ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Sequence Classification using Ensembles of Recurrent Generative Expert Modules

Marius Hobbhahn, Martin V. Butz, Sarah Fabi and Sebastian Otte

University of Tübingen - Neuro-Cognitive Modeling Group
Sand 14, 72076 Tübingen - Germany

Abstract.
Successful discriminative deep learning relies on large amounts of data and proper domain coverage. We introduce an ensemble of recurrent generative modules, achieving robust and effective sequence classification facing sparse data. Each module is an expert for only a few variations of a certain class. Given an input trajectory, the latent codes of the experts are adapted via back-propagation of the reconstruction error and the most accurate expert yields the class. In comparison with direct discriminative models, our approach achieves better classification rates with fewer training examples, can be easily extended, and provides fully transparent decisions.

1 Introduction

It is astonishing how fast we partially learn from few examples. For example, we can easily recognize new objects and faces under various background or lighting conditions. In contrast, learning to recognize and distinguish particular types of objects with deep learning models relies on hundreds of thousands and even millions of supervised example presentations and model adjustment steps. Furthermore, the quality of such discriminative models and their robustness against disturbances strongly depend on how well the training data covers the application domain. Still, even well-trained models can be rather easily fooled [1]. In contrast, the brain is a predictive machine on many levels (cf. *the predictive mind* [2] or *predictive coding* [3]). It constructs a generative model of the encountered reality, instead of just processing the input in a forward pipeline fashion. Even object recognition is a highly recurrent process [4].

In our recent work [5, 6], we have shown that adaptive, goal-directed behavior and learning can unfold in generative forward models. These architectures continuously predict future sensory signals and simultaneously reflect on the recent past. They integrate *active inference* [7], *event segmentation theory* [8], and *predictive coding* [3]. One particular insight is that temporal gradients can be used to infer stable latent codes parameterizing the dynamic generator (cf. parametric biases, e.g. [9]), such that it best explains the recently experienced sensorimotor input history. As a result, event sequences and their boundaries become encoded in a somewhat symbolic fashion [6].

This paper asks the question whether temporal gradient-based latent input inference is applicable in regular sequence classification tasks, specifically when facing sparse data. Preliminary experiments indicated that overly complex recurrent generative models lead to a very irregular latent space and less useful gradient signals. As shown elsewhere [10], an adequate pre-modularization with low-complexity recurrent modules can dramatically improve the overall performance.

We show that temporal gradient-based latent code inference enables the fitting of the predictor's dynamics onto unseen patterns. The system can reconstruct previously unseen target sequences by adjusting its latent code by back-propagating prediction error signals. Moreover, we demonstrate the effectiveness of an ensemble of competing pattern-fitting modules for classification purposes, comparing it with traditional recurrent neural networks (RNN) classifiers with very few training data.

2 Classification using Generative Models

The basis of this paper is a recurrent generative model, which is trained to generate a few variants of a particular type of sequence (class), such as different variants of one letter. The target variant is indicated via a latent input code (one-hot encoding). Once the model is trained, it can be used to generate an unseen sequence by iteratively adapting an initial latent input vector $\mathbf{v}^{\tau}$ via temporal gradient information based on the error $\mathcal{L}$ between the currently generated output sequence (unrolled from the current latent vector) and the given example sequence, as shown in Figure 1. Over time, the output converges towards the target. Our hypothesis is that if the considered model is an expert for the presented example it will quickly find a generative latent code to approximately generate the target, while this will hardly be possible for a non-expert model.

In order to perform classification, however, we established an ensemble of n generative expert modules (EGEM)—one expert module per class. Specifically, the modules are implemented using LSTMs [11] with ten units per network. The individual modules are experts for their learned samples and can therefore reconstruct plausible variations of them. A z-module, for example, is a bad expert for a 'b' target sequence. The error between the reconstruction and the target is large, as exemplified in Figure 2. We control the size of the individual modules—and thus their complexity—thus ensuring efficiency and keeping the latent spaces simple and more stable. The final classification result is determined through competition within the ensemble, i.e., the expert with the best reconstruction determines the class.

