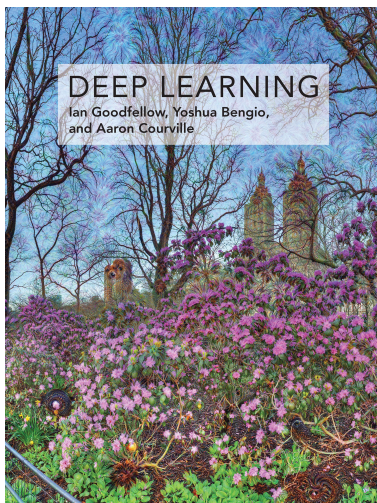


# Recurrent Nets and Attention for System 2 Processing

Yoshua Bengio

July 30th, 2018, CIFAR Deep Learning &  
Reinforcement Learning Summer School, Toronto



Université   
de Montréal

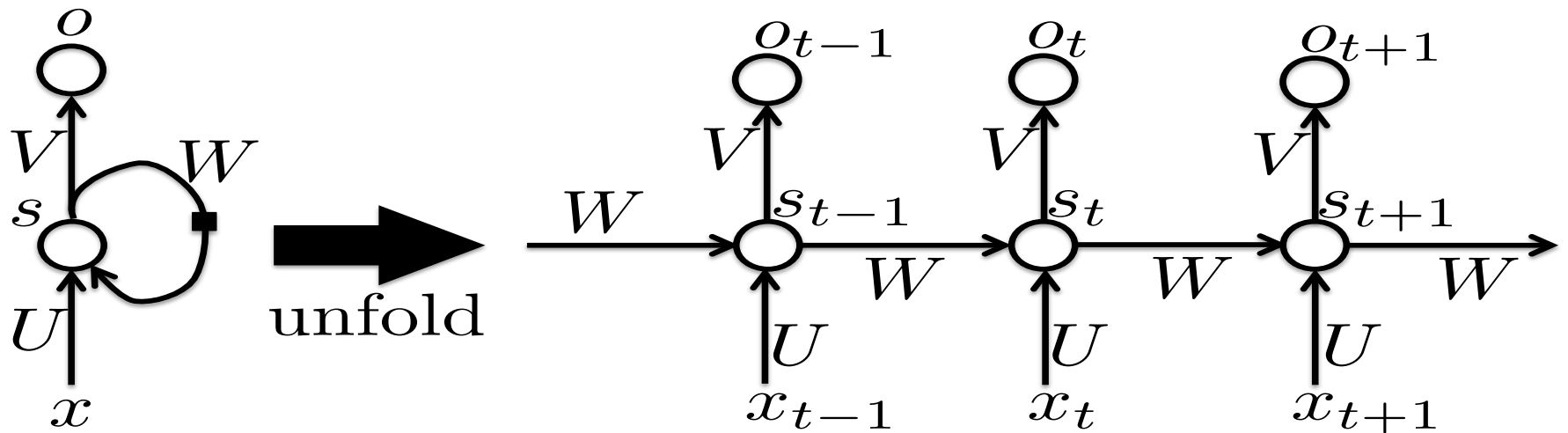


**CIFAR**  
CANADIAN  
INSTITUTE  
FOR  
ADVANCED  
RESEARCH

**ICRA**  
INSTITUT  
CANADIEN  
DE  
RECHERCHES  
AVANCÉES

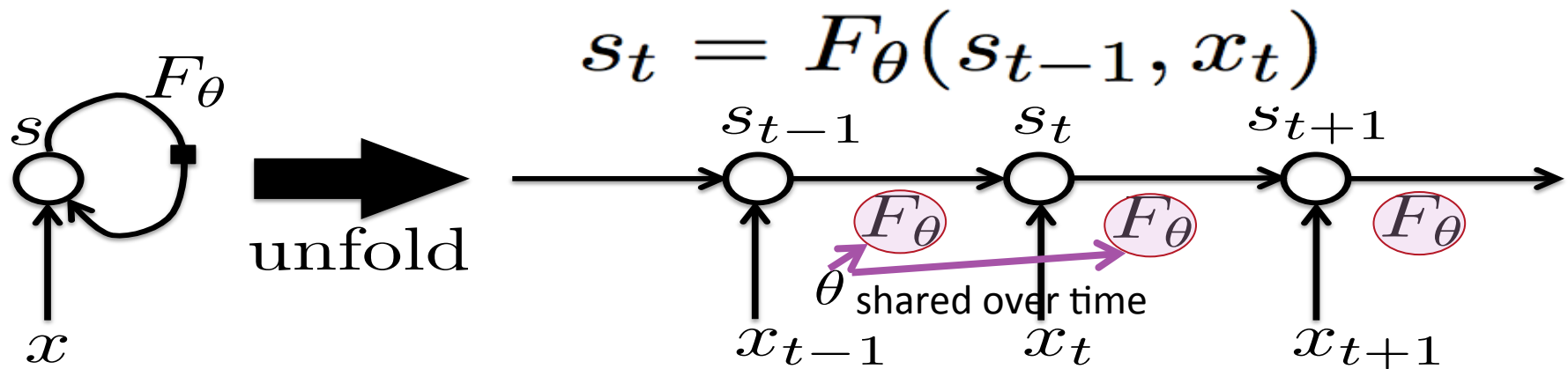
# Recurrent Neural Networks

- Can read or produce an output at each time step: unfolding the graph tells us how to back-prop through time.



# Recurrent Neural Networks

- Selectively summarize an input sequence in a fixed-size state vector via a recursive update



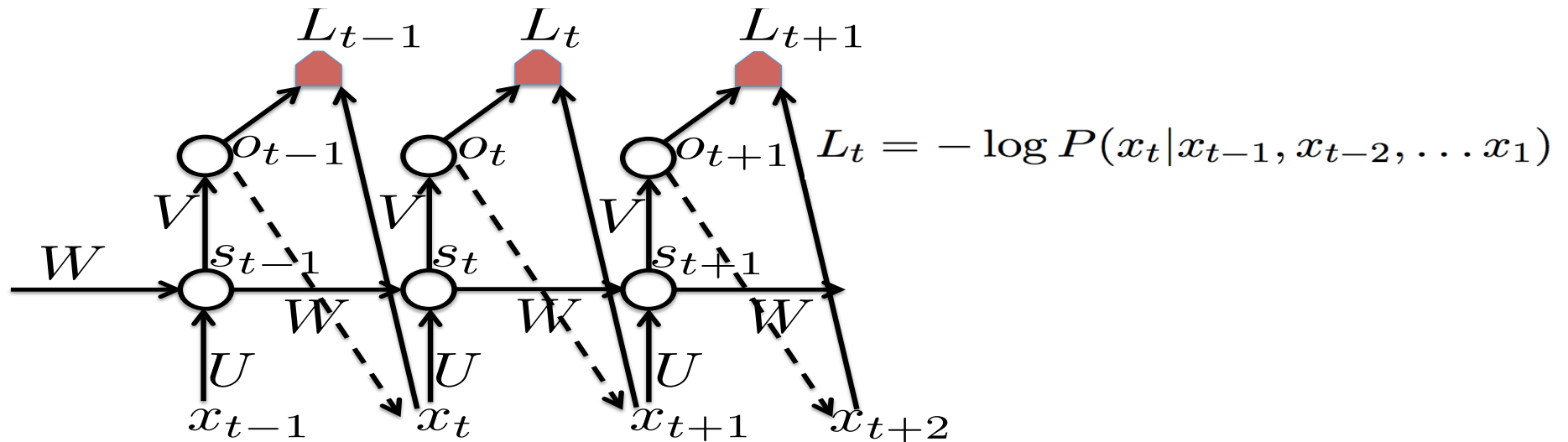
$$s_t = G_t(x_t, x_{t-1}, x_{t-2}, \dots, x_2, x_1)$$

➔ Generalizes naturally to new lengths not seen during training

# Generative RNNs

- An RNN can represent a fully-connected **directed generative model**: every variable predicted from all previous ones.

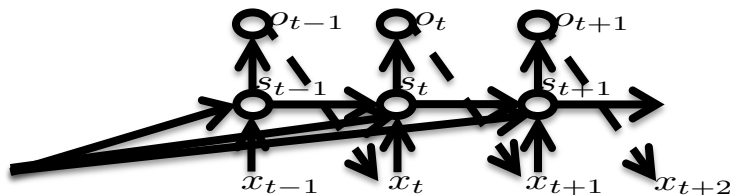
$$P(\mathbf{x}) = P(x_1, \dots, x_T) = \prod_{t=1}^T P(x_t | x_{t-1}, x_{t-2}, \dots, x_1)$$



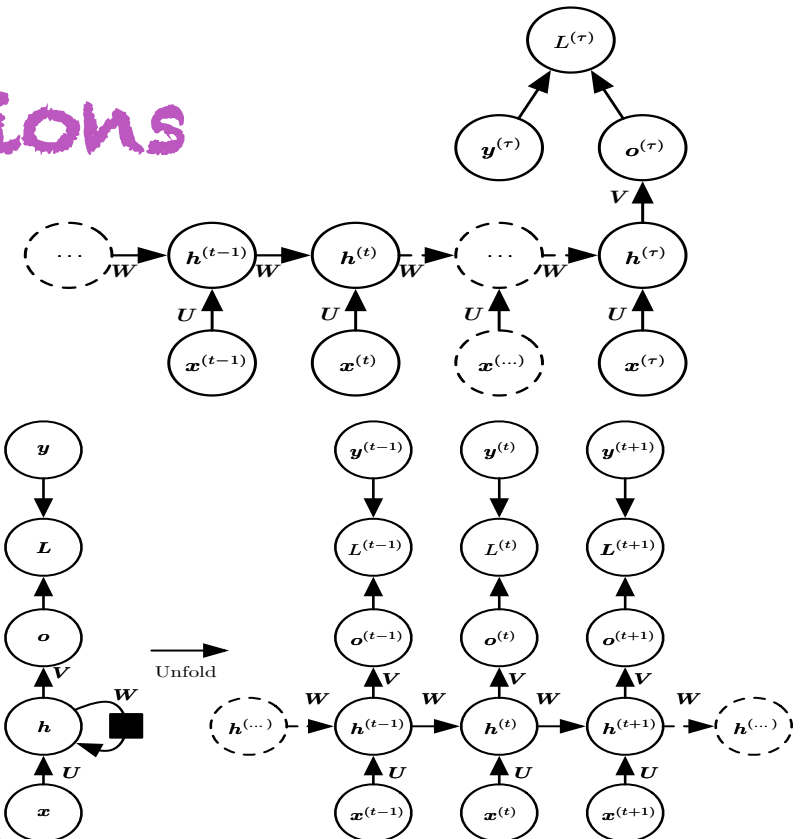
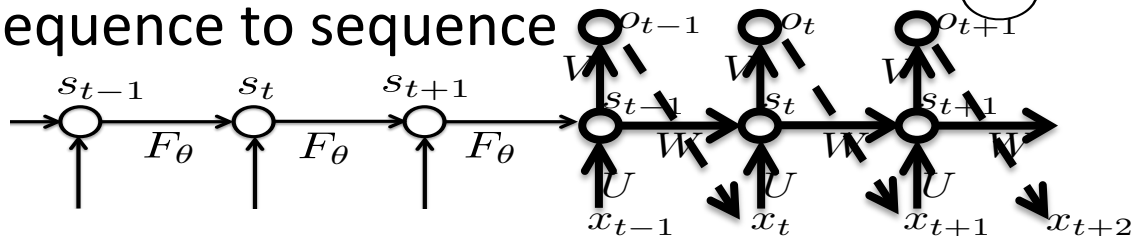


