PhD Research Proposal

Unsupervised Tuning of Neural Networks for Object Classification in Sequential Data with Dynamically Chosen Object Classes

Adam Kosiorek

1 Introduction

Suppose that an autonomous agent moves through a large highly differentiated space. It can enter private and public buildings, parks, factories, crowded or desolate places. In order to interact with its environment, it has to recognize surrounding objects. High accuracy classification of variable environments with hundreds of object classes requires computationally expensive approaches, huge labeled datasets and is not good enough for many practical applications yet [1].

One approach of solving this problem relies on the fact that filters used in Convolutional Neural Networks (CNNs) respond strongly to visually similar inputs. It suggests using several simpler CNN-based classifiers specialized for subsets of possible object classes. Some of these classifiers, however, could have huge overlaps in types of objects they are trained to classify. Therefore, I suggest another approach. I believe that this problem can be solved by a (possibly recurrent) CNN classifier, whose filters are tuned online in an unsupervised way and whose final fully-connected layers are constantly calibrated on a small labeled training set. Additionally, if the set of classes this classifier predicts could be altered at run-time, it might be possible to achieve better accuracy at a lower computational cost. Finally, it results in a classifier that is constantly adapting to its changing environment.

2 Project Outline

There are three essential phases in this project: (1) Developing a recurrent CNN (RCNN) for object classification in sequential data, (2) unsupervised tuning of the network with simultaneous supervised calibration and (3) dynamically changing prediction classes. I will discuss each of the phases in detail.

2.1 RCNN for Object Classification in Sequential Data

CNNs are well suited to object classification, while RNNs with *e.g.* LSTMs can model short and long term dependencies in the data [2]. The first stage of the project is to develop an RCNN for object classification in RGB videos, possibly augmented with depth data or working on pure geometrical data *e.g.* point clouds. Training problems might arise due

to high correlation between subsequent video frames. They can be mitigated by holding a reservoir of seen samples [3] or by an experience replay mechanism [4]. The resulting system should resemble [2] and it should take around 3 months to complete.

2.2 Unsupervised Tuning with Supervised Calibration

Unsupervised learning is superior to supervised learning in that it needs no labels and can act on raw data, which is typically available in unlimited quantities. It has been shown that unsupervised training of neural networks provides good initialization for supervised finetuning [5] as well as delivers good features that can be used for classification by feeding them to a classifier directly [6]. The next step is to use a combination of supervised and unsupervised approach for lifelong learning and adaptation to the changing environment.

Firstly, the CNN part of the system (and possibly the recurrent part as well) is tuned in an unsupervised way as an autoencoder or in an adversarial setting [6]. This should adjust the convolutional filters so that they are highly responsive to recently seen object classes. Secondly, due to changing response pattern of the CNN, the last fully-connected layers of the network, which act as a classifier, can decalibrate. This could be addressed by simultaneous fine-tuning of the whole network on a (small) training set.

In order for this approach to work, appropriate RCNN architectures need to be found and some adjustments to learning algorithms might be needed. It would be beneficial to merge unsupervised and supervised tuning into a unified end-to-end approach. For now, it is not clear how those two tasks can be done at the same time and yet separately. Some challenges with computational complexity of this approach might arise. Since it operates on a real working system (mobile robot), the adjustment mechanism should work in or close to real-time. I expect this phase to take 12 to 15 months.

2.3 Dynamically Changing Prediction Classes

As an agent moves through the environment, a type of a scene it is in might change and with it categories of objects that are likely to be present there. A classifier of fixed architecture might be very complex and it might waste processing power if it is to care about every possible object class regardless of the scene type. The purpose of this stage of the project would be to find out how the classes a classifier can predict could be changed at runtime.

One possible solution is the following. Let A the main classifier be fitted to the most prevalent classes in the training set and let B another classifier that is able to recognize whole groups of object classes. At any given point in time A can recognize only a subset of all classes, while B maintains a ranking of groups of classes according to their probability given the data seen so far. When the ranking changes significantly, *i.e.* a group becomes very probable that hasn't been probable before, the final fully connected layers of A could be refitted to predict

a union of the most probable groups of classes as given by B. It can be seen as an attention mechanism that pays higher attention to most probable classes given the characteristic of the environment, which slightly resembles [7].

Challenges of this phase might arise from computational complexity, since refitting might require considerable resources and from finding a well-performing algorithm for scoring groups of classes or detecting changes of a scene type. The final goal would be to produce an end-to-end differentiable architecture, that can learn the classes it should predict from the data by gradient-based methods. I anticipate it to take around 12 months, with the remaining research time dedicated to unifying and improving the whole architecture, experiments and casting parts of the approach as a Reinforcement Learning problem.

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[2] M. S. Pavel, H. Schulz, and S. Behnke, "Recurrent convolutional neural networks for object-class segmentation of rgb-d video," in *Neural Networks (IJCNN), 2015 International Joint Conference on*, pp. 1–8, IEEE, 2015.

[3] J. S. Vitter, "Random sampling with a reservoir," *ACM Transactions on Mathematical Software (TOMS)*, vol. 11, no. 1, pp. 37–57, 1985.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[5] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[6] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[7] S. Sukhbaatar, J. Weston, R. Fergus, *et al.*, "End-to-end memory networks," in *Advances in Neural Information Processing Systems*, pp. 2431–2439, 2015.