# Incremental Learning

## Machine Learning and Deep Learning

*Cordaro Nicolò*
*S272145*

*Di Nepi Marco*
*S277959*

*Falletta Alberto*
*S277971*

# Roadmap

- Finetuning baseline
- Learning Without Forgetting
- iCaRL
- Ablation studies on Losses and Classifiers
- Our own proposal

# Introduction

Incremental Learning is an open problem in the field of machine learning and image recognition.

**Scenario**: Dataset is not fixed and known apriori. New classes are progressively available over time.

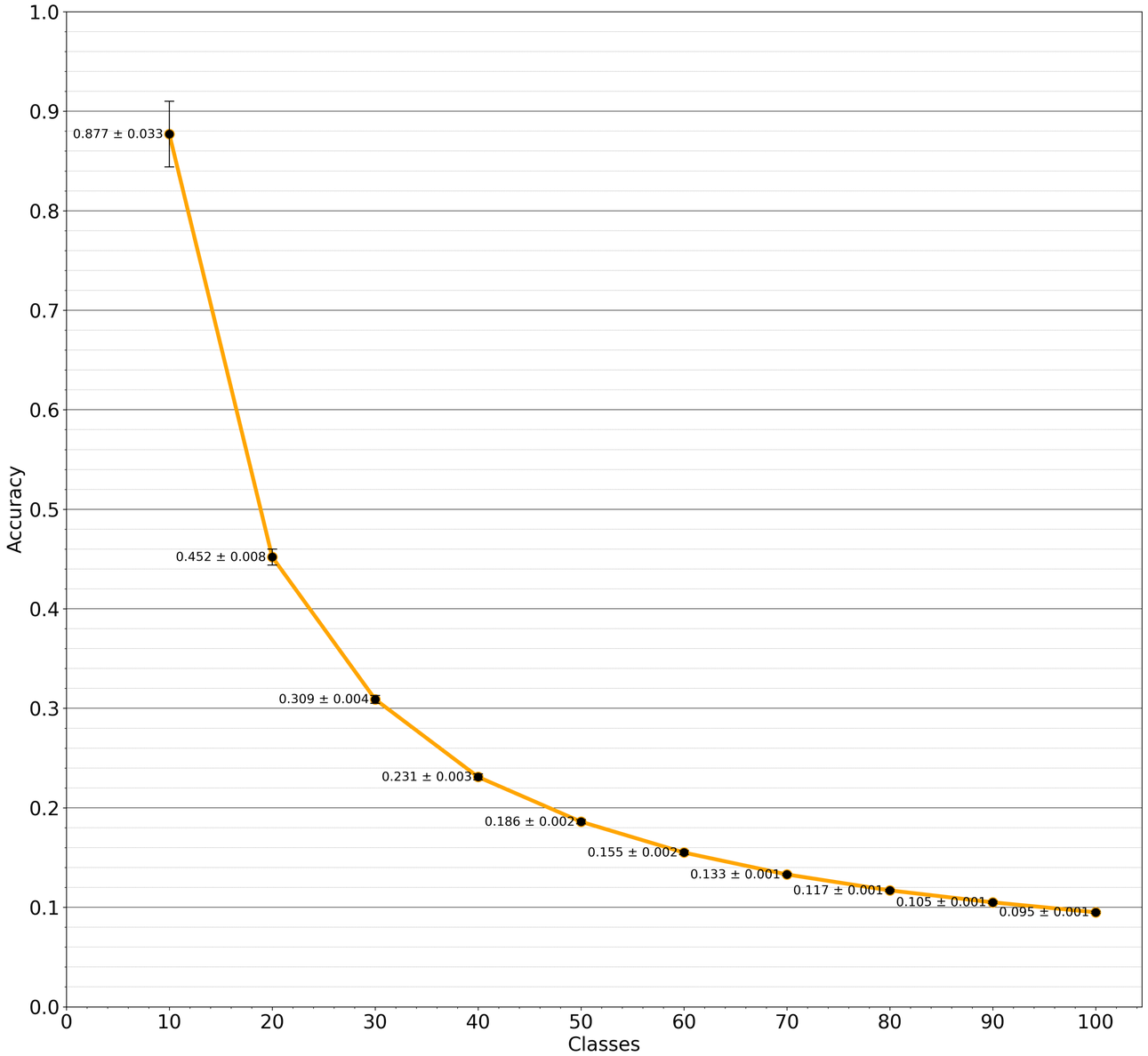
**Goal**: Find a robust model that is able to learn from a stream of data, without suffering of catastrophic forgetting.

