Application of Lifelong Learning with CNNs to Visual Robotic Classification Tasks

Abel S. Zacarias
http://www.di.ubi.pt
Luís A. Alexandre
http://http://www.di.ubi.pt/~lfbaa/

Departamento de Informática
Universidade da Beira Interior
6201-001 Covilhã, Portugal
Instituto de Telecomunicações

Abstract

The field of robotics is becoming continuously more important, due to the impact it can bring to our everyday life. A long standing problem with neural network learning is the catastrophic forgetting when one tries to use the same network to learn more than one task. In this paper we present results of the application of a method to avoid catastrophic forgetting while using Convolutional Neural Networks (CNNs) to some visual recognition tasks relevant to the field of robotics. The results show that with this method a robot can learn new tasks without forgetting the previous learned tasks. Results also showed that if we applied this method, the performance on isolated tasks increases and it is better to use it than train a CNN in an isolated way (single task). We use for our experiments two well known data sets, namely, Olivetti Faces and Fashion-MNIST.

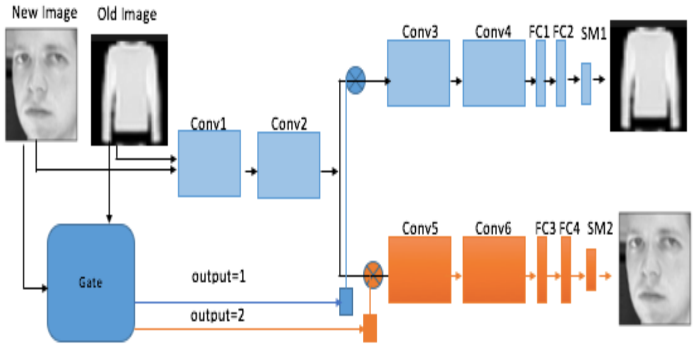

Figure 1: Network architecture: one branch is used for each task and the gate selects the branch to deploy at test time.

1 Introduction

Object recognition is an area of computer vision that deals with ability of an intelligent system to recognize instances of objects belonging to a certain category. This task has been important to the scientific community because of its many applications. One of these applications is in robotics, where a robot learns to distinguish objects (eg. boats, cars, clothes, plates, etc.) such that it can interact with the world. Another important computer vision area is biometrics, where a system processes data to recognize persons, such as using face images. Both these tasks are important if one wants to have robots interacting with people. Currently, deep learning approaches, such as the use of Convolutional Neural Networks (CNNs), has shown to be very effective in these types of visual recognition tasks. One goal is to have a single system learning to solve several tasks by reusing information from previously learned tasks to improve new ones, without forgetting what was learned before. This forgetting, in the neural network context, is a problem called catastrophic forgetting [2]. Catastrophic forgetting means that the performance of an agent trained to recognize a given task decreases when new tasks are added in an incremental manner.

There are some approaches to mitigate or overcome this the problem of catastrophic forgetting. For example [1] proposed solving the catastrophic forgetting problem in incremental learning scenario with support data inspired by the two major neurophysiology theories (Hebbian Learning System and complementary learning system). Also, in [6] an approach based on reinforcement learning is proposed where an agent interacts with an unknown environment to solve a specific task according to a policy and reward signal. It consists of three networks: controller, value network and task network. After learning the first task, the controller network decides how many filters or nodes should be added to each layer corresponding the new tasks, and only train the network on new added filters or nodes. The task network correspond to the expanded child network obtained from the controller network. During training, the parameters from the first task were frozen and only back-propagated the new added nodes or layers. Here we apply the SENA-CNN approach [7] to avoid catastrophic forgetting problem where the agent can incrementally learn new tasks without forgetting what was previous learned, while working with CNNs, and apply it to learning two problems relevant in the context of robotics: object and face recognition, using a single network.

2 Proposed Method

In this paper we aim to apply the method for lifelong learning proposed in [7] to the field of robotics. Our idea is to start by teaching a robot how to recognize faces and progressively teach it more capabilities in such a way that as it learns how to solve new tasks it does not forget the ones it previously learned.

