Proceedings of Machine Learning Research vol 120:1–11, 2020

Contracting Implicit Recurrent Neural Networks: Stable Models with Improved Trainability

Max Revay M.REVAY@ACFR.USYD.EDU.AU
Ian Manchester IAN.MANCHESTER@SYDNEY.EDU.AU
Australian Center for Field Robotics,
Sydney Institute for Robotics and Intelligent Systems,
The University of Sydney,
NSW, 2006. Australia

Abstract

Stability of recurrent models is closely linked with trainability, generalizability and in some applications, safety. Methods that train stable recurrent neural networks, however, do so at a significant cost to expressibility. We propose an implicit model structure that allows for a convex parametrization of stable models using contraction analysis of non-linear systems. Using these stability conditions we propose a new approach to model initialization and then provide a number of empirical results comparing the performance of our proposed model set to previous stable RNNs and vanilla RNNs. By carefully controlling stability in the model, we observe a significant increase in the speed of training and model performance.

Keywords: System Identification, Contraction, Stability, Recurrent Neural Network, Vanishing Gradient, Exploding Gradient, Nonlinear Systems, Echo State Network

Notation

Most of our notation is standard. For a matrix A, $A \succ 0$ or $A \succeq 0$ means that A is positive definite or positive semi-definite. Similarly $A \prec 0$ or $A \preceq 0$ means that A is negative definite or negative semi-definite. We use $\text{vec}(A)$ to refer to the vector obtained by stacking A into a vector. $A \in \mathbb{D}_+$ means that A is a positive definite diagonal matrix. The normal distribution with mean μ and variance σ^2 is $\mathcal{N}[\mu, \sigma^2]$ and the uniform distribution between a and b is $\mathcal{U}[a, b]$.

1. Introduction

Recurrent neural networks (RNNs) are a common class of dynamical system used to model sequential data (Yi, 2004; Mandic and Chambers, 2001; Graves, 2012). They have been used extensively in areas such as system identification (Sjberg et al., 1995), learning based control systems (Anderson et al., 2007; Knight and Anderson, 2011), natural language processing (Zhou et al., 2016) and others. Instability of dynamical systems can lead to unpredictable behaviour, and as such, stability should be a consideration when training and deploying RNNs. Systems using RNNs have also been proposed in safety critical applications such as autonomous driving (Zyner et al., 2017), surgical robotics (Mayer et al., 2008) or active prosthetics (Boudali et al., 2019a). In such cases, dynamical instability could lead to injury or even death. As noted by a RAND corporation report on that state of AI: "The current state of AI verification, validation, test, and evaluation (VVT&E) is nowhere

close to ensuring the performance and safety of AI applications, particularly where safety-critical systems are concerned" (Tarraf et al., 2019). In practice, there are few approaches to training RNNs with stability guarantees.

The importance of model stability has more subtle implications than just safety; it is also closely related to the difficulty in fitting models. Training recurrent models using gradient descent is complicated by the exploding and vanishing gradients problem (Pascanu et al., 2013). If a model is too stable the gradients will vanish, and if it is unstable they will explode. A common approach to this problem, used for instance in the Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (GRU) (Cho et al., 2014), is to alter the model structure to one less susceptible to these issues. Alternatively, one can initialize or constrain the weights to be isometries that do not change the magnitude of the gradients. For instance, it has been reported that models such as the iRNN (Le et al., 2015), orthogonal RNN (Mhammedi et al., 2017) and spectral RNN (Zhang et al., 2018) can offer similar performance and trainability to the LSTM with fewer parameters. An alternative approach avoids this issue by initializing a sufficiently rich bank of dynamics so that only the input and output mappings must be learned. This is the basis for the Laguerre filter (Wahlberg, 1991) and the echo state network (Jaeger, 2003). These two approaches suggest that a more effective initialization scheme may be to initialize with a rich set of dynamics, of which some modes have long memory.

Another area closely linked to stability is model generalization. Empirically, it has been observed that stability is an effective regularizer in system identification (Umenberger and Manchester, 2019a,b). There are also theoretical results that relate generalization to a form of stability (Zhang et al., 2018).

