Multi-Class Incremental Learning

Yaoyao Liu
liuyaoyao@tju.edu.cn
Outline

- Background
- Methods
- Experiments
- Takeaways
Background
Motivation

Thousands of new users and items everyday
Update the model with incremental data

Limited memory
Taking too long time to retrain the model

(Images from Internet)
Problem Definitions

Incremental learning (also lifelong learning, continual learning)
Problem Definitions

Incremental learning (also lifelong learning, continual learning)

10-class train

DNN classifier

10-class test

10-class train

10-class test
Problem Definitions

Incremental learning (also lifelong learning, continual learning)
Problem Definitions

Incremental learning (also lifelong learning, continual learning)

DNN classifier

10-class train

10-class train

10-class test

......
Problem Definitions

Incremental learning (also lifelong learning, continual learning)

Multi-task setting
- ImageNet train
- CUB train
- DNN classifier
- Test data

Multi-class setting
- ImageNet Class 1-10
- ImageNet Class 11-20
- ImageNet Class 21-30
- DNN classifier
- Test data

Test data
Problem Definitions

Incremental learning (also lifelong learning, continual learning)

Multi-task setting
- ImageNet train
- CUB train
- DNN classifier
- Test data

Multi-class setting
- ImageNet Class 1-10
- DNN classifier
- Test data

- ImageNet Class 11-20
- DNN classifier
- Test data

- ImageNet Class 21-30
- DNN classifier
- Test data

This talk
## Related Learning Methods

<table>
<thead>
<tr>
<th></th>
<th>Transfer learning</th>
<th>Multi-task learning</th>
<th>Multi-task incremental learning</th>
<th>Multi-class incremental learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target task(s)</strong></td>
<td>Single</td>
<td>Multiple</td>
<td>Multiple</td>
<td>Single</td>
</tr>
<tr>
<td><strong>Source task(s)</strong></td>
<td>Multiple</td>
<td>Multiple</td>
<td>Multiple</td>
<td>Single</td>
</tr>
<tr>
<td><strong>Data arrival</strong></td>
<td>Constantly / Once</td>
<td>Once</td>
<td>Constantly</td>
<td>Constantly</td>
</tr>
</tbody>
</table>
Challenges

1. **Catastrophic Forgetting**
   Model bias on the latest class group

2. **FC classifier is not extendable**
   10 classes -> 20 classes

3. **Memory resources may be limited**
   Not able to retain all previous samples
Catastrophic Forgetting

Distribution of DNN parameters

Joint distribution
Catastrophic Forgetting

ImageNet Class 1-10

DNN classifier

ImageNet Class 11-20

DNN classifier

ImageNet Class 21-30

DNN classifier

Distribution of DNN parameters

Overfit to the latest class group
**Literature Review**

- To improve the feature extractor
  
  LwF[1], ......

- To improve the classifier:
  
  iCaRL[4], BiC[3], ......

- To improve both:
  
  Hou et al.[5], EEIL[2]......

---

**Reference**

Methods
Learning without Forgetting (LwF)

Joint Training

Input:
- all images

Target:
- old classes’ ground truth
- new class’ ground truth

Learning without Forgetting

Input:
- new class images

Target:
- previous model’s response for old classes
- new class’ ground truth

Reference
Learning without Forgetting (LwF)

Idea: discouraging the old classes output to change [1]

Knowledge distillation

Proposed by Hilton et al.[2] for ensemble modeling

Classification
\[
L_{\text{new}}(y_n, \hat{y}_n) = -y_n \log \hat{y}_n
\]

Distillation
\[
L_{\text{old}}(y_o, \hat{y}_o) = -y_o \log \hat{y}_o
\]

Full objective
\[
L = L_{\text{old}} + L_{\text{new}}
\]

In which,
\[
y_o = \Phi_{\text{old}}(x)
\]
\[
y_n: \text{ground truth}
\]
\[
[\hat{y}_o, \hat{y}_n] = \Phi_{\text{current}}(x)
\]

Reference
Learning without Forgetting (LwF)

Summary

+ Distillation loss -> improve the learning of feature extractor
+ Don’t need to retain data for old classes

- Using a simple way to deal with the FC classifier without solving the bias problem

* This method is proposed for multi-task setting. However, it is usually used as a baseline of multi-class incremental learning papers

Reference

iCaRL: Incremental Classifier and Representation Learning

Algorithm 1 iCaRL CLASSIFY

\[
\text{input } x \quad \text{// image to be classified}
\]
\[
\text{require } \mathcal{P} = (P_1, \ldots, P_t) \quad \text{// class exemplar sets}
\]
\[
\text{require } \varphi : \mathcal{X} \rightarrow \mathbb{R}^d \quad \text{// feature map}
\]
\[
\text{for } y = 1, \ldots, t \text{ do}
\]
\[
\mu_y \leftarrow \frac{1}{|P_y|} \sum_{p \in P_y} \varphi(p) \quad \text{// mean-of-exemplars}
\]
\[
\text{end for}
\]
\[
y^* \leftarrow \text{argmin}_{y=1,\ldots,t} \| \varphi(x) - \mu_y \| \quad \text{// nearest prototype}
\]

\[
\text{output } \text{class label } y^*
\]

**Idea:** FC classifier -> nearest-mean-of-exemplars (NME) *NME is used only in test phase*

Reference
iCaRL: Incremental Classifier and Representation Learning

Algorithm 4 iCaRL \textsc{ConstructExemplarSet}

\textbf{input} image set $X = \{x_1, \ldots, x_n\}$ of class $y$

\textbf{input} $m$ target number of exemplars

\textbf{require} current feature function $\varphi: \mathcal{X} \to \mathbb{R}^d$

\[ \mu \leftarrow \frac{1}{n} \sum_{x \in X} \varphi(x) \]  // current class mean

