Annotation-Efficient Learning: Class-Incremental Learning and Few-Shot Learning

Yaoyao Liu

11 Sept. 2020
Outline of today’s talk

1. Class-Incremental Learning
   Mnemonics Training: Multi-Class Incremental Learning without Forgetting
   CVPR 2020

2. Few-Shot Learning
   An Ensemble of Epoch-wise Empirical Bayes for Few-shot Learning
   ECCV 2020
**Background: class-incremental learning**

**Phase 1**

- Train on Data 1
- Classifier
- Test for Data 1
Background: class-incremental learning

Phase 2

Data 2

Classifier

train

Classifier

Test for Data 1

Test for Data 1+2
Background: class-incremental learning

Phase 3

Classifier

Test for Data 1

Classifier

Test for Data 1+2

Classifier

Test for Data 1+2+3

Data 3

train
Background: Class-Incremental Learning

Phase 3

Challenge: catastrophic forgetting

Classifier
Test for Data 1

Classifier
Test for Data 1+2

Classifier
Test for Data 1+2+3

Data 3

train

overfit
Literature review

**Technique 1: Replay samples for the old classes:**

iCaRL\(^1\), IL2M\(^2\), ...

\(\rightarrow\) **Mnemonics exemplars**

**Technique 2: Preserve the knowledge for the old model:**

LwF\(^3\), LUCIR\(^4\), PODNet\(^5\)...

\(\rightarrow\) **Weight transfer operations**

**References**

Replay samples for the old classes

Phase 1

Data 1

Exemplar 1

Classifier

Train

Test for Data 1
Replay samples for the old classes

**Phase 2**

- **Exemplar 1**
- **Data 2**

**Classifier**

**train**

**Question:** How to extract the exemplars?
**Question: how to extract the exemplars?**

**Existing methods:**

E.g., herding\(^{[1, 4, 6]}\): select the samples near the average embedding

**Limitations for existing methods:**
- Heuristic selection, not performance-based
- Select from finite sets (real images)

**Our method: Mnemonics exemplars**

Question: Can we generate the optimal exemplars?

**Benefits for our method:**
- Optimal selection by end-to-end training
- Select from continuous (infinite) synthetic data

References

Question: how to formulate the optimization of the exemplars?

In the $i$-th incremental phase,

We have: Exemplars for previous phases $\mathcal{E}_{0:i-1} \cup D_i$ → $\Theta^*_i$ (model)

We aim to get: Exemplars for the current phase $\mathcal{E}_{0:i}$ → $\Theta^\varepsilon_i$

Bilevel optimization formulation:

$$\min_{\Theta^\varepsilon_i} \mathcal{L}(\Theta^\varepsilon_i; \mathcal{E}_{0:i-1} \cup D_i)$$

s. t. $\Theta^\varepsilon_i = \min_{\Theta_i} \mathcal{L}(\Theta_i; \mathcal{E}_{0:i})$
Literature review

Technique 1: Replay samples for the old classes:

iCaRL\textsuperscript{[1]}, IL2M\textsuperscript{[2]}, ...

→ Mnemonics exemplars

Technique 2: Preserve the knowledge for the old model:

LwF\textsuperscript{[3]}, LUCIR\textsuperscript{[4]}, PODNet\textsuperscript{[5]}...

→ Weight transfer operations

References
Preserve the knowledge for the old model

Phase 1

Data 1

train

Classifier

Exemplar 1

Test for Data 1
Preserve the knowledge for the old model

Phase 2

Motivation:
Force the new model and the old model to be similar

Predictions for old classes

Groundtruth for new classes

Distillation loss
Classification loss
Preserve the knowledge for the old model

iCaRL\textsuperscript{[1]} (CVPR 2017) $\rightarrow$ Distillation on predictions

LUCIR\textsuperscript{[4]} (CVPR 2019) $\rightarrow$ Distillation on the final feature maps

PODNet\textsuperscript{[5]} (ECCV 2020) $\rightarrow$ Distillation on the feature maps from all layers

\textit{Distillation: preserve high-level knowledge for the old model}

\textit{It is better to transfer low-level knowledge among tasks…}\textsuperscript{[7]}

\textit{Q: Can we preserve the low-level knowledge for the old model?}

\textbf{References}
How to transfer low-level knowledge for class-incremental learning?

