RECURSIVE CONSTRUCTION OF STABLE ASSEMBLIES OF RECURRENT NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Advanced applications of modern machine learning will likely involve combinations of trained networks, as are already used in spectacular systems such as DeepMind's AlphaGo. Recursively building such combinations in an effective and stable fashion while also allowing for continual refinement of the individual networks - as nature does for biological networks - will require new analysis tools. This paper takes a step in this direction by establishing contraction properties of broad classes of nonlinear recurrent networks and neural ODEs, and showing how these quantified properties allow in turn to recursively construct stable networks of networks in a systematic fashion. The results can also be used to stably combine recurrent networks and physical systems with quantified contraction properties. Similarly, they may be applied to modular computational models of cognition. We perform experiments with these combined networks on benchmark sequential tasks (e.g permuted sequential MNIST) to demonstrate their capacity for processing information across a long timescale in a provably stable manner.

1 INTRODUCTION

Neuro-inspired machine learning has profoundly altered many fields such as computer vision, natural language processing, and computational neuroscience (Bengio et al., 2017; Hassabis et al., 2017). While models trained with e.g. deep learning are remarkably powerful, they are for the most part 'black boxes'. This opaqueness can be dangerous in safety-critical applications, such as autonomous driving or human-centered robotics, and it limits conceptual progress. In the case of recurrent models, one difficulty is that providing a certificate of *stability* is currently impossible or computationally impractical. Given that stability is a fundamental property of dynamical systems – and is intimately linked to concepts of control, generalization, data-efficiency, and robustness – being able to guarantee the stability of a recurrent model is an important step towards making sure deep models behave as we expect them to.

In this spirit, there has been a recent flux of work focusing on applications of contraction analysis (Lohmiller & Slotine, 1998) to recurrent models. Loosely speaking, a dynamical system is said to be contracting if any two of its trajectories converge to each other exponentially, regardless of initial conditions. A primary advantage of contraction analysis is that it is directly applicable to non-autonomous systems, which the vast majority of recurrent models are, allowing in turn modular contraction-preserving combination properties to be derived (Slotine & Lohmiller, 2001; Slotine, 2003). We include a brief mathematical primer on contraction analysis in section 5.1.

To briefly review the current literature on application of contraction analysis to recurrent models, we first note that the 'Echo-State Condition' introduced in Jaeger (2001) is equivalent to discrete-time contraction in the identity metric. A later generalization of this condition included a diagonal metric (Buehner & Young, 2006). In the context of neuroscience, contraction analysis has been applied to analyzing the dynamics of winner-take-all networks (Rutishauser et al., 2011; 2015) as well as networks with synaptic plasticity (Kozachkov et al., 2020). In the machine learning context, Miller and Hardt recently rederived an 'echo-state property' for discrete recurrent models, and went on to prove that these contracting recurrent models could be well-approximated by feedforward networks in certain cases (Miller & Hardt, 2018). More recently still, in a series of papers Revay, Wang, and Manchester applied contraction analysis to discrete-time recurrent networks (Revay & Manchester, 2020; Revay et al., 2021; 2020a) and greatly expanded the class of models considered in Miller

& Hardt (2018). Within this line of research, our paper has two main aims: 1) To provide simple contraction conditions for *continuous-time* recurrent neural networks and 2) To show how these continuous-time contraction conditions imply a combination property. Using both aims, we proceed to implement stable combination networks that exhibit near state-of-the-art (SOTA) performance on multiple sequential image classification tasks with a small number of trainable parameters.

The combination and reuse of primitive "modules" has enabled a great deal of progress in computer science, and is also a key theme in biological evolution, particularly apparent in cortical structure of the human brain. In fact, it is thought that the majority of traits that have developed over the last 400+ million years are the result of evolutionary forces acting on regulatory elements that combine core components, rather than mutations to the core components themselves. This mechanism of action makes meaningful variation in population phenotypes much more feasible to achieve, and is appropriately titled "facilitated variation" (Gerhart & Kirschner, 2007). In addition to the biological evidence for facilitated variation, computational models have demonstrated that this approach produces populations that are better able to generalize to new environments (Parter et al., 2008), an ability that will be critical to further develop in deep learning systems.

While the benefits of building modular systems are clear (Simon, 1962), as e.g in DeepMind's AlphaGo (Silver et al., 2016), in general there is no guarantee that a combination of stable systems will itself be stable. Thus the tractability of these evolutionary processes hinges on some mechanism for ensuring stability of combinations. As contraction analysis tools allow complicated contracting systems to be built up recursively from simpler elements, this form of stability is well suited for neural and other biological systems (Slotine & Lohmiller, 2001; Slotine & Liu, 2012). Here, we describe two common forms of contracting system combinations–hierarchical and feedback–that automatically guarantee overall system stability, as depicted in Figure 1. Note also that contracting combinations can be made between systems with very different dynamics–so long as those dynamics are contracting. In our particular case, this means that the contracting RNNs discussed herein can be combined with physical systems with quantified contraction properties.

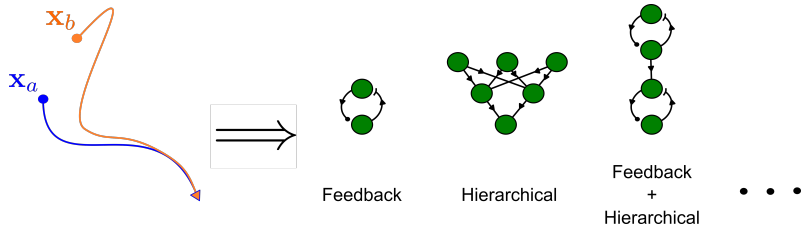

Figure 1: Our stability certificate implies a modularity principle. It may be used to recursively construct complicated 'networks of networks' while automatically maintaining stability.

Ultimately, stability is a basic presupposition of experimental neuroscience research. In particular, the act of having an animal perform many repeated trials of a task and then averaging the neural responses across tasks assumes an underlying stability or consistency to the observed neural dynamics. Moreover, cognitive models are moving increasingly toward study of multi-area dynamics, but many questions remain on how to best train and evaluate such networks (Yang et al., 2018; Yang & Molano-Mazon, 2021). Understanding how different brain regions interact harmoniously to produce a unified percept/action will require new ideas and analysis tools.

2 Mathematical Results

2.1 Contraction Conditions for Continuous-Time RNNs

We consider the following continuous-time RNN, evolving according to the following equations:

$$\begin{aligned} \tau\dot{\mathbf{x}} &= -\mathbf{x} + \mathbf{W}\phi(\mathbf{x}) + \mathbf{u}(t) \\ \mathbf{y} &= g(\mathbf{x}) \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$, ϕ is a static nonlinearity such that $0 \leq \phi' \leq g$, $\mathbf{u}(t)$ is some input (potentially time-varying) and $\tau > 0$. We do not constrain the sign of ϕ. Example nonlinearities $\phi(x)$ that

satisfy the constraints are $\tanh(ax)$, $\log(1+e^x)$, and $\max(0,x)$, with $g = a, 1, 1$ respectively. Here $\mathbf{y}$ is a readout vector, and $g(\cdot)$ is a static nonlinearity with bounded derivative. Note that this class of RNNs is equivalent to another commonly used class of RNNs where the terms $\mathbf{W}\mathbf{x} + \mathbf{u}$ appears inside $\phi(\cdot)$. See Appendix 5.3 or (Miller & Fumarola, 2012) for details. Our mathematical results therefore apply equally well to both types of RNNs.

We seek a stability certificate for this system in terms of the recurrent weight matrix $\mathbf{W}$. We are specifically interested in restricting $\mathbf{W}$ such that the RNN is globally *contracting*. We do this with the goal of recursively combining these contracting RNNs with other contracting RNNs to make large, complicated, stable 'networks of networks'. Here we derive five restrictions on $\mathbf{W}$ which ensure contraction for the continuous-time RNN defined by equation 1. To the best of our knowledge these contraction conditions are novel. All proofs can be found in the supplemental Section 6.

Theorem 1 (Absolute Value Restricted Weights)**.** *Let $|\mathbf{W}|$ denote the matrix formed by taking the element-wise absolute value of $\mathbf{W}$. If there exists a positive, diagonal $\mathbf{P}$ such that:*

$$\mathbf{P}(g|\mathbf{W}| - \mathbf{I}) + (g|\mathbf{W}| - \mathbf{I})^T\mathbf{P} \prec 0$$

then equation 1 is contracting in metric $\mathbf{P}$. Moreover, if $W_{ii} \leq 0$, then $|W|_{ii}$ may be set to zero to reduce conservatism.

Note that when $g = 1$, Theorem 1 can be checked by checking for linear stability of $|\mathbf{W}| - \mathbf{I}$.

Theorem 2 (Symmetric Weights)**.** *If $\mathbf{W} = \mathbf{W}^T$ and $g\mathbf{W} \prec \mathbf{I}$, then (1) is contracting.*

Theorem 3 (Product of Diagonal and Orthogonal Weights)**.** *If there exists positive diagonal matrices $\mathbf{P}_1$ and $\mathbf{P}_2$, as well as $\mathbf{Q} = \mathbf{Q}^T \succ 0$ such that*

$$\mathbf{W} = -\mathbf{P}_1\mathbf{Q}\mathbf{P}_2$$

then (1) is contracting in metric $\mathbf{M} = (\mathbf{P}_1\mathbf{Q}\mathbf{P}_1)^{-1}$.

Theorem 4 (Triangular Weights)**.** *If $g\mathbf{W} - \mathbf{I}$ is triangular and Hurwitz, then (1) is contracting in a diagonal metric.*

Theorem 5 (Singular Value Restricted Weights)**.** *If there exists a positive diagonal matrix $\mathbf{P}$ such that:*

$$g^2\mathbf{W}^T\mathbf{P}\mathbf{W} - \mathbf{P} \prec 0$$

then (1) is contracting in metric $\mathbf{P}$.

2.2 On Stability Theorems for RNNs

Several recent papers in machine learning, e.g (Haber & Ruthotto, 2017; Chang et al., 2019), claim that a sufficient condition for stability of the nonlinear system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$$

is that the associated Jacobian matrix $\mathbf{J}(\mathbf{x}, t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ has eigenvalues whose real parts are strictly negative, i.e:

$$\max_i \mathrm{Re}(\lambda_i(\mathbf{J}(\mathbf{x}, t)) \leq -\alpha$$

with $\alpha > 0$. This claim is generally false. For a counter-example, see section 4.4.2 in (Slotine & Li, 1991). However, in the *specific* case of the RNN (1), it appears that the eigenvalues of the symmetric part of $\mathbf{W}$ do provide information on global stability in a number of applications. For example, in (Matsuoka, 1992) it was shown that if $\mathbf{W}_s = \frac{1}{2}(\mathbf{W} + \mathbf{W}^T)$ has all its eigenvalues less than unity, and $\mathbf{u}$ is constant, then (1) has a unique fixed point that is globally asymptotically stable. It is easy to see that this condition also implies that the real parts of the eigenvalues of the Jacobian are uniformly negative. Moreover, in (Chang et al., 2019) it was shown that setting the symmetric part of $\mathbf{W}_s = \frac{1}{2}(\mathbf{W} + \mathbf{W}^T)$ almost equal to zero (yet slightly negative) led to rotational, yet stable dynamics in practice. This leads us to the following theorem, which shows that if the slopes of the activation functions change sufficiently slowly as a function of time, then the condition in (Matsuoka, 1992) in fact implies global contraction of (1).

