Visual meta-learning for planning and control

Seminar on Current Works in Computer Vision
© Chair of Pattern Recognition and Image Processing.

Samuel Roth
Winter Semester 2018/19

Albert-Ludwigs-Universität Freiburg
Content

- Motivation, Goal.
- Meta learning.
- Concept acquisition through meta learning (CAML).
- Implementation of CAML.
- Use observation classifier as reward functions.
- Experimental results.
Motivation

- Reinforcement Learning needs rewards for learning.
Motivation

- Reinforcement Learning needs rewards for learning.
Motivation

- Reinforcement Learning needs rewards for learning.
- Specifying rewards are often hand crafted and domain specific.

Example Tasks
Motivation

- Reinforcement Learning needs rewards for learning.
- Specifying rewards are often hand-crafted and domain specific.
- Learning rewards for a specific tasks requires large labeled dataset for training.
Introduction - Goal and Approach

Goal

- Derive reward function.
- From camera input.
- Only few positive examples of success.
- Utilize information given by other tasks.

Meta learning approach

- “Learn to learn”
- Train a model to classify tasks for multiple different tasks.
- Expect the model to learn internal representations.
- For a new task is given ⇒ Model has a good initialization.
Introduction - Goal and Approach

Goal

- Derive reward function.
- From camera input.
- Only few positive examples of success.
- Utilize information given by other tasks.

Meta learning approach

- ”Learn to learn”
- Train a model to classify tasks for multiple different tasks.
- Expect the model to learn internal representations.
- For a new task is given ⇒ Model has a good initialization.
Learn Structure among relevant tasks

Task 1

Task 2

...

Task N

Meta-training
Introduction - Framework

Infer Objective for New Tasks

1. Show examples of success
2. Parameter adaptation
3. Task-specific classifier
4. Derive reward

Run Planning or Reinforcement Learning
Meta Learning for few shot objective learning

Concept

• Given a dataset of labeled observations over multiple tasks

\[ D_i := \{(o_j, y_j) | j = 1, \cdots, k; y_j \in \{0, 1\}\} \]

• A meta-learner \( f_L \) and a classifier \( f_C \)

\[ f_C = f_L(D_{new}^+; \theta) : Observation \rightarrow [0, 1] \]

• With the objective to

\[ \min_{\theta} \sum_{i}^{n} \sum_{(o_j, y_j) \in D_{i}^{train}} \mathcal{L}(y_j, f_L(D_{i}^+; \theta)(o_j)) \]
CAML - Concept Acquisition through meta learning.

- Incorporate gradient descent for parameter adaption.
- Adapt $\theta$ to new task $\tau$:
  $$g(\theta, D^{+} + \tau) := \theta - \alpha \nabla_{\theta} \sum_{(o_i, y_i) \in D^{+} + \tau} L(y_i, f_{CAML}(\theta)(o_i)) = \theta'_{\tau}$$
- Initial $\theta$ learned in meta training:
  $$\min \theta \sum_{i} \sum_{(o_j, y_j) \in D^{train}} L(y_j, f_{CAML}(\theta'(\theta, D^{+} + \tau_i))(o_j))$$
CAML - Concept Acquisition through meta learning.

- Incorporate gradient descent for parameter adaption.
CAML - Concept Acquisition through meta learning.

- Incorporate gradient descent for parameter adaptation.
- Adapt $\theta$ to new task $\tau$:

$$
g(\theta, D^+_{\tau}) := \theta - \alpha \nabla_{\theta} \sum_{(o_i, y_i) \in D^+_{\tau}} \mathcal{L}(y_i, f_{CAML}(\theta)(o_i)) = \theta'_{\tau}
$$
CAML - Concept Acquisition through meta learning.

- Incorporate gradient descent for parameter adaption.
- Adapt $\theta$ to new task $\tau$:

$$g(\theta, D^+_\tau) := \theta - \alpha \nabla_\theta \sum_{(o_i, y_i) \in D^+_\tau} \mathcal{L}(y_i, f_{CAML}(\theta)(o_i)) = \theta'_\tau$$

- Initial $\theta$ learned in meta training:

$$\min_{\theta} \sum_i^n \sum_{(o_j, y_j) \in D_{\tau_i}^{train}} \mathcal{L}(y_j, f_{CAML}(g(\theta, D^+_\tau))(o_j))$$
Algorithm 1 CAML training phase

Require: $\alpha$, $\beta$ learning rate Hyper-parameters. $k$ examples size ($k$-shot).

