Learning Long-Term Dependencies with Gradient Descent is Difficult



Y. Bengio, P. Simard & P. Frasconi, IEEE Trans. Neural Nets, **1994**

June 23, 2016, ICML, New York City
Back-to-the-future Workshop
Yoshua Bengio
Montreal Institute for Learning Algorithms
Université de Montréal

PLUG: **Deep Learning**, MIT Press book in press,
Chapters will remain online

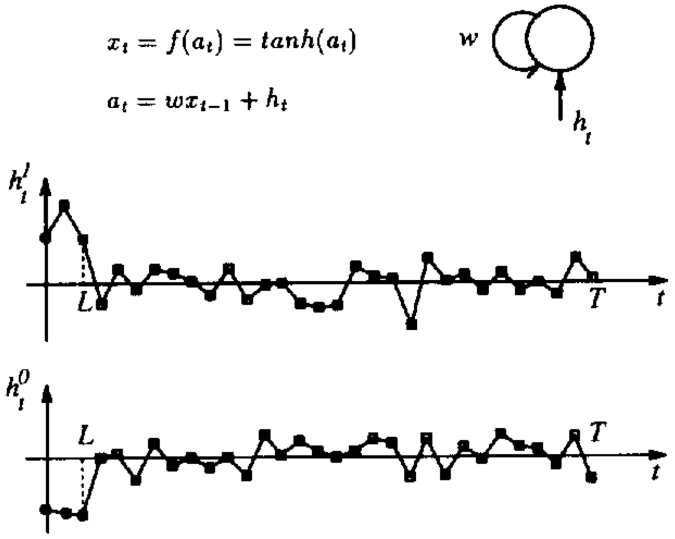
Université de Montréal

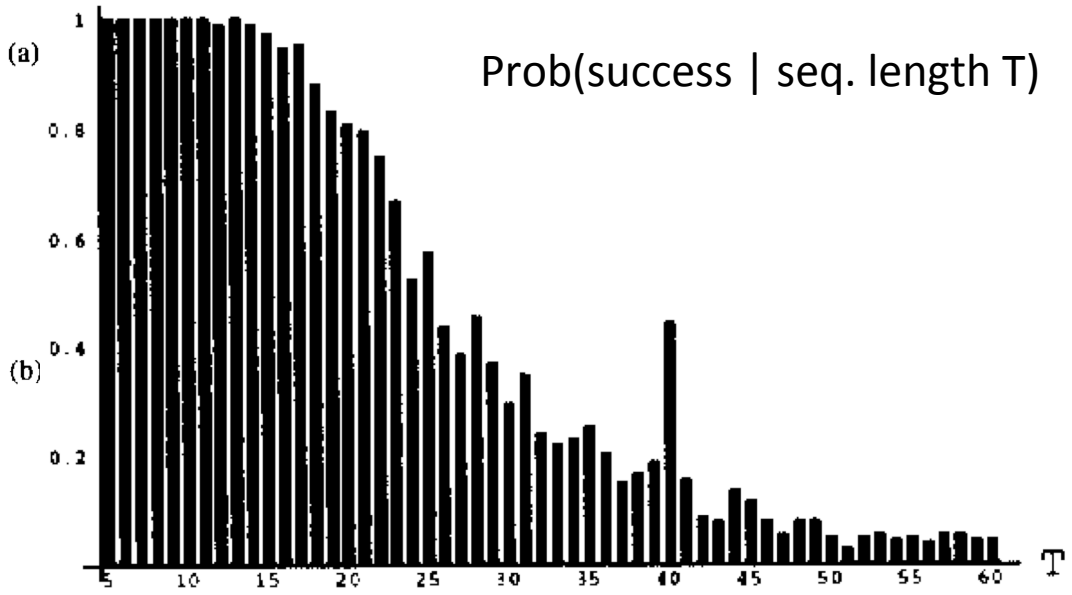
CIFAR | ICRA

MILA

Simple Experiments from 1992 while I was at MIT

- 2 categories of sequences
- Can the single tanh unit learn to store for T time steps 1 bit of information given by the sign of initial input?

