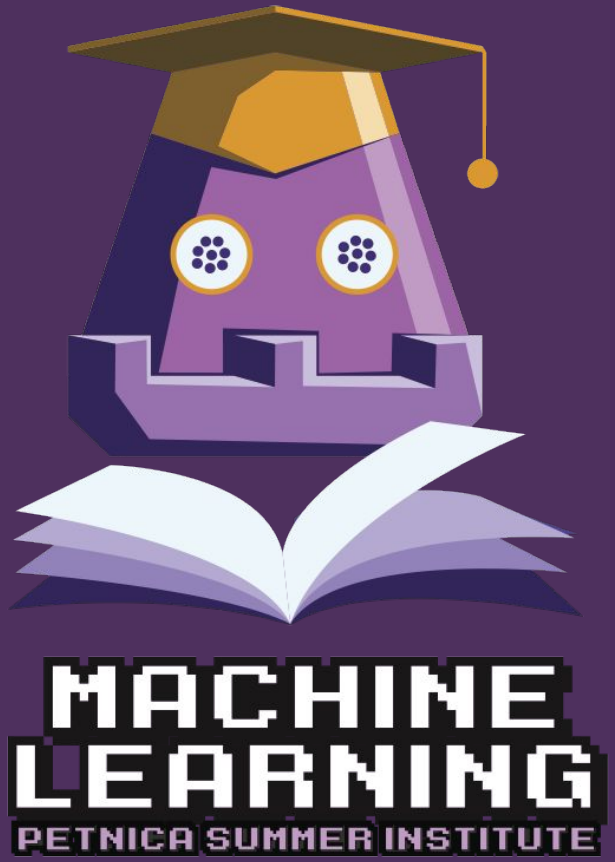
Unitary Evolution RNN

Nikola Jovanović
Ognjen Đuričić

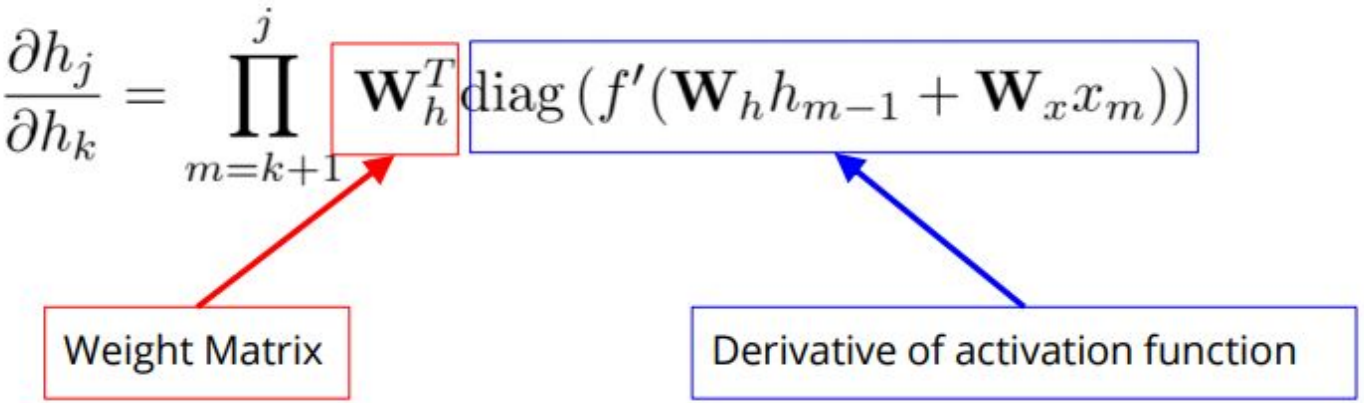
29/07/2017



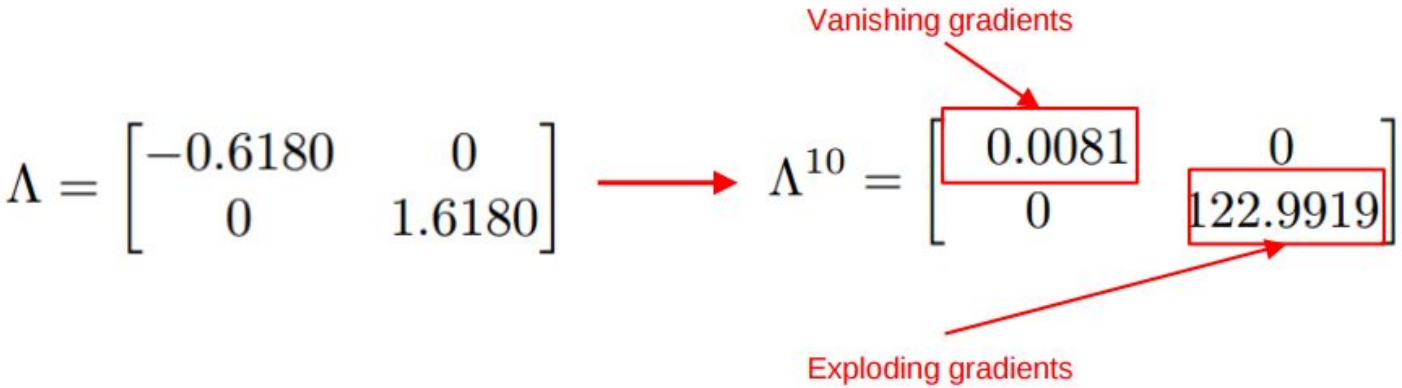
Microsoft Development Center Serbia

PETNICA Science Center

Problems with RNN



Vanishing and exploding gradients



- Gradient clipping

Minimal amount of LinAlg

$$W_h = Q^{-1} * \Lambda * Q$$

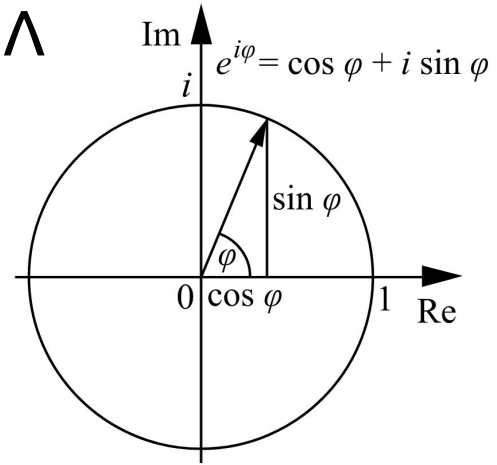
$$W_h^n = Q^{-1} * \Lambda^n * Q$$

$$\begin{bmatrix} R_1 & & & & \\ & \ddots & & 0 & \\ & & R_k & & \\ & & & \pm 1 & \\ & 0 & & & \ddots \\ & & & & & \pm 1 \end{bmatrix}$$

Making it stay unitary

- iRNN
- Projecting back when it drifts away
- Fix Q, learn parameter φ to generate Λ

$$W_h = Q^{-1} * \Lambda * Q$$



uRNN - the key idea

- M. Arjovsky, A. Shah, Y. Bengio, *Unitary Evolution Recurrent Neural Networks*, 2015
- Unitary * Unitary = Unitary
- Generate W as a product of several unitary matrices!
- 4 matrix types with linear number of parameters

uRNN

$$\mathbf{W} = \mathbf{D}_3\mathbf{R}_2\mathcal{F}^{-1}\mathbf{D}_2\mathbf{\Pi}\mathbf{R}_1\mathcal{F}\mathbf{D}_1.$$

- Makes gradients work
- O(n) parameters to learn in total
  - Larger depth, deeper temporal connections