1.1. Contraction Analysis

Stability can be defined in many ways. For input-output systems, Zames (1966) argued for two properties: Firstly, bounded inputs should produce bounded outputs. Secondly, outputs should not be critically sensitive to small changes in inputs.

Many approaches to RNN stability analysis focus on global stability of a particular equilibrium. This however, guarantees neither of these properties (Sontag, 2008). An additional complication is that the analysis is centred on a particular trajectory and in practice, we do not know the trajectories for unknown inputs. For instance, Kaszkurewicz and Bhaya (1993, 1994, 2000) propose diagonal stability. For a certain class of non-linearity, less conservative stability guarantees can be found using diagonally dominant Lyapunov functions (Chu and Glover, 1999). Absolute stability theory (Barabanov and Prokhorov, 2002) has also been used to reduce reduce conservatism.

Contraction (Lohmiller and Slotine, 1998) and incremental stability (Angeli, 2002), on the other hand, both guarantee stability in the sense of Zames (1966) and are independent of the input or equilibrium. We provide a brief introduction to contraction analysis. Additional details can be found in Lohmiller and Slotine (1998). Suppose that we have a non-linear dynamical system with dynamics:

$$x_{k+1} = f(x_k, k), \tag{1}$$

where x_k is the state of the system at time k. If f is piecewise differentiable, then we can study the differential dynamics given by:

$$\delta_{x_{k+1}} = F(x_k, k, \delta_{x_k}) \tag{2}$$

where $F(x, k, \delta_k)$ is the directional derivative of $f(\cdot)$ at x_k in the direction δ_k. If $f(\cdot)$ is differentiable then this can be written as $F(x, k, \delta_x) = \frac{\partial f(x,k)}{\partial x}\delta_x$. The vector δ_x can be interpreted as a infinitesimal displacement between two neighbouring trajectories. Stability of the differential dynamics (2) imposes a strong type of stability on the dynamical system (1), whereby, all trajectories of the original system (1) converge exponentially to a single trajectory. This is done by searching for a contraction metric or differential Lyapunov function (Forni and Sepulchre, 2014) $V(x_k, \delta_{x_k}) > 0,\ \forall\ \delta_x \neq 0$ such that $V(x_{k+1}, \delta_{x_{k+1}}) \leq \lambda V(x_k, \delta_{x_k})$ for $0 < \lambda < 1$. In this work we also allow for the case where $\lambda = 1$ to allow for non-expansive systems.

In principle there are many metric structures that can be used. A common approach however, parametrizes a quadratic form $V(\delta_{x_k}) = \delta_k^\top M \delta_k$, where $M \succ 0$ is a positive definite matrix. In this case a sufficient condition for contraction when $f(x, k)$ is differentiable is:

$$\frac{\partial f}{\partial x}^\top M \frac{\partial f}{\partial x} - \lambda M \preceq 0, \tag{3}$$

and M is called a contraction metric. Methods that analyse stability by bounding the maximum singular value (e.g Miller and Hardt (2018); Zhang et al. (2018)) can be seen as a special case of (3), where $M = I$ and $\lambda = 1$. The use of a parametrized metric provides considerable flexibility to the model set. This can be seen in the following example:

Example 1 *Consider the simple* 1 *layer RNN:*

$$h_1^{k+1} = \sigma(A h_2^k), \quad A = \begin{pmatrix} 0.8 & 1 \\ 0 & 0.8 \end{pmatrix} \tag{4}$$

Where $\sigma(z) = \max(0, z)$ *is a ReLU non-linearity with Lipschitz constant* $L_\sigma = 1$. *The matrix* A *has a maximum singular value of* 1.44 *so it does not satisfy the condition used by Miller and Hardt (2018). On the other hand, using condition* (3)*, we can construct a contraction metric* $V = \delta_{h_k}^\top P \delta_{h_k}$ *with* $P = diag(1, 10)$ *in which the system is contracting.*

1.2. Convex Parametrizations

Contraction analysis is a powerful tool for studying dynamical systems. Synthesis (e.g. control design or system identification) with contraction constraints is complicated by non-convexity in the model parameters and contraction metric. To be precise, (3) is convex in M or f, but not M and f. This significantly complicates optimization and even ensuring feasibility of the model becomes difficult. The situation is greatly improved if the constraints are convex (even if the objective is not) as projected gradient, barrier or penalty methods can be employed without making the optimization problem much harder than the unconstrained problem.

