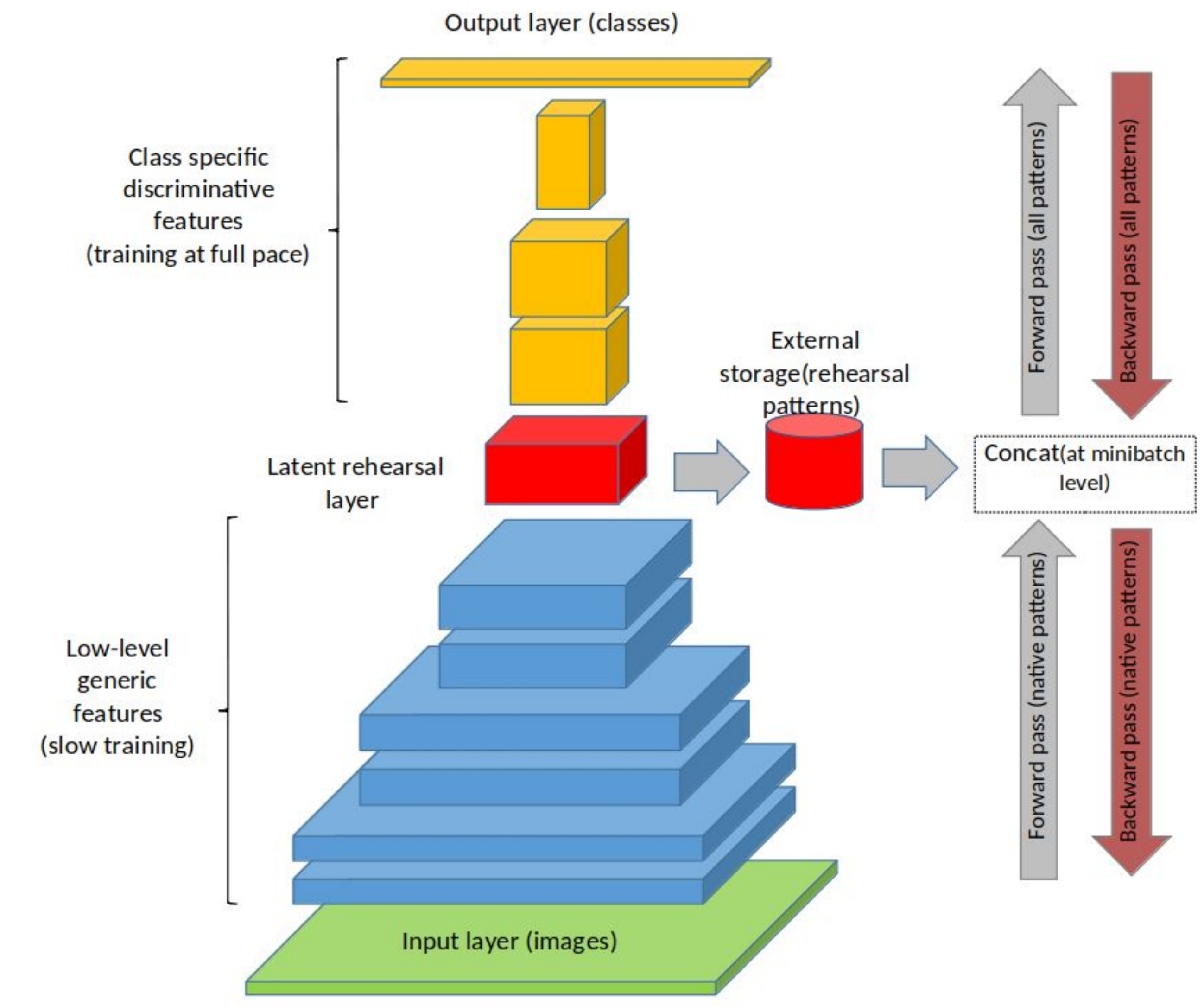
IROS - Lifelong Object Recognition Challenge Unibo Team

