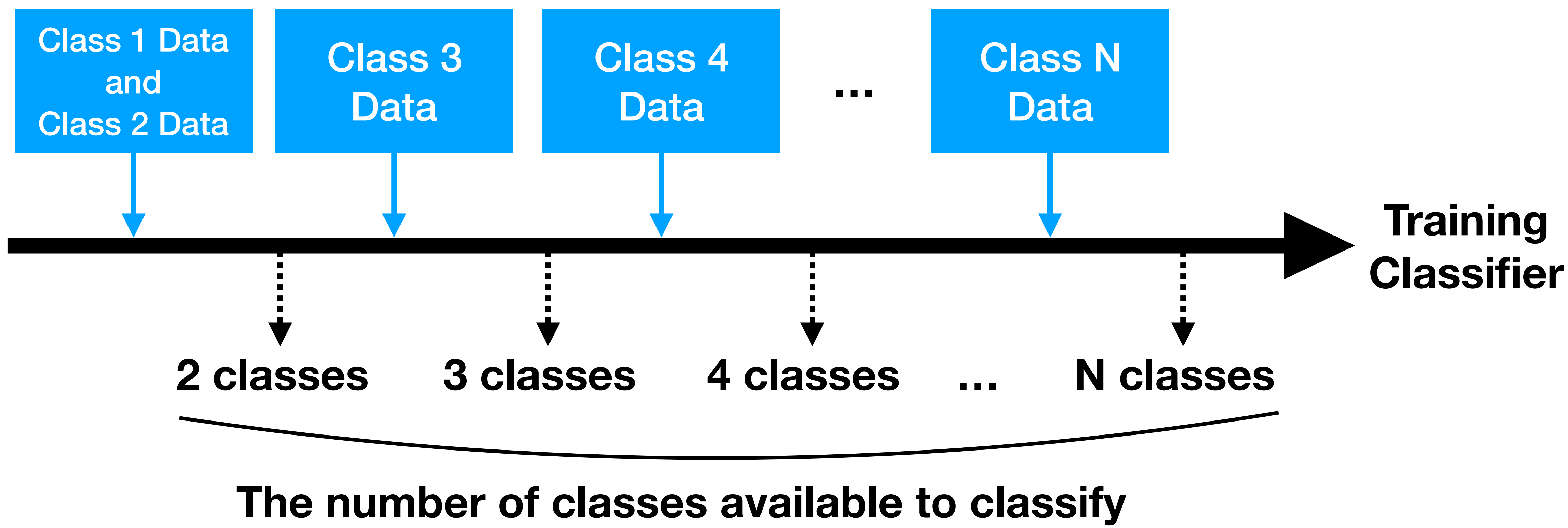
CRIL: Compact Representation for Incremental Learning

Presented by Youngki Hong, Seongmin Lee, Minsuk Choi

Contents

Incremental Learning



- **Class-incremental learning** is the scheme of learning that is able to learn about new classes, when the training data for new classes is available.

Incremental Learning

- There are 3 properties of an algorithm to qualify as class-incremental.

  i) Trainability for a class-incremental data stream.

  ii) Competitive performance as a multi-class classifier for the classes observed so far, at any time during the training.

  iii) Bounded computational requirements and memory footprint.

Incremental Learning

- The most of the artificial object recognition systems can only be trained in a batch setting where **all object classes are known**.

- Training the classifiers with class-incremental data streams incurs quick deterioration of accuracy, which is known as **catastrophic forgetting**.

- Existing solutions are limited to the case with the fixed data representation. **They cannot learn classifiers and feature representations simultaneously**.

iCaRL

- **iCaRL, incremental classifier and representation learning,** provides a practical strategy for simultaneous learning of classifiers and a feature representation in the class-incremental setting.

- There are 3 main components.
  - Classification by **a nearest-mean-of-exemplars** rule.
  - **Prioritized exemplar selection** based on examples from observed classes.
  - Representation learning using knowledge distillation and prototype rehearsal.