Project goals

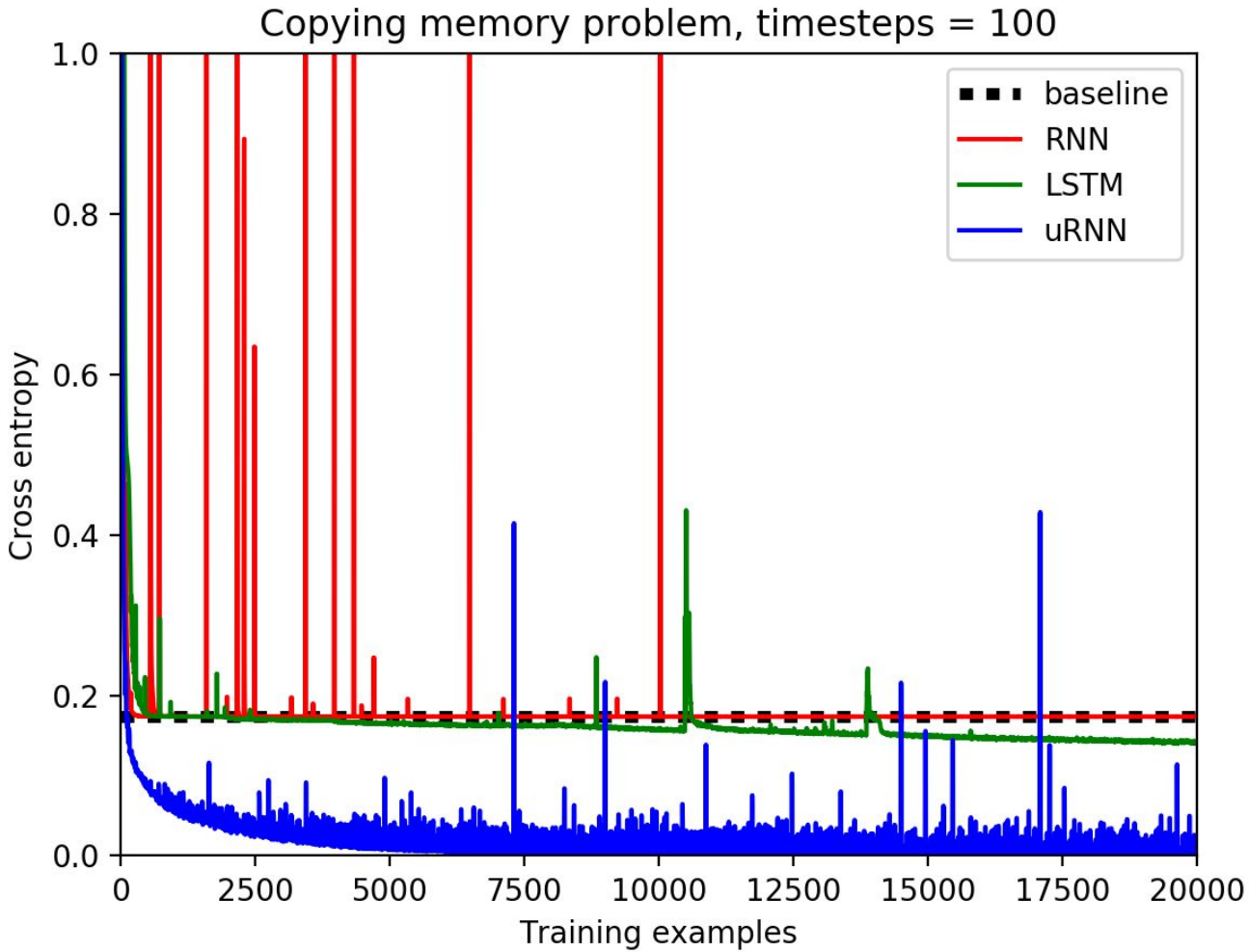
- Figure out the paper
- Devise an implementation
- Compare with LSTM/SimpleRNN
- Confirm the results
- ~~Try new problems and extensions~~