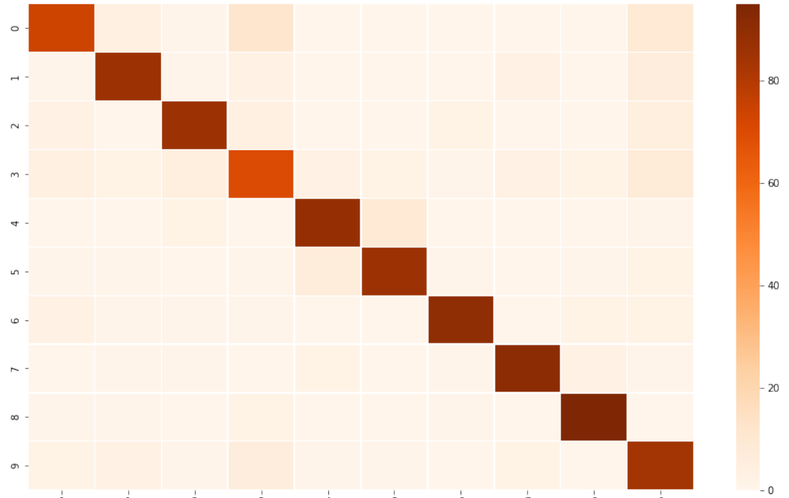
# Finetuning

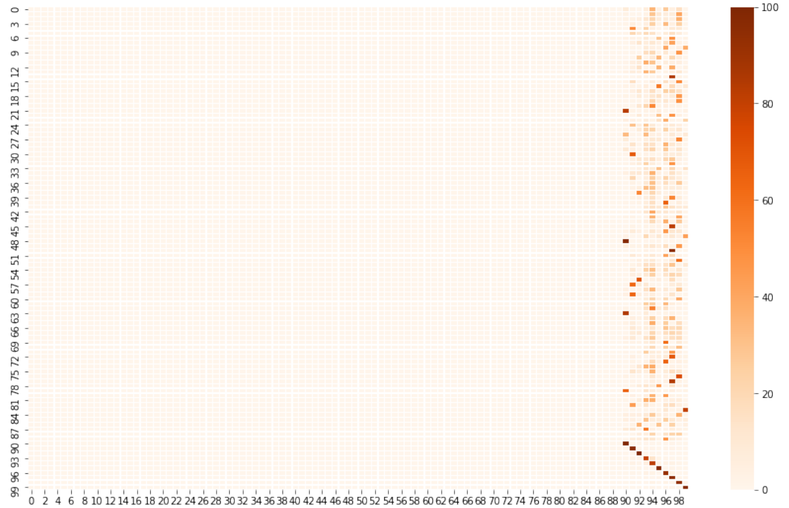
The model implemented in this section, doesn’t use any strategy to prevent **catastrophic forgetting**.

Every iteration 10 new classes are selected **randomly** and loaded into the training set. The new trained model is tested on the whole dataset used so far.

As can be expected, the model **completely forgets** the existence of the previously seen classes.


Accuracy
Classes
0.877 ± 0.033
0.452 ± 0.008
0.309 ± 0.004
0.231 ± 0.003
0.186 ± 0.002
0.155 ± 0.002
0.133 ± 0.001
0.117 ± 0.001
0.105 ± 0.001
0.095 ± 0.001