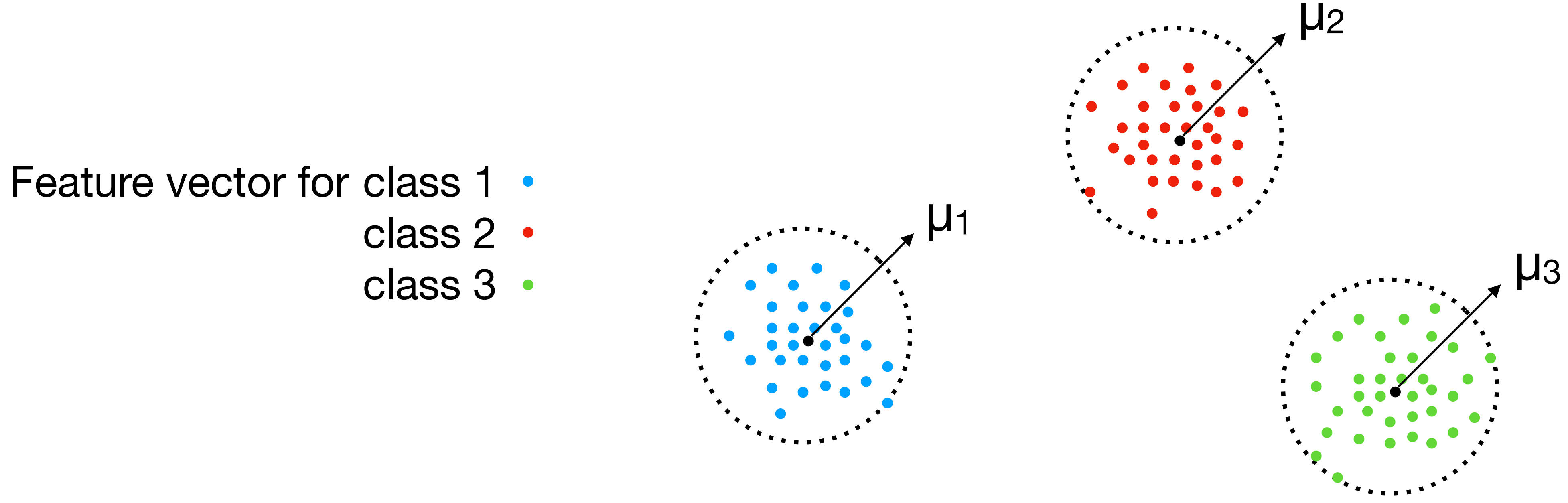
- iCaRL uses **naive linear classifier** for representation learning.

CRIL

- We presents **compact representation for incremental learning, CRIL,** using variational auto encoder, VAE, for representation learning.

- With VAE, the network is trained to have the output features that follow the gaussian distribution with given mean μ_i and variance σ_i^2 for class i.

- The network is expected to generate the output feature vectors which are more compact within each class and distinctive between classes.



Class-Incremental Classifier Learning

- Classification
  - Classifying by **nearest-mean-of-exemplars** algorithm.
  - Using **the sets of exemplar images** for each class observed so far: if there are 5 observed classes so far, there are 5 exemplar sets.
- Training
  - Using batches of classes observed so far.
  - Updating on representation, exemplar set occurs when the data with new classes becomes available.

Class-Incremental Classifier Learning

- Architecture
  - In case with **linear classifier**, using CNN for feature extractor and single classification layer to classify in classes observed so far for training.
  - In case with **VAE**, using encoder layer to generate the output features following certain gaussian distributions for each classes observed so far for training.
  - Using network only for representation learning.

- Resource Usage
  - Considering finite number of classes.
  - Using limited number of exemplar to limit the resource.

Nearest-Mean-of-Exemplars Classification

- Predict a label y* of the input image x by compare the distances between the feature vector of the image x and the average feature vector of all exemplars for a class y.

$$y^* = \underset{y=1,\ldots,t}{\operatorname{argmin}} \|\varphi(x) - \mu_y\|$$

- **x**: an input image
- **μ_y**: the average feature vector for all exemplars for a class y
- **y***: a label for a image x
- **φ**(x): a feature vector of x

Exemplar Management

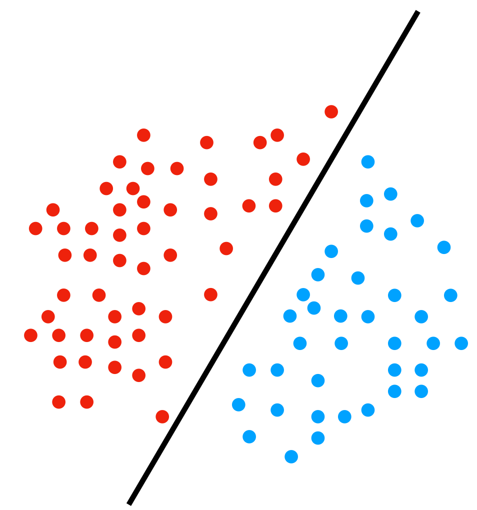
- Update exemplar set when new classes become available.
- For K exemplars, divide them into number of classes.
  - If there are t observed classes so far, $\frac{K}{t}$ exemplars for each class.
- Select exemplars for new classes: add examples which causes **best approximation of the average feature vector** over all training examples.
- Reduce exemplars from old classes: discard exemplars from the end of the set of each class.

