

**Introduction to
iCaRL
Incremental Classifier and Representation Learning
(CVPR2017)**

ChangJun Ahn
Hanbat National University Dept. of Electronic Engineering
qkqk1009@gmail.com
July 4, 2023

논문 소개 : iCaRL Incremental Classifier and Representation Learning (CVPR 2017)

- 본 논문에서는 class-incremental learning 상황에서 효과적인 학습 전략인 iCaRL을 제안합니다.
- Class-incremental learning은 새로운 클래스가 점진적으로 추가되는 상황을 가정합니다.

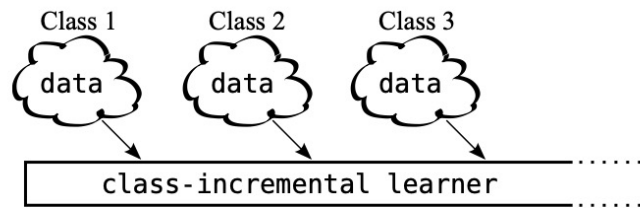


Figure 1: Class-incremental learning: an algorithm learns continuously from a sequential data stream in which new classes occur. At any time, the learner is able to perform multi-class classification for all classes observed so far.

Continual Learning의 당면한 상황

- 문제 1. Catastrophic Forgetting : Neural Network가 새로운 지식을 배울 때 이전에 배웠던 지식을 쉽게 잊게 됩니다.
- 문제 2. 메모리를 포함한 컴퓨팅 자원은 한정적입니다. 무한정 메모리를 증가시킬 수 없습니다.

iCaRL의 소개

1. Nearest-mean-of-exemplars
2. Prioritized exemplar selection
3. Representation learning

iCaRL 1. Nearest-Mean-of-Exemplars Classification

- $P = (P_1 \dots P_t)$: 1번 부터 t 번 까지의 각 클래스에 포함되는 *exemplar* 이미지 집합(각 클래스당 $\frac{K}{t} = m$ 개의 이미지 저장)
- iCaRL의 분류 알고리즘 요약
 1. 각 클래스마다 저장된 exemplar들의 feature에 대한 평균값(mean)을 계산합니다.
 2. 한 장의 이미지 x 가 주어졌을 때, feature를 계산하여 가장 가까운 클래스를 반환합니다.
- iCaRL에서는 총 K 개의 이미지를 저장할 수 있는 저장 공간을 준비합니다.

Algorithm 1 iCaRL CLASSIFY

```

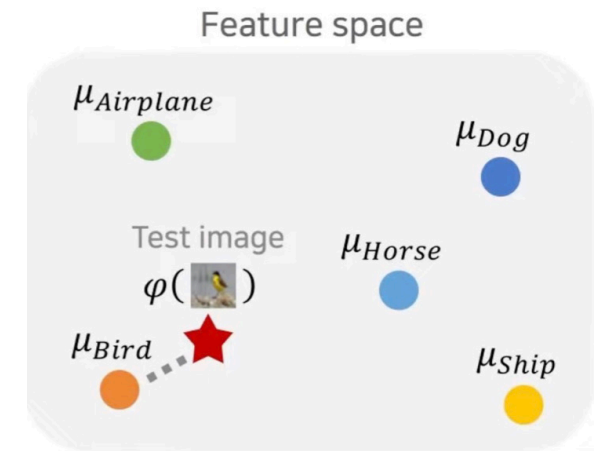
input  $x$  // image to be classified
require  $\mathcal{P} = (P_1, \dots, P_t)$  // class exemplar sets
require  $\varphi : \mathcal{X} \rightarrow \mathbb{R}^d$  // feature map
for  $y = 1, \dots, t$  do
     $\mu_y \leftarrow \frac{1}{|P_y|} \sum_{p \in P_y} \varphi(p)$  // mean-of-exemplars
end for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

[Train]
Representation
learning



CNN

[Test]
Find nearest
neighbor



iCaRL 2. Training Procedure

- iCaRL 의 학습 알고리즘 요약
 - 새로운 데이터와 이전까지 저장한 k 개의 이미지를 이용해 representation learning을 진행합니다.
 - 학습 이후 각 클래스마다 $k/t = m$ 개의 이미지만 남깁니다.
- (Assume) 총 k 개의 이미지만을 저장할 수 있는 공간이 있습니다.

(새로운 데이터 + k 개의 저장된 데이터)로 학습,
각 데이터 마다 m 개의 데이터만 남김

기존 클래스의 exemplar 크기 줄이기

새로운 클래스의 exemplar 만들기

Algorithm 2 iCaRL INCREMENTALTRAIN

