ETH zürich

@_hylandSL

Learning Unitary Operators with Help from $\mathfrak{u}(n)$

Stephanie L. Hyland[1,2], Gunnar Rätsch[1]

BIOMEDICAL INFORMATICS

[1]Department of Computer Science, ETH Zurich

[2]Tri-Institutional Training Program in Computational Biology and Medicine, Weill Cornell Medicine

February 8th, AAAI-17 San Francisco

this work in a nutshell

context:

- vanishing/exploding gradients make training RNNs hard, especially for long sequences
- having a unitary transition matrix helps gradients

problem:

- learning the unitary matrix

solution:

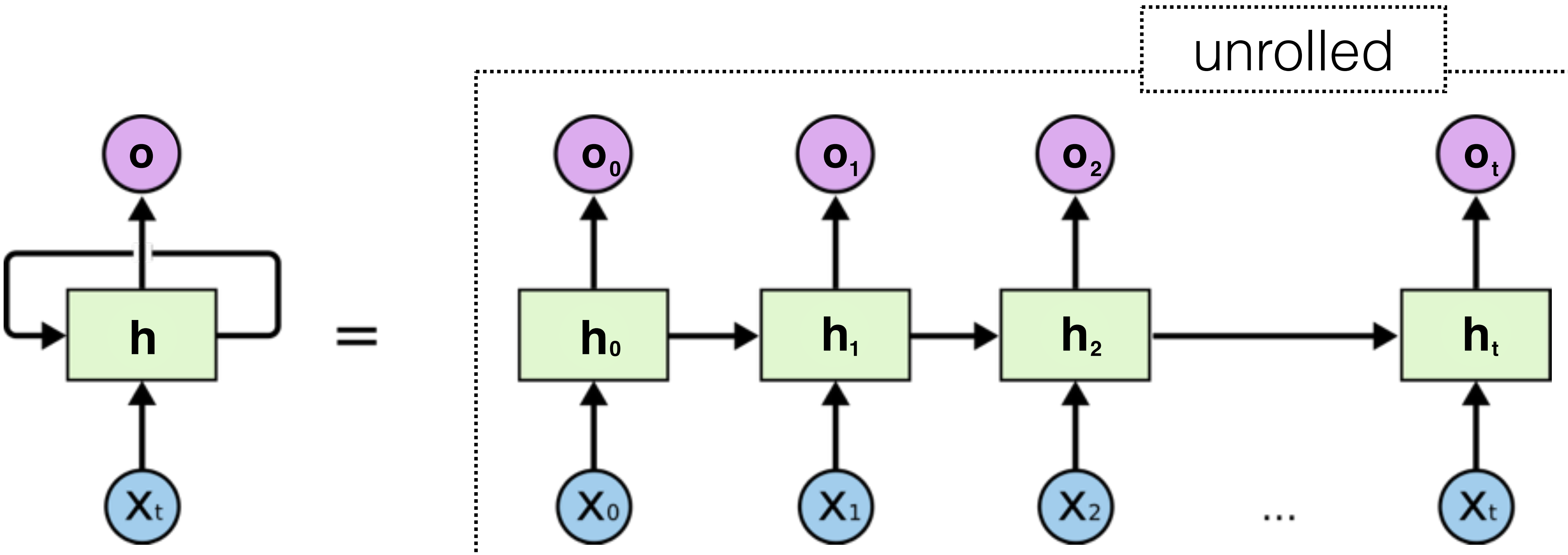
- parametrisation: use Lie group-Lie algebra correspondence

… and experiments to show it works

context

recurrent neural networks

RNN: ‘deep’ neural network with **recurrent units**:



modified from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

hidden state previous hidden state

$$\mathbf{h}_t = f\left(V\mathbf{h}_{t-1} + W\mathbf{x}_t + \mathbf{b}\right)$$

transition matrix input data

long sequences

"And then nothing can protect us against a complete falling away from our Ideas of duty, or can preserve in the soul a grounded reverence for its law, except the clear conviction that even if there never have been actions springing from such pure sources, the question at issue here is not whether this or that has happened; that, on the contrary, reason by itself and independently of all appearances commands what ought to happen; that consequently actions of which the world has perhaps hitherto given no example—actions whose practicability might well be doubted by those who rest everything on experience—are nevertheless commanded unrelentingly by reason; and that, for instance, although up to now there may have existed no loyal friend, pure loyalty in friendship can be no less required from every man, inasmuch as this duty, prior to all experience, is contained as duty in general in the Idea of a reason which determines the will by a priori grounds."