Weight transfer operations: Channel-wise masks
Global computing glow

Our method: **Technique 1 + Technique 2**

Data $D_0$ \[\xrightarrow{\text{Model } \Theta_0} \] Exemplars $\mathcal{E}_0$ \[\xrightarrow{\text{BOP}} \] Data $D_1$

Data $D_1$ \[\xrightarrow{\text{Model } \Theta_1} \] Exemplars $\tilde{\mathcal{E}}_0, \mathcal{E}_1$

**Weight transfer operations**

**BOP** = Bilevel Optimization Program
Our method boosts the performance

Dataset: ImageNet-Subset

References
Our method boosts the performance

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LwF° (2016)[2]</td>
<td>49.59</td>
<td>46.98</td>
<td>45.51</td>
<td>53.62</td>
<td>47.64</td>
<td>44.32</td>
<td>44.35</td>
<td>38.90</td>
<td>36.87</td>
</tr>
<tr>
<td></td>
<td>LwF w/ ours</td>
<td>54.21</td>
<td>52.72</td>
<td>51.59</td>
<td>60.94</td>
<td>59.25</td>
<td>59.71</td>
<td>52.70</td>
<td>50.37</td>
<td>50.79</td>
</tr>
<tr>
<td>Average acc. (%) ↑</td>
<td>iCaRL (2017)[1]</td>
<td>57.12</td>
<td>52.66</td>
<td>48.22</td>
<td>65.44</td>
<td>59.88</td>
<td>52.97</td>
<td>51.50</td>
<td>46.89</td>
<td>43.14</td>
</tr>
<tr>
<td></td>
<td>iCaRL w/ ours</td>
<td>60.00</td>
<td>57.37</td>
<td>54.13</td>
<td>72.34</td>
<td>70.50</td>
<td>67.12</td>
<td>60.61</td>
<td>58.62</td>
<td>53.46</td>
</tr>
<tr>
<td>( \bar{A} = \frac{1}{N+1} \sum_{i=0}^{N} A_i )</td>
<td>BiC (2019) [6]</td>
<td>59.36</td>
<td>54.20</td>
<td>50.00</td>
<td>70.07</td>
<td>64.96</td>
<td>57.73</td>
<td>62.65</td>
<td>58.72</td>
<td>53.47</td>
</tr>
<tr>
<td></td>
<td>BiC w/ ours</td>
<td>60.67</td>
<td>58.11</td>
<td>55.51</td>
<td>71.92</td>
<td>70.73</td>
<td>69.22</td>
<td>64.63</td>
<td>62.71</td>
<td>60.20</td>
</tr>
<tr>
<td></td>
<td>LUCIR (2019)[4]</td>
<td>63.17</td>
<td>60.14</td>
<td>57.54</td>
<td>70.84</td>
<td>68.32</td>
<td>61.44</td>
<td>64.45</td>
<td>61.57</td>
<td>56.56</td>
</tr>
<tr>
<td></td>
<td>LUCIR w/ ours</td>
<td><strong>63.34</strong></td>
<td><strong>62.28</strong></td>
<td><strong>60.96</strong></td>
<td><strong>72.58</strong></td>
<td><strong>71.37</strong></td>
<td><strong>69.74</strong></td>
<td><strong>64.54</strong></td>
<td><strong>63.01</strong></td>
<td><strong>61.00</strong></td>
</tr>
</tbody>
</table>

