Revisiting Distillation & Incremental Classifier Learning

Khurram Javed[2] and Faisal Shafait[1,2]

[1]Deep Learning Laboratory, National Center of Artificial Intelligence, Islamabad

[2]School of Electrical Engineering and Computer Science, National University of Sciences and Technology

1. Abstract

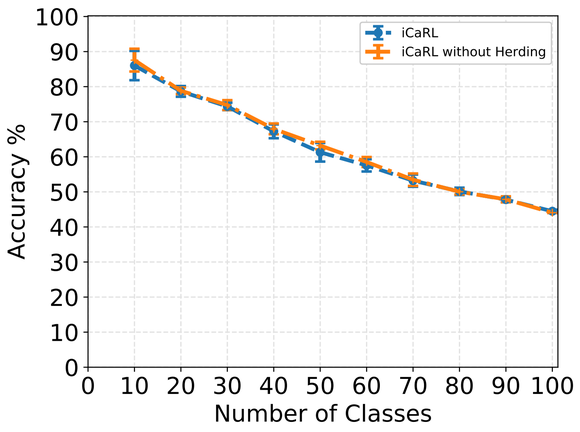
Artificial Neural Networks are unable to learn from data in an incremental way. This lack of ability, called Catastrophic Forgetting, is considered a major hurdle in building a true AI system. In this work, we isolate the truly effective existing ideas for incremental learning from those that only work under certain conditions. To this end, we first thoroughly analyze the current state of the art (iCaRL [1]) for incremental learning and make some non-trivial observations. We conclude that the success of iCaRL is primarily due to knowledge distillation and recognize a key limitation of knowledge distillation, i.e. it often leads to bias in classifiers. Finally, we propose a dynamic threshold moving algorithm that is able to successfully remove this bias. We demonstrate the effectiveness of our algorithm on CIFAR100 and MNIST datasets showing near-optimal results. Our implementation is available at : https://github.com/Khurramjaved96/incremental-learning.

2. Poster Overview

The current SOTA for incremental learning, iCaRL, proposed two algorithms, Herding and NEM, for incremental classifier learning. Former is used to select instances of old classes for rehearsal, whereas the latter is used for classification. In section 3 and 4, we show that both Herding and NEM are unnecessary for incremental learning, and by removing bias of the classifier during training or testing, we can get the same performance as iCaRL without using Herding or NEM. In section 5 we present our algorithm to remove bias at test time, and in section 6 we visualize the results of our algorithm.

3. iCaRL: Herding

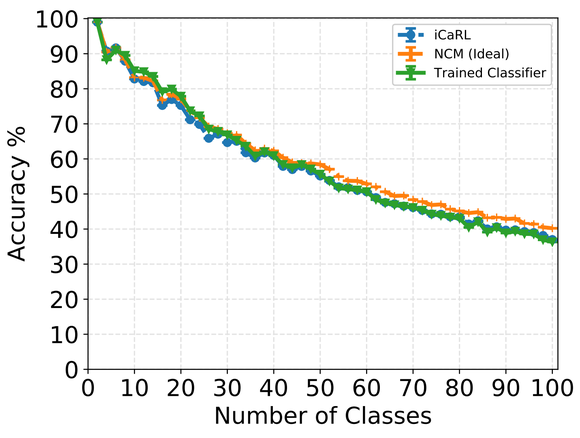
iCaRL proposed an algorithm called Herding for exemplar selection, but we discovered that herding does not work any better than random instance selection.



Experiment comparing iCaRL with herding and iCaRL with random instance selection; there is no meaningful difference in the performance.

4. iCaRL: Nearest Exemplar Mean

iCaRL proposed Nearest Exemplar Mean (NEM) algorithm as an improvement to trained classifier, however we showed that by removing bias for new classes during training or testing, the trained classifier is as good as NEM.

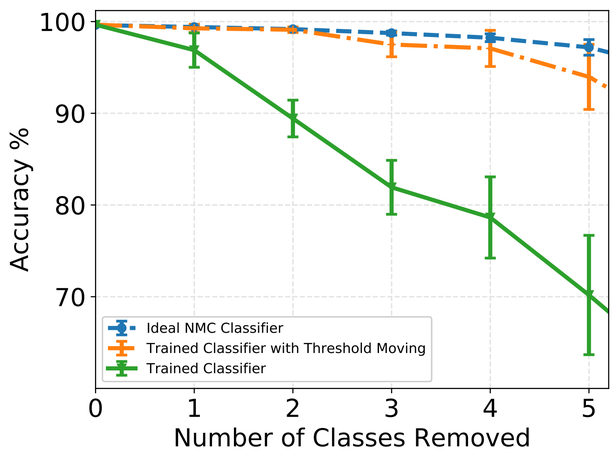


The bias can be removing by either using high temperature when distilling knowledge (shown in the graph above using temperature of 3), or by scaling the predictions of the classifier by a vector $\mathcal{S}$ at test time (explained in Section 5).