# Conditional Distributions

- Sequence to vector
- Sequence to sequence of the same length, aligned
- Vector to sequence

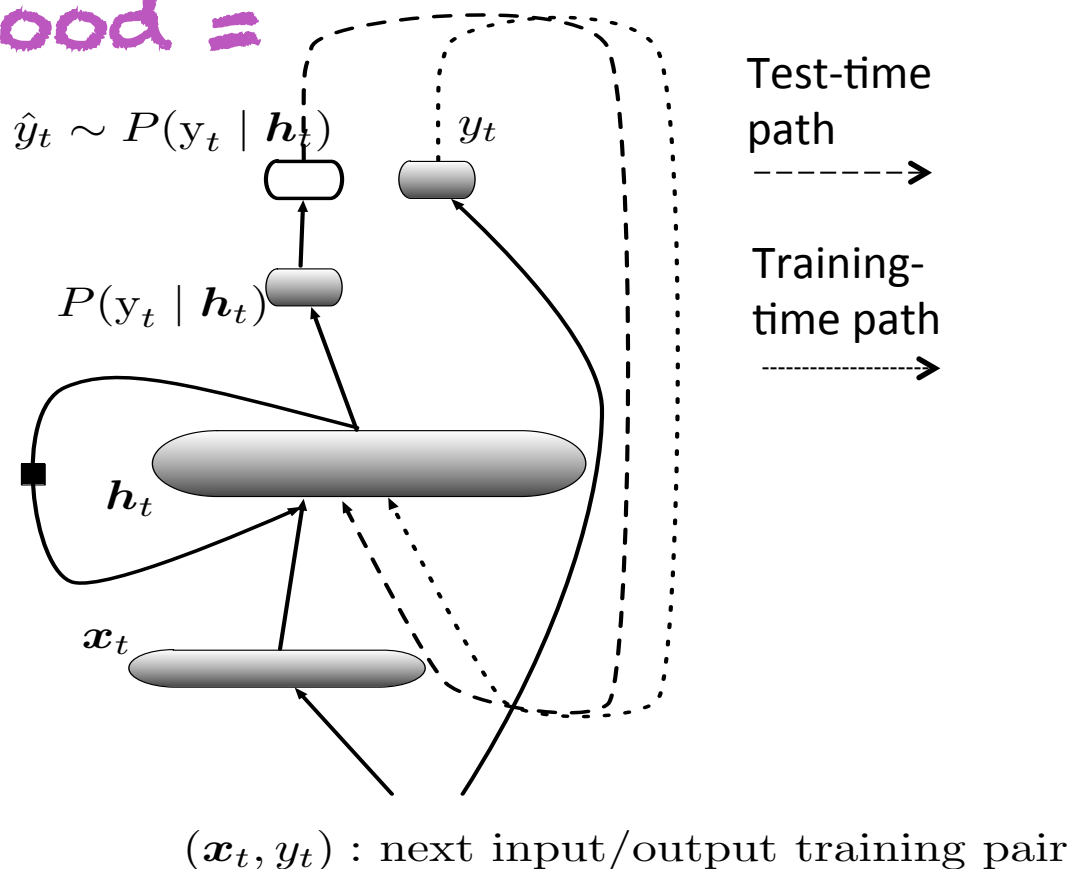


- Sequence to sequence



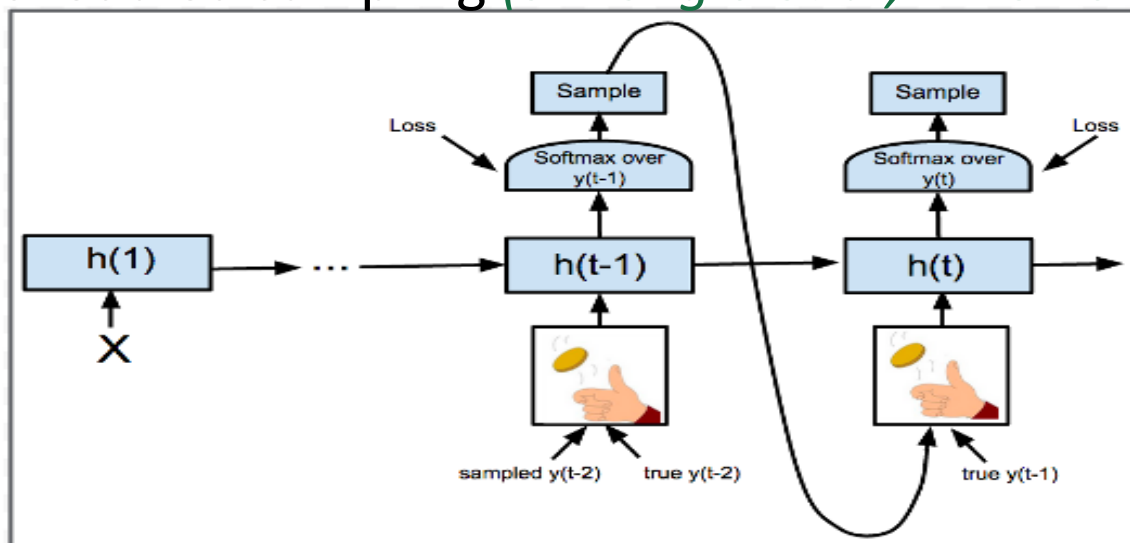
# Maximum Likelihood = Teacher Forcing

- During training, past  $y$  in input is from training data
- At generation time, past  $y$  in input is generated
- Mismatch can cause "compounding error"



# Ideas to reduce the train/generate mismatch in teacher forcing

- Scheduled sampling (*S. Bengio et al, NIPS 2015*)

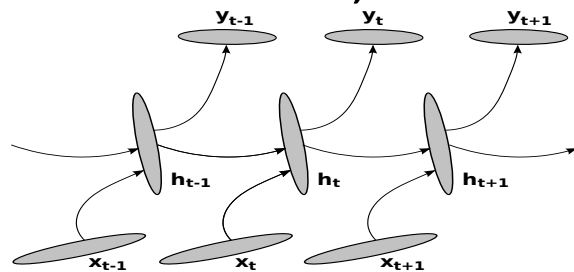


Related to  
SEARN (Daumé et al 2009)  
DAGGER (Ross et al 2010)  
Gradually increase the  
probability of using  
the model's samples  
vs the ground truth  
as input.

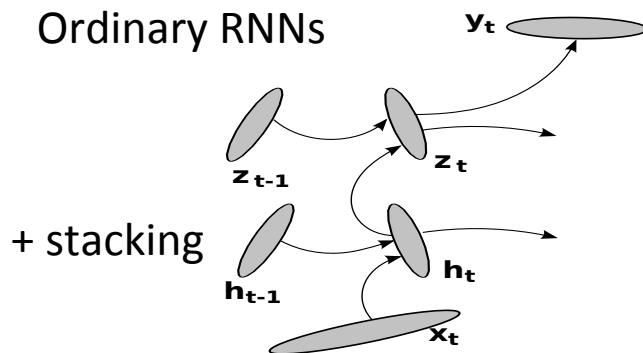
- Backprop through open-loop sampling recurrence & minimize long-term cost (but which one? GAN would be most natural → *Professor Forcing, NIPS'2016*)

# Increasing the Expressive Power of RNNs with more Depth

- ICLR 2014, *How to construct deep recurrent neural networks*

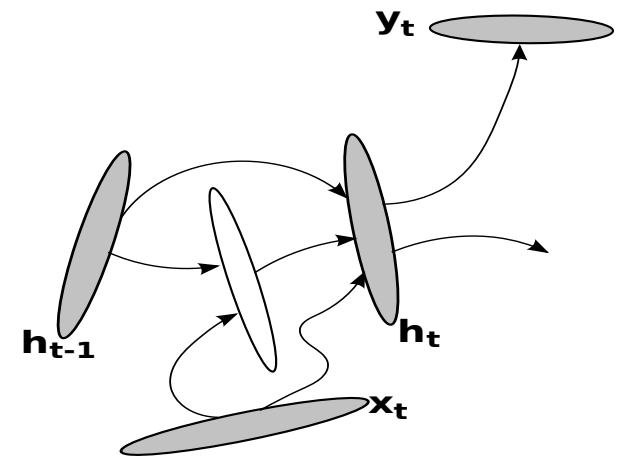
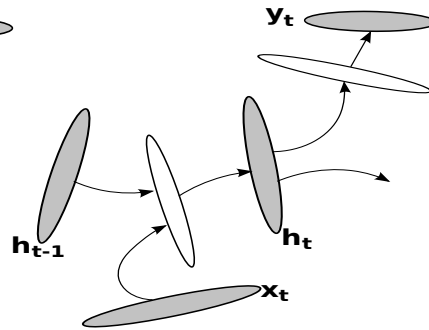


