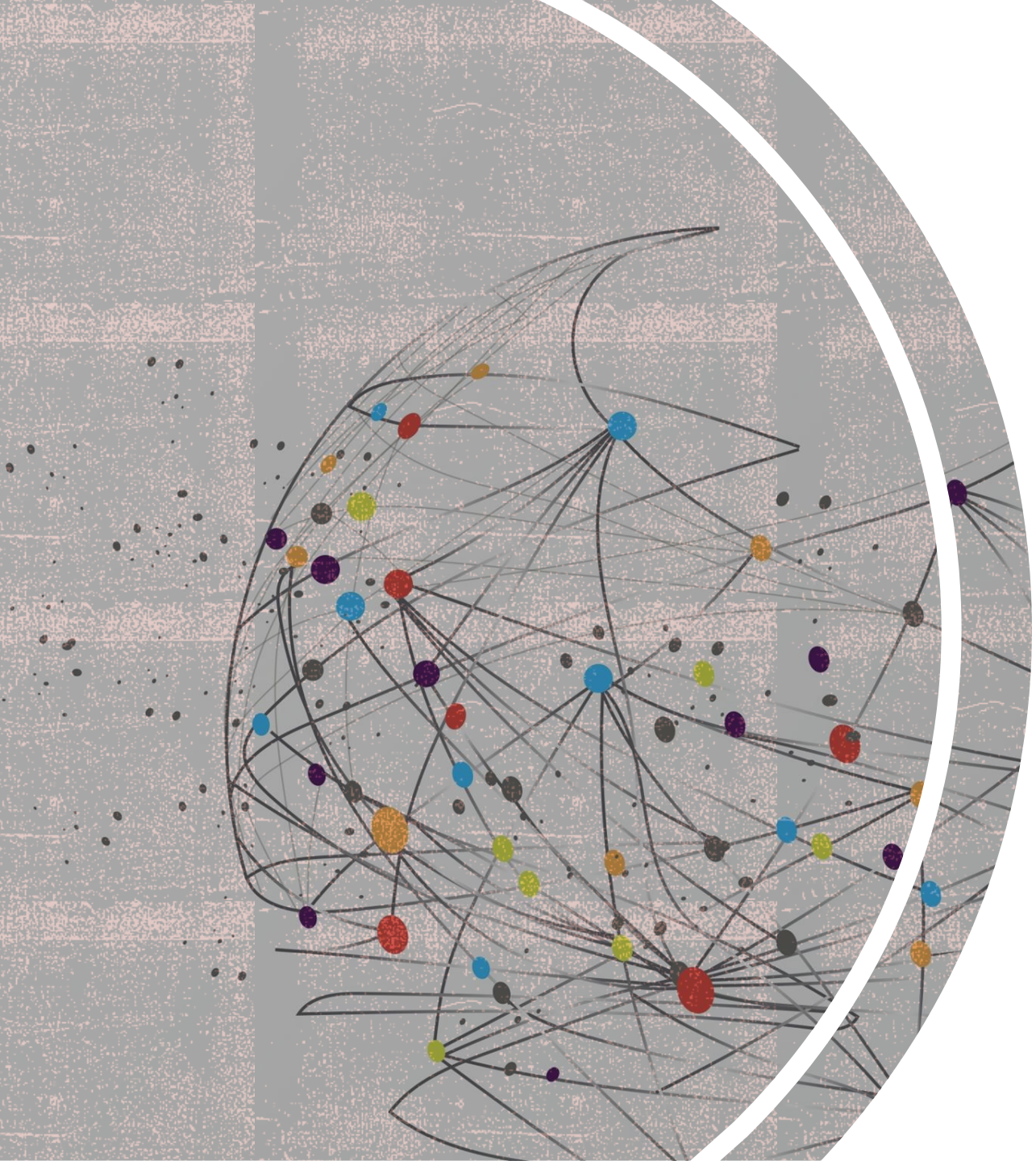


서울과학기술대학교
SEOULTECH SEOUL NATIONAL UNIVERSITY OF SCIENCE & TECHNOLOGY

Big Data Management and Application Laboratory

ICARL: INCREMENTAL CLASSIFIER AND REPRESENTATION LEARNING

Presentor: Minseon kim

BACKGROUND

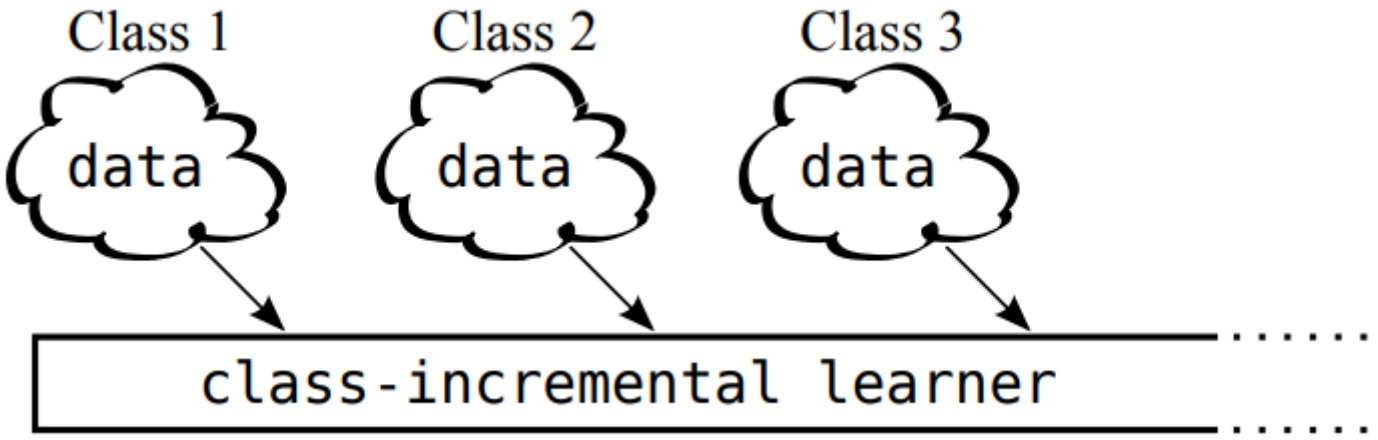
- Natural vision system are inherently incremental
  : 새로운 정보가 기존의 정보를 유지함과 동시에 지속적으로 통합된다.

- 하지만, 대부분의 artificial object는 모든 클래스에 대한 정보를 선행적으로 주어져 접근가능하다는 조건하에서 배치 형태로 학습이 된다.

- 따라서, 새로운 클래스에 대한 정보가 생성되었을 때 학습 가능한 ***class-incremental learning***이 필요하다.

BACKGROUND

- Class-incremental의 3가지 조건
  1) 서로 다른 클래스가 다른 시간대에 발생하는 data stream로 부터 학습가능해야함
  2) 특정시점에서 현재까지 관찰된 클래스에 대한 분류 가능해야함
  3) Computational requirements와 메모리 사용량이 일정한 수준에서 유지되어야함



INTRODUCTION

- Naively, class-incremental data stream으로 부터 SGD optimization을 이용해서 classifier를 재학습 시킬 수 있다. 하지만 이 경우, 정확도가 빠른 속도로 악화하는 양상을 보인다. (catastrophic forgetting)

