Complex unitary memory units

Moritz Wolter
Uni Bonn

Angela Yao
Uni Bonn

May 10, 2018

Abstract

RNN optimization often suffers from numberically unstable gradients. We propose a novel complex RNN architecture, which can be shown to be numerically stable. Building on top of recent sucesses in the optimization of complex valued neural networks we propose a novel memory cell, which allows us to take gated recurrent units to the complex domain. In short we optimize:

$$\min_{\mathbf{W}} \text{cost}(\{\mathbf{x}\}, \{\mathbf{W}\}) \tag{1}$$

$$\text{such that } \forall m \; \|\phi_m\| = 1, \tag{2}$$

$$\forall n \; \|f'(h_n)\| \leq 1. \tag{3}$$

Where $\{\mathbf{W}\}$ denotes the set of network weights $\{\mathbf{x}\}$ the set of network inputs and $\{\phi_m\}$ the set of all weight matrix eigenvalues and finally $\|f'(h_n)\|$ the hidden actication derivatives. We show that our complex gated memory cells are practically stable and use wirtinger calculus to overcome limitations on skalar activations set by Liouville's theorem. It turns out that we do not need to work with a constrained optimization algorithm here, but can instead rewrite the problem in an unconstrained way, and use libraries optimized for large scale unconstrained optimization such as tensorflow or pytorch.

1 Introduction

Recurrent neural networks (RNNs) are widely used for processing time series and sequential information. The difficulties of training RNNs, especially when trying to learn long-term dependencies, are well-established, as RNNs are prone to vanishing and exploding gradients [2, 12, 18]. Heuristics have been developed to alleviate some of the optimization instabilities and learning difficulties. They include gradient clipping **(AY: ref?)**, gating, as used in gated recurrent units (GRUs) and long short-term memory (LSTM) networks, and using norm-preserving weight matrices. RNNs of the latter type are particularly interesting because they are guaranteed to be mathematically stable [1]. To be norm-preserving, weight matrices need to be either orthogonal or unitary[1].

Arjovsky *et al.* [1], as well as follow-up works [13, 26] advocate working in the complex domain, since enforcing weight matrices to be unitary is less restrictive than enforcing orthogonality. Unitary Matrices can spread their eigenvalues over the entire unit circle, while orthogonal matrices have eigenvalues of either minus one or one. However, complex neural networks have been receiving attention in their own right **(AY: add citations of complex networks)**. **(AY: add reasons why we want to work in the complex domain; see benefits form chapter 2 of Mandic, complex filtering book)**. **(AY: It is our interest to develop the mathematical theory for working with complex recurrent neural networks)**.

We (along with many others **(AY: citations)**) posit that working in the complex domain can significantly increase the functionality of neural networks while simplifying the learning task. Complex neural networks are more than simply a real-valued counterpart with twice as many parameters / dimensions.

Directly extending the mathematics of RNNs to complex networks is non-trivial because of **(AY: x and Y)**. **(AY: To date, has only been shown for CNNs, no gates, etc.)** In this work, we propose a stable gated RNN **(AY: discuss contributions of paper)**.

2 Related work

Normalized complex matrices where first introduced into the literature by [1]. Since then [26], expanded the reach of the unitary matrix basis. An idea that is taken further by [13], which makes use Lie group theory. A holomorph non-linearity was used in [11], [1] introduces a novel non-linearity which is not complex-differentiable. [22] compares complex non-linearities and systematically measures performance. Furthermore complex batch-normalization is introduced. Finally [14], proposes a gated unitary RNN, but is restricted to the real numbers.
Finding a gradient for functions from $\mathbb{C}$ to $\mathbb{R}$, is a problem which has been adressed in the digital signal processing literature. Where complex problems with real cost functions had to be solved [4][23][7][15]. Applications to neural network cost functions where considered later [16]. All solutions essentially utilize Wirtinger calculus [25], to come up with an approximate gradient for a non-holomorph function.

3 Preliminaries

A complex-valued function f mapping

[1] Unitary matrices are the complex analogue of orthogonal matrices, *i.e.*a complex matrix W is unitary if $WW^* = W^*W = I$, where W^* is its conjugate transpose and I is the identity matrix. **(AY: check necessity of square in definition)**.

(AY: add small primer on requirements for complex networks, holomorph, Wirtinger calculus / pseudo-gradients; see section II of Scardapane section II)

3.1 $\mathbb{CR}$ Calculus

(AY: definition of z = u + iv; separation of complex into real and imaginary)
(AY: to be truly differentiable / holomorph, needs to satisfy Cauchy-Riemann conditions; most people are put off by Liouville's condition.)
(AY: however, can leverage the use of CR-Calculus, also known as Wirtinger calulus. In such a scenario, add equations of real and imaginary derivative; approximations)

3.2 Complex Activation Functions

- non-linear and bounded
- for learning, partial derivatives should exist and also be bounded

Based on Liousville's theorem, functions from $\mathbb{C} \to \mathbb{R}$ cannot be holomorph unless they are constant

4 Complex Gated Network

4.1 Norm-Preserving RNNs

Suppose we are given a neural network with T hidden layers and an objective function C. Let x_t and h_t represent the input and hidden unit vectors at layer t, f be a point-wise non-linearity function, and W_t and V_t be the hidden and input weight matrices respectively. The network can be defined as

$$z_{t+1} = \mathbf{W}_t h_t + \mathbf{V}_t x_{t+1} \tag{4}$$
$$h_{t+1} = f(z_{t+1}). \tag{5}$$

A matrix $\mathbf{W}$ is norm-preserving if its repeated multiplication with a vector leaves the vector norm unchanged, *i.e.*$\|\mathbf{W}h\|_2 = \|h\|_2$. If $\mathbf{W}$ contains only real-valued entries, and $\mathbf{W}^\intercal\mathbf{W} = \mathbf{W}\mathbf{W}^\intercal = \mathbf{I}$. If $\mathbf{W}$ contains entries from the complex domain, it is *unitary* and $\mathbf{W}^*\mathbf{W} = \mathbf{W}\mathbf{W}^* = \mathbf{I}$, where $\mathbf{W}^*$ is the complex conjugate transpose of $\mathbf{W}$.

In [1], Arjovsky *et al.* prove that with an orthogonal or unitary weight matrices $\mathbf{W}_k$, one can avoid exploding gradients of the cost function C with respect to h_t. More specifically, the gradient magnitude can be bounded as follows

$$\left\|\frac{\partial C}{\partial h_t}\right\| \leq \left\|\frac{\partial C}{\partial h_T}\right\| \prod_{k=t}^{T-1} \|\mathbf{D}_{k+1}\mathbf{W}_k^T\| = \left\|\frac{\partial C}{\partial h_T}\right\| \prod_{k=t}^{T-1} \|\mathbf{D}_{k+1}\| = \left\|\frac{\partial C}{\partial h_T}\right\|, \tag{6}$$

where $\mathbf{D}_k = \text{diag}(f'(z_k))$ is the Jacobian matrix of the pointwise non-linearity.

Since $\mathbf{D}_k$ is a diagonal matrix, $\|\mathbf{D}_k\| = \max_{j=1,\ldots,n} |f'(z_k^{(j)})|$, where $z_k^{(j)}$ is the j^{th} pre-activation of the k^{th} hidden layer. This proof hinges the critical assumption that $\|\mathbf{D}_k\| = 1$ for all layers k. For real pre-activations, *i.e.*$z \in \mathbb{R}$, this constraint is easily met with the standard rectified linear unit or ReLU. For complex pre-activations $z \in \mathbb{C}$, however, one needs a non-linearity that is applicable to complex inputs. More importantly,

4.2 Complex Activation Functions

The *modReLU* proposed by Arjovsky *et al.*in [1], defined as

$$\sigma_{\text{modReLU}}(z) = \text{Relu}(\|z\| + b)\frac{z}{\|z\|}. \tag{7}$$

As noted by Arjovsky, this *is* the case for the standard rectified linear unit or ReLU. However, the standard ReLU is defined only if the pre-activation is real *i.e.*$z_k^{(j)} \in \mathbb{R}$. for a complex pre-activation, one needs a specially defined non-linearity applicable to complex inputs. Arjovsky define a modRelu, however, . With a complex pre-activation $z_k^{(j)} \in \mathbb{C}$, it is complex[2], as is proposed σ must be a bounded holomorph function However, this due to liouville theroem, cannot be both analytic and bounded; resort to using an analytic version (always differentiable), i.e. complex tanh / complex sigmoid, then we end up with singularities and lose on the boundedness; may be difficult to avoid whilst learning; alternative, settle for approximations without being analytic i.e. go for partial derivatives not defined everywhere; stay for bounded modReLU is neither analytic nor bounded but if we stay mostly in the linear range, then won't have problems (check values in arjovsky's experiment results to verify)

4.2.1 Stiefel-manifold optimization

The group of unitary/orthogonal matrices is not closed under addition. Using orthogonal initialization and standard addition based gradient descent leads to loss orthogonality. In order to overcome this problem Wisdom et al.[26] uses the Stiefel-Manifold optimization scheme described in[21]. Tagare proposes to use a gradient, which points along the Stiefel-Manifold of unitary matrices, which itself unitary. This gradient may be computed using [21]:

$$\mathbf{A} = \mathbf{W}\mathbf{G}^H - \mathbf{W}^H\mathbf{G} \tag{8}$$

$$Y^k(\lambda) = (\mathbf{I} + \frac{\lambda}{2}\mathbf{A}^k)^{-1}(\mathbf{I} - \frac{\lambda}{2}\mathbf{A}^k)\mathbf{W}^k \tag{9}$$

With the resulting weight update given by $\mathbf{W}^{k+1} = Y^k(\lambda)$.

