Towards Adaptive and Stable Compositional Assemblies of Recurrent Neural Network Modules

Valerio De Caro, Andrea Ceni, Davide Bacciu, Claudio Gallicchio *

Department of Computer Science, University of Pisa
Largo Bruno Pontecorvo 3, Pisa, Italy school

Abstract. Recurrent neural networks (RNNs) are computational models regarded as dynamical systems. Modularity is a key ingredient of complex systems. Thus, the composition of RNN modules provides a simple paradigm for building complex computational models, with the potential to approach the human brain capability. We devise strategies for training RNNs assembled into a larger RNN of RNNs, provided with theoretical guarantees of stability that hold during training for the composed global network. Experiments on pixel-by-pixel image classification benchmarks prove the effectiveness of this approach.

1 Introduction

In the modern landscape of AI, compositional learning stands out as a fundamental feature for combining "primitive" learning modules within integrated systems leveraging the representation capabilities of its subparts. This concept mirrors the current understanding of the human brain where, while sub-areas are delegated to specific tasks, the interactions among the areas provide enhanced and more complex representations [1]. Changing the perspective to one of semantic representations of concepts, combining simple concepts into more complex ones is a cornerstone of the human ability to understand, reason, and learn [2]. Equipping learning modules with compositional abilities allows to benefit from the reuse of specialized knowledge, as well as from richer representations to solve intricate problems more accurately [3]. Joining the notion of compositionality to the landscape of dynamical systems leads to the desiderata of achieving the ability to compose multiple dynamical systems into an assembly of dynamical systems, whose dynamics emerging from appropriate modelling of their mutual interaction enrich the information of the singles. This property is particularly beneficial in the context of sequential data processing tasks, where Recurrent Neural Networks (RNNs) represent a model of choice. Interpreting these models as an input-driven dynamical system is a widely adopted practice in literature [4] and, as such, ensuring their *stability* is a crucial aspect to address. In this regard, the seminal work in [5] first investigated stable assemblies of RNNs and provided solution for coupling different stable RNN modules to preserve overall stability. The main idea is to define RNNs that correspond to *contractive* dynamical systems and exploit contract theory results in the literature [6] to

*This work has been supported by NEURONE, a project funded by the European Union - Next Generation EU, M4C1 CUP I53D23003600006, under program PRIN 2022 (prj code 20229JRTZA), and EU-EIC EMERGE (Grant No. 101070918).

build up larger RNNs of RNNs with stability guarantees. Their theoretical investigation is accompanied by the proposal of approaches to guarantee the contractive dynamics of the single RNN modules accordingly. A first strategy relies on complex optimization schemes to allow the adaptation of the RNN modules and the connections between them with stability guarantees. However, a second, more straightforward and better-performing strategy, denoted as Sparse Combo Net, consists of simply *fixing the RNN modules' recurrent weights in a contractive configuration* and only learning the connections between RNN modules. These results highlight how it remains an open and pressing problem to find strategies for adapting the internal weights of the RNN modules to outperform the simple strategy of keeping them fixed. In this paper, we take a step further and focus on constructing *stable* and *fully adaptive* assemblies of RNNs, while relaxing from the complexity arising from the optimization of the RNN modules. A necessary step toward the ambitious goal to unlock the expressive power of an RNN composed of smaller specialized and adaptive RNNs is to ensure the stability of the whole network as a dynamical system, without the burden of dealing with complex optimization schemes to foster stability in the training phase. To achieve this objective, we propose two strategies to allow adaptivity of the internal weights of RNN modules while ensuring contractive dynamics from each. We test these new strategies on the popular permuted sequential MNIST (psMNIST) and sequential MNIST (sMNIST) tasks and compare them to Sparse Combo Net. Experiments demonstrate the effectiveness of our proposal, with the new strategies outperforming the original Sparse Combo Net (SCN) in all cases.

