

Contents

1	Introduction	2
2	Related work	2
3	Complex Unitary Memory cells	3
3.1	Proof of stability	3
3.2	Results	4
4	Other Ideas	4
4.1	Fourier Space rotations.	4
4.2	Unitary dynamic filter networks	5
5	Background	5
5.1	Phase-Relus	5
5.1.1	Phase-Relu approximate stability proof in Cartesian co-ordinates.	7
5.2	Analysis of existing complex activation functions.	8
5.2.1	zRelu	8
5.2.2	modRelu	9
5.2.3	cRelu	10
5.2.4	Cardioid	11
5.2.5	Phase-amplitude activation	11
5.3	Backward mode automatic differentiation gradients	13
5.3.1	The linear case	13
5.3.2	Non-linear Cayley-Networks	14
5.4	Convolutions and circulant matrices	14
5.5	1D-convolutions and circulant matrices	14
5.6	The linear one-dimensional case	15
5.7	Effects on linear network stability	16
5.7.1	Matrix power	16
5.7.2	Network conditioning	16
5.8	Two dimensional convolutions	17
5.8.1	Doubly block circulant matrices	17
5.8.2	Doubly block circulant matrices and their eigenvalues	18
5.8.3	The spectrum of real doubly block circulant matrices.	19
6	Other related ideas	22
6.1	Rotation-GRU in \mathbb{R}	22
6.1.1	Gradients of the Rotation-GRU	22

Complex gated memory cells.

Moritz Wolter
Uni Bonn

Angela Yao
Uni Bonn

April 24, 2018

Abstract

RNN optimization often suffers from numerically unstable gradients. We propose a novel complex RNN architecture, which can be shown to be numerically stable. Building on top of recent successes 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:

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

$$\text{such that } \forall m \|\phi_m\| = 1, \quad (2)$$

$$\forall n \|f'(h_n)\| \leq 1. \quad (3)$$

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 activation derivatives. We show that our complex gated memory cells are practically stable and use Wirtinger calculus to overcome limitations on scalar 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

The training process of artificial neural networks is not necessarily stable. Unstable problem formulations lead to problems which are hard to optimize and converge slowly. Recently [1] has been able to prove that recurrent neural networks with normalized eigenvalues and bounded activation derivatives must be stable.

2 Related work

Normalized complex matrices were first introduced into the literature by [1]. Since then [23], expanded the reach of the unitary matrix basis. An idea that is

taken further by [12], which makes use Lie group theory. A holomorph non-linearity was used in [10], [1] introduces a novel non-linearity which is not complex-differentiable. [19] compares complex non-linearities and systematically measures performance. Furthermore complex batch-normalization is introduced. Finally [13], 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 addressed in the digital signal processing literature. Where complex problems with real cost functions had to be solved [3][20][6][14]. Applications to neural network cost functions were considered later [15]. All solutions essentially utilize Wirtinger calculus [22], to come up with an approximate gradient for a non-holomorph function.

3 Complex Unitary Memory cells

Setting up complex gating mechanisms is no trivial task, because functions from $\mathbb{C} \rightarrow \mathbb{R}$ cannot be holomorph unless they are constant [2, page 9]¹. Furthermore bounded holomorph complex functions must be constant [2, page 38]². 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. We define:

$$\mathbf{i}_g = \sigma(\mathbf{U}_i[\Re(\mathbf{h}_t) \Im(\mathbf{h}_t)]^T + \mathbf{W}_i[\Re(\mathbf{x}) \Im(\mathbf{x})]^T) \quad (4)$$

$$\mathbf{f}_g = \sigma(\mathbf{U}_f[\Re(\mathbf{h}_t) \Im(\mathbf{h}_t)]^T + \mathbf{W}_f[\Re(\mathbf{x}) \Im(\mathbf{x})]^T) \quad (5)$$

$$\mathbf{h}_{t+1} = \mathbf{U}f(\mathbf{f}_g \odot \mathbf{h}_t) + \mathbf{W}(\mathbf{i}_g \odot \mathbf{x}) \quad (6)$$

Where \mathbf{U} is 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 [6][14][22], to define a pseudogradient, which we argue is sufficient to train the gates. Please note that we do not distribute our derivatives over a sum, a major difference to the classical formulation in [11].

TODO: explain our gradient approximation.