3 Experiments

The experiments were conducted on two trajectory datasets. The first dataset contains handwritten digits (character trajectories data set) taken from the UCI machine learning repository [12]. A character is represented as the sequence of its coordinates over time. It contains only characters that can be written in a single stroke, leading to 20 different classes with 2 858 example characters

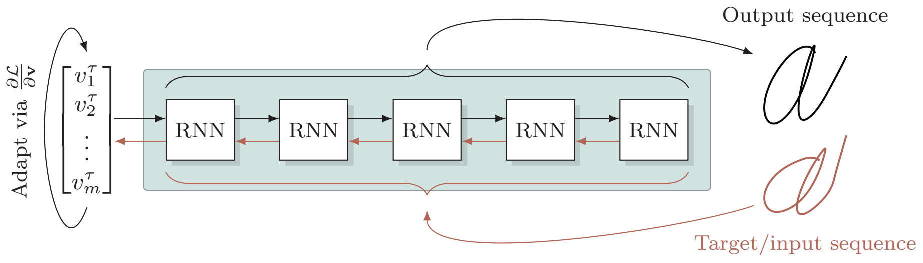

Fig. 1: Depiction of one expert module. The output is generated through the RNN and the latent vector is adapted using the temporal gradient information.

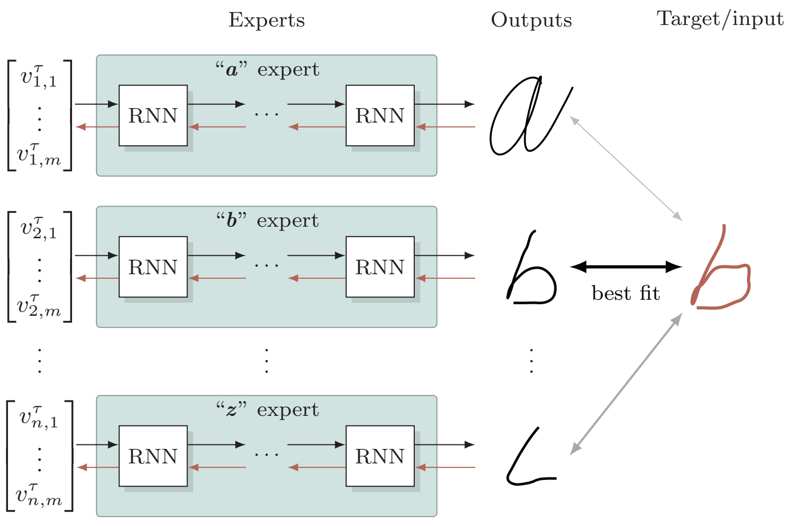

Fig. 2: Illustration of the ensemble-based classification. The experts generate their best possible input reconstructions. The expert with the closest fit 'wins'.

in total. The sequences are padded to have the same length and are globally standardized. The second dataset is MNIST, a well known, large database of handwritten digits. Every example is represented as a 28 by 28 pixel gray scale image, here normalized to the value range $[0, 1]$. In total there are 60 000 training and 10 000 test images. In this application we used MNIST in a sequential manner: every image is interpreted a sequence of 28 pixel rows of size 28.

Since we explore the low-data regime, we trained on only 80 samples for the character dataset and 40 for MNIST, i.e. with only four examples per class. The generative modules were trained for

We first compare performance of EGEM with and without latent vector adaption, by comparing performance between a zero latent vector and the latent vector values generated by 50 iterations of gradient-based updates. Second, we analyzed effects of training sequence selection. In some settings we might

Table 1: Achieved accuracies for EGEM without noise (base), EGEM with noise, a conventional forward LSTM classifier, and nearest neighbor (NN).

