Meta-Learning

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks
Chelsea Finn, Pieter Abbeel, Sergey Levine. ICML 2017

RL2: Fast Reinforcement Learning via Slow Reinforcement Learning
Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, Pieter Abbeel. ICLR 2017

Presented by Chen Jinfan
[Meta-Learning is to tell] agents to learn how to learn new tasks faster by reusing previous experience, rather than considering each new task in isolation.

—Chelsea Finn
Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks
Intuition

• Maximising the ‘sensitivity’ of the loss function of tasks w.r.t parameters

• By pre-training parameters for all tasks

• Sensitivity is high if small local changes lead to large improvement for tasks

Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation $\theta$ that can quickly adapt to new tasks.
Algorithm

- The parameters after gradient decent updates on task $i$
  \[ \theta_i' = \theta - \alpha \nabla_{\theta} \mathcal{L}_{T_i}(f_\theta). \]

- Our objective function (for a distribution of tasks) is
  \[
  \min_{\theta} \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta_i'}) = \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_\theta - \alpha \nabla_{\theta} \mathcal{L}_{T_i}(f_\theta))
  \]

- So one gradient update w.r.t. our objective is
  \[
  \theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta_i'})
  \]
Regression Experiment

- Sinusoid Function with amplitude in $[0.1, 0.5]$ and phase in $[0, \pi]$

- A model of 2 layers each with size 40 and ReLu-activation

- Compared with ground truth and model pre-trained on same metadata
Regression Experiment

**Pretrained, $K=5$, step size=0.01**

- Solid red line: ground truth
- Dashed blue line: 1 grad step
- Dotted black line: 10 grad steps

**Pretrained, $K=10$, step size=0.02**

- Solid red line: ground truth
- Dashed blue line: 1 grad step
- Dotted black line: 10 grad steps

**MAML, $K=5$**

- Solid red line: ground truth
- Dashed blue line: 1 grad step
- Solid black line: 10 grad steps

**MAML, $K=10$**

- Solid red line: ground truth
- Dashed blue line: 1 grad step
- Solid black line: 10 grad steps

Legend:
- ▲ ▲ ▲: used for grad
- ◀ ◀ ◀: pre-update
Regression Experiment

$k$-shot regression, $k=10$

Mean squared error vs number of gradient steps for different methods:
- **MAML (ours)**
- **pretrained, step=0.02**
- **oracle**

The graph shows a decrease in mean squared error as the number of gradient steps increases.
RL Experiment

- Continuous control as proposed in Duan et al. 2016
- 2 hidden layers of size 100 with ReLu activation
- TRPO as metaoptimizer and vanilla policy gradient as actual update
- Compared with ground truth and model pre-trained on same metadata
RL Experiment

ant, goal velocity

ant, forward/backward

MAML (ours)
pretrained
random
oracle

half-cheetah, goal velocity

half-cheetah, forward/backward

number of gradient steps

number of gradient steps
RL Experiment

More videos on: https://sites.google.com/view/maml
Wrap up MAML

- Model-agnostic: compatible with any gradient trained model
- Flexible: take advantage of any amount of data with any number of gradient steps
- Simple: No additional parameters needed
- Disadvantage: need to compute higher order derivatives during meta-training
RL2
RNN
General Architecture
Implementation

- RL problems seen as MDPs or POMDPs
- RNN implemented by GRU network
- First-order TRPO as training algorithm
- GAE to further reduce variance
Multi-Armed Bandits
Multi-Armed Bandits

<table>
<thead>
<tr>
<th>Setup</th>
<th>Random</th>
<th>Gittins</th>
<th>TS</th>
<th>OTS</th>
<th>UCB1</th>
<th>$\epsilon$-Greedy</th>
<th>Greedy</th>
<th>$RL^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 10, k = 5$</td>
<td>5.0</td>
<td>6.6</td>
<td>5.7</td>
<td>6.5</td>
<td>6.7</td>
<td>6.6</td>
<td>6.6</td>
<td>6.7</td>
</tr>
<tr>
<td>$n = 10, k = 10$</td>
<td>5.0</td>
<td>6.6</td>
<td>5.5</td>
<td>6.2</td>
<td>6.7</td>
<td>6.6</td>
<td>6.6</td>
<td>6.7</td>
</tr>
<tr>
<td>$n = 10, k = 50$</td>
<td>5.1</td>
<td>6.5</td>
<td>5.2</td>
<td>5.5</td>
<td>6.6</td>
<td>6.5</td>
<td>6.5</td>
<td>6.8</td>
</tr>
<tr>
<td>$n = 100, k = 5$</td>
<td>49.9</td>
<td>78.3</td>
<td>74.7</td>
<td>77.9</td>
<td>78.0</td>
<td>75.4</td>
<td>74.8</td>
<td>78.7</td>
</tr>
<tr>
<td>$n = 100, k = 10$</td>
<td>49.9</td>
<td>82.8</td>
<td>76.7</td>
<td>81.4</td>
<td>82.4</td>
<td>77.4</td>
<td>77.1</td>
<td>83.5</td>
</tr>
<tr>
<td>$n = 100, k = 50$</td>
<td>49.8</td>
<td>85.2</td>
<td>64.5</td>
<td>67.7</td>
<td>84.3</td>
<td>78.3</td>
<td>78.0</td>
<td>84.9</td>
</tr>
<tr>
<td>$n = 500, k = 5$</td>
<td>249.8</td>
<td>405.8</td>
<td>402.0</td>
<td>406.7</td>
<td>405.8</td>
<td>388.2</td>
<td>380.6</td>
<td>401.6</td>
</tr>
<tr>
<td>$n = 500, k = 10$</td>
<td>249.0</td>
<td>437.8</td>
<td>429.5</td>
<td>438.9</td>
<td>437.1</td>
<td>408.0</td>
<td>395.0</td>
<td>432.5</td>
</tr>
<tr>
<td>$n = 500, k = 50$</td>
<td>249.6</td>
<td>463.7</td>
<td>427.2</td>
<td>437.6</td>
<td>457.6</td>
<td>413.6</td>
<td>402.8</td>
<td>438.9</td>
</tr>
</tbody>
</table>

