Incremental Learning in Image Classification

Manuele Macchia
s277309@studenti.polito.it

Francesco Montagna
s277596@studenti.polito.it

Giacomo Zema
s269614@studenti.polito.it

Abstract

Extending the knowledge of a model is an open problem in deep learning. A central issue in incremental learning is catastrophic forgetting, resulting in degradation of previous knowledge when gradually learning new information.

The scope of the described implementation is to reproduce some existing baselines that address the difficulties posed by incremental learning, to propose variations to the existing frameworks in order to gain a deeper knowledge of their components and in-depth insights, and finally to define new approaches to overcome existing limitations.

1. Introduction

Incremental learning is a paradigm that allows extending the knowledge of an existing model, gradually incorporating new information, as opposed to training the model on a complete dataset. Training a system on a dataset that includes all classes is unfeasible in many real-world scenarios, *e.g.*, where data comes in a stream or where previously collected data is not available anymore. Therefore, the possibility of incrementally learning new data while maintaining existing knowledge has great importance for training more flexible and dynamic systems.

A central issue in incremental learning is *catastrophic forgetting* or *catastrophic interference* [5], *i.e.*, training a model with new data interferes with previously acquired knowledge. This leads to a performance decrease or, in the worst case, to the old knowledge being overwritten.

Considerable research has been devoted to limiting the effects of catastrophic forgetting. We mainly consider two incremental learning baselines in computer vision, Learning without Forgetting [4] and iCaRL [6]. These methods mitigate catastrophic forgetting by adopting techniques such as knowledge distillation and prototype rehearsal, which we further explore in our study.

In this work, we implement the existing incremental learning baselines of Learning without Forgetting and iCaRL to gain a deeper knowledge of the problem and the proposed techniques. We explore the effect of each component of the baselines on the performance of the model, and study the effectiveness of alternative components, such as different loss functions and classifiers. Finally, we propose novel approaches to overcome existing limitations.

2. Method

In this section we describe the implementation details of our work, the underlying convolutional neural network and the dataset partitioning for incremental learning.

2.1. Network

We carry out all experiments using a 32-layers ResNet [2], a deep convolutional neural network. Residual networks address the vanishing gradient problem by allowing shortcut connections between layers, allowing for deeper architectures. From now on, we treat the network up to the second to last layer as a feature extractor $\varphi : \chi \to \mathbb{R}^d$ and the last fully connected layer as a classifier. This choice is in line with the implementation of iCaRL.

In our incremental setting, we initialize the last fully connected layer to ten output nodes, and gradually increment this number as the network learns new classes. This approach is coherent with the assumption that the total number of classes is not known a priori. We use the Xavier normal initializer [1] to initialize the network weights. In our experiments, we find that this approach leads to better accuracy.

2.2. Data

We execute our experiments on the CIFAR-100 dataset [3]. It is a popular dataset for incremental learning, and both our baselines [4, 6] use it to test their strategies. It comprises 60000 tiny images belonging to 100 classes.

CIFAR-100 provides 500 training images and 100 testing images per class. We hold out 50 images per class from the training set to form a validation set. We use this data to validate our models in some of our studies and to monitor the performance of the network during training.

We artificially increase the cardinality of the dataset through data augmentation. This technique may help in improving the results and avoiding overfitting. We apply the same data preprocessing that the iCaRL authors use in the

official code[1], which consists in applying a random horizontal flip with 50% probability and a 32 by 32 random crop with 4 pixels of padding to the training set. These transformations are actually effective, increasing the accuracy of the model on the first incremental step by 10%.

All three RGB channels are normalized with mean 0.5 and standard deviation 0.5. Using the true mean and standard deviation of the dataset would violate the incremental learning assumptions.

2.3. Incremental approach

Our incremental approach is consistent with the benchmark protocol of iCaRL. The dataset is split into ten batches of ten classes each, with classes randomly assigned to a batch. At each learning step, the network is trained on a batch of data and evaluated on a test set that contains all known classes up to that point. To obtain statistically significant results, we fix three class orders and run all experiments three times, as different class splits lead to very different accuracy scores. We present the results in terms of mean and standard deviation of classification accuracies after each batch of classes, using an error bar plot or reporting the average of the mean accuracies, called *average incremental accuracy*.

This approach does not eliminate all sources of randomness, which are still present in the initialization of the network weights. We do not explore this aspect due to time and computational resources limitations.

2.4. Settings

We train the model using stochastic gradient descent with momentum as optimization algorithm, with mini-batch size set to 64 and weight decay equal to 0.00001. The initial learning rate is set to 2.0 and is divided by 5 after 49 and 63 epochs. For prototype rehearsal, we store a maximum of K=2000 exemplars. These are the same hyper-parameters used by the authors of iCaRL. The only exception is the size of the mini-batches, which we lower to 64 to obtain results comparable to the ones reported in their study.

This is the default hyper-parameter set-up. All further variations that we apply to these settings are reported in the description of the experiments.

2.5. Code

For our experiments we rely on the *PyTorch* framework. Our source code is publicly available at `https://github.com/manuelemacchia/incremental-learning-image-classification`.

[1] `https://github.com/srebuffi/iCaRL`

3. Experiments

3.1. Joint Training

3.2. Finetuning

3.3. Learning without Forgetting

3.4. iCaRL

References

[1] X. Glorot and Y. Bengio. Understanding the difculty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE.

[3] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. page 60, 2009.

[4] Z. Li and D. Hoiem. Learning without Forgetting. *European Conference on Computer Vision (ECCV)*, 2016.

[5] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, May 2019.

[6] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. iCaRL: Incremental Classifier and Representation Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, Honolulu, HI, July 2017. IEEE.