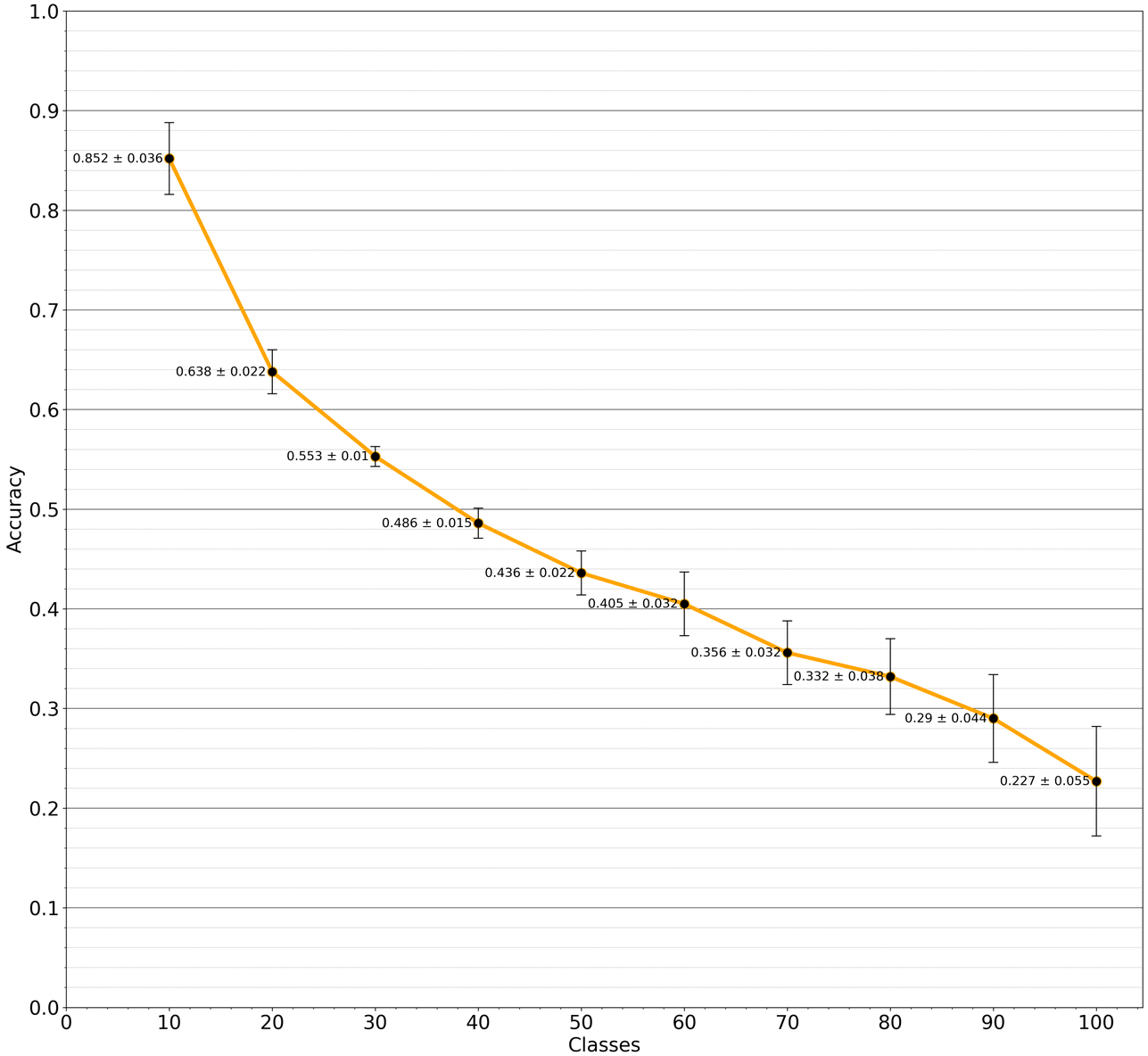


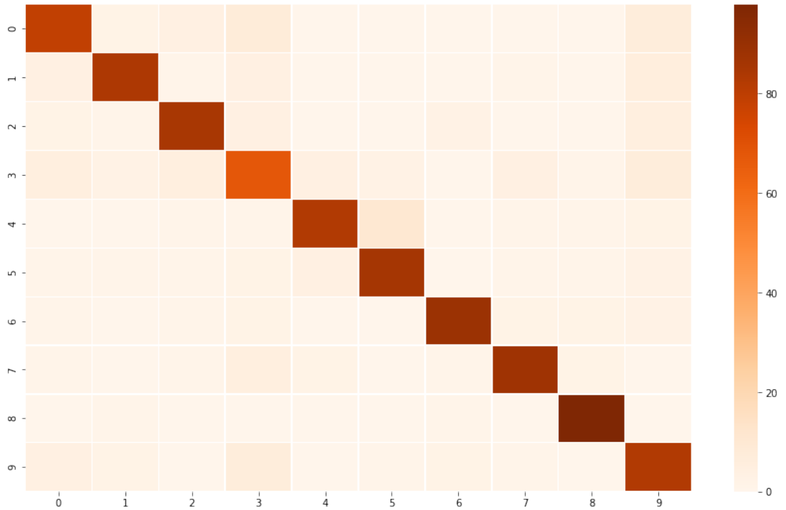
# Learning Without Forgetting

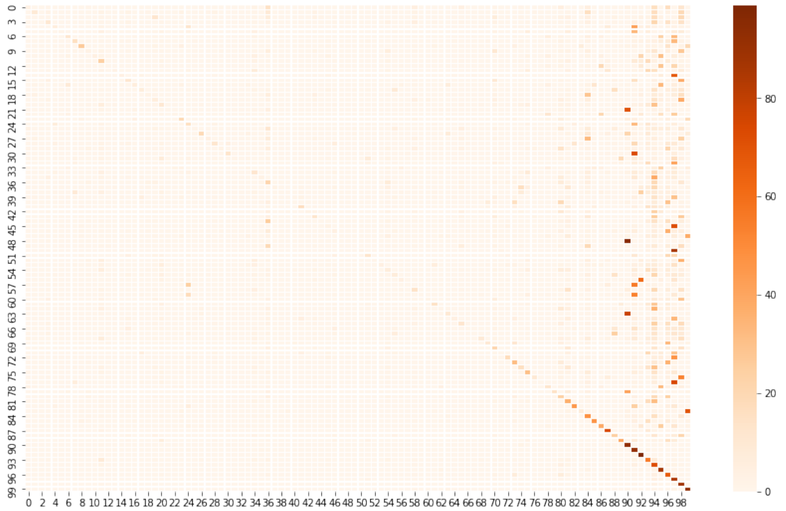
This model implementation adopts the principle of **distillation**: to tackle catastrophic forgetting the network is simulated with both new and old data.

Loss is composed by two terms: **classification loss** between the outputs of the network and the new labels, and **distillation loss** between the outputs of both, new and old network for the old classes.

This solution tries to address catastrophic forgetting, but still presents **bias** toward the new classes.


Accuracy
Classes
0.852 ± 0.036
0.638 ± 0.022
0.553 ± 0.01
0.486 ± 0.015
0.436 ± 0.022
0.405 ± 0.032
0.356 ± 0.032
0.332 ± 0.038
0.29 ± 0.044
0.227 ± 0.055





# iCaRL

iCaRL implementation, proposes a different strategy: the use of a set of **exemplars** and classification by a **nearest-mean-of-exemplars**.

Exemplars ensure to have a fixed number of images from previous classes, organised in a priority list. It is chosen the **best** possible **representative subset** according to the principle of **herding**.

**Nearest Mean of Exemplars** consists in computing the **prototype** vector for each class observed, assigning the label of the new image to the most similar vector.

# iCaRL functions

**Classify:**

Given a batch of images find the nearest prototype and return the labels

$$y* \leftarrow \underset{y=1..t}{argmin} \|\varphi(\mathbf{x}) - \mu_{\mathbf{y}}\|$$