\textbf{for} $k = 1, \ldots, m$ \textbf{do}

\[ p_k \leftarrow \text{argmin}_{x \in X} \| \mu - \frac{1}{k} [\varphi(x) + \sum_{j=1}^{k-1} \varphi(p_j)] \| \]

\textbf{end for}

$P \leftarrow (p_1, \ldots, p_m)$

\textbf{output} exemplar set $P$

Reference
iCaRL: Incremental Classifier and Representation Learning

Summary

+ Solving the bias problem for the classifier

- Need to retain parts of old data
- Non-parametric classifier may fail in some novel similar classes
- Training and testing using different types of classifier (train: fc, test: NME)

Reference
End-to-end Incremental Learning (EEIL)

Reference
End-to-end Incremental Learning (EEIL)

Summary

+ End-to-end, improvement on both feature extractor and classifier
+ A series of data augmentation techniques

- Improvements may come from tricks

Reference
Large Scale Incremental Learning (BiC)

**Problem: bias on novel classifier**

![Diagram showing bias correction](image)

**BiC: Bias Correction**

$$q_k = \begin{cases} o_k & 1 \leq k \leq n \\ \alpha o_k + \beta & n + 1 \leq k \leq n + m \end{cases}$$

$$L_b = -\sum_{k=1}^{n+m} \delta_{y=k} \log[\text{softmax}(q_k)]$$

Reference

Large Scale Incremental Learning (BiC)

Summary

+ Solve the bias problem on classifier

- The correction function works only on large scale datasets

Reference
Learning a Unified Classifier Incrementally via Rebalancing (Hou et al.)

### Cosine normalization

- **Imbalanced Magnitudes**
  - Old class embeddings
  - New class embeddings

- **Cosine Normalization**

### FC classifier

\[
p_i(x) = \frac{\exp(\theta_i^T f(x) + b_i)}{\sum_j \exp(\theta_j^T f(x) + b_j)}
\]

### Cosine distance

\[
p_i(x) = \frac{\exp(\eta \langle \theta_i, f(x) \rangle)}{\sum_j \exp(\eta \langle \theta_j, f(x) \rangle)}
\]

### Improve classifier

Reference

Learning a Unified Classifier Incrementally via Rebalancing (Hou et al.)

**Less-forget constraint**

\[ L^C_{\text{dis}}(x) = - \sum_{i=1}^{|C_o|} \left\| \langle \tilde{\theta}_i, \bar{f}(x) \rangle - \langle \tilde{\theta}_i^*, \bar{f}^*(x) \rangle \right\| \]

\[ L^G_{\text{dis}}(x) = 1 - \langle \bar{f}^*(x), \bar{f}(x) \rangle \]

**Distillation loss**

**Cosine distance of feature**

**Improve feature extractor**

Reference

Learning a Unified Classifier Incrementally via Rebalancing (Hou et al.)

*Inter-class separation*

Add margin threshold to top-K classes

\[
L_{mr}(x) = \sum_{k=1}^{K} \max(m - \langle \bar{\theta}(x), \bar{f}(x) \rangle + \langle \bar{\theta}^k, \bar{f}(x) \rangle, 0)
\]

Improve classifier

Reference

Learning a Unified Classifier Incrementally via Rebalancing (Hou et al.)

Summary

+ Improvement on both classifier (cosine distance, inter-class separation) and feature extractor (less-forgot constraint)

- The first group requires more classes than other groups (require a good initialization for the CONV networks)
- Extremely slow with inter-class separation strategy

Reference
## Comparison

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Distillation Loss</td>
<td>Distillation Loss</td>
<td>Distillation Loss</td>
<td>Distillation Loss</td>
<td>Distillation Loss</td>
<td>Less-Forget Constraint</td>
</tr>
<tr>
<td>Exemplar, Balanced Fine-tuning</td>
<td>Exemplar</td>
<td>Exemplar</td>
<td>Exemplar</td>
<td>Exemplar,</td>
<td></td>
</tr>
<tr>
<td>Classifier</td>
<td>FC</td>
<td>NME</td>
<td>FC</td>
<td>FC, Bias Correction</td>
<td>FC, Cosine Normalization, Inter-Class Separation</td>
</tr>
</tbody>
</table>

### Reference

Experiments
Datasets and Benchmark

Datasets: CIFAR100, ImageNet-Sub (100 classes subset), ImageNet

E.g. CIFAR-100

Number of class groups

- 2 groups
  - 50 classes
  - 50 classes

- 10 groups
  - 10 10 10 10 10
  - 10 10 10 10 10
Experiments on CIFAR-100

(a) 20 groups
(b) 10 groups
(c) 5 groups
(d) 2 groups
Experiments on ImageNet

(a) ImageNet-100

(b) ImageNet-1000

Accuracy (%) vs. Number of classes for different methods: LwF, iCaRL, EEIL, BIC, UpperBound.
Confusion Matrices
Takeaways

Important techniques:
1. Distillation Loss -> retain knowledge for old classes
2. Nearest-Mean-of-Exemplar Classifier -> no-bias classifier

Future work:
1. Other strategy for retaining knowledge for old classes
2. Shareable parametric classifier -> meta-learning?
Thanks!
Any questions?

liuyaoyao@tju.edu.cn