From Attention to Memory and towards Longer-Term Dependencies

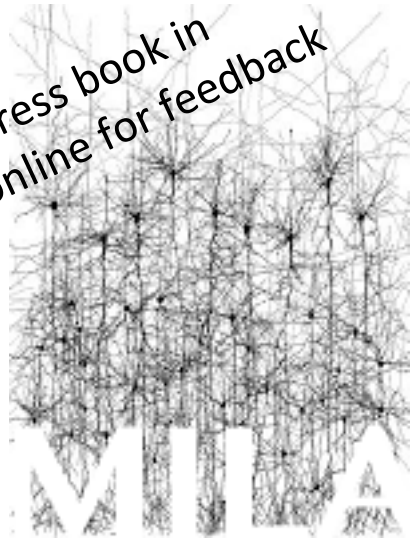
Yoshua Bengio

December 12, 2015

NIPS'2015 Reasoning, Attention & Memory Workshop

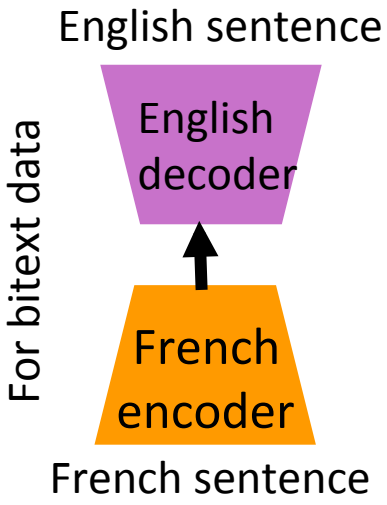
Université de Montréal

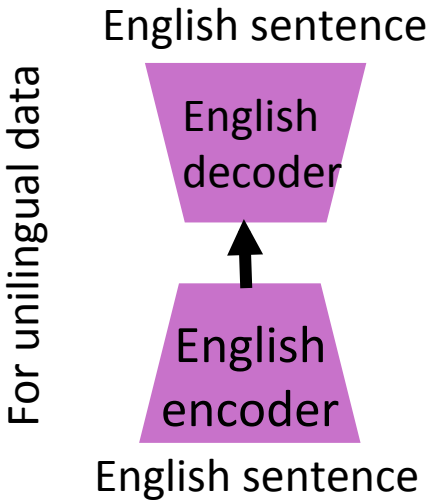
PLUG: **Deep Learning**, MIT Press book in preparation, draft chapters online for feedback

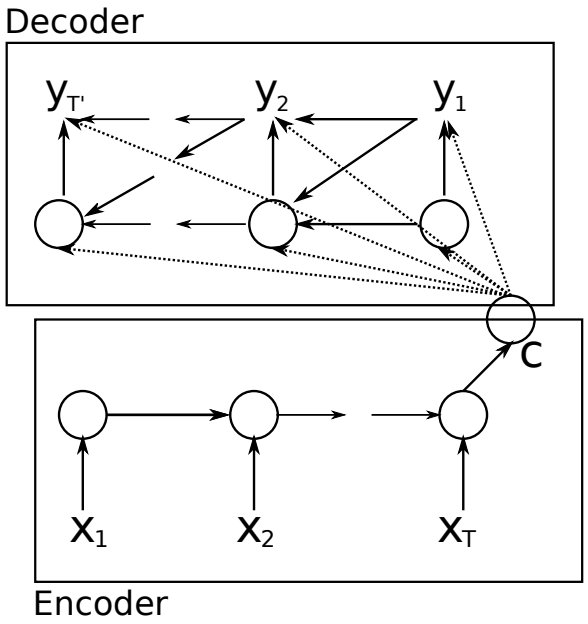


Encoder-Decoder Framework

- Intermediate representation of meaning
  = 'universal representation'
- Encoder: from word sequence to sentence representation
- Decoder: from representation to word sequence distribution

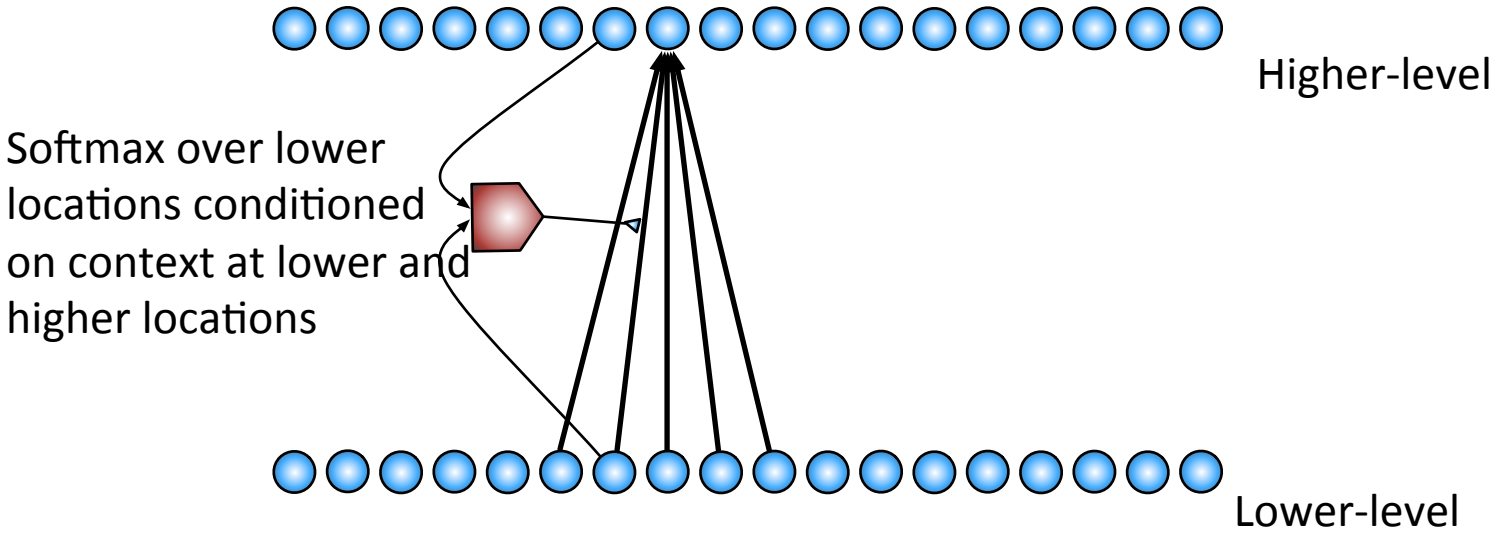






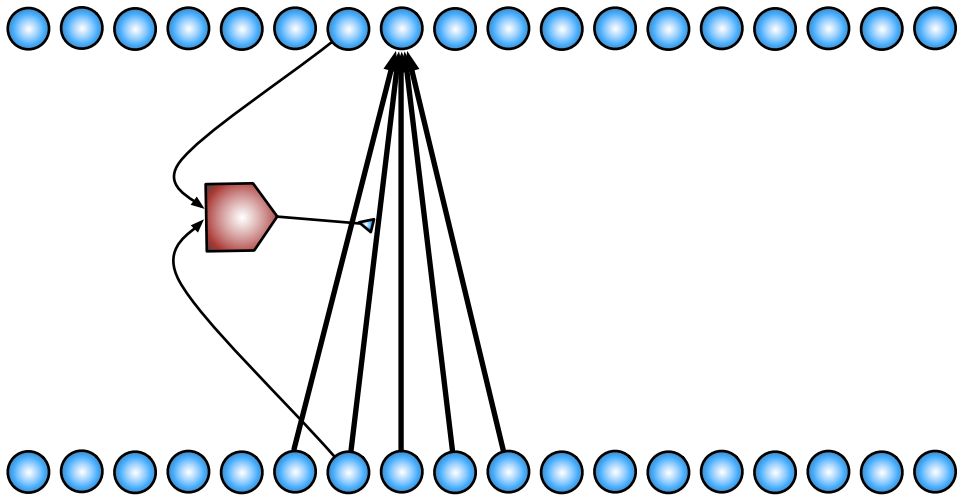
Attention Mechanism for Deep Learning

- Consider an input (or intermediate) sequence or image
- Consider an upper level representation, which can choose « where to look », by assigning a weight or probability to each input position, as produced by an MLP, applied at each position



Content-Based & Location-Based Attention Mechanisms

- *(Graves 2013)*: location-based, handwriting generation
- *(Bahdanau et al 2014)*: content-based, machine translation
- *(Weston et al 2014, Graves et al 2014, Chorowski et al 2014 & NIPS 2015)*: content-based + location-based, speech recognition