3.1 Proof of stability

Hochreiter et al. [11] have shown, that a similar cell defined on \mathbb{R} , must be stable. Following [1] we prove that Complex Unitary Memory cells are also be

¹Proof: <https://math.stackexchange.com/questions/1004672/prove-that-a-real-valued-constant-function-is-holomorphic-and-vice-versa>

²[https://en.wikipedia.org/wiki/Liouville%27s_theorem_\(complex_analysis\)](https://en.wikipedia.org/wiki/Liouville%27s_theorem_(complex_analysis))

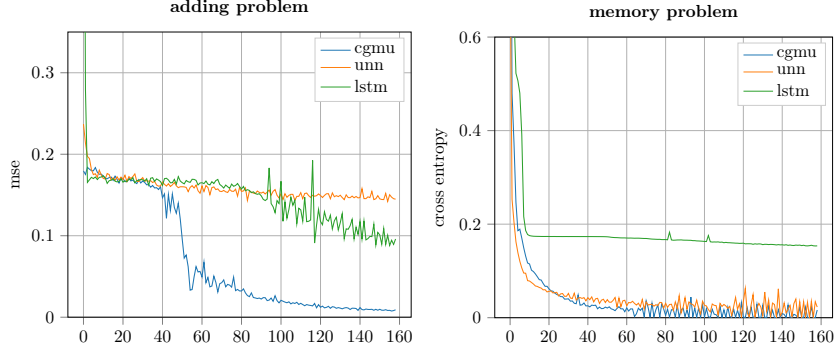


Figure 1: Performance of the complex gated memory unit (cgmu, ours), the unitary neural network (unn, [1]), and long short term memory (lstm, [11]).

stable:

$$\frac{\partial C}{\partial h_t} = \frac{\partial C}{\partial h_T} \frac{\partial h_T}{\partial h_t} \quad (7)$$

$$= \frac{\partial C}{\partial h_T} \prod_{k=t}^{T-1} \frac{\partial h_{k+1}}{\partial h_k} \quad (8)$$

$$= \frac{\partial C}{\partial h_T} \prod_{k=t}^{T-1} \mathbf{U} \mathbf{G}_{k+1} \frac{\partial f}{\partial h_k} \quad (9)$$

3.2 Results

To test our ideas we used the benchmark originally proposed in [11] following the implementation of [1]. A visualization of our results is shown in figure 1. All models were 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 threshold. However the convergence behaviour differs. We argue that our approach gets the best of both worlds in terms of performance and converges well on both the adding and memory problems, it shows the superior 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.

4 Other Ideas

4.1 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

theorem [4, page 173].³ ⁴

$$\mathcal{D}(f_{m+m_0, n+n_0}) = \omega_M^{-m_0 j} \omega_N^{-n_0 k} F_{jk} \quad (10)$$

$$\text{with } \omega_N^{nk} = e^{i2\pi nk/N} \quad (11)$$

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 interpolation⁵. 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?

4.2 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 [7] was the basis for the original dynamic filter paper. Which is why I think the fourier extension of this paper [16] could hold some cues for a nice extension. In particular, because outside of the vision domain, [12] has already shown that this is an interesting idea.

5 Background

5.1 Phase-Relus

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

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \text{ and } \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \quad (12)$$

This form has been considered in [19] and used to evaluate 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 ⁶:

$$\frac{\partial u}{\partial r} = \frac{1}{r} \frac{\partial v}{\partial \theta} \text{ and } \frac{\partial v}{\partial r} = -\frac{1}{r} \frac{\partial u}{\partial \theta} \quad (13)$$

³http://www.nontrivialzeros.net/KGL_Papers/27_Rotation_Paper_1997_qualityscan_OCR.pdf

⁴<http://bigwww.epfl.ch/publications/unser9502.pdf>

⁵https://en.wikipedia.org/wiki/Trigonometric_interpolation

⁶Proof see: <https://math.stackexchange.com/questions/1245754/cauchy-riemann-equations-in-polar-form>

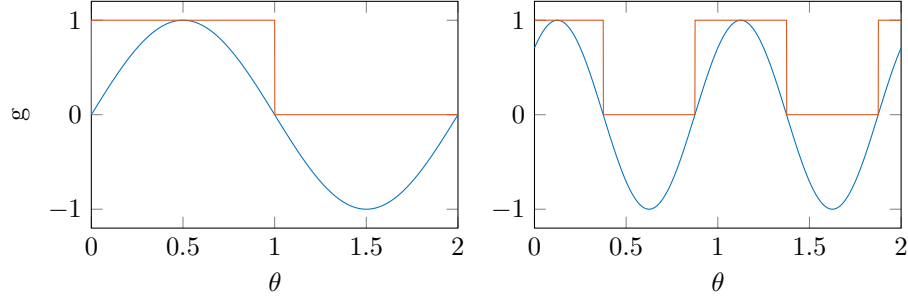


Figure 2: Plot of the $\sin(\theta a\pi + b\pi)$ and $H(\sin(\theta a\pi + b\pi))$ with $a = 1, b = 0$ (left) and $a = 2, b = 0.1$

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:

$$f(r, \theta) = g(r, \theta, a, b)e^{i\theta} \quad (14)$$

$$= g(r, \theta, a, b) \cos(\theta) + ig(r, \theta, a, b) \sin(\theta) \quad (15)$$