Ordinary RNNs



+ stacking

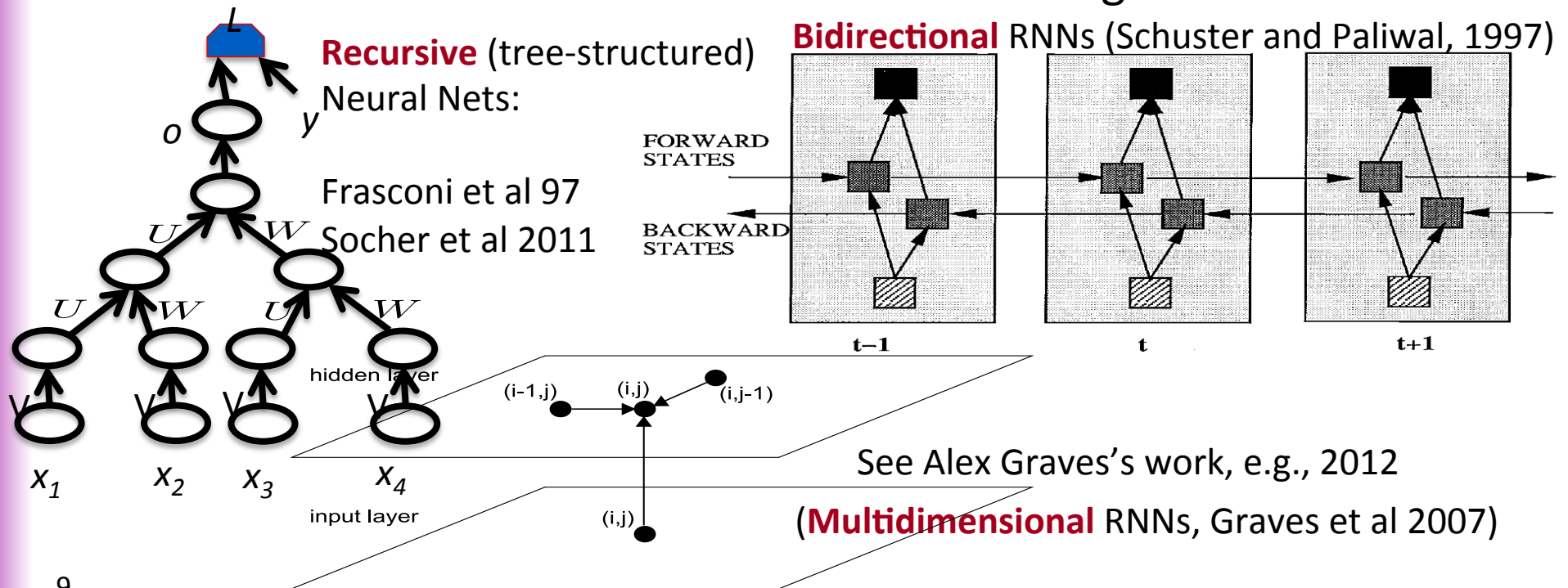
+ deep hid-to-out  
+ deep hid-to-hid  
+ deep in-to-hid



+ skip connections for  
creating shorter paths

# Bidirectional RNNs, Recursive Nets, Multidimensional RNNs, etc.

- The unfolded architecture needs not be a straight chain



# Multiplicative Interactions

(Wu et al, 2016, arXiv:1606.06630)

- Multiplicative Integration RNNs:

- Replace

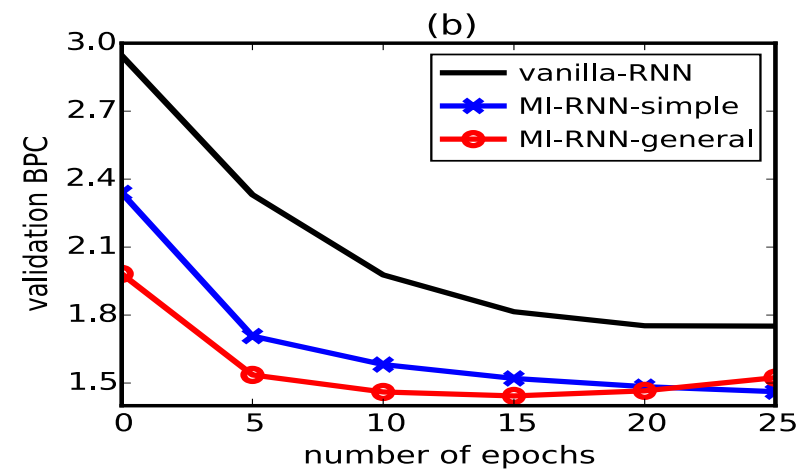
$$\phi(\mathbf{W}x + \mathbf{U}z + \mathbf{b})$$

- By

$$\phi(\mathbf{W}x \odot \mathbf{U}z + \mathbf{b})$$

- Or more general:

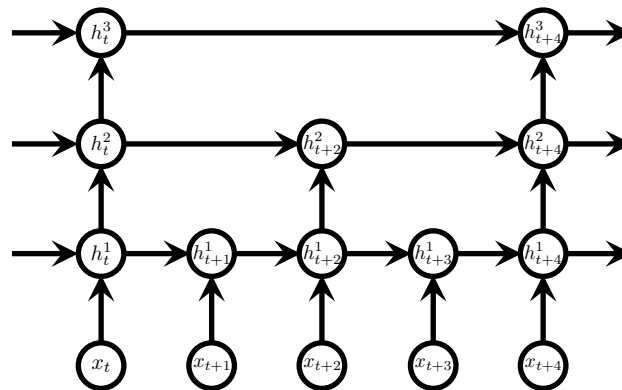
$$\phi(\alpha \odot \mathbf{W}x \odot \mathbf{U}z + \beta_1 \odot \mathbf{U}z + \beta_2 \odot \mathbf{W}x + \mathbf{b})$$



# Multiscale or Hierarchical RNNs

(Bengio & Elhihi, NIPS 1995)

- Motivation :
  - Gradients can propagate over longer spans through slow time-scale paths
- Approach :
  - Introduce a network architecture that update the states of its hidden layers with different speeds in order to capture multiscale representation of sequences.



# Learning Long-Term Dependencies with Gradient Descent is Difficult

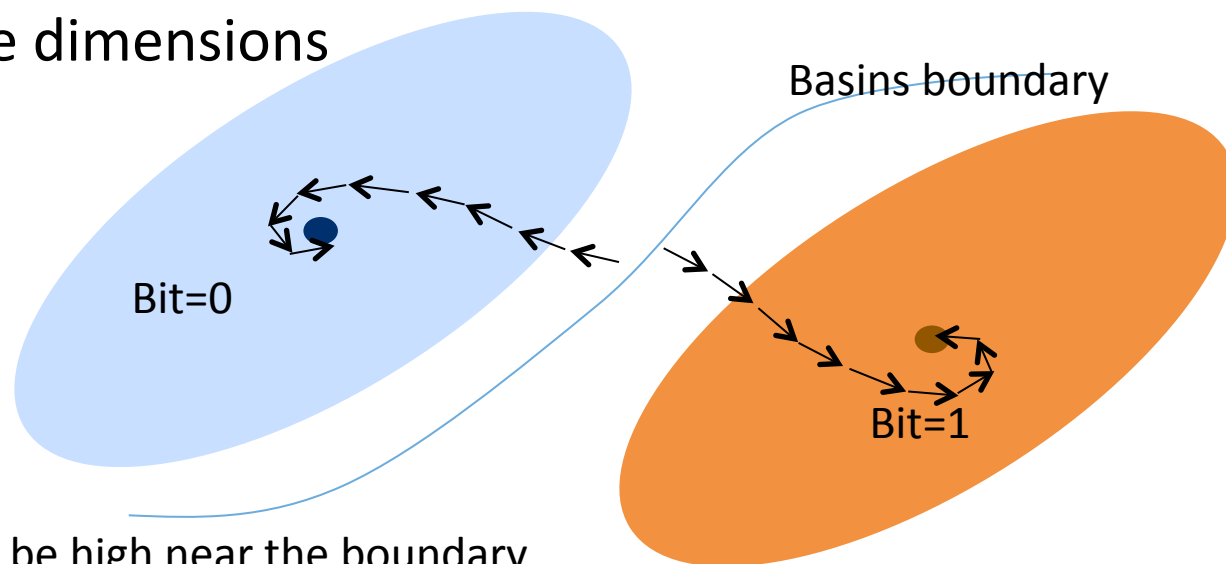
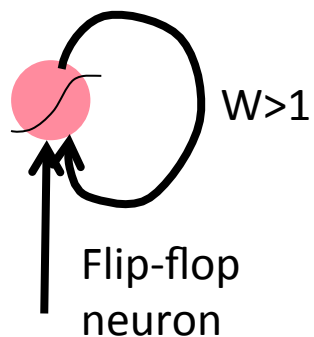


Y. Bengio, P. Simard & P. Frasconi, IEEE Trans. Neural Nets, **1994**



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

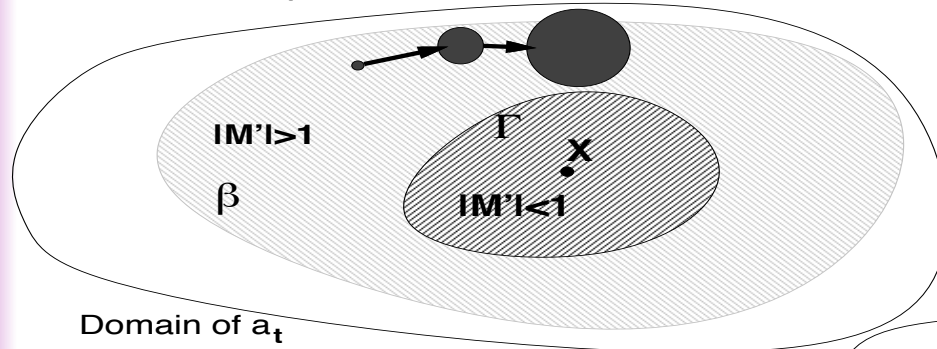
- Some subspace of the state can store 1 or more bits of information if the dynamical system has multiple basins of attraction in some dimensions



Note: gradients MUST be high near the boundary

# Robustly storing 1 bit in the presence of bounded noise

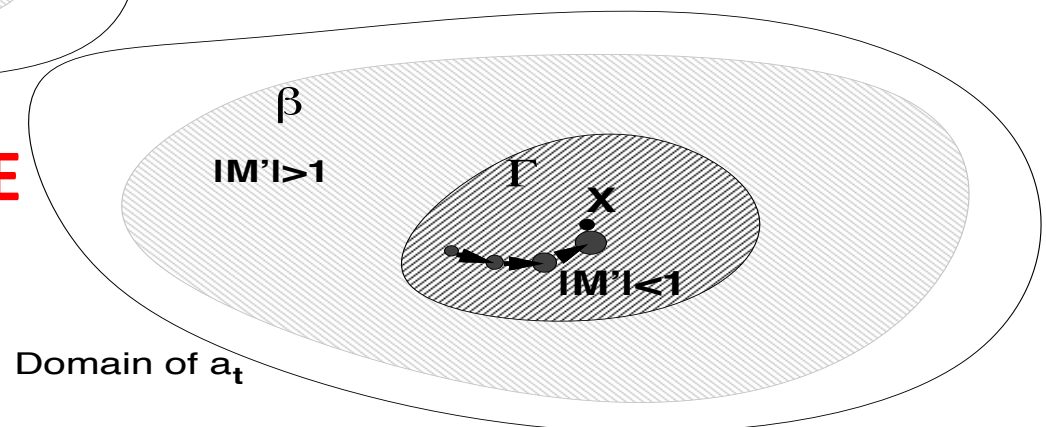
- With spectral radius  $> 1$ , noise can kick state out of attractor