- Kant, *Groundwork of the Metaphysics of Morals*

(407-408; or pp. 75-76 of the H. J. Paton translation [New York: Harper & Row, 1964])

http://www.stephenhicks.org/2009/06/21/philosophys-longest-sentences-part-2/

long sequences

"And then nothing can protect us against a complete falling away from our Ideas of duty, or can preserve in the soul a grounded reverence for its law, except the clear conviction that even if there never have been actions springing from such pure sources, the question at issue here is not whether this or that has happened; that, on the contrary, reason by itself and independently of all appearances commands what ought to happen; that consequently actions of which the world has perhaps hitherto given no example—actions whose practicability might well be doubted by those who rest everything on experience—are nevertheless commanded unrelentingly by reason; and that, for instance, although up to now there may have existed no loyal friend, pure loyalty in friendship can be no less required from every man, inasmuch as this duty, prior to all experience, is contained as duty in general in the Idea of a reason which determines the will by a priori grounds."

- Kant, *Groundwork of the Metaphysics of Morals*

(407-408; or pp. 75-76 of the H. J. Paton translation [New York: Harper & Row, 1964])

http://www.stephenhicks.org/2009/06/21/philosophys-longest-sentences-part-2/

gradient instability

recurrence relation again:

$$\mathbf{h}_t = f\left(V\mathbf{h}_{t-1} + W\mathbf{x}_t + \mathbf{b}\right)$$

$$f(V\mathbf{h}_{t-2} + W\mathbf{x}_{t-1} + \mathbf{b})$$

$$f(V\mathbf{h}_{t-3} + W\mathbf{x}_{t-2} + \mathbf{b})$$

nested functions

etc.

chain rule (back-propagation) → products of derivatives

gradient of cost with respect to hidden state **explodes or vanishes** deeper into network

unitary RNN

standard RNN:

$$\mathbf{h}_t = f\left(V\mathbf{h}_{t-1} + W\mathbf{x}_t + \mathbf{b}\right)$$

unitary RNN:

$$\mathbf{h}_t = f\left(\boxed{U}\mathbf{h}_{t-1} + W\mathbf{x}_t + \mathbf{b}\right)$$

U is **unitary**: $U^\dagger U = \mathbb{I}$

U preserves norms: $||U\mathbf{v}|| = ||\mathbf{v}||$

how to learn U?

for deep learning: stochastic gradient descent

general update rule $$U \to \underbrace{U - \alpha \frac{\partial C}{\partial U}}_{\text{not necessarily unitary}}$$

to keep U unitary:

1. update and project
2. use a parametrisation
3. ???

parametrising $U(n)$

our approach

observations:

- unitary matrices from a group - a **Lie group** - *U(n)*
- Lie groups have associated **Lie algebras**
- Lie algebras are vector spaces - closed under addition
- for the unitary group, the map from Lie algebra to Lie group is **surjective**

