iTAML: An Incremental Task-Agnostic Meta-learning Approach

Jathushan Rajasegaran\textsuperscript{1} \hspace{1cm} Salman Khan\textsuperscript{1} \hspace{1cm} Munawar Hayat\textsuperscript{1} \\
Fahad Shahbaz Khan\textsuperscript{1,2} \hspace{1cm} Mubarak Shah\textsuperscript{3}

\textsuperscript{1}Inception Institute of Artificial Intelligence \hspace{1cm} \textsuperscript{2}CVL, Linköping University \hspace{1cm} \textsuperscript{3}University of Central Florida
Problem Definition

• Continual learning is essential for intelligent systems.

• Continual learning algorithms need to retain the past knowledge while learning new concepts on newly revealed data sets.

   In other words, these algorithms needs to achieve generalization.

• Meta-learning is an ideal tool for such problems.
The Challenge

• Achieving generalization to new data while preserving past knowledge remains a challenge for existing incremental learning algorithms.

• *Meta-learning* suffers on incremental learning setting due to,
  • Out of Order Distribution (OOD).
  • Often requires fine-tuning at the end.
  • Skewed data distribution with limited memory.

*iTAML tries to bridge the gap between meta-learning and incremental learning.*
Incremental Task-Agnostic Meta-learning

The tasks are observed sequentially.

Each task is a set of classes.

iTAML incrementally learns new tasks with meta-updates and tries to retain previous knowledge.

At inference, given a data continuum, iTAML first predicts the task and then quickly adapts to it.
Incremental Task-Agnostic Meta-learning

• The experimental setting for iTAML:
  
  • Involves learning a single model which can generalize to all the tasks (*old* as well as *new*).
  
  • We make a weak assumption that a data continuum is available with all the samples belongs to a single task (yet the *task* is unknown).
  
  • Our meta-learned generic model is good enough to find the correct task.
Incremental Task-Agnostic Meta-learning

- iTAML uses the following novel learning and inference strategies:
  - A momentum based meta-update rule to avoid forgetting.
  - Disentangling the network into a generic feature extractor and task-specific classification weights.
  - A task-agnostic prediction mechanism, with two stage classification.
  - A sampling rate selection approach for data continuum.
Meta Training of iTAML

- **Network** \( \Phi \)
- **Train data**
  - New Task Data \( D(t) \)
  - Memory \( M(t - 1) \)
- **Group by tasks**
- **Mini batch** \( B_m = \{(x_k, y_k, \ell_k)\}_{k=1}^{K} \)
- **Inner loop**
  - \( \Phi_1 \) \( \downarrow \)
  - \( \Phi_2 \) \( \downarrow \)
  - \( \Phi_i \) \( \downarrow \)
  - \( \Phi_t \) \( \downarrow \)
  - \( \frac{1}{t} \sum_{i=1}^{t} \Phi_i \) \( \leftarrow \) Controller
  - \( \Phi_{base} \)
  - \( \Phi_{new} \leftarrow \eta \frac{1}{t} \sum_{i=1}^{t} \Phi_i + (1 - \eta) \Phi_{base} \)

- **Outer loop**
  - \( \Phi_{base} \leftarrow \Phi \)
Meta Training of iTAML

• Each mini-batch is further broken into task specific micro batches.

• In the inner loop, task specific models $\Phi_i$ are trained for each seen task.

• Then, a momentum controller combines these task specific weights in the outer loop.

$$
\Phi_{new} = \eta \frac{1}{t} \sum_{i=1}^{t} \Phi_i + (1-\eta) \Phi_{base}.
$$
Meta Training of iTAML

\[ \mathcal{W}_i^* \]

\[ \mathcal{W}_j^* \]

\[ \{\theta, \phi_i, \phi_j\} \]

Initial parameters

Path of \( \theta \)  
Path of \( \phi_i \)  
Path of \( \phi_j \)
Meta Training of iTAML

- Since, the feature space parameters and classification parameters are tuned separately, iTAML remains task-agnostic.

- Feature space parameters, are tuned for each task and combined in the outer loop, hence they remain close to optimal solution manifold of all the tasks.

- Classification parameters are tuned only for the specific task; hence they remain close to the corresponding task’s optimal solution manifold.
Inference of iTAML

Data continuum $C(p)$

Network $\Phi^t$

Task Prediction

Task wise maximum response

Task scores $(S)$

Class Prediction

$\Phi_{new}$

Gradient Update

$\Phi^t$

Filter from memory $B'_m \sim \hat{M}_{t_{pred}}$

Memory

Final Predictions

$t_{pred}$
Inference of iTAML

At inference, iTAML uses a two-stage prediction.