**Construct Exemplars Set:**

For each new class build a list of representative items via herding

**for** k = 1,...,m **do**

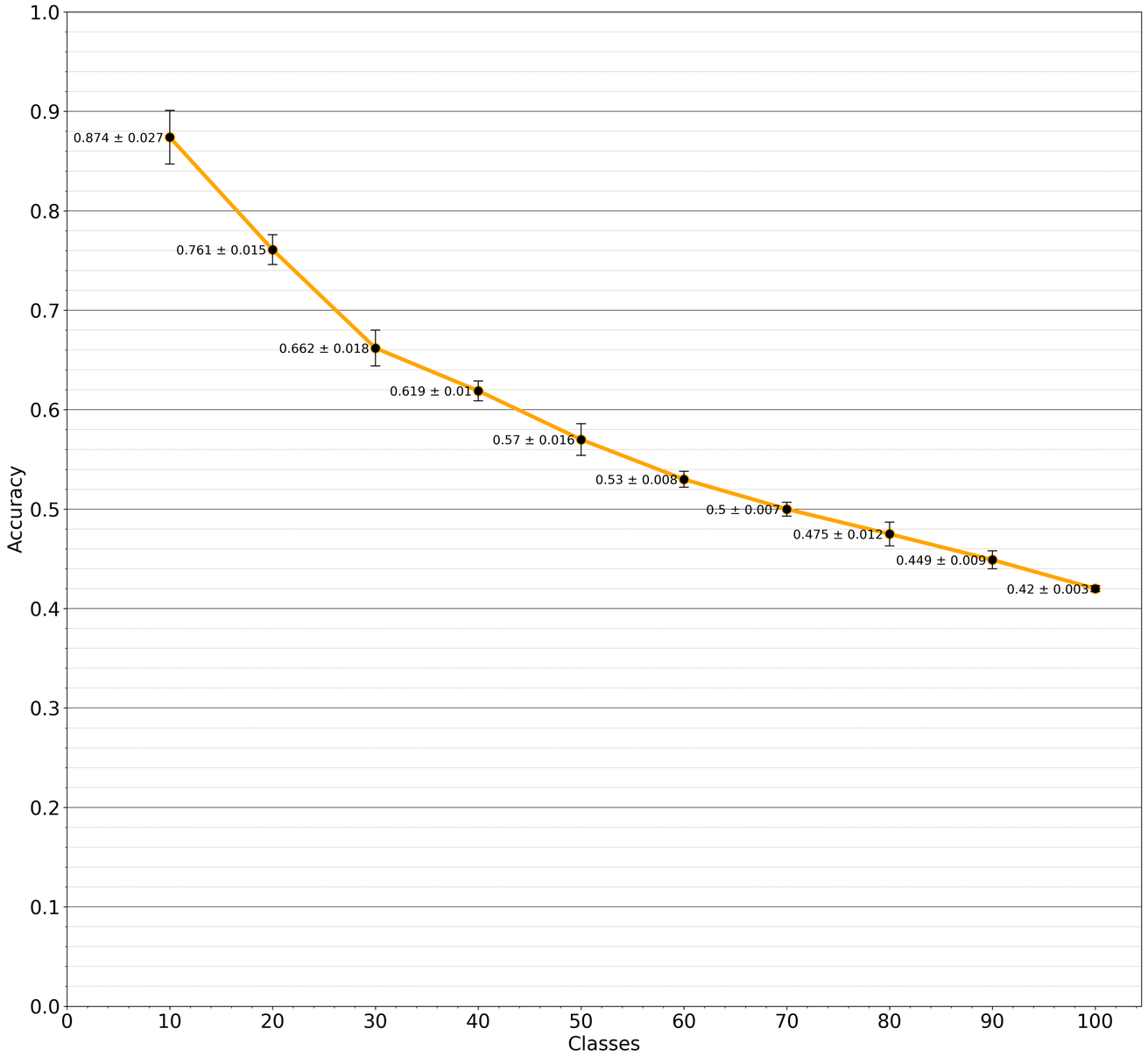
$$p_k \leftarrow \underset{x \in X}{argmin} \|\mu - \frac{\mathbf{1}}{\mathbf{k}}[\varphi(\mathbf{x}) + \sum_{\mathbf{j=1}}^{\mathbf{k}} \varphi(\mathbf{p_j})]\|$$

**end for**

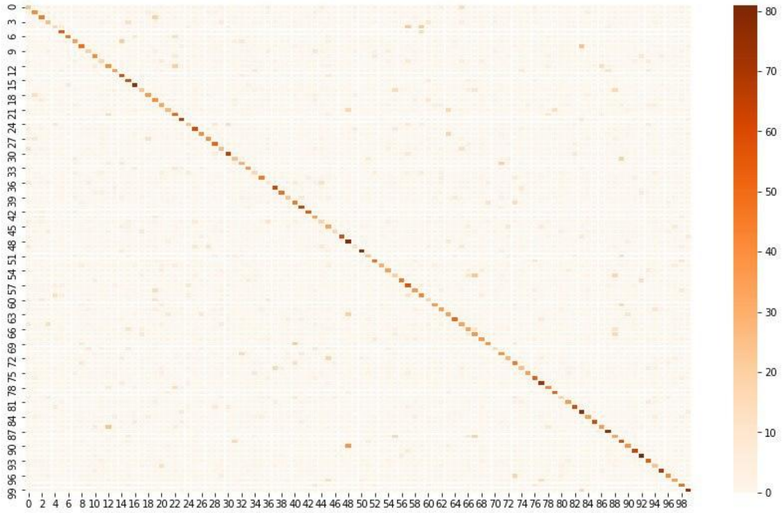
**Update Representation:**

Training on the union of new data and exemplars with distillation loss.

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




As can be seen by the graphs, there is a clear **improvement** of the accuracy, and the classification is more **homogenous**, and not so biased toward some classes.



# Ablation studies on losses

In this section some experiments with different **combinations** of classification and distillation losses are made.