End-to-End Machine Translation with Recurrent Nets and Attention Mechanism

(Bahdanau et al 2014, Jean et al 2014, Gulcehre et al 2015, Jean et al 2015)

- Reached the state-of-the-art in one year, from scratch

(a) **English→French (WMT-14)**

	NMT(A)	Google	P-SMT
NMT	32.68	30.6*	**37.03**$^\bullet$
+Cand	33.28	–	
+UNK	33.99	32.7°	
+Ens	**36.71**	**36.9**$^\circ$	

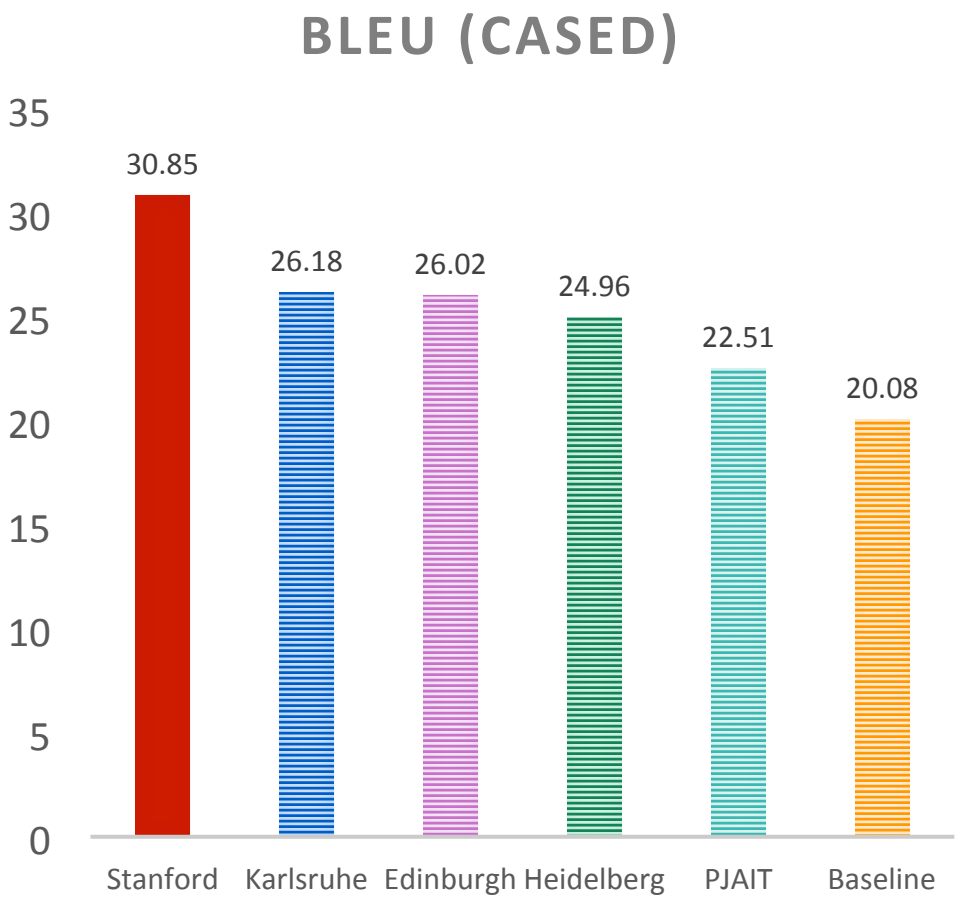
(b) **English→German (WMT-15)**

Model	Note
24.8	Neural MT
24.0	U.Edinburgh, Syntactic SMT
23.6	LIMSI/KIT
22.8	U.Edinburgh, Phrase SMT
22.7	KIT, Phrase SMT

(c) **English→Czech (WMT-15)**

Model	Note
18.3	Neural MT
18.2	JHU, SMT+LM+OSM+Sparse
17.6	CU, Phrase SMT
17.4	U.Edinburgh, Phrase SMT
16.1	U.Edinburgh, Syntactic SMT

IWSLT 2015 – Luong & Manning (2015)
TED talk MT, English-German

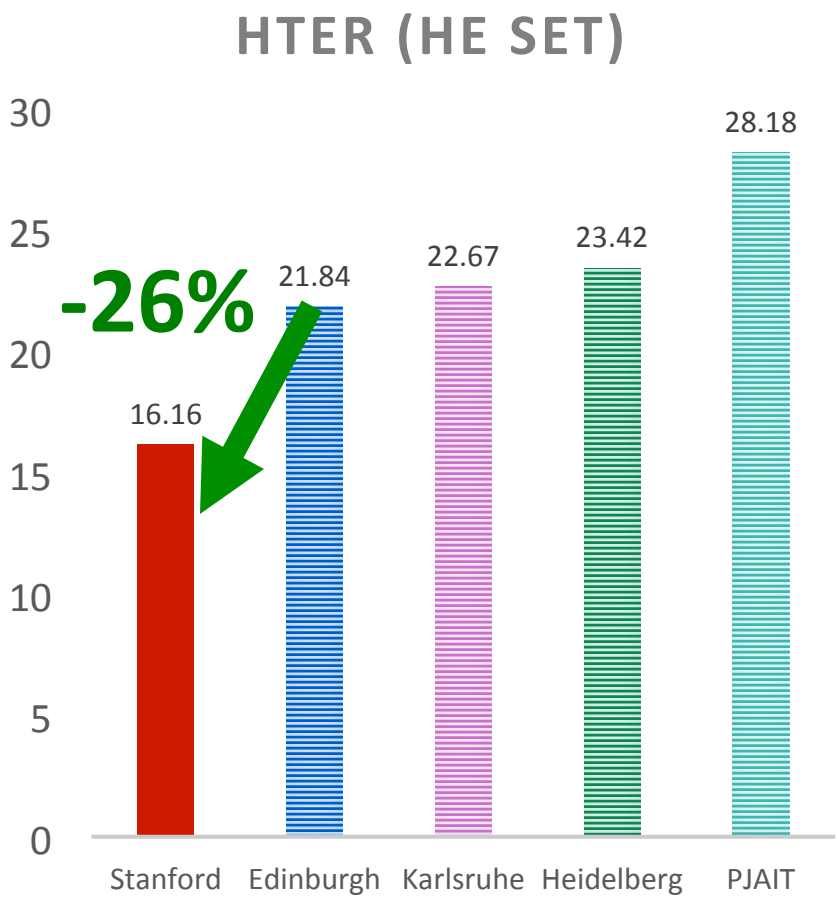
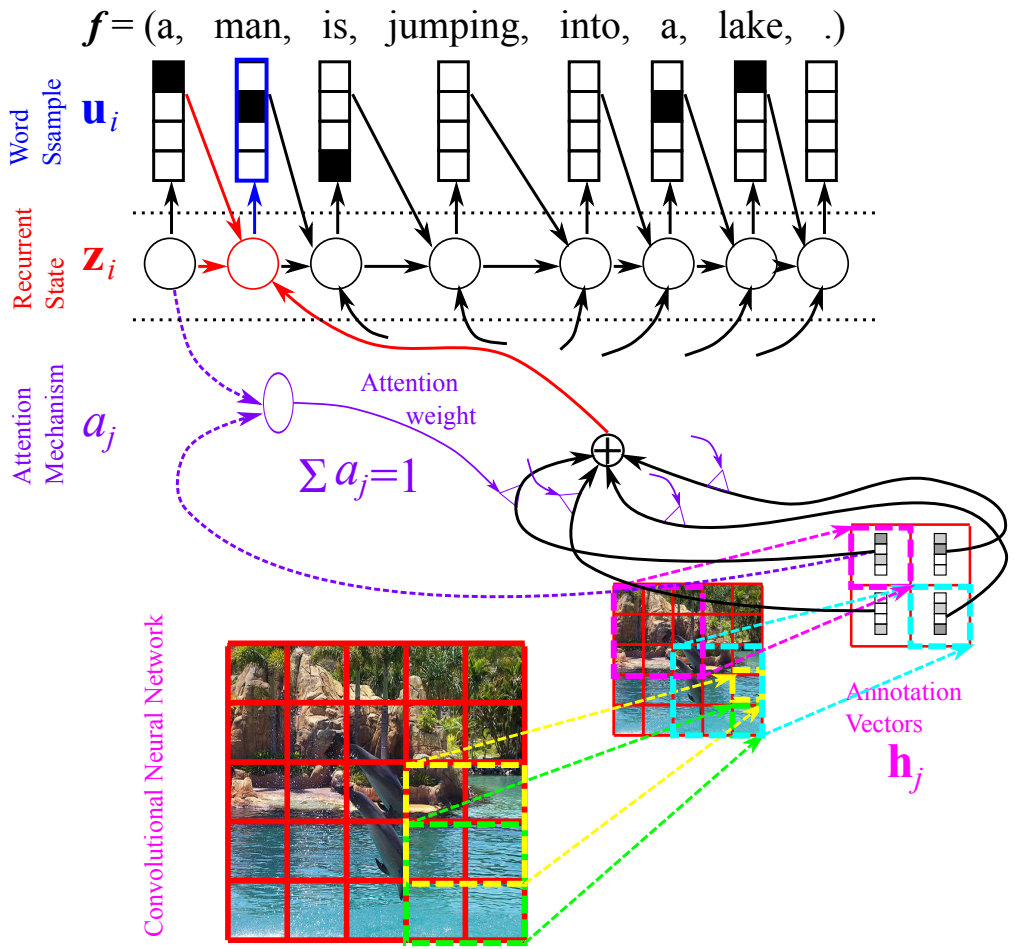
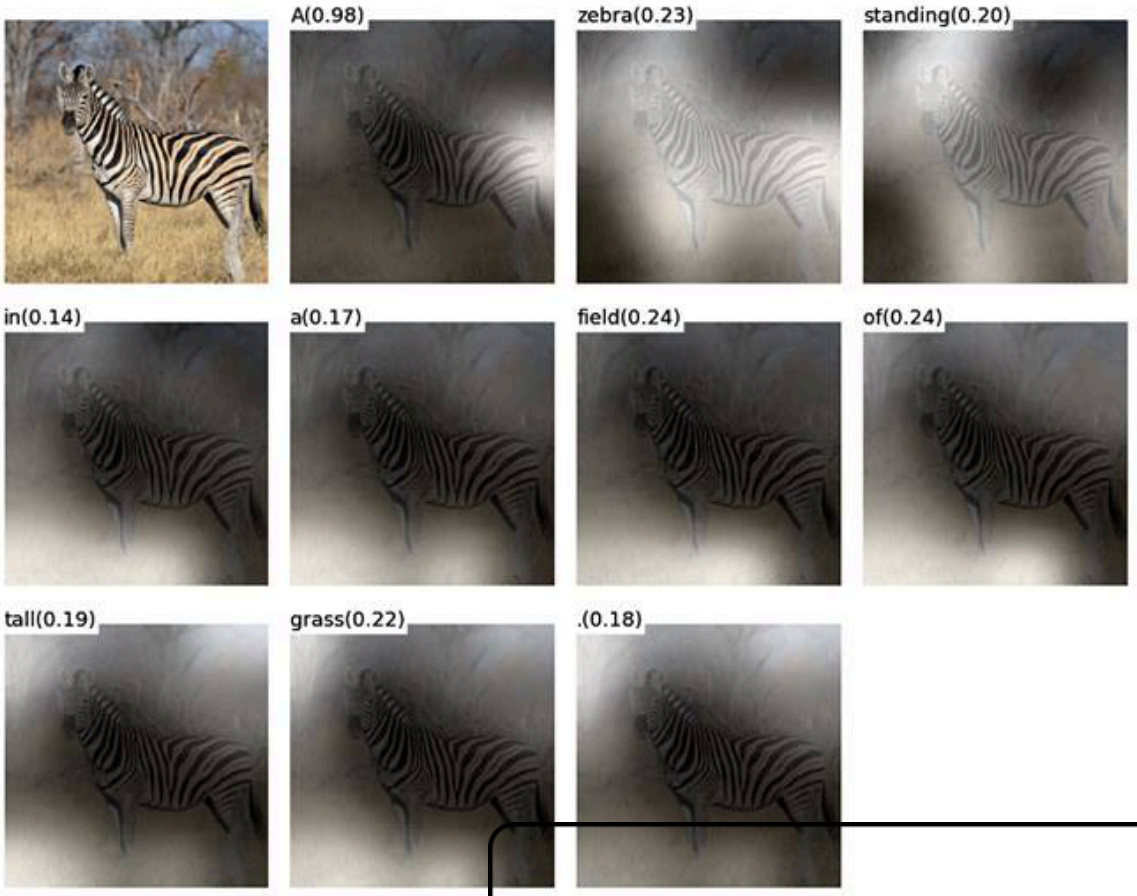