[2] the only case it can be real with complex $\mathbf{W}$ is if h_t is the complex conjugate.

4.3 Complex Gating

desirable properties for complex gate (1) we want a $\mathbb{C} \rightarrow \mathbb{R}$ mapping for the gates because we want to preserve (not modify) the phase (cite Arjovsky, also show experimentation, showing those which change the phase and those which do not) (2) have a non-linearity, so as to get a bounded gate value (from 0 to 1); non-linearity is good for learning approximation –> which non-linearity is good? show different baselines (3) including the gates should still conserve the overall stability of the architecture (distribution over a sum is a no-no –> LSTM distributes over a sum; compare with baseline which the gates are not constrained in any way)

5 Complex gated recurrent units

5.1 The gated unitary evolution network

$$\mathbf{i}_g = \sigma(\mathbf{W}_i[\Re(\mathbf{x})\ \Im(\mathbf{x})]^T), \tag{10}$$

$$\mathbf{f}_g = \sigma(\mathbf{W}_f[\Re(\mathbf{x})\ \Im(\mathbf{x})]^T), \tag{11}$$

$$\mathbf{h}_{t+1} = \mathbf{U}_h f(\mathbf{f}_g \odot \mathbf{h}_t) + \mathbf{W}_x(\mathbf{i}_g \odot \mathbf{x}). \tag{12}$$

For notational brevity bias terms are omitted. $\mathbf{U}_h$ denotes a unitary matrix and $\mathbf{i}_g, \mathbf{f}_g$ are computed by mappings from $\mathbb{C} \rightarrow \mathbb{R}$. Our gates are therefore non-holomorph. We leverage Wirtinger calculus [7][15][25], to define a pseudo-gradient, which we argue is sufficient to train the gates. Because $\mathbf{U}_h$ is unitary as in [1], we do not have to distribute our derivatives over a sum, a major difference with respect to the classical formulation in [12]. Because gate output is independent of the cell state $\mathbf{h}$, the proof in [1] holds without modification since $\sigma(\cdot) \in [0, 1]$.

5.2 The complex gated recurrent unit

Setting up complex gating mechanisms is no trivial task, because functions from $\mathbb{C} \rightarrow \mathbb{R}$ cannot be holomorph unless they are constant [3, page 9][3]. Furthermore bounded holomorph complex functions must be constant [3, page 38][4]. Classic multiplication gates with $0 \leq |f(x)| \leq 1$ and $\mathbb{C} \rightarrow \mathbb{R}$ which rely on $f(x) \cdot h$ are therefore hard to implement, because there is no obvious complex gradient to train these gates.

In order to take the classic GRU to the complex domain, while inheriting as

[3] Proof: https://math.stackexchange.com/questions/1004672/prove-that-a-real-valued-constant-function-is-holomorphic-and-vice-versa

[4] https://en.wikipedia.org/wiki/Liouville%27s_theorem_(complex_analysis)

many of the favorable properties from [1] as possible we define:

$$\mathbf{g} = \mathbf{O}_g\mathbf{h} + \mathbf{W}_g\mathbf{x} + \mathbf{b}_g, \tag{13}$$

$$\mathbf{z} = \sigma(\Re(\mathbf{g})), \tag{14}$$

$$\mathbf{r} = \sigma(\Im(\mathbf{g})), \tag{15}$$

$$\overline{\mathbf{h}}_t = f(\mathbf{W}_x\mathbf{x}_t + \mathbf{U}_h(\mathbf{r} \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \tag{16}$$

$$\mathbf{h} = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \overline{\mathbf{h}_t} \tag{17}$$

With $\mathbf{U}_h$ defined as a unitary matrix. And $\mathbf{O}$ defined as split orthogonal, which means we define $\Re(\mathbf{O})^T\Re(\mathbf{O}) = \mathbf{I} = \Im(\mathbf{O})^T\Im(\mathbf{O})$, a definition we motivate in the upcoming section. f denotes the non-linearity. Please note that since $\mathbf{r}$ and $\mathbf{f}$ are real vectors. Multpilication with these vectors therefore changes only the magnitude, but leaves the phase unchanged. A major difference with respect to real valued memory cells. This definition allows the model to scale data points according to their current relevance, while leaving phase information untouched. For $\mathbf{z}, \mathbf{r} = \mathbf{1}$ the equations above simplify to the formulation proposed in [1]. Previous work has shown that initializing memory gates to be open is beneficial **(MW: TODO: Find citations)**.

5.3 Model stability

For $\mathbf{z} \neq \mathbf{1}$ derivatives will distribute over a sum, forming a constant error carrousel as described in [12]. Choosing the gate bias to be large we initially have $\mathbf{z} \approx \mathbf{1}$ and a situation similar to [1] considering their formulation, while taking into account the added reset gate we obtain:

$$\frac{\partial C}{\partial \mathbf{h}_t} = \frac{\partial C}{\partial \mathbf{h}_T}\frac{\partial \mathbf{h}_T}{\partial \mathbf{h}_t}, \tag{18}$$

$$= \frac{\partial C}{\partial \mathbf{h}_T}\prod_{k=t}^{T-1}\frac{\partial \mathbf{h}_{k+1}}{\partial \mathbf{h}_k}, \tag{19}$$

$$= \frac{\partial C}{\partial \mathbf{h}_T}\prod_{k=t}^{T-1}\mathbf{U}_h\mathbf{G}_{k+1}\mathbf{R}_{k+1} \tag{20}$$

With $\mathbf{U}$ unitary and $\mathbf{G} = \text{diag}(f'(\mathbf{r} \odot \mathbf{h}_t))$. The matrix $\mathbf{R}$ and $\mathbf{Z}$ represent the change introduced by the gates. The change in gradient dynamics introduced by the reset gate is therefore governed by $\partial\mathbf{r}/\partial\mathbf{h}_k$ as well as $\partial\mathbf{z}/\partial\mathbf{h}_k$. Wirtinger-calculus tells us that [19][page 55],[16][page 61, eq 5.14]:

$$\frac{\partial \mathbf{r}}{\partial \mathbf{h}_k} = \frac{\partial \mathbf{r}}{\partial \mathbf{g}_k}\frac{\partial \mathbf{g}_k}{\partial \mathbf{h}_k} + \frac{\partial \mathbf{r}}{\partial \overline{\mathbf{g}_k}}\frac{\partial \overline{\mathbf{g}_k}}{\partial \mathbf{h}_k} \qquad \text{Wirtinger chain rule} \tag{21}$$

$$= \frac{1}{2i}\mathrm{diag}(\sigma'(\frac{\mathbf{g}-\overline{\mathbf{g}}}{2i}))\mathbf{O}_g - \frac{1}{2i}\mathrm{diag}(\sigma'(\frac{\mathbf{g}-\overline{\mathbf{g}}}{2i}))\overline{\mathbf{O}}_g \tag{22}$$

$$= \mathbf{R}(\frac{\mathbf{O}_g - \overline{\mathbf{U}}_g}{2i}) \qquad \mathbf{R} = \mathrm{diag}(\sigma'(\frac{\mathbf{g}-\overline{\mathbf{g}}}{2i})) \tag{23}$$

$$= \mathbf{R}\Im(\mathbf{O}_g) \tag{24}$$

$$\frac{\partial \mathbf{z}}{\partial \mathbf{h}_k} = \frac{\partial \mathbf{z}}{\partial \mathbf{g}_k}\frac{\partial \mathbf{g}_k}{\partial \mathbf{h}_k} + \frac{\partial \mathbf{z}}{\partial \overline{\mathbf{g}_k}}\frac{\partial \overline{\mathbf{g}_k}}{\partial \mathbf{h}_k} \tag{25}$$

$$= \frac{1}{2}\mathrm{diag}(\sigma'(\frac{\mathbf{g}+\overline{\mathbf{g}}}{2}))\mathbf{O}_g + \frac{1}{2}\mathrm{diag}(\sigma'(\frac{\mathbf{g}+\overline{\mathbf{g}}}{2}))\overline{\mathbf{O}}_g \tag{26}$$

$$= \mathbf{Z}(\frac{\mathbf{O}_g + \overline{\mathbf{U}}_g}{2}) \qquad \mathbf{Z} = \mathrm{diag}(\sigma'(\frac{\mathbf{g}+\overline{\mathbf{g}}}{2})) \tag{27}$$

$$= \mathbf{Z}\Re(\mathbf{O}_g) \tag{28}$$

$$\tag{29}$$

Taking the norm we obtain:

$$|\frac{\partial \mathbf{r}}{\partial \mathbf{h}_k}| = |\mathbf{R}|\ |\Im(\mathbf{O}_g)| \tag{30}$$

$$|\frac{\partial \mathbf{z}}{\partial \mathbf{h}_k}| = |\mathbf{Z}\ |\Re(\mathbf{O}_g)| \tag{31}$$

