ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Continual Learning with Echo State Networks

Andrea Cossu[1,2], Davide Bacciu[1], Antonio Carta[1],
Claudio Gallicchio[1] and Vincenzo Lomonaco[1] *

1- University of Pisa - Department of Computer Science
Largo B. Pontecorvo, 3, 56127, Pisa - Italy

2- Scuola Normale Superiore
Piazza dei Cavalieri, 7, 56126, Pisa - Italy

Abstract. Continual Learning (CL) refers to a learning setup where data is non stationary and the model has to learn without forgetting existing knowledge. The study of CL for sequential patterns revolves around trained recurrent networks. In this work, instead, we introduce CL in the context of Echo State Networks (ESNs), where the recurrent component is kept fixed. We provide the first evaluation of catastrophic forgetting in ESNs and we highlight the benefits in using CL strategies which are not applicable to trained recurrent models. Our results confirm the ESN as a promising model for CL and open to its use in streaming scenarios.

1 Introduction

Real world environments where data is highly non stationary represent a challenge for current machine learning solutions. Continual Learning (CL) focuses on the design of new models and techniques able to learn new information while preserving existing knowledge [1]. CL models receive a (possibly infinite) stream of experiences $e_1, e_2, e_3, ...$, where each experience e_i contains data sampled from an associated distribution D_i. Drifts in data distribution usually occur during a known transition to a new experience. We study CL from the perspective of catastrophic forgetting: when trained sequentially on multiple experiences, neural networks tend to forget previous knowledge, that is, they reduce their performance on previously seen data and tasks.
We address CL in sequential data processing (where each pattern is a sequence) with a family of randomized recurrent neural networks called Echo State Networks (ESNs) [2]. ESNs process sequences by means of a fixed recurrent component initialized with stable dynamics (reservoir) whose output is fed to a trainable output layer (readout). ESNs are promising architectures for challenging contexts such as neuromorphic hardware and embedded intelligence applications.
The topic of CL with recurrent neural networks has started to gain attention from the CL community. Recent works on the behavior of recurrent models in non stationary environments highlighted the fact that common CL strategies may behave differently than expected when applied to recurrent models (e.g. their performance may be influenced by the sequence length) [3]. While these studies focus on fully trained recurrent models, we instead provide the first experimental analysis of catastrophic forgetting in ESNs, and their use with popular CL

*This work has been partially supported by the H2020 TEACHING project (GA 871385).

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

strategies. By treating the untrained reservoir of an ESN as a feature extractor for sequences, we were able to 1) restrict the application of CL strategies to the final, linear readout and 2) to leverage efficient CL strategies operating solely on the final layer. The latter point marks a clear advantage in using ESNs, since many approaches in CL for computer vision often exploit pretrained feature extractors. While such methods could be directly applied to ESNs, it is generally difficult to do the same for other recurrent models, where the use of pretrained networks is not as effective. Our results indicate that ESN exhibits competitive performance with respect to LSTM and it is amenable to be applied with CL solutions operating in challenging settings like streaming learning.

2 Continual Learning and Recurrent Models

Continual Learning The study of recurrent models in CL currently focuses on deep recurrent networks trained by backpropagation [3, 4]. The problem has been tackled both by designing new techniques and learning algorithms [5] and by applying popular CL strategies not designed for sequential data [6, 4]. Notably, the CL performance of alternative recurrent paradigms like spiking neural networks [7] and reservoir computing [8] is under-documented. To the best of our knowledge, there is only one work about CL and ESNs [8] which, however, focuses on a specific application and does not give insights on how to use ESNs in different CL contexts. This highlights the need for a broad experimental evaluation with different families of CL strategies on popular benchmarks.

Continual Learning strategies Here, we briefly introduce the CL strategies used in our experiments. These strategies are not specifically designed for sequential data processing. Therefore, they allow us to draw more general conclusions on the model behavior.