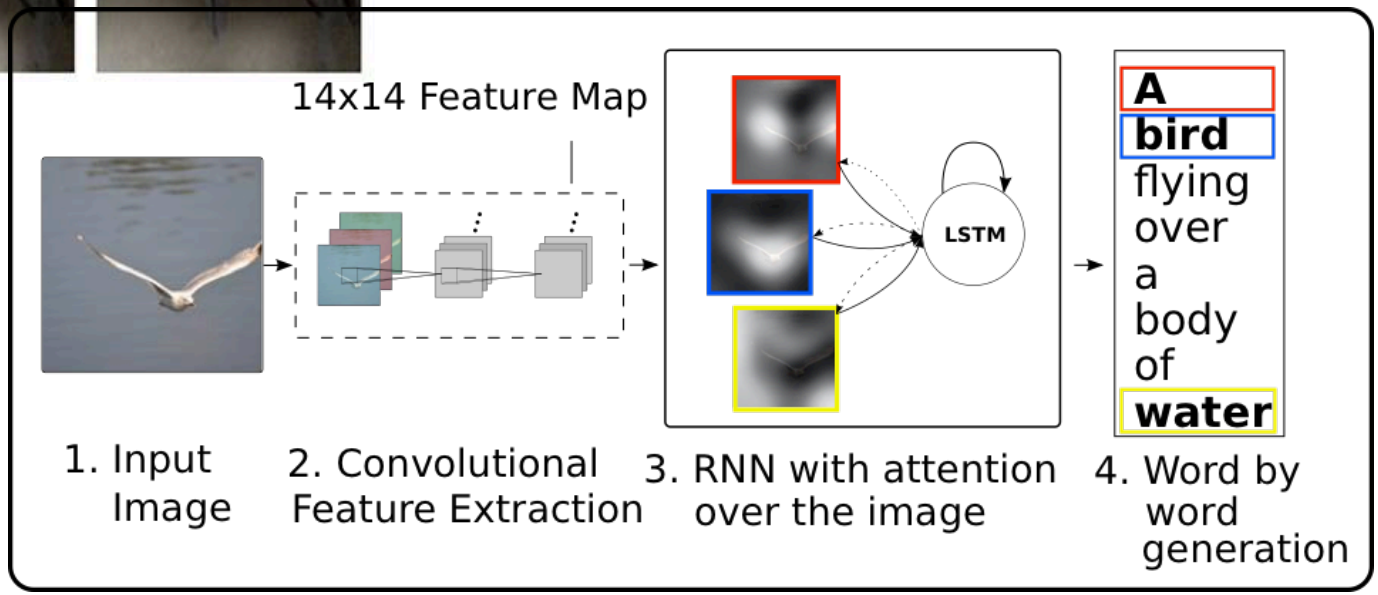
Image-to-Text: Caption Generation with Attention