- 본 논문에서는 classifier와 동시에 feature representation을 학습할 수 있는 iCaRL 학습 방법론을 제안한다.
  1) Classification by a nearest-mean-of-exemplars rule
  2) Prioritized exemplar selection based on herding
  3) Representation learning using knowledge distillation and prototype rehersal

CLASS-INCREMENTAL LEARNING

- Classifier과 동시에 Feature Representation을 학습

Algorithm 1 iCaRL CLASSIFY <Classification>

input x // image to be classified
require $\mathcal{P} = (P_1, \ldots, P_t)$ // class exemplar sets
require φ // feature map
Conv layers
for $y = 1, \ldots, t$ **do** t개의 class
$\mu_y \leftarrow \frac{1}{|P_y|} \sum_{p \in P_y} \varphi(p)$ // mean-of-exemplars
Prototype vector
Feature vector
end for
분류하고자 하는 이미지의 feature vector
$y^* \leftarrow \underset{y=1,\ldots,t}{\operatorname{argmin}} \|\varphi(x) - \mu_y\|$ // nearest prototype
output class label y^* 가장 가까운 prototype으로 assign

$$g_y(x) = \frac{1}{1 + \exp(-a_y(x))} \quad \text{with} \quad a_y(x) = w_y^\top \varphi(x)$$

