Recurrent neural networks in $\mathbb{C}$

Moritz Wolter

wolter@cs.uni-bonn.de

June 6, 2019

Motivation

- Quantum computers require complex unitary weights.
- Fourier transforms produce complex representations.
- Encoding data in magnitude and phase may enable us create a richer representation.
- Complex analysis is a well studied (and very interesting!) subject, lets merge it with machine learning and see what happens.

Memory and adding benchmark problems for RNNs

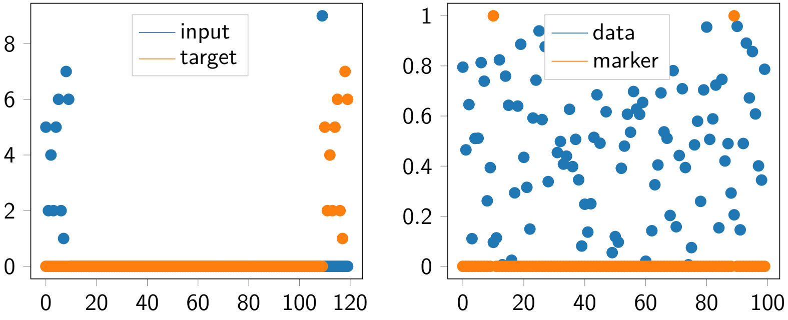


Figure: Illustrations of the memory problem on the left and the adding problem on the right.

Wirtinger-Calculus [Wir27][MG09][KD09]

For a complex function $f(z) = u(x, y) - iv(x, y)$ we have:

$$\mathbb{R}\text{-derivative} \triangleq \frac{\partial f}{\partial z}\Big|_{\bar{z}=\text{const}} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - i\frac{\partial f}{\partial y}\right), \tag{1}$$

$$\overline{\mathbb{R}}\text{-derivative} \triangleq \frac{\partial f}{\partial \bar{z}}\Big|_{z=\text{const}} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + i\frac{\partial f}{\partial y}\right). \tag{2}$$

Based on these derivatives, one can define the chain rule for a function $g(f(z))$ as follows:

$$\frac{\partial g(f(z))}{\partial z} = \frac{\partial g}{\partial f}\frac{\partial f}{\partial z} + \frac{\partial g}{\partial \bar{f}}\frac{\partial \bar{f}}{\partial z} \text{ where } \bar{f} = u(x, y) - iv(x, y). \tag{3}$$

Theoretical tool to convince ourselves, that it's ok to work with equivalent real networks.

Unitary Evolution matrix RNN-Motivation [ASB16][Pas13]

$$\mathbf{x}_t = \mathbf{W}_{\text{rec}} f(\mathbf{x}_{t-1}) + \mathbf{W}_{\text{in}} \mathbf{u}_t + \mathbf{b}. \tag{4}$$

$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{1 \leq t \leq T} \frac{\mathcal{E}_t}{\partial \theta}, \tag{5}$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{1 \leq k \leq t} \left(\frac{\mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\mathbf{x}_k} \frac{\partial^+ \mathbf{x}_k}{\partial \theta} \right), \tag{6}$$

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \geq i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \prod_{t \geq i > k} W_{\text{rec}}^T \text{diag}(f'(\mathbf{x}_{i-1})). \tag{7}$$

Stiefel Manifold Weight Updates [WPH+16]

$$\mathbf{W}_{k+1} = (\mathbf{I} + \frac{\lambda}{2}\mathbf{A}_k)^{-1}(\mathbf{I} - \frac{\lambda}{2}\mathbf{A}_k)\mathbf{W}_k, \tag{8}$$

where $$\mathbf{A} = \mathbf{W}\overline{\nabla_{\mathbf{w}} F}^T - \overline{\mathbf{W}}^T \nabla_{\mathbf{w}} F. \tag{9}$$

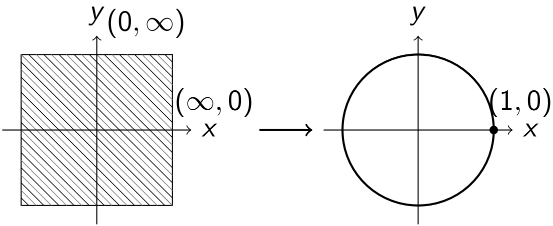


Figure: Fix the optimized matrix eigenvalues onto the unit circle. The key idea behind stiefel-manifold optimization.