Elastic Weight Consolidation (EWC) [9] and Learning without Forgetting (LwF) [10] are regularization strategies: they add a penalty to the loss function to improve model stability. EWC prevents large changes in parameters deemed important for previous experiences. The loss penalty at experience n takes the form of $\sum_{t=1}^{n-1} \mathbf{\Omega_t}(\theta_\mathbf{t} - \theta_\mathbf{n})^2$, where $\theta_\mathbf{t}$ are the model parameters at experience t and $\mathbf{\Omega}$ their corresponding importances. At the end of each experience, the parameters importances for that experience are computed by freezing the model, performing an additional pass over training data and averaging the squared gradients across all patterns.

LwF enforces stability in the output layer activations by keeping a copy of the previous model and by taking its output on the current training patterns as soft targets in the distillation loss. Therefore, the loss penalty takes the form of the the KL-divergence $\mathrm{KL}[p_{\theta_\mathbf{t}}(\mathbf{x_t})||p_{\theta_{\mathbf{t-1}}}(\mathbf{x_t})]$, where $p_{\theta_\mathbf{t}}(\mathbf{x_t})$ represents the output of the model parameterized by $\theta_\mathbf{t}$.

Replay [11] leverages a different paradigm: on each experience, it randomly samples a number of patterns from the training set and adds them to the replay memory. During training, the current minibatch is augmented with an addi-

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

tional minibatch of patterns sampled directly from the memory. This is one of the most effective strategies in CL [12].
Finally, Streaming Linear Discriminant Analysis (SLDA) [13] is a strategy which leverages a pretrained feature extractor G (a ResNet-18 in the original paper) combined with a linear layer to compute the final output $\mathbf{y} = \mathbf{W}G(\mathbf{x}) + \mathbf{b}$, where $\mathbf{x}$ is the current input pattern. This strategy operates in a streaming setting, where patterns are seen one at a time and only once (one epoch only). The parameters $\mathbf{W}$ and $\mathbf{b}$ of the linear layer are computed through an online approximation of the Linear Discriminant Analysis. The algorithm keeps a mean vector together with an associated counter per class and a shared covariance matrix, updated during training. Since SLDA requires a fixed feature extractor, we can only apply it to ESNs thanks to its untrained recurrent layer.
To provide lower and upper bounds performance, we ran experiments with other 2 strategies: Naive, which simply finetunes the network across experiences without any CL strategy, and Joint Training, which trains the model in an offline setting with all the data available from the beginning.

Echo State Networks Reservoir Computing provides a general framework to build recurrent networks [14]. ESNs [2] belong to the reservoir computing paradigm since they are composed by an untrained reservoir and a trained linear readout. The reservoir is composed by a set of randomly connected units and it represents the recurrent component of the architecture. The initialization of recurrent weights matrix in the reservoir is a crucial hyperparameter: usually, the matrix values are scaled such that its spectral radius is slightly smaller than one (necessary condition for the Echo State Property). The linear readout parameters can be trained with closed-form solutions like pseudo-inverse or ridge regression, which however requires to store the entire set of reservoir activations.

3 Experiments

We compared the performance of LSTM and ESN when equipped with the 4 CL strategies already presented: EWC, LwF, Replay, SLDA. We tested our methods on 2 different sequence classification tasks, in which each pattern (a sequence) is associated to a target class. We chose 2 popular class-incremental CL benchmarks, where each experience provides examples from new classes, which will be present only in that experience. At the end of training on the last experience, the model performance is measured against data coming from all experiences. During testing, the model has no knowledge about the experience from which each pattern is coming from.

Experimental setup We used Split MNIST (SMNIST) and SSC [3] as the benchmarks for our experiments. SMNIST provides 5 different experiences, each of which contains examples of MNIST dataset from 2 digit classes. In order to use SMNIST as a sequence classification task, we took each image one row at a time, resulting in input sequences with 28 steps. SSC is a dataset of spoken words.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

SMNIST	LSTM†	ESN
EWC	$0.21_{\pm 0.02}$	$0.20_{\pm 0.00}$
LWF	$0.31_{\pm 0.07}$	$0.47_{\pm 0.07}$
REPLAY	$\mathbf{0.85_{\pm 0.03}}$	$0.74_{\pm 0.03}$
SLDA	—	$\mathbf{0.88_{\pm 0.01}}$
NAIVE	$0.20_{\pm 0.00}$	$0.20_{\pm 0.00}$
JOINT	$0.97_{\pm 0.00}$	$0.97_{\pm 0.01}$