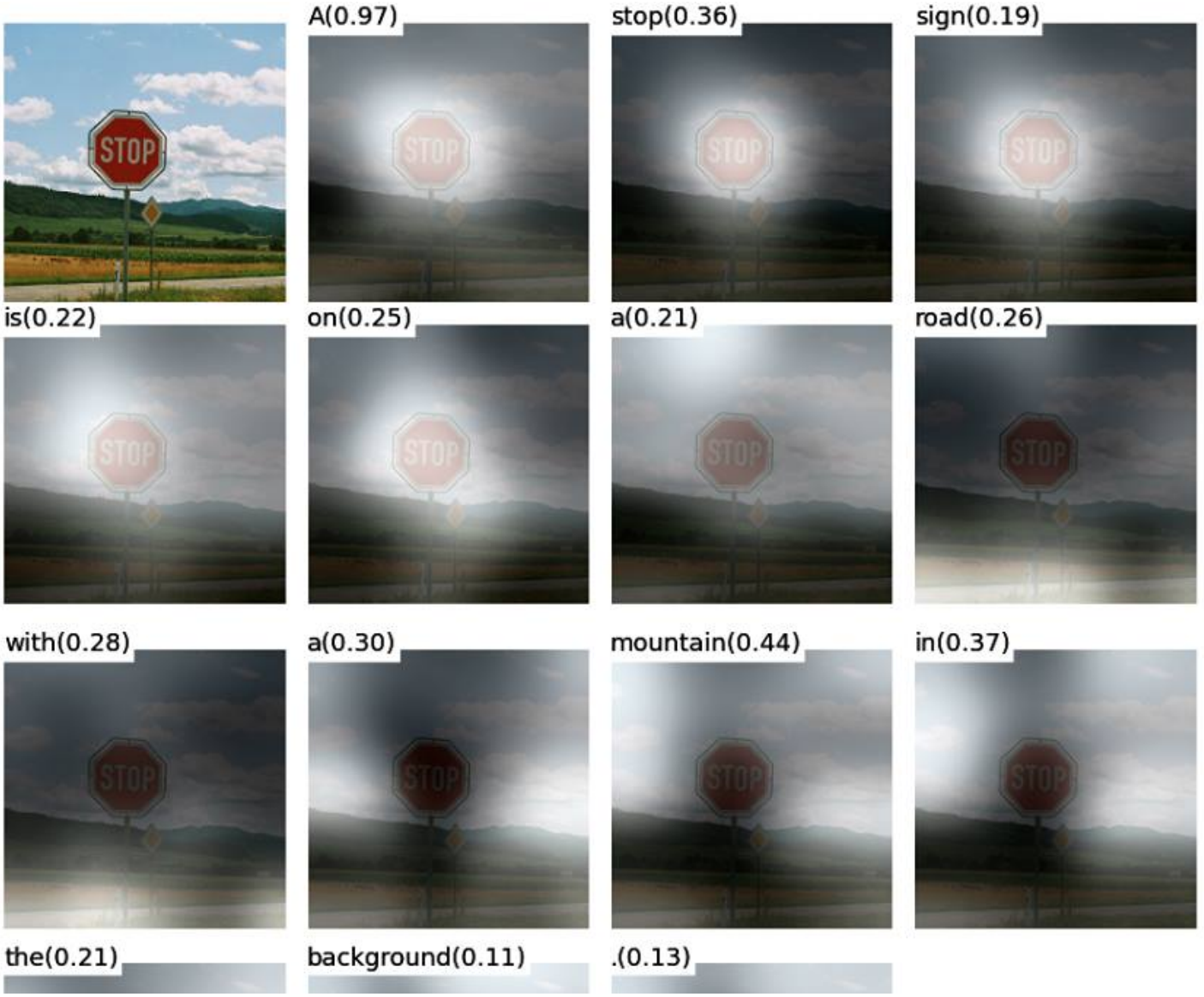
(Xu et al., 2015), (Yao et al., 2015)



Paying Attention to Selected Parts of the Image While Uttering Words



Speaking about what one sees



The Good

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

And the Bad

A large white bird standing in a forest.

A woman holding a clock in her hand.

A man wearing a hat and
a hat on a skateboard.

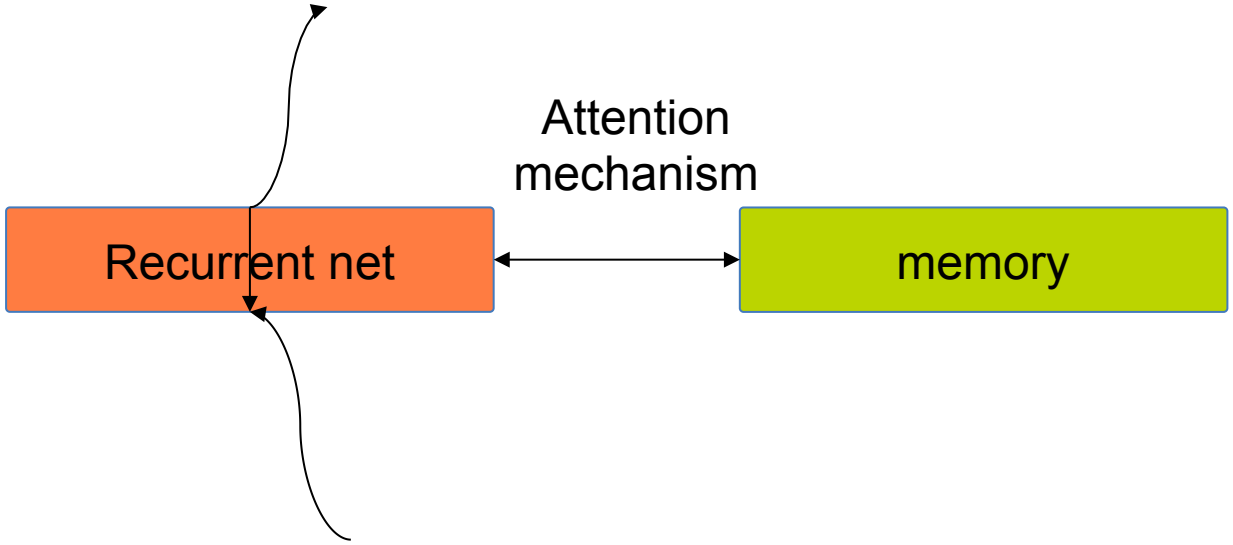
A person is standing on a beach
with a surfboard.

A woman is sitting at a table
with a large pizza.

A man is talking on his cell phone
while another man watches.

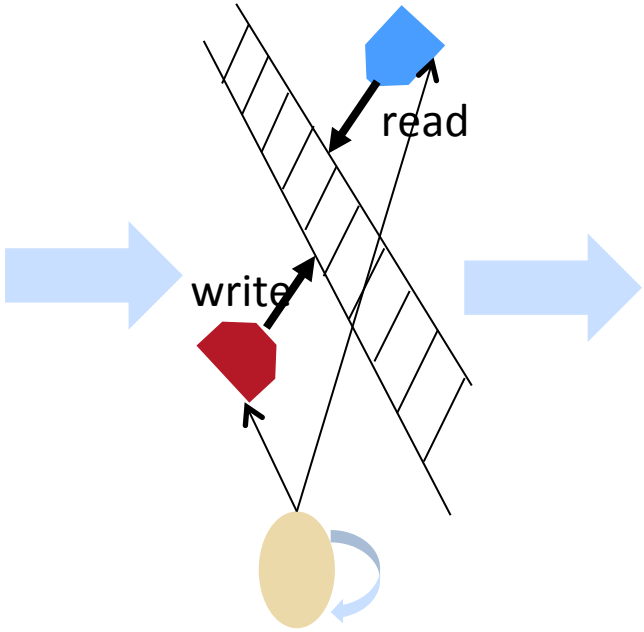
Attention on Memory Elements

- **Recurrent networks cannot remember things for very long**
  - The cortex only remember things for 20 seconds
- **We need a "hippocampus" (a separate memory module)**
  - LSTM [Hochreiter 1997], registers
  - **Memory networks** [Weston et 2014] (FAIR), associative memory
  - NTM [Graves et al. 2014], "tape".



Ongoing Project: Knowledge Extraction

- Learn to fill the memory network from natural language descriptions of facts
- Force the neural net to understand language
- Extract knowledge from documents into a usable form



Long-Term Dependencies

- The RNN gradient is a product of Jacobian matrices, each associated with a step in the forward computation. To store information robustly in a finite-dimensional state, the dynamics must be contractive [Bengio et al 1994].