The losses tried are four:

- Kullback-Leibler Distance
- L2 Loss
- L1 Loss
- Cosine embedding Loss

# Losses formulas

**Kullback-Leibler Distance:**
Measures the distance between two probability distributions

$$l_n = y_n \cdot (\log y_n - x_n)$$

**L2 Loss:**
Measures the L2 Loss computed between input and target

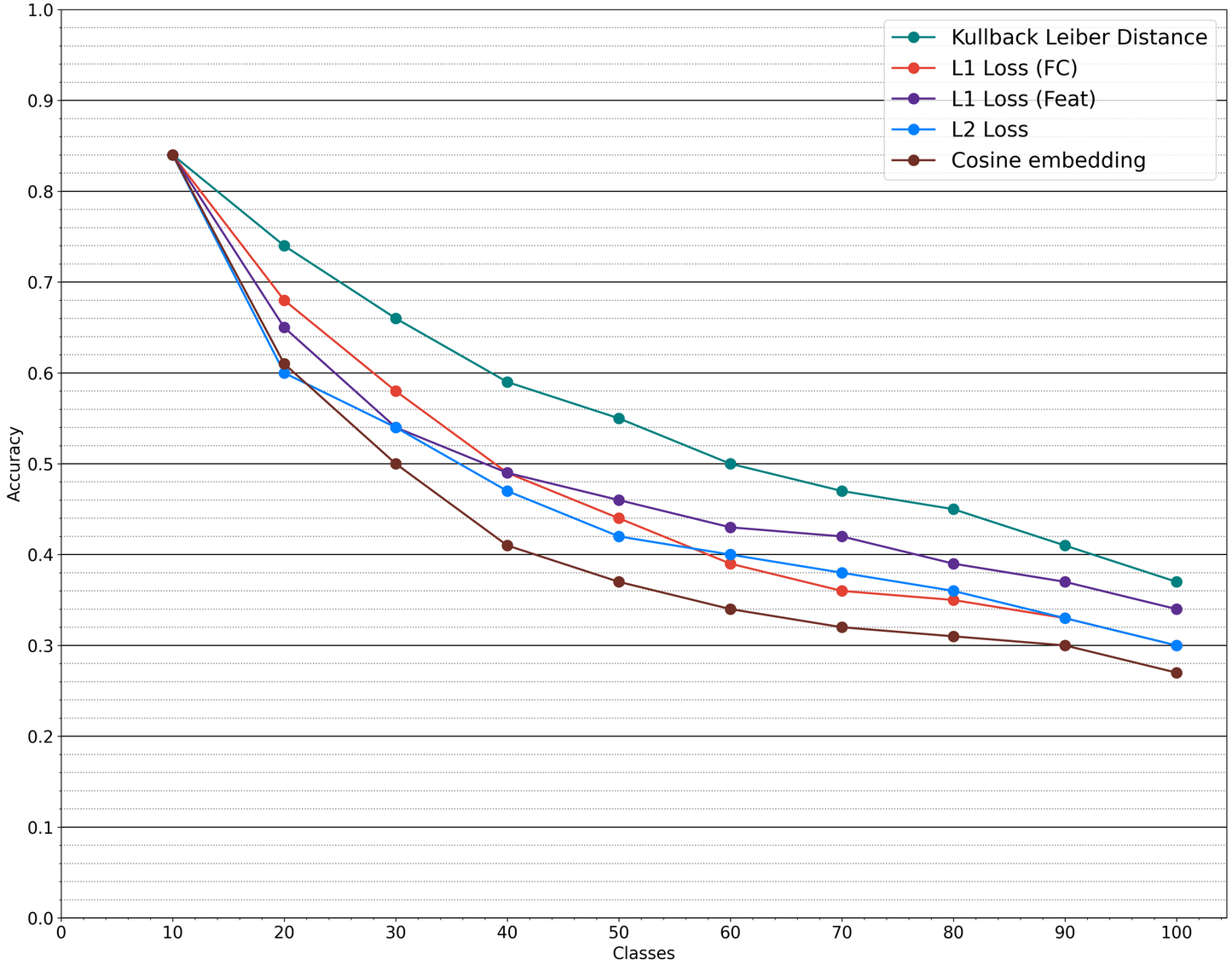
$$l_n = (y_n - x_n)^2$$

**L1 Loss:**
Measures the L1 Loss computed between input and target

$$l_n = |y_n - x_n|$$

**Cosine embedding Loss:**
Maximises the cosine similarity between two vectors

$$l(x_1, x_2) = 1 - cos(x_1, x_2)$$




| | KL | L1(fc) | L1(feat) | L2 | Cos |
|---|---|---|---|---|---|
| 20 | 0.74 | 0.68 | 0.65 | 0.60 | 0.61 |
| 30 | 0.66 | 0.58 | 0.54 | 0.54 | 0.50 |
| 40 | 0.59 | 0.49 | 0.49 | 0.47 | 0.41 |
| 50 | 0.55 | 0.44 | 0.46 | 0.42 | 0.37 |
| 60 | 0.50 | 0.39 | 0.43 | 0.40 | 0.34 |
| 70 | 0.47 | 0.36 | 0.42 | 0.38 | 0.32 |
| 80 | 0.45 | 0.35 | 0.39 | 0.36 | 0.31 |
| 90 | 0.41 | 0.33 | 0.37 | 0.33 | 0.30 |
| 100 | 0.37 | 0.30 | 0.34 | 0.30 | 0.27 |

# Ablation studies on classifiers

In this section some experiments with different **alternatives** to the nearest class mean classifier.