Constraining $\mathbf{O}_g$ to have orthogonal real and imaginary part, $|\Im(\mathbf{O}_g)| = |\Re(\mathbf{O}_g)| = 1$ and going back to the cost function term 18, considering it's norm leads to:

$$|\frac{\partial C}{\partial \mathbf{h}_t}| = |\frac{\partial C}{\partial \mathbf{h}_T}| \prod_{k=t}^{T-1} |\mathbf{G}_{k+1}| \cdot |\mathbf{R}_{k+1}|. \tag{32}$$

With $\mathbf{G} = \mathrm{diag}(f'(\mathbf{r} \odot \mathbf{h}_t))$ and $\mathbf{R} = \mathrm{diag}(\sigma'(\frac{\mathbf{g}+\overline{\mathbf{g}}}{2}))$. We desire $|\frac{\partial C}{\partial h_t}| = |\frac{\partial C}{\partial h_T}|$, which would hold only if we could guarantee $|\mathbf{G}_{k+1}\mathbf{R}_{k+1}| = 1$. $\mathbf{R}$ is populated with values drawn from a bell-shaped curve, we therefore expect $|\mathbf{R}| \leq 1$ **(MW: The experiments with the gate-relu failed, because gate gradients vanish after a while. Here I am talking about sigmoid(4x))**. To prevent gradients from vanishing for longer time sequences, equation 17 introduces a constant error carousel as described in [12]. We argue that our hybrid approach inherits some of the capabilities of unitary evolution networks by bounding the update in equation in 17, while at the same time gaining the noise resistance that comes with gated memory management.

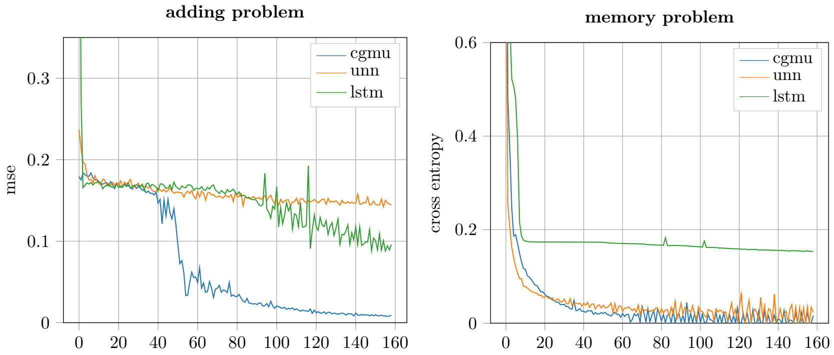

Figure 1: Performance of the complex gated memory unit (cgmu, ours), the unitary neural network (unn, [1]), and long short term memory (lstm, [12]), T=100 unitary U, free O, no GRU equations.

5.4 Results

To test our ideas we used the benchmark originally proposed in [12] following the implementation of [1]. A visualization of our results is shown in figure 1. All modes where run with a state size of 512, step size of 0.001 and a batch size of 250. For the memory problem, the baseline is at 0.173, which all models beat. For the adding problem it is 0.167. Again all models crack this treshhold. However the convergence behaviour differes. We argue that our apprach gets the bost of both worlds in terms of performance an converges well on both the adding and memory problems, it shows the supereor UNN dynamics on the memory problem, while at the same time behaving more like the LSTM on the adding problem, where it significantly outperforms both other models.

5.5 Computational Expense

Experimental baseline to show why complex networks are better than their counterpart with same number of params

6 Conclusion

TODO.

7 Other Ideas

7.1 Fourier Space rotations.

Earlier work has found that rotations can be implemented in the frequency domain by shearing along the two dimensions. Making use of the DFTs shift

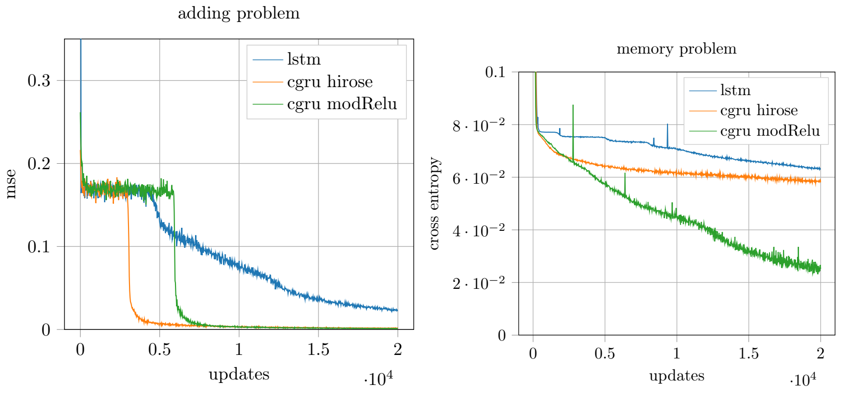

Figure 2: T=250, unitary U, free O, GRU, nlstm=128, ncgru=48

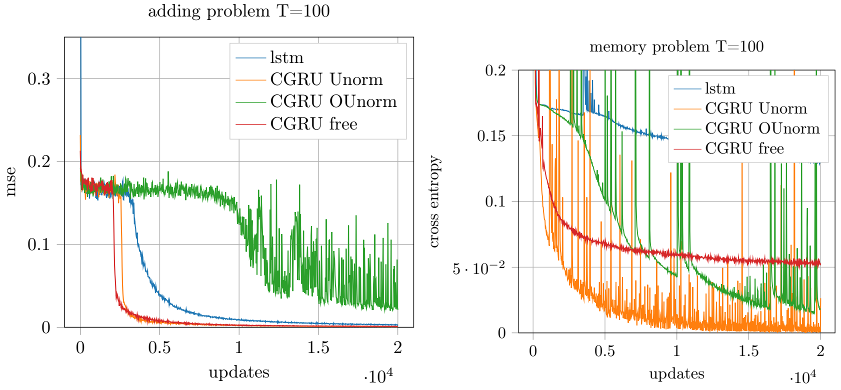

Figure 3: T=100, orthogonal unitary comparison, mod relu for memory, hirose for adding, GRU equation. nlstm=128, ncgru=48

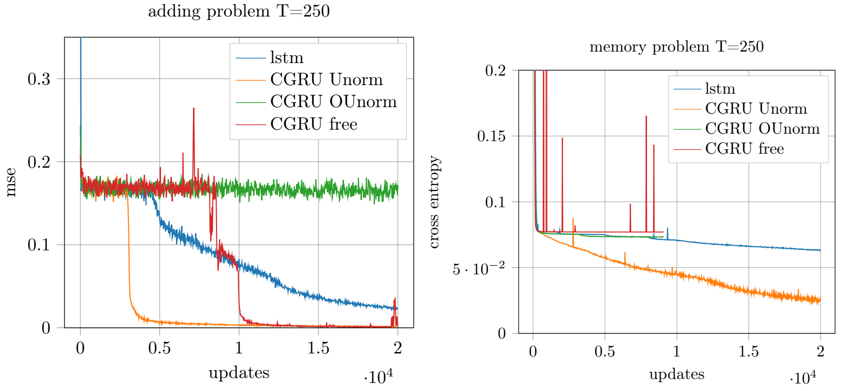

Figure 4: T=250, orthogonal unitary comparison, mod relu for memory, hirose for adding, GRU equation. nlstm=128, ncgru=48

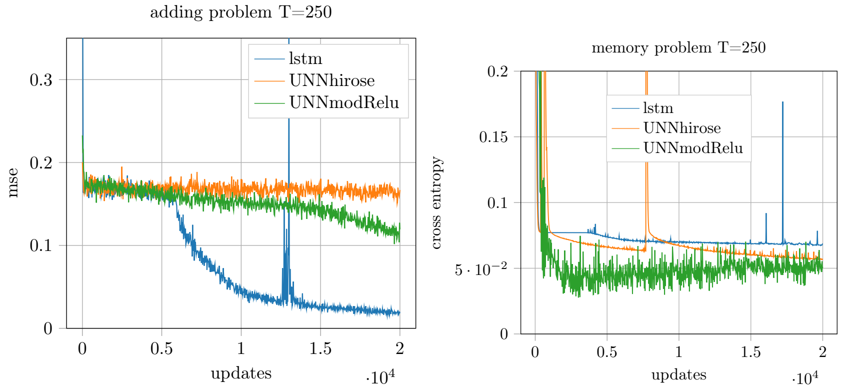

Figure 5: T=250, unitary evolution RNN proposed by wisdom et al. vs. LSTM nlstm=128, nUNN=128

theorem [5, page 173]:[5] [6]

$$\mathcal{D}(f_{m+m_0,n+n_0}) = \omega_M^{-m_0 j} \omega_N^{-n_0 k} F_{jk} \tag{33}$$

$$\text{with } \omega_N^{nk} = e^{i2\pi nk/N} \tag{34}$$

Transformation to the frequency domain multiplication, rotation and inverse transformation, can be implemented using three matrix multiplications, when working with the DFT or as FFT, multiplication and ifft. The inverse transformation is a way to implicitly apply trigonometric interpolation[7]. Which takes care of interpolating the pixel values of the new rotated image.