Copying Memory Problem

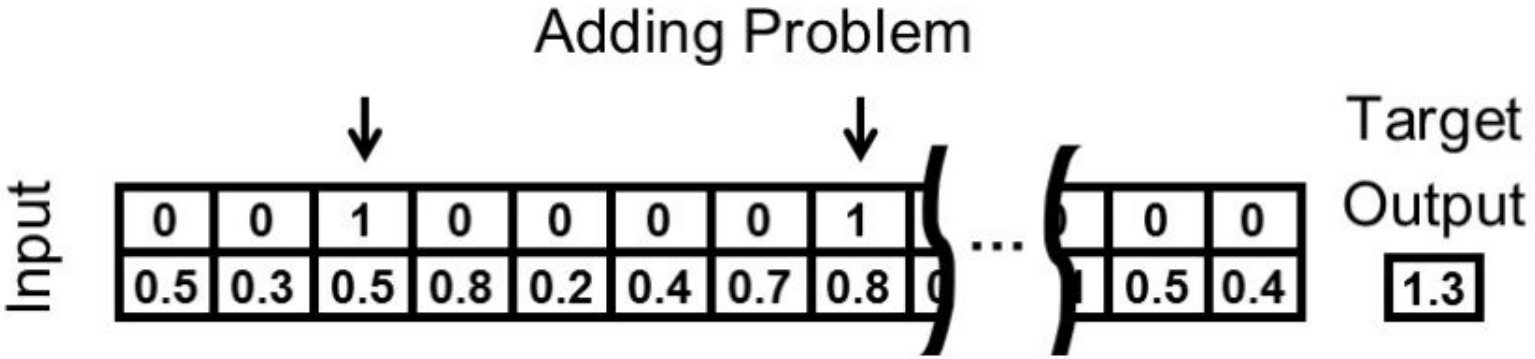
- Good for testing the understanding of very distant temporal connections

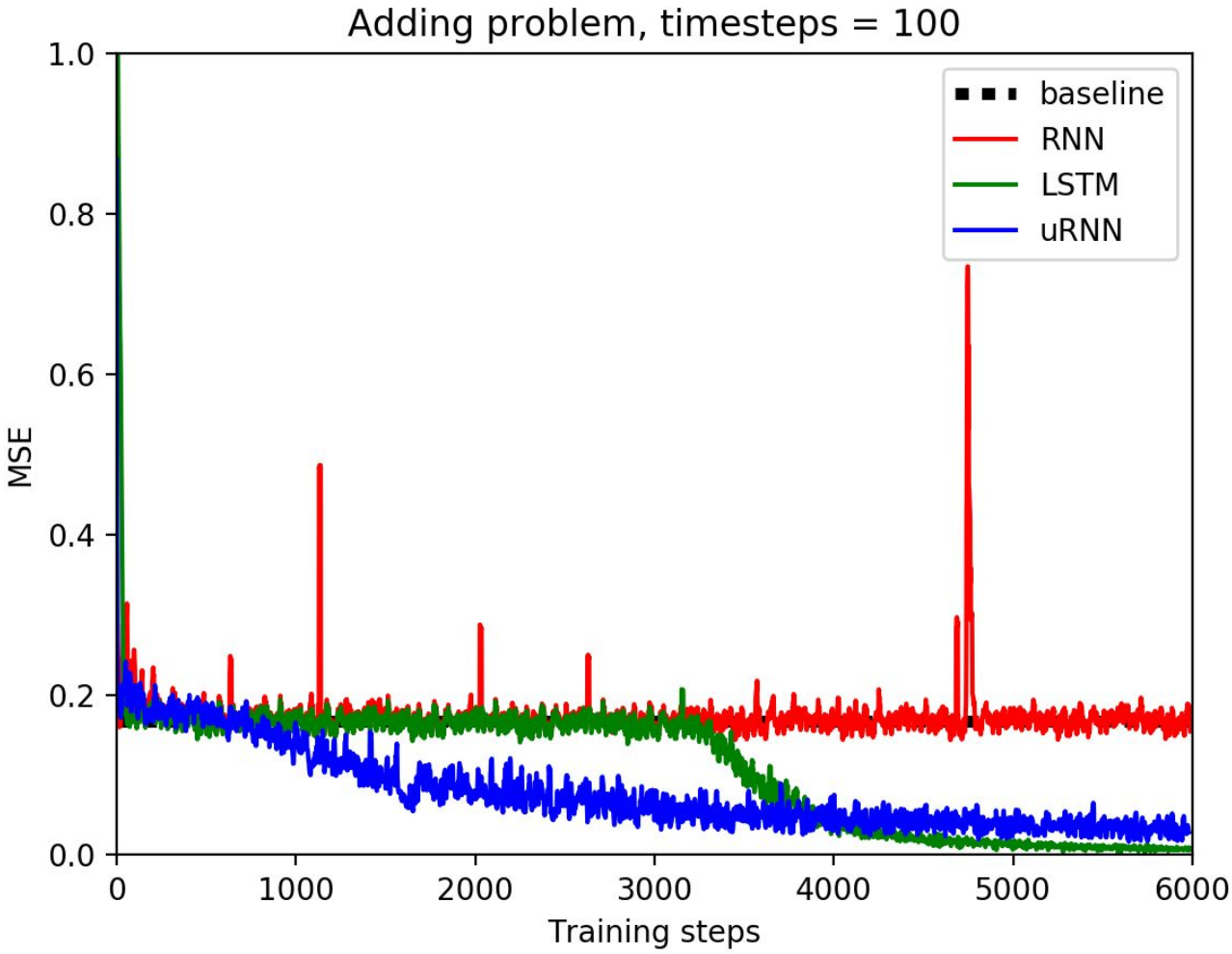
Input: 5 2 5 3 2 1 7 8 4 2 0 0 0 0 0 0 0 0 . . . 9 0 0 0 0 0 0 0 0 0
Output: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 . . . 0 5 2 5 3 2 1 7 8 4 2