Theorem 6. *Let* $\mathbf{D}$ *be a positive, diagonal matrix with* $D_{ii} = \frac{d\phi_i}{dx_i}$*, and let* $\mathbf{P}$ *be an arbitrary, positive diagonal matrix. If:*

$$(g\mathbf{W} - \mathbf{I})\mathbf{P} + \mathbf{P}(g\mathbf{W}^T - \mathbf{I}) \preceq -c\mathbf{P}$$

and

$$\dot{\mathbf{D}} - cg^{-1}\mathbf{D} \preceq -\beta\mathbf{D}$$

for $c, \beta > 0$*, then (1) is contracting in metric* $\mathbf{D}$ *with rate* β*.*

It has been conjectured that diagonal stability of $g\mathbf{W} - \mathbf{I}$ is a sufficient condition for global contraction of the RNN 1 (Revay et al., 2020b), however this has been difficult to prove. To better characterize this conjecture, we present Theorem 7, which shows by way of counterexample that diagonal stability of $g\mathbf{W} - \mathbf{I}$ does not imply global contraction in a *constant* metric for (1).

Theorem 7. $g\mathbf{W}_{sym} - \mathbf{I} = g\frac{\mathbf{W}+\mathbf{W}^T}{2} - \mathbf{I} \prec 0$*, i.e. contraction of the linear system in the identity metric, is not a sufficient condition for the general nonlinear system equation 1 to be contracting in a constant metric. High levels of antisymmetry in* $\mathbf{W}$ *can make it impossible to find such a metric, which we demonstrate via a* 2×2 *counterexample of the following form, with* $c \geq 2$*:* $\mathbf{W} = \begin{bmatrix} 0 & -c \\ c & 0 \end{bmatrix}$

3 Experiments

It is possible that imposing a strong stability constraint on the RNNs precludes them from performing well on tasks that require information processing over long timescales. To investigate whether or not this occurs with combination networks meeting our stability conditions, we trained a variety of provably stable RNNs on the benchmark sequential MNIST, permuted sequential MNIST (permuted seqMNIST), and sequential CIFAR10 tasks (Le et al., 2015; Chang et al., 2019). Images are presented pixel-by-pixel in these tests, and in the case of permuted seqMNIST, they are presented in a fixed but random order. At the end of the sequence, the network must predict the image label. We chose these tasks because they require the use of data points from across a long sequence, and because they are well-studied in the literature.

As we will detail below, our networks either performed comparably to or better than other stability constrained RNNs across all tasks, particularly when leveraging sparsity in the component RNNs. Our best achieved test accuracies of 96.94% on permuted seqMNIST and 64.75% on sequential CIFAR10 not only provide a new SOTA for stability-guaranteed RNNs, but are also competitive with the overall SOTA networks despite having many fewer parameters (Table 1).

3.1 Network Architecture and Initialization

For our experiments we focused on networks whose nonlinear (ReLU) subnetworks satisfied either Theorem 1 or Theorem 5. These subnetworks were then combined in negative feedback, in a way that automatically ensured contraction of the overall network (see Appendix 5.2 for details).

For subnetworks constrained by Theorem 5 we ensured contraction by directly parameterizing each of the $\mathbf{W}_i$ $(i = 1, 2 \dots p)$ as:

$$\mathbf{W}_i = \mathbf{\Phi}_i^{-1}\mathbf{U}_i\mathbf{\Sigma}_i\mathbf{V}_i^T\mathbf{\Phi}_i \tag{2}$$

where $\mathbf{\Phi}_i$ is diagonal and nonsingular, $\mathbf{U}_i$ and $\mathbf{V}_i$ are orthogonal, and $\mathbf{\Sigma}_i$ is diagonal with $\Sigma_{ii} \in [0, g^{-1})$. Two additional parameterizations are needed to ensure the orthogonality of $\mathbf{U}_i$ and $\mathbf{V}_i$. We achieve this by writing $\mathbf{U}_i$ and $\mathbf{V}_i$ as matrix exponentials of skew-symmetric matrices:

$$\mathbf{U}_i = e^{\mathbf{Q}_i - \mathbf{Q}_i^T}, \quad V = e^{\mathbf{R}_i - \mathbf{R}_i^T}$$

where $\mathbf{Q}_i$ and $\mathbf{R}_i$ are arbitrary matrices. We used $\phi(x) = \text{ReLU}(x)$, $g(\mathbf{x}) = \sigma\mathbf{B}\mathbf{x} + \mathbf{d}$ and initialized $\mathbf{W}_i$ as a scaled identity matrix (Le et al., 2015).

For subnetworks constrained by Theorem 1 on the other hand, we were not able to find a direct parameterization to use during training - but it is straightforward to randomly generate matrices with a particular likelihood of meeting the condition by selecting an appropriate density level and

limit on entry magnitude. As every RNN meeting this contraction condition has a well-defined stable LTI system contracting in the same metric, it is not only easy to verify, but also easy to find a contraction metric to use in our training algorithm. Solving for $\mathbf{M}$ in $-\mathbf{I} = \mathbf{M}\mathbf{A} + \mathbf{A}^T\mathbf{M}$ will produce a valid metric for any stable LTI system $\mathbf{A}$ (Slotine & Li, 1991). We therefore randomly generate fixed subnetworks satsifying Theorem 1 and train only the linear connections between them (Figure 2), as well as the linear input and output layers. Taking this approach can greatly speed up training as opposed to the singular value condition parameterization, and confers the benefits of a sparse network - which has implications in both machine learning and biology.

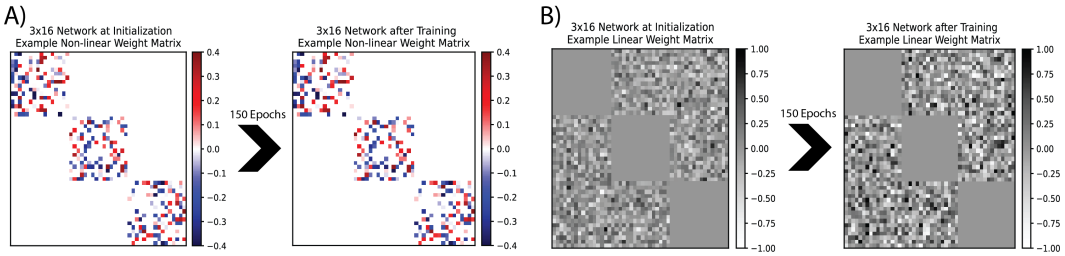

Figure 2: Network construction and training example (3x16 combination RNN). Nonlinear connections (A) are formed only within the component RNNs, are initialized based on a set sparsity, and do not changed over training. Linear connections (B) are formed only between the component RNNs, are constrained to be antisymmetric (with respect to the overall network metric), and are updated in the training process.

3.2 Permuted Sequential MNIST

3.2.1 Sparse Combo Networks

Unless specified otherwise, the sparse combination networks were trained for 150 epochs, using an Adam optimizer with initial learning rate 1e-3 and weight decay 1e-5. The learning rate was cut to 1e-4 after 90 epochs and to 1e-5 after 140. Density and scalar settings for the component nonlinear RNNs were chosen for each network size using the percentage of random networks that met the Theorem 1 condition. For scalar s, a component network would have non-zero entries sampled uniformly between $-s$ and s. Detailed information on network initialization and hyperparameter tuning is provided in Appendix 5.4.

First, we held static the number of units and initialization settings for each component RNN, and tested the effect of changing the number of components in the combination network. 1, 3, 5, 10, 15, 20, 22, 25, and 30 component RNNs were tested in this experiment (Table S2). Increasing the number of components initially lead to great improvements in test accuracy, but had diminishing returns - test accuracy consistently hit $\sim 93\%$ with a large enough number of modules, but neither loss nor accuracy showed meaningful improvement past the $22x16$ network (Figure 3A). Interestingly, early training loss and accuracy became substantially worse once the number of components increased past a certain point, falling from 70% to 43% epoch 1 test accuracy between the $22x16$ and $30x16$ networks.

To better understand how the modularity of the combination networks affects performance, the next experiment held the number of total units constant at 352, selected due to the prior success of the $22x16$ network, and tested different allocations of these units amongst component RNNs. Thus $1x352$, $11x32$, $44x8$, and $88x4$ networks were trained to compare against the $22x16$ (Table S3). Increasing the modularity improved performance to a point, with the $44x8$ network resulting in final test accuracy of 94.44%, while conversely the $11x32$ resulted in decreased test accuracy (Figure 3B). However, the $88x4$ network was unable to learn, and a $352x1$ network would theoretically just be a scaled linear anti-symmetric network.

While the $11x32$ network performed notably worse than the $22x16$ did, using larger component RNNs provides more room to test different sparsity levels. Therefore, our next step was to compare initialization settings within the modular networks (Table S4). Decreasing the density was able to consistently improve performance, with the strongest effect seen in the $11x32$ network, as predicted

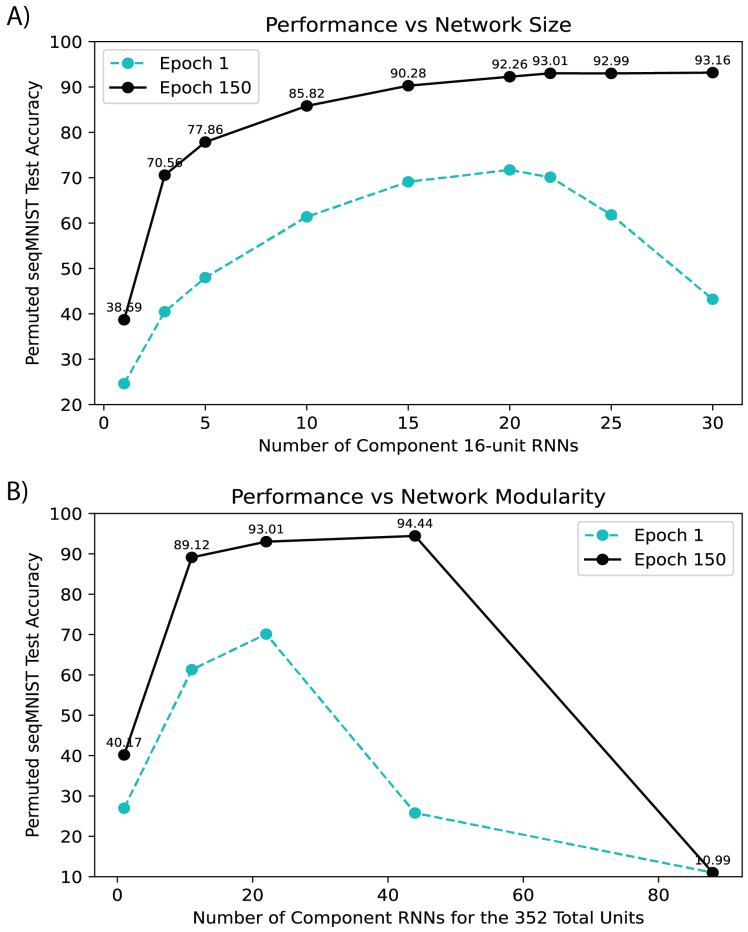

Figure 3: Permuted seqMNIST performance by combination network size. For total size comparisons, first epoch and final test accuracy are plotted against the number of 16 unit module RNNs used (A). For modularity comparisons, first epoch and final test accuracy are plotted against the number of module RNNs that 352 total units were distributed amongst (B).

(Figure 4A). To further optimize test accuracy, we moved onto a $16x32$ network, as it reliably outperformed the $11x32$ (Table S5).

Ultimately, both sparsity and magnitude had a clear effect on performance (Figure 4B/C). Increases in both had a positive correlation with accuracy and loss through most parameters tested, seemingly driven by the increase in non-zero element magnitude enabled by sparsity. Best test accuracy overall was 96.79%, which was obtained by both a $16x32$ network with 5% density and entries between -5 and 5, and a $16x32$ network with 3.3% density and entries between -6 and 6. The latter also achieved the best epoch 1 test accuracy observed of 86.79%. For a component RNN size of just 32 units, these density levels are quite small, with only one or two directional connections per neuron on average (Figure S2).

To further improve performance once network settings were explored, an extended training run was tested on the best performing option. Settings were kept the same as above using a 3.3% density $16x32$ network, except training now ran for over 200 epochs, with just a single learning rate cut occurring after epoch 200 (exact number of epochs varied based on runtime limit). This experiment was repeated four times and resulted in 96.94% best test accuracy.