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

$$g(r, \theta, a, b) = rH(\sin(\theta \cdot a\pi + b)) \quad (16)$$

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

$$\frac{\partial u}{\partial r} = H(\sin(\theta \cdot a\pi + b)) \cos \theta, \quad (17)$$

$$\frac{\partial v}{\partial r} = H(\sin(\theta \cdot a\pi + b)) \sin \theta, \quad (18)$$

$$\begin{aligned} \frac{\partial u}{\partial \theta} &= -rH(\sin(\theta \cdot a\pi + b)) \sin \theta \\ &\quad + r\delta(\sin(\theta \cdot a\pi + b)) \cos(\theta \cdot a\pi + b)a\pi \cos(\theta) \end{aligned} \quad (19)$$

$$\begin{aligned} \frac{\partial v}{\partial \theta} &= rH(\sin(\theta \cdot a\pi + b)) \cos \theta \\ &\quad + r\delta(\sin(\theta \cdot a\pi + b)) \cos(\theta \cdot a\pi + b)a\pi \sin(\theta) \end{aligned} \quad (20)$$

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 holomorph⁷.

$H(\sin(\theta \cdot a\pi + b))$ 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

⁷Strictly speaking it is holomorph, when excluding all points where $\sin(\theta \cdot a\pi + b) = 0$.

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 16:

$$g(r, \theta, a, b, c, d) = rH(\sin(\theta \cdot a\pi + b))H(\cos(\theta \cdot c\pi + d)) \quad (21)$$

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 [10][19]. 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.

5.1.1 Phase-Relu approximate stability proof in Cartesian coordinates.

We have shown that

$$f(r, \theta) = rH\sin(\theta \cdot a\pi + b)e^{i\theta} \quad (22)$$

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

$$f(x + iy) = H(\sin(\text{atan2}(x, y)))(x + iy) \quad (23)$$

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

$$f(x + iy) = xH(\sin(\text{atan2}(x, y))) + iyH(\sin(\text{atan2}(y/x))) \quad (24)$$

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

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}, \quad (25)$$

and using the facts that the derivatives of $\text{atan2}(x, y)$ 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:

$$\begin{aligned} \frac{\partial u}{\partial x} &= H(\sin(\tan^{-1}(y/x))) \\ &\quad + x\delta(\sin(\tan^{-1}(y/x))) \cos(\tan^{-1}(y/x)) \left(\frac{-y}{x^2+y^2}\right) \left(\frac{-y}{x^2}\right) \end{aligned} \quad (26)$$

$$\frac{\partial u}{\partial y} = \delta(\sin(\tan^{-1}(y/x))) \cos(\tan^{-1}(y/x)) \left(\frac{x}{x^2+y^2}\right) \quad (27)$$

$$\frac{\partial v}{\partial x} = y\delta(\sin(\tan^{-1}(y/x))) \cos(\tan^{-1}(y/x)) \left(\frac{-y}{x^2+y^2}\right) \left(\frac{-y}{x^2}\right) \quad (28)$$

$$\begin{aligned} \frac{\partial v}{\partial y} &= H(\sin(\tan^{-1}(y/x))) \\ &\quad + y\delta(\sin(\tan^{-1}(y/x))) \cos(\tan^{-1}(y/x)) \left(\frac{x}{x^2+y^2}\right) \left(\frac{1}{x}\right) \end{aligned} \quad (29)$$

Most of the time the Dirac terms $\delta(\cdot)$ will be zero and $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = H(\sin(\tan^{-1}(y/x)))$ 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.

5.2 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:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}, \quad (30)$$

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

$$\frac{\partial u}{\partial r} = \frac{1}{r} \frac{\partial v}{\partial \theta} \text{ and } \frac{\partial v}{\partial r} = -\frac{1}{r} \frac{\partial u}{\partial \theta} \quad (31)$$

which is equivalent.

5.2.1 zRelu

$$\text{zRelu}(z) = \begin{cases} z & \text{if } \theta \in [0, \pi/2], \\ 0 & \text{else.} \end{cases} \quad (32)$$

Following [19] we have for the first quadrant:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 1, \quad (33)$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0, \quad (34)$$

and elsewhere:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 0, \quad (35)$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0, \quad (36)$$

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 5.1.1:

$$\text{zRelu}(z) = \text{H}(\sin(\text{atan2}(x, y)))\text{H}(\cos(\text{atan2}(x, y)))(x + iy) \quad (37)$$

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.

5.2.2 modRelu

The modRelu is defined as [1]:

$$f(z) = \text{Relu}(\|z\| + b) \frac{z}{\|z\|}. \quad (38)$$

Conversion to polar coordinates yields:

$$f(r, \theta) = \text{Relu}(r + b)e^{i\theta}, \quad (39)$$

$$f(r, \theta) = \text{Relu}(r + b) \cos(\theta) + i\text{Relu}(r + b) \sin(\theta). \quad (40)$$

$$(41)$$

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

$$\frac{\partial u}{\partial r} = \text{H}(r + b) \cos(\theta), \quad (42)$$

$$\frac{\partial u}{\partial \theta} = -\text{Relu}(r + b) \sin(\theta), \quad (43)$$

$$\frac{\partial v}{\partial r} = \text{H}(r + b) \sin(\theta), \quad (44)$$

$$\frac{\partial v}{\partial \theta} = \text{Relu}(r + b) \cos(\theta). \quad (45)$$