Some first numerical evidence suggests that fourier rotation matrices are unitary. This could allow us to prove stability. TODO: Proof?

7.2 Unitary dynamic filter networks

Motivation: Current dynamic RNNs do not worry about stability.
Idea: Adapt RNN stability theory to come up with stable dynamic RNNs.
Extra motivation: I think that the steerable filter paper [8] was the basis for the original dynamic filter paper. Which is why I think the fourier extension of this paper [17] could hold some cues for a nice extension. In particular, because outside of the vision domain, [13] has already shown that this is an interesting idea.

[5] `http://www.nontrivialzeros.net/KGL_Papers/27_Rotation_Paper_1997_qualityscan_OCR.pdf`

[6] `http://bigwww.epfl.ch/publications/unser9502.pdf`

[7] `https://en.wikipedia.org/wiki/Trigonometric_interpolation`

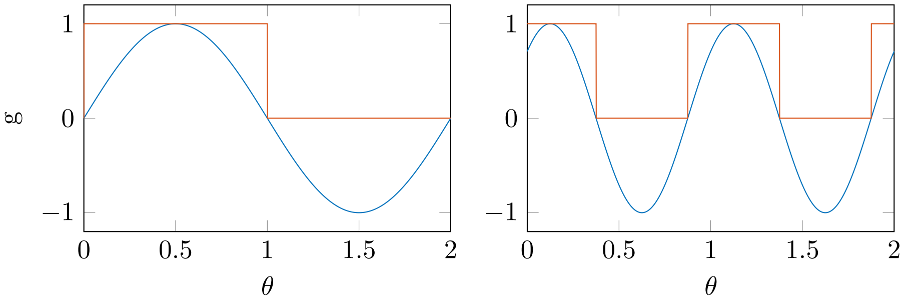

Figure 6: Plot of the $\sin(\theta a\pi + b\pi)$ and $\mathrm{H}(\sin(\theta a\pi + b\pi))$ with $a = 1, b = 0$ (left) and $a = 2, b = 0.1$

8 Background

8.1 Phase-Relus

Holomorph functions $f(x, y) = u(x, y) + iv(x, y)$ must satisfy the Cauchy-Riemann equations:

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial y} \text{ and } \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \tag{35}$$

This form has been considered in [22] and used to evalute existing non-linearities such as the zRelu, cRelu or Mod-Relu. However we believe it is much more intuitive to consider the Cauchy-Riemann equations in polar form [8]:

$$\frac{\partial u}{\partial r} = \frac{1}{r}\frac{\partial v}{\partial \theta} \text{ and } \frac{\partial v}{\partial r} = -\frac{1}{r}\frac{\partial u}{\partial \theta} \tag{36}$$

Which allows us to design a non-linearity using $z = re^{i\theta}$ and $f(r, \theta) = u(r, \theta) + iv(r, \theta)$. We will focus on non-linearities of the form:

$$\begin{aligned} f(r, \theta) &= g(r, \theta, a, b)e^{i\theta} && (37) \\ &= g(r, \theta, a, b)\cos(\theta) + ig(r, \theta, a, b)\sin(\theta) && (38) \end{aligned}$$

With a, b as lernable function parameters. Setting $g(r, \theta, a, b)$ to:

$$g(r, \theta, a, b) = r\mathrm{H}(\sin(\theta \cdot a\pi + b)) \tag{39}$$

With H denoting the Heaviside step function and $a, b \in \mathbb{R}$. Leads to the condi-

[8] Proof see: `https://math.stackexchange.com/questions/1245754/cauchy-riemann-equations-in-polar-form`

tions:

$$\frac{\partial u}{\partial r} = \mathrm{H}(\sin(\theta \cdot a\pi + b)) \cos\theta, \tag{40}$$

$$\frac{\partial v}{\partial r} = \mathrm{H}(\sin(\theta \cdot a\pi + b)) \sin\theta, \tag{41}$$

$$\begin{aligned}\frac{\partial u}{\partial \theta} &= -r\mathrm{H}(\sin(\theta \cdot a\pi + b)) \sin\theta \\ &\quad + r\delta(\sin(\theta \cdot a\pi + b)) \cos(\theta \cdot a\pi + b)) a\pi \cos(\theta)\end{aligned} \tag{42}$$

$$\begin{aligned}\frac{\partial v}{\partial \theta} &= r\mathrm{H}(\sin(\theta \cdot a\pi + b)) \cos\theta \\ &\quad + r\delta(\sin(\theta \cdot a\pi + b)) \cos(\theta \cdot a\pi + b)) a\pi \sin(\theta)\end{aligned} \tag{43}$$

Above δ denotes Dirac's distribution, which we consider to be zero for all practical purposes. We therefore argue that this non-linearity which we call Polar-Relu is approximately holomorph[9].
$\mathrm{H}(\sin(\theta \cdot a\pi + b)$ sets the output to zero, whenever $\sin(\theta \cdot a\pi + b) < 0$. We must have $\theta \in [0, 2\pi]$. This means for $a = 1, b = 0$ this non-linearity removes the lower-half of the complex plane with $\theta > \pi$ where $\Re(z) < 0$. When keeping $b = 0$, for $0.5 < |a| < 1$ the filtered spectrum is reduced, and for $|a| < 0.5$, no values are filtered. Working with $|a| > 1$ introduces periodically spaced smaller filters. Because the sine wave will complete more than one iteration for θ. Finally b rotates the filter around the origin, this parameter enables layered phase relus to individually remove different areas of the complex plane.
An interesting variant of this approach can be created by adding a cosine term to equation 39:

$$g(r, \theta, a, b, c, d) = r\mathrm{H}(\sin(\theta \cdot a\pi + b))\mathrm{H}(\cos(\theta \cdot c\pi + d)) \tag{44}$$

This will kill any incoming complex number with a phase angle of either zero sine or cosine. The above equation can be considered a generalization of the zRelu from [11][22]. Its is equivalent for $a = 1, b = 0, c = 1, d = 0$ because both cosine and sine are positive in the first quadrant. This approach works when choosing the function parameters manually. Unfortunately, the same mechanics, that makes this approach approximately holomorph also kills the derivative, which one would want to use to train the function parameters.

8.1.1 Phase-Relu approximate stability proof in Cartesian coordinates.

We have shown that

$$f(r, \theta) = r\mathrm{H}\sin(\theta \cdot a\pi + b)e^{i\theta} \tag{45}$$

[9] Strictly speaking it is holomorph, when excluding all points where $\sin(\theta \cdot a\pi + b) = 0$.

is stable in polar coordinates. For the extremely skeptical reader we will now show that its equivalent form:

$$f(x+iy) = \mathrm{H}(\sin(\mathrm{atan2}(x,y)))(x+iy) \tag{46}$$

is stable in Cartesian coordinates. Splitting the above formulation into real and imaginary parts leads to:

$$f(x+iy) = x\mathrm{H}(\sin(\mathrm{atan2}(x,y))) + iy\mathrm{H}(\sin(\mathrm{atan2}(y/x))) \tag{47}$$

We recognize the form $f(z) = u + iv$. Working with the unchanged Cauchy-Riemann equations:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}, \tag{48}$$

and using the facts that the derivatives of $\mathrm{atan2}(x,y)$ are equal to those of $\tan^{-1}(y/x)$ which are $\frac{\partial \tan^{-1}(y/x)}{\partial x} = -\frac{y}{x^2+y^2}$ and $\frac{\partial \tan^{-1}(y/x)}{\partial y} = \frac{x}{x^2+y^2}$, we derive:

$$\begin{aligned}\frac{\partial u}{\partial x} &= \mathrm{H}(\sin(\tan^{-1}(y/x))) \\ &\quad + x\delta(\sin(\tan^{-1}(y/x)))\cos(\tan^{-1}(y/x))(\frac{-y}{x^2+y^2})(\frac{-y}{x^2}) \qquad (49)\\ \frac{\partial u}{\partial y} &= \delta(\sin(\tan^{-1}(y/x)))\cos(\tan^{-1}(y/x))(\frac{x}{x^2+y^2}) \qquad (50)\\ \frac{\partial v}{\partial x} &= y\delta(\sin(\tan^{-1}(y/x)))\cos(\tan^{-1}(y/x))(\frac{-y}{x^2+y^2})(\frac{-y}{x^2}) \qquad (51)\\ \frac{\partial v}{\partial y} &= \mathrm{H}(\sin(\tan^{-1}(y/x))) \\ &\quad + y\delta(\sin(\tan^{-1}(y/x)))\cos(\tan^{-1}(y/x))(\frac{x}{x^2+y^2})(\frac{1}{x}) \qquad (52)\end{aligned}$$

Most of the time the Dirac terms $\delta(\cdot)$ will be zero and $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = \mathrm{H}(\sin(\tan^{-1}(y/x)))$ as well as $\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0$ will hold. The derivative is zero if the non-linearity is inactive and 1 when its active and therefore bounded.