As the nonlinear RNNs do not have their internal connections trained, network performance could be particularly susceptible to random initialization. However, across the four trials with the same network settings, best test accuracy always fell between 96.65% and 96.94%, a range much smaller than the differences seen with changing sparsity settings and network size. Three of the four trials

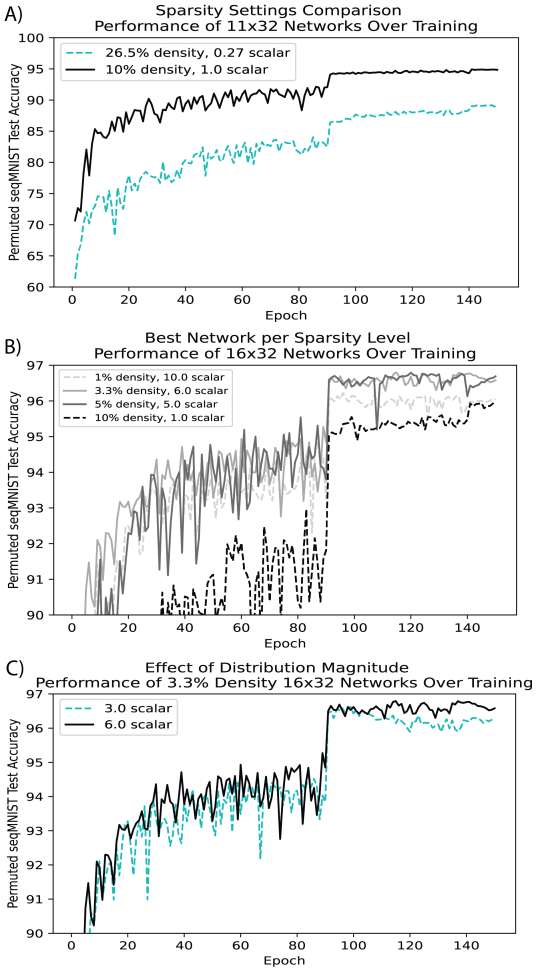

Figure 4: Permuted seqMNIST performance by component RNN initialization settings. Test accuracy is plotted over the course of training for two 11x32 networks with different density levels and entry magnitudes (A), and the same is repeated for four 16x32 networks (B), highlighting the role of sparsity in network performance. Test accuracy is then plotted over the course of training for two 3.3% density 16x32 networks with different entry magnitudes (C), to demonstrate the role of the scalar. When the magnitude becomes too high however, performance is out of view of the current axis limits.

showed best test accuracy $\geq 96.88\%$, despite some variability in early training performance (Figure S1).

It is worth noting that upon investigation of the nonlinear weight matrices across these trials, the high performing/low density networks had substantially lower max eigenvalue of $|\mathbf{W}|$, suggesting that stronger stability can actually correlate with improved performance on sequential tasks. This could be due to a mechanism such as that described in (Radhakrishnana et al., 2020).

3.3 Singular Value Constrained Networks

For these networks numerical integration of 1 was performed by using standard Euler integration, with step size dt and $\frac{dt}{\tau} = .03$. We trained the networks using the Adam optimizer with an initial learning rate of 0.003 and a learning rate scheduler which reduces the learning rate by a factor of 0.1 when the validation accuracy plateaued for 10 epochs. We performed two sets of experiments to investiage the effects of modularity and network size. In the first set of experiments, we held the number of neurons per module fixed at 16, and retrained the networks with different number

of modules $p = [1, 5, 18, 30]$. The results are shown in Figure 5a) and 5b). These curves show an inverted 'U'-curve as a function of the number of subnetworks. Initially increasing p helps improve accuracy, but beyond a certain point the network becomes hard to train and performance plummets. In the second set of experiments, we held the total number of neurons fixed at $N = 352$ and then divided the network into $p = [1, 4, 8, 22]$ subnetworks The results are shown in Figure 5c) and 5d). We find that a small amount of modularity ($p = 4$) helps improve performance but adding more modules does not improve performance.

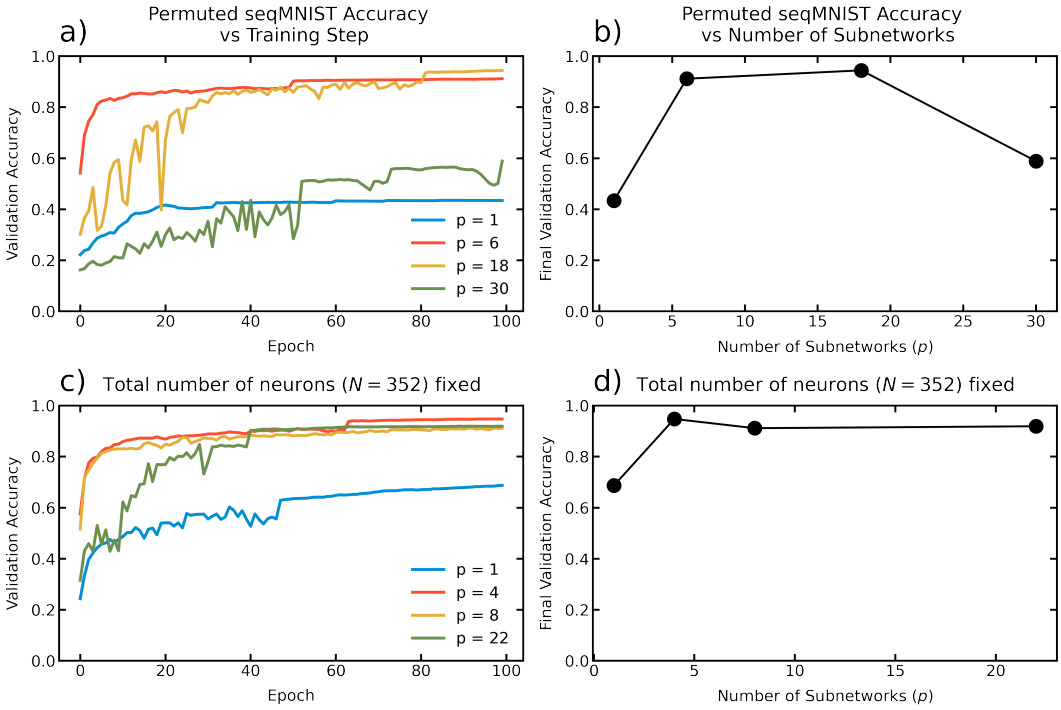

Figure 5: a) Permuted seqMNIST validation accuracy curves for svd-constrained networks (see Theorem 5) with p subnetworks and 16 neurons per subnetwork. The connections between subnetworks ensured that the overall network was contracting at any given moment of the training process. b) Final validation accuracies plotted against number of subnetworks. c) Accuracy curves for networks with 352 total neurons, split into p subnetworks. d) Final validation accuracies for the networks described in c).

3.4 Sequential CIFAR10

To test the combination networks on a more challenging sequential image classification task, we next trained select networks on sequential CIFAR10. For this section we just used the sparse combo networks, as these consistently trained faster and better than the singular-value constrained networks. The initial test used the same training hyperparameters as described for permuted sequential MNIST, with one of the best performing network architectures: a $16x32$ sparse combination network with 3.3% density and entries between -6 and 6. On this very first trial, we achieved 64.63% best test accuracy, the highest to date for a provably stable RNN, and higher than the 1 million parameter CKConv network, which holds the record for permuted sequential MNIST accuracy (Table 1). By simply extending the training time to 200 epochs (with learning rate cuts now at epochs 140 and 190), we further improved the best test accuracy to 64.75% (Figure S3). However, changes to learning rate initialization and Adam weight decay were unable to improve performance on the new task (Table S6).

To complete a final benchmarking table (Table 1), we again used the same settings as the first CIFAR10 trial to run a single trial on sequential MNIST (non-permuted). To our knowledge, our results on permuted seqMNIST and sequential CIFAR10 are the best to date for a provably stable network.

Name	Stable?	Parameters	Sequential MNIST	Permuted MNIST	Sequential CIFAR10
LSTM (Chang et al., 2019)		68K	97.3%	92.7%	59.7%
Transformer (Trinh et al., 2018)		0.5M	98.9%	97.9%	62.2%
Antisymmetric (Chang et al., 2019)	?	36K	98%	95.8%	58.7%
Sparse Combo Net	✓	130K	99.04%	**96.94%**	**64.75%**
Lipschitz (Erichson et al., 2021)	✓	134K	**99.4%**	96.3%	64.2%
Dense IndRNN (Li et al., 2020)		83K	**99.48%**	97.2%	-
CKConv (Romero et al., 2021)		1M	99.32%	**98.54%**	63.74%
Trellis (Bai et al., 2019)		8M	99.2%	98.13%	**73.42%**

Table 1: Published benchmarks for sequential MNIST, permuted MNIST, and sequential CIFAR10 test accuracy. Networks are grouped into three categories: baselines, best performing networks with claimed stability guarantee*, and networks achieving the current SOTA on each task. Within each grouping, networks are ordered by their number of trainable parameters (for CIFAR10 if this differed across tasks). Our network is highlighted. *For more information on stability guarantees in machine learning, see Section 2.2.

4 Discussion

Most work on stability of task-trained RNNs has focused on *single* RNNs. Here we leverage tools from nonlinear control theory to derive novel 'local' stability conditions which enable the recursive construction of stable *assemblies* of RNNs. We then show that these modular networks perform better than existing stable RNNs on key sequence classification tasks.

There are numerous future directions enabled by this work. For example, Theorem 6 suggests that a less restrictive contraction condition on $\mathbf{W}$ in terms of the eigenvalues of the symmetric part is possible and desirable. Furthermore, the beneficial impact of sparsity on training these stable models suggests a potential avenue for additional work – in particular adding a regularizing sparsity term during training. As RNN variants have produced SOTA performance on a variety of important machine learning tasks, such experimental work has many possible extensions. Moreover, because sparsity and modularity have important theoretical and empirical implications in neuroscience (Kozachkov et al., 2020; Slotine & Liu, 2012) we expect to be able to apply the present work to those topics as well.

Our conditions could also be used in theoretical applications distinct from those already outlined. For instance, questions about synchronization of networks could be addressed via a virtual systems perspective as in Wang & Slotine (2005). Related methods could even be applied to models that include weight updates, such as the RNN with Hebbian and anti-Hebbian learning rules discussed in Kozachkov et al. (2020). Input-dependent stability criteria for updating networks are of particular interest, to provide theoretical groundwork on topics such as curriculum learning and transfer learning, as well as to better understand development in the biological brain.

Ultimately, our work represents a step forward in understanding the stability properties of recurrent neural networks. Stability is a fundamental property of dynamical systems, and is inextricably linked to concepts such as generalization, control, predictability, and robustness. Therefore, as systems trained with deep learning become more modular, complex, and integrated into our lives, understanding the conditions under which these systems are stable will become increasingly important. Furthermore, it will be necessary to identify training techniques that can reliably generate provably stable RNNs with minimal performance loss. We provide a proof-of-concept of the use of combination networks to achieve this goal, and show that sparsity enables additional performance gains.

Acknowledgements This work benefited from stimulating discussions with Michael Happ and Quang-Cuong Pham.

REFERENCES

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Trellis networks for sequence modeling. *ICLR*, 2019.

Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press Massachusetts, USA:, 2017.

Michael Buehner and Peter Young. A tighter bound for the echo state property. *IEEE Transactions on Neural Networks*, 17(3):820–824, 2006.

Bo Chang, Minmin Chen, Eldad Haber, and Ed H Chi. Antisymmetricrnn: A dynamical system view on recurrent neural networks. *arXiv preprint arXiv:1902.09689*, 2019.

