Input-Output Equivalence of Unitary and Contractive RNNs

Melikasadat Emami, Mojtaba Sahraee-Ardakan, Alyson K. Fletcher
University of California, Los Angeles

Sundeep Rangan
New York University

Background

RNNs:

- Sequence-to-sequence mappings of the form:

$$h^{(k)} = \phi\left(Wh^{(k-1)} + F\,x^{(k)} + b\right),$$
$$y^{(k)} = Ch^{(k)}, \quad h^{(-1)} = h_{-1}.$$

- Parameters: $\Theta = (W, F, b\, C, h_{-1})$
- Input- output mapping: $y = G(x, \Theta)$

Equivalence of RNNs:

- Given Θ_1 and Θ_2:

$$G(x, \Theta_1) = G(x, \Theta_2) \quad \text{for all} \quad x = (x^{(0)}, \ldots, x^{(T-1)})$$

- Internal states may be different
- Does not imply that parameters are identical
- Example: invertible T, identity activation
- $W \to TWT^{-1}, \quad C \to CT^{-1}, \quad F \to TF, \quad h_{-1} \to Th_{-1}$

Contractive RNNs:

- $\|W\| := \max_{h \neq 0} \frac{\|Wh\|_2}{\|h\|_2}$
- Contractive: $\|W\| < 1$, non-expansive: $\|W\| \leq 1$.
- Non expansive activation function:

$$\|\phi(x) - \phi(y)\| \leq \|x - y\| \quad \text{for all} \quad x, y$$

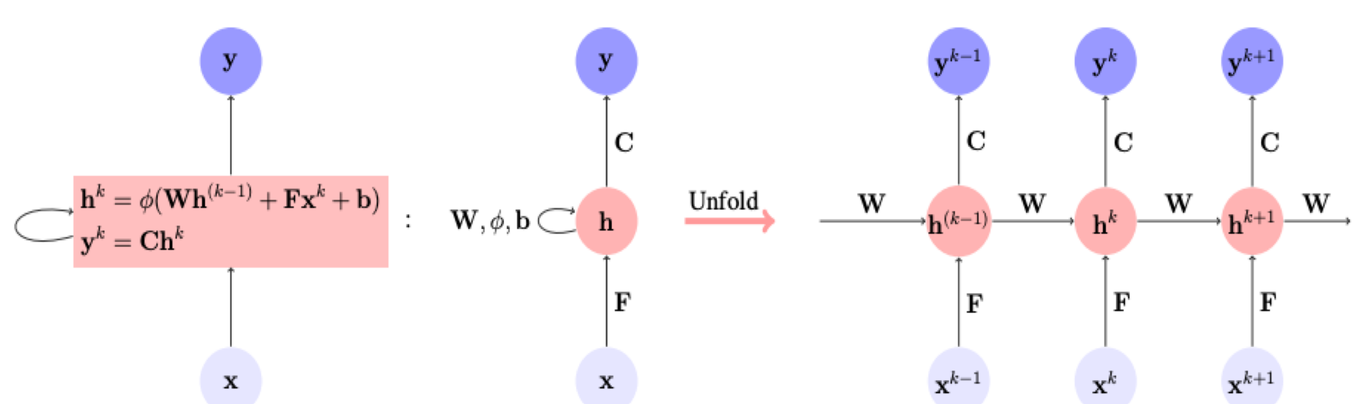


Figure 1: Recurrent neural network (RNN) model

Unitary RNNs (URNN):

- $W^H W = WW^H = I$
- Overcome the vanishing/exploding gradient problem
- Improve the stability of the network

This work:

- Characterizes how restrictive the unitary constraint is on an RNN
- Compares input-output mappings achievable by URNNs and RNNs

Main Results

Equivalence Results for RNNs with ReLU Activations:

Theorem 1: Given any contractive RNN with n hidden states, bounded input, and ReLU activations, there exists a URNN with at most $2n$ hidden states and the identical input-output mapping.

- No loss in modeling with URNNs compared to RNNs
- Cost: two-fold increase in state dimension

Proof idea:

- Construct a URNN with $2n$ states
- Match the first n states with the original RNN
- Last n states are zero

Theorem 2: For every positive n, there exists a contractive RNN with ReLU activations and state dimension n such that every equivalent URNN has at least $2n$ states.