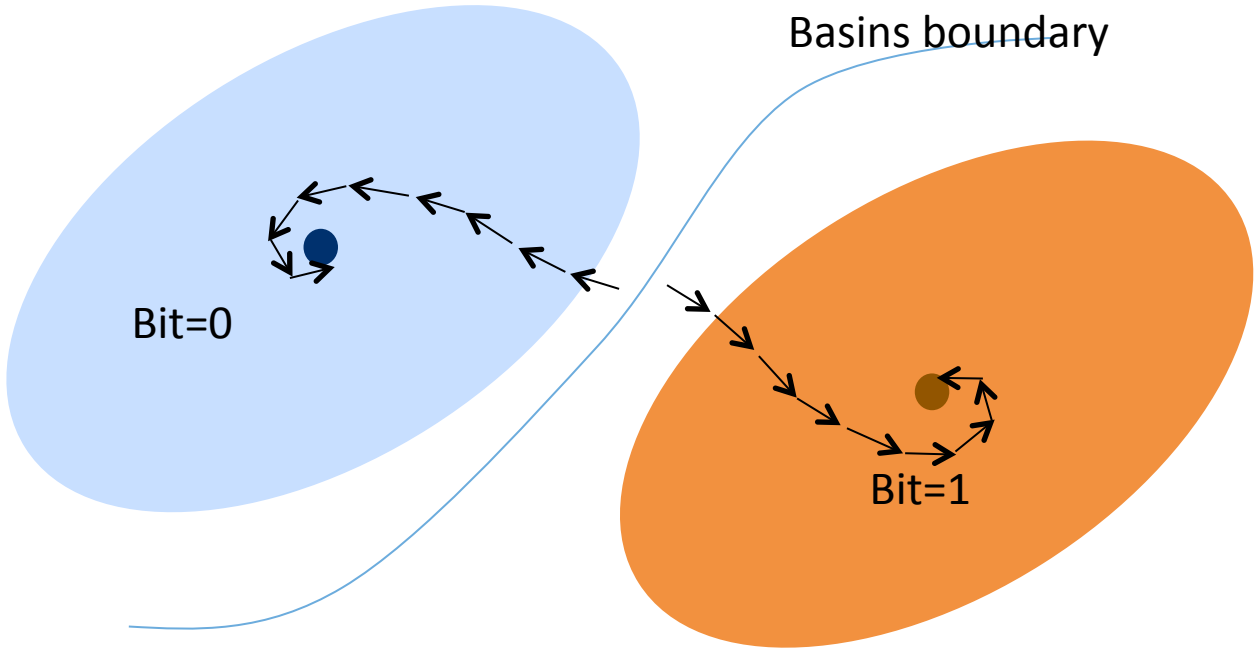




Prob(success | seq. length T)

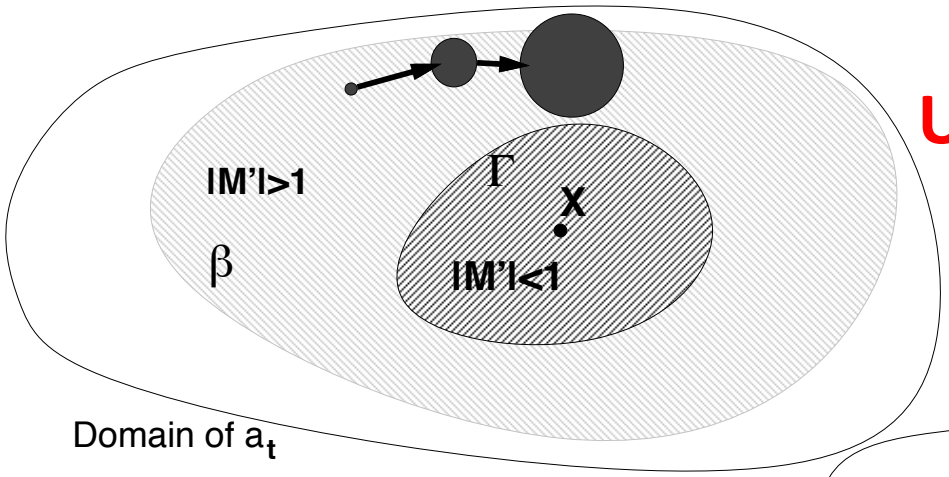
How to store 1 bit? Dynamics with multiple basins of attraction in some dimensions

- Some subspace of the state can store 1 or more bits of information if the dynamical system has multiple basins of attraction in some dimensions



Robustly storing 1 bit in the presence of bounded noise

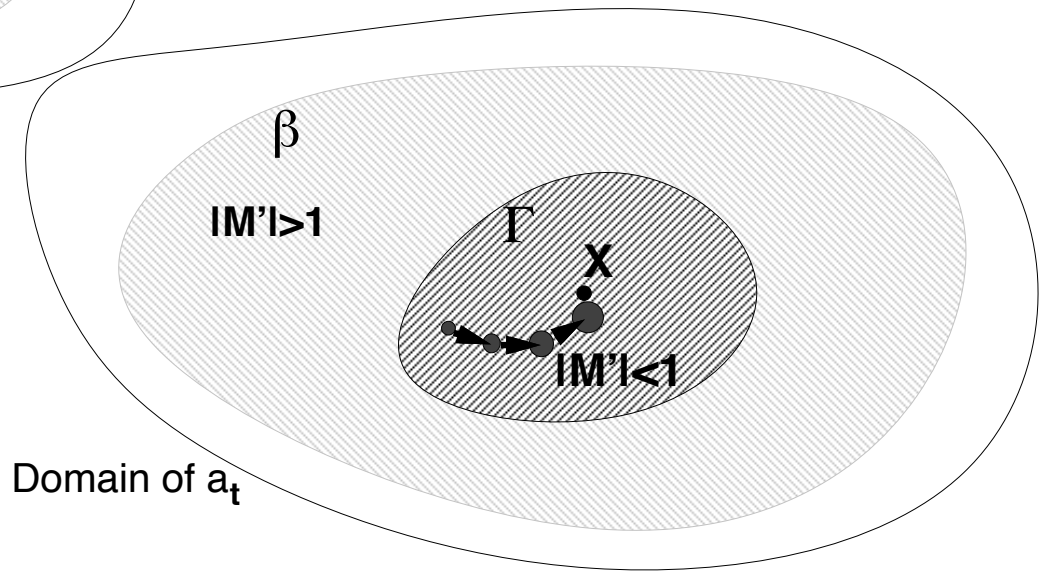
- With spectral radius > 1, noise can kick state out of attractor



UNSTABLE

- Not so with radius<1

CONTRACTIVE
→ STABLE



Storing Reliably → Vanishing gradients

- Reliably storing bits of information requires spectral radius<1
- The product of T matrices whose spectral radius is < 1 is a matrix whose spectral radius converges to 0 at exponential rate in T