Yimian Dai, Stefan Oehmcke, Fabian Gieseke, Yiquan Wu, and Kobus Barnard. Attention as activation. *arXiv preprint arXiv:2007.07729v2*, 2020.

N. Benjamin Erichson, Omri Azencot, Alejandro Queiruga, Liam Hodgkinson, and Michael W. Mahoney. Lipschitz recurrent neural networks. *ICLR*, 2021.

John Gerhart and Marc Kirschner. The theory of facilitated variation. *Proceedings of the National Academy of Sciences*, 104(1):8582–8589, 2007. doi: 10.1073/pnas.0701035104.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.

Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. 2001.

Leo Kozachkov, Mikael Lundqvist, Jean-Jacques Slotine, and Earl K Miller. Achieving stable dynamics in neural circuits. *PLoS computational biology*, 16(8):e1007659, 2020.

Dmitry Krotov and John Hopfield. Large associative memory problem in neurobiology and machine learning. *arXiv preprint arXiv:2008.06996*, 2020.

Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.

Shuai Li, Chris Cook, and Yanbo Gao. Deep independently recurrent neural network (indrnn). *arXiv preprint arXiv:1910.06251v3*, 2020.

Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

Kiyotoshi Matsuoka. Stability conditions for nonlinear continuous neural networks with asymmetric connection weights. *Neural networks*, 5(3):495–500, 1992.

John Miller and Moritz Hardt. Stable recurrent models. *arXiv preprint arXiv:1805.10369*, 2018.

Kenneth D Miller and Francesco Fumarola. Mathematical equivalence of two common forms of firing rate models of neural networks. *Neural computation*, 24(1):25–31, 2012.

Kumpati S. Narendra and Robert Shorten. Hurwitz stability of metzler matrices. *IEEE Transactions On Automatic Control*, 55(6):1484–1487, 2010.

Merav Parter, Nadav Kashtan, and Uri Alon. Facilitated variation: How evolution learns from past environments to generalize to new environments. *PLOS Computational Biology*, 4(11), 2008.

Adityanarayanan Radhakrishnana, Mikhail Belkin, and Caroline Uhler. Overparameterized neural networks implement associative memory. *PNAS*, 117:27162–27170, 2020.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

Max Revay and Ian Manchester. Contracting implicit recurrent neural networks: Stable models with improved trainability. In *Learning for Dynamics and Control*, pp. 393–403. PMLR, 2020.

Max Revay, Ruigang Wang, and Ian R Manchester. A convex parameterization of robust recurrent neural networks. *IEEE Control Systems Letters*, 5(4):1363–1368, 2020a.

Max Revay, Ruigang Wang, and Ian R Manchester. Lipschitz bounded equilibrium networks. *arXiv e-prints*, pp. arXiv–2010, 2020b.

Max Revay, Ruigang Wang, and Ian R Manchester. Recurrent equilibrium networks: Unconstrained learning of stable and robust dynamical models. *arXiv preprint arXiv:2104.05942*, 2021.

David W. Romero, Anna Kuzina, Erik J. Bekkers, Jakub M. Tomczak, and Mark Hoogendoorn. Ckconv: Continuous kernel convolution for sequential data. *arXiv preprint arXiv:2102.02611v1*, 2021.

Ueli Rutishauser, Rodney J Douglas, and Jean-Jacques Slotine. Collective stability of networks of winner-take-all circuits. *Neural computation*, 23(3):735–773, 2011.

Ueli Rutishauser, Jean-Jacques Slotine, and Rodney Douglas. Computation in dynamically bounded asymmetric systems. *PLoS Comput Biol*, 11(1):e1004039, 2015.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Herbert A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 1962. ISSN 0003-049X.

Jean-Jacques Slotine. Modular stability tools for distributed computation and control. *Int. J. Adaptive Control and Signal Processing*, 17(6), 2003.

Jean-Jacques Slotine and Yang-Yu Liu. The missing link. *Nature Physics*, 8(7):512–513, 2012.

Jean-Jacques Slotine and Winfried Lohmiller. Modularity, evolution, and the binding problem: a view from stability theory. *Neural Networks*, 14(2):137–145, 2001.

Jean-Jacques E Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice-Hall, 1991.

Nicolas Tabareau and Jean-Jacques Slotine. Notes on contraction theory. *arXiv preprint nlin/0601011*, 2006.

Trieu H. Trinh, Andrew M. Dai, Minh-Thang Luong, and Quoc V. Le. Learning longer-term dependencies in rnns with auxiliary losses. *arXiv preprint arXiv:1803.00144v3*, 2018.

Wei Wang and Jean-Jacques Slotine. On partial contraction analysis for coupled nonlinear oscillators. *Biological Cybernetics*, 92:38–53, 2005.

Eric W. Weisstein. Positive definite matrix. URL https://mathworld.wolfram.com/PositiveDefiniteMatrix.html.

Guangyu R Yang and Manuel Molano-Mazon. Next-generation of recurrent neural network models for cognition, Apr 2021. URL psyarxiv.com/w34n2.

Guangyu Robert Yang, Igor Ganichev, Xiao-Jing Wang, Jonathan Shlens, and David Sussillo. A dataset and architecture for visual reasoning with a working memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 714–731, 2018.

5 APPENDIX

5.1 SUPPLEMENTARY MATH

It can be shown that the non-autonomous system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$$

is contracting if there exists a metric $\mathbf{M}(\mathbf{x}, t) = \mathbf{\Theta}(\mathbf{x}, t)^T \mathbf{\Theta}(\mathbf{x}, t) \succ 0$ such that uniformly

$$\dot{\mathbf{M}} + \mathbf{M}\mathbf{J} + \mathbf{J}^T\mathbf{M} \preceq -\beta\mathbf{M}$$

where $\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ and $\beta > 0$. For more details see the main reference (Lohmiller & Slotine, 1998). Similarly, a non-autonomous discrete-time system

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, t)$$

is contracting if

$$\mathbf{J}^T\mathbf{M}_{t+1}\mathbf{J} - \mathbf{M}_t \preceq -\beta\mathbf{M}_t$$

5.2 FEEDBACK AND HIERARCHICAL COMBINATIONS

Consider two systems, independently contracting in constant metrics $\mathbf{M}_1$ and $\mathbf{M}_2$, which are combined in feedback:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, t) + \mathbf{B}\mathbf{y} \\ \dot{\mathbf{y}} &= \mathbf{g}(\mathbf{y}, t) + \mathbf{G}\mathbf{x} \end{aligned} \qquad \text{(Feedback Combination)}$$

If the following relationship between $\mathbf{B}, \mathbf{G}, \mathbf{M}_1$, and $\mathbf{M}_2$ is satisfied:

$$\mathbf{B} = -\mathbf{M}_1^{-1}\mathbf{G}^T\mathbf{M}_2$$

then the combined system is contracting as well. This may be seen as a special case of the feedback combination derived in (Tabareau & Slotine, 2006). The situation is even simpler for hierarchical combinations. Consider again two systems, independently contracting in some metrics, which are combined in hierarchy:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, t) \\ \dot{\mathbf{y}} &= \mathbf{g}(\mathbf{y}, t) + \mathbf{h}(\mathbf{x}, t) \end{aligned} \qquad \text{(Hierarchical Combination)}$$

where $\mathbf{h}(\mathbf{x}, t)$ is a function with *bounded* Jacobian. Then this combined system is contracting in a diagonal metric, as shown in (Lohmiller & Slotine, 1998). By recursion, this extends to hierarchies of arbitrary depth.

5.3 TWO DIFFERENT RNNS

Note that in neuroscience, the variable $\mathbf{x}$ in equation (1) is typically thought of as a vector of neural membrane potentials. It was shown in (Miller & Fumarola, 2012) that the RNN (1) is equivalent via an affine transformation to another commonly used RNN model,

$$\tau\dot{\mathbf{y}} = -\mathbf{y} + \phi(\mathbf{W}\mathbf{y} + \mathbf{b}(t)) \tag{3}$$

where the variable $\mathbf{y}$ is interpreted as a vector of firing rates, rather than membrane potentials. The two models are related by the transformation $\mathbf{x} = \mathbf{W}\mathbf{y} + \mathbf{b}$, which yields

$$\tau\dot{\mathbf{x}} = \mathbf{W}(-\mathbf{y} + \phi(\mathbf{W}\mathbf{y} + \mathbf{b})) + \tau\dot{\mathbf{b}} = -\mathbf{x} + \mathbf{W}\phi(\mathbf{x}) + \mathbf{v}$$

where $\mathbf{v} \equiv \mathbf{b} + \tau\dot{\mathbf{b}}$. Thus $\mathbf{b}$ is a low-pass filtered version of $\mathbf{v}$ (or conversely, $\mathbf{v}$ may be viewed as a first order prediction of $\mathbf{b}$) and the contraction properties of the system are unaffected by the affine transformation. Note that the above equivalence holds even in the case where $\mathbf{W}$ is not invertible. In this case, the two models are proven to be equivalent, provided that $\mathbf{b}(0)$ and $\mathbf{y}(0)$ satisfy certain conditions–which are always possible to satisfy (Miller & Fumarola, 2012). Therefore, any contraction condition derived for the x (or y) system automatically implies contraction of the other system. We exploit this freedom freely throughout the paper.

5.4 Supplementary Experiment Details

5.4.1 Sparse Combo RNNs Methodology

All networks described in the main text were trained using a single GPU on Google Colab, and all had a runtime of less than 7 minutes per epoch. An exported Colab notebook with the code to replicate all experiments is provided in the supplementary attachment, along with 3 model checkpoints (pre-training, best test accuracy epoch, final epoch) and a training stats CSV for the best performing network from each of the benchmarked tasks.

To obtain the metric necessary for training the linear connections, scipy.integrate.quad was used with default settings to solve for $\mathbf{M}$ in the equation $-\mathbf{I} = \mathbf{M}\mathbf{W} + \mathbf{W}^T\mathbf{M}$, as described in the main text. This is done by integrating $e^{\mathbf{W}^T t}\mathbf{Q}e^{\mathbf{W}t}dt$ from 0 to ∞. For efficiency reasons, and due to the guaranteed existence of a diagonal metric in the case of Theorem 1, integration was only performed to solve for the diagonal elements of $\mathbf{M}$. Therefore a check was added prior to training to confirm that the initialized network indeed satisfied Theorem 1 with metric $\mathbf{M}$. However, it was never triggered by our initialization method.

Initial training hyperparameter tuning was done primarily with $10x16$ combination networks on the permuted seqMNIST task, starting with settings based on existing literature on this task, and verifying promising settings using a $15x16$ network. Initialization settings were held the same throughout, matching what was later done for the size comparison trials (described below). The results of all of the attempted trials are reported in Table **??**, found in the next section. Once hyperparameters were decided upon, the trials reported on in the main text began.

For the size comparison trials (Figure 3A), the nonlinear RNN weights were set by drawing uniformly from between -0.4 and 0.4 with 40% density using scipy.sparse.random, and then zeroing out the diagonal entries. These settings were chosen because they resulted in $\sim 1\%$ of $16x16$ weight matrices meeting the Theorem 1 condition. During initialization only the matrices meeting this condition were kept, finishing when the desired number of component RNNs had been set - producing a block diagonal $\mathbf{W}$ like pictured in Figure 2. This same initialization process was used throughout our experiments.

Because larger networks require different sparsity settings to meet the Theorem 1 condition, these were not held constant between trials in the modularity comparison experiment (Figure 3B), but rather selected in the same way between trials - looking for settings that keep density and scalar balanced and result in $\sim 1\%$ of the matrices meeting the condition. The scalar was applied after sampling non-zero entries from a uniform distribution between -1 and 1. The resulting settings were 7.5% density and 0.077 scalar for 352 unit component RNN, 26.5% density and 0.27 scalar for 32 unit component RNN, 60% density and 0.7 scalar for 8 unit component RNN, and 100% density and 1.0 scalar for 4 unit component RNN.

