ISS-P-117

Autoencoders with Deformable Templates for Image Reconstruction

SATRE Grégoire*, OCHIAI Tsubasa*, KATAGIRI Shigeru*, OHSAKI Miho*
*Doshisha University

1. Introduction

The current state-of-the-art models in image recognition make principal use of deep Convolutional Neural Networks (CNNs). While these networks are powerful feature extractors, these features are often not very efficient. Pooling layers in CNNs offer some translational invariance but, for example, the same lower-level feature are sometimes repeated several times along different orientations [1]. It would thus be desirable to explore more robust features that are invariant to spatial transformations in general.

2. Purpose

This study is directly inspired by transforming autoencoders [2]. Instead of learning scalar “neurons”, the objective is to learn “capsules” which not only detect the presence of a feature but also, for example, its position, orientation, or scale. The issue considered in this work is how to learn the first layer of capsules from the pixel intensities of input images. While a slight variation on classical autoencoders was suggested in [2], the team in [3] built a decoder with the specific purpose of learning such capsules from the code produced by the encoder. The aim of this work is to verify the effects of the approach in [3].

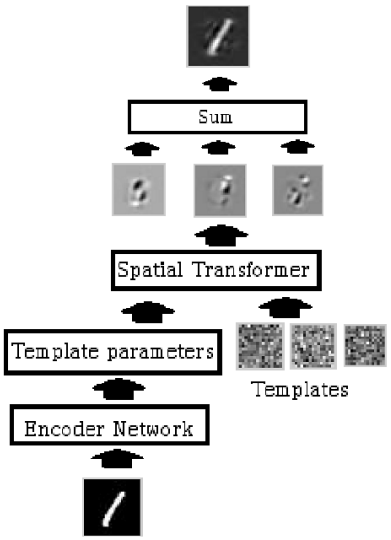

Figure 1. Autoencoder with deformable templates.

3. Model

The main issue related to capsules is to ensure that the autoencoder learns a few salient features, each of which consists of a certain amount of parameters, instead of many simpler features. The model is expected to enforce this feature extraction through the structure of the decoder. The encoder, a standard feedforward neural network, learns a set of templates of canonical representations of features (e.g. straight lines, curves) and parameters describing them in the input image. The decoder transforms each template using the corresponding parameters [4] and sums them all together to form the final output. Figure 1 shows a graphical representation.

4. Experiments and results

In our experiments, the encoder consisted of a feedforward neural network with 3 layers of size 500. The encoder outputs 10 capsules with 5 parameters each (1 for intensity, 2 for position, 1 for orientation, and 1 for scale), with size 15×15 corresponding templates. The loss to minimize was the sum of the squared errors between input images and their reconstructions, averaged over the size of the batch.

Two types of experiments were conducted on MNIST digits. In the first, the model was trained to overfit on a small sample of 100 images. The second was conducted on the full 60,000 MNIST images. In both experiments, reconstructions improved throughout learning, with the first reaching zero training error. The templates, however, did not change from their initial state of Gaussian noise.

5. Conclusion

The model seems to be learning parameters so that the sum of transformed templates gives decent reconstructions, but does not learn the templates themselves. Investigations into such hyper-paramaters as gradient norm and locally adaptive learning rates have not solved this unexpected result. Work into examining the precise behavior of the transformer block in the decoder is ongoing.

Acknowledgements

This work was supported in part by JSPS Grants-in-Aid for Scientific Research No. 26280063 and MEXT-Supported Program ″Driver-in-the-Loop″.

References

[1] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer vision-ECCV 2014* (pp. 818-833). Springer International Publishing.

[2] Hinton, G. E., Krizhevsky, A., & Wang, S. D. (2011). Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning-ICANN 2011* (pp. 44-51).

[3] Tieleman, T. (2014). *Optimizing neural networks that generate images* (Doctoral dissertation, University of Toronto).

[4] Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. In *Advances in Neural Information Processing Systems* (pp. 2008-2016).