- First, given a data continuum $C(i, p)$, it predicts the task using average predictions over data samples.
- Then, it uses exemplar data to adapt for the task using a single gradient update.
- Finally, it processes the continuum and gives class-wise predictions.
Experimental Results

CIFAR-100: Learning 10 Classes at a time

CIFAR-100: Learning 5 Classes at a time

CIFAR-100: Learning 20 Classes at a time

Accuracy % vs Number of Classes

- DMC
- LwF
- RWalk
- SI
- MAS
- EWC
- Finetuning
- FixedRep
- iCaRL
- RPS
- Ours
Experimental Results

Note that, with about 15 samples in a continuum, the model can accurately predict that correct task with 95% accuracy!
## Experimental Results

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Methods</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Final</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Finetuning</td>
<td>99.3</td>
<td>49.4</td>
<td>32.6</td>
<td>24.7</td>
<td>20.0</td>
<td>16.7</td>
<td>13.9</td>
<td>12.3</td>
<td>11.1</td>
<td>9.9</td>
</tr>
<tr>
<td></td>
<td>FixedRep</td>
<td>99.3</td>
<td>88.1</td>
<td>73.7</td>
<td>62.6</td>
<td>55.7</td>
<td>50.2</td>
<td>42.9</td>
<td>41.3</td>
<td>39.2</td>
<td>35.3</td>
</tr>
<tr>
<td></td>
<td>LwF(TPAMI’18)[15]</td>
<td>99.3</td>
<td>95.2</td>
<td>85.9</td>
<td>73.9</td>
<td>63.7</td>
<td>54.8</td>
<td>50.1</td>
<td>44.5</td>
<td>40.7</td>
<td>36.7</td>
</tr>
<tr>
<td></td>
<td>iCaRL(CVPR’17)[23]</td>
<td>99.3</td>
<td>97.2</td>
<td>93.5</td>
<td>91.0</td>
<td>87.5</td>
<td>82.1</td>
<td>77.1</td>
<td>72.8</td>
<td>67.1</td>
<td>63.5</td>
</tr>
<tr>
<td></td>
<td>RPSnet(NeurIPS’19)[22]</td>
<td>100.0</td>
<td>97.4</td>
<td>94.3</td>
<td>92.7</td>
<td>89.4</td>
<td>86.6</td>
<td>83.9</td>
<td>82.4</td>
<td>79.4</td>
<td>74.1</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>99.4</td>
<td>96.4</td>
<td>94.4</td>
<td>93.0</td>
<td>92.4</td>
<td>90.6</td>
<td>89.9</td>
<td>90.3</td>
<td>90.3</td>
<td>89.8+15.7</td>
</tr>
<tr>
<td></td>
<td>Finetuning</td>
<td>90.2</td>
<td>43.1</td>
<td>27.9</td>
<td>18.9</td>
<td>15.6</td>
<td>14.0</td>
<td>11.7</td>
<td>10.0</td>
<td>8.9</td>
<td>8.2</td>
</tr>
<tr>
<td></td>
<td>FixedRep</td>
<td>90.1</td>
<td>76.1</td>
<td>66.9</td>
<td>58.8</td>
<td>52.9</td>
<td>48.9</td>
<td>46.1</td>
<td>43.1</td>
<td>41.2</td>
<td>38.5</td>
</tr>
<tr>
<td></td>
<td>LwF(TPAMI’18)[15]</td>
<td>90.2</td>
<td>77.6</td>
<td>63.6</td>
<td>51.6</td>
<td>42.8</td>
<td>35.5</td>
<td>31.5</td>
<td>28.4</td>
<td>26.1</td>
<td>24.2</td>
</tr>
<tr>
<td></td>
<td>iCaRL(CVPR’17)[23]</td>
<td>90.1</td>
<td>82.8</td>
<td>76.1</td>
<td>69.8</td>
<td>63.3</td>
<td>57.2</td>
<td>53.5</td>
<td>49.8</td>
<td>46.7</td>
<td>44.1</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>91.5</td>
<td>89.0</td>
<td>85.7</td>
<td>84.0</td>
<td>80.1</td>
<td>76.7</td>
<td>70.2</td>
<td>67.0</td>
<td>67.9</td>
<td>63.2+19.1</td>
</tr>
<tr>
<td></td>
<td>iCaRL(CVPR’17)[23]</td>
<td>94.2</td>
<td>93.7</td>
<td>90.8</td>
<td>86.5</td>
<td>80.8</td>
<td>77.2</td>
<td>74.9</td>
<td>71.1</td>
<td>68.5</td>
<td>65.5</td>
</tr>
<tr>
<td></td>
<td>RPSnet(NeurIPS’19)[22]</td>
<td>92.8</td>
<td>92.0</td>
<td>92.3</td>
<td>90.8</td>
<td>86.3</td>
<td>83.6</td>
<td>80.0</td>
<td>76.4</td>
<td>71.8</td>
<td>65.0</td>
</tr>
<tr>
<td></td>
<td>BiC(CVPR’19)[30]</td>
<td>95.7</td>
<td>96.5</td>
<td>96.5</td>
<td>95.7</td>
<td>95.1</td>
<td>94.2</td>
<td>93.2</td>
<td>91.7</td>
<td>90.0</td>
<td>87.6</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>94.0</td>
<td>95.6</td>
<td>96.0</td>
<td>95.8</td>
<td>95.5</td>
<td>95.4</td>
<td>95.2</td>
<td>95.1</td>
<td>95.0</td>
<td>95.0+7.4</td>
</tr>
</tbody>
</table>
Experimental Results

(a) Task Agnostic
(b) Task Aware
(c) No Meta Updates

Accuracy % vs Number of Classes
Thank You!

https://github.com/brjathu/iTAML