5. Dynamic Threshold Moving

Let $F(x_i)$ be the model that outputs the probability distribution over N classes i.e. $\forall x_i \in X$, $F(x_i) = P(n|x_i)$ where $0 <= n < N$. Suppose now that we want to find another model, $G(X)$, that gives a distribution over the old N classes and k new classes. Furthermore, we want to find G given only the data of the k new classes, and original model $F(X)$.
Let y_i be ground truth of new classes and $F^T(x_i))$ represents output of F with temperature T. Then Hinton et.al. [2] showed that we can train the model using the following loss function:

$$\sum_{x_i, y_i \in \mathcal{D}} (1-\gamma) \times C_{entropy}(G(x_i), y_i) + T^2 \gamma \times C_{entropy}(G^T(x_i), F^T(x_i))$$



However this results in a model that is biased in favor of new classes. We discovered that we can remove the bias by scaling the predictions of $G(X)$ using a vector $\mathcal{S}$ given by:

$$\mathcal{S} = \sum_{x_i, y_i \in \mathcal{D}} (1-\gamma) \times y_i + T^2 \gamma \times F^T(x_i) \quad (1)$$

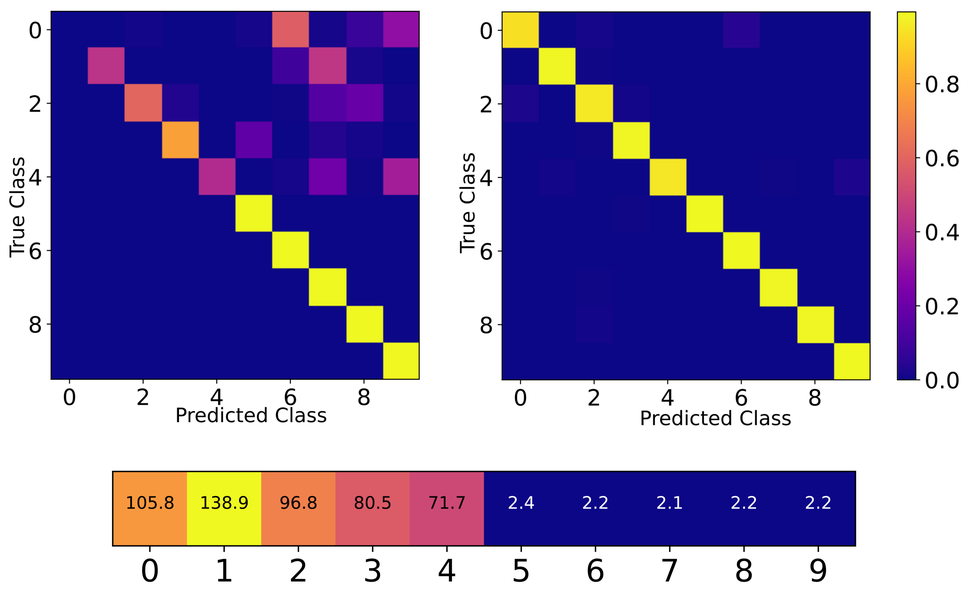
Where the scaled predictions $G'(X)$ are given by:

$$G'(X) = G(x) \circ \frac{\|\mathcal{S}\|}{\mathcal{S}} \quad (2)$$

It can be seen from that figure that by removing bias (Classifier with Threshold Moving), it is possible to improve on the current method (Trained Classifier) and achieve near ideal results (Ideal NCM Classifier).

6. Dynamic Threshold Moving : Results

In this section, we elaborate on the results of our algorithm presented in last section; we train our model on last 5 classes of MNIST and test it on all classes (Where knowledge of unseen 5 classes is distilled from a model trained on all classes). Without threshold moving (left confusion matrix), the model performs poorly on the old classes.



With threshold moving (right confusion matrix), however, not only is it able to classify unseen classes nearly perfectly, but also its performance does not deteriorate on new classes. The scaling vector $\frac{\|\mathcal{S}\|}{\mathcal{S}}$ learned while training is also visualized below the confusion matrices; Note that the predictions of unseen classes are scaled approximately 50 times compared to those of seen classes; this shows that while training, the signal model gets for old classes is extremely small, which is precisely the reason of the large bias.

7. Implementation Details

We open-source our **PyTorch** implementation of iCaRL, as well as dynamic threshold moving, at https://github.com/Khurramjaved96/incremental-learning. We hope that our implementation will promote reproducibility in science, and allow future researchers to validate our work.

8. References

[1] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. pages 2001–2010, 2017.

[2] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.