Methods such as Miller and Hardt (2018) essentially avoid this problem by fixing the metric at the cost of model expressibility. It has been found, however, that using an implicit model structure allows parametrizations jointly convex in the model parameters and contraction metric (Tobenkin et al., 2017). Our work extends this approach to the model class of RNNs.

1.3. Contributions

We propose a class of contracting implicit recurrent neural network that is jointly convex in the model parameters and stability certificate. The proposed set has less conservative stability conditions which leads to greater expressibility when compared to previous stable RNNs - particularly

in the multilayer case. Additionally, we propose an initialization procedure that ensures both a rich set of dynamics and that the model is not too stable to train which improves the model trainability. We then provide empirical results on a simulated model and a gait prediction task highlighting the benefits of our approach.

2. Model Set

We are interested in fitting state space models parametrized by $\theta \in \Theta \subseteq \mathbb{R}^p$, of the following form:

$$x_{k+1} = f_\theta(x_k, u_k) \tag{5}$$
$$y_k = g_\theta(x_k, u_k) \tag{6}$$

where the function $f_\theta(x, u)$ can be represented by an L-layer Neural Network with skip connections, and Θ refers to a convex set of parameters to be defined later. In this case, we write the dynamics in (5) as follows:

$$z^0 = x, \qquad z^{\ell+1} = \phi(A_\ell z^\ell + B_\ell u + b_\ell) \quad \text{for } \ell = 0, ..., L-1, \quad f_\theta(x, u) = z^L. \tag{7}$$

Here, the superscript to refers to the layer in the network, z^ℓ is the output of the ℓ'th hidden layer of the network and are not necessarily of the same size. The weight matrices and bias for the ℓ'th layer are denoted A_ℓ, B_ℓ and b_ℓ respectively.

We define the set of admissible activations $\phi(\cdot)$ to be the set of piecewise differentiable, scalar, non-linearities with slope restricted to $[-\gamma, \gamma]$. For simplicity we will assume $\gamma = 1$, however this can be relaxed. This includes any collection of standard activation functions.

Skip connections from the input to the ℓ'th hidden are contained within B_ℓ and skip connections between hidden layers are included by replacing part of the activation with a linear activation.

The contraction properties are independent of the output mapping so that $g_\theta(x, u)$ can be any function. In all examples we will take the output to be linear in the input and final hidden layer so that $g_\theta(x, u) = Cx + Du$.

2.1. Implicit RNNs

We will refer to the dynamics in (7) as the explicit model. We can also parametrize the same set of models using the following implicit, redundant parametrization:

$$E_0 h^0 = x, \quad E_{\ell+1} h^{\ell+1} = \phi(W_\ell h^\ell + B_\ell u + b_\ell) \text{ for } \ell = 0, ..., L-1, \quad f_\theta(x, u) = E_L h^L, \tag{8}$$

where W_ℓ and E_ℓ are learnable weight matrices and E_ℓ are invertible. Note that the implicit and explicit models are input/output equivalent under the coordinate transformation $z_\ell = E_\ell h_\ell$ and $A_\ell = W_\ell E_{\ell-1}^{-1}$.

We can treat multi-layer networks as a time-varying, periodic, non-linear system by dividing up each k step into L sub-steps so that

$$h_k^{\ell+1} = f^\ell(h_k^\ell, u_k), \quad \ell = 0, ..., L-1 \tag{9}$$

with f^ℓ defined in (8) and $h_{k+1}^0 = h_k^L$. The associated differential dynamics of the network are given by $E_{\ell+1}\delta_k^{\ell+1} = \Lambda\left(h_k, W_\ell \delta_k^\ell\right)$, $\ell = 0, ..., L-1$, where $\Lambda(h_k, \delta_k^\ell)$ is the directional derivative of ϕ at h_k in the direction δ_k^ℓ and δ_k^ℓ is a differential in layer ℓ at time k.