2 Skew-symmetric coupling of RNN modules

We consider a number p of RNN subnetworks of the following type [5]:

$$\tau\dot{\mathbf{x}}_i = -\mathbf{x}_i + \mathbf{W}_i\phi(\mathbf{x}_i) + \mathbf{u}_i(t), \qquad i = 1, \dots, p, \tag{1}$$

where $\mathbf{x}_i$ is the hidden state of the i-th RNN, $\mathbf{W}_i \in \mathbb{R}^{N\times N}$ are the recurrent connections, $\mathbf{u}_i(t)$ the input driving the i-th subnetwork, and ϕ is the nonlinear activation function, tanh in our experiments. We couple the i-th RNN module with the j-th RNN module them via matrices $\mathbf{L}_{ij} \in \mathbb{R}^{N\times N}$ with the skew-symmetric constraint as follows:

$$\mathbf{L}_{ij} = -\mathbf{L}_{ji}^T, \tag{2}$$

so that the overall RNN of RNNs defined by:

$$\tau\dot{\mathbf{x}}_i = -\mathbf{x}_i + \mathbf{W}_i\phi(\mathbf{x}_i) + \sum_{j=1}^{p}\mathbf{L}_{ij}\mathbf{x}_j + \mathbf{u}_i(t), \qquad i = 1, \dots, p, \tag{3}$$

is guaranteed to be a stable system whenever the single RNN modules in (1) are themselves stable [5].

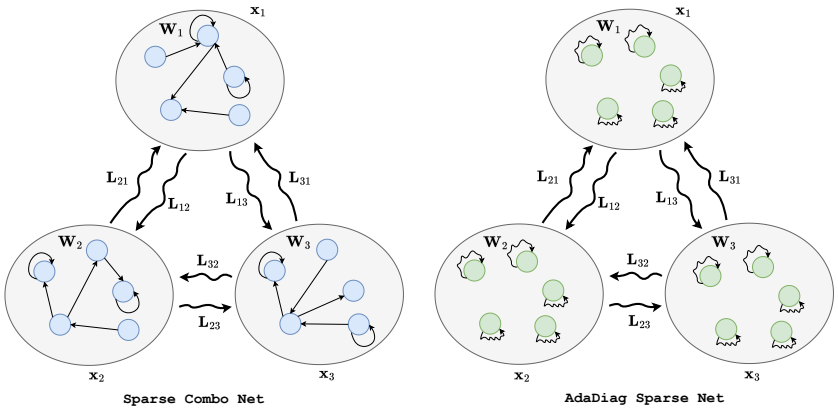

Fig. 1: Depiction of an assembly of RNNs. Straight arrows denote untrained connections, while wavy arrows denote trained connections. **Left**: Sparse Combo Net trains only the connections between RNN modules leaving the internal connections of the single RNN modules untrained. **Right**: AdaDiag Sparse Net (our) trains the connections between RNN modules and also the internal connections of the single RNN modules, which result in adaptive self-loops.

Sparse Combo Net SCN initializes the weights $\mathbf{W}_i$ in eq. (1) according to the criterion in [5, Theorem 1] to achieve a contractive parameterization. Now, such a criterion is too computationally intensive to be checked at each weights-update iteration, they are left untrained throughout the learning process, while the coupling matrices $\mathbf{L}_{ij}$ are trained under the constraint of eq. (2).

Contrariwise, we propose a couple of strategies for adapting the internal weights of RNN modules while ensuring contraction dynamics during training. We use the same methodology as SCN for training the coupling matrices $\mathbf{L}_{ij}$, but we allow the internal connections of the RNN modules to be trained. Both these strategies ensure that the spectral norm of the matrix $\mathbf{W}_i$ is less or equal than 1 during training. This guarantees that each single RNN module, if decoupled from all the others, is contracting [7]. Both our proposals involve the use of single RNN modules whose internal units do not communicate with each other, i.e. the matrices $\mathbf{W}_i$ are structured to be diagonal. We call this model *AdaDiag Sparse Net*. Restricting individual RNN modules to a diagonal form limits their expressiveness, as their neurons remain independent. However, the overall network preserves its representational ability since the coupling matrices $\mathbf{L}_{ij}$ enable communication between neurons across different modules.

AdaDiag Sparse Net with tanh. We consider diagonal matrices $\mathbf{W}_i$ with entries constrained via the component-wise application of the hyperbolic tangent on the entries of $\mathbf{W}_i$. This strategy ensures that the diagonal elements of $\mathbf{W}_i$ assume values in $(-1, 1)$. Therefore, the spectral norm of $\mathbf{W}_i$ is necessarily less than 1.

AdaDiag Sparse Net with clip. We consider diagonal matrices $\mathbf{W}_i$ with entries clipped to 0.99 whenever they exceed the value 1, or clipped to -0.99 whenever they assume a value less than -1. This strategy ensures that the diagonal elements of $\mathbf{W}_i$ assume values in $(-1, 1)$. Therefore, the spectral norm of $\mathbf{W}_i$ is necessarily less than 1.