Normalised total reward

**Iteration**

(a) $n = 10$

(b) $n = 100$

(c) $n = 500$
Tabular MDPs
Tabular MDPs

<table>
<thead>
<tr>
<th>Setup</th>
<th>Random</th>
<th>PSRL</th>
<th>OPSRL</th>
<th>UCRL2</th>
<th>BEB</th>
<th>$\varepsilon$-Greedy</th>
<th>Greedy</th>
<th>$RL^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 10$</td>
<td>100.1</td>
<td>138.1</td>
<td>144.1</td>
<td>146.6</td>
<td>150.2</td>
<td>132.8</td>
<td>134.8</td>
<td>156.2</td>
</tr>
<tr>
<td>$n = 25$</td>
<td>250.2</td>
<td>408.8</td>
<td>425.2</td>
<td>424.1</td>
<td>427.8</td>
<td>377.3</td>
<td>368.8</td>
<td>445.7</td>
</tr>
<tr>
<td>$n = 50$</td>
<td>499.7</td>
<td>904.4</td>
<td>930.7</td>
<td>918.9</td>
<td>917.8</td>
<td>823.3</td>
<td>769.3</td>
<td>936.1</td>
</tr>
<tr>
<td>$n = 75$</td>
<td>749.9</td>
<td>1417.1</td>
<td>1449.2</td>
<td>1427.6</td>
<td>1422.6</td>
<td>1293.9</td>
<td>1172.9</td>
<td>1428.8</td>
</tr>
<tr>
<td>$n = 100$</td>
<td>999.4</td>
<td>1939.5</td>
<td>1973.9</td>
<td>1942.1</td>
<td>1935.1</td>
<td>1778.2</td>
<td>1578.5</td>
<td>1913.7</td>
</tr>
</tbody>
</table>
Visual navigation

(a) Sample observation
(b) Layout of the $5 \times 5$ maze in (a)
(c) Layout of a $9 \times 9$ maze
## Visual navigation

<table>
<thead>
<tr>
<th>Episode</th>
<th>Small</th>
<th>Large</th>
<th>Episode</th>
<th>Small</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>52.4 ± 1.3</td>
<td>180.1 ± 6.0</td>
<td>1</td>
<td>99.3%</td>
<td>97.1%</td>
</tr>
<tr>
<td>2</td>
<td>39.1 ± 0.9</td>
<td>151.8 ± 5.9</td>
<td>2</td>
<td>99.6%</td>
<td>96.7%</td>
</tr>
<tr>
<td>3</td>
<td>42.6 ± 1.0</td>
<td>169.3 ± 6.3</td>
<td>3</td>
<td>99.7%</td>
<td>95.8%</td>
</tr>
<tr>
<td>4</td>
<td>43.5 ± 1.1</td>
<td>162.3 ± 6.4</td>
<td>4</td>
<td>99.4%</td>
<td>95.6%</td>
</tr>
<tr>
<td>5</td>
<td>43.9 ± 1.1</td>
<td>169.3 ± 6.5</td>
<td>5</td>
<td>99.6%</td>
<td>96.1%</td>
</tr>
</tbody>
</table>

(c) %Improved

<table>
<thead>
<tr>
<th>Small</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>91.7%</td>
<td>71.4%</td>
</tr>
</tbody>
</table>
Visual navigation

(a) Good behavior, 1st episode  
(b) Good behavior, 2nd episode

(c) Bad behavior, 1st episode  
(d) Bad behavior, 2nd episode
Visual navigation
Wrap up RL2

- Fast reinforcement learning via slow reinforcement learning using RNN states
- Comparable to theoretical optimum in small problem setting
- Scalable to complicated vision tasks
- Potential improvement for RL algorithm and network architecture
Summary

Learning How to Learn
Thanks for listening
References

- Paper && Quotes:
  - https://bair.berkeley.edu/blog/2017/07/18/learning-to-learn/

- Pictures:
  - https://paperswithcode.com/task/multi-armed-bandits
  - https://www.coursera.org/learn/learning-how-to-learn