8.2 Analysis of existing complex activation functions.

This section is dedicated to the analysis of previously proposed non-linearities and relies on using the Cauchy-Riemann equations given by:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}, \tag{53}$$

for $f(z) = u(x,y) + iv(x,y)$ or if $f(r,\theta) = u(r,\theta) + iv(r,\theta)$, we make use of:

$$\frac{\partial u}{\partial r} = \frac{1}{r}\frac{\partial v}{\partial \theta} \text{ and } \frac{\partial v}{\partial r} = -\frac{1}{r}\frac{\partial u}{\partial \theta} \tag{54}$$

which is equivalent.

8.2.1 zRelu

$$\text{zRelu}(z) = \begin{cases} z \text{ if } \theta \in [0, \pi/2], \\ 0 \text{ else.} \end{cases} \tag{55}$$

Following [22] we have for the first quadrant:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 1, \tag{56}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0, \tag{57}$$

and elsewhere:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 0, \tag{58}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0, \tag{59}$$

holds. On the real and imaginary axes, the two areas are not smoothly connected, which is why we must include them. This argument may be backed up by considering the equivalent Phase-Relu formulation as defined in section 8.1.1:

$$\text{zRelu}(z) = \text{H}(\sin(\text{atan2}(x, y)))\text{H}(\cos(\text{atan2}(x, y)))(x + iy) \tag{60}$$

Which is an equivalent way to write the zRelu, its Dirac pulse derivatives are one on the real and imaginary axis, which is why the this non-linearity is not holomorph there. The derivative is either zero or one and therefore bounded. These desirable properties come at the cost of having to throw away three quarters of the complex plane, which seems unnecessarily wasteful.

8.2.2 modRelu

The modRelu is defined as [1]:

$$f(z) = \text{Relu}(\|z\| + b)\frac{z}{\|z\|}. \tag{61}$$

Conversion to polar coordinates yields:

$$f(r, \theta) = \text{Relu}(r + b)e^{i\theta}, \tag{62}$$

$$f(r, \theta) = \text{Relu}(r + b)\cos(\theta) + i\text{Relu}(r + b)\sin(\theta). \tag{63}$$

$$\tag{64}$$

we find $u(r,\theta) = \mathrm{Relu}(r+b)\cos(\theta)$ and $v(r,\theta) = \mathrm{Relu}(r+b)\sin(\theta)$. The polar Cauchy-Riemann equations yield:

$$\frac{\partial u}{\partial r} = \mathrm{H}(r+b)\cos(\theta), \tag{65}$$

$$\frac{\partial u}{\partial \theta} = -\mathrm{Relu}(r+b)\sin(\theta), \tag{66}$$

$$\frac{\partial v}{\partial r} = \mathrm{H}(r+b)\sin(\theta), \tag{67}$$

$$\frac{\partial v}{\partial \theta} = \mathrm{Relu}(r+b)\cos(\theta). \tag{68}$$

For holomorphy we require:

$$\frac{\partial u}{\partial r} = \frac{1}{r}\frac{\partial v}{\partial \theta}, \tag{69}$$

$$\Leftrightarrow r\mathrm{H}(r+b)\cos(\theta) = \mathrm{Relu}(r+b)\cos(\theta); \tag{70}$$

$$\frac{\partial v}{\partial r} = -\frac{1}{r}\frac{\partial u}{\partial \theta}, \tag{71}$$

$$\Leftrightarrow r\mathrm{H}(r+b)\sin(\theta) = \mathrm{Relu}(r+b)\sin(\theta). \tag{72}$$

Taking into account the fact that $r\mathrm{H}(r) = \mathrm{Relu}(r)$ we have $r\mathrm{H}(r+b) \approx \mathrm{Relu}(r+b)$ if $b \approx 0$. The modRelu non-linearity is therefore only holomorph when it is approximately linear, not a useful property. Furthermore we find:

$$\frac{\partial}{\partial z}\sigma_{\mathrm{Relu}}(\|z\|+b)\frac{z}{\|z\|} = \sigma'_{\mathrm{Relu}}(\|z\|+b)\frac{z}{\|z\|} + \sigma_{\mathrm{Relu}}(\|z\|+b)(\frac{z}{\|z\|})'. \tag{73}$$

By applying the product rule. The left part of the resulting sum is stable, but the right part is not bounded and therefore unstable, which is yet another undesirable property.

8.2.3 cRelu

[22] defines the cRelu as:

$$\mathrm{cRelu}(z) = \mathrm{Relu}(x) + i \cdot \mathrm{Relu}(y). \tag{74}$$

Thus $u = \mathrm{Relu}(x)$ and $v = \mathrm{Relu}(y)$. In the first quadrant we have:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 1, \tag{75}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0. \tag{76}$$

For the second we find:

$$\frac{\partial u}{\partial x} = 0 \neq \frac{\partial v}{\partial y} = 1, \tag{77}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0, \tag{78}$$

The third quadrant has:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 0, \tag{79}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0, \tag{80}$$

Finally considering the fourth:

$$\frac{\partial u}{\partial x} = 1 \neq \frac{\partial v}{\partial y} = 0, \tag{81}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0, \tag{82}$$

In its holomorph region the derivative of the function is one just like in the real case which is stable. We have shown this definition to be holomorph when $\text{sign}(\Re(z)) = \text{sign}(\Im(z))$[22], which is the case in the first an third quadrant. When restricting this definition to the first quadrant and setting it to zero elsewhere, one obtains the zRelu which is a holomorph function.

8.2.4 Cardioid

[24] introduces the complex cardioid,:

$$f(z) = \frac{1}{2}(1 + cos(\theta))z \tag{83}$$

Which we express in polar-coordinates as:

$$f(r, \theta) = \frac{1}{2}(1 + cos(\theta))re^{i\theta} \tag{84}$$

Using the definition of the complex exponential we obtain:

$$f(r, \theta) = \frac{1}{2}(1 + cos(\theta))r\cos(\theta) + i\frac{1}{2}(1 + cos(\theta))r\sin(\theta) \tag{85}$$

Reading of $u = \frac{1}{2}(1 + cos(\theta))r\cos(\theta)$ and $v = \frac{1}{2}(1 + cos(\theta))r\sin(\theta)$ we find:

$$\frac{\partial u}{\partial r} = \frac{1}{2}(1 + \cos(\theta))\cos(\theta) \tag{86}$$

$$\frac{\partial u}{\partial \theta} = -r\frac{1}{2}(1 + \cos(\theta))\sin(\theta) - r\frac{1}{2}\sin(\theta)\cos(\theta) \tag{87}$$

$$\frac{\partial v}{\partial r} = \frac{1}{2}(1 + \cos(\theta))\sin(\theta) \tag{88}$$

$$\frac{\partial v}{\partial \theta} = r\frac{1}{2}(1 + \cos(\theta))\cos(\theta)) - r\frac{1}{2}\sin(\theta)\cos(\theta) \tag{89}$$

And therefore we require:

$$\frac{\partial u}{\partial r} = \frac{1}{2}(1+\cos(\theta))\cos(\theta) = \frac{\partial v}{r\partial\theta} = \frac{1}{2}(1+\cos(\theta)\cos(\theta)) - r\frac{1}{2}\sin(\theta)\cos(\theta) \tag{90}$$

$$\Leftrightarrow 0 = -r\frac{1}{2}\sin(\theta)\cos(\theta) \tag{91}$$

$$\tag{92}$$

Which is holomorph at $r = 0$. When $r \neq 0$ excluding everything except for the real and imaginary axes is necessary, there the trigonometric functions are zero. Where the derivative exists it is unstable when $\cos(\theta) > 0$, which happens if $0 \leq \theta < \pi/2$ and $3/2\pi < \theta \leq 2\pi$. This is the case on the real axis where $\theta = 0$. In other words the Cardioid has a defined and stable derivative only on the imaginary axis.

8.2.5 Phase-amplitude activation

[20] cites these as:

$$f(z) = \tanh(r/m)e^{i\theta} \tag{93}$$

Considering the polar C.R. equations:

$$\begin{aligned}
f(r,\theta) &= \cos(\theta)\tan^{-1}(r/m) + i\sin(\theta)\tan^{-1}(r/m) && (94)\\
&\Rightarrow u + iv\\
\frac{\partial u}{\partial r} &= \cos(\theta)\text{sech}^2(r/m)(1/m) && (95)\\
\frac{\partial u}{\partial \theta} &= -\sin(\theta)\tanh(r/m) && (96)\\
\frac{\partial v}{\partial r} &= \sin(\theta)\text{sech}^2(r/m)(1/m) && (97)\\
\frac{\partial v}{\partial \theta} &= \cos(\theta)\tanh(r/m) && (98)\\
\Rightarrow \frac{\partial u}{\partial r} &= \frac{\partial v}{r\partial\theta}\\
\Leftrightarrow \qquad \cos(\theta)\text{sech}^2(r/m)(1/m) &= \frac{1}{r}\cos(\theta)\tanh(r/m) && (99)\\
\Leftrightarrow \qquad \text{sech}^2(r/m)(r/m) &= \tanh(r/m) && (100)
\end{aligned}$$

This activation is holomorph at $r/m = 0$. And approximately analytic for $r/m \approx 0$ as shown in figure 7, the problem here is that this non-linearity breaks magnitude information by rescaling them, it must therefore be non-holomorph for large inputs.

