Meta-transfer Learning for Few-shot Learning

Yaoyao Liu
Tianjin University and NUS School of Computing
OUTLINE

• Research Background

• Methods
  • Meta-transfer Learning
  • Hard-task Meta Batch

• Experiments and Conclusions
Deep learning achieved a lot of success in many fields: Computer Vision, NLP…

Limitation: most algorithms are based on **supervised learning**, so we need lots of **labeled samples** to train the model.
Research Background

• Limitation: most algorithms are based on *supervised learning*, so we need lots of *labeled samples* to train the model.
Task: Few-shot Learning
Our focus: few-shot image classification
Few-shot Classification

Using only *a few labeled samples* to train the classifier

<table>
<thead>
<tr>
<th>1-shot, 4-class</th>
<th>train-set</th>
<th>test-set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>Dog</td>
<td>Lion</td>
</tr>
</tbody>
</table>

**Shot number:** how many samples for one class

**Class number:** how many classes in the small dataset
Few-shot Classification

Using only *a few labeled samples* to train the classifier

1-shot, 4-class
- Cat
- Dog
- Lion
- Bowl

```
train-set
```
```
test-set
```

5-shot, 3-class
- Cat
- Dog
- Lion

```
train-set
```
```
test-set
```
1. **Meta learning based:**
   Meta-LSTM\(^1\), MAML\(^2\), ...

2. **Metric learning based:**
   MatchingNets\(^3\), ProtoNets\(^4\), ...

3. **Others (based on augmentation, domain adaptation...):**
   Data Augmentation GAN\(^5\), CCN+\(^6\), ...

---

1. **Meta learning based:**
   Meta-LSTM\(^1\), MAML\(^2\), ...

2. **Metric learning based:**
   MatchingNets\(^3\), ProtoNets\(^4\), ...

3. **Others (based on augmentation, domain adaptation...):**
   Data Augmentation GAN\(^5\), CCN+\(^6\), ...

---

\(^1\) Ravi et al. "Optimization as a model for few-shot learning." ICLR 2016;
\(^3\) Vinyals et al. "Matching networks for one shot learning." NIPS 2016;
\(^5\) Antoniou et al. "Data augmentation generative adversarial networks." In ICLR Workshops 2018;
\(^6\) Hsu et al. "Learning to cluster in order to transfer across domains and tasks." ICLR 2018.
• Research Background

• Methods
  • Meta-transfer Learning
  • Hard-task Meta Batch

• Experiments and Conclusions
Classic Algorithm: MAML

Learn initialization weights for different tasks using meta-learning.

Classic Algorithm: MAML

MAML

CONV1
CONV2
CONV3
CONV4
FC

$\Theta_0$

$M$ epochs

base learning

$\Theta_M$

CONV1
CONV2
CONV3
CONV4
FC

$\mathcal{T}(tr):$

1 2 3 4

$D_{train}$

$\mathcal{T}(te):$

$\mathcal{T}(te):$

$D_{test}$

meta-test phase

Problems of MAML

- Failure on deeper networks

\[ \Theta_0 \]

\[
\begin{align*}
\text{CONV1} \\
\text{CONV2} \\
\text{CONV3} \\
\text{CONV4} \\
\text{FC}
\end{align*}
\]

\[ M \text{ epochs} \]

\[ \mathcal{T}^{(tr)}: \]

\[ \Theta_M \]

\[
\begin{align*}
\text{CONV1} \\
\text{CONV2} \\
\text{CONV3} \\
\text{CONV4} \\
\text{FC}
\end{align*}
\]
Problems of MAML

- Failure on deeper networks

- Slow convergence speed
  For the networks with only 4 conv layers, MAML trains $60k$ iterations. It takes more than 30 hours on a NVIDIA V100 GPU.
Our Methods

- Failure on deeper networks \(\rightarrow\) Meta-transfer Learning

- Slow convergence speed \(\rightarrow\) Hard Task Meta Batch
Overview of the Methods

- **Meta-transfer Learning**
  
  Explore the structure of the classifier, control the degree of freedom

- **Hard Task Meta Batch**

Convolution Networks in \textit{MAML}

A Conv Layer

A Filter

CONV1
CONV2
CONV3
CONV4
FC

\begin{itemize}
\item learnable
\item fixed
\end{itemize}
Learn the Structure by Many-shot Classification