First, the system must learn how to do face recognition, as it is a fundamental task in human-robot interaction; second, we will teach it to do object recognition, as it is also a key capability for a robot to be able to interact with its environment. To do so, we begin by training a CNN in one task. After the model reaches convergence, we make it learn the second task. For each new task, we add a new branch to the first model trained in an isolated way. For the second task, we only train our model on this added branch and keep the parameters of the previous layer frozen.

We do not add the first two layers of the model corresponding to the second task, instead, we use the two layers of the first learned task. That's because the neurons in those layers find several simple structures, such as oriented edges as demonstrated in [3] and these are applicable to many different tasks. The remaining layers seem to be devoted to more complex representations, and hence, are more specific to each problem, and that is why we choose to create those new layers instead of re-using the original ones.

2.1 Using the Gate to Select the Correct Branch

Our goal is to train a gate network with a super-class of both tasks. Each task is labelled $\{0,1,\ldots,n-1\}$, where n represents the total number of tasks to be deployed. This way, there is no need for all branches to process the input and produce an output and hence, only one branch is chosen by the gate to make the final decision. Figure 1 shows the procedure used by the gate mechanism for the case we are solving in this paper ($n=2$).

2.2 Training Methodology

The network is going to learn how to solve several tasks. First the gate is trained by using images from each task, with labels that represent those tasks (and not the original task classes). Second, we train a first network branch, starting with randomly initialized weights. Then, for each new task, a new branch is added to the network, where the first two convolutional layers are reused from the first branch.

3 Experiments

In this section we present the results of our method applied to the robotics field. We conducted our experiment using two data sets namely, Fashion-MNIST [5] which consists of 28×28 grey-scale images with 60000 training set and 10000 test set. We also used for our experiments the Olivetti data set [4] which comprises 400 pictures of people, 10 per person. All images are in grey-scale with size 64×64, all frontal and with a slight tilt of the head.

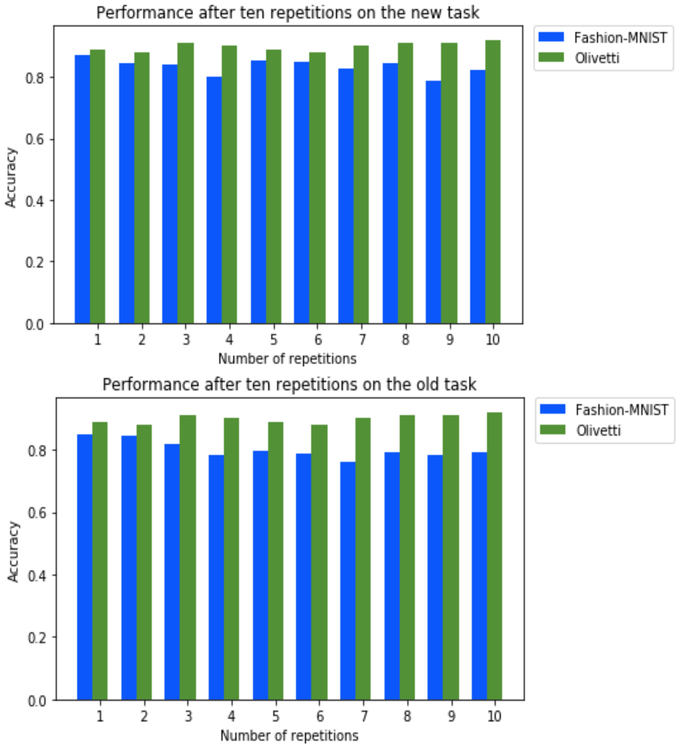

Figure 2: Network performance when we add a new task to an existing network trained on isolated learning.These results correspond to the ten repetitions of train and test of the proposed model. After testing the new task we test again on the old task: (top) performance on new added task; (bottom) performance on old task after training the network with the new task.

3.1 Network Architecture

To run our experiments we used a convolutional neural network with four convolutional layers each with a ReLU activation layer, maxpooling layer, dropout, flatten layer and two dense layers. The last dense layer is connected with a softmax layer. The gate network has the same architecture as the CNNs used in each branch.