Representation Learning

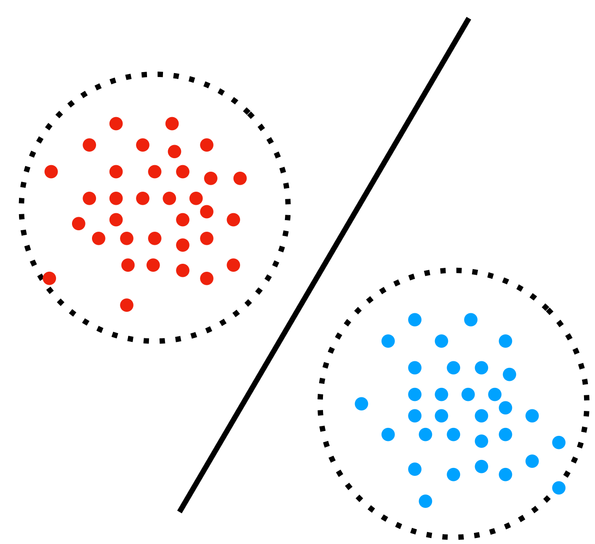
- When new classes are available, the feature extractor is updated in following three steps.
  - Construct an augmented training set with **the current training examples** and **the stored exemplars**.
  - Evaluate current network for each example for old classes.
  - Update the network parameters by minimizing a loss function with a **classification loss** for new classes and a **distillation loss** for previous classes.

Representation Learning

- With naive linear classifier, the network is trained to generate the output feature vectors to be linearly separable.



- With variational auto encoder, the network is trained to generate the output feature vectors following the distinctive gaussian distribution for each of the observed class.



Experiments

- Using MNIST and CIFAR-100 as data sets.
- iCaRL is constructed with ResNet; CRIL is constructed with MLP.
- Experiment 1
  - Train iCaRL network with K=100, 200, 500, 1000 exemplars.
  - To evaluate the performance changing as the number of exemplar changing.

Experiments

- Experiment 2
  - Train CRIL with K=100, 200, 500, 1000 exemplars.
  - Give μ_i and variance σ_i^2 for class i as following.

$$\mu_i = [v_j],\ \mu_i \in \mathbb{R}^n,\ v_j = \begin{cases} a & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \text{ for } i = 1, 2, \ldots, n,$$