Output은 확률로 나오게 되지만, 이는 단지 representation learning의 일부 (classification X)

Algorithm 2 iCaRL INCREMENTALTRAIN <Training>
새로운 클래스가 available할때의 routine

input $X^s, \ldots, X^t$ // training examples in per-class sets
input K // memory size
require Θ // current model parameters
require $\mathcal{P} = (P_1, \ldots, P_{s-1})$ // current exemplar sets
$\Theta \leftarrow$ UPDATEREPRESENTATION$(X^s, \ldots, X^t; \mathcal{P}, \Theta)$
네트워크 parameter 업데이트
$m \leftarrow K/t$ // number of exemplars per class
for $y = 1, \ldots, s-1$ **do**
$P_y \leftarrow$ REDUCEEXEMPLARSET(P_y, m)
end for
for $y = s, \ldots, t$ **do** exemplars 업데이트
$P_y \leftarrow$ CONSTRUCTEXEMPLARSET(X_y, m, Θ)
end for
새로운 클래스에 대한 exemplar sets
$\mathcal{P} \leftarrow (P_1, \ldots, P_t)$ // new exemplar sets

NEAREST-MEAN-OF-EXEMPLARS CLASSIFICATION

- 새로운 이미지에 대한 라벨을 생성하기 위해서, 각 클래스 별로 해당 클래스에 속한 exemplars들에 대한 feature vector를 평균내어 prototype vector를 구한다. 또한 새로운 image에 대한 feature vector를 계산하여 그차이를 비교한다.
- 기존의 multiclass-classification 학습 방식과의 비교
  : weight vector 값이 φ의 값과 분리되는 문제점

$$\operatorname{argmax}_y g_y(x) = \operatorname{argmax}_y w_y^\top \varphi(x)$$

Weight vector

feature extraction routine

- 반면, 채택한 방식은 weight-vector를 분리 하지 않는다. Prototype vector (good approximation to the class mean)의 지속적인 변화 반영.

$$y^* = \operatorname*{argmin}_{y=1,\dots,t} \|\varphi(x) - \mu_y\| = \operatorname{argmax}_y \mu_y^\top \varphi(x)$$

Automatically change whenever the feature representation chagnes

REPRESENTATION LEARNING

- 새로운 클래스가 들어왔을 때, feature extraction routine의 업데이트가 이루어진다.(Algorithm 3)

1. Augmented train set (new+stored)
2. Resulted output for current network is stored
3. Nework parameters updated

- Catastrophic forgetting 완화를 위한 기존 fine-tuning 대비 두가지 변화
  1) Augmented training set
  2) Augmented loss

Algorithm 3 iCaRL UPDATEREPRESENTATION — New classes

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

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

// store network outputs with pre-update parameters:
for $y = 1, \ldots, s-1$ **do** (2)
$q_i^y \leftarrow g_y(x_i)$ for all $(x_i, \cdot) \in \mathcal{D}$
end for
run network training (*e.g.* BackProp) with loss function (3)

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

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

EXEMPLAR MANAGEMENT

- 새로운 클래스가 들어왔을 때, exemplar set의 조정이 이루어진다.(Algorithm 4, 5)

1. 새로운 클래스 exemplar 선정
2. 기존 클래스 exemplar sets 사이즈 축소

- Two objectives
  1) 초기 exemplar set은 실제 클래스의 평균 벡터에 근사해야한다.
  2) 앞선 1) 조건을 위배하지않고, 어느 순간에서도 exemplar의 제거가 가능해야 한다.

- Prioritized exemplar selection from distribution data with iterative selection results in high approximation of the mean vector with fewer samples