Gabriele Graffieti, Lorenzo Pellegrini, Vincenzo Lomonaco, Davide Maltoni
University of Bologna, via dell'Università 50, 47522 Cesena - Italy
email: {gabriele.graffieti, l.pellegrini, vincenzo.lomonaco, davide.maltoni}@unibo.it

Abstract

The ability to **learn continuously** is particularly desirable for many applications. In particular, **robotic vision** might highly benefit from a continual learning approach, since training the model from scratch as soon as new data becomes available is prohibitive in terms of storage and computation.
A new approach based on **latent rehearsal** is proposed, where a replay memory contains latent space patterns extracted from the trained model. This approach can highly reduce the memory and the computation needed during training, making possible to train the proposed model on mobile and low-computational-power devices. It is also possible to **tune** the algorithm in order to reach a higher accuracy at the cost of more resources and vice-versa.



Proposed Approach

Rehearsal

Many continual learning approaches are based on rehearsal, namely the replay of old patterns to the network, in order to contrast the problem of forgetting. However, storing and replaying old patterns requires extra amount of memory and extra steps of computation, because of the extra forward and backward steps. This makes the training too resource demanding for low computational power devices.

Latent Rehearsal

In order to address the problems of classic rehearsal, we propose a new approach based on a latent representation of patterns in the replay memory. The **memory contains feature vectors** extracted from some level of the network. This reduces the memory usage, since feature vectors are often smaller that the input images, and **highly reduces the amount of required computation,** since the backward pass has to be done only up to the rehearsal level. The algorithm can be summarized as follow:

1. Take n patterns from the current batch.
2. Forward them through the network until the rehearsal layer
3. Select k patterns from the rehearsal memory
4. Concat the original and the replay patterns
5. Forward all the patterns through the rest of the network
6. Backpropagate the loss only until the rehearsal layer

Rehearsal Memory Management

The feature are **not pre-extracted** before the beginning of the training, but the memory is updated after every batch.
The pseudocode of the memory update is the following:

Let $M = \varnothing$, M_{size} = number of patterns to be stored in M

for each training batch B_i
  train the model on shuffled $B_i \cup M$
  $h = M_{size} / i$
  R_{add} = random sampling h patterns from B_i
  if $i > 1$
    $R_{replace}$ = random sampling h patterns from M
  else
    $R_{replace} = \varnothing$
  $M = (M - R_{replace}) \cup R_{add}$

Continual Learning Strategy

We have used the Learning without Forgetting (LwF) strategy alongside latent rehearsal in order to improve the performance of our approach. The main advantages of LwF are:

- Lightweight, so it can be used even in low computational power devices
- Only one additional computational step for every new batch
- One of the best strategies for NI (new instances) CL tasks

Experiments

Dataset

We used the OpenLoris dataset during the training and testing of our model
- 69 classes of everyday objects
- Different levels of illumination, clutter, object size and occlusion
- Images at different sizes but same aspect ratio