2.2. Contracting Implicit RNNs

We now define the set of contracting implicit RNNs (ci-RNNs): A ci-RNN is an implicit RNN defined as (5) with $f_\theta(x,u)$ defined in (8) with an additional contraction constraint. We propose to use the following constraints to ensure model stability:

$$\begin{pmatrix} E_\ell + E_\ell^\top - P_\ell & W_\ell^\top \\ W_\ell & P_{\ell+1} \end{pmatrix} \succeq 0, \quad \ell = 0, ..., L-1 \tag{10}$$

with $P_0 = \lambda P_L$. The set of ci-RNNs, denoted Θ_{ci} is defined as:

$$\Theta_{ci} := \Big\{ \theta \ : \ \exists P_0, ..., P_L \in \mathbb{D}_+ \text{ s.t. } P_0 = \lambda P_L, \ E + E^\top \succ 0, \ (10) \Big\}$$

Note that Θ_{ci} is convex as it is the intersection a number of semi-definite cones and a linear equality constraint, and for all $\theta \in \Theta_{ci}$, there exists a corresponding explicit RNN (7). Fixing $E_\ell = I$ and $P_\ell = I$ recovers the model set used by Miller and Hardt (2018).

Theorem 1 *Suppose that $\theta \in \Theta_{ci}$, then the model (5), (8) is contracting with rate λ in the metric $V = \delta_x^\top E_0^\top P_0^{-1} E_0 \delta_x$.*

Proof We would like to show that the condition (10) implies the existence of a contraction metric V_k for the system (5), (8), for which $V_{k+1} \leq \lambda V_k$. Via Schur complement (10) is equivalent to:

$$E_\ell + E_\ell^\top - P_\ell - W_\ell^\top P_{\ell+1}^{-1} W_\ell \succeq 0.$$

For all admissible activation functions (slope restricted to the interval $[-1, 1]$) and diagonal $P \succ 0$, we have $\delta^\top P^{-1} \delta \geq \Lambda(h,\delta)^\top P^{-1} \Lambda(h,\delta)$. Left and right multiplying by δ_h gives

$$\delta_{h_{\ell+1}}^\top E_{\ell+1}^\top P_{\ell+1}^{-1} E_{\ell+1} \delta_{h_{\ell+1}} - \delta_h^\top (E + E^\top - P) \delta_h \leq 0.$$

Introducing the storage function $V_k^\ell(\delta_{h_k}^\ell) = {\delta_{h_k}^\ell}^\top E_\ell^\top P_\ell^{-1} E_\ell \delta_{h_k}^\ell$ and using the bound $E^\top P^{-1} E \succ E + E^\top - P$, we can see that $V_k^{\ell+1} - V_k^\ell < 0$. Summing this from $\ell = 0, ..., L-1$ gives $V_k^L - V_k^0 \leq 0$. Due to the periodicity, we have $V_{k+1}^0 = \lambda V_k^L$, so $V_{k+1}^0 \leq \lambda V_k^0$, and the system is contracting in the metric V_k^0. ■

3. Method

We demonstrate the use of the proposed model set in a system identification context. In particular, we are interested in finding functions f_θ and g_θ that minimize the simulation error:

$$\min_{\theta \in \Theta, h_0} J_{sim} = \sum_{k=0}^{T} |y_k - \tilde{y}_k|^2 \quad s.t. \quad h_{k+1} = f_\theta(h_k, \tilde{u}_k), \quad y_k = g_\theta(h_k, \tilde{u}_k) \tag{11}$$

where Θ is the domain of the parameters and $(\tilde{u}_k, \tilde{y}_k)$ are the measured inputs and outputs to system we would like to identify. We will compare the proposed ci-RNN with two others. The first is a regular RNN defined by the equations (7) and the second is the stable RNN (s-RNN) defined by the explicit dynamics (7) with A_ℓ having spectral norm less than 1. We enforce this using the following LMIs:

$$\begin{pmatrix} I & A_\ell^\top \\ A_\ell & I \end{pmatrix} \succeq 0, \quad \ell = 0, ..., L-1. \tag{12}$$

In the one layer case, this is the same model set used in Miller and Hardt (2018).