The classifiers tried are three:

- Bias Correction Layer
- Cosine Layer
- Naive Bayes Classifier

# Classifiers Formulas

**Bias correction layer:**

Tries to remove bias present in the last layer.

$$q_k = \begin{cases} o_k, & \text{if } 1 < k < n. \\ \alpha o_k + \beta, & \text{otherwise.} \end{cases}$$

**Cosine layer:**

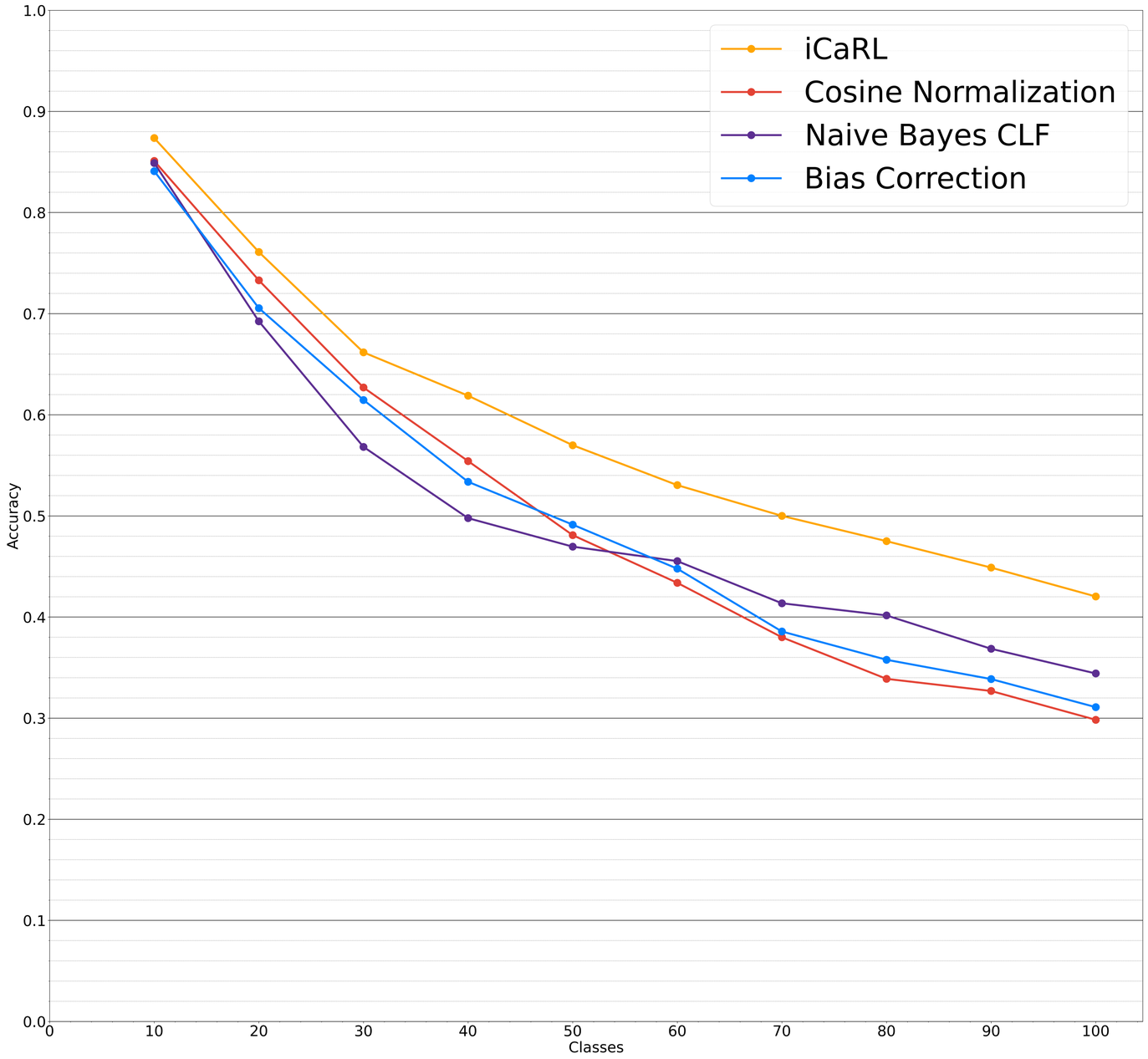
Tries to remove bias present in the last layer.