As we previously said, the gate network is used to choose which branch to deploy at test time. Table 1 presents the mean accuracy and (standard deviation) when training each task in isolated learning, after ten repetition. These two networks trained on isolated are then used to add the branch corresponding to each new task.

Train	Test	Baseline
Gate	Gate	99.29 (2.17)
Olivetti	Olivetti	94.00 (3.25)
Fashion-MNIST	Fashion-MNIST	77.55 (11.35)

Table 1: Performance results on isolated learning of the gate, Olivetti and fashion-MNIST.

3.2 Adding New Tasks to the Model

As previously said, after training the robot to perform the first task, is necessary to teach the robot to perform new tasks as they come sequentially with the ability of not forgetting what was learned before. A robot with this ability will be a good improvement in the field of artificial intelligence, because the robot can learn many tasks without forgetting over its life time.

Table 2, presents the mean accuracy (and standard deviation) of our method when learning the new task. As we can see in the table, results show that if we see the two layers of the model trained on isolated learning to learn a new task, it can increase the performance of the new task compared to using a model trained isolated on the correspondent task. In this case, the performance of a robot will increase if we use this approach to learn new tasks.

After learning the new task it is necessary to test if the model did not forget what it learned previously. Table 3 presents the performance of our model after learning the new task. The Table shows that our model was able to preserve the performance on the old learned task and so it was possible for the robot to learn a new task without forgetting what it learned previously.

Old	New	Accuracy
Fashion-MNIST	Olivetti	94.41 (2.02)
Olivetti	Fashion-MNIST	83.23 (2.39)

Table 2: Network performance on new task, after learning a first task.

New	Old	Accuracy
Fashion-MNIST	Olivetti	91.43 (2.06)
Olivetti	Fashion-MNIST	79.50 (2.71)

Table 3: Network performance on old task, after learning a new task.

Verifying the experiments results, it is possible to observe the ability of the proposed approach to deal with the catastrophic forgetting problem. Comparing the results of a model trained on isolated learning with the proposed method, there is a slight degradation of performance on old task for Olivetti data set that is less than 3%. Another interesting observation is that the proposed method showed in the two scenarios (Fashion-MNIST $\mapsto$ Olivetti and Olivetti $\mapsto$ Fashion-MNIST) increased performance when compared to isolated training. This is understandable since by reusing partial information from previous tasks, we are somehow doing fine-tuning on the new task.

4 Conclusion

In this paper we present the results of the application of lifelong learning with CNNs to object and face recognition tasks. The results showed that we can use the SENA-CNN model to perform lifelong learning in this context and a robot using this approach is more efficient than if it is trained to learn each task with independent CNNs. Next we will focus on using theses results in a real scenario with a robot.

5 Acknowledgment

This work was supported by National Founding from the FCT- Fundação para a Ciência e a Tecnologia, through the UID/EEA/50008/2013 Project. The GTX Titan X used in this research was donated by the NVIDIA Corporation.

References

[1] Y. Li, Z. Li, L. Ding, Y. Pan, C. Huang, Y. Hu, W. Chen, and X. Gao. SupportNet: solving catastrophic forgetting in class incremental learning with support data. *ArXiv e-prints*, June 2018.

[2] Zhizhong Li and Derek Hoiem. Learning without forgetting. *CoRR*, abs/1606.09282, 2016. URL `http://arxiv.org/abs/1606.09282`.

[3] Ivet Rafegas, María Vanrell, and Luís A. Alexandre. Understanding trained cnns by indexing neuron selectivity. *CoRR*, abs/1702.00382, 2017. URL `http://arxiv.org/abs/1702.00382`.

[4] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pages 138–142, Dec 1994. doi: 10.1109/ACV.1994.341300.

[5] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL `http://arxiv.org/abs/1708.07747`.

[6] Ju Xu and Zhanxing Zhu. Reinforced continual learning. *CoRR*, abs/1805.12369, 2018. URL `http://arxiv.org/abs/1805.12369`.

[7] A. Zacarias and L.A. Alexandre. Improving sena-cnn by automating task recognition. In *19th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, LNCS, Spain, Madrid, 21-23 November 2018. Springer.