For holomorphy we require:

$$\frac{\partial u}{\partial r} = \frac{1}{r} \frac{\partial v}{\partial \theta}, \quad (46)$$

$$\Leftrightarrow r\text{H}(r + b) \cos(\theta) = \text{Relu}(r + b) \cos(\theta); \quad (47)$$

$$\frac{\partial v}{\partial r} = -\frac{1}{r} \frac{\partial u}{\partial \theta}, \quad (48)$$

$$\Leftrightarrow r\text{H}(r + b) \sin(\theta) = \text{Relu}(r + b) \sin(\theta). \quad (49)$$

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

$$\frac{\partial}{\partial z} \sigma_{\text{Relu}}(\|z\| + b) \frac{z}{\|z\|} = \sigma'_{\text{Relu}}(\|z\| + b) \frac{z}{\|z\|} + \sigma_{\text{Relu}}(\|z\| + b) \left(\frac{z}{\|z\|} \right)'. \quad (50)$$

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.

5.2.3 cRelu

[19] defines the cRelu as:

$$\text{cRelu}(z) = \text{Relu}(x) + i \cdot \text{Relu}(y). \quad (51)$$

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

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 1, \quad (52)$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0. \quad (53)$$

For the second we find:

$$\frac{\partial u}{\partial x} = 0 \neq \frac{\partial v}{\partial y} = 1, \quad (54)$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0, \quad (55)$$

The third quadrant has:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 0, \quad (56)$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0, \quad (57)$$

Finally considering the fourth:

$$\frac{\partial u}{\partial x} = 1 \neq \frac{\partial v}{\partial y} = 0, \quad (58)$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} = 0, \quad (59)$$

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 $\text{sign}(\Re(z)) = \text{sign}(\Im(z))$ [19], which is the case in the first and 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.

5.2.4 Cardioid

[21] introduces the complex cardioid,:

$$f(z) = \frac{1}{2}(1 + \cos(\theta))z \quad (60)$$

Which we express in polar-coordinates as:

$$f(r, \theta) = \frac{1}{2}(1 + \cos(\theta))re^{i\theta} \quad (61)$$

Using the definition of the complex exponential we obtain:

$$f(r, \theta) = \frac{1}{2}(1 + \cos(\theta))r \cos(\theta) + i\frac{1}{2}(1 + \cos(\theta))r \sin(\theta) \quad (62)$$

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:

$$\frac{\partial u}{\partial r} = \frac{1}{2}(1 + \cos(\theta)) \cos(\theta) \quad (63)$$

$$\frac{\partial u}{\partial \theta} = -r\frac{1}{2}(1 + \cos(\theta)) \sin(\theta) - r\frac{1}{2} \sin(\theta) \cos(\theta) \quad (64)$$

$$\frac{\partial v}{\partial r} = \frac{1}{2}(1 + \cos(\theta)) \sin(\theta) \quad (65)$$

$$\frac{\partial v}{\partial \theta} = r\frac{1}{2}(1 + \cos(\theta)) \cos(\theta) - r\frac{1}{2} \sin(\theta) \cos(\theta) \quad (66)$$

And therefore we require:

$$\frac{\partial u}{\partial r} = \frac{1}{2}(1 + \cos(\theta)) \cos(\theta) = \frac{\partial v}{r\partial \theta} = \frac{1}{2}(1 + \cos(\theta)) \cos(\theta) - r\frac{1}{2} \sin(\theta) \cos(\theta) \quad (67)$$

$$\Leftrightarrow 0 = -r\frac{1}{2} \sin(\theta) \cos(\theta) \quad (68)$$

$$(69)$$

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.

5.2.5 Phase-amplitude activation

[18] cites these as:

$$f(z) = \tanh(r/m)e^{i\theta} \quad (70)$$

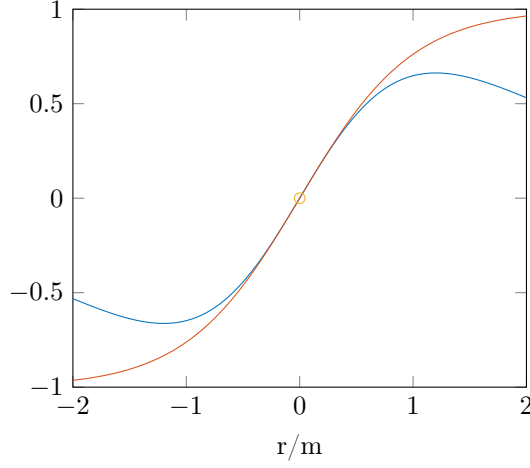


Figure 3: Plot of the holomorphy condition's two sides for the Phase-amplitude activation. $\tanh(r/m)$ is shown in red, $\text{sech}^2(r/m)(r/m)$ is shown in blue. The yellow circle indicates the point at $(0, 0)$.