3.1. Model Initialization

We propose to initialize the models in a two step procedure. Firstly, we sample weights for the explicit model (7) as follows

$$A^{ij} \sim \mathcal{N}\left[0, \frac{\alpha^2}{n}\right], \tag{13}$$

where α is a hyper parameter that relates to how close to instability we expect our model to operate and n is the width of the weight matrix. According to the random matrix circular law (Tao and Vu, 2008), we expect the eigenvalues to be approximately distributed over a circle of radius α. The intuition is to try and generate a rich set of dynamics so that we only need to learn the input and output mappings of the dynamical system. Depending on α, we may find that this method of sampling generates a number of unstable models that complicate training. We project onto the set of contracting implicit models by solving the following convex optimization problem:

$$\min_{\theta \in \Theta_{ci}} \quad \sum_{\ell=0}^{L-1} |A_\ell E_\ell - W_\ell|_F^2 \tag{14}$$

We solve this optimization problem using the cvxpy toolbox (Diamond and Boyd, 2016). For the s-RNN model set, we project onto the set of stable models by clipping the singular values as in Miller and Hardt (2018). We leave the regular RNNs as they are.

3.2. Training Procedure

Fitting the models ci-RNN or s-RNN require a number of LMIs to be satisfied. We do this using the Burer-Montero method that has been shown to be both empirically (Burer and Monteiro, 2003) and theoretically (Boumal et al., 2016) effective for a wide range of problems. This involves replacing a semi-definite constraint $M \succeq 0$ with a series of equality constraints: $M = \mathcal{L}\mathcal{L}^\top$ where $\mathcal{L}$ is an auxiliary matrix variable.

We then use ADAM optimizer to minimize the following objective $J = MSE(y_{0:T}, \tilde{y}_{0:T}) + \mu c^\top c$ where MSE is the mean square error, $y_{0:T}$ are the outputs from simulating the model using (5) and (6) and $c = \text{vec}(M - \mathcal{L}\mathcal{L}^\top)$ are the equality constraints. We use an initial learning rate of 0.5×10^{-3} which decays by a factor of 0.96 at each epoch and an initial penalty parameter of 500. If the equality constraints are violated by more than 1×10^{-3}, we increase the penalty parameter by a factor of 10. The models are trained until more than 20 epochs have passed without seeing a model better than the best seen so for (on validation).

4. Results

We test the proposed approach on two systems. Firstly, we will look at a simple simulated system and explore the effects of implicit parametrizations, stability constraints and model initialization. Then, we will compare the ci-RNN to both the RNN and s-RNN using a human gait prediction task based on data gathered from Motion Capture (MOCAP) experiments. Code to reproduce all examples can be found at https://github.com/imanchester/ci-rnn.

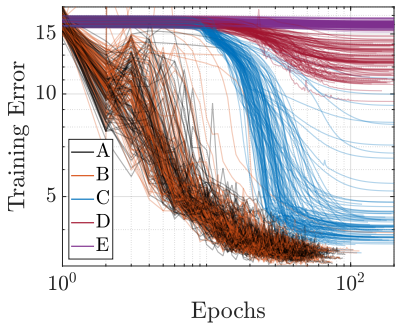


Figure 1: Best viewed in colour. Figure explained in text (Section 4.1).

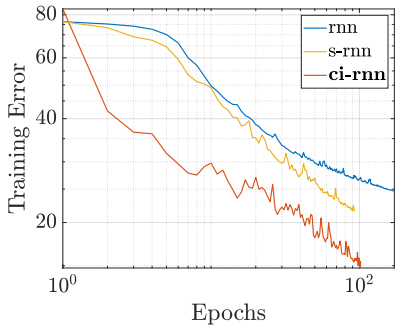


Figure 2: Training error versus epochs for gait prediction task for subject 1.