3 Experimental Assessment

The purpose of our experiments is to provide an analysis of the proposed strategies in comparison with the best model from [5] as baseline. We implemented our version of the models based on the available code at `https://github.com/kozleo/rnns-of-rnns`. In our setup, the assembly size is fixed to 16 recurrent modules, each consisting of 32 units. For an assembly of 16 modules, the total amount of possible coupling blocks is 240, but under the constraint of eq. (2), the total trainable coupling blocks is 120 (i.e., $\frac{16\times15}{2}$). We analyzed the behaviour under different levels of sparsity in the modules' coupling by setting the number of coupling blocks C to 5 and 20. To pursue the aforementioned objective, we configured the initialization and the adaptivity of diagonal blocks in the following four ways: (1) fixed, sparse matrix with 3% of nonzero entries (as in [5]); (2) fixed, sparse matrix with 30% of nonzero entries; (3) diagonal matrix adapted with strategy 1; (4) diagonal matrix adapted with strategy 2. The nonlinearity in eq. (3) is tanh, and the discretization step is 0.03 (using forward Euler method).

We assessed all the configurations on the regular and the permuted version of Sequential MNIST. We trained and validated each configuration on the given train/test split three times. Each run was limited to a maximum of 200 training epochs, and we applied early stopping when reaching a plateau in the validation accuracy.

3.1 Results

In Table 1, we report the performance of all the assessed configurations, plus the performance of a Vanilla RNN as reference. Starting from the configurations with lower coupling among the RNN modules, i.e., with $C = 5$, we can observe that our model outperforms the baseline by $\sim 2\%$ with the clipping method on sMNIST and by $\sim 5\%$ with the tanh method on psMNIST. From these results, we deduce that the accuracy benefits from the adaptivity arising from the interaction between the models when the coupling is low. Even more relevant, when the coupling is higher, i.e., with $C = 20$, we experience a significant improvement in the performance on psMNIST, as the adaptivity of the RNN modules tackles the higher complexity of the task. From a general perspective, we can observe that our strategies are more consistent in terms of performance across the runs, as the standard deviation is significantly lower than the one of Sparse Combo Net in most of the cases as adaptivity of the RNN modules allows to cope with bad parameterization in the initialization phase. This is further supported by the representation of the weights in Figure 2, where we can observe that the entries of the coupling blocks tend to saturate more on the Sparse Combo Net. Hence, we

Block	Method	sMNIST		psMNIST	
		$C = 5$	$C = 20$	$C = 5$	$C = 20$
SCN [5]	Sparse (3%)	$77.99_{\pm 1.58}$	$\mathbf{90.04_{\pm 0.72}}$	$74.32_{\pm 2.51}$	$82.08_{\pm 0.98}$
SCN [5]	Sparse (30%)	$85.44_{\pm 1.67}$	$90.04_{\pm 0.92}$	$80.18_{\pm 0.51}$	$85.91_{\pm 1.36}$
AdaDiag (our)	tanh	$86.60_{\pm 0.68}$	$88.71_{\pm 0.07}$	$\mathbf{85.11_{\pm 0.25}}$	$88.53_{\pm 1.23}$
AdaDiag (our)	clip	$\mathbf{87.74_{\pm 0.11}}$	$88.89_{\pm 0.06}$	$84.27_{\pm 1.10}$	$\mathbf{89.63_{\pm 0.25}}$
Vanilla RNN [8]	dense	49.10	–	–	71.60

Table 1: Performance of different configurations of assemblies (including number of coupling blocks C) on sMNIST and psMNIST. The first two rows correspond to two sparsity settings of the SCN model found in [5]. The third and fourth rows correspond to our methods, as described in Section 2. In the last row, the performance of a fully-connected RNN trained as a monolithic block, with a comparable number of trainable parameters. We report the mean and standard deviation of the test set accuracy averaged over three runs, apart from Vanilla RNN which is taken from [8]. The best result for each dataset and coupling block configuration is highlighted in bold.

conjecture that adapting the inner dynamics of each RNN modules by accounting also for the interactions with other networks allows to better accomodate the knowledge across the assembly, ultimately leading to a performance improvement. We remark that adapting the RNN modules in the training phase does not produce a significant overhead from a computational perspective, as the Sparse Combo Nets required an average time per epoch of $\sim 185s$, against the $\sim 206s$ of our strategies. Finally, we conclude noticing that, training as a monolithic block a Vanilla RNN results in poorer performance in the considered classification benchmarks, despite its theoretically greater expressive power due to lack of architectural constraints.