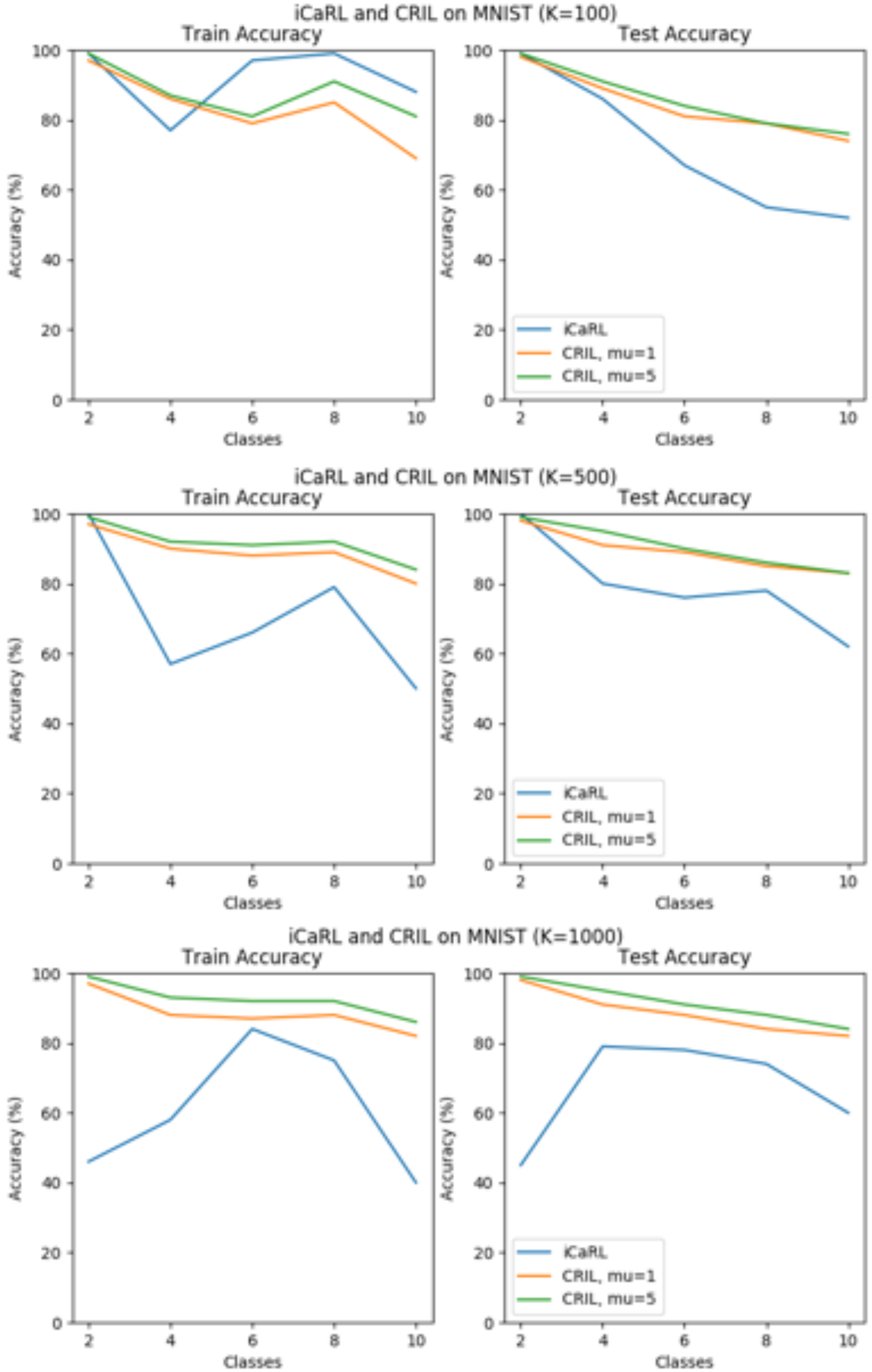
$$\sigma_i = I,\ \sigma_i \in \mathbb{R}^n,\ i = 1, 2, \ldots, n,$$

where $a = 1, 5$, $n =$ the number of the observed classes

Experiments

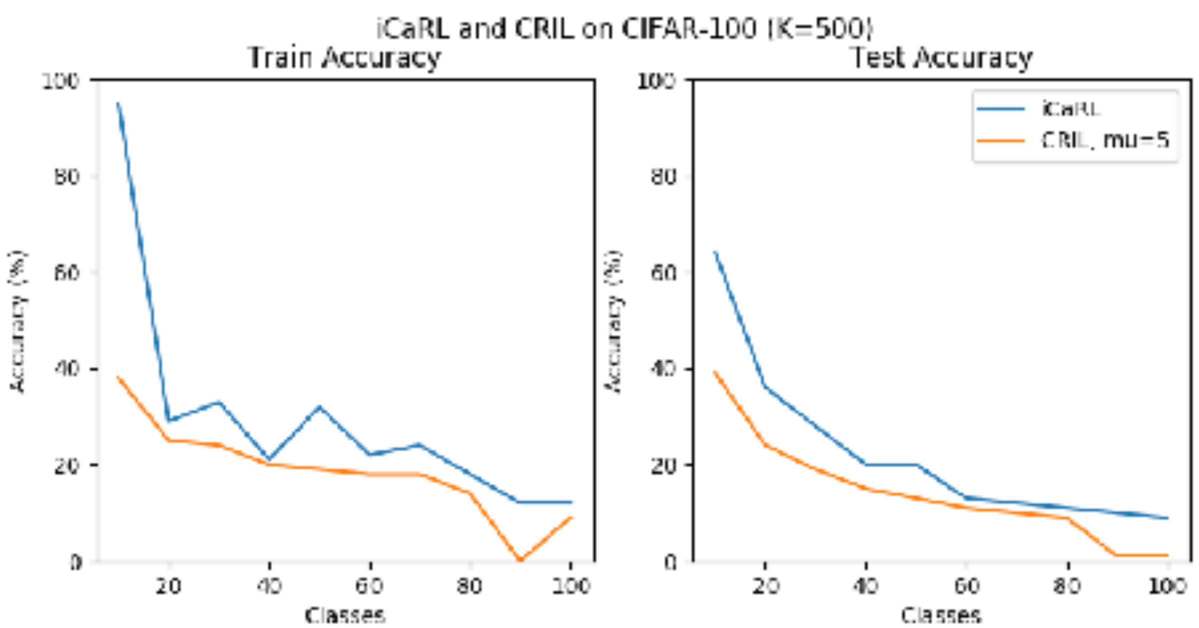
- Experiment 3 (will be on report)
  - Compare the distribution of the output feature vectors from previous experiments by t-sne visualization.
  - To compare the compactness of the output feature vectors for iCaRL and CRIL.