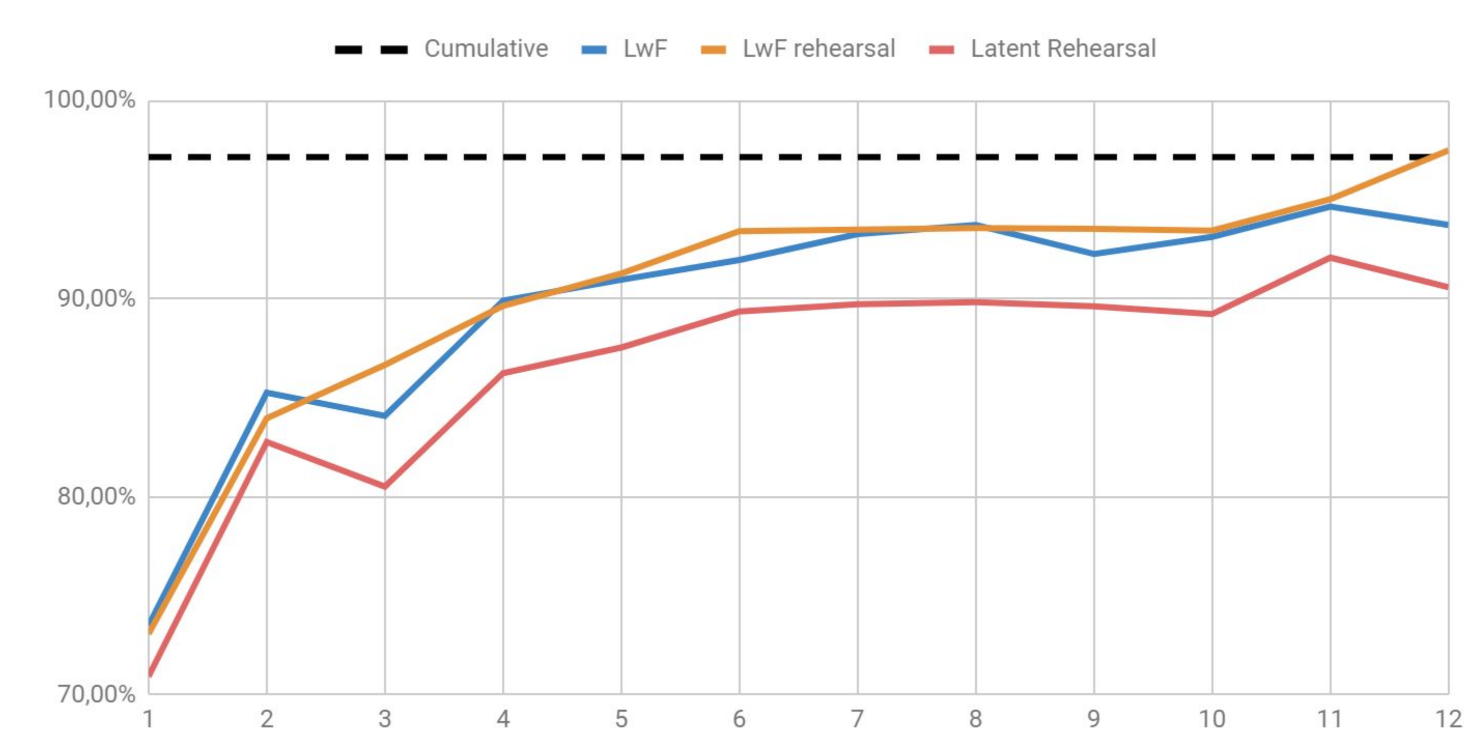
Training

- We used a MobileNet V2 0.75 for our experiments, in order to save memory and computation time.
- Images resized to 224x396, then cropped to 224x300 then a random central crop of size 224x224 is taken (data augmentation).
- Batch size of 32 (30 images from the current batch + 2 from the rehearsal memory)
- LwF α = 0.3, epochs per batch = 3, rehearsal memory size = 966 (69x14)
- Rehearsal layers: input and last inverted residual

Testing (on OpenLoris validation set)

- Test set not disclosed (competition)
- Images resized to 224x396, then the central 224x224 crop is taken

	Cumulative	LwF	Input rehea.	latent rehea.
Accuracy	97.14%	93.72%	97.48%	90.57%
Model size	9.1MB	5.7MB	5.7MB	5.7MB
Rehearsal Memory size	0	0	150MB	7.8MB
Inf. time	~1m47s	~1m35s	~1m32s	~1m33s



Conclusions

- Rehearsal is an effective strategy to address the problem of forgetting
- An approach based on latent rehearsal is agnostic regarding the CL strategy, and can be tuned in order to increase accuracy or performance
- Paired with a lightweight approach as LwF, the algorithm can be used to **train a model directly on mobile** or low computational power devices.
- This approach has made possible the development of the **first mobile app that performs training on device**