- Converse result for Theorem 1
- $2n$ achievability is tight

Acknowledgements:
The work of M. Emami, M. Sahraee-Ardakan, A. K. Fletcher was supported in part by the NSF Grants 1254204 and 1738286, and the Office of Naval Research under Grant N00014-15-1-2677. S. Rangan was supported in part by the NSF Grants 1116589, 1302336, and 1547332, NIST, the industrial affiliates of NYU WIRELESS, and the SRC.

Main Results

Equivalence Results for RNNs with Sigmoid Activations:

Theorem 3: There exists a contractive RNN with sigmoid activations such that there is no URNN with any finite number of states that exactly matches the input-output mapping.

- No exact equivalence even with arbitrary number of states
- Difference due to ReLU zeroing out states

Numerical Simulations

Data Generation:

- Multiple instances of a synthetic RNN with 4 hidden units
- b, F, C matrices ~ iid Gaussian
- Contractive $W_g = I - \epsilon A^T A / \|A\|^2$, A: Gaussian iid
- $\epsilon = 0.01 \to$ slow varying system $\to$ long term dependencies
- Biases adjusted to ensure hidden states are on 60% of the time
- Additive output noise with SNR= 15, 20 dB
- Each trial: $T = 1000$ i.e.(10×time constant($\frac{1}{\epsilon}$)), test ratio = 0.3

Learning the system:

- Using standard RNNs, URNNs with [2,4,6,8,10,12,14] hidden units
- MSE loss , batch-size =10 , learning-rate = 0.01
- Averaged over 30 realization of original contractive system
- Unitary constraint: projection on unitary space using SVD .

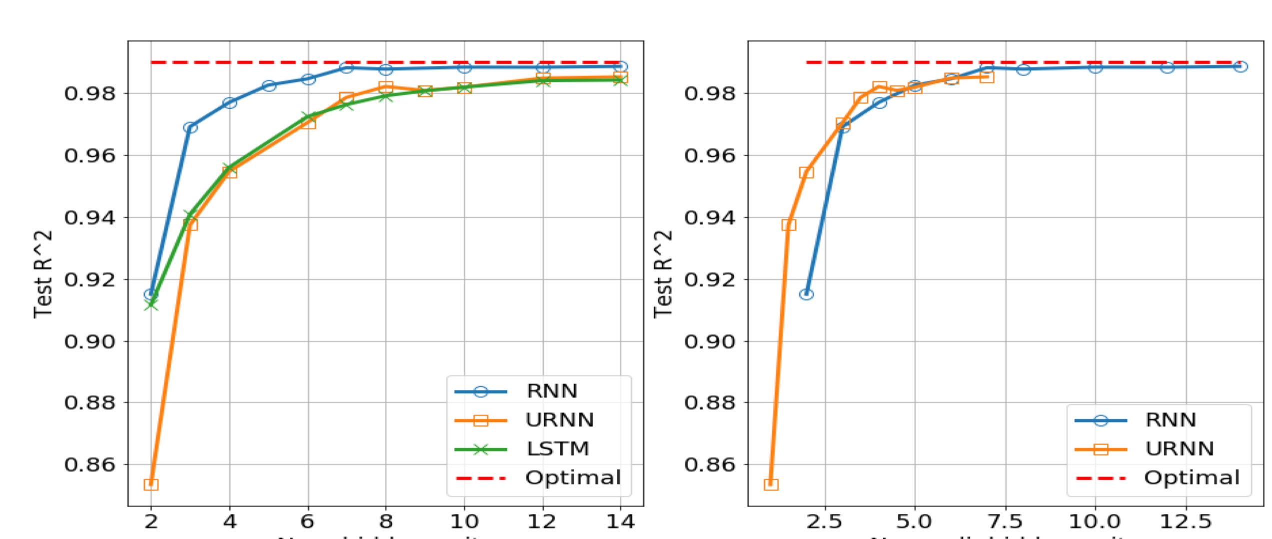


Figure 2: Test R^2 on synthetic data for a Gaussian i.i.d. input and output SNR=20 dB.