Dataset (method)	MNIST	Character sequences
clustered (EGEM base)	0.834 ± 0.020	0.791 ± 0.021
clustered (EGEM noise average)	0.874 ± 0.008	0.889 ± 0.014
clustered (EGEM noise best)	**0.892**	0.905
clustered (forward LSTM)	0.645± 0.015	0.620 ± 0.065
clustered (NN)	0.873	**0.927**
random (EGEM base)	0.525 ± 0.030	0.771 ± 0.025
random (EGEM noise average)	0.595 ± 0.028	0.834 ± 0.034
random (EGEM noise best)	**0.643**	**0.9**
random (forward LSTM)	0.480 ± 0.020	0.575 ± 0.112
random (NN)	0.586 ± 0.034	0.828 ± 0.023
full dataset (forward LSTM)	0.99 ± 0.0	0.978 ±0.006

encounter representative samples of a dataset, in others not. We simulated representative sample selection by training with the K-Means clusters of each class (with dynamic time warping-based distance measures), referred to as "clustered" training samples. We compare performance with the one achieved when randomly sampling data trajectories for each class.

Third, we add noise to input and output during training. The addition of noise on the input was used to smooth the latent space. The noise was normally distributed with different standard deviations. It was additionally clipped such that every value lied between 0 and 1. When a recurrent generative model is trained only on clean one-hot vectors, it is less likely to have a smooth latent space. The addition of noise to the output was intended to make the model more robust.

Lastly, EGEM was compared to conventional forward LSTMs (with 100 units for MNIST and 200 units for characters), trained on the same data, and nearest neighbor classifiers (NN). NN methods are, despite their simplicity, often the most accurate techniques in the low data regime and therefore a good baseline. We conducted every experiment five times and report averages and standard deviations.

4 Results

First, we found that EGEM was able to reconstruct previously unseen sequences reliably. Selected examples can be found in Figure 3. Its capability becomes especially clear when some features of the target cannot be found in any of the training samples, e.g., the bend in Target 2 of the characters dataset.

The results of the first subexperiment show that 50 gradient-based updates of the latent vector code improve accuracy by $10\% - 20\%$ on average and improve the MSE significantly. The results of the other three subexperiments are shown

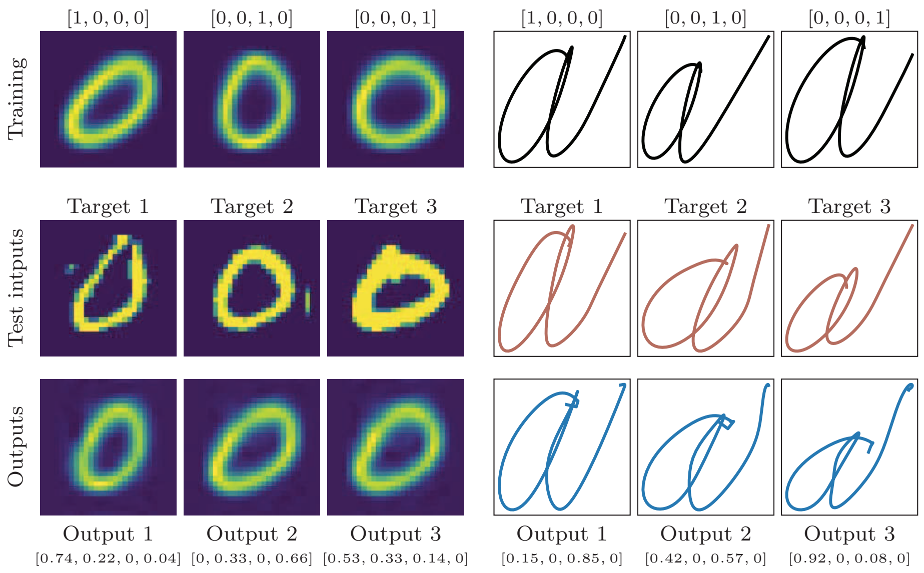

Fig. 3: Reconstruction examples of single modules for MINST (left) and character trajectories (right). The upper row shows exemplary training sequences. The second row shows some reconstruction targets, whereas the third row shows the respective reconstructions including final latent vectors.

in Table 1. They show that on average (a) all methods perform better when trained on representative samples compared to random ones; (b) the addition of noise to the input is always increasing accuracy, while adding noise to the output does so only sometimes; (c) EGEM is able to outperform a forward model by large margins in all cases and beats the NN in three out of four cases. Additionally, in the random case EGEM clearly outperforms NN. Since data points in real application scenarios are unlikely to be clustered and very likely to be random representatives, this strongly speaks in favor of EGEM.