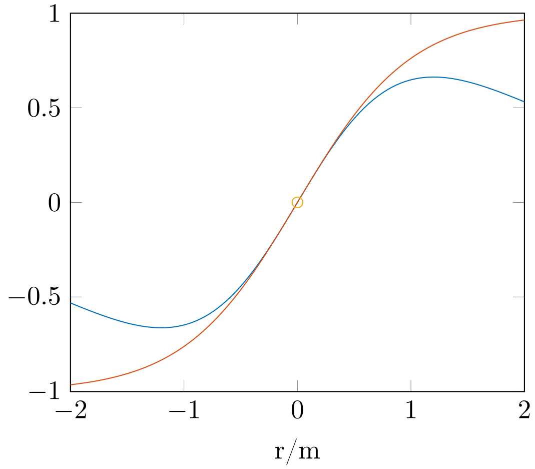

Figure 7: Plot of the holomorphy condition's two sides for the Phase-amplitude activation. $\tanh(r/m)$ is shown in red, $\operatorname{sech}^2(r/m)(r/m)$ is shown in blue. The yellow circle indicates the point at $(0, 0)$.

8.3 Backward mode automatic differentiation gradients

Consider the non-linear network proposed in [18]:

$$\mathbf{x}_t = W_{\text{rec}} f(\mathbf{x}_{t-1}) + W_{\text{in}} \mathbf{u}_t + \mathbf{b} \tag{101}$$

Following [18] we define the error function $\mathcal{E}_t = \mathcal{L}$ and work with BPTT gradients given by:

$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{1 \leq t \leq T} \frac{\mathcal{E}_t}{\partial \theta} \tag{102}$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{1 \leq k \leq t} \left(\frac{\mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\mathbf{x}_k} \frac{\partial^+ \mathbf{x}_k}{\partial \theta} \right) \tag{103}$$

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \geq i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \prod_{t \geq i > k} W_{\text{rec}}^T \operatorname{diag}(f'(\mathbf{x}_{i-1})) \tag{104}$$

The above equations are essential a consistent application of the chain rule. It is important to note that $\partial^+ \mathbf{x}_k / \partial W_{\text{rec}} = f(\mathbf{x}_{k-1})$.

8.3.1 The linear case

Working with $f(x) = x$ according the long term behavior is determined by the matrix product $\partial \mathbf{x}_t / \partial \mathbf{x}_k$ [18]. We define $W_{\text{rec}} = C$, and normalize the spectrum of C to $\forall k \; \|\phi_k\| = 1$ for $k \in \{1 \ldots n\}$. Focusing on the term in the sum of equation 103, we express $\partial\mathcal{E}/\partial\mathbf{x}_t$ in terms of a Fourier basis:

$$\frac{\partial\mathcal{E}}{\partial\mathbf{x}_t} = \sum_{i=1}^{n} \mathbf{f}_i^T d_i \tag{105}$$

Knowing the Fourier vectors are eigenvectors of C, which leads to $\mathbf{f}_i^T(C^T)^l = \phi_i^l \mathbf{f}_i^T$ therefore we have:

$$\frac{\partial\mathcal{E}}{\partial\mathbf{x}_t}\frac{\partial\mathbf{x}_t}{\partial\mathbf{x}_k} = \sum_{i=1}^{n} \mathbf{f}_i^T d_i \phi_i \tag{106}$$

Having chosen $\|\phi_k\| = 1$, we can approximate:

$$\frac{\partial\mathcal{E}}{\partial\mathbf{x}_t}\frac{\partial\mathbf{x}_t}{\partial\mathbf{x}_k} = \sum_{i=1}^{n} \mathbf{f}_i^T d_i \phi_i \approx \sum_{i=1}^{n} \mathbf{f}_i^T d_i = \frac{\partial\mathcal{E}}{\partial\mathbf{x}_t} \tag{107}$$

Because $\phi = \exp(i\omega)$ merely represents a rotation of the errors phase angle, but leaves its magnitude intact. Using the train of tought borrowed from [18], we claim to establish an error carousel similar to Hochreiter 1998, through which errors can pass in a stable manner.

8.3.2 Non-linear Cayley-Networks

We could employ a non-linearity based on the Cayley-Transform [3, p. 100]:

$$C'(z) = \frac{z-i}{z+i} \tag{108}$$

Which is guaranteed to map the upper half of the complex plane into the unit circle. Integrating $C(z)$ leads to;

$$C(z) = z - 2i\ln(z+i) \tag{109}$$

Which leads to a possible non-linearity for $\Re(z) > 0$. The unstable lower part of the complex plane where $\Re(z) < 0$, could be removed by defining:

$$D'(z) = \frac{z+i}{z-i} \tag{110}$$

And working with $D(z) = z + 2i\ln(z-i)$ where $\Re(z) < 0$. Working with this definition all z with $\Im(z) = 0$, would not be defined, because there is no smooth connection when crossing from $\Re(z) > 0$ to $\Re(z) < 0$ and the complex logarithm is not defined for all $\Re(z) < 0$ with $\Im(z) = 0$. Cayley transforms are known to be holomorph, which is a general property of all Möbius transforms.

$$C = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 \\ c_4 & c_1 & c_2 & c_3 \\ c_3 & c_4 & c_1 & c_2 \\ c_2 & c_3 & c_4 & c_1 \end{pmatrix}$$

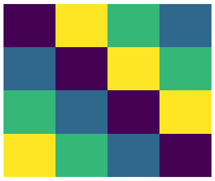

Figure 8: Circulant matrix structure as formula and in plotted form.

8.4 Convolutions and circulant matrices

One dimensional convolutions can be expressed as multiplication with a circulant matrix. The convolution operations used in neural networks may be expressed as matrix multiplication with doubly circulant matrices [9, page 324], doubly referring to a circulant block matrix consisting of circulant blocks. The eigen-decompositions of both cases seem to be well understood in the specialized mathematical literature[10].

8.5 1D-convolutions and circulant matrices

Consider for example the four by four circulant matrix $C = circ(c_1, c_2, c_3, c_4)$ as shown in figure 8. The matrix vector product $C\mathbf{x}$ with $\mathbf{x} = (x_1, x_2, x_3, x_4)^T$, can be written as:

$$c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \tag{111}$$
$$c_4x_1 + c_1x_2 + c_2x_3 + c_3x_4 \tag{112}$$
$$c_3x_1 + c_4x_2 + c_1x_3 + c_2x_4 \tag{113}$$
$$c_2x_1 + c_3x_2 + c_4x_3 + c_1x_4 \tag{114}$$

Above one can nicely see how the kernel moves over the one dimensional signal in x. If the wrapping effect is not desired the edges of x must be padded with zeros and parts of the circulant matrix be set to zero. For example $x_1, x_4 = 0$ and $c_3, c_4 = 0$, will remove the wrap-around.

8.6 The linear one-dimensional case

According to [10, page 33], the eigenbasis of all circulant matrices is given by:

$$\mathbf{f}^{(m)} = \frac{1}{\sqrt{n}}(1, \exp(-2\pi i m/n), \ldots, \exp(-2\pi i m(n-1)/n))' \tag{115}$$

For all m eigenvectors. Given a set of complex eigenvalues $\{\phi_m\}$, the corresponding circulant matrix can be computed using:

$$C = F\Phi F^{-1} \tag{116}$$

[10] http://nzjm.math.auckland.ac.nz/images/8/8e/18-36.pdf

For reasons which will become clear later we will express our eigenvalues in polar coordinates as:

$$\phi_m = r \exp i\omega \tag{117}$$

We propose the normalize network convolutions by setting their $r = 1$ for all convolutions and all eigenvalues and optimize only the eigenangles ω. Which amounts to requiring all convolution matrices to have eigenvalues located on the unit circle or equivalently, we enforce $\|\phi_m\| = 1$. To construct C we must transform all ϕ_ms to Cartesian coordinates using $x_m = \cos(\omega)$ and $y_m = \sin(\omega)$. When then place the Cartesian eigenvalues $x_m + iy_m$ on the diagonal of Φ. Next we can construct C from $C = U\Phi U^{-1}$. Where U is known and can be attached as a constant matrix to the computational graph. Furthermore the previous operations involve only trigonometric functions and matrix products, these operations are all differentiable, therefore we can find their gradient using standard AD tools. By running the optimization in the ω space we can enforce $\|\phi_m\| = 1$, without having to work with a constrained optimization algorithm.