When we began experimenting with sparsity in the initialization (4, we split the previously described scalar setting into two different scalars - one applied before a random matrix was checked against the Theorem 1 condition, and one applied after a matrix was selected. Of course the latter must be ≤ 1 to guarantee stability is preserved. The scalar was separated out after we noticed that at 5% density, random $32x32$ weight matrices met the condition roughly 1% of the time whether the scalar was 10 or 100000 - $\sim 85\%$ of sampled matrices using scalar 10 would continue to meet the condition even if multiplied by a factor of 10000. Therefore we wanted a mechanism that could bias selection towards matrices that are stable due to their sparsity and not due to magnitude constraints, while still keeping the elements to a reasonable size for training purposes.

After training was completed, we inspected the state of all networks described in the main text, pulling both the nonlinear ($\mathbf{W}$) and linear ($\mathbf{L}$) weight matrices from both initialization time and the final model. For $\mathbf{W}$, we confirmed it did not change over training, and inspected the max real part of the eigenvalues of $|\mathbf{W}|$ in accordance with Theorem 1. The densest tested matrices tended to have $\lambda_{max}(|\mathbf{W}|) > 0.9$, while the sparsest ones tended to have $\lambda_{max}(|\mathbf{W}|) < 0.1$. For $\mathbf{L}$, we checked the maximum element and the maximum singular value before and after training. In general, both went up over the course of training, but by a modest amount.

To report on the number of trainable parameters, we used the following formula:

$\frac{n^2 - M*C^2}{2} + i*n + n*o + n + o$

Where n is the total number of units in the MxC combination network, o is the total number of output nodes for the task, and i is the total number of input nodes for the task. Thus for the $16x32$ networks highlighted here, we have 129034 trainable parameters for the MNIST tasks, and 130058 trainable parameters for sequential CIFAR10.

Note that the naive estimate for the number of trainable parameters would be $n^2 + i*n + n*o + n + o$, corresponding to the number of weights in $\mathbf{L}$, the number of weights in the feedforward linear input layer, the number of weights in the feedforward linear output layer, and the bias terms for the input and output layers, respectively. However, because of the combination property constraints on $\mathbf{L}$, only the lower triangular portion of a block off-diagonal matrix is actually trained, and $\mathbf{L}$ is then defined in terms of this matrix and the metric $\mathbf{M}$. Thus we subtract $M * C^2$ to remove the block diagonal portions corresponding to nonlinear RNN components, and then divide by 2 to obtain only the lower half.

5.4.2 Tables of Results by Trial

Table S1 shows all trials run on permuted sequential MNIST before beginning the more systematic experiments reported on in the main text, presented in chronological order. Notably, our networks did not require an extensive hyperparameter tuning process.

Tables S2 and S3 report additional details on the size and modularity experiments described in the main text (Figure 3).

Tables S4 and S5 report additional details on the sparsity experiments described in the main text (Figure 4), including results of all relevant trials, as some were left out of the main text for brevity.

Finally, Table S6 reports the results of all trials of different hyperparameters on the sequential CIFAR10 task, in chronological order.

Size	Epochs	Adam WD	Initial LR	LR Schedule	Final Test Acc.
10x16	150	5e-5	5e-3	0.1 after 91	84%
10x16	150	1e-5	1e-2	0.1 after 50,100	85%
15x16	150	2e-4	5e-3	0.1 after 50,100	84%
10x16	150	2e-4	1e-2	0.5 every 10	81%
10x16	200	2e-4	1e-2	0.5 after 10 then every 30	81%
10x16	171*	5e-5	1e-2	0.75 after 10,20,60,100 then every 15	84%
15x16	179*	1e-5	1e-3	0.1 after 100,150	90%

Table S1: Training hyperparameter tuning trials, presented in chronological order. * indicates that training was cut short by the 24 hour Colab runtime limit. LR Schedule describes the scalar the learning rate was multiplied by, and at what epochs. The best performing network is highlighted, and represents the training settings we used throughout most of the main text.

Size	Final Test Acc.	Epoch 1 Test Acc.	Final Train Loss
1x16	38.69%	24.61%	1.7005
3x16	70.56%	40.47%	0.9033
5x16	77.86%	47.99%	0.7104
10x16	85.82%	61.38%	0.4736
15x16	90.28%	69.09%	0.3156
20x16	92.26%	71.72%	0.2392
22x16	93.01%	70.11%	0.2073
25x16	92.99%	61.81%	0.2017
30x16	93.16%	43.21%	0.1991

Table S2: Results for combination networks containing different numbers of component 16-unit RNNs. Training hyperparameters and network initialization settings were kept the same across all trials.

Size	Final Test Acc.	Epoch 1 Test Acc.	Final Train Loss
1x352	40.17%	26.97%	1.662
11x32	89.12%	61.29%	0.3781
22x16	93.01%	70.11%	0.2073
44x8	94.44%	25.78%	0.1500
88x4	10.99%	10.99%	2E+35

Table S3: Results for different distributions of 352 total units across a combination network. This number was chosen based on prior 22x16 network performance. For each component RNN size tested, the same procedure was used to select appropriate density and scalar settings.

Size	Density	Scalar	Final Test Acc.	Epoch 1 Test Acc.	Final Train Loss
11x32	26.5%	0.27	89.12%	61.29%	0.3781
11x32	10%	1.0	94.86%	70.67%	0.1278
22x16	40%	0.4	93.01%	70.11%	0.2073
22x16	20%	1.0	95.27%	76.58%	0.0924
22x16	10%	1.0	94.26%	71.53%	0.1425
44x8	60%	0.7	94.44%	25.78%	0.1500
44x8	50%	1.0	95.05%	30.52%	0.1180

Table S4: Results for different initialization settings - varying sparsity and magnitude of the component RNNs for different network sizes.

Density	Pre-select Scalar	Post-select Scalar	Final Test Acc.	Epoch 1 Test Acc.	Final Train Loss
10%	1.0	1.0	95.87%	73.67%	0.074
5%	10.0	0.1	95.11%	73.10%	0.1311
5%	10.0	0.2	96.15%	82.50%	0.0051
5%	10.0	0.5	96.69%	75.76%	0.0001
5%	6.0	1.0	96.41%	21.55%	3.3E-5
5%	7.5	1.0	16.75%	11.39%	3068967
3.3%	30.0	0.1	96.24%	83.89%	0.0005
3.3%	30.0	0.2	96.54%	86.79%	4E-5
1%	10.0	1.0	96.04%	81.2%	0.0001

Table S5: Further optimizing the sparsity settings for high performance using a 16x32 network. The final scalar is the product of the pre-selection and post-selection scalars. Note that the 5% density and 7.5 scalar network was killed after 18 epochs due to exploding gradient.

Density	Pre-select Scalar	Post-select Scalar	Epochs	Adam WD	Initial LR	LR Schedule	Best Test Acc.
3.3%	30	0.2	150	1e-5	1e-3	0.1 after 50,100	64.63%
3.3%	30	0.2	34*	1e-5	5e-3	0.1 after 50,100	35.42%
5%	6	1	150	1e-5	1e-3	0.1 after 50,100	60.9%
5%	10	0.5	150	1e-5	1e-4	0.1 after 50,100	54.86%
3.3%	30	0.2	150	1e-5	5e-4	0.1 after 50,100	61.83%
3.3%	30	0.2	200	1e-6	2e-3	0.1 after 150,190	62.31%
3.3%	30	0.2	186*	1e-5	1e-3	0.1 after 150,190	64.75%
3.3%	30	0.2	132*	1e-6	1e-3	0.1 after 150,190	62.31%
5%	10	0.5	195*	1e-5	1e-3	0.1 after 150,190	64.68%

Table S6: Additional hyperparameter tuning for the CIFAR10 task, presented in chronological order. * indicates that training was cut short by the 24 hour Colab runtime limit, or in the case of high learning rate was killed intentionally due to exploding gradient. LR Schedule describes the scalar the learning rate was multiplied by, and at what epochs. The best performing network is highlighted.

5.4.3 Supplemental Figures

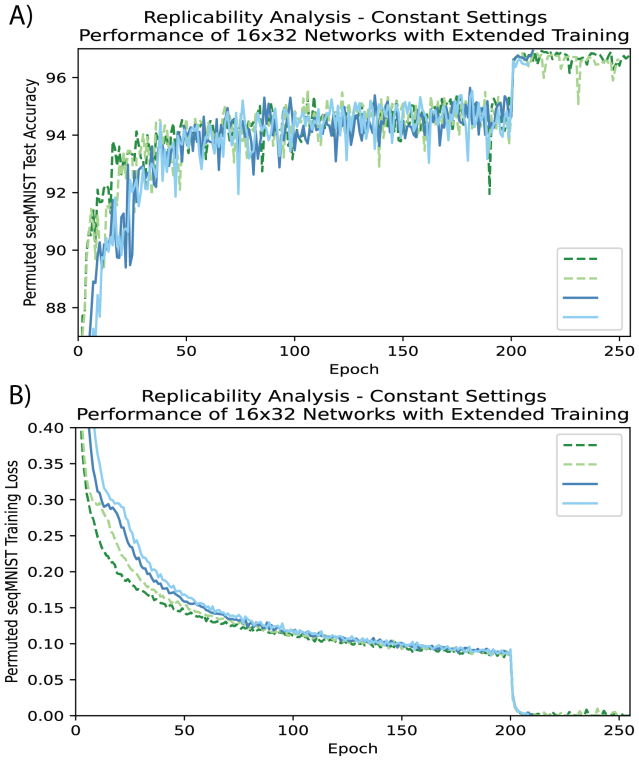

Figure S1: Permuted seqMNIST performance on repeated trials. Four different 16x32 networks with 3.3% density and entries between -6 and 6 were trained. (A) depicts test accuracy for each of the networks over the course of training. (B) depicts the training loss for the same networks.

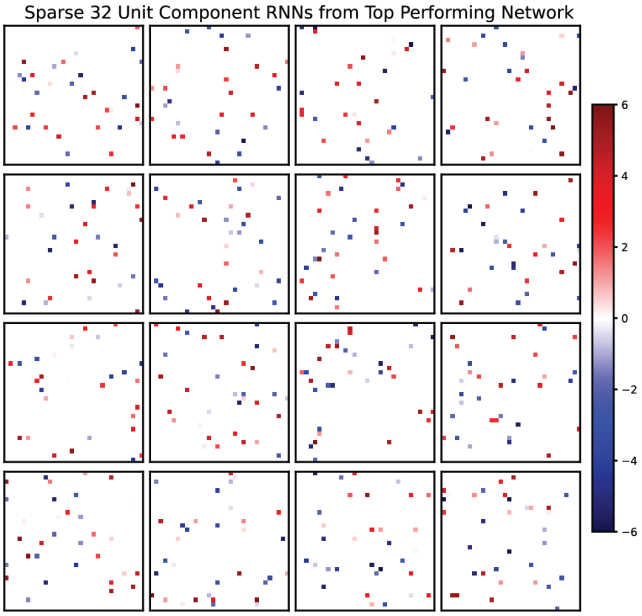

Figure S2: Weight matrices for each of the 32 unit non-linear component RNNs that were used in the best performing 16x32 network on permuted sequential MNIST.