SSC	LSTM†	ESN
EWC	$0.10_{\pm 0.00}$	$0.09_{\pm 0.02}$
LWF	$0.12_{\pm 0.01}$	$0.12_{\pm 0.02}$
REPLAY	$\mathbf{0.74_{\pm 0.07}}$	$0.36_{\pm 0.07}$
SLDA	—	$\mathbf{0.57_{\pm 0.03}}$
NAIVE	$0.10_{\pm 0.00}$	$0.10_{\pm 0.00}$
JOINT	$0.89_{\pm 0.02}$	$0.91_{\pm 0.02}$

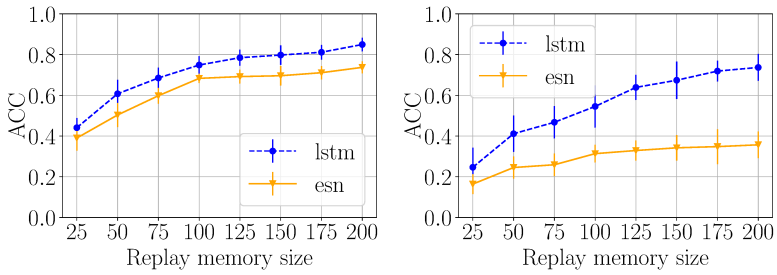
Table 1: Mean ACC and standard deviation over 5 runs on SMNIST and SSC benchmarks. SLDA is applied only to ESN since it assumes a fixed feature extractor. SMNIST contains 5 experiences, while SSC contains 10 experiences. † results are taken from [3], except for replay which has been recomputed to guarantee the use of the same replay policy (200 patterns in memory).

We took 10 experiences, each of which containing patterns representing 2 words. Sequences have 101 steps. We performed grid search on all the strategies for ESN and on Replay for LSTM. The other results for LSTM are taken from [3], since the experimental setup is the same. To perform grid search for SSC, we took 3 held-out experiences for model selection and 10 for model assessment. To fairly compare ESN and LSTM, we select a model configuration whose only requirement is to be able to learn effectively at training time. Then, the performance in terms of forgetting depends mostly on the CL strategy (subjected to grid search) and not on the specific model setting. We train the ESN readout with Adam optimizer and backpropagation, since CL requires to update the model continuously, possibly without storing its activations. We used the Avalanche [15] framework for all our CL experiments. We make publicly available the code together with configurations needed to reproduce all experiments[1]. We monitored the average accuracy (ACC) metric: after training on all experiences we measure the accuracy averaged over test patterns from all the experiences.

Results Table 1 reports the ACC metric and its standard deviation over 5 runs for the best configuration found in model selection. Our results show that ESN performs better or comparably to the LSTM network. EWC is not able to tackle class-incremental scenarios, neither with LSTM nor with ESN. It achieves a performance equal to the Naive one. LwF, instead, manages to reduce forgetting in the simplest SMNIST benchmarks. In this context, applying LwF only on the feedforward component results in a better performance with respect to the LSTM. This is compatible with results presented in [3] and highlights one of the advantage in using ESNs for CL. However, when facing more complex scenarios like SSC, LwF fails to provide any benefits with respect to Naive finetuning. For replay, we studied the performance of ESN and LSTM with different memory sizes (Fig. 1). ESN exhibits a lower performance with respect to a fully trained

[1] `https://github.com/Pervasive-AI-Lab/ContinualLearning-EchoStateNetworks`

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.



(a) Split MNIST (b) SSC

Fig. 1: Accuracy over increasing replay memory sizes.

LSTM. On SMNIST, the gap becomes significant only for large memory sizes. On SSC, instead, the difference in performance occurs also for smaller memories. This result is surprising since replay is not usually sensitive to the choice of the model. A deeper investigation of different replay policies and ESNs architectures will be needed in order to discover possible solutions to the problem.
The advantage in using ESNs clearly emerges when studying the behavior of SLDA strategy. This strategy effectively tackles class-incremental benchmarks like SSC. The absolute performance of SLDA in SSC is still far from the joint training upper bound. However, this does not mean that SLDA suffers from large forgetting effect. In fact, it achieves an average experience forgetting (difference between the accuracy after training on a certain experience and the corresponding accuracy after training on all experiences) of 0.14 ± 0.02. The remaining 20% of difference from the joint training is explained by the fact that SLDA is a streaming strategy which trains the model only on a single epoch. Therefore, on complex scenarios like SSC it achieves a lower accuracy.