The linear unitary case

$$\mathbf{x}_t = \mathbf{W}_{\text{rec}}\mathbf{x}_{t-1} + \mathbf{W}_{\text{in}}\mathbf{u}_t + \mathbf{b}. \tag{10}$$

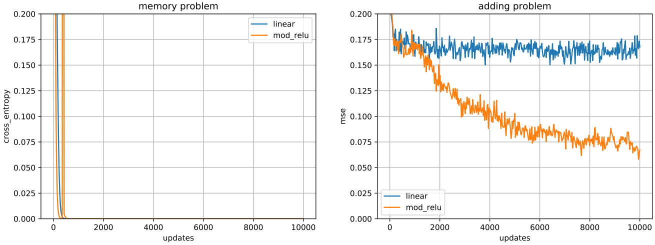
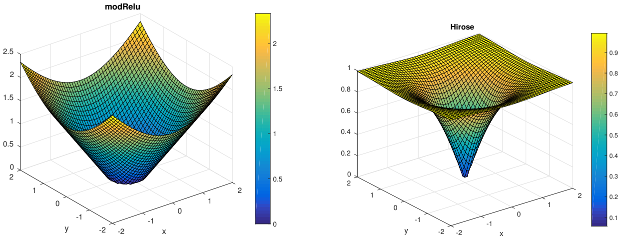


Figure: Performance of linear and mod-Relu activated unitary RNNs on the memory (left) and adding (right) problems for T=50. All networks have approx. 40k weights.

Complex equivalents of tanh and Relu



$$f_{\text{Hirose}}(z) = \tanh\left(\frac{|z|}{m^2}\right) e^{-i\cdot\theta_z} = \tanh\left(\frac{|z|}{m^2}\right) \frac{z}{|z|}, \tag{11}$$

$$f_{\text{modReLU}}(z) = \text{ReLU}(|z| + b) e^{-i\cdot\theta_z} = \text{ReLU}(|z| + b) \frac{z}{|z|}. \tag{12}$$

We will compare their performance as state-to-state non-linearities.

Unitary evolution network performance

$$\mathbf{x}_t = \mathbf{U}_{\text{rec}} f(\mathbf{x}_{t-1}) + \mathbf{W}_{\text{in}} \mathbf{u}_t + \mathbf{b}. \tag{13}$$

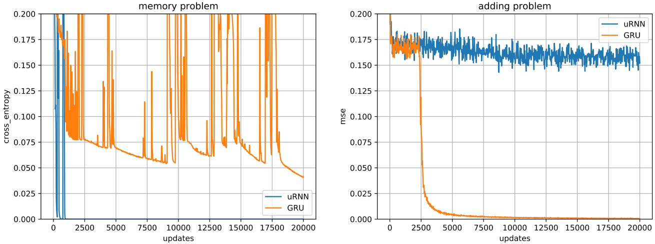
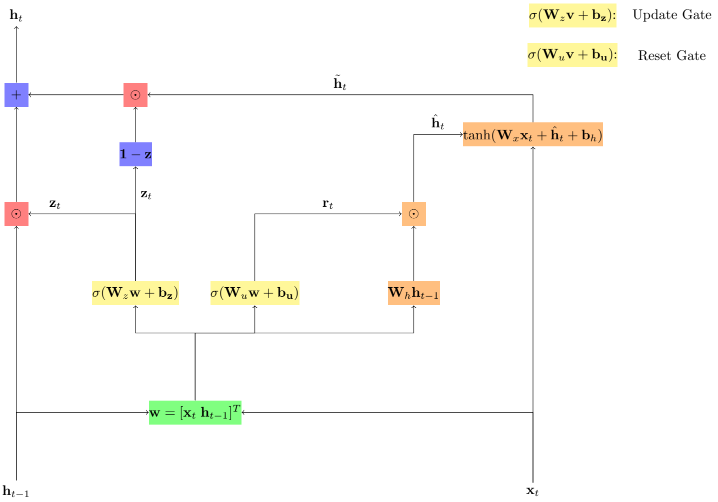


Figure: Current state of the art performance on memory and adding problem for T=250. Models have approximately 40k weights.

The gated recurrent unit



Complex gated Recurrent Recurrent Nets [WY18a]

Gate equation:

$$\mathbf{g}_r = f_g(\mathbf{z}_r), \qquad \text{where} \qquad \mathbf{z}_r = \mathbf{W}_r\mathbf{h} + \mathbf{V}_r\mathbf{x}_t + \mathbf{b}_r, \tag{14}$$

$$\mathbf{g}_z = f_g(\mathbf{z}_z), \qquad \text{where} \qquad \mathbf{z}_z = \mathbf{W}_z\mathbf{h} + \mathbf{V}_z\mathbf{x}_t + \mathbf{b}_z, \tag{15}$$

Update equations:

$$\widetilde{\mathbf{z}}_t = \mathbf{W}(\mathbf{g}_r \odot \mathbf{h}_{t-1}) + \mathbf{V}\mathbf{x}_t + \mathbf{b}, \tag{16}$$

$$\mathbf{h}_t = \mathbf{g}_z \odot f_a(\widetilde{\mathbf{z}}_t) + (1 - \mathbf{g}_z) \odot \mathbf{h}_{t-1}, \tag{17}$$