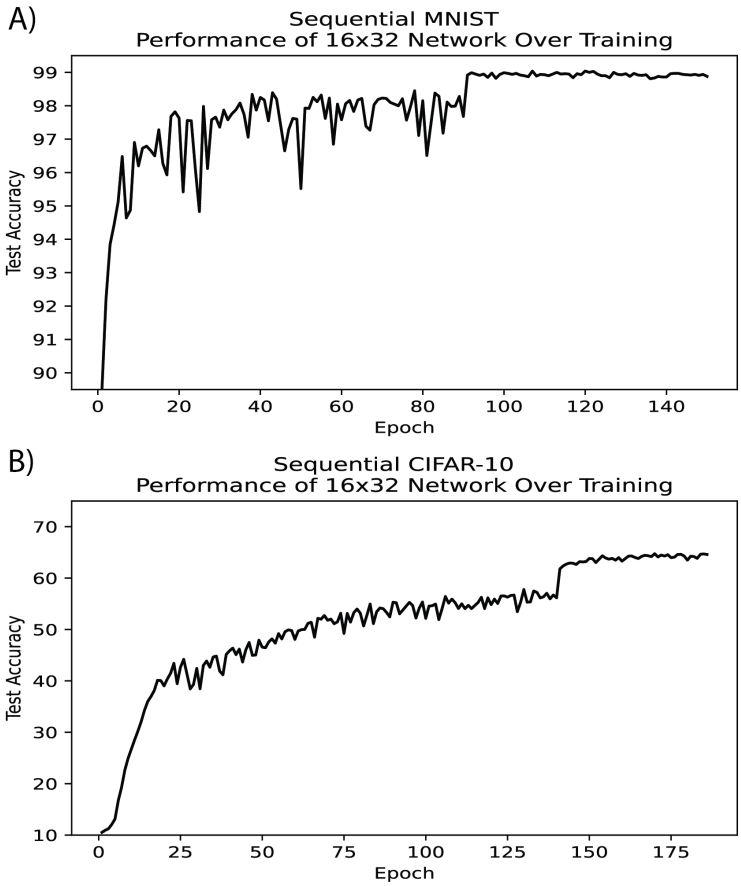

Figure S3: Test accuracy over the course of training for benchmark trials of the sparse combination network on sequential MNIST (A) and sequential CIFAR10 (B) tasks. 16x32 networks with 3.3% density and entries between -6 and 6 were again used.

6 PROOFS FOR MAIN RESULTS

6.1 PROOF OF THEOREM 1

Our first theorem is motivated by the observation that if the y-system is to be interpreted as a vector of firing rates, it must stay positive for all time. For a linear, time-invariant system with positive states, diagonal stability is equivalent to stability. Therefore a natural question is if diagonal stability of a linearized y-system implies anything about stability of the nonlinear system. More formally, given an excitatory neural network (i.e $\forall ij, W_{ij} \geq 0$), if the linear system

$$\dot{\mathbf{x}} = -\mathbf{x} + g\mathbf{W}\mathbf{x}$$

is stable, then there exists a positive diagonal matrix P such that:

$$\mathbf{P}(g\mathbf{W} - \mathbf{I}) + (g\mathbf{W} - \mathbf{I})^T\mathbf{P} \prec 0$$

The following theorem shows that the nonlinear system (1) is indeed contracting in metric $\mathbf{P}$, and extends this result to a more general $\mathbf{W}$ by considering only the magnitudes of the weights.

Theorem 1. *Let $|\mathbf{W}|$ denote the matrix formed by taking the element-wise absolute value of $\mathbf{W}$. If there exists a positive, diagonal $\mathbf{P}$ such that:*

$$\mathbf{P}(g|\mathbf{W}| - \mathbf{I}) + (g|\mathbf{W}| - \mathbf{I})^T\mathbf{P} \prec 0$$

then equation 1 is contracting in metric $\mathbf{P}$. Moreover, if $W_{ii} \leq 0$, then $|W|_{ii}$ may be set to zero to reduce conservatism.

This condition is particularly straightforward in the common special case where the network does not have any self weights, with the leak term driving stability. While it can be applied to a more general $\mathbf{W}$, the condition will of course not be met if the network was relying on highly negative values on the diagonal of $\mathbf{W}$ for linear stability. As demonstrated by counterexample in the proof of Theorem 1, it can be impossible to use the same metric $\mathbf{P}$ for the nonlinear RNN in such cases.

Theorem 1 allows many weight matrices with low magnitudes or a generally sparse structure to be verified as contracting in the nonlinear system equation 1, by simply checking a linear stability condition (as linear stability is equivalent to diagonal stability for Metzler matrices too (Narendra & Shorten, 2010)).

Beyond verifying contraction, Theorem 1 actually provides a metric, with little need for additional computation. Not only is it of inherent interest that the same metric can be shared across systems in this case, it is also of use in machine learning applications, where stability certificates are becoming increasingly necessary. Critically, it is feasible to enforce the condition during training via L2 regularization on $\mathbf{W}$. More generally, there are a variety of systems of interest that meet this condition but do not meet the well-known maximum singular value condition, including those with a hierarchical structure.

Proof. Consider the differential, quadratic Lyapunov function:

$$V = \delta\mathbf{x}^T\mathbf{P}\delta\mathbf{x}$$

where $\mathbf{P} \succ 0$ is diagonal. The time derivative of V is:

$$\dot{V} = 2\delta\mathbf{x}^T\mathbf{P}\dot{\delta\mathbf{x}} = 2\delta\mathbf{x}^T\mathbf{P}\mathbf{J}\delta\mathbf{x} = -2\delta\mathbf{x}^T\mathbf{P}\delta\mathbf{x} + 2\delta\mathbf{x}^T\mathbf{P}\mathbf{W}\mathbf{D}\delta\mathbf{x}$$

where $\mathbf{D}$ is a diagonal matrix such that $\mathbf{D}_{ii} = \frac{d\phi_i}{dx} \geq 0$. We can upper bound the quadratic form on the right as follows:

$$\delta\mathbf{x}^T\mathbf{P}\mathbf{W}\mathbf{D}\delta\mathbf{x} = \sum_{ij} P_i W_{ij} D_j \delta x_i \delta x_j \leq$$

$$\sum_i P_i W_{ii} D_i |\delta x_i|^2 + \sum_{ij, i\neq j} P_i |W_{ij}| D_j |\delta x_i||\delta x_j| \leq g|\delta\mathbf{x}|^T\mathbf{P}|\mathbf{W}||\delta\mathbf{x}|$$

If $W_{ii} \leq 0$, the term $P_i W_{ii} D_i |\delta x_i|^2$ contributes non-positively to the overall sum, and can therefore be set to zero without disrupting the inequality. Now using the fact that $\mathbf{P}$ is positive and diagonal, and therefore $\delta\mathbf{x}^T\mathbf{P}\delta\mathbf{x} = |\delta\mathbf{x}|^T\mathbf{P}|\delta\mathbf{x}|$, we can upper bound $\dot{V}$ as:

$$\dot{V} \leq |\delta\mathbf{x}|^T(-2\mathbf{P} + \mathbf{P}|\mathbf{W}| + |\mathbf{W}|\mathbf{P})|\delta\mathbf{x}| = |\delta\mathbf{x}|^T[(\mathbf{P}(|\mathbf{W}| - \mathbf{I}) + (|\mathbf{W}|^T - \mathbf{I})\mathbf{P})]|\delta\mathbf{x}|$$

where $|W|_{ij} = |W_{ij}|$, and $|W|_{ii} = 0$ if $W_{ii} \leq 0$ and $|W|_{ii} = |W_{ii}|$ if $W_{ii} > 0$. This completes the proof. Note that $\mathbf{W} - \mathbf{I}$ is Metzler, and therefore will be Hurwitz stable if and only if $\mathbf{P}$ exists (Narendra & Shorten, 2010).

It is also worth noting that highly negative diagonal values in $\mathbf{W}$ will prevent the same metric $\mathbf{P}$ from being used for the nonlinear system. Therefore the method used in this proof cannot feasibly be adapted to further relax the treatment of the diagonal part of $\mathbf{W}$. The intuitive reason behind this is that in the symmetric part of the Jacobian, $\frac{\mathbf{PWD}+\mathbf{DW}^T\mathbf{P}}{2} - \mathbf{P}$, the diagonal self weights will also be scaled down by small $\mathbf{D}$, while the leak portion $-\mathbf{P}$ remains untouched by $\mathbf{D}$. Now we actually demonstrate a counterexample, presenting a 2×2 symmetric Metzler matrix $\mathbf{W}$ that is contracting in the identity in the linear system, but cannot be contracting *in the identity* in the nonlinear system equation 1:

$$\mathbf{W} = \begin{bmatrix} -9 & 2.5 \\ 2.5 & 0 \end{bmatrix}$$

To see that it is not possible for the more general nonlinear system with these weights to be contracting in the identity, take $\mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$. Now

$$(\mathbf{WD})_{sym} - \mathbf{I} = \begin{bmatrix} -1 & 1.25 \\ 1.25 & -1 \end{bmatrix}$$

which has a positive eigenvalue of $\frac{1}{4}$.

□

6.2 Proof of Theorem 2

While regularization may push networks towards satisfying Theorem 1, strictly enforcing the condition during optimization is not straightforward. This motivates the rest of our theorems, which derive contraction results for specially structured weight matrices. Unlike Theorem 1, these results have direct parameterizations which can easily be plugged into modern optimization libraries.

Theorem 2. *If $\mathbf{W} = \mathbf{W}^T$ and $g\mathbf{W} \prec \mathbf{I}$, then (1) is contracting.*

When $\mathbf{W}$ is symmetric, (1) may be seen as a continuous-time Hopfield network. Continuous-time Hopfield networks with symmetric weights were recently shown to be closely related to Transformer architectures (Krotov & Hopfield, 2020; Ramsauer et al., 2020). Specifically, the dot-product attention rule may be seen as a discretization of the continuous-time Hopfield network with softmax activation function (Krotov & Hopfield, 2020). Our results here provide a simple sufficient (and nearly necessary, see above remark) condition for global exponential stability of a given *trajectory* for the Hopfield network. In the case where the input into the network is constant, this trajectory is a fixed point. Moreover, each trajectory associated with a unique input is guaranteed to be unique. Finally, we note that our results are flexible with respect to activation functions so long as they satisfy the slope-restriction condition. This flexibility may be useful when, for example, considering recent work showing that standard activation functions may be advantageously replaced by attention mechanisms (Dai et al., 2020).

Proof. We begin by writing $\mathbf{W} = \mathbf{R} - \mathbf{P}$ for some unknown $\mathbf{R} = \mathbf{R}^T$ and $\mathbf{P} = \mathbf{P}^T \succ 0$. The approach of this proof is to show by construction that the condition $g\mathbf{W} \prec \mathbf{I}$ implies the existence of an $\mathbf{R}$ and $\mathbf{P}$ such that the system is contracting in metric $\mathbf{P}$. We consider the y version of the RNN, which as discussed above is equivalent to the x version via an affine transformation.