**UNSTABLE**

**CONTRACTIVE  
→ STABLE**

- Stable with radius  $< 1$



## Storing Reliably $\rightarrow$ 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  $T$

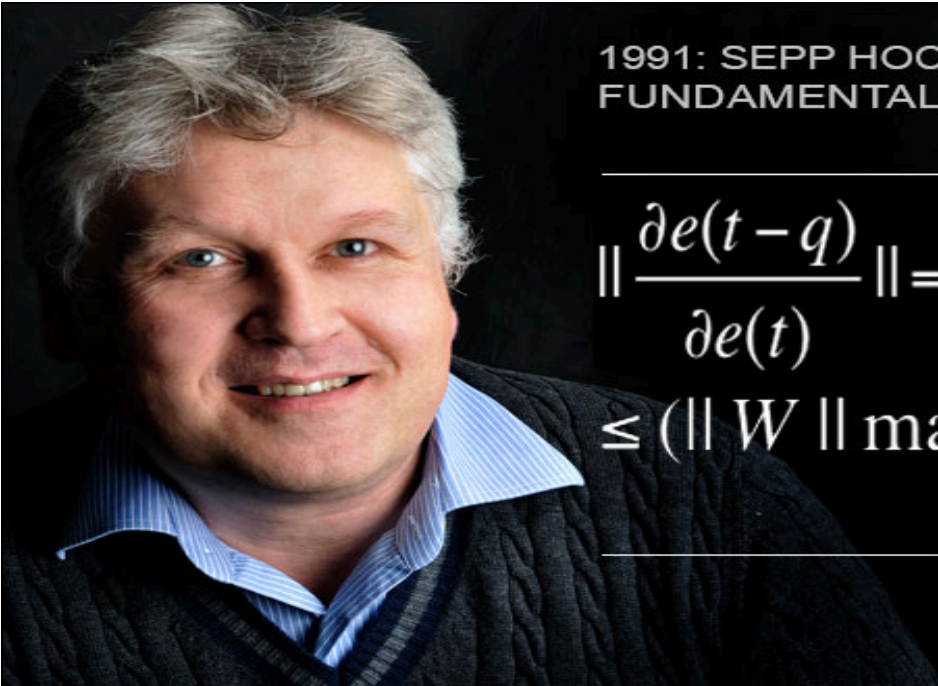
$$L = L(s_T(s_{T-1}(\dots s_{t+1}(s_t, \dots))))$$

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

- If spectral radius of Jacobian is  $< 1 \rightarrow$  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

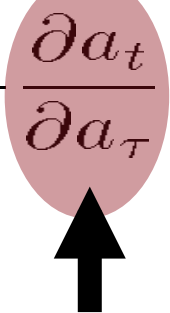


1991: SEPP HOCHREITER'S ANALYSIS OF THE  
FUNDAMENTAL DEEP LEARNING PROBLEM

$$\left\| \frac{\partial e(t-q)}{\partial e(t)} \right\| = \left\| \prod_{m=1}^q W F'(Net(t-m)) \right\|$$
$$\leq (\|W\| \max_{Net} \{ \|F'(Net)\| \})^q$$

## Why it hurts gradient-based Learning

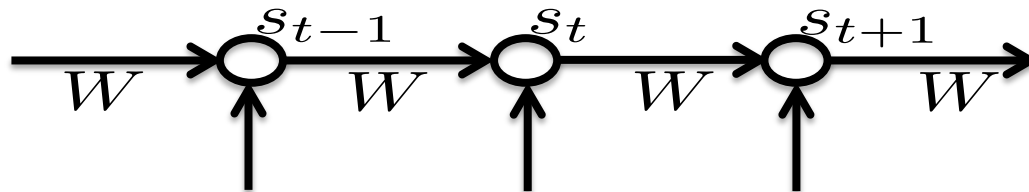
- Long-term dependencies get a weight that is exponentially smaller (in T) compared to short-term dependencies

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


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

# Vanishing Gradients in Deep Nets are Different from the Case in RNNs

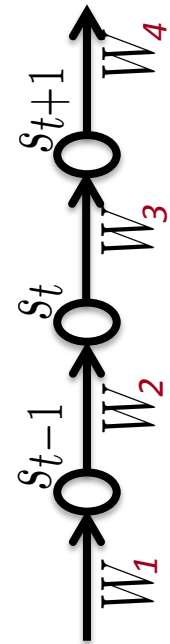
- If it was just a case of vanishing gradients in deep nets, we could just rescale the per-layer learning rate, but that does not really fix the training difficulties.



- Can't do that with RNNs because the weights are shared, & total true gradient = sum over different

"depths"

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



# To store information robustly the dynamics must be contractive

- 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].

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

Storing bits  
robustly requires  
e-values < 1

- Problems:

- e-values of Jacobians > 1 → *gradients explode*
- or e-values < 1 → *gradients shrink & vanish*
- or random → *variance grows exponentially*

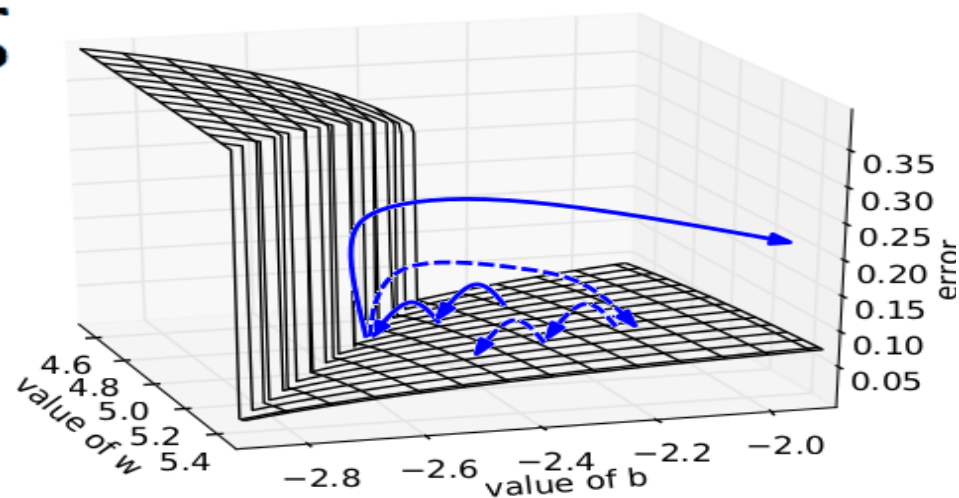
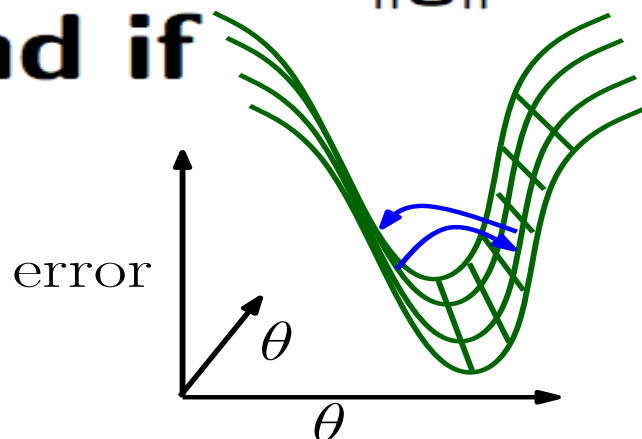


**Gradient  
clipping**

# Dealing with Gradient Explosion by Gradient Norm Clipping

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

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





# RNN Tricks

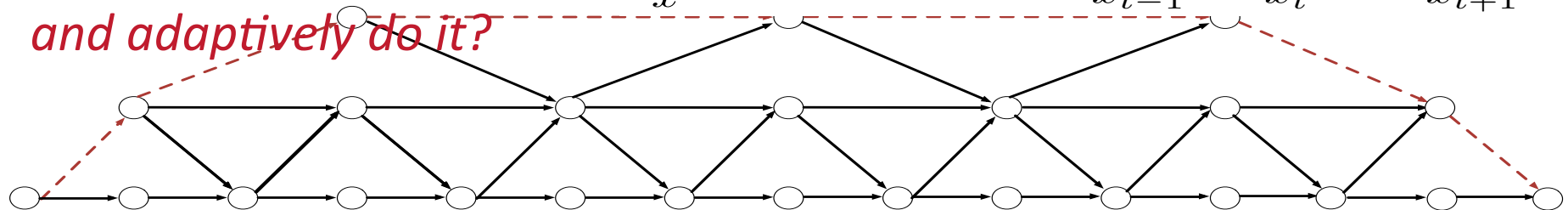
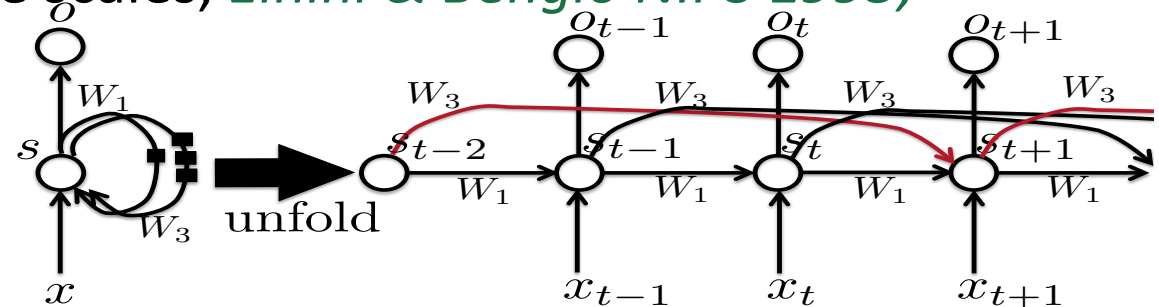
(Pascanu, Mikolov, Bengio, ICML 2013; Bengio, Boulanger & Pascanu, ICASSP 2013)