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

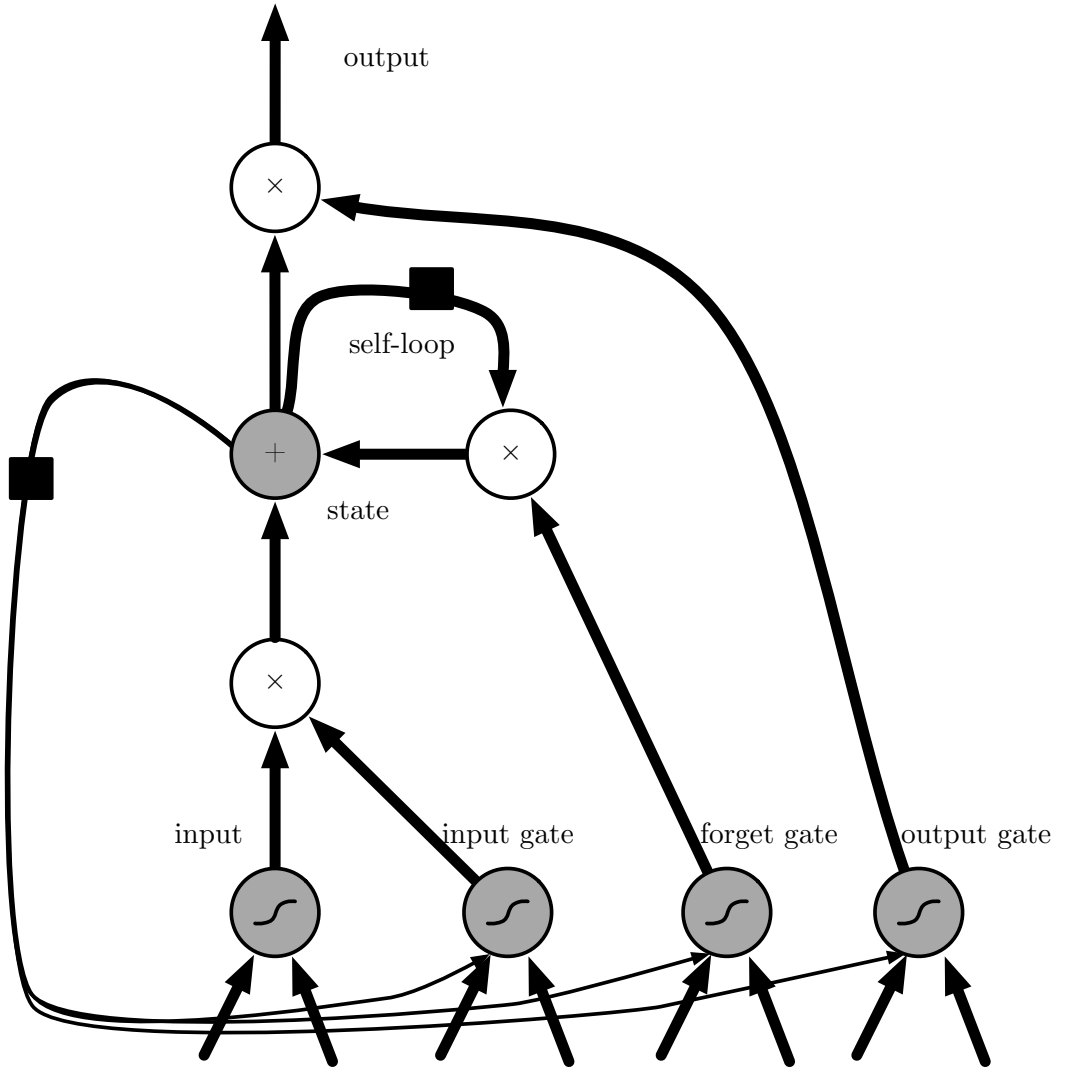
$$\frac{\partial L}{\partial s_t} = \frac{\partial L}{\partial s_T}\frac{\partial s_T}{\partial s_{T-1}}\cdots\frac{\partial s_{t+1}}{\partial s_t}$$

Storing bits robustly requires sing. values<1

- Problems:
  - sing. values of Jacobians > 1 → *gradients explode* → **Gradient clipping**
  - or sing. values < 1 → *gradients shrink & vanish* *(Hochreiter 1991)*
  - or random → variance grows exponentially

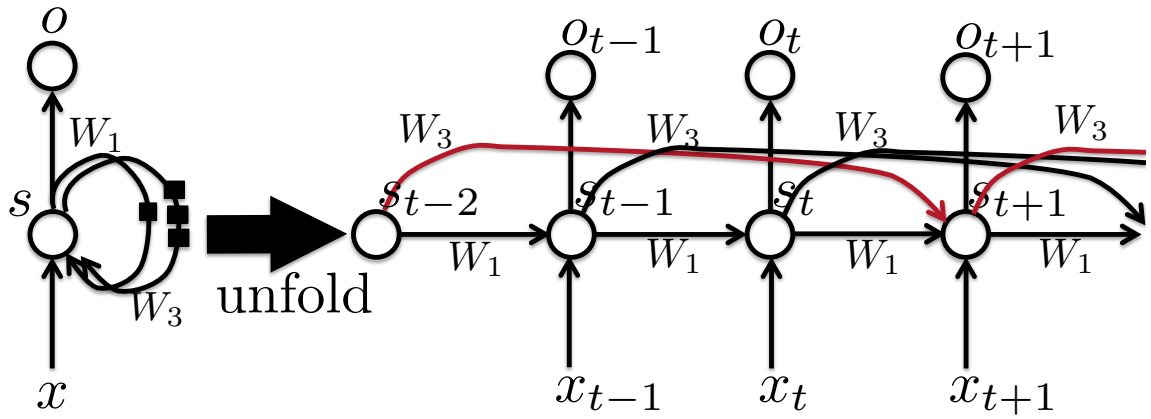
Gated Recurrent Units & LSTM

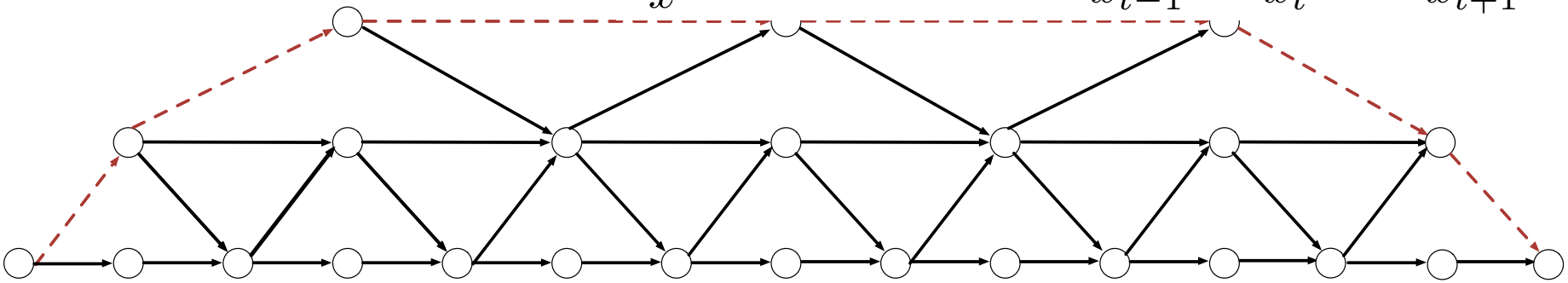
- **Create a path where gradients can flow for longer with self-loop**
- Corresponds to an eigenvalue of Jacobian slightly less than 1
- LSTM is **heavily used** *(Hochreiter & Schmidhuber 1997)*
- GRU light-weight version *(Cho et al 2014)*



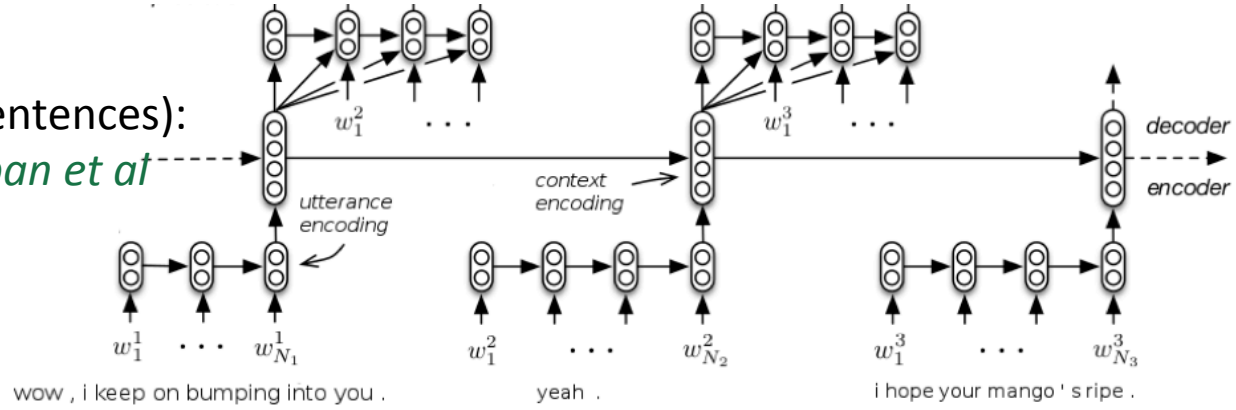
Delays & Hierarchies to Reach Farther

- Delays and multiple time scales, *Elhihi & Bengio NIPS 1995, Koutnik et al ICML 2014*