Consider the contraction condition:

$$-2\mathbf{M} + \mathbf{MDW} + \mathbf{W}^T\mathbf{DM} \preceq -\beta\mathbf{M}$$

with $\beta > 0$. Substituting in the definitions of $\mathbf{W}$ and $\mathbf{M}$, this condition becomes:

$$-2\mathbf{P} + \mathbf{PD}(\mathbf{R} - \mathbf{P}) + (\mathbf{R} - \mathbf{P})\mathbf{DP} \preceq -\beta\mathbf{P}$$

Simplifying the terms and collecting them all on one side, the above may be written as:

$$(\beta - 2)\mathbf{P} + \mathbf{RDP} + \mathbf{PDR} - 2\mathbf{PDP} \preceq 0$$

via the Schur complement, the above term will be satisfied if:

$$(2 - \beta)\mathbf{P} - \mathbf{RDP}(2\mathbf{PDP})^{-1}\mathbf{PDR} =$$

$$(2 - \beta)\mathbf{P} - \frac{1}{2}(\mathbf{RDR}) \succeq (2 - \beta)\mathbf{P} - \frac{g}{2}(\mathbf{RR}) \succeq 0$$

We continue by setting $\mathbf{P} = \gamma^2\mathbf{RR}$ with $\gamma^2 = \frac{g}{2(2-\beta)}$, so that the above inequality is satisfied. At this point, we have shown that if $\mathbf{W}$ can be written as:

$$\mathbf{W} = \mathbf{R} - \gamma^2\mathbf{RR}$$

then (1) is contracting in metric $\mathbf{M} = \gamma^2\mathbf{RR}$. What remains to be shown is that if the condition:

$$g\mathbf{W} - \mathbf{I} \prec 0$$

Is satisfied, then this implies the existence of an $\mathbf{R}$ such that the above is true. To show that this is indeed the case, assume that:

$$\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{W} \succeq 0$$

Substituting in the definition of γ, this is just the statement that:

$$\frac{2(2 - \beta)}{4g}\mathbf{I} - \mathbf{W} \succeq 0$$

Setting $\beta = 2\lambda > 0$, this yields:

$$(1 - \lambda)\mathbf{I} \succeq g\mathbf{W}$$

Since $\mathbf{W}$ is orthogonal, we have the eigendecomposition:

$$\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{W} = \mathbf{V}(\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{\Lambda})\mathbf{V}^T$$

where $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{W}$. Denote the symmetric square-root of this expression as $\mathbf{S}$:

$$\mathbf{S} = \mathbf{V}\sqrt{(\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{\Lambda})}\mathbf{V}^T = \mathbf{S}^T$$

Which implies that:

$$\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{W} = \mathbf{S}^T\mathbf{S}$$

We now define $\mathbf{R}$ in terms of $\mathbf{S}$ as follows:

$$\mathbf{R} = \frac{1}{\gamma}\mathbf{S} + \frac{1}{2\gamma^2}\mathbf{I}$$

Which means that:

$$\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{W} = (\gamma\mathbf{R} - \frac{1}{2\gamma}\mathbf{I})(\gamma\mathbf{R} - \frac{1}{2\gamma}\mathbf{I})$$

Expanding out the right side, we get:

$$\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{W} = \gamma^2\mathbf{RR} - \mathbf{R} + \frac{1}{4\gamma^2}\mathbf{I}$$

Subtracting $\frac{1}{4\gamma^2}\mathbf{I}$ from both sides yields:

$$\mathbf{W} = \mathbf{R} - \gamma^2\mathbf{RR}$$

As desired.

□

6.3 PROOF OF THEOREM 3

Theorem 3. *If there exists positive diagonal matrices $\mathbf{P}_1$ and $\mathbf{P}_2$, as well as $\mathbf{Q} = \mathbf{Q}^T \succ 0$ such that*

$$\mathbf{W} = -\mathbf{P}_1 \mathbf{Q} \mathbf{P}_2$$

then (1) is contracting in metric $\mathbf{M} = (\mathbf{P}_1 \mathbf{Q} \mathbf{P}_1)^{-1}$.

Proof. Consider again a differential Lyapunov function:

$$V = \delta \mathbf{x}^T \mathbf{M} \delta \mathbf{x}$$

the time derivative is equal to:

$$\dot{V} = -2V + \delta \mathbf{x}^T \mathbf{M} \mathbf{W} \mathbf{D} \delta \mathbf{x}$$

Substituting in the definitions of $\mathbf{W}$ and $\mathbf{M}$, we get:

$$\dot{V} = -2V - \delta \mathbf{x}^T \mathbf{P}_1^{-1} \mathbf{P}_2 \mathbf{D} \delta \mathbf{x} \leq -2V$$

Therefore V converges exponentially to zero.

□

6.4 PROOF OF THEOREM 4

Theorem 4. *If $g\mathbf{W} - \mathbf{I}$ is triangular and Hurwitz, then (1) is contracting in a diagonal metric.*

Note that in the case of a triangular weight matrix, the system (1) may be seen as a feedforward (i.e hierarchical) network. Therefore, this result follows from the combination properties of contracting systems. However, our proof provides a means of explicitly finding a metric for this system.

Proof. Without loss of generality, assume that $\mathbf{W}$ is lower triangular. This implies that $W_{ij} = 0$ if $i \leq j$. Now consider the generalized Jacobian:

$$\mathbf{F} = -\mathbf{I} + \mathbf{\Gamma} \mathbf{W} \mathbf{D} \mathbf{\Gamma}^{-1}$$

with $\mathbf{\Gamma}$ diagonal and $\Gamma_i = \epsilon^i$ where $\epsilon > 0$. Because $\mathbf{\Gamma}$ is diagonal, the generalized Jacobian is equal to:

$$\mathbf{F} = -\mathbf{I} + \mathbf{\Gamma} \mathbf{W} \mathbf{\Gamma}^{-1} \mathbf{D}$$

Now note that:

$$(\mathbf{\Gamma} \mathbf{W} \mathbf{\Gamma}^{-1})_{ij} = \epsilon^i W_{ij} \epsilon^{-j} = W_{ij} \epsilon^{i-j}$$

Where $i \leq j$, we have $W_{ij} = 0$ by assumption. Therefore, the only nonzero entries are where $i \geq j$. This means that by making ϵ arbitrarily small, we can make $\mathbf{\Gamma} \mathbf{W} \mathbf{\Gamma}^{-1}$ approach a diagonal matrix with W_{ii} along the diagonal. Therefore, if:

$$\max_i g W_{ii} - 1 < 0$$

the nonlinear system is contracting. Since $\mathbf{W}$ is triangular, W_{ii} are the eigenvalues of $\mathbf{W}$, meaning that this condition is equivalent to $g\mathbf{W} - \mathbf{I}$ being Hurwitz.

□

6.5 PROOF OF THEOREM 5

Theorem 5. *If there exists a positive diagonal matrix $\mathbf{P}$ such that:*

$$g^2 \mathbf{W}^T \mathbf{P} \mathbf{W} - \mathbf{P} \prec 0$$

then (1) is contracting in metric $\mathbf{P}$.

Note that this is equivalent to the discrete-time diagonal stability condition developed in (Revay & Manchester, 2020), for a constant metric. Note also that when $\mathbf{M} = \mathbf{I}$, Theorem 5 is identical to checking the maximum singular value of $\mathbf{W}$, a previously established condition for stability of equation 1. However a much larger set of weight matrices are found via the condition when $\mathbf{M} = \mathbf{P}$ instead.

Proof. Consider the generalized Jacobian:

$$\mathbf{F} = \mathbf{P}^{1/2}\mathbf{J}\mathbf{P}^{-1/2} = -\mathbf{I} + \mathbf{P}^{1/2}\mathbf{W}\mathbf{P}^{-1/2}\mathbf{D}$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \frac{d\phi_i}{dx_i}$. Using the subadditivity of the matrix measure μ_2 of the generalized Jacobian we get:

$$\mu_2(\mathbf{F}) \leq -1 + \mu_2(\mathbf{P}^{1/2}\mathbf{W}\mathbf{P}^{-1/2}\mathbf{D})$$

Now using the fact that $\mu_2(\cdot) \leq ||\cdot||_2$ we have:

$$\mu_2(\mathbf{F}) \leq -1 + ||\mathbf{P}^{1/2}\mathbf{W}\mathbf{P}^{-1/2}\mathbf{D})||_2 \leq -1 + g||\mathbf{P}^{1/2}\mathbf{W}\mathbf{P}^{-1/2}||_2$$

Using the definition of the 2-norm, imposing the condition $\mu_2(\mathbf{F}) \leq 0$ may be written:

$$g^2\mathbf{W}^T\mathbf{P}\mathbf{W} - \mathbf{P} \prec 0$$

which completes the proof.

□

6.6 Proof of Theorem 6

Theorem 6. *Let $\mathbf{D}$ be a positive, diagonal matrix with $D_{ii} = \frac{d\phi_i}{dx_i}$, and let $\mathbf{P}$ be an arbitrary, positive diagonal matrix. If:*

$$(g\mathbf{W} - \mathbf{I})\mathbf{P} + \mathbf{P}(g\mathbf{W}^T - \mathbf{I}) \preceq -c\mathbf{P}$$

and

$$\dot{\mathbf{D}} - cg^{-1}\mathbf{D} \preceq -\beta\mathbf{D}$$

for $c, \beta > 0$, then (1) is contracting in metric $\mathbf{D}$ with rate β.

Proof. Consider the differential, quadratic Lyapunov function:

$$V = \delta\mathbf{x}^T\mathbf{P}\mathbf{D}\delta\mathbf{x}$$

where $\mathbf{D} \succ 0$ is as defined above. The time derivative of V is:

$$\dot{V} = \delta\mathbf{x}^T\mathbf{P}\dot{\mathbf{D}}\delta\mathbf{x} + \delta\mathbf{x}^T(-2\mathbf{P}\mathbf{D} + \mathbf{P}\mathbf{D}\mathbf{W}\mathbf{D} + \mathbf{D}\mathbf{W}^T\mathbf{D}\mathbf{P})\delta\mathbf{x}$$

The second term on the right can be factored as:

$$\begin{aligned}
\delta\mathbf{x}^T(-2\mathbf{P}\mathbf{D} + \mathbf{P}\mathbf{D}\mathbf{W}\mathbf{D} + \mathbf{D}\mathbf{W}^T\mathbf{D}\mathbf{P})\delta\mathbf{x} &= \\
\delta\mathbf{x}^T\mathbf{D}(-2\mathbf{P}\mathbf{D}^{-1} + \mathbf{P}\mathbf{W} + \mathbf{W}^T\mathbf{P})\mathbf{D}\delta\mathbf{x} &\leq \\
\delta\mathbf{x}^T\mathbf{D}(-2\mathbf{P}g^{-1} + \mathbf{P}\mathbf{W} + \mathbf{W}^T\mathbf{P})\mathbf{D}\delta\mathbf{x} &= \\
\delta\mathbf{x}^T\mathbf{D}[\mathbf{P}(\mathbf{W} - g^{-1}\mathbf{I}) + (\mathbf{W}^T - g^{-1}\mathbf{I})\mathbf{P}]\mathbf{D}\delta\mathbf{x} &\leq \\
-cg^{-1}\delta\mathbf{x}^T\mathbf{P}\mathbf{D}^2\delta\mathbf{x}
\end{aligned}$$

where the last inequality was obtained by substituting in the first assumption above. Combining this with the expression for $\dot{V}$, we have:

$$\dot{V} \leq \delta\mathbf{x}^T\mathbf{P}\dot{\mathbf{D}}\delta\mathbf{x} - cg^{-1}\delta\mathbf{x}^T\mathbf{P}\mathbf{D}^2\delta\mathbf{x}$$

Substituting in the second assumption, we have:

$$\dot{V} \leq \delta\mathbf{x}^T\mathbf{P}(\dot{\mathbf{D}} - cg^{-1}\mathbf{D}^2)\delta\mathbf{x} \leq -\beta\delta\mathbf{x}^T\mathbf{P}\mathbf{D}\delta\mathbf{x} = -\beta V$$

and thus V converges exponentially to 0 with rate β. □

6.7 Proof of Theorem 7

Theorem 7. *$g\mathbf{W}_{sym} - \mathbf{I} = g\frac{\mathbf{W}+\mathbf{W}^T}{2} - \mathbf{I} \prec 0$, i.e. contraction of the linear system in the identity metric, is not a sufficient condition for the general nonlinear system equation 1 to be contracting in a constant metric. High levels of antisymmetry in $\mathbf{W}$ can make it impossible to find such a metric, which we demonstrate via a 2×2 counterexample of the form*

$$\mathbf{W} = \begin{bmatrix} 0 & -c \\ c & 0 \end{bmatrix}$$

with $c \geq 2$.

The main intuition behind this counterexample is that high levels of antisymmetry can prevent a constant metric from being found in the nonlinear system. This is because $\mathbf{D}$ is a diagonal matrix with values between 0 and 1, so the primary functionality it can have in the symmetric part of the Jacobian is to downweight the outputs of certain neurons selectively. In the extreme case of all 0 or 1 values, we can think of this as selecting a subnetwork of the original network, and taking each of the remaining neurons to be single unit systems receiving input from the subnetwork. For a given static configuration of $\mathbf{D}$ (think linear gains), this is a hierarchical system that will be stable if the subnetwork is stable. But as $\mathbf{D}$ can evolve over time when a nonlinearity is introduced, we would need to find a constant metric that can serve completely distinct hierarchical structures simultaneously - which is not always possible.

