Network Adaptation Strategies for Learning New Classes Without Forgetting the Original Ones

Hagai Taitelbaum[1], *Gal Chechik*[2], *Jacob Goldberger*[1]

[1] Faculty of Engineering, Bar-Ilan University, Ramat-Gan, Israel

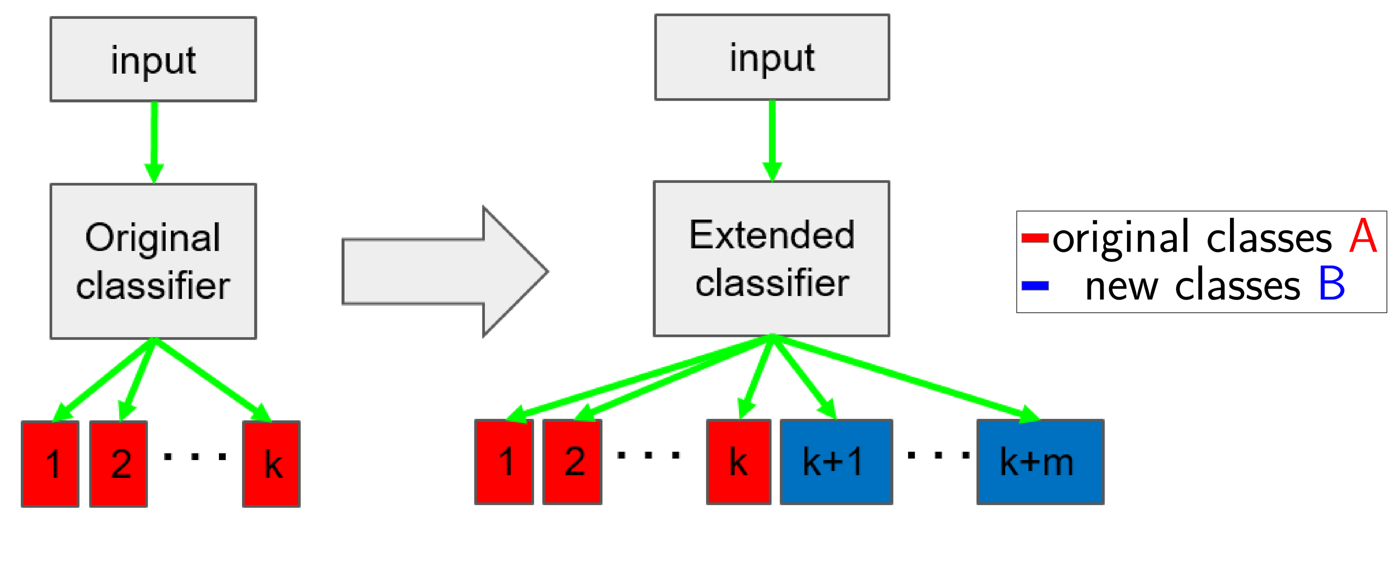
[2] The Gonda Brain Research Center, Bar-Ilan University and NVIDIA Research

Overview

- We propose a deep learning method for adding new classes to a given classifier without access to the original data.
- This problem arises frequently since models are often shared without their training data, due to privacy and data ownership concerns.
- We modify the original classifier by retraining a suitable subset of layers using a knowledge-distillation regularization
- The achieved accuracy is almost as good as that obtained by a system trained from both the original and new classes.

Problem formulation

- We are given a classifier C_A for k original classes A $= \{1,2,...,k\}$ and training data for m **new** classes B $= \{k+1,...,k+m\}$.
- We wish to build an extended classifier C_{AB}, that can handle samples from all classes $A \cup B$.
- We can access to the parameters of C_A but not its training data.



$p_{or}(y = i|\mathbf{x}), \quad i \in A$ $\qquad$ $p_{ex}(y = i|\mathbf{x}), \quad i \in A \cup B$

Challenges

- Catastrophic Forgetting
  - Forget previously learned information upon learning new one
- Privacy
  - No samples from original classes at training time
- In contrast to *Transfer Learning*, we interested in the extended class-set, rather than the new one

Our training approach

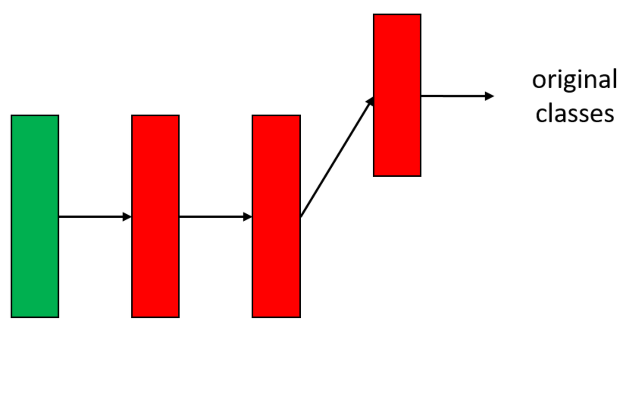
- Retrain a subset of the layers of C_A
  Motivated by Transfer Learning
- Use a regularized term:
  Motivated by Knowledge Distillation

$$L = (1-\epsilon)\sum_{t=1}^{n} \log p_{ex}(y_t|\mathbf{x}_t) + \epsilon \sum_{t=1}^{n}\sum_{i \in A} p_{or}(y=i|\mathbf{x}_t) \log p_{ex}(y=i|\mathbf{x}_t)$$