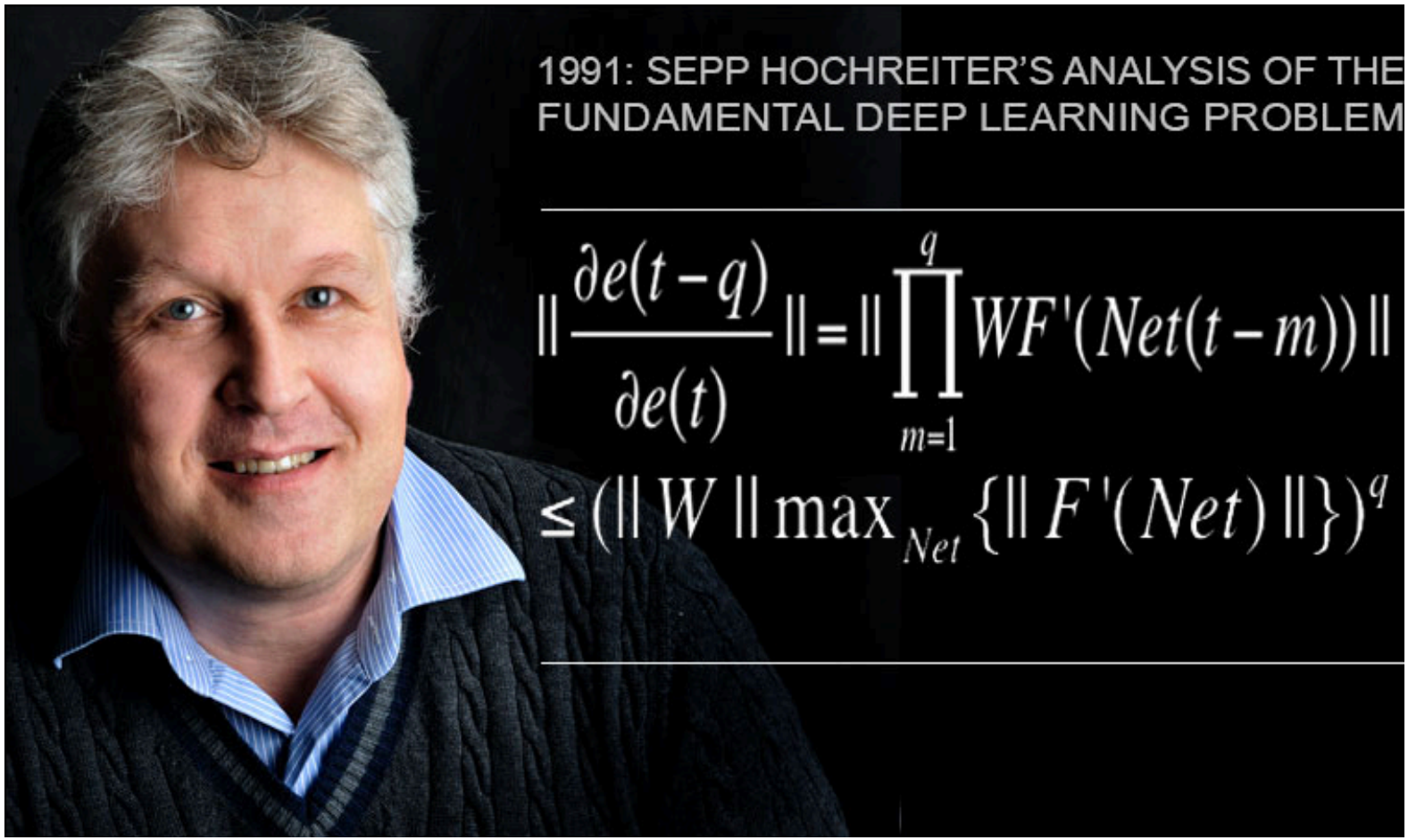
$$L = L(s_T(s_{T-1}(\dots s_{t+1}(s_t, \dots))))$$

$$\frac{\partial L}{\partial s_t} = \frac{\partial L}{\partial s_T}\frac{\partial s_T}{\partial s_{T-1}}\cdots\frac{\partial s_{t+1}}{\partial s_t}$$

- If spectral radius of Jacobian is < 1 → propagated gradients vanish

Vanishing or Exploding Gradients

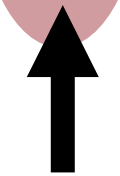
- Hochreiter's 1991 MSc thesis (in German) had independently discovered that backpropagated gradients in RNNs tend to either vanish or explode as sequence length increases



Why it hurts gradient-based learning

- Long-term dependencies get a weight that is exponentially smaller (in T) compared to short-term dependencies

$$\frac{\partial C_t}{\partial W} = \sum_{\tau \leq t} \frac{\partial C_t}{\partial a_\tau} \frac{\partial a_\tau}{\partial W} = \sum_{\tau \leq t} \frac{\partial C_t}{\partial a_t} \frac{\partial a_t}{\partial a_\tau} \frac{\partial a_\tau}{\partial W}$$