$$\theta_i^T f(x) + b_i \quad \text{->} \quad \eta < \bar{\theta}_i \, , \bar{f}(x) >$$

**Naive Bayes Classifier:**

Online classification algorithm that relying on Bayes theorem, constantly updates posterior probabilities

$$p(c|X) = \frac{p(X|c)\,p(c)}{p(X)} \qquad p(c|X) = p(x_1|c)\,p(x_2|c)\,...p(x_n|c)\,p(c)$$




**Cosine Normalization** and **Bias Correction**, that are in practise an improved version of a fully connected layer, perform similar and follow almost the same curve.

The incremental classifier proposed, **NB Classifier**, has a steeper curve in the first part of the process, but after the sixth batch is able to outperform the other two.

# Our own proposal

While working on iCaRL implementation we found one main **weak point**.
The dataset used for training the network is strongly **unbalanced**.

Our work is mainly inspired by advanced augmentations techniques:

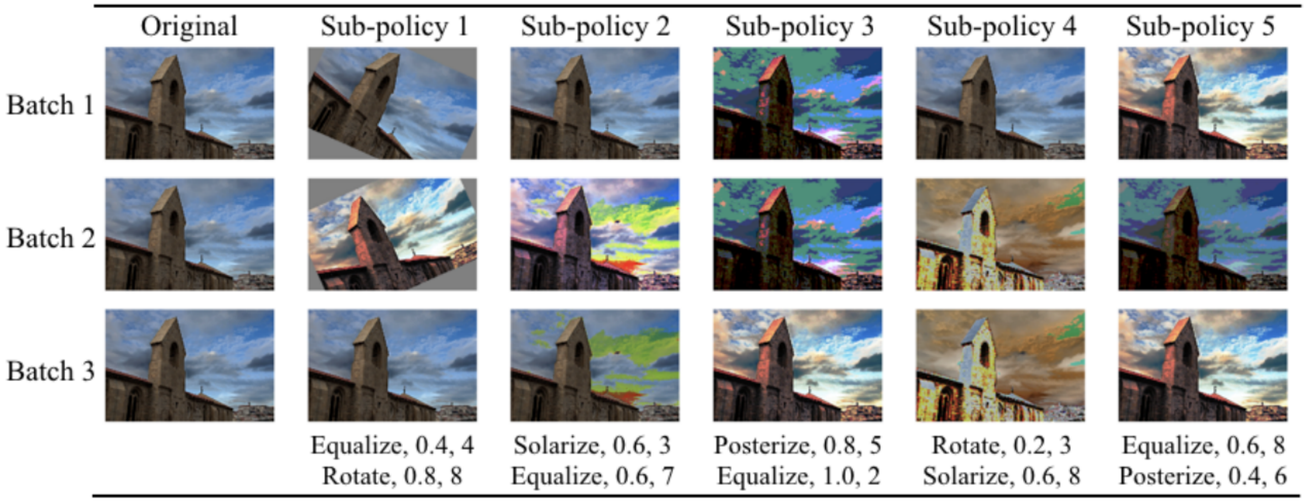
- AutoAugment
- Data Augmentation by Pairing Samples for Images Classification
- End-to-End Incremental Learning

# Techniques

**AutoAugment**: automatically search for the best augmentations policies. On our dataset, resulted of a policy made of 24 sub-policies.

**SamplePairing**: given a small dataset new images can be synthesised by taking the average of two random images at pixel level.

**Balanced fine-tuning**: in addition to new transformations, a balanced fine-tuning phase using a small subset of the new images is added.




Examples of results after **AutoAugment** implementation: new images are “created” according to the best augmentation policies.


Examples of results after **SamplePairing**: the newly synthesised image is labeled with the same label of *img_a*.

# Conclusions

Despite **our efforts**, our implementation does not provide any significant accuracy improvement, but only a slight decrease after the fine-tuning.

It's **not trivial** to write from scratch a new model that is able to **outperform** existing models. **Nearest Mean** classification and the use of **exemplars jumpstarted** the research in this field. Even provided this, **Incremental learning** in image recognition it's still an **open problem**.

**Thanks for your attention.**