therefore…

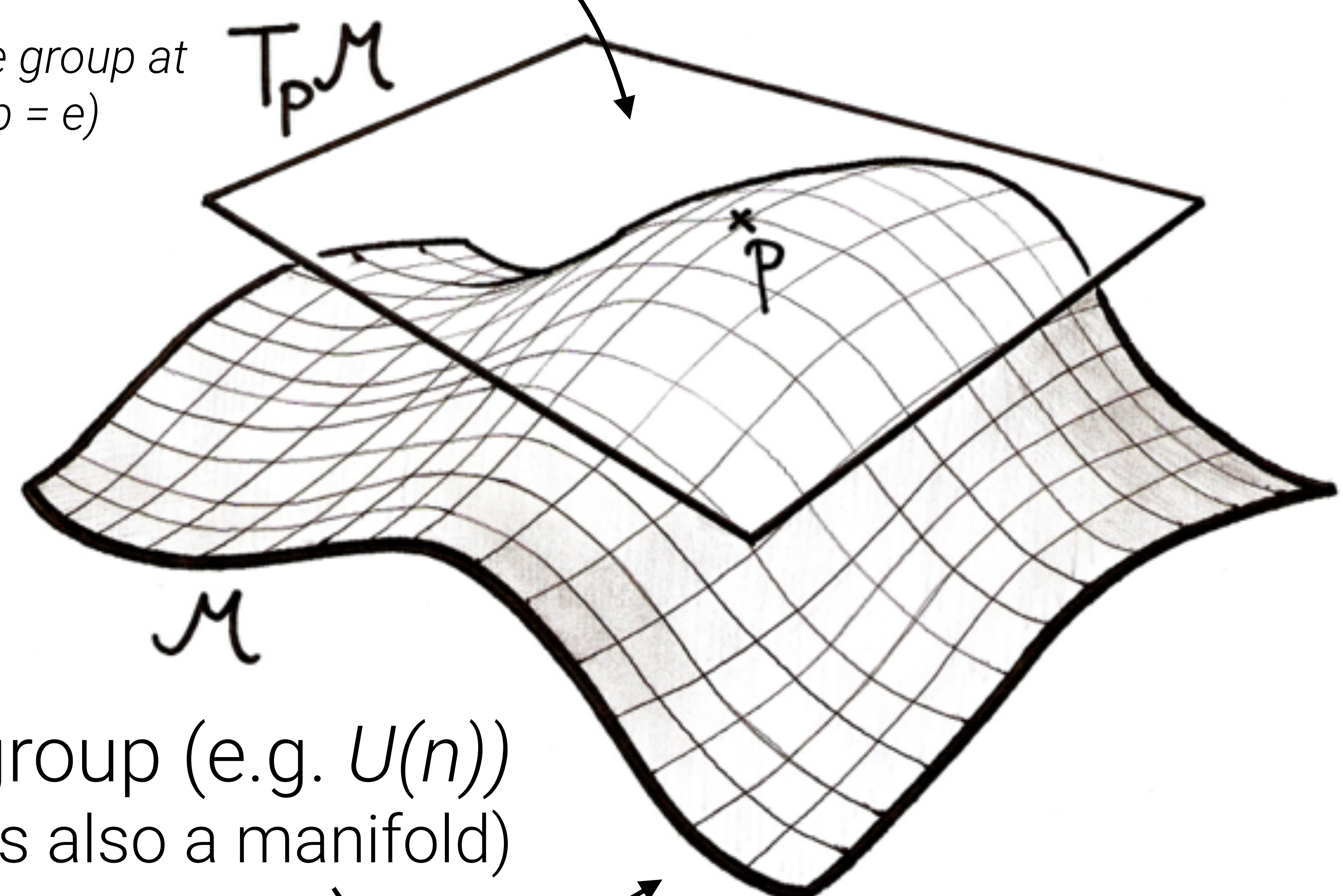
- we can parametrise unitary matrices by the **coefficients** (relative to a basis) of elements of the Lie algebra
- this can describe *any* unitary matrix

Lie groups and algebras

Lie algebra (e.g. $\mathfrak{u}(n)$)

(vector space)

(Tangent space to the Lie group at its identity element, p = e)



Lie group (e.g. *U(n))*

(a group, which is also a manifold)

figure modified from Blagoje Oblak, ‘From the Lorentz Group to the Celestial Sphere’, arXiv 2015

the exponential map

$$\exp : \mathfrak{u}(n) \to U(n)$$

Lie algebra   Lie group

$$L \in \mathfrak{u}(n)$$
$$\to \exp(L) \in U(n)$$

U(n) is a matrix group, so $\exp$ is the matrix exponential:

$$\exp(L) = \sum_{j=0}^{\infty} \frac{L^j}{j!}$$

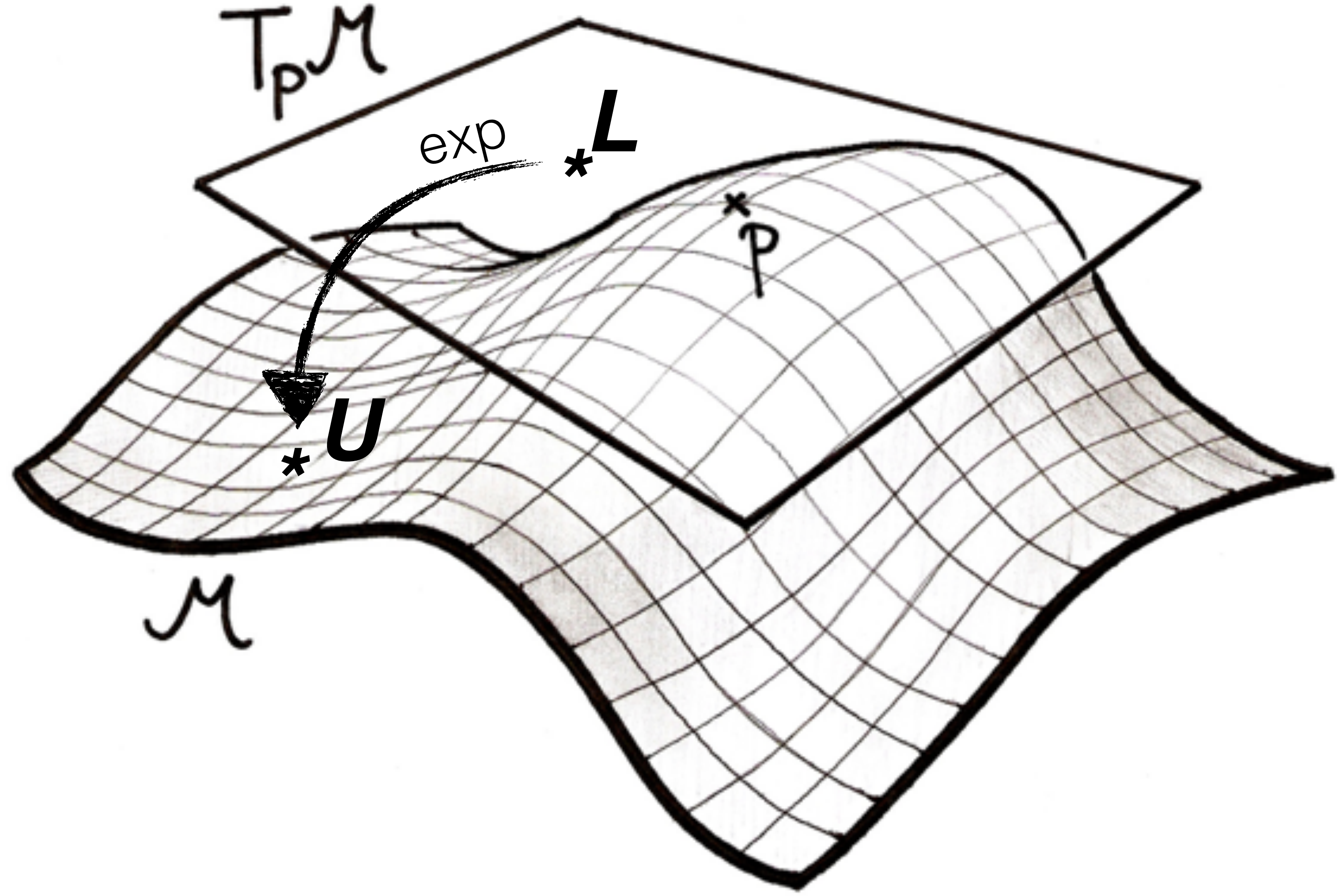


figure modified from Blagoje Oblak, 'From the Lorentz Group to the Celestial Sphere', arXiv 2015

the parametrisation

$$U = U(\lambda_1, \cdots, \lambda_{n^2})$$

λ are coefficients relative to a **basis** of the Lie algebra $\mathfrak{u}$(n) of L;

$$L = \sum_{j=1}^{n^2} \lambda_j T_j$$

basis matrix (sparse)

and we get U from L using the exponential map:

$$U = \exp(L) = \exp\left(\sum_{j=1}^{n^2} \lambda_j T_j\right)$$

experiments and results

learning matrices

simple task: learn **matrix** relating pairs of vectors:

$$\underset{\text{given}}{\mathbf{y}_i} = \boxed{\tilde{U}}\underset{\text{given}}{\mathbf{x}_i} + \underset{\text{noise (Gaussian)}}{\epsilon_i}$$

minimise squared loss between true and predicted vectors:

$$U^* = \underset{U}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \|\hat{\mathbf{y}}_j - \mathbf{y}_j\|^2 = \underset{U}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \|U\mathbf{x}_j - \mathbf{y}_j\|^2$$

compare multiple **approaches** and matrix sizes:

n	true	projection	arjovsky	lie algebra	rand
3	$6.004 \pm 0.005 \times 10^{-4}$	8 ± 1	$\mathbf{6.005 \pm 0.003 \times 10^{-4}}$	$\mathbf{6.003 \pm 0.003 \times 10^{-4}}$	12.5 ± 0.4
6	~ 0.001	15 ± 1	0.09 ± 0.01	$\mathbf{0.03 \pm 0.01}$	24 ± 1
8	~ 0.002	14 ± 1	1.17 ± 0.06	$\mathbf{0.014 \pm 0.006}$	31.6 ± 0.6
14	~ 0.003	24 ± 4	10.8 ± 0.3	$\mathbf{0.07 \pm 0.02}$	52 ± 1
20	~ 0.004	38 ± 3	29.0 ± 0.5	$\mathbf{0.47 \pm 0.03}$	81 ± 2

Arjovsky, Shah & Bengio - Unitary Evolution Recurrent Neural Networks, ICML 2016

uRNN task 1: adding

task: add two marked entries in a long sequence of numbers

input sequence: (example)