4 Conclusions

In this paper, we addressed the problem of constructing a stable and adaptive assembly of Recurrent Neural Networks (RNNs). We laid the foundations of our work on the theoretical results from [5], where the best-performing approach proposed requires all the RNNs of the assembly to stay *fixed* throughout the training process. Motivated by the potential benefit in the representation capabilities of the assembly by allowing the adaptivity of the modules, we proposed two strategies where the RNNs' weight matrices are diagonal (thus encompassing the exact eigenspectrum of the recurrent transformation), and their values are bounded in the $(-1, 1)$ interval. In our experiments, compared the proposed methodologies with the fixed baseline in [5] on sMNIST and psMNIST, with different levels of coupling among the RNNs. The results showed that the adaptivity of the RNNs leads to outperform the baseline in most of the cases, while maintaining better consistency across the runs. This highlighted that the inner dynamics

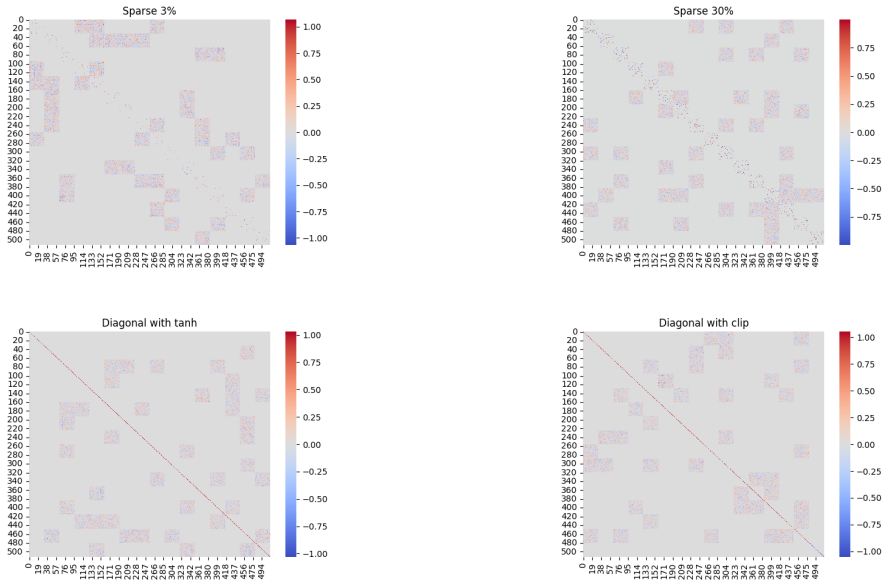

Fig. 2: Representation of the weights of the RNN of RNNs for each strategy on psMNIST with $C = 20$.

of each RNN module benefit from adapting according to the dynamics of the others. In the future, we aim to generalize our approach to different architectures while investigating further the theoretical foundation of stable RNNs composed of smaller adaptive RNNs.

References

[1] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, p. e253, 2017.

[2] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, S. Mack, *et al.*, *Principles of neural science*, vol. 4. McGraw-hill New York, 2000.

[3] S. Sinha, T. Premsri, and P. Kordjamshidi, "A survey on compositional learning of ai models: Theoretical and experimetnal practices," *arXiv preprint arXiv:2406.08787*, 2024.

[4] B. Chang, M. Chen, E. Haber, and E. H. Chi, "Antisymmetricrnn: A dynamical system view on recurrent neural networks," *arXiv preprint arXiv:1902.09689*, 2019.

[5] L. Kozachkov, M. Ennis, and J.-J. Slotine, "Rnns of rnns: Recursive construction of stable assemblies of recurrent neural networks," *Advances in neural information processing systems*, vol. 35, pp. 30512–30527, 2022.

[6] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for non-linear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[7] I. B. Yildiz, H. Jaeger, and S. J. Kiebel, "Re-visiting the echo state property," *Neural networks*, vol. 35, pp. 1–9, 2012.

[8] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. A. Hasegawa-Johnson, and T. S. Huang, "Dilated recurrent neural networks," *Advances in neural information processing systems*, vol. 30, 2017.