```

input  $X^s, \dots, X^t$  // training examples in per-class sets
input  $K$  // memory size
require  $\Theta$  // current model parameters
require  $\mathcal{P} = (P_1, \dots, P_{s-1})$  // current exemplar sets

 $\Theta \leftarrow \text{UPDATEREPRESENTATION}(X^s, \dots, X^t; \mathcal{P}, \Theta)$ 
 $m \leftarrow K/t$  // number of exemplars per class
for  $y = 1, \dots, s-1$  do
     $P_y \leftarrow \text{REDUCEEXEMPLARSET}(P_y, m)$ 
end for
for  $y = s, \dots, t$  do
     $P_y \leftarrow \text{CONSTRUCTEXEMPLARSET}(X_y, m, \Theta)$ 
end for
 $\mathcal{P} \leftarrow (P_1, \dots, P_t)$  // new exemplar sets
    
```

iCaRL 3. Representation Learning

- iCaRL의 학습 알고리즘 요약

- (새로운 데이터 + 저장된 k개의 이미지)로 데이터 세트를 구성합니다.
- 전체 데이터 세트에 대하여 분류 결과를 임시로 저장합니다.
 - 이때 새로운 데이터를 포함한 모든 데이터를 입력으로 넣습니다.
 - 이전까지의 클래스에 대한 결과를 계산합니다. (새 클래스 제외)
- 두 가지 loss function을 이용하여 네트워크를 업데이트 합니다.
 - Classification loss** : 새로운 클래스에 대해 학습합니다.
 - Distillation loss** : 이전 클래스에 대해서는 기존 기능을 유지합니다.

- 각 클래스를 개별적으로 binary classification 하도록 학습합니다.

- 네트워크의 파라미터 θ
- Feature extractor $\varphi : X : R^d$

- 각 클래스에 대한 분류 결과 $g_y(x) = \frac{1}{1 + \exp(-w_y^T \varphi(x))}$

Algorithm 3 iCaRL UPDATERPRESENTATION

input X^s, \dots, X^t // training images of classes s, \dots, t
require $\mathcal{P} = (P_1, \dots, P_{s-1})$ // exemplar sets
require Θ // current model parameters
 // form combined training set:

$$\mathcal{D} \leftarrow \bigcup_{y=s, \dots, t} \{(x, y) : x \in X^y\} \cup \bigcup_{y=1, \dots, s-1} \{(x, y) : x \in P^y\}$$

 // store network outputs with pre-update parameters:
for $y = 1, \dots, s-1$ **do**
 $q_i^y \leftarrow g_y(x_i)$ for all $(x_i, \cdot) \in \mathcal{D}$
end for
 run network training (e.g. BackProp) with loss function

$$\ell(\Theta) = - \sum_{(x_i, y_i) \in \mathcal{D}} \left[\sum_{y=s}^t \delta_{y=y_i} \log g_y(x_i) + \delta_{y \neq y_i} \log(1 - g_y(x_i)) \right]$$

$$+ \sum_{y=1}^{s-1} \left[q_i^y \log g_y(x_i) + (1 - q_i^y) \log(1 - g_y(x_i)) \right]$$

 that consists of *classification* and *distillation* terms.

Conclusion

- Class-incremental learning에 효과적인 **iCaRL** 학습 전략을 제안합니다.
 - Classifier와 feature representation을 동시에 학습합니다.
- iCaRL은 다음과 같은 요소로 구성됩니다.
 1. A nearest-mean-of-exemplars
 2. Herding-based step for prioritized exemplar selection
 3. Representation learning step : distillation을 사용해 catastrophic forgetting 문제를 개선합니다.
- 핵심적인 데이터 위주로 데이터를 저장하므로, 메모리를 절약할 수 있습니다.
 - 모든 클래스가 동시에 학습될 수 있는 batch setting보다는 성능이 낮은 한계가 있습니다.
 - 추후에 이미지 원본을 저장하지 않고, 오토 인코더를 이용해 이미지 인코딩된 정보를 저장하는 방식을 사용할 수 있습니다.
 - 보안이 중요시되는 상황에서는 조금의 데이터도 저장할 수 없는 경우가 존재합니다.