1: Randomly initialize $\theta$
2: while not done do
3: Sample batch of tasks of size $k$.
4: for all $\tau_i$ in batch do
5: Sample $D_i^{\text{train}}$ and $D_i^{+}$ s.t. $D_i^{\text{train}} \cap D_i^{+} = \emptyset$
6: Calculate task parameters with gradient decent:
7: $\theta_i = \theta - \alpha \nabla_{\theta} \sum_{(o_j, y_j) \in D_i^{+}} \mathcal{L}(y_j, f_{\text{CAML}}(\theta)(o_j))$
8: end for
9: Update $\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\tau_i \in \text{batch}} \sum_{(o_j, y_j) \in D_i^{\text{train}}} \mathcal{L}(y_j, f_{\text{CAML}}(\theta_i)(o_j))$
10: end while
Algorithm 2 CAML test phase

Require: $D^+_{new}$ positive examples of success for a new task.
Require: $\theta$ Pre-trained meta-learner parameters.
Require: $\alpha$ learning rate Hyper-parameters.

1: Adapt parameters with gradient decent:
2: $\theta' = \theta - \alpha \nabla_\theta \sum_{(o_j, y_j) \in D^+_{new}} \mathcal{L}(y_j, f_{CAML}(\theta)(o_j))$
3: return $f_{CAML}(\theta')$
From observation classifier to rewards.

- $f_C(o) \in [0, 1]$ interpreted as probability of $o$ to be a successful execution of the task.
- Directly use the $f_C$ output as reward.
- To reduce the effect of false positives.
- Use a threshold $t \in (0, 1)$ and set $f_C(o) \leftarrow 0$ if $f_C(o) < t$. 
Experimental results - Single task relative position

example 1  example 2
Experimental results - Multi task relative position

task 1

task 2
Experimental evaluation competitors

- **Pixel distance**: Measure the $\ell_2$ distance between observation and successful observation.
- **Latent space distance**: Measure distance in a learned latent space using an auto encoder. Here ”Deep spatial autoencoders for visuomotor learning” by Finn et al. 2016.
- **Oracle**: Use the ground truth if available to get an upper bound of performance (only applied in the rope manipulation domain).
Experimental results - Relative position
Experimental results - Relative position planning costs
Experimental results - Rope manipulation

example 1

example 2

Time
Experimental results - Rope manipulation

<table>
<thead>
<tr>
<th>method</th>
<th>median distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>pixel distance</td>
<td>0.54 (0.34, 0.70)</td>
</tr>
<tr>
<td>AE distance</td>
<td>0.59 (0.50, 0.69)</td>
</tr>
<tr>
<td>FLO (ours)</td>
<td>0.27 (0.18, 0.34)</td>
</tr>
<tr>
<td>oracle</td>
<td>0.16 (0.13, 0.23)</td>
</tr>
</tbody>
</table>

example 1  example 2

---

time
Experimental results - 3D navigation

example 1  example 2

time
Experimental results - 3D navigation

<table>
<thead>
<tr>
<th>method</th>
<th>success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>pixel distance</td>
<td>33.0%</td>
</tr>
<tr>
<td>AE distance</td>
<td>44.0%</td>
</tr>
<tr>
<td>FLO (ours)</td>
<td>77.8%</td>
</tr>
</tbody>
</table>

Example 1: Image 1, Image 2

Example 2: Sequence of images showing navigation through a 3D environment over time.
Conclusion

• Model that can achieve rapid adaptation.
• Internal representation that is broadly suitable to many tasks.
• If tasks have distinct visual concept unlikely that meta-learning will acquire useful knowledge.
• Experiments showed good performance on vision-based manipulation skills.
Conclusion

- Model that can achieve rapid adaptation.
Conclusion

- Model that can achieve rapid adaptation.
- Internal representation that is broadly suitable to many tasks.
• Model that can achieve rapid adaptation.
• Internal representation that is broadly suitable to many tasks.
• If tasks have distinct visual concept unlikely that meta-learning will acquire useful knowledge.
Conclusion

- Model that can achieve rapid adaptation.
- Internal representation that is broadly suitable to many tasks.
- If tasks have distinct visual concept unlikely that meta-learning will acquire useful knowledge.
- Experiments showed good performance on vision-based manipulation skills.
Thanks!