8.7 Effects on linear network stability

8.7.1 Matrix power

In this section we will evaluate the effect of $\|\phi_m\| = 1$ on a linear one dimensional bias-free convnet consisting of n layers.

$$y = C_1 \cdot C_2 \dots C_n \cdot x \tag{118}$$

$$y = F\Phi_1 F^{-1} \cdot F\Phi_2 F^{-1} \cdot \dots \cdot F\Phi_n F^{-1} \cdot x \tag{119}$$

$$y = F\Phi_1 \cdot \Phi_2 \cdot \dots \cdot \Phi_n F^{-1} \cdot x \tag{120}$$

$$\tag{121}$$

All convolution eigenvalues will be of the form $\phi_{m,n} = \exp i\omega_{m,n}$, Φ amounts to element wise multiplication considering the rows therefore will lead to eigenvalues of:

$$\phi_m = \exp(i\omega_{m,1} + i\omega_{m,2} \dots i\omega_{m,n}) \tag{122}$$

For the equivalent one convolution network. We therefore claim that adding convolutions to this kind of eigenspace normalized linear network will add additional degrees of freedom to eigenspace rotations around the unit circle. Having set all $r_{m,n} = 1$ we claim to run a more stable network, because we only rotate, but do not rescale with additional layers. This convolution should remain stable when added to the recurrent convLSTM state update equation.
However to apply this idea to convNets in space the non-linearity needs to be taken care of.

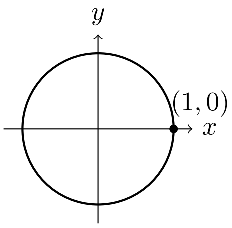

Figure 9: Illustration of the unit circle, on which we place all convolution eigenvalues $\phi = x + iy$.

8.7.2 Network conditioning

In linear algebra when solving $A\mathbf{x} = b$ or $\min_x \|Ax - b\|$ an important property is the condition number. It is a measure of the solutions sensitivity to small perturbations in $\mathbf{x}$. A problem is considered to be ill conditioned when A's assiciated condition number is very large. A problem's conditioning is measured using:

$$\kappa = \max_{i,j} |\frac{\phi_i}{\phi_j}| \tag{123}$$

In other words the matrix condition κ is the norm of the ratio of the largest and smallest eigenvalue. By enforcing $\|\phi\| = 1$ we also ensure a constant condition number of one for our convolution matrices. We hope to increase overall network stability this way, because our convolutions should not react very sensitively to small input perturbations.

8.8 Two dimensional convolutions

Discrete convolution is often described as sliding a kernel over an image. This operation may be expressed in terms of matrix-vector multiplication. For example the two dimensional convolution:

$$A * B = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} * \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \tag{124}$$

May be expressed using matrix multiplication as:

$$A * B = K^T \cdot B_{\text{flat}} \tag{125}$$

Where b_{flat} is a vector constructed by concatenation of B's rows. And the matrix K defined as:

$$K = \begin{pmatrix} a_1 & a_2 & 0 & a_3 & a_4 & 0 & 0 & 0 & 0 \\ 0 & a_1 & a_2 & 0 & a_3 & a_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & a_2 & 0 & a_3 & a_4 & 0 \\ 0 & 0 & 0 & 0 & a_1 & a_2 & 0 & a_3 & a_4 \end{pmatrix} \tag{126}$$

Matrix K, describes a convolution, but is not circulant.

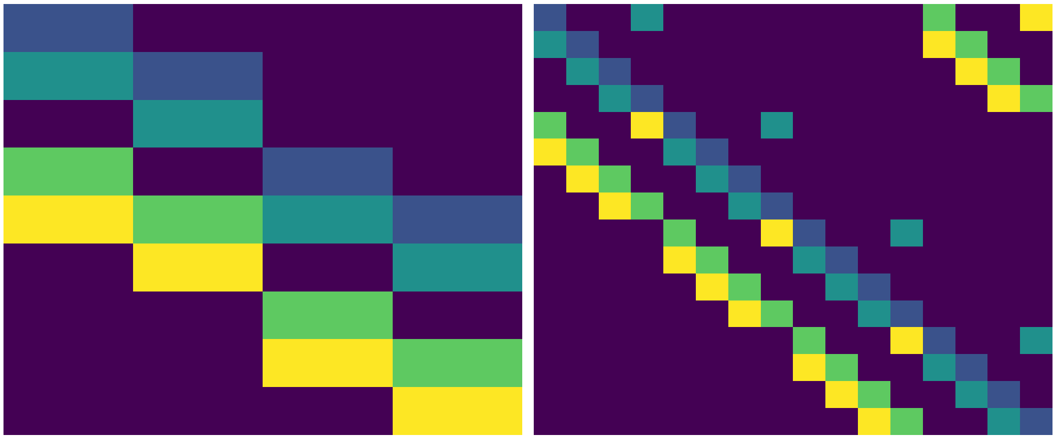

Figure 10: Visualization of a two dimensional convolution matrix and its square doubly circulant cousin.

8.8.1 Doubly block circulant matrices

In order to turn the convolution matrix into a square doubly circulant matrix, padding is required in both kernel and target matrix. A doubly circulant matrix is a block matrix consisting out of circulant blocks which are arranged in a circular pattern. In order to obtain circulant blocks the circular pattern must be finished, which is why the resulting matrix will be square by definition. Padding A and B leads to[11]:

$$A_p = \begin{pmatrix} a_1 & a_2 & 0 & 0 \\ a_3 & a_4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} B_p = \begin{pmatrix} b_1 & b_2 & 0 & 0 \\ b_3 & b_4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{127}$$

In this case the circular convolution matrix can be set up according to:

$$\begin{align} C_0 = circ(c_0) = \begin{pmatrix} a_1 & a_2 & 0 & 0 \end{pmatrix} \tag{128} \\ C_1 = circ(c_1) = \begin{pmatrix} a_2 & a_3 & 0 & 0 \end{pmatrix} \tag{129} \\ C_2 = circ(c_2) = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix} \tag{130} \\ C_3 = circ(c_3) = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix} \tag{131} \\ \tag{132} \end{align}$$

[11] I think its probably possible to come up with a less wasteful way to do the padding i.e. remove the second zero row and column.

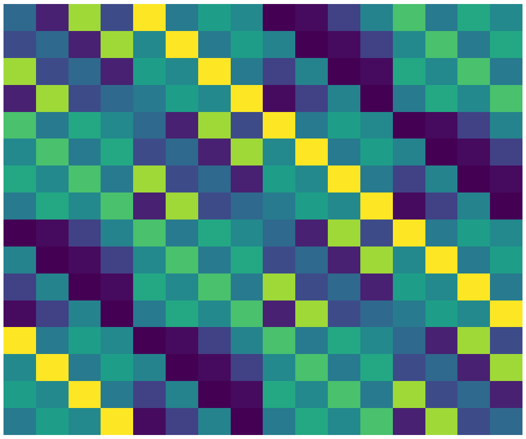

Figure 11: Absolute values of complex doubly block circulant matrix constructed in the frequency domain.

Which leads to the resulting matrix C:

$$C_b = \begin{pmatrix} C_0 & C_1 & C_2 & C_3 \\ C_3 & C_0 & C_1 & C_2 \\ C_2 & C_3 & C_0 & C_2 \\ C_1 & C_2 & C_3 & C_0 \end{pmatrix} \tag{133}$$

A visualization of this matrix is shown in figure 10 on the right. Multiplication of $C_b \cdot B_{p \text{ flat}}$ will lead to a zero padded version of $K^T \cdot B_{\text{flat}}$.

8.8.2 Doubly block circulant matrices and their eigenvalues

In order to be able to enforce $\|\phi\| = 1$. We would like to be able to construct doubly block circulant matrices in their eigenspace. According to [6, page 185], their diagonalization is given by:

$$C_b = \overline{(F_m \otimes F_n)^T} \Lambda (F_m \otimes F_n) \tag{134}$$

F_m and F_n denote fourier matrices. Λ has complex eigenvalues sitting on its diagonal. Their choice determines the block structure of the resulting matrix, which will be square with m block containing n rows each. Given a real input matrix we can find Lambda from:

$$\Lambda = (F_m \otimes F_n) C_b \overline{(F_m \otimes F_n)^T} \tag{135}$$

We believe that 134 is differentiable and should enable use to construct and optimize doubly block circulant matrices in the frequency domain. In order to ensure a real valued output Λ must be symmetric with respect to the real axis.

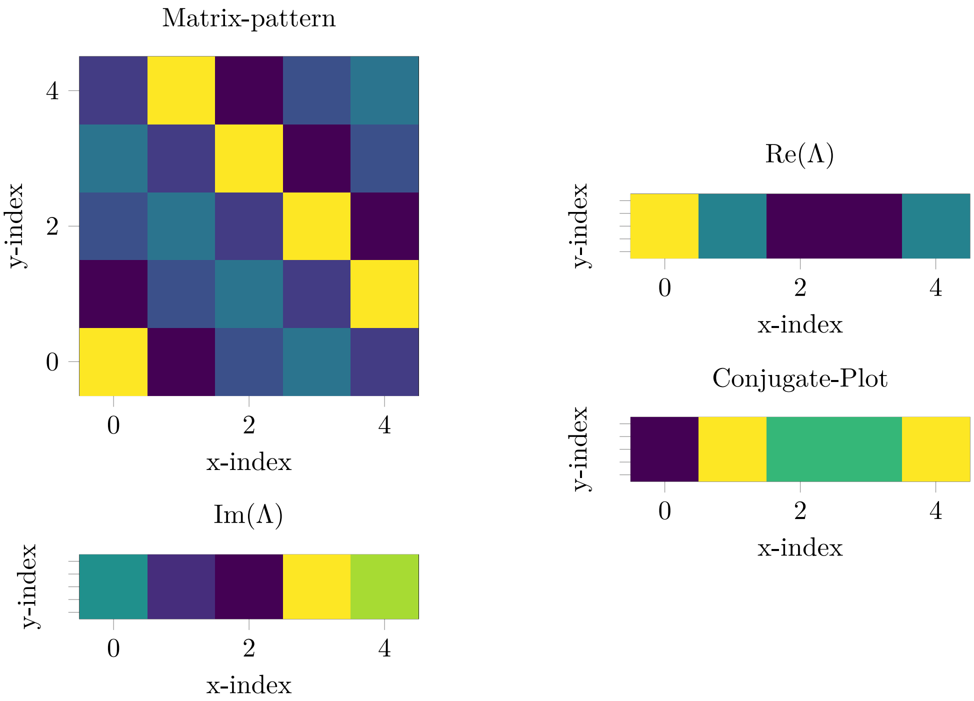

Figure 12: Circulant matrix pattern and spectrum.