4 Conclusion and Future Work

We studied the ability of ESNs to mitigate forgetting in CL environments. We provided the first experimental evaluation of ESNs trained with popular CL strategies. Our analysis showed that, apart from replay strategies, ESNs perform comparably with fully trained recurrent networks like LSTM. Moreover, since the reservoir is a fixed feature extractor, it is possible to train ESNs with CL strategies like SLDA which are not applicable to LSTM. SLDA obtains a good performance in class-incremental scenarios.
This work could foster new studies and applications of CL with ESNs: the renowned computational efficiency of ESNs may be particularly interesting for streaming or task-free CL. The possibility to implement ESNs in neuromorphic hardware opens to the continuous training of such models on low resources devices. ESNs are not the only family of models with an untrained component. More in general, CL with partially trained networks constitutes an interesting avenue of research, due to the fact that fixed connections are not subjected to

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

catastrophic forgetting. Alternatively, reservoir in ESNs may also be finetuned during training to better adapt to new experiences. In particular, unsupervised finetuning through backpropagation-free methods (e.g. Hebbian learning, intrinsic plasticity) may provide quicker adaptation and more robust representations. The design of new CL strategies which exploit reservoir finetuning while keeping forgetting into consideration would provide us with a deeper understanding of the CL learning capabilities of ESNs.

References

[1] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58, 2020.

[2] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304, 2004.

[3] Andrea Cossu, Antonio Carta, Vincenzo Lomonaco, and Davide Bacciu. Continual Learning for Recurrent Neural Networks: An Empirical Evaluation. *arXiv*, 2021.

[4] Shagun Sodhani, Sarath Chandar, and Yoshua Bengio. Toward Training Recurrent Neural Networks for Lifelong Learning. *Neural Computation*, 32, 2019.

[5] Lea Duncker, Laura N Driscoll, Krishna V Shenoy, Maneesh Sahani, and David Sussillo. Organizing recurrent network dynamics by task-computation to enable continual learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[6] Andrea Cossu, Antonio Carta, and Davide Bacciu. Continual Learning with Gated Incremental Memories for sequential data processing. In *International Joint Conference on Neural Networks*, 2020.

[7] Alexander Ororbia. Spiking Neural Predictive Coding for Continual Learning from Data Streams. *arXiv*, 2020.

[8] Taisuke Kobayashi and Toshiki Sugino. Continual Learning Exploiting Structure of Fractal Reservoir Computing. In *ICANN*, 2019.

[9] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114, 2017.

[10] Zhizhong Li and Derek Hoiem. Learning without Forgetting. In *European Conference on Computer Vision*, 2016.

[11] Anthony Robins. Catastrophic Forgetting; Catastrophic Interference; Stability; Plasticity; Rehearsal. *Connection Science*, 7, 1995.

[12] Ameya Prabhu, Philip H. S. Torr, and Puneet K. Dokania. GDumb: A Simple Approach that Questions Our Progress in Continual Learning. In *ECCV*, 2020.

[13] Tyler L Hayes and Christopher Kanan. Lifelong Machine Learning with Deep Streaming Linear Discriminant Analysis. In *CLVision Workshop at CVPR*, 2020.

[14] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3, 2009.

[15] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L. Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido van de Ven, Martin Mundt, Qi She, Keiland Cooper, Jeremy Forest, Eden Belouadah, Simone Calderara, German I. Parisi, Fabio Cuzzolin, Andreas Tolias, Simone Scardapane, Luca Antiga, Subutai Amhad, Adrian Popescu, Christopher Kanan, Joost van de Weijer, Tinne Tuytelaars, Davide Bacciu, and Davide Maltoni. Avalanche: An End-to-End Library for Continual Learning. In *CLVision Workshop at CVPR*, 2021.