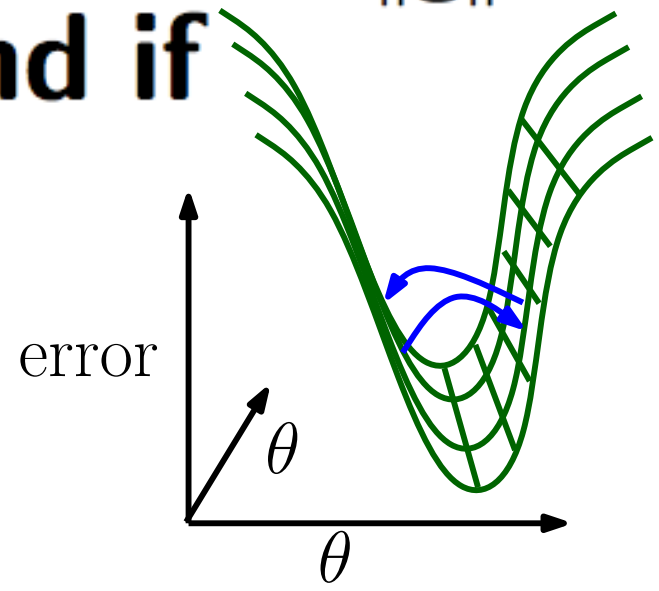


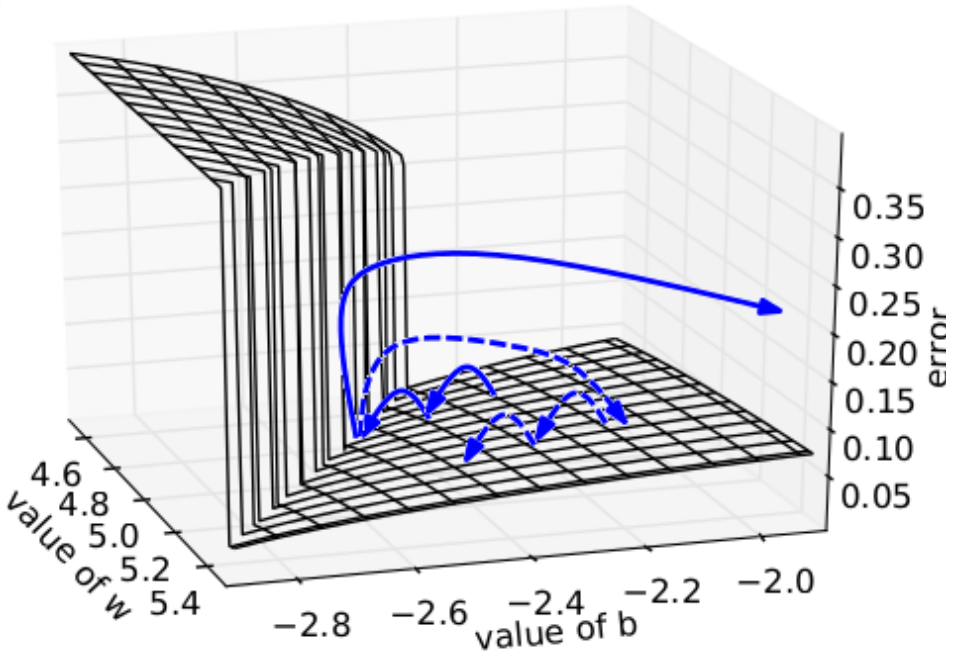
Becomes exponentially smaller for longer time differences, when spectral radius < 1

Dealing with Gradient Explosion by Gradient Norm Clipping

(Mikolov thesis 2012;
Pascanu, Mikolov, Bengio, ICML 2013)

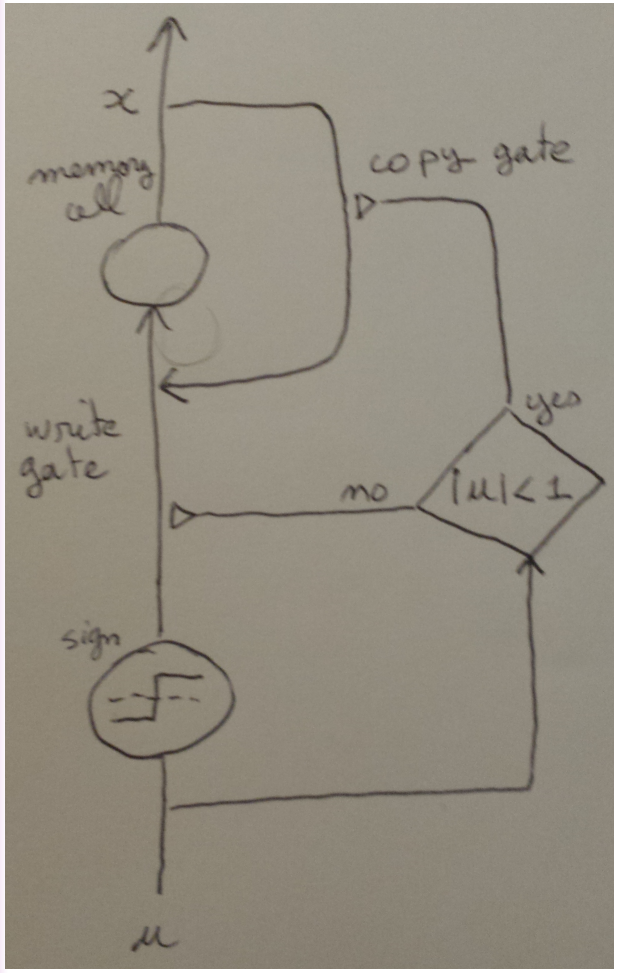
$$\hat{\mathbf{g}} \leftarrow \frac{\partial error}{\partial \theta}$$
$$\textbf{if } \|\hat{\mathbf{g}}\| \geq threshold \textbf{ then}$$
$$\hat{\mathbf{g}} \leftarrow \frac{threshold}{\|\hat{\mathbf{g}}\|}\hat{\mathbf{g}}$$
$$\textbf{end if}$$