Considering the polar C.R. equations:

$$f(r, \theta) = \cos(\theta) \tan^{-1}(r/m) + i \sin(\theta) \tan^{-1}(r/m) \quad (71)$$

$$\Rightarrow u + iv$$

$$\frac{\partial u}{\partial r} = \cos(\theta) \text{sech}^2(r/m)(1/m) \quad (72)$$

$$\frac{\partial u}{\partial \theta} = -\sin(\theta) \tanh(r/m) \quad (73)$$

$$\frac{\partial v}{\partial r} = \sin(\theta) \text{sech}^2(r/m)(1/m) \quad (74)$$

$$\frac{\partial v}{\partial \theta} = \cos(\theta) \tanh(r/m) \quad (75)$$

$$\Rightarrow \frac{\partial u}{\partial r} = \frac{\partial v}{r \partial \theta}$$

$$\Leftrightarrow \cos(\theta) \text{sech}^2(r/m)(1/m) = \frac{1}{r} \cos(\theta) \tanh(r/m) \quad (76)$$

$$\Leftrightarrow \text{sech}^2(r/m)(r/m) = \tanh(r/m) \quad (77)$$

This activation is holomorph at $r/m = 0$. And approximately analytic for $r/m \approx 0$ as shown in figure 3, the problem here is that this non-linearity breaks magnitude information by rescaling them, it must therefore be non-holomorph for large inputs.

5.3 Backward mode automatic differentiation gradients

Consider the non-linear network proposed in [17]:

$$\mathbf{x}_t = W_{\text{rec}} f(\mathbf{x}_{t-1}) + W_{\text{in}} \mathbf{u}_t + \mathbf{b} \quad (78)$$

Following [17] we define the error function $\mathcal{E}_t = \mathcal{L}$ and work with BPTT gradients given by:

$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{1 \leq t \leq T} \frac{\mathcal{E}_t}{\partial \theta} \quad (79)$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{1 \leq k \leq t} \left(\frac{\mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} \frac{\partial^+ \mathbf{x}_k}{\partial \theta} \right) \quad (80)$$

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \geq i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \prod_{t \geq i > k} W_{\text{rec}}^T \text{diag}(f'(\mathbf{x}_{i-1})) \quad (81)$$

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

5.3.1 The linear case

Working with $f(x) = x$ according the long term behavior is determined by the matrix product $\partial \mathbf{x}_t / \partial \mathbf{x}_k$ [17]. We define $W_{\text{rec}} = C$, and normalize the spectrum of C to $\forall k \|\phi_k\| = 1$ for $k \in \{1 \dots n\}$. Focusing on the term in the sum of equation 80, we express $\partial \mathcal{E} / \partial \mathbf{x}_t$ in terms of a Fourier basis:

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_t} = \sum_{i=1}^n \mathbf{f}_i^T d_i \quad (82)$$

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:

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \sum_{i=1}^n \mathbf{f}_i^T d_i \phi_i \quad (83)$$

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

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \sum_{i=1}^n \mathbf{f}_i^T d_i \phi_i \approx \sum_{i=1}^n \mathbf{f}_i^T d_i = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_t} \quad (84)$$

Because $\phi = \exp(i\omega)$ merely represents a rotation of the errors phase angle, but leaves its magnitude intact. Using the train of thought borrowed from [17], we claim to establish an error carousel similar to Hochreiter 1998, through which errors can pass in a stable manner.


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


Figure 4: Circulant matrix structure as formula and in plotted form.

5.3.2 Non-linear Cayley-Networks

We could employ a non-linearity based on the Cayley-Transform [2, p. 100]:

$$C'(z) = \frac{z - i}{z + i} \quad (85)$$

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

$$C(z) = z - 2i \ln(z + i) \quad (86)$$

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:

$$D'(z) = \frac{z + i}{z - i} \quad (87)$$

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.

5.4 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 [8, 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 literature⁸.

5.5 1D-convolutions and circulant matrices

Consider for example the four by four circulant matrix $C = \text{circ}(c_1, c_2, c_3, c_4)$ as shown in figure 4. The matrix vector product $C\mathbf{x}$ with $\mathbf{x} = (x_1, x_2, x_3, x_4)^T$,

⁸<http://nzjm.math.auckland.ac.nz/images/8/8e/18-36.pdf>

can be written as:

$$c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \quad (88)$$

$$c_4x_1 + c_1x_2 + c_2x_3 + c_3x_4 \quad (89)$$

$$c_3x_1 + c_4x_2 + c_1x_3 + c_2x_4 \quad (90)$$

$$c_2x_1 + c_3x_2 + c_4x_3 + c_1x_4 \quad (91)$$

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.

5.6 The linear one-dimensional case

According to [9, page 33], the eigenbasis of all circulant matrices is given by:

$$\mathbf{f}^{(m)} = \frac{1}{\sqrt{n}}(1, \exp(-2\pi im/n), \dots, \exp(-2\pi im(n-1)/n))' \quad (92)$$

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

$$C = F\Phi F^{-1} \quad (93)$$

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

$$\phi_m = r \exp i\omega \quad (94)$$

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_m s 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.