- **Generic**
- Boost the performance for **FOUR** different baselines

References
## Ablation study

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR-100</th>
<th></th>
<th></th>
<th>ImagNet-Subset</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N=5</td>
<td>10</td>
<td>25</td>
<td>5</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>Baseline (LUCIR [4])</td>
<td>63.17</td>
<td>60.14</td>
<td>57.54</td>
<td>70.84</td>
<td>68.32</td>
<td>61.44</td>
</tr>
<tr>
<td>+ weight transfer operations</td>
<td>62.98</td>
<td>61.23</td>
<td>60.36</td>
<td>71.66</td>
<td>71.02</td>
<td>69.40</td>
</tr>
<tr>
<td>+ weight transfer operations and mnemonics exemplars</td>
<td><strong>63.34</strong></td>
<td><strong>62.28</strong></td>
<td><strong>60.96</strong></td>
<td>72.58</td>
<td><strong>71.37</strong></td>
<td><strong>69.74</strong></td>
</tr>
</tbody>
</table>

**References**

**t-SNE results: clearer separation in the data**

**Phase 25**

*One region for one class*

*Light color: original data*

*Deep color: exemplars*

Dataset: ImageNet

**Herding [1, 4]**

**Mnemonics (ours)**

**Our method:**

- Clearer separation in data
- Exemplars locate on the class boundaries

References

Outline of today’s talk

1. Class-Incremental Learning
   Mnemonics Training: Multi-Class Incremental Learning without Forgetting
   CVPR 2020

2. Few-Shot Learning
   An Ensemble of Epoch-wise Empirical Bayes for Few-shot Learning
   ECCV 2020
Research background

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

Medical images: expensive to label the data

*Mitosis detection*

(From Yao Lu)
**Few-shot learning: learning with limited data**

**Question:** how to learn a model with limited labeled data?

**Task:** few-shot image classification

(Images from Ravi and Larochelle)
Review: meta-learning

**Seen classes**
Many-shot

**Unseen classes**
Few-shot

Same class number, same shot number

(Images from Ravi and Larochelle)
Review: meta-learning

Training tasks

Seen classes

Meta-train

Meta-test

Test task

Unseen classes

(Images from Ravi, Larochelle)
**Existing methods vs. our E³BM**

(a) MAML [8] 
(b) MTL [9] 
(c) SIB [10] 
(d) E³BM (ours)

**Existing methods:**
- A single base-learner
- Arbitrary base-learning hyperparameters
  - Unstable

**Our E³BM:**
- An ensemble of multiple base-learners
- Task-specific base-learning hyperparameters
  + Stable and robust

**References**
Existing method: MAML\cite{8}

For one training task:

\[ \nabla L_1^{(tr)} \rightarrow \Theta_0 \rightarrow \Theta_1 \rightarrow \Theta_2 \rightarrow \cdots \rightarrow \Theta_M \]

\[ \nabla L_1^{(tr)} \rightarrow \nabla L_2^{(tr)} \rightarrow \cdots \rightarrow \nabla L_M^{(tr)} \]

Predictions from a single base-learner

\[ z_M^{(te)} \rightarrow \mathcal{L}^{(te)} \]

\[ \Theta \quad \text{Epoch-wise base-learner} \quad \theta \quad \text{Base-learner initializer} \quad \alpha \quad \text{Learning rate} \quad \nu \quad \text{Combination weight} \]

References

**Existing method: MAML**\[^8\]

For one training task:

- **\(\Theta\)**: Epoch-wise base-learner
- **\(\theta\)**: Base-learner initializer
- **\(\alpha\)**: Learning rate
- **\(\upsilon\)**: Combination weight

References

Our method: $E^3BM$ framework

For one training task:

- $\theta$: Epoch-wise base-learner
- $\Theta_0$: Base-learner initializer
- $\alpha$: Learning rate
- $\nu$: Combination weight