Conference version (1993) of the 1994 paper by the same authors had a predecessor of GRU and targetprop

(The problem of learning long-term dependencies in recurrent networks, Bengio, Frasconi & Simard ICNN'1993)



IV. A Trainable Flip-Flop

- Flip-flop unit to store 1 bit, with gating signal to control when to write

$$x_{t+1} = f(x_t, u_t)$$

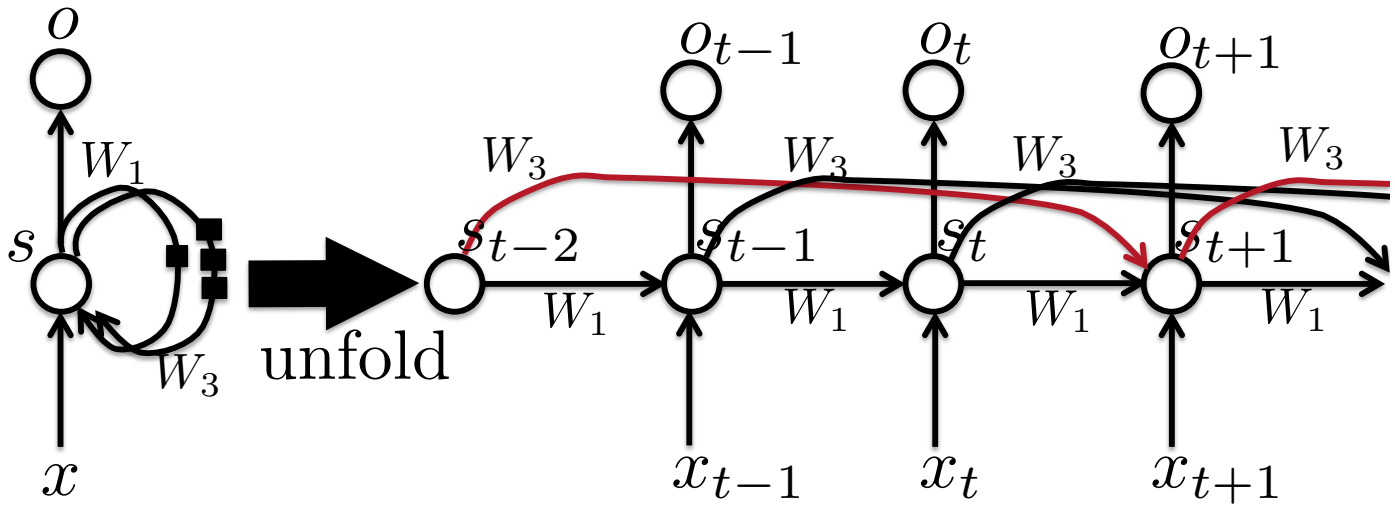
$$f(x,u) = \begin{cases} 1 & \text{if } |u| < 1 \text{ and } x \geq 0 \\ & \text{or if } u \geq 1 \\ -1 & \text{otherwise} \end{cases} \quad (8)$$

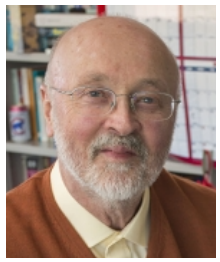
- Pseudo-backprop through it by a form of targetprop

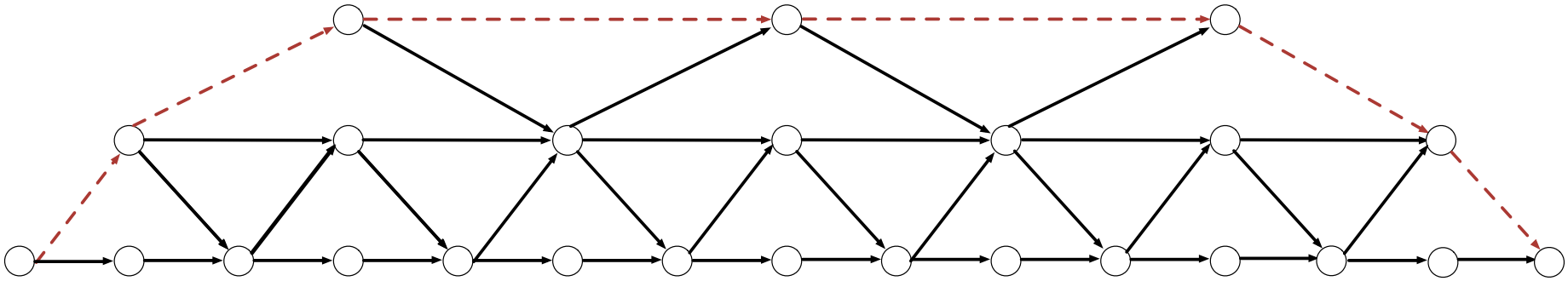
$$\Delta x(\Delta f, u) = \begin{cases} \Delta f & \text{if } |u| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Bypassing nonlinearities to learn longer term dependencies

- Delays *(Lin et al & Giles 1995)*
- Multiple time scales *(Elhihi & Bengio NIPS 1995)*