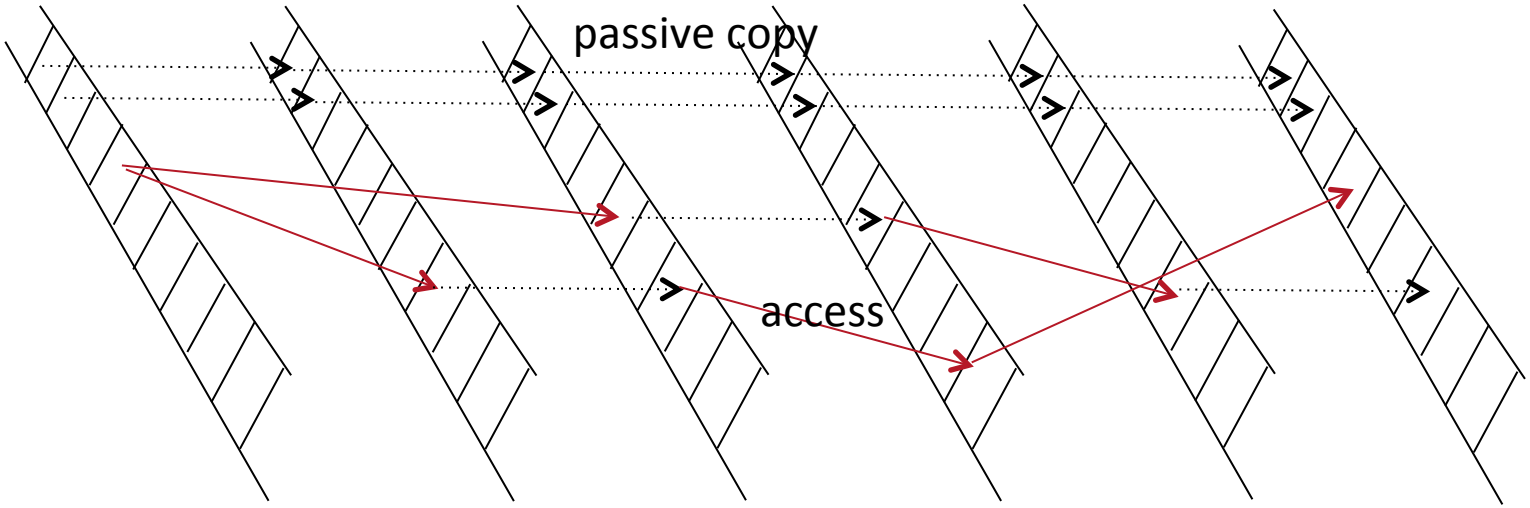


Hierarchical RNNs (words / sentences):
Sordoni et al CIKM 2015, Serban et al AAAI 2016



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 evoked for read or write
- Forgetting = vanishing gradient.
- Memory = larger state, avoiding the need for forgetting/vanishing



Paths where gradient flows unhampered → |eigenvalue|=1

- Self-loop with weight = 1
- Copy operation in memory network

→ For each scalar that is so preserved, gradient can flow back unhampered

→ And there must exist a direction in which the Jacobian preserves vector magnitude

→ Corresponds to an eigenvalue whose magnitude is 1

Unitary Evolution RNNs

Martin Arjowski, Amar Shah & Yoshua Bengio

arXiv 1511.06464

Submitted to ICLR 2016

Large state: cannot afford $O(n^2)$ recurrent computation

Orthogonal matrices $\subset$ Unitary matrices

Non-trivial to parametrize efficiently

What if W had all |eigenvalues|=1

$$z_{t+1} = \mathbf{W}_t h_t + \mathbf{V}_t x_{t+1}$$
$$h_{t+1} = \sigma(z_{t+1})$$

$$\frac{\partial C}{\partial h_t} = \frac{\partial C}{\partial h_T}\frac{\partial h_T}{\partial h_t} = \frac{\partial C}{\partial h_T}\prod_{k=t}^{T-1}\frac{\partial h_{k+1}}{\partial h_k} = \frac{\partial C}{\partial h_T}\prod_{k=t}^{T-1}\mathbf{D}_{k+1}\mathbf{W}_k^T,$$

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

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

Guaranteed no explosion, no need for gradient clipping

Building Blocks for Unitary W

*U = V D V** with V fixed would have a bad ratio of Computation *O(n^2)* to parameters *O(n), and D complex even for orthogonal matrices*

- $\mathbf{D}$, a diagonal matrix with $\mathbf{D}_{j,j} = e^{iw_j}$, with parameters $w_j \in \mathbb{R}$,
- $\mathbf{R} = \mathbf{I} - 2\frac{vv^*}{\|v\|^2}$, a reflection matrix in the complex vector $v \in \mathbb{C}^n$,
- $\mathbf{\Pi}$, a fixed random index permutation matrix, and
- $\mathcal{F}$ and $\mathcal{F}^{-1}$, the Fourier and inverse Fourier transforms.

O(n) computation & params

O(n log *n)* computation

Building Blocks for Unitary W

- $\mathbf{D}$, a diagonal matrix with $\mathbf{D}_{j,j} = e^{iw_j}$, with parameters $w_j \in \mathbb{R}$,
- $\mathbf{R} = \mathbf{I} - 2\frac{vv^*}{\|v\|^2}$, a reflection matrix in the complex vector $v \in \mathbb{C}^n$,
- $\mathbf{\Pi}$, a fixed random index permutation matrix, and
- $\mathcal{F}$ and $\mathcal{F}^{-1}$, the Fourier and inverse Fourier transforms.

A Recipe Inspired by FastFood (Le et al ICML 2013) and ACDC (Moczulski et al arXiv 2015)

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

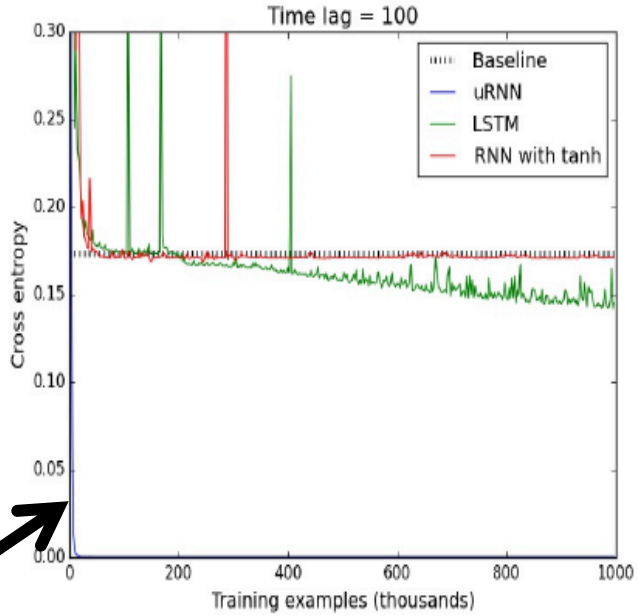
Nonlinearity

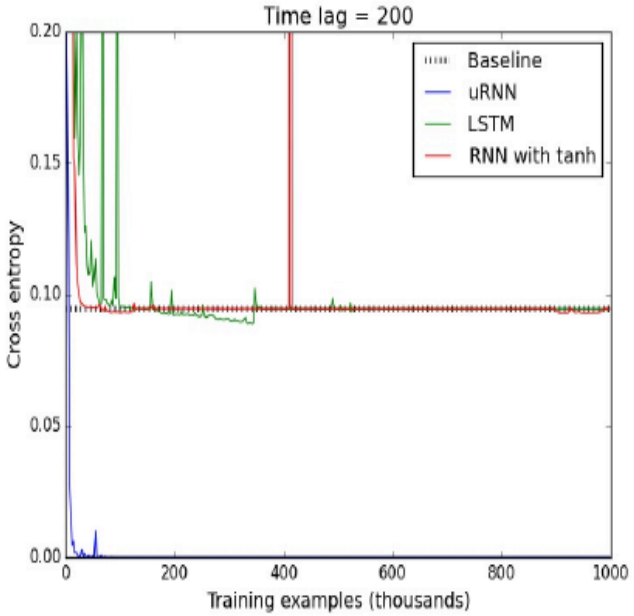
We do not want to destroy the information present in the phase

$$\sigma_{\text{modReLU}}(z) = \begin{cases} (|z| + b)\frac{z}{|z|} & \text{if } |z| + b \geq 0 \\ 0 & \text{if } |z| + b < 0 \end{cases}$$