4.1. Simulated System

We generate data from a slight modification of the system used in Chen et al. (1990). The system has been modified so that it operates closer to the edge of stability, as follows:

$$x_k = 1.4\Bigg[\left(0.8 - 0.5e^{-x_{k-1}^2}\right)x_{k-1} - \left(0.3 + 0.9e^{-x_{k-1}^2}\right)x_{k-2} + u_{k-1} + 0.2u_{k-2} + 0.1u_{k-1}u_{k-2} + w_k\Bigg] \quad (15)$$

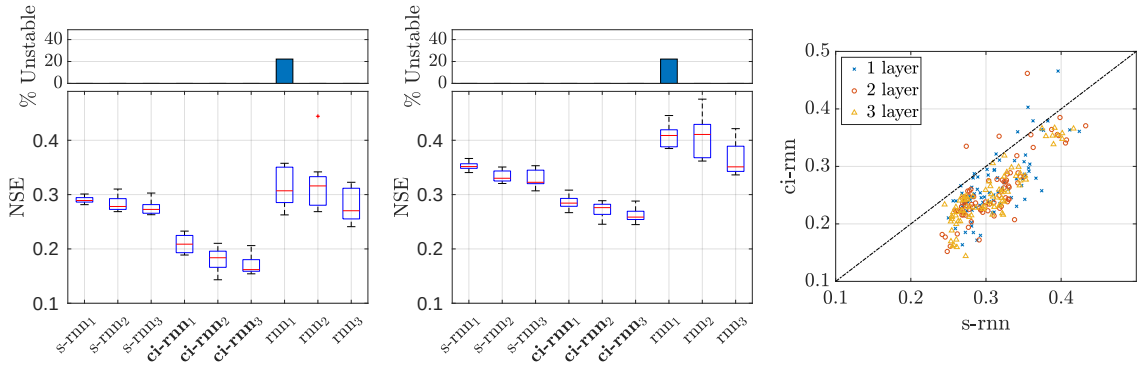
with process noise $w_k \sim \mathcal{N}(0, 0.5)$ and inputs $u_k \sim \mathcal{N}(0, 1)$. For each model realization, we generate a data set consisting of 20 batches of 500 measurements and train models with 2 layers of 60 hidden units per layer and ReLU activations. We train 5 different types model denoted A-E. The models 'A' are ci-RNNs with the initialization scheme in Section 3.1, $\alpha = 1.2$. 'B' are implicit models with the same initialization but without the stability constraint (10). 'C' are implicit models initialized so that $E = I$ and $W_{ij} \sim \mathcal{U}[-\frac{1}{\sqrt{60}}, \frac{1}{\sqrt{60}}]$. 'D' are explicit models with no stability constraints initialized by sampling $A_{ij} \sim \mathcal{N}[0, \frac{1}{60}]$ and finally, 'E' are explicit models initialized by sampling $A_{ij} \sim \mathcal{N}[0, \frac{1}{60}]$ and projecting onto the unit spectral norm ball.

Comparing the models 'C' with 'D' and 'E', we can see that the implicit model structure appears to make training much easier. Comparing the models 'A' and 'B' with 'C', we also see that the initialization procedure appears to significantly speed up training. Finally, comparing A with B we see that the proposed contraction constraint did not hinder training compared to unconstrained models of the same structure. This is in contrast to the spectral norm constraint: comparing D and E we can see that the constraint dramatically hinders training.

4.2. Gait Prediction

The problem is to determine a mapping to the trajectory of the left leg joint angles from the trajectories of the remaining limbs. Such a model can be used, for example, to generate trajectories for an actuated prosthetic limb or exoskeleton. As noted by the authors, this is a system where stability is an important concern as unstable models can lead to unpredictable or dangerous behaviour.

The problem data consists of measurements of joint angles from 9 participants who instructed to walk across flat ground, up a flight of stairs and then stop at the top. Data was gathered using a



(*a*) Subject 1 Training NSE (*b*) Subject 1 Test NSE (*c*) All subjects Test NSE

Figure 3: Performance of models on gait prediction task. The subscripts refer to the number of layers in the model.