- Clipping gradients (avoid exploding gradients)
- Skip connections & leaky integration (propagate further)
- Multiple time scales / hierarchy (propagate further)
- Momentum (cheap 2<sup>nd</sup> order)
- Initialization (start in right ballpark avoids exploding/vanishing)
- Sparse Gradients (symmetry breaking)
- Gradient propagation regularizer (avoid vanishing gradient)
- Gated self-loops (LSTM & GRU, reduces vanishing gradient)

# Delays & Hierarchies to Reach Farther

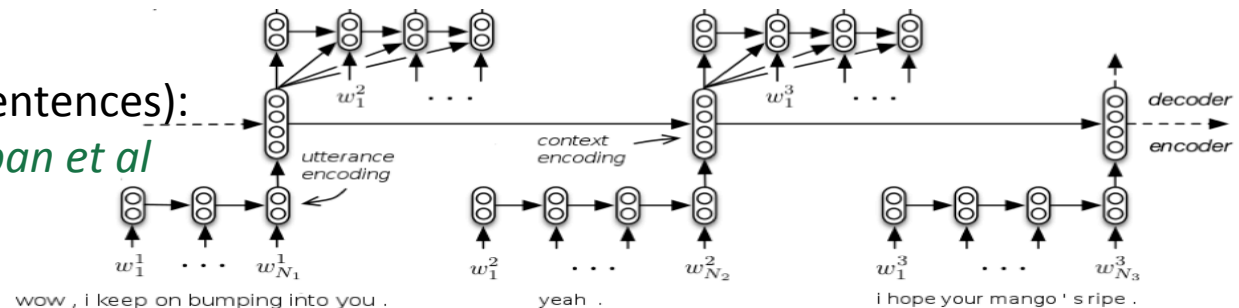
- Delays and multiple time scales, *Elhihi & Bengio NIPS 1995*, *Koutnik et al ICML 2014*

- How to do this right?
- How to automatically and adaptively do it?



Hierarchical RNNs (words / sentences):

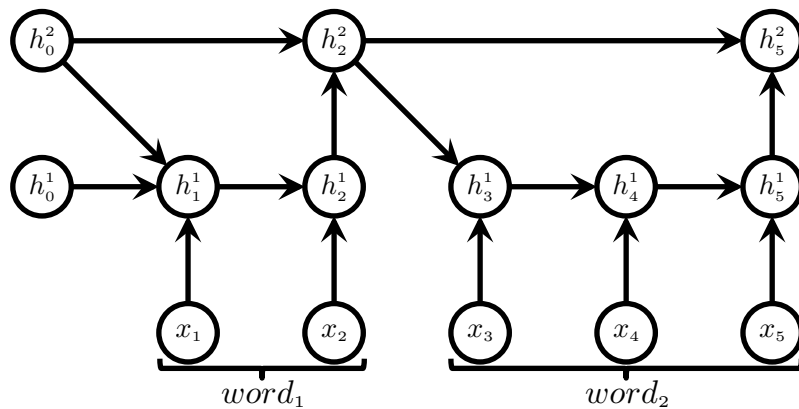
*Sordoni et al CIKM 2015*, *Serban et al AAAI 2016*



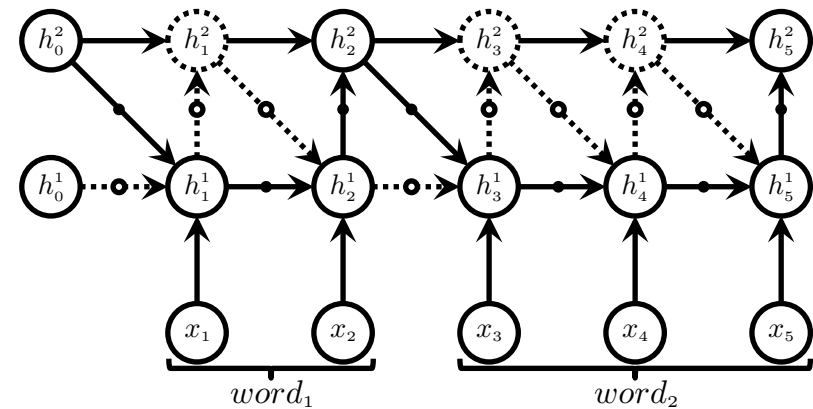
# Multi-Scale: Chung, Cho & Bengio ACL'2016



Hand-crafted segmentation



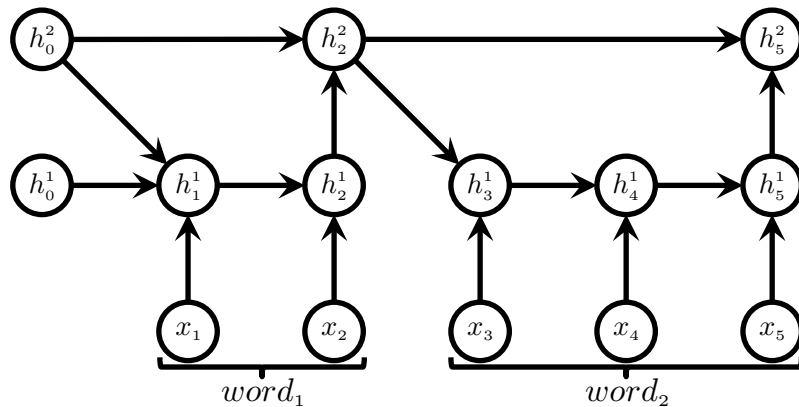
Learned segmentation



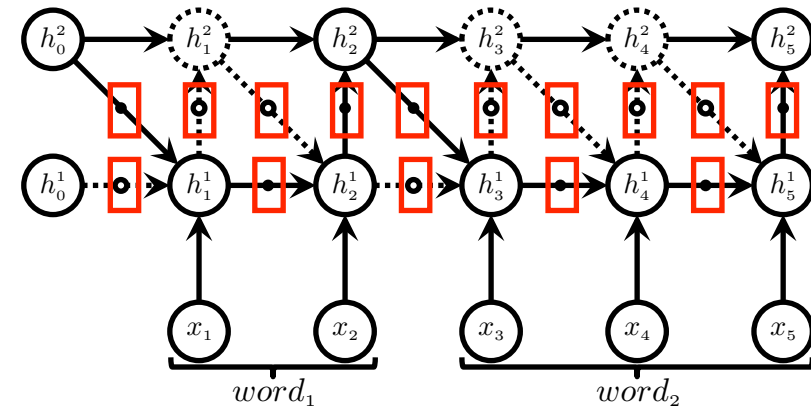
soft segmentation:  
can be trained by backprop

# Hierarchical Multiscale RNNs

Chung, Ahn & Bengio ICLR'2017



Boundary detectors have binary states!



Text8	
Model	BPC
<i>td</i> -LSTM (Zhang et al., 2016)	1.63
HF-MRNN (Mikolov et al., 2012)	1.54
MI-RNN (Wu et al., 2016)	1.52
Skipping-RNN (Pachitariu & Sahani, 2013)	1.48
MI-LSTM (Wu et al., 2016)	1.44
BatchNorm LSTM (Cooijmans et al., 2016)	1.36
HM-LSTM	1.32
LayerNorm HM-LSTM	<b>1.29</b>

Gradient signal:

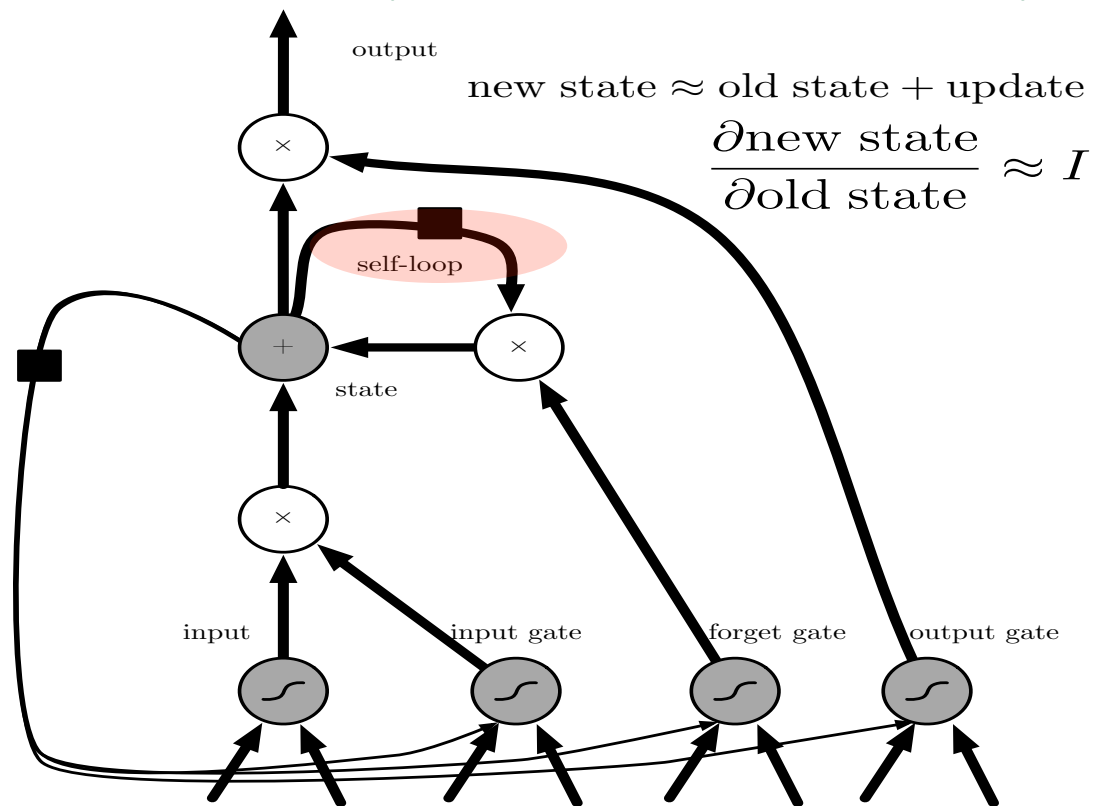
- straight-through
- REINFORCE

# 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)



# Gating for Attention-Based Neural Machine Translation

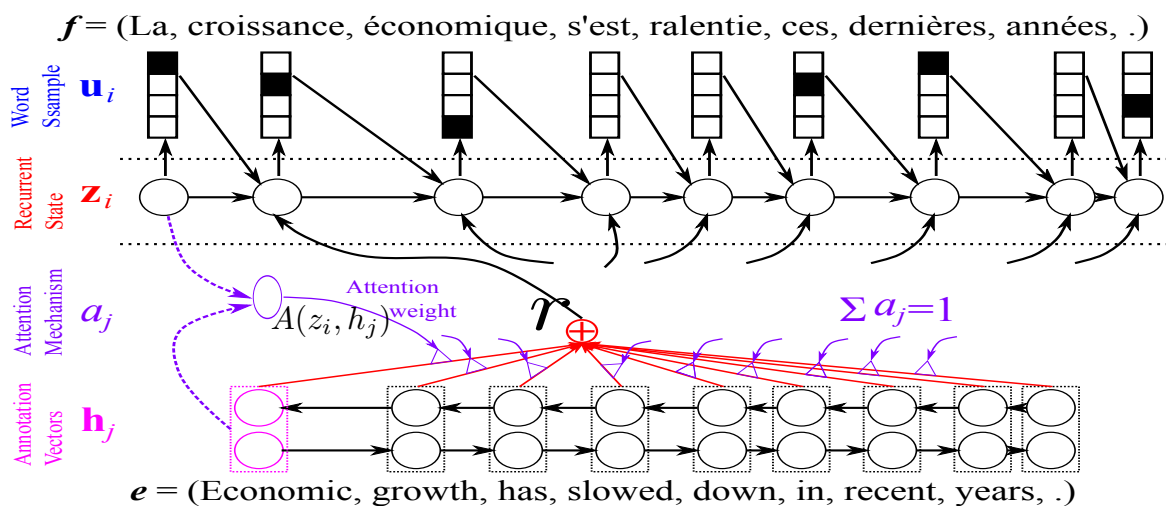
Related to earlier Graves 2013 for generating handwriting

- (Bahdanau, Cho & Bengio, arXiv sept. 2014, ICLR 2015)
- (Jean, Cho, Memisevic & Bengio, arXiv dec. 2014, ACL 2015)

$$a_j = \frac{e^{A(z_i, h_j)}}{\sum_{j'} e^{A(z_i, h_{j'})}}$$

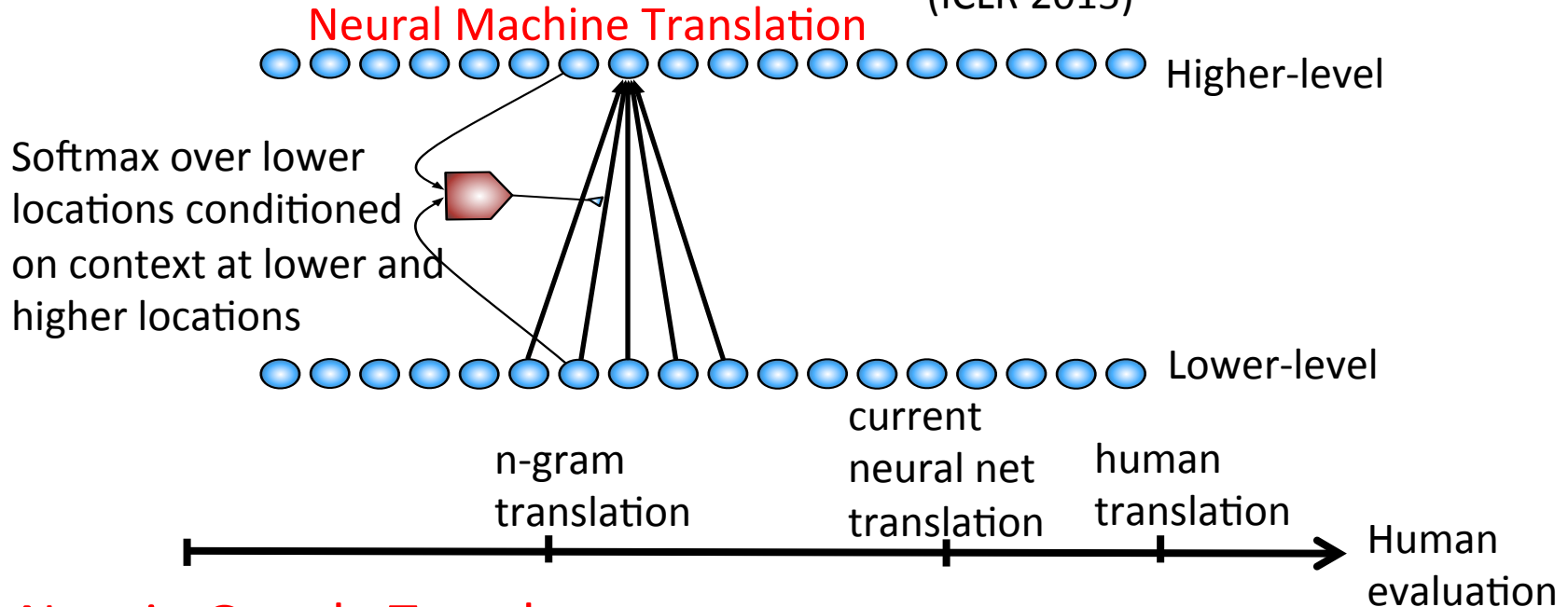
$$r = \sum_j a_j h_j$$

Read = weighted average of attended contents



# Gating for Attention-Based Neural Machine Translation

- Incorporating the idea of **attention**, using **GATING units**, has unlocked a breakthrough in machine translation: (ICLR'2015)

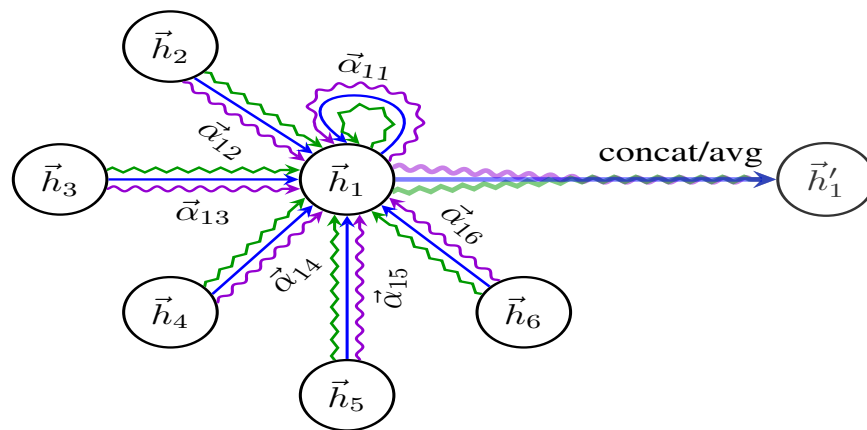


- Now in Google Translate

# Graph Attention Networks

Velickovic et al, ICLR 2018

- Handle variable-size neighborhood of each node using the same neural net by using an attention mechanism to aggregate information from the neighbors
- Use multiple attention heads to collect different kinds of information

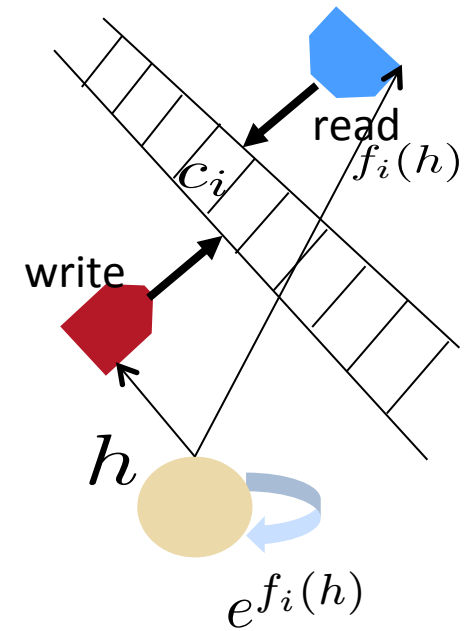




# Attention Mechanisms for Memory Access

- 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

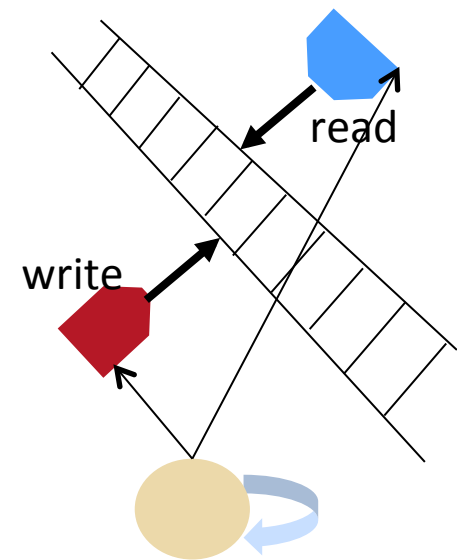
Read = weighted average of  
attended contents



$$\alpha_i = \frac{e^{f_i(h)}}{\sum_j e^{f_j(h)}}$$
$$r = \sum_i \alpha_i c_i$$

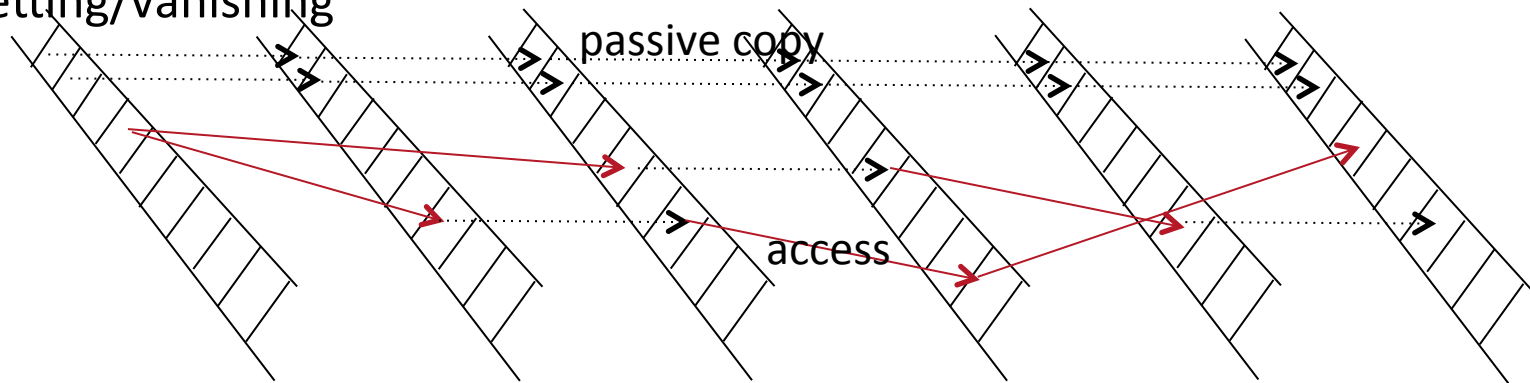
## From Memory to System 2

- Attention has also opened the door to neural nets which can **write to and read from a memory**
  - 2 systems:
    - Cortex-like (state controller and representations)
      - **System 1, intuition, fast heuristic answer (what current DL does quite well)**
    - Hippocampus-like (memory) + prefrontal cortex
      - System 2, slow, logical, sequential
- Memory-augmented networks gave rise to
  - Systems which reason
    - Sequentially combining several selected pieces of information (from the memory) in order to obtain a conclusion
  - Systems which answer questions
    - Accessing relevant facts and combining them



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

- Memory = part of the state
- Memory-based networks are special RNNs
- 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



# Pointing the Unknown Words

Gulcehre, Ahn, Nallapati, Zhou & Bengio ACL 2016

Based on 'Pointer Networks', Vinyals et al 2015

The next word generated can either come from vocabulary or is copied from the input sequence.