Predictions from multiple base-learners
Our method: $E^3$BM framework

For one training task:

\[ \mathcal{T}^{(tr)} \rightarrow \text{Hyperprior Learner} \]

\[ \nabla L_1^{(tr)} \rightarrow \alpha_1 \rightarrow \Theta_1 \]

\[ \nabla L_2^{(tr)} \rightarrow \alpha_2 \rightarrow \Theta_2 \]

\[ \vdots \]

\[ \nabla L_M^{(tr)} \rightarrow \alpha_M \rightarrow \Theta_M \]

\[ \Theta_0 \rightarrow \Theta \]

\[ \text{Predictions from multiple base-learners} \]

\[ \bar{z}_1^{(te)} \rightarrow \theta \rightarrow \bar{z}_2^{(te)} \rightarrow \cdots \rightarrow \bar{z}_M^{(te)} \]

\[ \mathcal{L}^{(te)} \rightarrow \text{meta update} \]

\[ \text{Task-specific base-learning hyperparameters} \]

\[ \Theta \] Epoch-wise base-learner \quad \theta \] Base-learner initializer \quad \alpha \] Learning rate \quad \nu \] Combination weight
The architecture of the hyperprior learner

For the $m$-th base epoch:

\[ \nabla_\Theta \mathcal{L}_m^{(tr)} \]

\[ x^{(tr)} \]

(a) Epoch-independent
Boost the performance on **THREE baselines**

The 5-class few-shot classification results (%).

<table>
<thead>
<tr>
<th>Methods</th>
<th>Backbone</th>
<th>miniImageNet</th>
<th>tieredImageNet</th>
<th>FC100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
<td>1-shot</td>
</tr>
<tr>
<td>MAML</td>
<td>4CONV</td>
<td>48.70</td>
<td>63.11</td>
<td>49.0</td>
</tr>
<tr>
<td>MTL</td>
<td>ResNet-25</td>
<td>63.4</td>
<td>80.1</td>
<td>69.1</td>
</tr>
<tr>
<td>MAML+E³BM</td>
<td>4CONV</td>
<td>(53.2↑4.5)</td>
<td>(65.1↑2.0)</td>
<td>(52.1↑3.1)</td>
</tr>
<tr>
<td>(+time, +param)</td>
<td>–</td>
<td>(8.9, 2.2)</td>
<td>(9.7, 2.2)</td>
<td>(10.6, 2.2)</td>
</tr>
<tr>
<td>MTL+E³BM</td>
<td>ResNet-25</td>
<td><strong>64.3</strong>↑0.9</td>
<td><strong>81.0</strong>↑0.9</td>
<td><strong>70.0</strong>↑0.9</td>
</tr>
<tr>
<td>(+time, +param)</td>
<td>–</td>
<td>(5.9, 0.7)</td>
<td>(10.2, 0.7)</td>
<td>(6.7, 0.7)</td>
</tr>
</tbody>
</table>

(a) Inductive Methods

| SIB       | WRN-28-10| 70.0         | 79.2          | 72.9   | 82.8   | **45.2** | **55.9** |
| SIB+E³BM  | WRN-28-10| **71.4**↑1.4 | **81.2**↑2.0  | **75.6**↑2.7 | **84.3**↑1.5 | **46.0**↑0.8 | **57.1**↑1.2 |
| (+time, +param) | –   | (2.1, 0.04)  | (5.7, 0.04)   | (5.2, 0.04) | (4.9, 0.04) | (6.1, 0.04) | (7.3, 0.04) |

(b) Transductive Methods

**References**

1. Class-Incremental Learning

*Mnemonics Training: Multi-Class Incremental Learning without Forgetting*

GitHub: [https://github.com/yaoyao-liu/mnemonics-training](https://github.com/yaoyao-liu/mnemonics-training)

2. Few-Shot Learning

*An Ensemble of Epoch-wise Empirical Bayes for Few-shot Learning*

GitHub: [https://github.com/yaoyao-liu/e3bm](https://github.com/yaoyao-liu/e3bm)
Thanks!
Any questions?

Yaoyao Liu
yaoyao.liu@mpi-inf.mpg.de
References