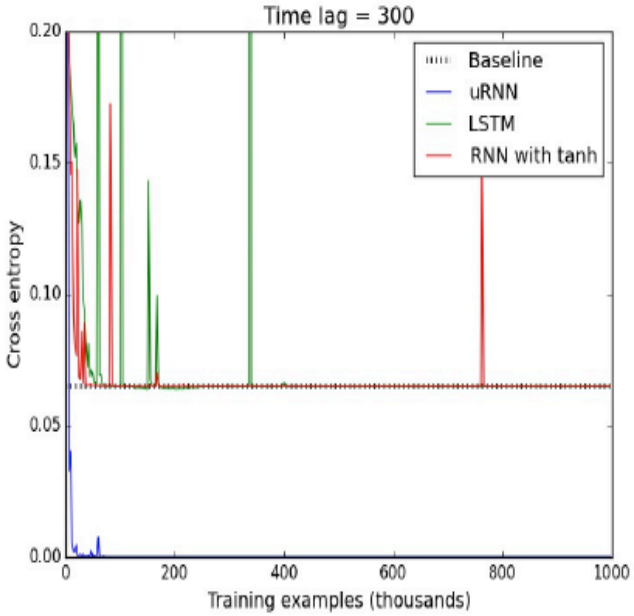
$$\sigma_{\text{modReLU}}(z) = \sigma_{\text{ReLU}}(|z| + b)\frac{z}{|z|}$$

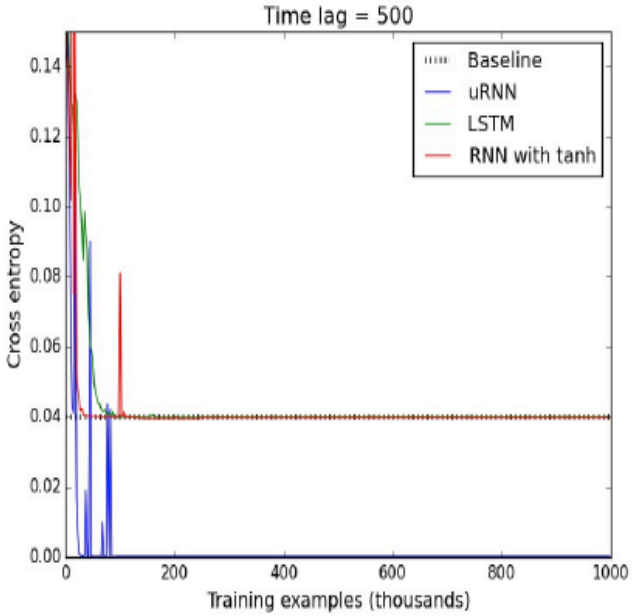
Memory Problem



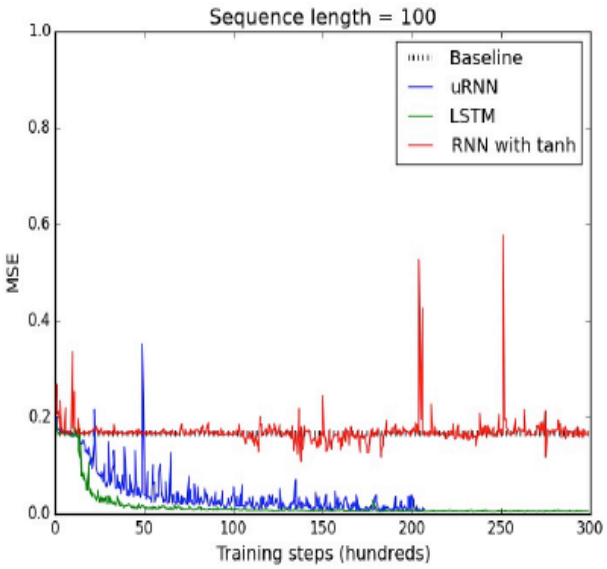


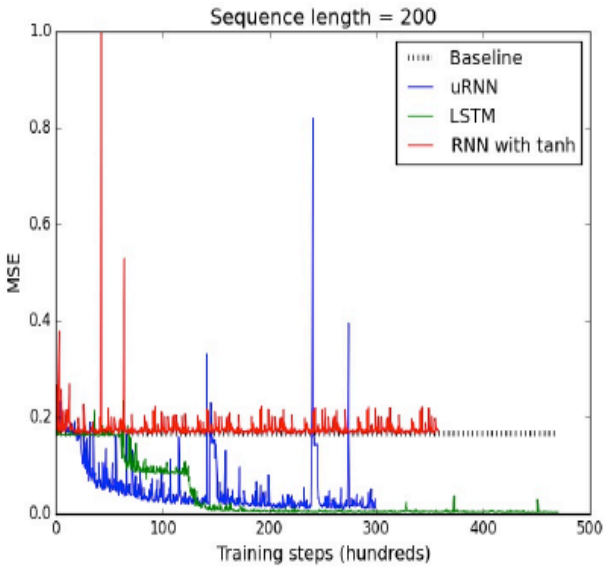
uRNN so good that we do not see the error curve