MOCAP system. Additional details on the data collection can be found in Boudali et al. (2019b,a). The exercise was repeated 11 times for each participant and the final two trials were withheld as a test set. The remaining 9 datasets were used to train 9 models using 9-fold cross validation. All models have 64 hidden units per layer, ReLU activations and layers varying from 1 to 3.

In Figure 2, for each model, we have plotted the mean square error of the outputs of one trial on the training data versus the epochs. We observe a significant increase in the training speed of the ci-RNN compared to the s-RNN and RNN. We believe that this is due to the proposed initialization scheme and the increased flexibility provided by the redundant parametrization. In order to compare the performance of the resulting models across different participants and model outputs, we use Normalized Simulation Error (NSE) as a performance metric, calculated as:

$$\text{NSE} = \frac{\sum_t |y_t - \tilde{y}_t|^2}{\sum_t |\tilde{y}_t|^2}. \tag{16}$$

The box-plots in Figure 3(*a*) and 3(*b*) show the average NSE across the 6 outputs for the training and test datasets for a single participant across the 9 models trained. We see that in each case the ci-RNN outperforms the models RNN and s-RNN. Additionally we also observe a number of unstable models in the RNN model set that have unbounded NSE. Figure 3(*c*) compares the NSE of the ci-RNN and s-RNN for all models trained and all participants. As the vast majority of the point lie beneath the line $y = x$, we can see that almost all ci-RNNs trained outperform the corresponding s-RNNs, does so in every case for the 3-layer networks.

References

Charles W. Anderson, Peter Michael Young, Michael R. Buehner, James N. Knight, Keith A. Bush, and Douglas C. Hittle. Robust Reinforcement Learning Control Using Integral Quadratic Constraints for Recurrent Neural Networks. *IEEE Transactions on Neural Networks*, 18(4):993–1002, July 2007.

David Angeli. A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, March 2002.

Nikita. E. Barabanov and Danil. V. Prokhorov. Stability analysis of discrete-time recurrent neural networks. *IEEE Transactions on Neural Networks*, 13(2):292–303, March 2002.

Mounir A. Boudali, Peter J. Sinclair, and Ian R. Manchester. Predicting Transitioning Walking Gaits: Hip and Knee Joint Trajectories From the Motion of Walking Canes. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(9):1791–1800, September 2019a.

Mounir A. Boudali, Peter J. Sinclair, and Ian R. Manchester. Prediction of Smooth Gait Transitioning for Active Lower Limb Prosthetics. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2424–2429, July 2019b.

Nicolas Boumal, Vladislav Voroninski, and Afonso S. Bandeira. The Non-convex BurerMonteiro Approach Works on Smooth Semidefinite Programs. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2765–2773, USA, 2016. Curran Associates Inc. event-place: Barcelona, Spain.

Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, February 2003.

Sheng. Chen, Stephen. A. Billings, and P. M. Grant. Non-linear system identification using neural networks. *International Journal of Control*, 51(6):1191–1214, January 1990.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN EncoderDecoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 2014. Association for Computational Linguistics.

Yun-Chung Chu and Keith Glover. Bounds of the induced norm and model reduction errors for systems with repeated scalar nonlinearities. *IEEE Transactions on Automatic Control*, 44(3): 471–483, March 1999.

Steven Diamond and Stephen Boyd. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *The Journal of Machine Learning Research*, 17(1):2909 – 2913, 2016.

Fulvio Forni and Rodolphe Sepulchre. A Differential Lyapunov Framework for Contraction Analysis. *IEEE Transactions on Automatic Control*, 59(3):614–628, March 2014.

Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Studies in Computational Intelligence. Springer-Verlag, Berlin Heidelberg, 2012.

Sepp Hochreiter and Jurgen Schmidhuber. Long Short-Term Memory. *Neural computation*, 9: 1735–1780, 1997.

Herbert Jaeger. Adaptive Nonlinear System Identification with Echo State Networks. In *Advances in Neural Information Processing Systems*, pages 609–616. MIT Press, 2003.

Eugenius Kaszkurewicz and Amit Bhaya. Robust Stability and Diagonal Liapunov Functions. *SIAM Journal on Matrix Analysis and Applications*, 14(2):508–520, 1993.