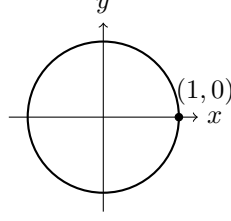


Figure 5: Illustration of the unit circle, on which we place all convolution eigenvalues $\phi = x + iy$.

5.7 Effects on linear network stability

5.7.1 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.

$$y = C_1 \cdot C_2 \dots C_n \cdot x \quad (95)$$

$$y = F\Phi_1 F^{-1} \cdot F\Phi_2 F^{-1} \dots F\Phi_n F^{-1} \cdot x \quad (96)$$

$$y = F\Phi_1 \cdot \Phi_2 \dots \Phi_n F^{-1} \cdot x \quad (97)$$

$$(98)$$

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:

$$\phi_m = \exp(i\omega_{m,1} + i\omega_{m,2} \dots i\omega_{m,n}) \quad (99)$$

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.

5.7.2 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 associated condition number is very large. A problem's conditioning is measured using:

$$\kappa = \max_{i,j} \left| \frac{\phi_i}{\phi_j} \right| \quad (100)$$

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.

5.8 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:

$$A * B = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} * \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \quad (101)$$

May be expressed using matrix multiplication as:

$$A * B = K^T \cdot B_{\text{flat}} \quad (102)$$

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

$$K = \begin{pmatrix} a_1 & a_2 & 0 & a_3 & a_4 & 0 & 0 & 0 & 0 \\ 0 & a_1 & a_2 & 0 & a_3 & a_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & a_2 & 0 & a_3 & a_4 & 0 \\ 0 & 0 & 0 & 0 & a_1 & a_2 & 0 & a_3 & a_4 \end{pmatrix} \quad (103)$$

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

5.8.1 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 to⁹:

$$A_p = \begin{pmatrix} a_1 & a_2 & 0 & 0 \\ a_3 & a_4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} B_p = \begin{pmatrix} b_1 & b_2 & 0 & 0 \\ b_3 & b_4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (104)$$

⁹I 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.

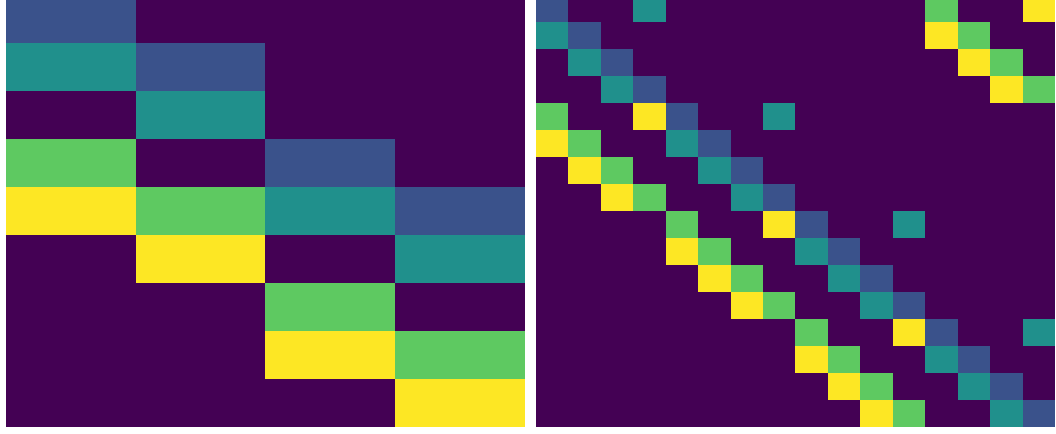


Figure 6: Visualization of a two dimensional convolution matrix and its square doubly circulant cousin.

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

$$C_0 = \text{circ}(c_0) = \begin{pmatrix} a_1 & a_2 & 0 & 0 \end{pmatrix} \quad (105)$$

$$C_1 = \text{circ}(c_1) = \begin{pmatrix} a_2 & a_3 & 0 & 0 \end{pmatrix} \quad (106)$$

$$C_2 = \text{circ}(c_2) = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix} \quad (107)$$

$$C_3 = \text{circ}(c_3) = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix} \quad (108)$$

$$(109)$$

Which leads to the resulting matrix C :

$$C_b = \begin{pmatrix} C_0 & C_1 & C_2 & C_3 \\ C_3 & C_0 & C_1 & C_2 \\ C_2 & C_3 & C_0 & C_2 \\ C_1 & C_2 & C_3 & C_0 \end{pmatrix} \quad (110)$$

A visualization of this matrix is shown in figure 6 on the right. Multiplication of $C_b \cdot B_{p \text{ flat}}$ will lead to a zero padded version of $K^T \cdot B_{\text{flat}}$.