$\mathbb{C} \rightarrow \mathbb{R}$, mapping:

$$\mathbf{o}_r = \mathbf{W}_o[\Re(\mathbf{h})\ \Im(\mathbf{h})] + \mathbf{b}_o. \tag{18}$$

Complex gate activations

$$f_{\text{mod sigmoid}}(\mathbf{z}) = \sigma(\alpha\Re(\mathbf{z}) + \beta\Im(\mathbf{z})). \tag{19}$$

With $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$.

Comparison to state of the art

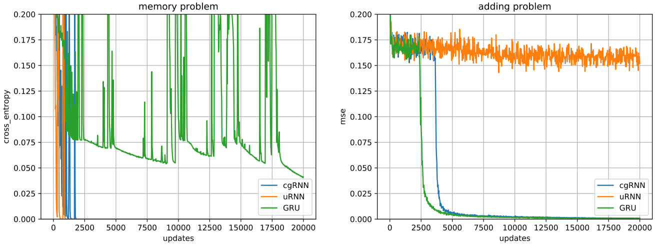


Figure: Comparison of our complex gated RNN (cgRNN, blue, $n_h = 80$) with the unitary RNN [ASB16](uRNN, orange, $n_h = 140$) and standard GRU [CvMG+14](orange, $n_h = 112$) on the memory (left) and adding (right) problem for $T = 250$.

Stiefel optimization and activations

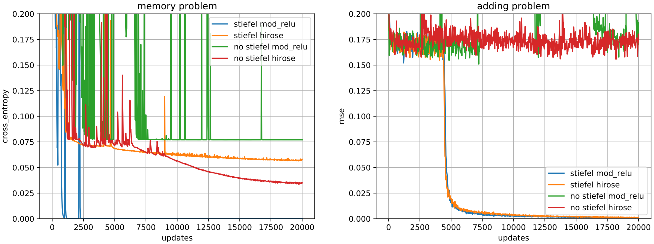


Figure: Comparison of non-linearities and norm preserving state transition matrices on the complex gated RNNs for the memory (a) and adding (b) problems for T=250. We use $n_h = 80$ for all experiments.

Weight reducitons on mocap data

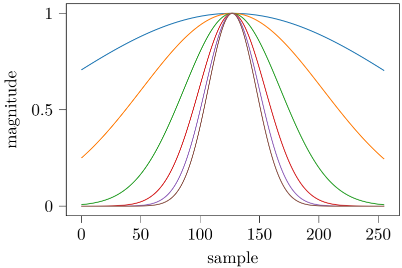
Table 2: Comparison of our cgRNN with the GRU [28] on human motion prediction.

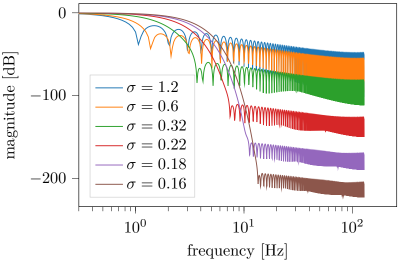
	cgRNN				GRU[28]			
Action	80ms	160 ms	320ms	400ms	80ms	160ms	320ms	400ms
walking	0.29	0.48	0.74	0.84	**0.27**	**0.47**	**0.67**	**0.73**
eating	0.23	**0.38**	0.66	0.82	0.23	0.39	**0.62**	**0.77**
smoking	**0.31**	**0.58**	**1.01**	**1.1**	0.32	0.6	1.02	1.13
discussion	0.33	0.72	**1.02**	**1.08**	**0.31**	**0.7**	1.05	1.12
directions	0.41	**0.65**	**0.83**	**0.93**	**0.41**	0.65	0.83	0.96
greeting	0.53	0.87	**1.26**	**1.43**	**0.52**	**0.86**	1.30	1.47
phoning	**0.58**	1.09	1.57	1.72	0.59	**1.07**	**1.50**	**1.67**
posing	**0.37**	**0.72**	**1.38**	**1.65**	0.64	1.16	1.82	2.1
purchases	0.61	0.86	**1.21**	1.31	**0.6**	**0.82**	1.13	**1.21**
sitting	0.46	0.75	1.22	**1.44**	**0.44**	**0.73**	**1.21**	1.45
sitting down	0.55	1.02	1.54	1.73	**0.48**	**0.89**	**1.36**	**1.57**
taking photo	**0.29**	**0.59**	**0.92**	**1.07**	0.29	0.59	0.95	1.1
waiting	0.35	0.68	1.16	**1.36**	**0.33**	**0.65**	**1.14**	1.37
walking dog	0.57	1.09	1.45	1.55	**0.54**	**0.94**	**1.32**	**1.49**
walking together	**0.27**	**0.53**	**0.77**	**0.86**	0.28	0.56	0.8	0.88
average	**0.41**	**0.73**	**1.12**	**1.26**	0.42	0.74	**1.12**	1.27

Our cgRNN (n_h =512, 1.8M params) predicts human motions which are either comparable or slightly better than the real-valued GRU [28] (n_h =1024, 3.4M params) despite having only approximately half the parameters.