- We can tune the “lag” to test the model’s limits


Copying memory problem, timesteps = 100
baseline
RNN
LSTM
uRNN
Cross entropy
1.0
0.8
0.6
0.4
0.2
0.0
0
2500
5000
7500
10000
12500
15000
17500
20000
Training examples

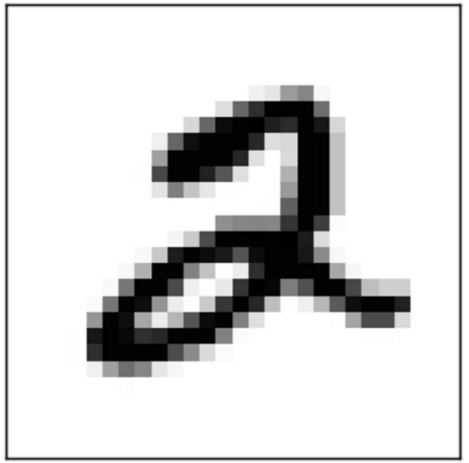
Adding Problem

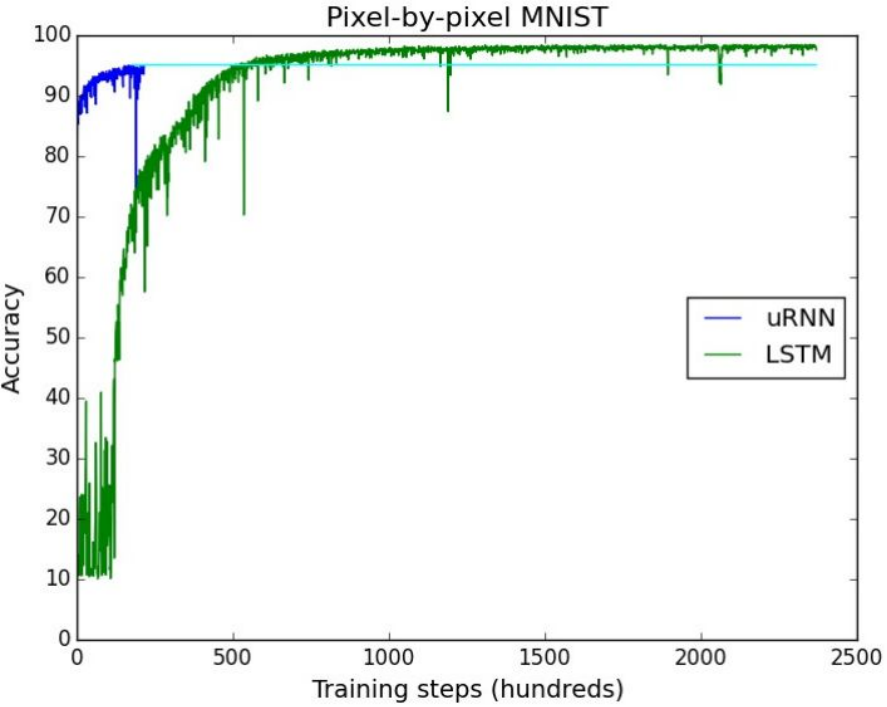


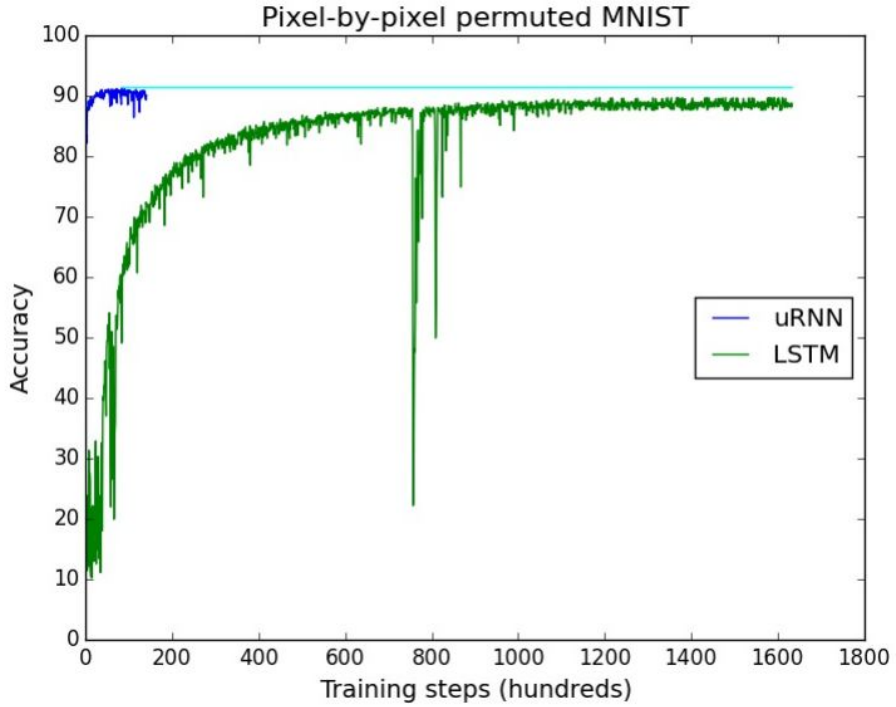

Adding problem, timesteps = 100
baseline
RNN
LSTM
uRNN
MSE
Training steps
1.0
0.8
0.6
0.4
0.2
0.0
0
1000
2000
3000
4000
5000
6000

Pixel-by-pixel MNIST

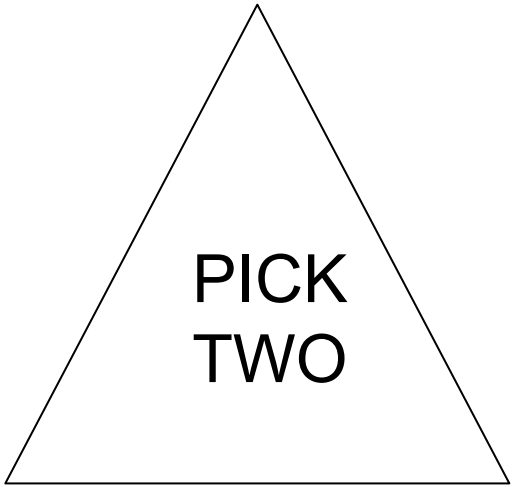
- RNN way of solving MNIST classification
- Results from the paper




Pixel-by-pixel MNIST
Accuracy
Training steps (hundreds)
uRNN
LSTM
0
10
20
30
40
50
60
70
80
90
100
0
500
1000
1500
2000
2500


Pixel-by-pixel permuted MNIST
Accuracy
Training steps (hundreds)
uRNN
LSTM
0
10
20
30
40
50
60
70
80
90
100
0
200
400
600
800
1000
1200
1400
1600
1800

The PSI:ML Triangle of Misery ™



Major takeaways

- Learned a lot:
  - Read a relevant paper -> understand -> implement
  - Tensorflow
  - First step towards proper ML intuition

Questions?

Thanks!