Algorithm 4 iCaRL CONSTRUCTEXEMPLARSET

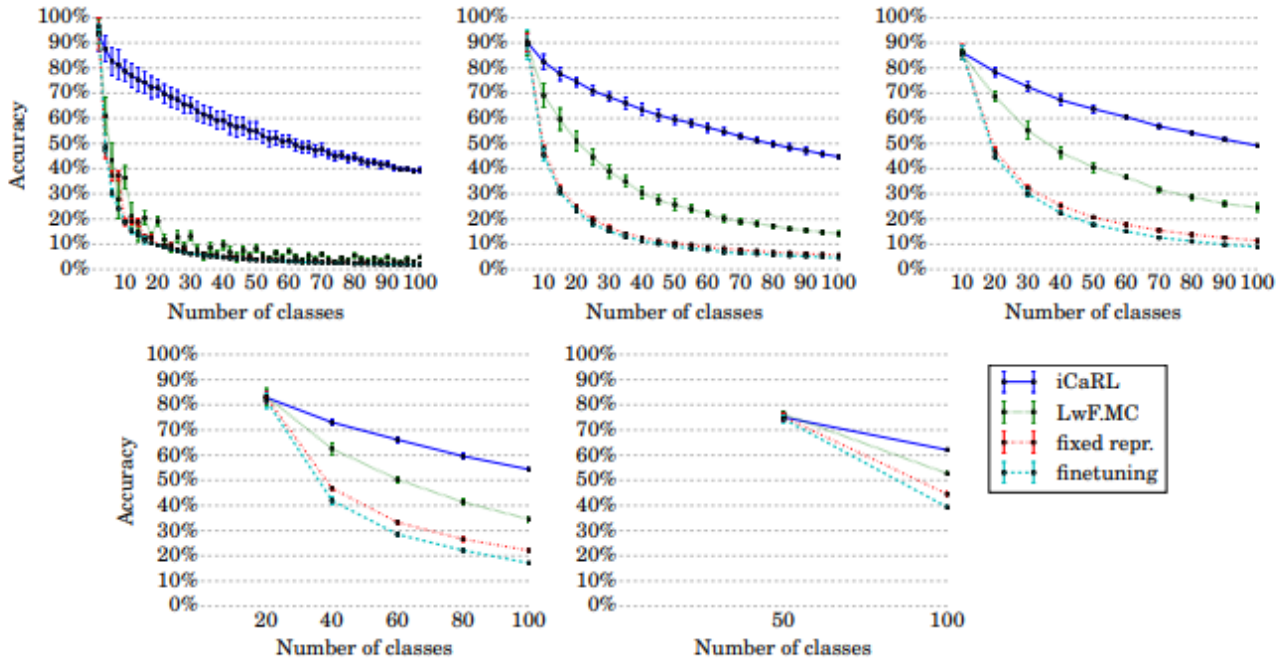
input image set $X = \{x_1, \ldots, x_n\}$ of class y
input m target number of exemplars
require current feature function $\varphi : \mathcal{X} \to \mathbb{R}^d$
  $\mu \leftarrow \frac{1}{n} \sum_{x \in X} \varphi(x)$ // current class mean
  for $k = 1, \ldots, m$ **do**
    $p_k \leftarrow \underset{x \in X}{\operatorname{argmin}} \left\| \mu - \frac{1}{k} [\varphi(x) + \sum_{j=1}^{k-1} \varphi(p_j)] \right\|$
  end for
  $P \leftarrow (p_1, \ldots, p_m)$
output exemplar set P

Algorithm 5 iCaRL REDUCEEXEMPLARSET

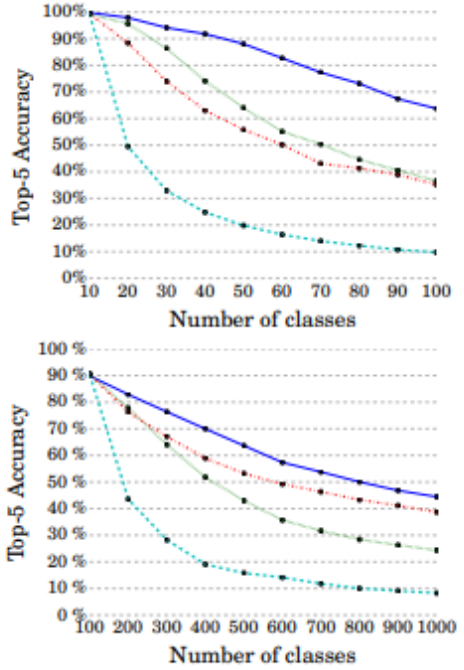
input m // target number of exemplars
input $P = (p_1, \ldots, p_{|P|})$ // current exemplar set
  $P \leftarrow (p_1, \ldots, p_m)$ // *i.e.* keep only first m
output exemplar set P

EXPERIMENTS

- Curves of the classification accuracies after each batch of classes
  - Evaluate with same test data (다만, 현재까지 학습된 클래스만을 대상으로)



(a) Multi-class accuracy (averages and standard deviations over 10 repeats) on iCIFAR-100 with 2 (top left), 5 (top middle), 10 (top right), 20 (bottom left) or 50 (bottom right) classes per batch.