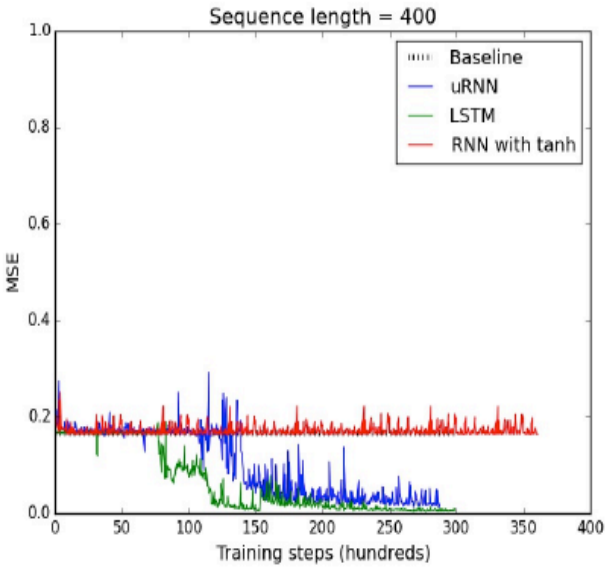


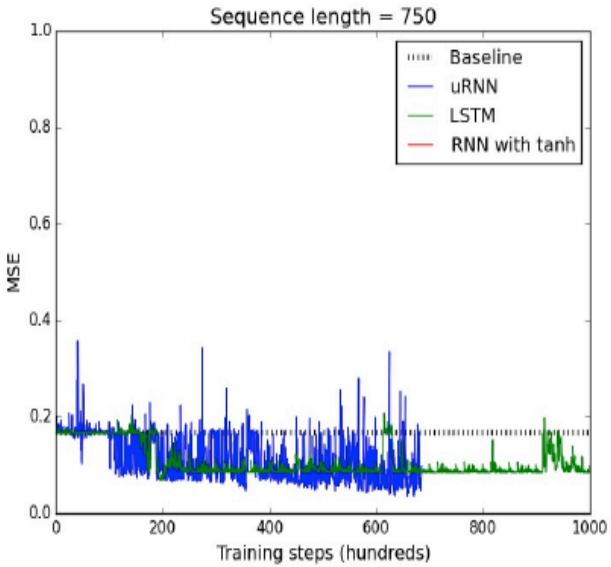


Adding Problem

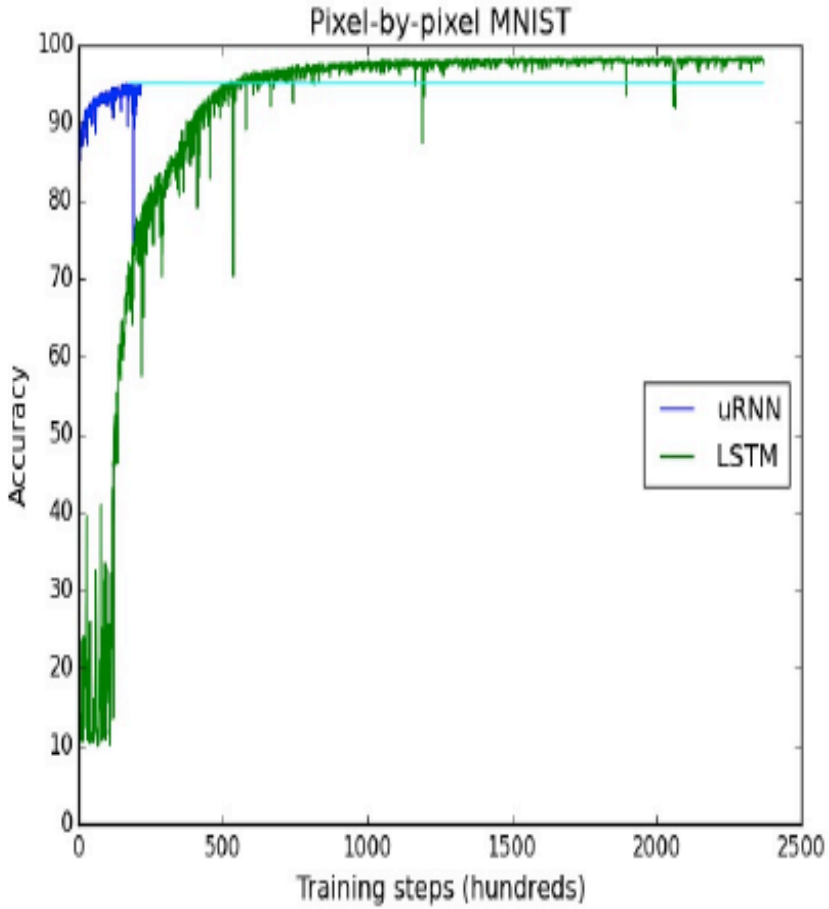


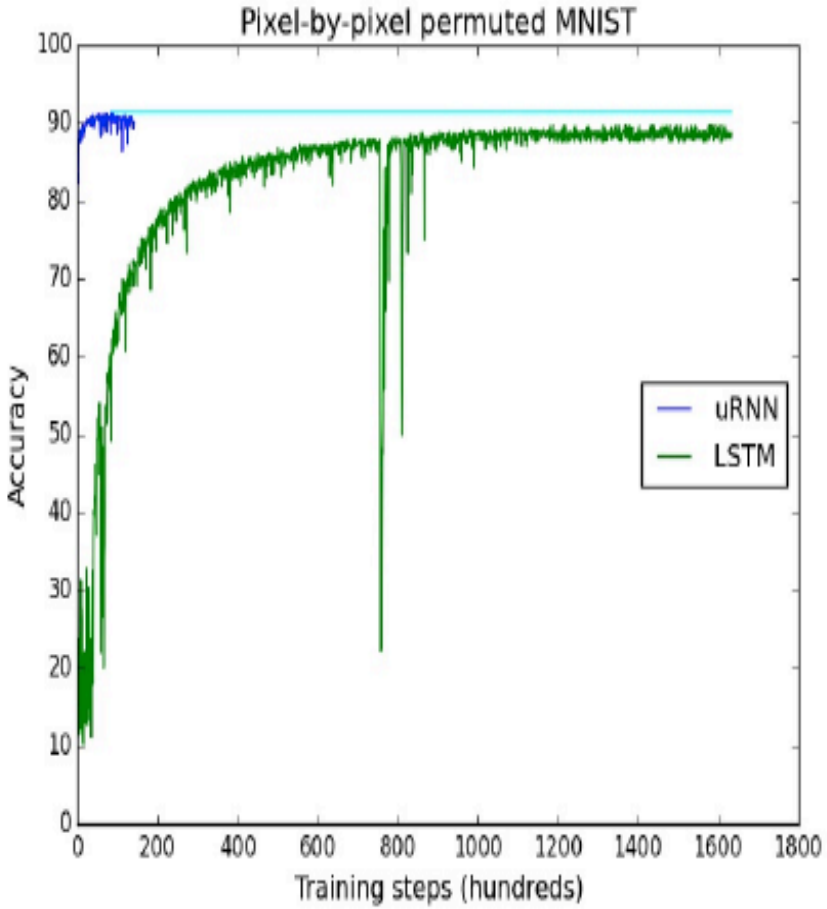




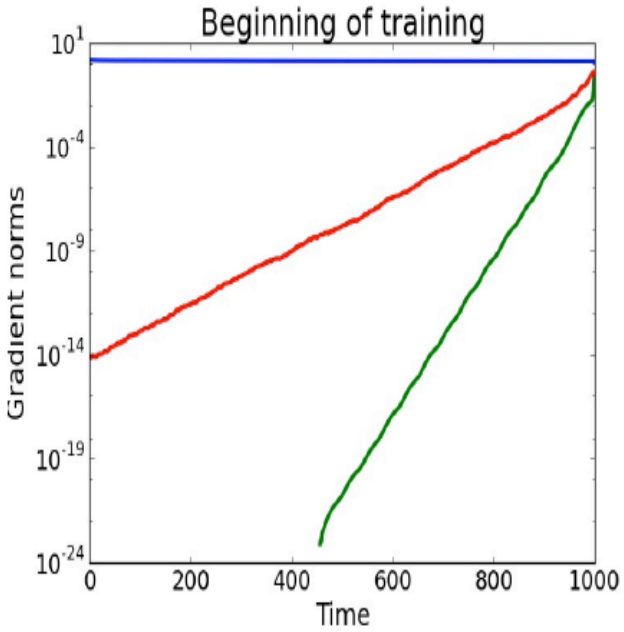


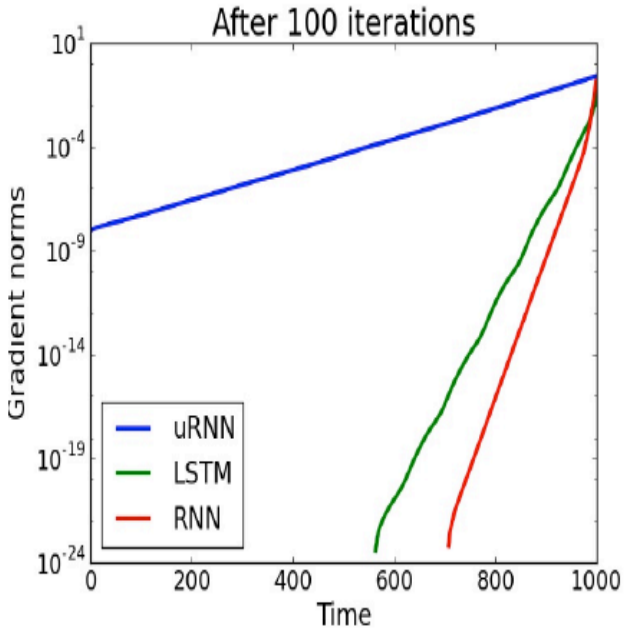
Sequential MNIST

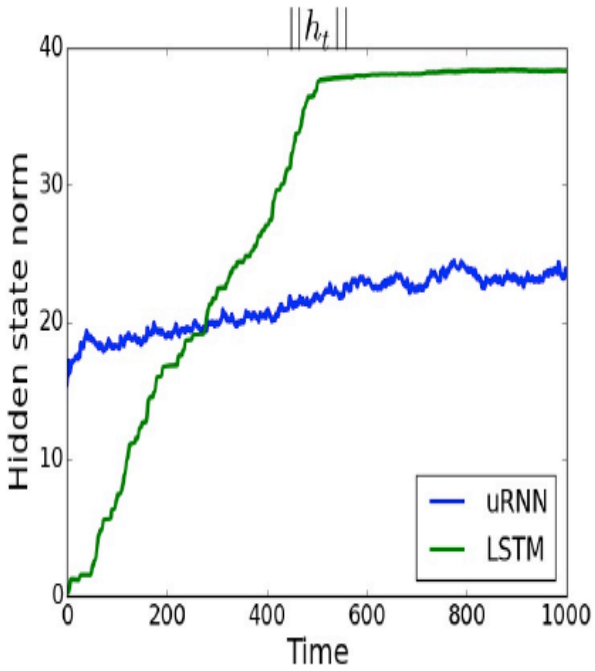


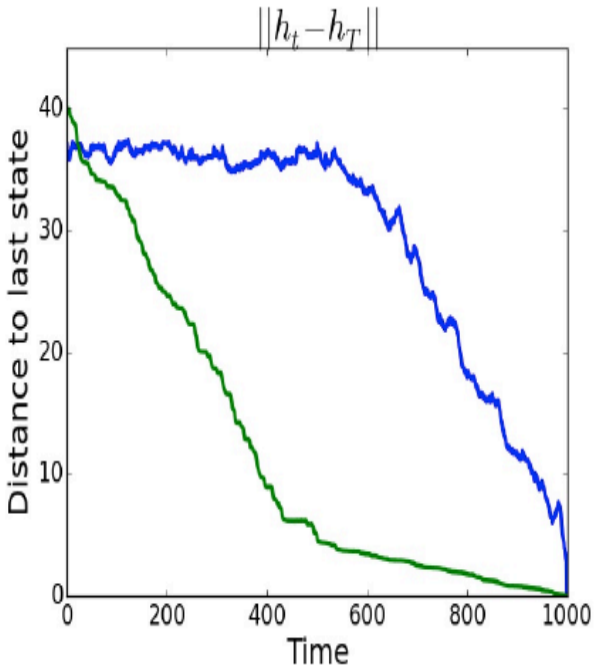


What is going on? uRNN forgets at a lower rate and LSTM state gets stuck at some point









MILA: Montreal Institute for Learning Algorithms