Put in terms of matrix algebra, D can zero out columns of $\mathbf{W}$, but not their corresponding rows. So for a given weight pair w_{ij}, w_{ji}, which has entry in $\mathbf{W}_{sym} = \frac{w_{ij}+w_{ji}}{2}$, if $D_i = 0$ and $D_j = 1$, the i, j entry in $(\mathbf{WD})_{sym}$ will be guaranteed to have lower magnitude if the signs of w_{ij} and w_{ji} are the same, but guaranteed to have higher magnitude if the signs are different. Thus if the linear system would be stable based on magnitudes alone $\mathbf{D}$ poses no real threat, but if the linear system requires antisymmetry to be stable, $\mathbf{D}$ can make proving contraction quite complicated (if possible at all).

Proof. The nonlinear system is globally contracting in a *constant* metric if there exists a symmetric, positive definite $\mathbf{M}$ such that the symmetric part of the Jacobian for the system, $(\mathbf{MWD})_{sym} - \mathbf{M}$ is negative definite uniformly. Therefore $(\mathbf{MWD})_{sym} - \mathbf{M} \prec 0$ must hold for all possible $\mathbf{D}$ if $\mathbf{M}$ is a constant metric the system *globally* contracts in with any allowed activation function, as some combination of settings to obtain a particular $\mathbf{D}$ can always be found.

Thus to prove the main claim, we present here a simple 2-neuron system that is contracting in the identity metric with linear activation function, but can be shown to have no $\mathbf{M}$ that simultaneously satisfies the $(\mathbf{MWD})_{sym} - \mathbf{M} \prec 0$ condition for two different possible $\mathbf{D}$ matrices.

To begin, take

$$\mathbf{W} = \begin{bmatrix} 0 & -2 \\ 2 & 0 \end{bmatrix}$$

Note that any off-diagonal magnitude ≥ 2 would work, as this is the point at which $\frac{1}{2}$ of one of the weights (found in $\mathbf{W}_{sym}$ when the other is zeroed) will have magnitude too large for $(\mathbf{WD})_{sym} - \mathbf{I}$

to be stable.

Looking at the linear system, we can see it is contracting in the identity because

$$\mathbf{W}_{sym} - \mathbf{I} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \prec 0$$

Now consider $(\mathbf{MWD})_{sym} - \mathbf{M}$ with $\mathbf{D}$ taking two possible values of

$$\mathbf{D}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad and \quad \mathbf{D}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

We want to find some symmetric, positive definite $\mathbf{M} = \begin{bmatrix} a & m \\ m & b \end{bmatrix}$ such that $(\mathbf{MWD}_1)_{sym} - \mathbf{M}$ and $(\mathbf{MWD}_2)_{sym} - \mathbf{M}$ are both negative definite.

Working out the matrix multiplication, we get

$$(\mathbf{MWD}_1)_{sym} - \mathbf{M} = \begin{bmatrix} 2m - a & b - m \\ b - m & -b \end{bmatrix}$$

and

$$(\mathbf{MWD}_2)_{sym} - \mathbf{M} = \begin{bmatrix} -a & -(a+m) \\ -(a+m) & -2m - b \end{bmatrix}$$

We can now check necessary conditions for negative definiteness on these two matrices, as well as for positive definiteness on $\mathbf{M}$, to try to find an $\mathbf{M}$ that will satisfy all these conditions simultaneously. In this process we will reach a contradiction, showing that no such $\mathbf{M}$ can exist.

A necessary condition for positive definiteness in a real, symmetric $n \times n$ matrix $\mathbf{X}$ is $x_{ii} > 0$, and for negative definiteness $x_{ii} < 0$. Another well known necessary condition for definiteness of a real symmetric matrix is $|x_{ii} + x_{jj}| > |x_{ij} + x_{ji}| = 2|x_{ij}| \ \forall i \neq j$. See (Weisstein) for more info on these conditions.

Thus we will require a and b to be positive, and can identify the following conditions as necessary for our 3 matrices to all meet the requisite definiteness conditions:

$$2m < a \tag{4}$$

$$-2m < b \tag{5}$$

$$|2m - (a+b)| > 2|b - m| \tag{6}$$

$$|-2m - (a+b)| > 2|a + m| \tag{7}$$

Note that the necessary condition for $\mathbf{M}$ to be PD, $a + b > 2|m|$, is not listed, as it is automatically satisfied if equation 4 and equation 5 are.

It is easy to see that if $m = 0$, conditions equation 6 and equation 7 will result in the contradictory conditions $a > b$ and $b > a$ respectively, so we will require a metric with off-diagonal elements. To make the absolute values easier to deal with, we will check $m > 0$ and $m < 0$ cases independently.

First we take $m > 0$. By condition equation 4 we must have $a > 2m$, so between that and knowing the signs of all unknowns are positive, we can reduce many of the absolute values. Condition equation 6 becomes $a+b-2m > |2b-2m|$, and condition equation 7 becomes $a+b+2m > 2a+2m$, which is equivalent to $b > a$. If $b > a$ we must also have $b > m$, so condition equation 6 further

reduces to $a + b - 2m > 2b - 2m$, which is equivalent to $a > b$. Therefore we have again reached contradictory conditions.

A very similar approach can be applied when $m < 0$. Using condition equation 5 and the known signs we reduce condition equation 6 to $2|m| + a + b > 2b + 2|m|$, i.e. $a > b$. Meanwhile condition equation 7 works out to $a + b - 2|m| > 2a - 2|m|$, i.e. $b > a$.

Therefore it is impossible for a single constant $\mathbf{M}$ to accommodate both $\mathbf{D}_1$ and $\mathbf{D}_2$, so that no constant metric can exist for $\mathbf{W}$ to be contracting in when a nonlinearity is introduced that can possibly have derivative reaching both of these configurations. One real world example of such a nonlinearity is ReLU. Given a sufficiently high negative input to one of the units and a sufficiently high positive input to the other, $\mathbf{D}$ can reach one of these configurations. The targeted inputs could then flip at any time to reach the other configuration.

An additional condition we could impose on the activation function is to require it to be a strictly increasing function, so that the activation function derivative can never actually reach 0. We will now show that a very similar counterexample applies in this case, by taking

$$\mathbf{D}_{1*} = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} \quad and \quad \mathbf{D}_{2*} = \begin{bmatrix} \epsilon & 0 \\ 0 & 1 \end{bmatrix}$$

Note here that the $\mathbf{W}$ used above produced a $(\mathbf{WD})_{sym} - \mathbf{I}$ that just barely avoided being negative definite with the original $\mathbf{D}_1$ and $\mathbf{D}_2$, so we will have to increase the values on the off-diagonals a bit for this next example. In fact anything with magnitude larger than 2 will have some $\epsilon > 0$ that will cause a constant metric to be impossible, but for simplicity we will now take

$$\mathbf{W}_* = \begin{bmatrix} 0 & -4 \\ 4 & 0 \end{bmatrix}$$

Note that with $\mathbf{W}_*$, even just halving one of the off-diagonals while keeping the other intact will produce a $(\mathbf{WD})_{sym} - \mathbf{I}$ that is not negative definite. Anything less than halving however will keep the identity metric valid. Therefore, we expect that taking ϵ in $\mathbf{D}_{1*}$ and $\mathbf{D}_{2*}$ to be in the range $0.5 \geq \epsilon > 0$ will also cause issues when trying to obtain a constant metric.

We will now actually show via a similar proof to the above that $\mathbf{M}$ is impossible to find for $\mathbf{W}_*$ when $\epsilon \leq 0.5$. This result is compelling because it not only shows that ϵ does not need to be a particularly small value, but it also drives home the point about antisymmetry - the larger in magnitude the antisymmetric weights are, the larger the ϵ where we will begin to encounter problems.

Working out the matrix multiplication again, we now get

$$(\mathbf{MW}_*\mathbf{D}_{1*})_{sym} - \mathbf{M} = \begin{bmatrix} 4m - a & 2b - m - 2a\epsilon \\ b - m - 2a\epsilon & -4m\epsilon - b \end{bmatrix}$$

and

$$(\mathbf{MW}_*\mathbf{D}_{2*})_{sym} - \mathbf{M} = \begin{bmatrix} 4m\epsilon - a & -(2a + m - 2b\epsilon) \\ -(2a + m - 2b\epsilon) & -4m - b \end{bmatrix}$$

Resulting in two new main necessary conditions:

$$|4m - a - b - 4m\epsilon| > 2|2b - m - 2a\epsilon| \tag{8}$$

$$|4m\epsilon - a - b - 4m| > 2|2a + m - 2b\epsilon| \tag{9}$$

As well as new conditions on the diagonal elements:

$$4m - a < 0 \tag{10}$$

$$-4m - b < 0 \tag{11}$$

We will now proceed with trying to find a, b, m that can simultaneously meet all conditions, setting $\epsilon = 0.5$ for simplicity.

Looking at $m = 0$, we can see again that $\mathbf{M}$ will require off-diagonal elements, as condition equation 8 is now equivalent to the condition $a+b > |4b-2a|$ and condition equation 9 is similarly now equivalent to $a + b > |4a - 2b|$.

Evaluating these conditions in more detail, if we assume $4b > 2a$ and $4a > 2b$, we can remove the absolute value and the conditions work out to the contradicting $3a > 3b$ and $3b > 3a$ respectively. As an aside, if $\epsilon > 0.5$, this would no longer be the case, whereas with $\epsilon < 0.5$, the conditions would be pushed even further in opposite directions.

If we instead assume $2a > 4b$, this means $4a > 2b$, so the latter condition would still lead to $b > a$, contradicting the original assumption of $2a > 4b$. $2b > 4a$ causes a contradiction analogously. Trying $4b = 2a$ will lead to the other condition becoming $b > 2a$, once again a contradiction. Thus a diagonal $\mathbf{M}$ is impossible

So now we again break down the conditions into $m > 0$ and $m < 0$ cases, first looking at $m > 0$. Using condition equation 10 and knowing all unknowns have positive sign, condition equation 8 reduces to $a + b - 2m > |4b - 2(a + m)|$ and condition equation 9 reduces to $a + b + 2m > |4a - 2(b - m)|$. This looks remarkably similar to the $m = 0$ case, except now condition equation 8 has $-2m$ added to both sides (inside the absolute value), and condition equation 9 has $2m$ added to both sides in the same manner. If $4b > 2(a + m)$ the $-2m$ term on each side will simply cancel, and similarly if $4a > 2(b - m)$ the $+2m$ terms will cancel, leaving us with the same contradictory conditions as before.

Therefore we check $2(a + m) > 4b$. This rearranges to $2a > 2(2b - m) > 2(b - m)$, so that from condition equation 9 we get $b > a$. Subbing condition equation 10 in to $2(a + m) > 4b$ gives $8b < 4a + 4m < 5a$ i.e. $b < \frac{5}{8}a$, a contradiction. The analogous issue arises if trying $2(b - m) > 4a$. Trying $2(a + m) = 4b$ gives $m = 2b - a$, which in condition equation 9 results in $5b - a > |6a - 6b|$, while in condition equation 10 leads to $5a > 8b$, so equation 9 can further reduce to $5b - a > 6a - 6b$ i.e. $11b > 7a$. But $b > \frac{7}{11}a$ and $b < \frac{5}{8}a$ is a contradiction. Thus there is no way for $m > 0$ to work.

Finally, trying $m < 0$, we now use condition equation 11 and the signs of the unknowns to reduce condition equation 8 to $a+b+2|m| > |4b-2(a-|m|)|$ and condition equation 9 to $a+b-2|m| > |4a - 2(b + |m|)|$. These two conditions are clearly directly analogous to in the $m > 0$ case, where b now acts as a with condition equation 11 being $b > 4|m|$. Therefore the proof is complete.

□