**French:** Guillaume et Cesar ont une voiture bleue a Lausanne.  
**English:** Guillaume and Cesar have a blue car in Lausanne.

Arrows labeled "Copy" point from the English words "Guillaume", "Cesar", and "Lausanne" to the corresponding French words "Guillaume", "Cesar", and "Lausanne".

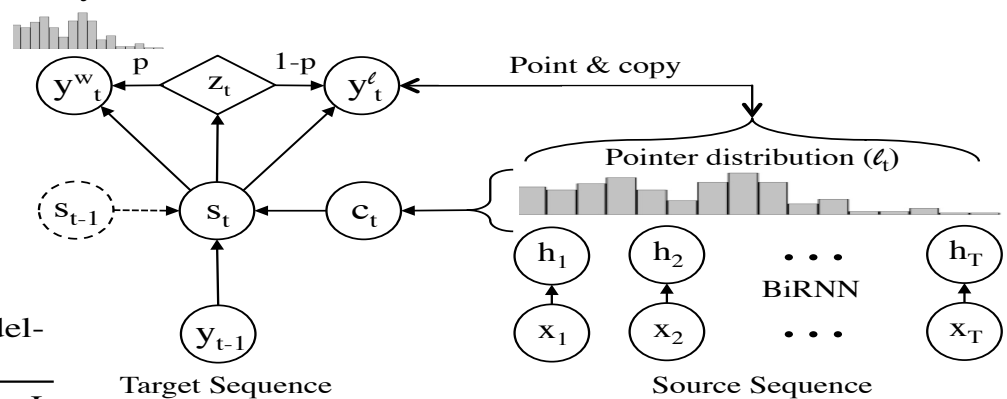
Table 5: Europarl Dataset (EN-FR)

Machine Translation	BLEU-4	
	NMT	NMT + PS
	20.19	<b>23.76</b>

Table 3: Results on Gigaword Corpus for modeling UNK's with pointers in terms of recall.

	Rouge-1	Rouge-2	Rouge-L
NMT + lvt	36.45	17.41	33.90
NMT + lvt + PS	<b>37.29</b>	<b>17.75</b>	<b>34.70</b>

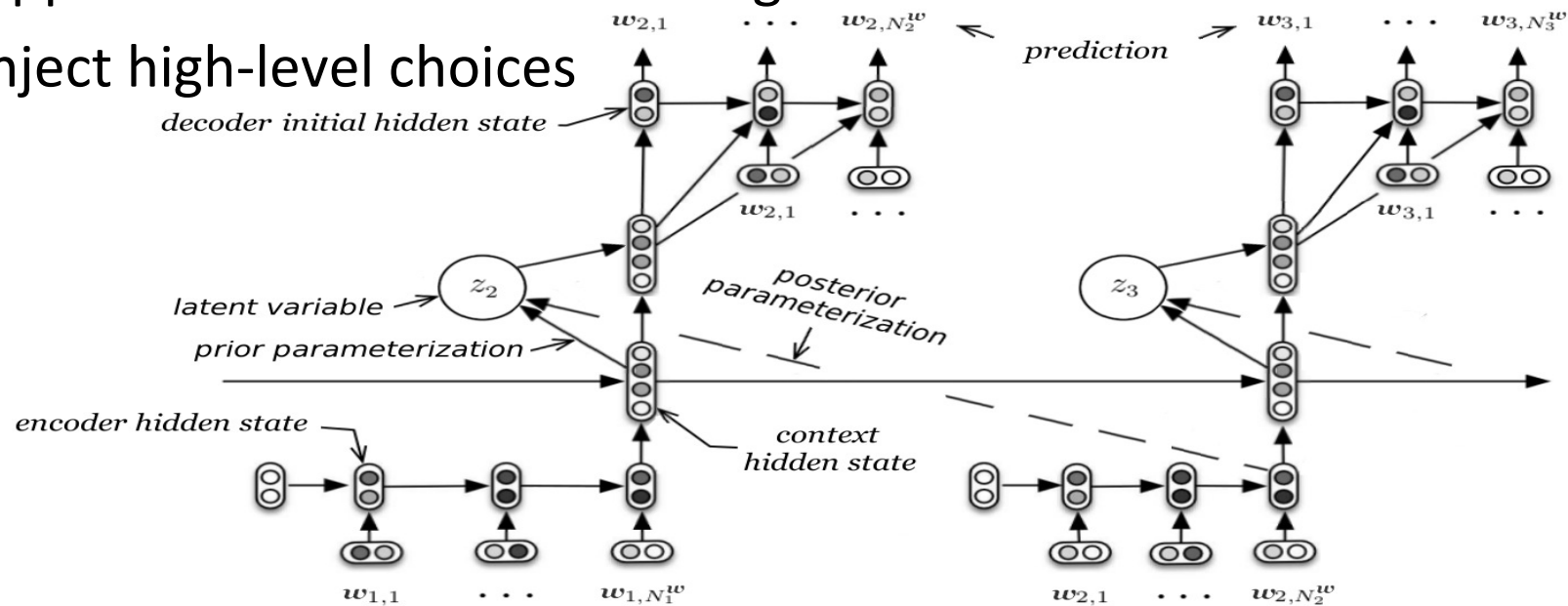
Vocabulary softmax



Text summarization

# Variational Hierarchical RNNs for Dialogue Generation (Serban et al 2016)

- Lower level = words of an utterance (turn of speech)
- Upper level = state of the dialogue
- Inject high-level choices



# Multi-Head Attention

We can run multiple attention mechanisms in parallel to focus on different aspects of the data

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$\text{MultiHeadAttention}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

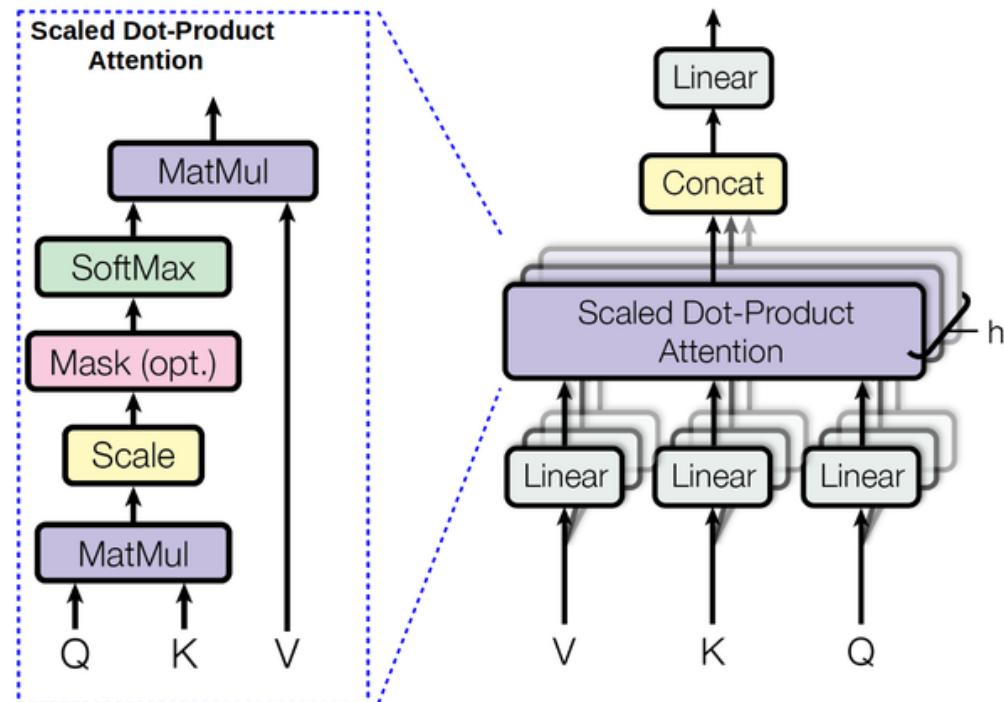


Fig: Michal Chromiak's blog

# Self-Attention & Transformers

*Vaswani et al Arxiv:1706.03762*

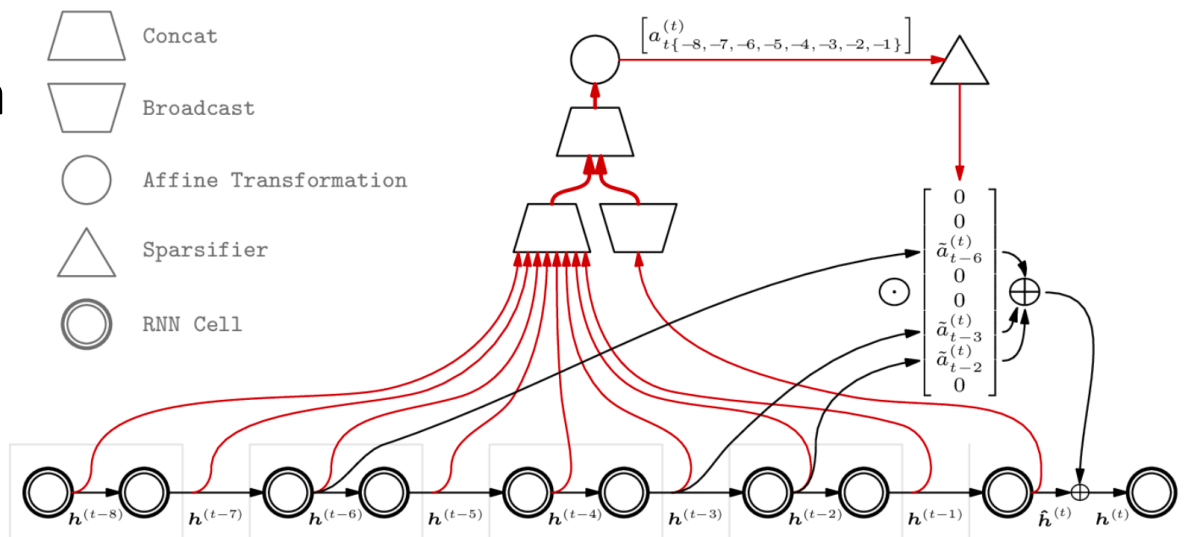
- **Parallelize** encoder
- Encode location of each item, no need for RNN
- Transform each location based on attention from all others
- See also Sparse Attentive Backtracking, Ke et al Arxiv: 1711.02326