5.8.2 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 [5, page 185], their diagonalization is given by:

$$C_b = \overline{(F_m \otimes F_n)^T} \Lambda (F_m \otimes F_n) \quad (111)$$

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,

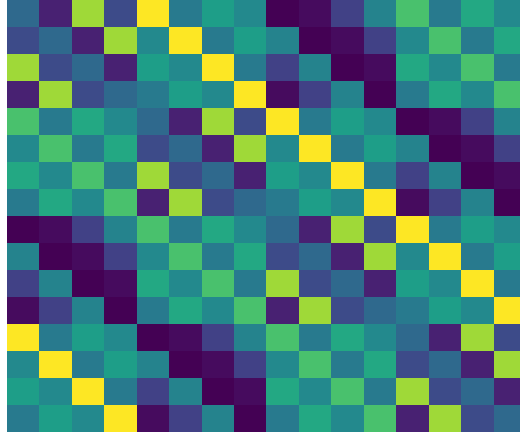


Figure 7: Absolute values of complex doubly block circulant matrix constructed in the frequency domain.

which will be square with m block containing n rows each. Given a real input matrix we can find Lambda from:

$$\Lambda = (F_m \otimes F_n) C_b \overline{(F_m \otimes F_n)^T} \quad (112)$$

We believe that 111 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.

5.8.3 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 8 and 9 illustrate this. The spectrum shown in 8 is symmetric along the a-Axis, when cutting after its third element and disregarding the first eigen-vale. The block circulant case shown in 8, we find the same symmetry in the zeroth column and 4th row. The middle block is point symmetric.

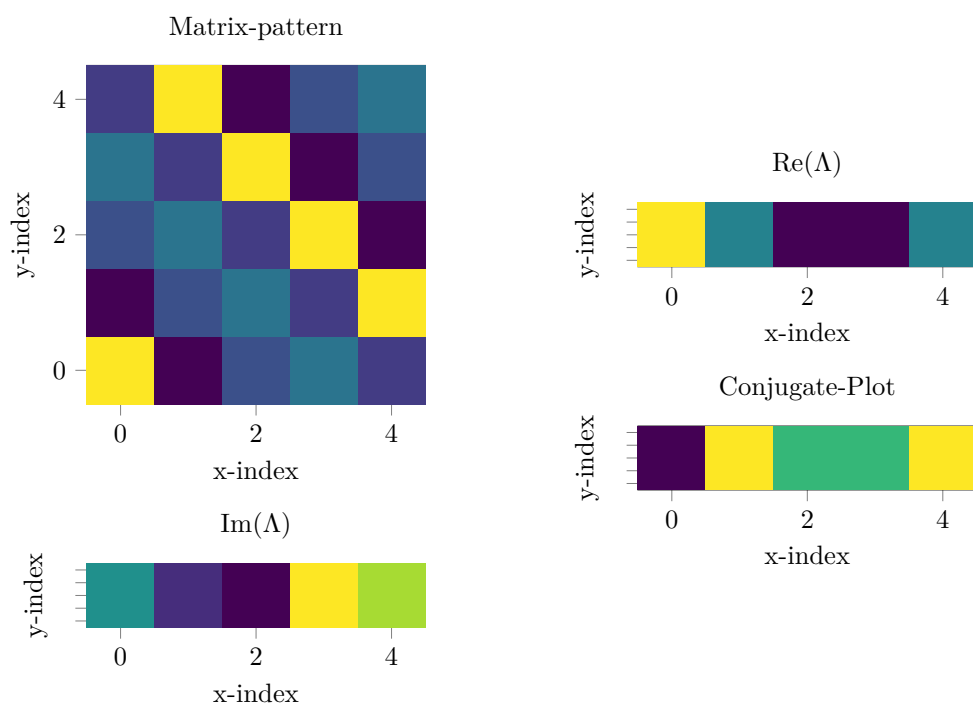


Figure 8: Circulant matrix pattern and spectrum.