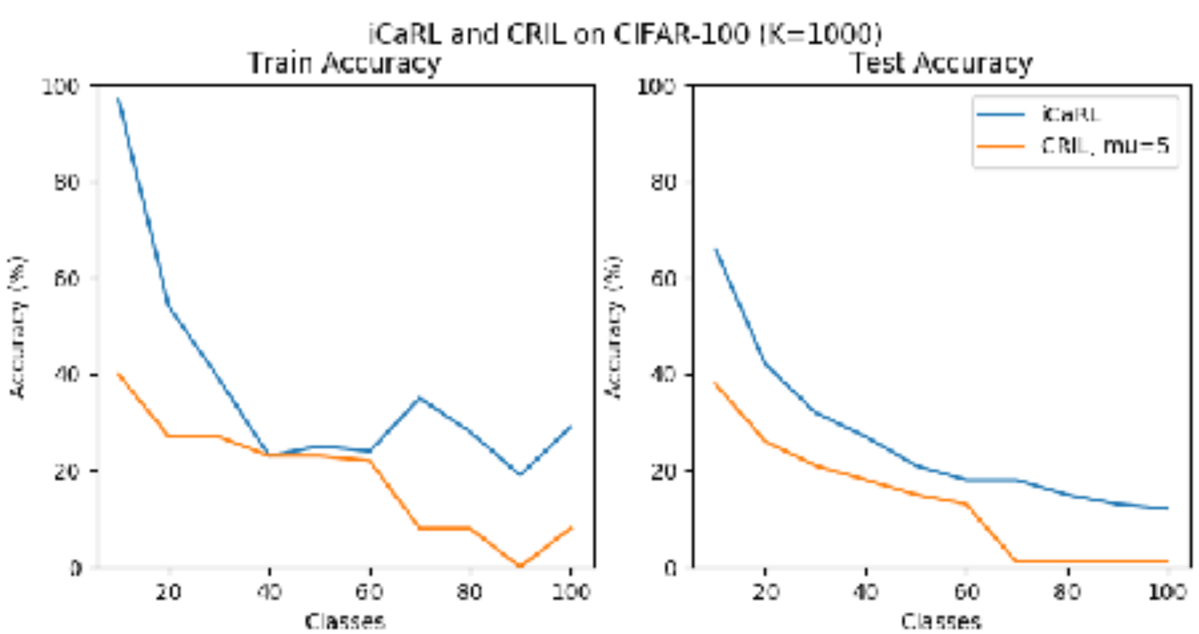
Results: MNIST



- CRIL outperformed **over 20%** in average accuracy to iCaRL for all sizes of exemplar.
- The performance tended to decrease for the smaller number of exemplars both in iCaRL and CRIL.

Results: CIFAR-100





- Both iCaRL and CRIL had poor performances, where the average accuracies were lower then 21%.
- iCaRL outperformed about 10% in average accuracy to CRIL for both K=500 and 1000.
- CRIL has pure MLP architecture, so the performance can be improved with CNN architecture.

Summary

- CRIL outperformed about 21% in average accuracy to iCaRL on MNIST dataset for all sizes of exemplars.
- CRIL had competitive performance with small size of exemplar, which was 23% higher than iCaRL.
  - CRIL: 75% test accuracy for 10 classes with K=100
  - iCaRL: 52% test accuracy for 10 classes with K=100
- iCaRL outperformed about 10% in average accuracy to CRIL on CIFAR dataset for all sizes of exemplars.
  - The performance of CRIL can be improved by adding the convolutional layers.

Conclusion

- CRIL showed competitive performance in class-incremental learning, which outperformed iCaRL on MNIST dataset.
- For CIFAR-100, CNN architecture can be added for CRIL for further comparison.
- The compactness of the output feature vectors will be analyzed with t-sne on the final report.