100 classes in batches of 10

1000 classes in batches of 100

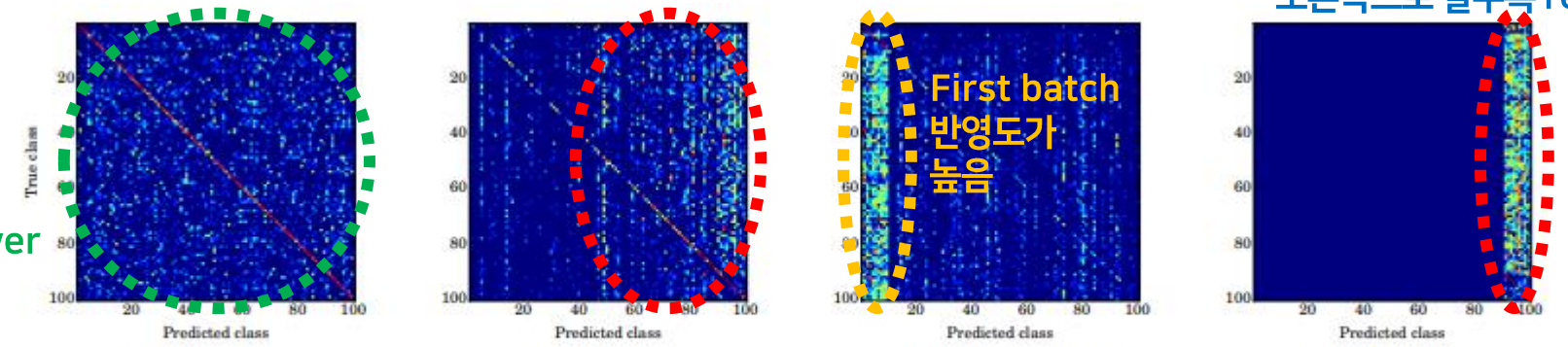
(b) Top-5 accuracy on iILSVRC-small (top) and iILSVRC-full (bottom).

RESULTS

- 4가지 방법론에 대한 classification accuracy의 비교 분석
  1) Finetuning: catastrophic forgetting 고려하지않고 네트워크 재조정
  2) Fixed Representation: prevent catastrophic forgetting (feature representation을 첫번째 배치 이후에 freeze, classification layer를 클래스의 학습후에 freeze)
  3) LwF.MC: exemplar set을 사용하지 않지만 network classifier에서의 distillation loss를 고려. Network output을 그대로 사용함.

오른쪽으로 갈수록 recently learned classes

distributed close to uniformly over all classes



(a) iCaRL (b) LwF.MC (c) fixed representation (d) finetuning

Bias towards classes from latest batch

Figure 3: Confusion matrices of different method on iCIFAR-100 (with entries transformed by $\log(1+x)$ for better visibility).

DIFFERENTIAL ANALYSIS

- Isolate individual aspects of the methods for futher insight
  1. 정확도 향상에 영향을 미친 요소 분석
     a) mean-of-exemplars classification rule
     b) Representation 학습을 위한 exemplars set의 활용
     c) Distillation loss의 활용
  2. Nearest-class-mean 알고리즘 대신 Menas-of-exemplars의 알고리즘을 사용한 결과 얼마나 정확도의 손실이 발생하였는가.

전체 데이터를 필요로 하므로 incremental한 방식이 아니다. 즉, incremental한 알고리즘의 구현이 batch learning에 얼마나 근접한 결과를 내는지를 비교하는것을 목적으로 함.