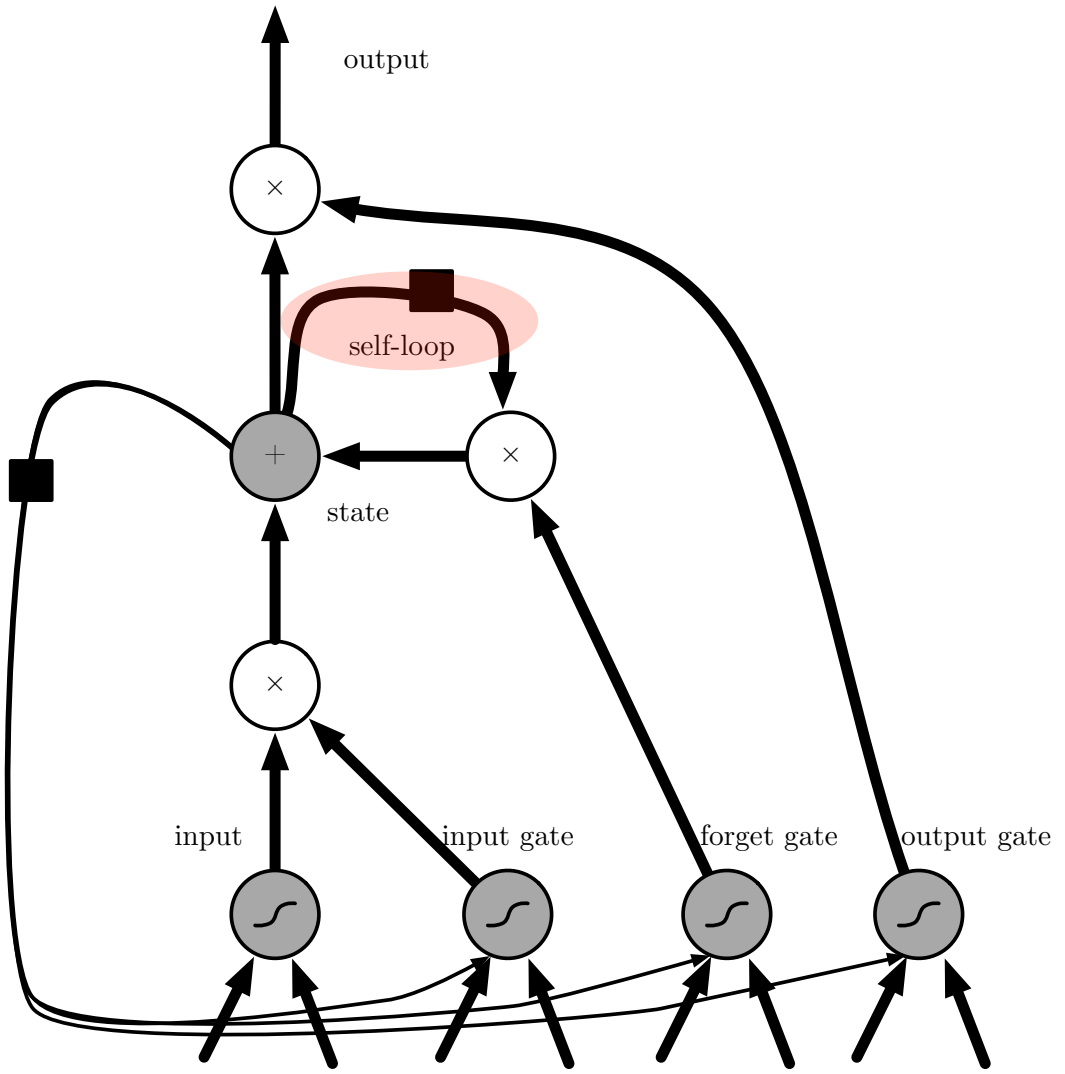


Fighting the vanishing gradient: LSTM & GRU

(Hochreiter 1991); first version of the LSTM, called Neural Long-Term Storage with self-loop

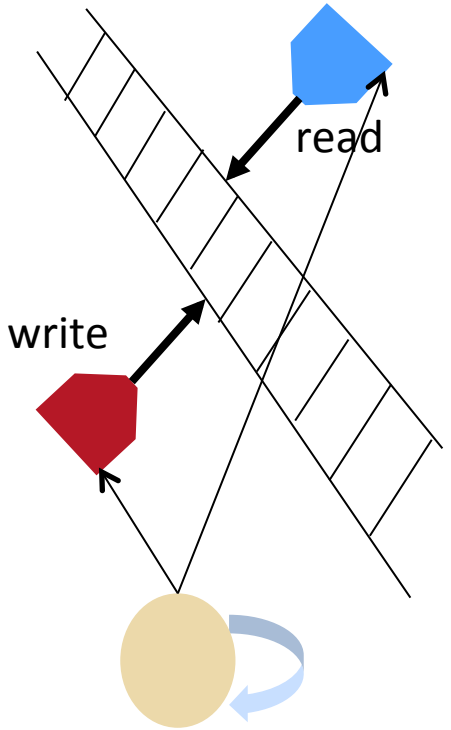
- Create a path where gradients can flow for longer with a self-loop
- Corresponds to an eigenvalue of Jacobian slightly less than 1
- LSTM is now **heavily used** *(Hochreiter & Schmidhuber 1997)*
- GRU light-weight version *(Cho et al 2014)*

LSTM: (Hochreiter & Schmidhuber 1997)