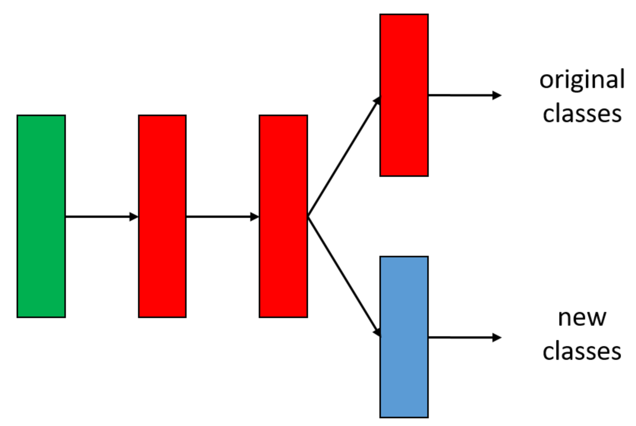
where ϵ weights the regularization term

Compared Methods

Baselines:

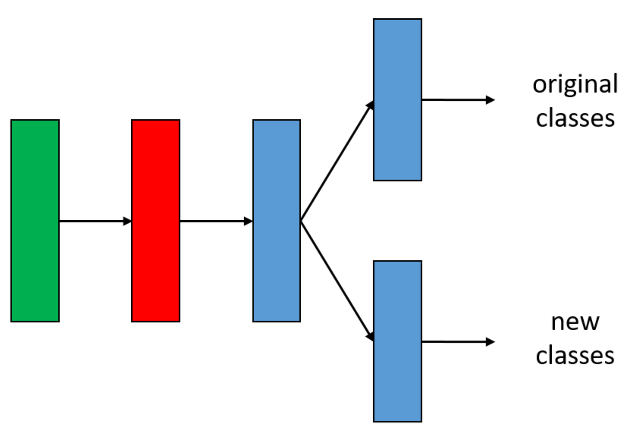


Original model

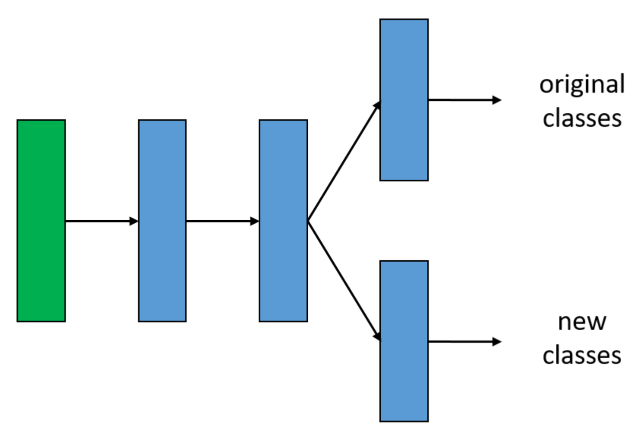


Adding new classes (ACWOD)

Our approach:

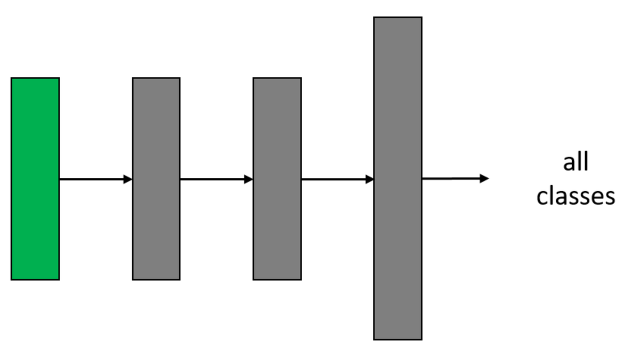


retraining part of the network

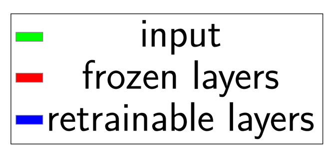


retraining all layers

Upper bound:

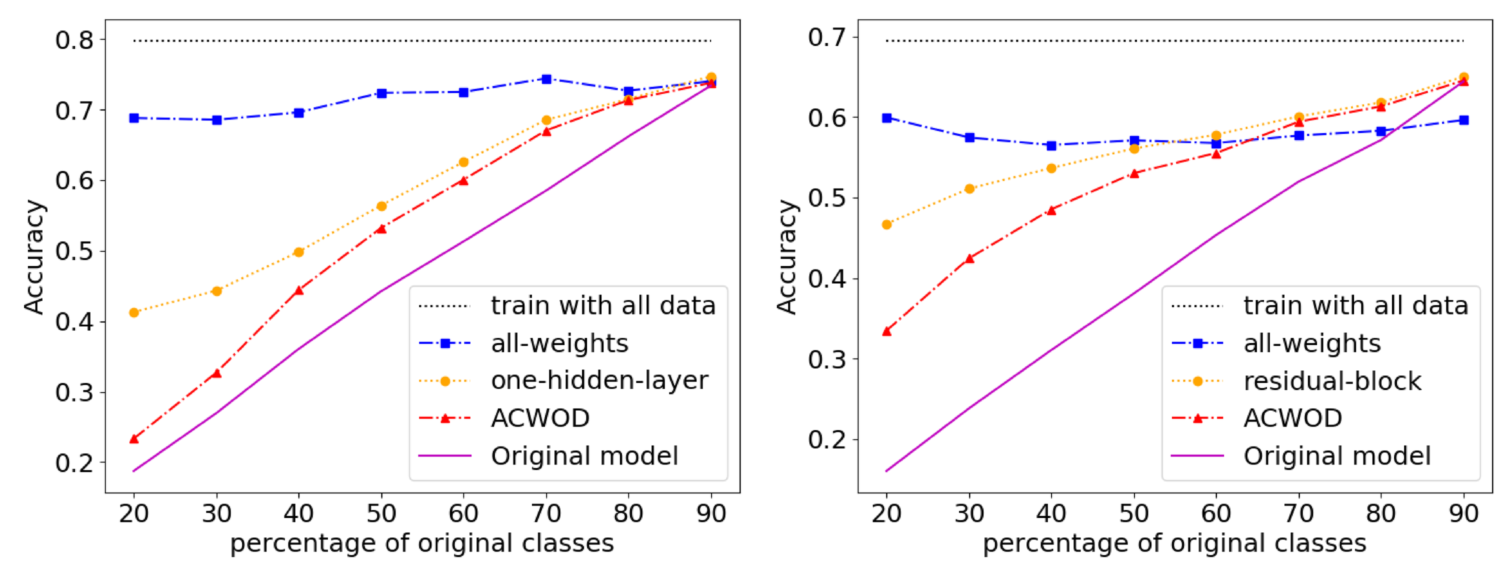


train with all data



Classification Results

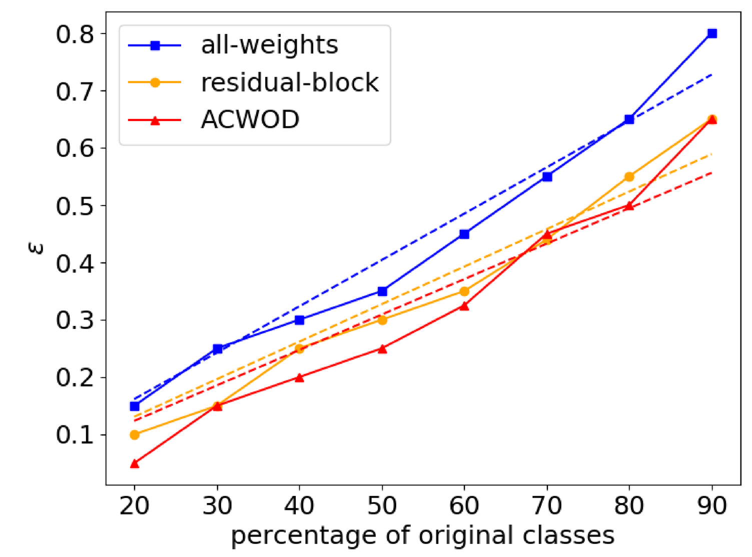
NIST-2015 (50 languages) $\qquad$ CIFAR-100 (100 image classes)



- NIST-2015
  - Accuracy improves as we retrain more layers
- CIFAR-100
  - When percentage of original classes is low performance improves as more layers are retrained
  - When percentage of original classes is high retraining all weights damages performance
- NIST-2015 vs. CIFAR-100
  - CIFAR-100 consists of raw images, while NIST-2015 consists of well-tuned and high-level features

Analyzing the weight of the regularization

CIFAR-100



- ϵ is linearly proportional to the number of original classes.
- As more layers are retrained, ϵ is bigger.
- Methods which constraint network layers (*ACWOD*, *residual-block*), allow the re-trainable layers to better adapt, using a smaller ϵ.