online Learning via Meta-Learning: Continual Adaptation for Model-Based RL* [Nagabandi et al., 2018b]

- **Inverse RL**
  - *Learning a prior over intent via meta-inverse reinforcement learning* [Xu et al., 2018a]
Outline

1. Introduction to Meta Learning
2. Overview of Meta Learning Approach
3. Meta Reinforcement Learning
4. General Discussion: Related Problems & Research Opportunities
Related Learning Paradigms

- **Knowledge Reuse via Prior Learning**
  - *Transfer Learning*
  - *Multi-task Learning*
  - *Lifelong & Continual Learning*
Transfer Learning

- Transfer learning: Leverage the information from the source domain(s) to help learning in the target domain.
- Meta Learning: A higher level knowledge transfer.
- Focus on learning fast (as well as accumulated performance).
Multi-task Learning

- Multi-task Learning: Optimize learning/performance across all tasks through shared knowledge.
- Multi-task Learning highly dependent on the model.
- Meta Learning: A kind of historical multi-task.

Example for image classification.
Lifelong Learning and Continual Learning

- **Lifelong Learning**: Leverage the relevant knowledge gained in the past N-1 tasks to help learning for the Nth task.
- **Continual Learning**: Similar concept and more focused on catastrophic forgetting for neural networks.
- **Online version and batch version.**
Potential Research Directions

Algorithms

- Integrate different Meta Learning Approaches
- Limitations of Current Meta Learning Approaches
- Meta RL: Exploration, Learning Rate, HRL, Model Based
- Generative Models
- Continual Learning
Potential Research Opportunities

**Applications** Go beyond Omniglot and MiniImageNet

- *Low Resource NLP: Few-shot QA*
- *Few-shot Speech to Text*
- *Few-shot Generation: Missing Data*
- *Fast Adaptation in Real-world Control Problems*
Q&A
Reference I

Learning to adapt: Meta-learning for model-based control.

RL2: Fast reinforcement learning via slow reinforcement learning.

Fe-Fei, L. et al. (2003).
A bayesian approach to unsupervised one-shot learning of object categories.
In Proceedings Ninth IEEE International Conference on Computer Vision, pages 1134–1141. IEEE.
Reference II


Reference III

Evolved policy gradients.

A simple neural attentive meta-learner.

Learning to adapt in dynamic, real-world environments through meta-reinforcement learning.


One-shot learning with memory-augmented neural networks.


Evolutionary principles in self-referential learning.


Prototypical networks for few-shot learning.

Some considerations on learning to explore via meta-reinforcement learning.

Learning to learn.

Matching networks for one shot learning.
In Advances in neural information processing systems, pages 3630–3638.
Reference VII