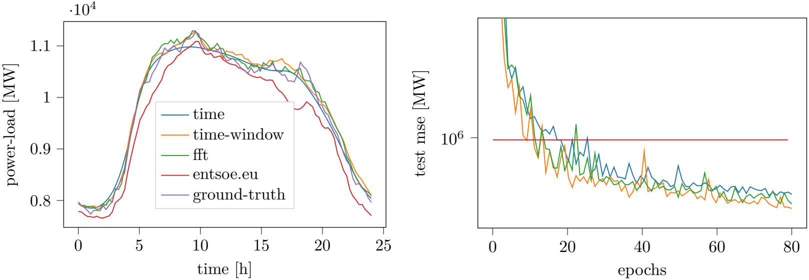
The Short time Fourier transform

$$\mathbf{X}[\omega, Sm] = \mathcal{F}_s(\mathbf{x}) = \mathcal{F}(\mathbf{w}[Sm - l]\mathbf{x}[l]) = \sum_{l=-\infty}^{\infty} \mathbf{w}[Sm - l]\mathbf{x}[l]e^{-j\omega l}, \quad (20)$$



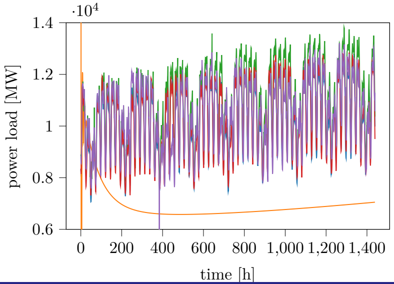


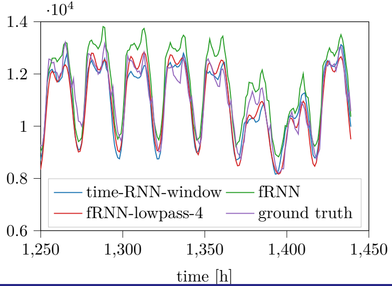
Learning trough the STFT [WY18b]



Learning trough the STFT [WY18b]

Network	mse [MW]	weights	run [min]
time-RNN	$1.3 \cdot 10^7$	13k	772
time-RNN-windowed	$8.8 \cdot 10^5$	28k	12
fRNN	$8.3 \cdot 10^5$	44k	13
fRNN-lowpass-1/4	$7.6 \cdot 10^5$	20k	13
fRNN-lowpass-1/8	$1.3 \cdot 10^6$	16k	13





Upcoming: Temporal Convolutions in the frequency domain

- The STFT turns sequences of numbers into images.
- What happens if we convolve these in time and frequency?
- What is the best way to convolve in frequency and time?
- How can recurrent connections and convolutions best be used together?

References I

- Martin Arjovsky, Amar Shah, and Yoshua Bengio, *Unitary evolution recurrent neural networks*, ICML, 2016.
- Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, *Learning phrase representations using RNN encoder–decoder for statistical machine translation*, EMNLP, October 2014.
- Ken Kreutz-Delgado, *The complex gradient operator and the cr-calculus*, arXiv preprint arXiv:0906.4835 (2009).

References II

- Danilo P Mandic and Vanessa Su Lee Goh, *Complex valued nonlinear adaptive filters: noncircularity, widely linear and neural models*, vol. 59, John Wiley & Sons, 2009.
- Pascanu, *On the difficulty of training recurrent neural networks*, Journal of Machine Learning Research (2013).
- W. Wirtinger, *Zur formalen theorie der funktionen von mehr komplexen veränderlichen*, 1927.
- Scott Wisdom, Thomas Powers, John R. Hershey, Jonathan Le Roux, , and Les Atlas, *Full-capacity unitary recurrent neural networks*, Advances in Neural Information Processing Systems, 2016.

References III

- Moritz Wolter and Angela Yao, *Complex gated recurrent neural networks*, 32nd Conference on Neural Information Processing Systems, 2018.
- ———, *Fourier rnns for sequence prediction*, arXiv preprint arXiv:1812.05645, 2018.

Discussion

Thanks for your attention and feedback.
Feel free to contact me at: wolter@cs.uni-bonn.de