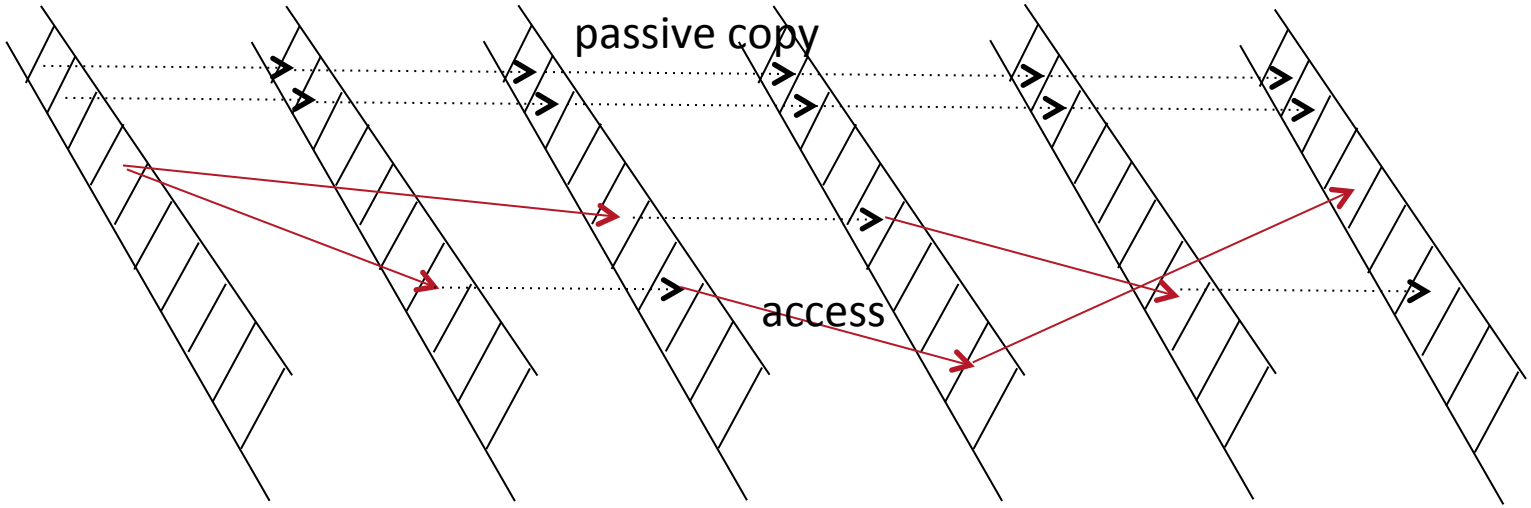
Fast Forward 20 years: Attention Mechanisms for Memory Access

- Neural Turing Machines *(Graves et al 2014)*
- and Memory Networks *(Weston et al 2014)*
- Use a content-based attention mechanism *(Bahdanau et al 2014)* to control the read and write access into a memory
- The attention mechanism outputs a softmax over memory locations



Large Memory Networks: Sparse Access Memory for Long-Term Dependencies

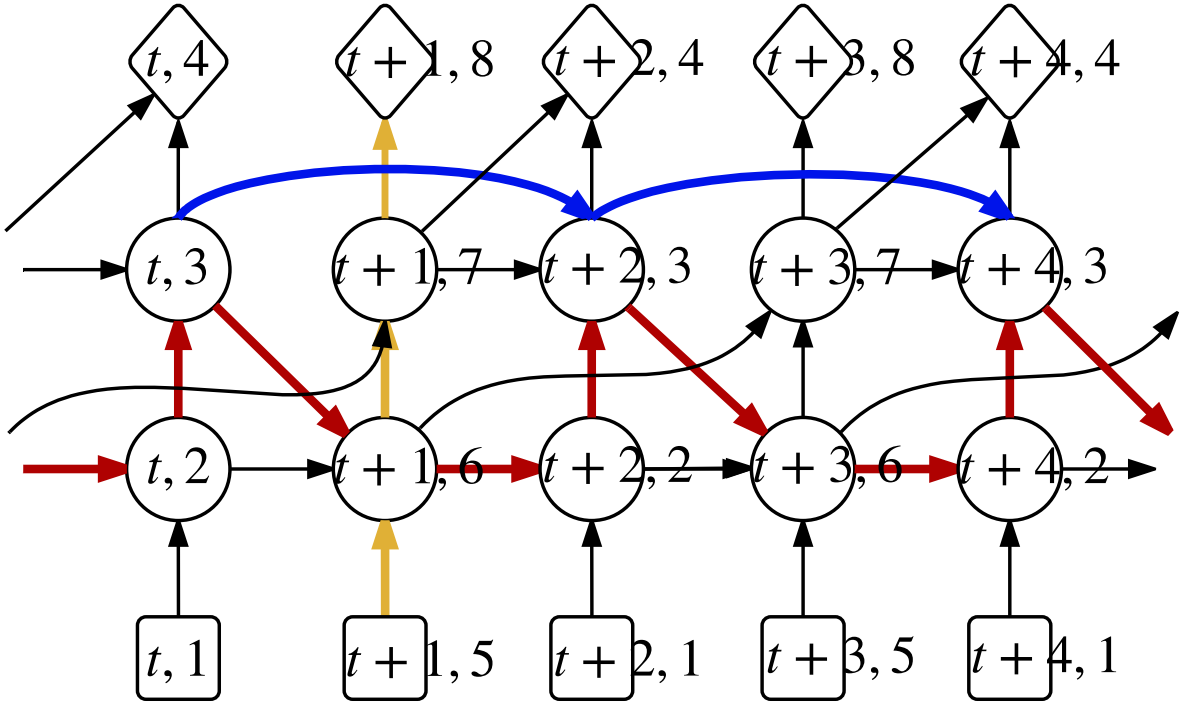
- A mental state stored in an external memory can stay for arbitrarily long durations, until it is overwritten (partially or not)
- Forgetting = vanishing gradient.
- Memory = **higher-dimensional state**, avoiding or reducing the need for forgetting/vanishing