Eugenius Kaszkurewicz and Amit Bhaya. On a class of globally stable neural circuits. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 41(2):171–174, February 1994.

Eugenius Kaszkurewicz and Amit Bhaya. *Matrix Diagonal Stability in Systems and Computation*. Birkhuser Basel, 2000.

James N. Knight and Charles W. Anderson. Stable reinforcement learning with recurrent neural networks. *Journal of Control Theory and Applications*, 9:410–420, 2011.

Quoc V. Le, Navdeep Jaitly, and Geoffrey E. Hinton. A Simple Way to Initialize Recurrent Networks of Rectified Linear Units. *arXiv:1504.00941 [cs]*, April 2015. arXiv: 1504.00941.

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

Danilo P. Mandic and Jonathon Chambers. *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

Hermann Mayer, Faustino Gomez, Daan Wierstra, Istvan Nagy, Alois Knoll, and Jrgen Schmidhuber. A System for Robotic Heart Surgery that Learns to Tie Knots Using Recurrent Neural Networks. *Advanced Robotics*, 22(13-14):1521–1537, January 2008.

Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. Efficient Orthogonal Parametrisation of Recurrent Neural Networks Using Householder Reflections. In *International Conference on Machine Learning*, pages 2401–2409, July 2017.

John Miller and Moritz Hardt. Stable Recurrent Models. *In Proceedings of ICLR 2019*, May 2018. arXiv: 1805.10369.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

Jonas Sjberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre-Yves Glorennec, H\a akan Hjalmarsson, and Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

Eduardo D. Sontag. Input to State Stability: Basic Concepts and Results. In *Nonlinear and Optimal Control Theory*, volume 1932, pages 163–220. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-77644-4 978-3-540-77653-6.

Terence Tao and Van Vu. Random matrices: the circular law. *Communications in Contemporary Mathematics*, 10(02):261–307, April 2008.

Danielle C. Tarraf, William Shelton, Edward Parker, Brien Alkire, Diana Gehlhaus Carew, Justin Grana, Alexis Levedahl, Jasmin Leveille, Jared Mondschein, James Ryseff, Ali Wyne, Dan Elinoff, Edward Geist, Benjamin N. Harris, Eric Hui, Cedric Kenney, Sydne Newberry, Chandler Sachs, Peter Schirmer, Danielle Schlang, Victoria M. Smith, Abbie Tingstad, Padmaja Vedula, and Kristin Warren. The Department of Defense Posture for Artificial Intelligence: Assessment and Recommendations, 2019.

Mark M Tobenkin, Ian R. Manchester, and Alexandre Megretski. Convex parameterizations and fidelity bounds for nonlinear identification and reduced-order modelling. *IEEE Transactions on Automatic Control*, 62(7):3679–3686, 2017.

Jack Umenberger and Ian R. Manchester. Convex Bounds for Equation Error in Stable Nonlinear Identification. *IEEE Control Systems Letters*, 3(1):73–78, 2019a.

Jack Umenberger and Ian R. Manchester. Specialized Interior-Point Algorithm for Stable Nonlinear System Identification. *IEEE Transactions on Automatic Control*, 64(6):2442–2456, June 2019b.

Bo Wahlberg. System identification using Laguerre models. *IEEE Transactions on Automatic Control*, 36(5):551–562, May 1991.

Zhang Yi. *Convergence Analysis of Recurrent Neural Networks*. Network Theory and Applications. Springer US, 2004.

George Zames. On the input-output stability of time-varying nonlinear feedback systems Part one: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Transactions on Automatic Control*, 11(2):228–238, April 1966.

Jiong Zhang, Qi Lei, and Inderjit Dhillon. Stabilizing Gradients for Deep Neural Networks via Efficient SVD Parameterization. In *International Conference on Machine Learning*, pages 5806–5814, July 2018.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany, August 2016. Association for Computational Linguistics.

Alex Zyner, Stewart Worrall, James Ward, and Eduardo Nebot. Long short term memory for driver intent prediction. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1484–1489. IEEE, June 2017.