A Conv Layer

A Filter

Pre-trained the network with many-shot classification task
Meta-transfer Learning

A Conv Layer

structure

The Scaling Weights

the degree of freedom

learnable

fixed
Meta-transfer Learning

A Conv Layer

Applying the scaling weights for each filter
Parameter number is reduced to approximately 1/9

- **Green**: learnable
- **Yellow**: fixed
The Pipeline

- **pre-train**
- **meta-train**
- **meta-test**

- **Pred**

- **reorganize**

- **target few-shot task**

- learnable
- fixed
Overview of the Methods

- **Meta-transfer Learning**

- **Hard Task Meta Batch**
  
The idea is from hard example mining\[^1\]
  
  Hard example $\rightarrow$ hard task

---

\[^1\] Shrivastava et al. "Training region-based object detectors with online hard example mining." CVPR 2016.
Hard Task Meta Batch

Meta learning iterations
OUTLINE

• Research Background

• Method
  • Meta-transfer Learning
  • Hard-task Meta Batch

• Experiments and Conclusions
Datasets

- **minImageNet**
  - Reorganized from ImageNet
  - Vinyals et al.\(^1\) first devised the dataset, and it is widely used in evaluating few-shot learning methods
  - 100 classes (64 meta-train, 16 meta-val, 20 meta-test)

- **Fewshot-CIFAR100 (FC100)**
  - Reorganized from CIFAR100
  - Splitted by Oreshkin et al.\(^2\)
  - 100 classes (60 meta-train, 20 meta-val 20 meta-test)
  - 20 super-classes (12 meta-train, 4 meta-val 4 meta-test)

---

Evaluation

- Image Classification Accuracy
  - 600 testing tasks randomly sampled from the meta-test set
  - 5-class
  - 1-shot and 5-shot on miniImageNet
  - 1-shot, 5-shot and 10-shot on FC100

* The same evaluation protocol with MAML[1]

<table>
<thead>
<tr>
<th>Methods</th>
<th>miniImageNet (5-class)</th>
<th>FC100 (5-class)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
</tr>
<tr>
<td>MatchingNets [1]</td>
<td>43.4 ± 0.8 %</td>
<td>55.3 ± 0.7 %</td>
</tr>
<tr>
<td>Meta-LSTM [2]</td>
<td>43.6 ± 0.8 %</td>
<td>60.6 ± 0.7 %</td>
</tr>
<tr>
<td>MAML [3]</td>
<td>48.7 ± 1.8 %</td>
<td>63.1 ± 0.9 %</td>
</tr>
<tr>
<td>ProtoNets [4]</td>
<td>49.4 ± 0.8 %</td>
<td>68.2 ± 0.7 %</td>
</tr>
<tr>
<td>TADAM [5]</td>
<td>58.5 ± 0.3 %</td>
<td><strong>76.7 ± 0.3 %</strong></td>
</tr>
<tr>
<td>Ours (MTL + HT)</td>
<td><strong>61.2 ± 1.8 %</strong></td>
<td>75.5 ± 0.8 %</td>
</tr>
</tbody>
</table>

## Ablation Study

<table>
<thead>
<tr>
<th>Method</th>
<th>minImageNet (5-class)</th>
<th>FC100 (5-class)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
</tr>
<tr>
<td>Train from scratch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>45.3</td>
<td>64.6</td>
</tr>
<tr>
<td>Finetune on pre-train model</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55.9</td>
<td>71.4</td>
</tr>
<tr>
<td>Ours (MTL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60.2</td>
<td>74.3</td>
</tr>
<tr>
<td>Ours (MTL + HT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>61.2</td>
<td>75.5</td>
</tr>
</tbody>
</table>
Validation Accuracy

(a) (b) minilImagenet, 1-shot and 5-shot  
(c) (d) (e) FC100, 1-shot, 5-shot, and 10-shot
Conclusions

❖ A novel MTL method that learns to transfer large-scale pre-trained DNN weights for solving few-shot learning tasks.

❖ A novel HT meta-batch learning strategy that forces meta-transfer to “grow faster and stronger through hardship”.

❖ Extensive experiments on minilmageNet and FC100, and achieving the state-of-the-art performance.
This work:
GitHub repo: https://github.com/y2l/meta-transfer-learning-tensorflow
Thank you!
Any questions?

Email: yaoyao.liu@u.nus.edu