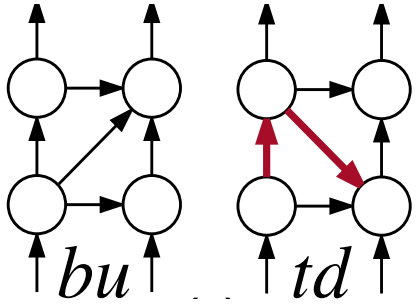
Designing the RNN Architecture

(Zhang et al 2016)

- **Recurrent depth**: max path length divided by sequence length
- **Feedforward depth**: max length from input to nearest output
- **Skip coefficient**: shortest path length divided sequence length



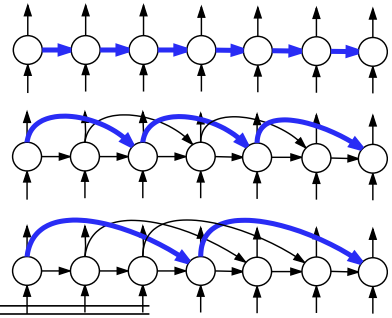
It makes a difference



- Impact of change in recurrent depth

DATASET	MODELS\ARCHS	*sh*	*st*	*bu*	*td*
PennTreebank	*tanh* RNN	1.54	1.59	1.54	**1.49**
text8	*tanh* RNN-SMALL	1.80	1.82	1.80	**1.77**
	tanh RNN-LARGE	1.69	1.67	1.64	**1.59**
	LSTM-SMALL	1.65	1.66	1.65	**1.63**
	LSTM-LARGE	1.52	1.53	1.52	**1.49**

- Impact of change in skip coefficient



RNN(*tanh*)	s = 1	s = 5	s = 9	s = 13	s = 21
MNIST	34.9	46.9	74.9	85.4	**87.8**
	s = 1	s = 3	s = 5	s = 7	s = 9
*p*MNIST	49.8	79.1	84.3	**88.9**	88.0

LSTM	s = 1	s = 3	s = 5	s = 7	s = 9
MNIST	56.2	**87.2**	86.4	86.4	84.8
	s = 1	s = 3	s = 4	s = 5	s = 6
*p*MNIST	28.5	25.0	60.8	62.2	**65.9**

Model	MNIST	*p*MNIST
*i*RNN[25]	97.0	≈82.0
*u*RNN[24]	95.1	91.4
LSTM[24]	**98.2**	88.0
RNN(*tanh*)[25]	≈35.0	≈35.0
*s*tanh(s = 21, 11)	98.1	**94.0**

Architecture, s		(1), 1	(2), 1	(3), $\frac{k}{2}$	(4), k
MNIST	k = 17	39.5	39.4	54.2	**77.8**
	k = 21	39.5	39.9	69.6	**71.8**
*p*MNIST	k = 5	55.5	66.6	74.7	**81.2**
	k = 9	55.5	71.1	78.6	**86.9**

Table 2: Results for MNIST/*p*MNIST. **Top-left**: test accuracies with different s for *tanh* RNN. **Top-right**: test accuracies with different s for LSTM. **Bottom**: compared to previous results. **Bottom-right**: test accuracies for architectures (1), (2), (3) and (4) for *tanh* RNN.

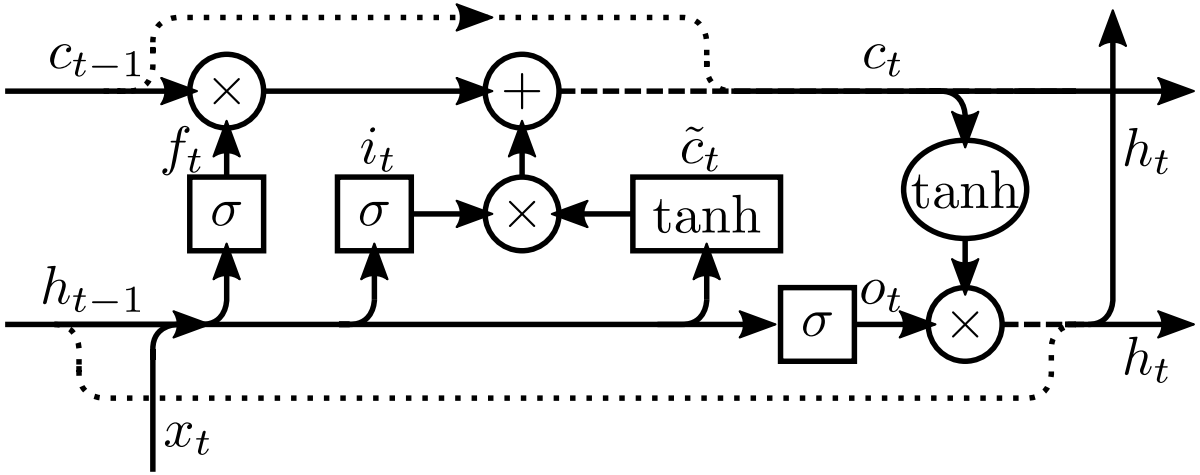
New Ideas to Help Information Propagation

- Unitary matrices: all e-values of matrix are 1

(Arjowski, Amar & Bengio ICML 2016)

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

- Zoneout: randomly choose to simply copy the state unchanged



(Krueger et al 2016, submitted)