5 Conclusion

This work has shown that iterative gradient based inference applied on an ensemble of generative expert modules (EGEM) is a good method for sequence classification, particularly when only few data instances are available for training. Expert modules that are able to reconstruct previously unseen patterns of a respective sequence type unfold classification capabilities via a competition process: the expert that best reconstructs the given input determines the label. In our experiments, EGEM outperformed conventional LSTM classifiers by large margins. Further advantages of EGEMs are that they are easily extendable and allow continuous, life-long learning by just adding further experts for new classes. Moreover, the decisions made are fully transparent and can be retraced, which is essential when striving for explainable AI. To which extent EGEMs are robust against adversarial attacks remains to be shown, but it is likely that the generative aspect of EGEMs could be advantageous here as well.

A clear limitation is that more training data does not necessarily mean a higher accuracy for EGEM. The generative model is likely to converge to one class template, which averages over all presented training samples, thereby limiting the effect of the gradient based method. However, strengthening the generative capabilities of the experts, e.g., by enhancing their representational richness, could improve the overall performance. Overall, we see strong connections to theories on event-predictive cognition [6, 13], where event-predictive inference determines perception, allows for meaningful compositions of previously learned knowledge, and the flexible and goal-directed generation of behavior. However, more loose and emergent modularizations still need to be pursued.

References

[1] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. *CoRR*, abs/1707.08945, 2017.

[2] Jakob Hohwy. *The Predictive Mind*. Oxford University Press, November 2013.

[3] Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, January 1999.

[4] Kohitij Kar, Jonas Kubilius, Kailyn Schmidt, Elias B. Issa, and James J. DiCarlo. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature Neuroscience*, 22(6):974–983, June 2019.

[5] Sebastian Otte, Theresa Schmitt, Karl Friston, and Martin V. Butz. Inferring Adaptive Goal-Directed Behavior within Recurrent Neural Networks. In *Artificial Neural Networks and Machine Learning – ICANN 2017*, number 10613 in Lecture Notes in Computer Science, pages 227–235. Springer International Publishing, September 2017.

[6] Martin V. Butz, David Bilkey, Dania Humaidan, Alistair Knott, and Sebastian Otte. Learning, planning, and control in a monolithic neural event inference architecture. *Neural Networks*, May 2019.

[7] Karl Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, and Giovanni Pezzulo. Active Inference: A Process Theory. *Neural Computation*, 29(1):1–49, November 2016.

[8] Jeffrey M. Zacks, Nicole K. Speer, Khena M. Swallow, Todd S. Braver, and Jeremy R. Reynolds. Event perception: A mind-brain perspective. *Psychological Bulletin*, 133(2):273–293, 2007.

[9] Yuuya Sugita, Jun Tani, and Martin V Butz. Simultaneously emerging braitenberg codes and compositionality. *Adaptive Behavior*, 19:295–316, 2011.

[10] Sebastian Otte, Patricia Rubisch, and Martin V. Butz. Gradient-based learning of compositional dynamics with modular rnns. In *Artificial Neural Networks and Machine Learning – ICANN 2019*, number 11727 in Lecture Notes in Computer Science, pages 484–496. Springer International Publishing, September 2019.

[11] Felix A. Gers, Jürgen A. Schmidhuber, and Fred A. Cummins. Learning to forget: Continual prediction with lstm. *Neural Comput.*, 12(10):2451–2471, October 2000.

[12] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[13] Martin V. Butz. Towards a unified sub-symbolic computational theory of cognition. *Frontiers in Psychology*, 7(925), 2016.