8.8.3 The spectrum of real doubly block circulant matrices.

Tricky because doubly block circulants are not also circulant. So we cannot simply apply the one dimensional insight gained from working with circulants to block circulants. However block circulant spectra are point-symmetric in tow dimensions. Figures 12 and 13 illustrate this. The spectrum shown in 12 is symmetric along the a-Axis, when cutting after its third element and disregarding the first eigen-vale. The block circulant case shown in 12, we find the same symmetry in the zeroth column and 4th row. The middle block is point symmetric.

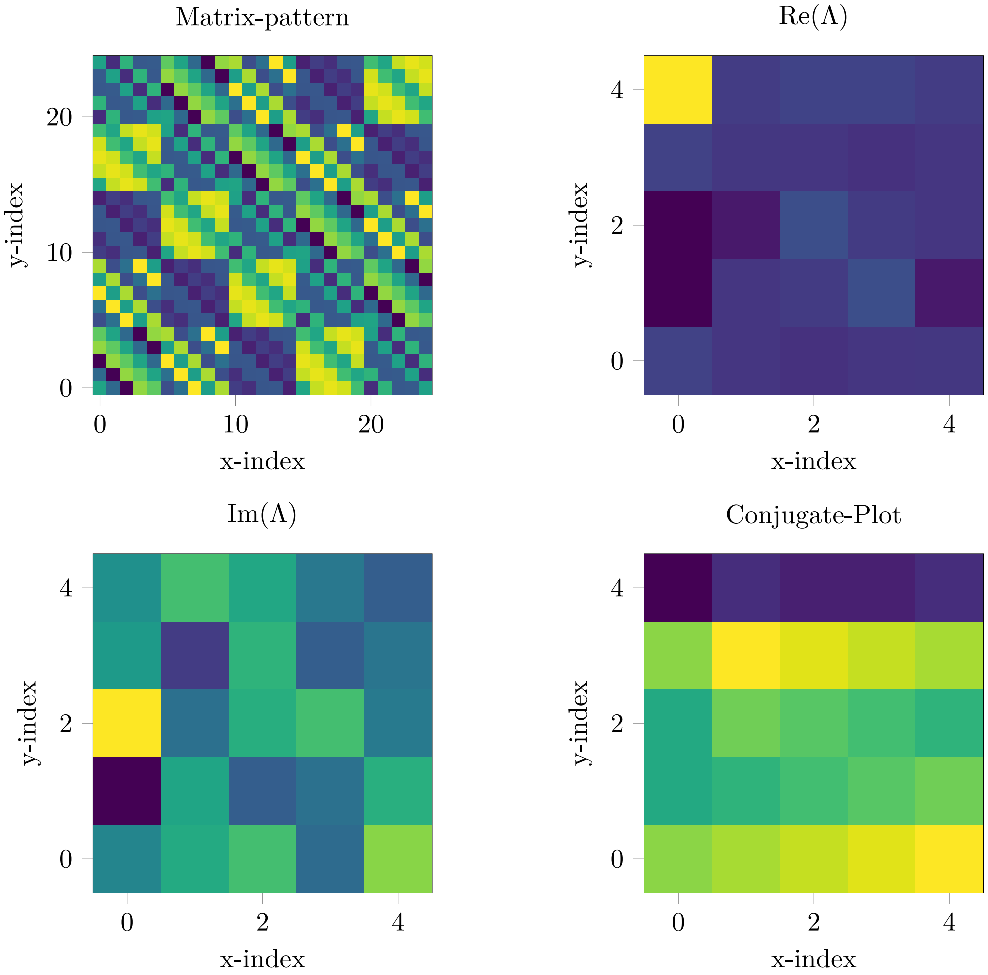

Figure 13: Block Circulant matrix pattern and spectrum.

9 Other related ideas

9.1 Rotation-GRU in $\mathbb{R}$

This section proposes the rotation-GRU, a modified version of the conv-GRU, which builds on the theory above. Recall the conv-GRU definition:

$$Z_t = \sigma(W_{xz} * X_t + W_{hz} * H_{t-1} + b_z), \tag{136}$$
$$R_t = \sigma(W_{xr} * X_t + W_{hr} * H_{t-1} + b_r), \tag{137}$$
$$H'_t = f(W_{xr} * X_t) + R_t \circ (W_{hp} * H_{t-1}), \tag{138}$$
$$H_t = (1 - Z_t) \circ H'_t + Z_t \circ H_{t-1}. \tag{139}$$

When optimizing the convolutions, while enforcing $\|\phi\| = 1$, changing the state update equation H_t to:

$$H_t = W_h * ((1 - Z_t) \circ H'_t + Z_t \circ H_{t-1}). \tag{140}$$

Assuming that the gates Z_t and R_t keep the absolute value of $((1-Z_t) \circ H'_t + Z_t \circ H_{t-1})$ under control, like they learn to do in the standard conv-GRU case, the network should remain stable, because the eigenvalues of W_h are normalized. A rational similar to the one in section 8.7.1 should hold.

9.1.1 Gradients of the Rotation-GRU

So far we have only considered the forward pass of the optimization process. In order for our ideas to work we must also consider the backward pass. The two are similar, because in time, input and error flow follow the dynamics of the state equation H_t. This section examines the gradient equations for the convGRU and rotationGRU in detail. .....TODO!

References

[1] M Arjovskya, A Shah, and Y Bengio. Unitary evolution recurrent neural networks. *Journal of Machine Learning Research*, 2016.

[2] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[3] Bornemann. *Funktionentheorie*. Birkhäuser, 2013.

[4] D.H. Brandwood. A complex gradient operator and its application in adaptive array theory, 1983.

[5] Briggs and Van Emden. *The DFT, an Owners Manual for the Discrete Fourier Transform*. Society for industrial and applied mathematics, 1995.

[6] Davis. *Circulant Matrices*. John Wiley and Sons, 1979.

[7] D. Franken. Complex digital networks: a sensitivity analysis based on thewirtinger calculus, 1997.

[8] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transaction on Pattern analysis and machine intelligence Vol 13.*, 1991.

[9] Goodfellow. *Deep Learning.* MIT Press, 2017.

[10] Gray. Toeplitz and circulant matrices: A review. *now publishing*, 2006.

[11] Nitzan Guberman. On complex valued convolutional neural networks. Technical report, The Hebrew University of Jerusalem Israel, 2016.

[12] Sepp Hochreiter and Juergen Schidhuber. Long short term memory. *Neural Computation*, 1997.

[13] Stephanie L. Hyland and Gunnar Raetsch. Learning unitary operators with help from u(n). In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[14] Li Jing, Caglar Gulcehre, John Peurifoy, Max Tegmark Yichen Shen, Marin Solja, and Yoshua Bengio. Gated orthogonal recurrent units: On learning to forget. 2017.

[15] Ken Kreutz-Delgado. The complex gradient operator and the cr-calculus. 2009.

[16] Danilo P. Mandic and Vanessa Su Lee Goh. *Complex Valued Nonlinear adaptive filters.* Wiley, 2009.

[17] Markus Michaelis and Gerald Sommer. A lie group approach to steerable filters. *Pattern Recognition Letters*, 1995.

[18] Pascanu. On the difficulty of training recurrent neural networks. *Journal of Machine Learning Research*, 2013.

[19] Reinhold Remmert. *Funktionentheorie 1.* Springer-Lehrbuch, 1992.

[20] Simone Scardapane, Steven Van Vaerenbergh, Amir Hussain, and Aurelio Uncini. Complex-valued neural networks with non-parametric activation functions. 2018.

[21] Hemant Tagare. Notes on optimization on stiefel manifolds. Technical report, Yale University, 2011.

[22] Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, Joao Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Joshua Bengio, and Christopher J Pal. Deep complex networks. In *ICLR*, 2018.

[23] A. van den Bos. Complex gradient and hessian, 1994.

[24] Patrick Virtue, Stella X. Yu, and Michael Lustig. Better than real: Complex-valued neural nets for mri fingerprinting. 2017.

[25] W. Wirtinger. Zur formalen theorie der funktionen von mehr komplexen veränderlichen, 1927.

[26] Scott Wisdom, Thomas Powers, John R. Hershey, Jonathan Le Roux, , and Les Atlas. Full-capacity unitary recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2016.