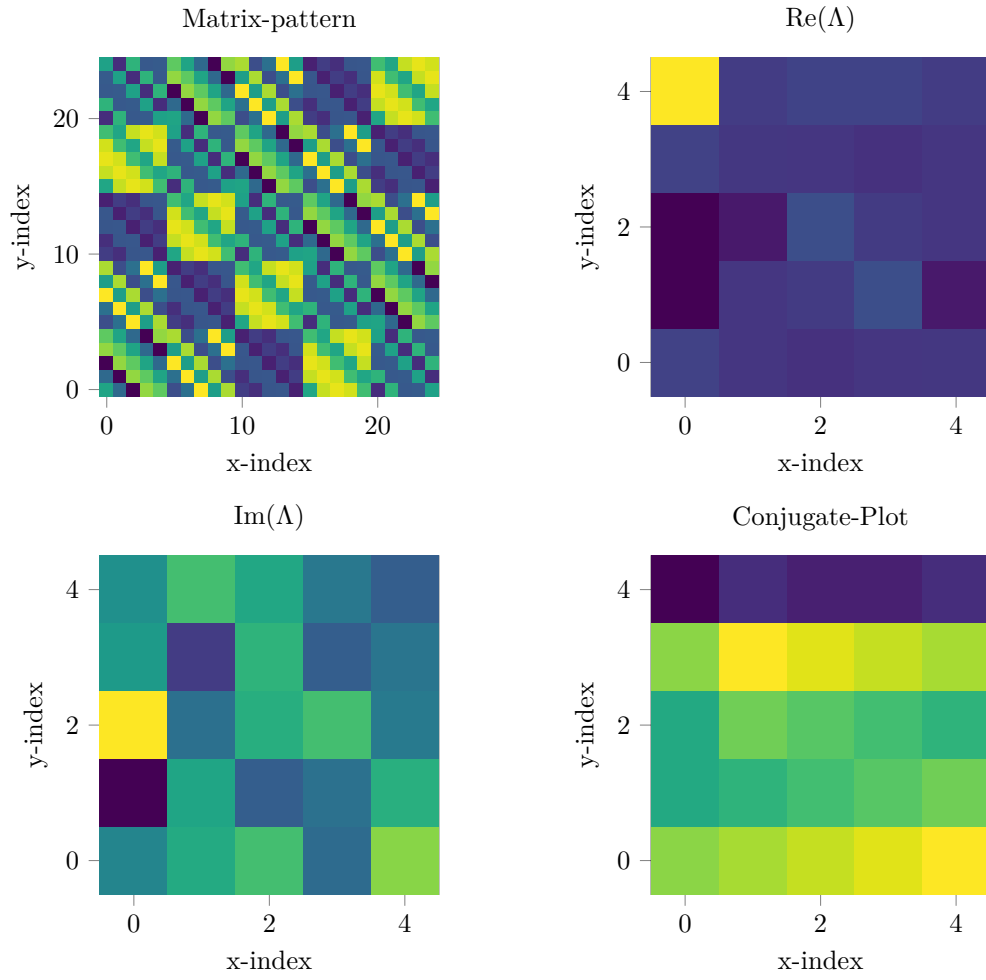


Figure 9: Block Circulant matrix pattern and spectrum.

6 Other related ideas

6.1 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:

$$Z_t = \sigma(W_{xz} * X_t + W_{hz} * H_{t-1} + b_z), \quad (113)$$

$$R_t = \sigma(W_{xr} * X_t + W_{hr} * H_{t-1} + b_r), \quad (114)$$

$$H'_t = f(W_{xr} * X_t) + R_t \circ (W_{hp} * H_{t-1}), \quad (115)$$

$$H_t = (1 - Z_t) \circ H'_t + Z_t \circ H_{t-1}. \quad (116)$$

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

$$H_t = W_h * ((1 - Z_t) \circ H'_t + Z_t \circ H_{t-1}). \quad (117)$$

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 5.7.1 should hold.

6.1.1 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] Bornemann. *Funktionentheorie*. Birkhäuser, 2013.
- [3] D.H. Brandwood. A complex gradient operator and its application in adaptive array theory, 1983.
- [4] Briggs and Van Emden. *The DFT, an Owners Manual for the Discrete Fourier Transform*. Society for industrial and applied mathematics, 1995.
- [5] Davis. *Circulant Matrices*. John Wiley and Sons, 1979.
- [6] D. Franken. Complex digital networks: a sensitivity analysis based on thewirtinger calculus, 1997.

- [7] 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.
- [8] Goodfellow. *Deep Learning*. MIT Press, 2017.
- [9] Gray. Toeplitz and circulant matrices: A review. *now publishing*, 2006.
- [10] Nitzan Guberman. On complex valued convolutional neural networks. Technical report, The Hebrew University of Jerusalem Israel, 2016.
- [11] Sepp Hochreiter and Juergen Schmidhuber. Long short term memory. *Neural Computation*, 1997.
- [12] 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.
- [13] Li Jing, Caglar Gulcehre, John Peurifoy, Max Tegmark Yichen Shen, Marin Solja, and Yoshua Bengio. Gated orthogonal recurrent units: On learning to forget. 2017.
- [14] Ken Kreutz-Delgado. The complex gradient operator and the cr-calculus. 2009.
- [15] Danilo P. Mandic and Vanessa Su Lee Goh. *Complex Valued Nonlinear adaptive filters*. Wiley, 2009.
- [16] Markus Michaelis and Gerald Sommer. A lie group approach to steerable filters. *Pattern Recognition Letters*, 1995.
- [17] Pascanu. On the difficulty of training recurrent neural networks. *Journal of Machine Learning Research*, 2013.
- [18] Simone Scardapane, Steven Van Vaerenbergh, Amir Hussain, and Aurelio Uncini. Complex-valued neural networks with non-parametric activation functions. 2018.
- [19] 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.
- [20] A. van den Bos. Complex gradient and hessian, 1994.
- [21] Patrick Virtue, Stella X. Yu, and Michael Lustig. Better than real: Complex-valued neural nets for mri fingerprinting. 2017.
- [22] W. Wirtinger. Zur formalen theorie der funktionen von mehr komplexen veränderlichen, 1927.

- [23] 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.