(a) Switching off different components of iCaRL (*hybrid1*, *hybrid2*, *hybrid3*, see text for details) leads to results mostly inbetween iCaRL and LwF.MC, showing that all of iCaRL's new components contribute to its performance.

batch size	iCaRL	*hybrid1*	*hybrid2*	*hybrid3*	LwF.MC
2 classes	57.0	36.6	57.6	57.0	11.7
5 classes	61.2	50.9	57.9	56.7	32.6
10 classes	64.1	59.3	59.9	58.1	44.4
20 classes	67.2	65.6	63.2	60.5	54.4
50 classes	68.6	68.2	65.3	61.5	64.5

Mean-of-exmplar 알고리즘은 작은 배치 사이즈에서 더 큰 효과

Distillation loss는 큰 배치 사이즈에서 advantageous

Exemplar set의 Catastrophic Forgetting 효과

(b) Replacing iCaRL's mean-of-exemplars by a nearest-class-mean classifier (NCM) has only a small positive effect on the classification accuracy, showing that iCaRL's strategy for selecting exemplars is effective.

batch size	iCaRL	NCM
2 classes	57.0	59.3
5 classes	61.2	62.1
10 classes	64.1	64.5
20 classes	67.2	67.5
50 classes	68.6	68.7

Only minor differences

DIFFERENTIAL ANALYSIS

- Effect of different memory budgets (parameter K와 관련)
  : 10000이상의 충분한 prototypes가 주어졌을때 NCM과 유사한 성능을 보여줌.

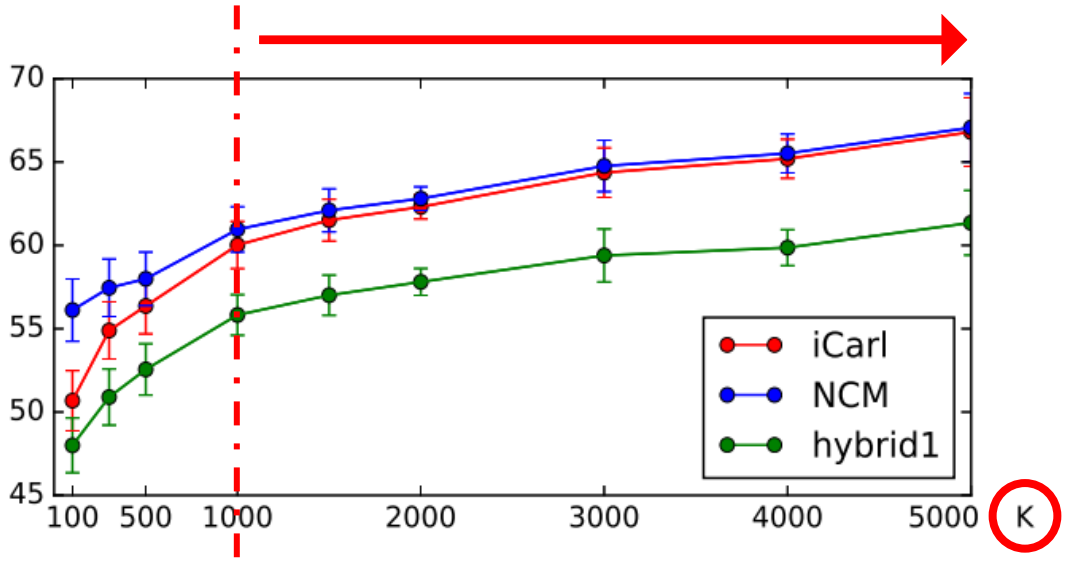
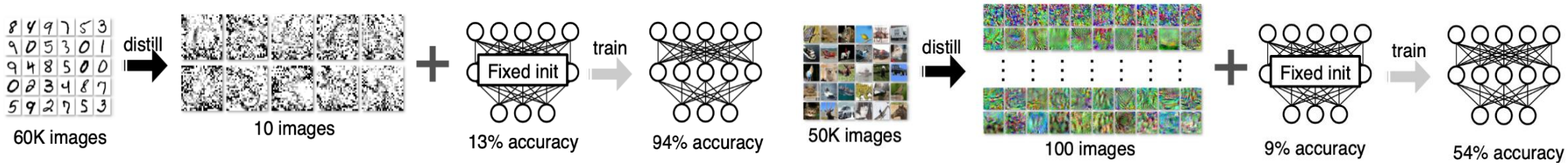


Figure 4: Average incremental accuracy on iCIFAR-100 with 10 classes per batch for different memory budgets K.

IMPRESSION

- ✓ Incremental learning을 위해서 *data distillation*의 기법의 활용 가능성
  -> computational cost remain bounded with increasing data size & benefit from the increase of memory budget



- ✓ Catastrophic forgetting의 효과를 prevent함과 동시에 효율적인 representation learning을 위해 *conditional computations*의 concept의 확장 가능성