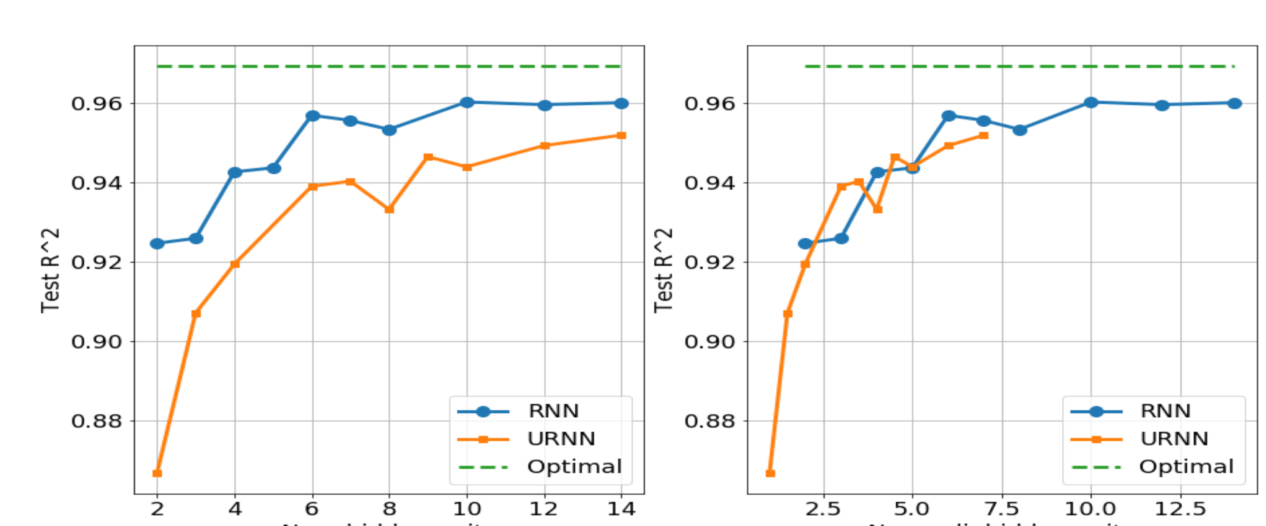
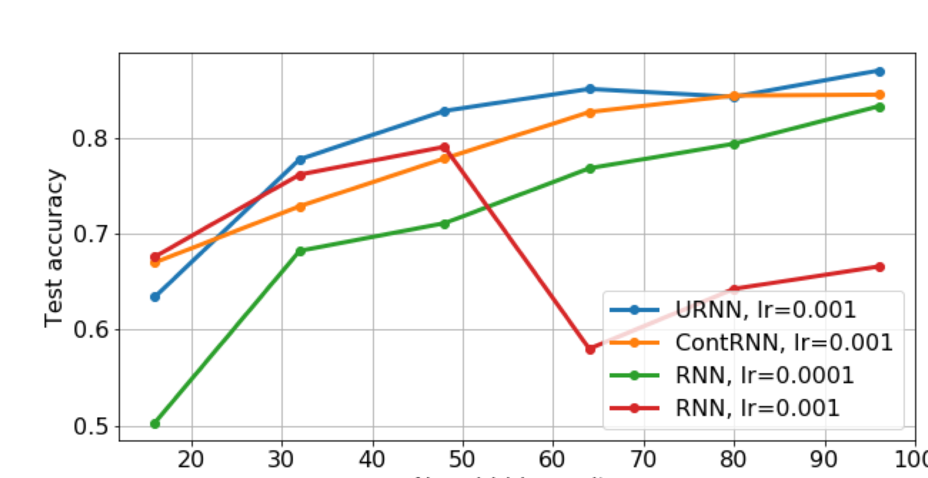


Figure 3: Test R^2 on synthetic data for a Gaussian i.i.d. input and output SNR=15 dB.



Accuracy on Permuted MNIST task for various models trained with RMSProp, validation-based early termination.

References:
[1] M. Emami, M. Sahraee-Ardakan, S. Rangan, A. K. Fletcher. Input-Output Equivalence of Unitary and Contractive RNNs. To appear in NeurIPS, 2019.
[2] M. Arjovsky, A. Shah, and Y. Bengio. Unitary evolution recurrent neural networks. In ICML, 2016.
[3] L. Jing, Y. Shen, T. Dubcek, J. Peurifoy, S. Skirlo, Y. LeCun, M. Tegmark, and M. Soljacic. Tunable efficient unitary neural networks (eunn) and their application to rnns. In ICML, 2017.
[4] S. Wisdom, T. Powers, J. Hershey, J. L. Roux, and L. Atlas. Full-capacity unitary recurrent neural networks. In NIPS, 2016.