035091231860**7**294332729793875990**8**347645093841572380
000000000000**1**000000000000000000**1**000000000000000000

desired output: 15 (= 7 + 8)

cost: MSE between predicted & correct answers

length of sequence: length of time to ‘remember’

(real version uses floats between 0 and 1, but that’s harder to show)

uRNN task 2: memory

task: remember a sequence of digits over a ‘long’ time period

input sequence: (example)

58329568929000000000000000000000000000000000000000

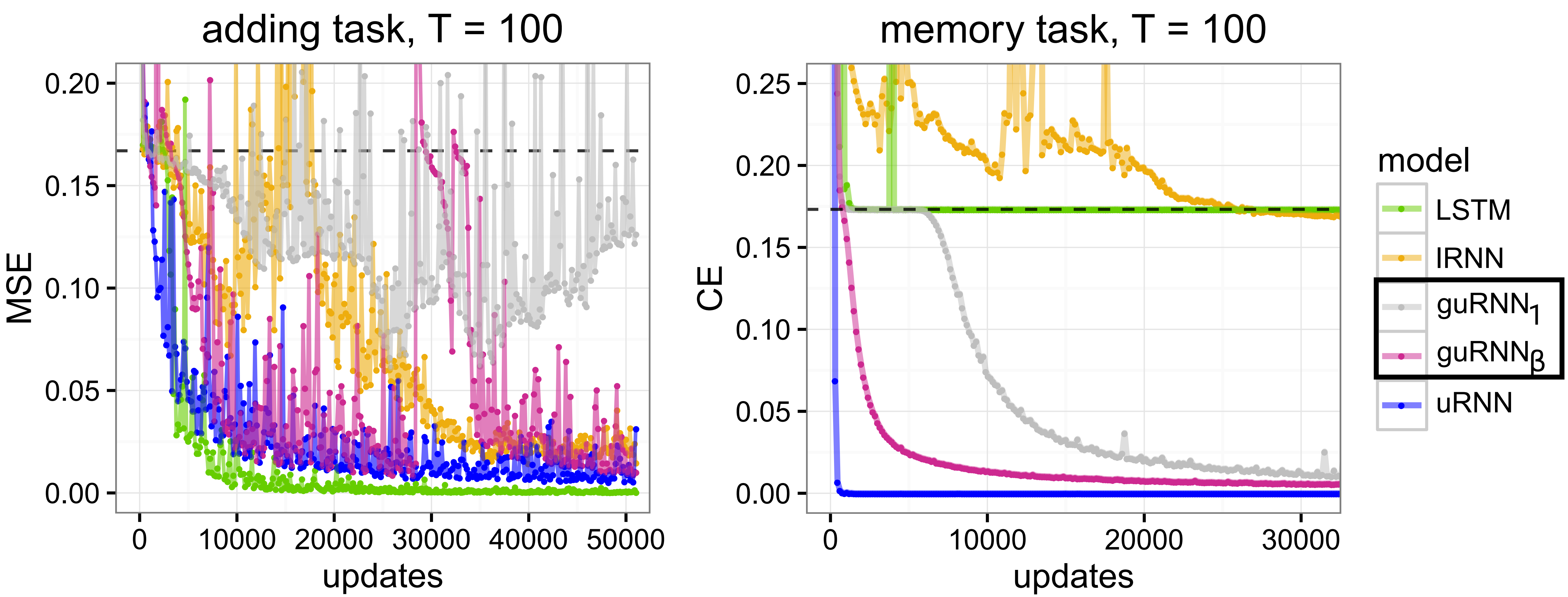
desired output sequence:

00000000000000000000000000000000000000005832956892

cost: cross-entropy between predicted & true output sequences

number of zeroes: length of time to ‘remember’

uRNN results (task 1 & 2)



Arjovsky, Shah & Bengio - Unitary Evolution Recurrent Neural Networks, ICML 2016

conclusion

this work in a nutshell

... again

context:

- vanishing/exploding gradients make training RNNs hard, especially for long sequences
- having a unitary transition matrix helps gradients

problem:

- learning the unitary matrix

solution:

- parametrisation: use Lie group-Lie algebra correspondence

... and experiments to show it works

✨further questions✨

geometry:

- optimisation directly on the Lie group (manifold)?
- how does the exponential map change the cost surface?

deep learning:

- role of nonlinearity in vanishing/exploding gradients: focusing on transition operator is not enough
- are complex-valued states necessary/helpful?

thanks!

code https://github.com/ratschlab/urnn

paper https://arxiv.org/abs/1607.04903

contact @_hylandSL

hyland@inf.ethz.ch