*From: Jakob Uszkoreit, Google AI Blog, 2017*

# Using an Associative Memory to Bridge Large Time Spans and Avoid BPTT

*Self-Attentive Backtracking, Ke et al Arxiv: 1711.02326*

- Associate past and present events using a predictor, which acts like a trainable attentive skip connection between associated events
- Sparse attention to select few such events



May be a way for brains to avoid implausible BPTT



# Still Far from Human-Level AI

- Industrial successes mostly based on **supervised** learning

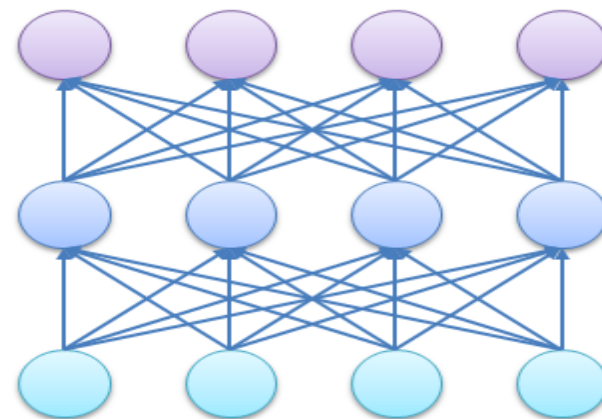


Adversarial  
example

- Learning superficial clues, not generalizing well outside of training contexts, easy to fool trained networks:
  - Current models cheat by picking on surface regularities
- Need to climb the ladder of higher-level abstractions

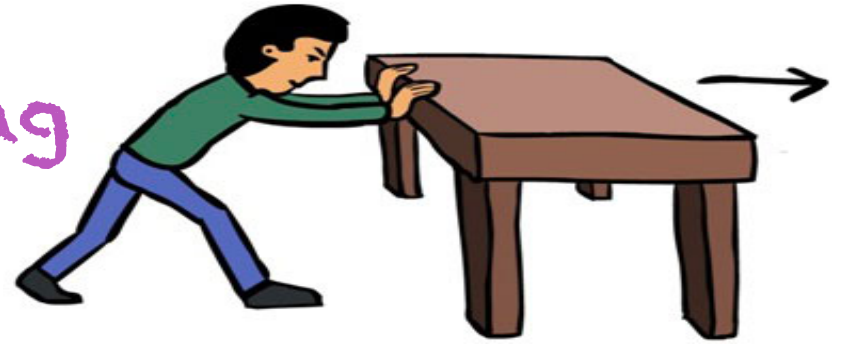
# How to Discover Good Disentangled Representations

- How to discover abstractions?
- What is a good representation? (*Bengio et al 2013*)
- Need clues (= priors) to help **disentangle** the underlying factors, e.g.
  - Spatial & temporal scales
  - Marginal independence
  - Simple dependencies between factors
    - *Consciousness prior*
  - Causal / mechanism independence
    - *Controllable factors*



# Acting to Guide Representation Learning & Disentangling

(E. Bengio et al, 2017; V. Thomas et al, 2017)



- **Some factors (e.g. objects) correspond to ‘independently controllable’ aspects of the world**
- *Can only be discovered by acting in the world*
  - *Control linked to notion of objects & agents*
  - *Causal but agent-specific & subjective: affordances*

# Abstraction Challenge for Unsupervised Learning

- Why is modeling  $P(\text{acoustics})$  so much worse than modeling  $P(\text{acoustics} \mid \text{phonemes})$   $P(\text{phonemes})$ ?
  - Wrong level of abstraction?
  - **many more entropy bits in acoustic details than linguistic content**
- **predict the future in in abstract space instead: non-trivial**

# The Consciousness Prior

Bengio 2017, arXiv:1709.08568

- Conscious thoughts are very low-dimensional objects compared to the full state of the (unconscious) brain
- Yet they have unexpected predictive value or usefulness
  - strong constraint or prior on the underlying state
- **Thought**: composition of few selected factors / concepts (key/value) at the highest level of abstraction of our brain
- Richer than but closely associated with short verbal expression such as a **sentence** or phrase, a **rule** or **fact** (link to classical symbolic AI & knowledge representation)



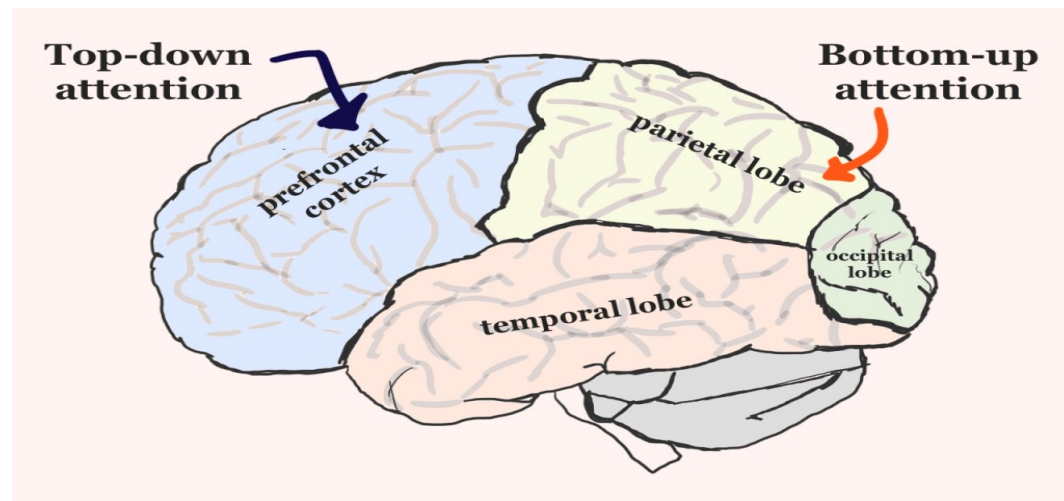
How to select a few  
relevant abstract  
concepts making a  
thought?

Content-based  
Attention

# On the Relation between Abstraction and Attention

- Attention allows to focus on a few elements out of a large set
- Soft-attention allows this process to be trainable with gradient-based optimization and backprop

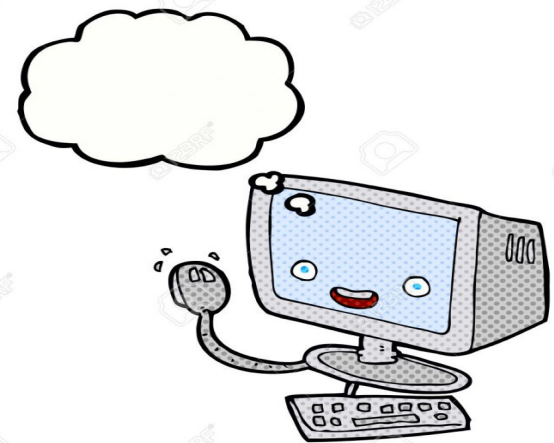
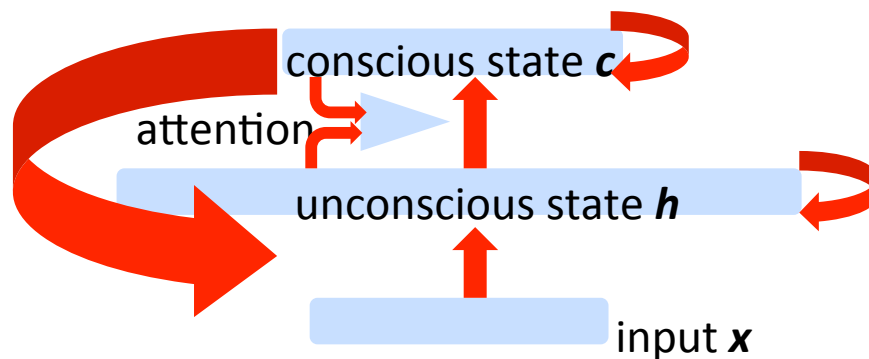
Attention focuses on a few appropriate abstract or concrete elements of mental representation



# The Consciousness Prior

Bengio 2017, arXiv:1709.08568

- 2 levels of representation:
  - High-dimensional abstract representation space (all known concepts and factors)  $h$
  - Low-dimensional conscious thought  $c$ ,





# Disentangling up to Linear Projection

- My old view of disentangling: each dimension of the representation = one 'nameable' (semantic) factor
- Potential problem: the number of 'nameable' factors is limited by the number of units, and brains don't use a completely localized representation for named things
- My current view of disentangling: it is enough that a linear projection exist to 'classify' or 'predict' any of the factors
- The 'number' of potential 'nameable' factors is now exponentially larger (e.g. subsets of dimensions, weights of these projections)

# The Consciousness Prior

Bengio 2017, arXiv:1709.08568

- Conscious prediction over attended variables  $A$  (soft attention)

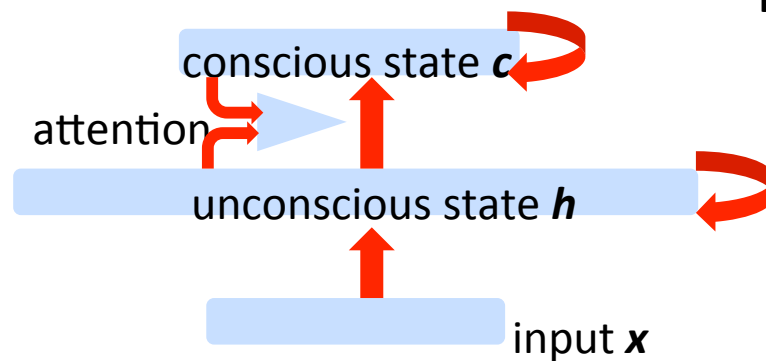
$$V = - \sum_A w_A \log P(h_{t,A} = a | c_{t-1})$$

Attention weights

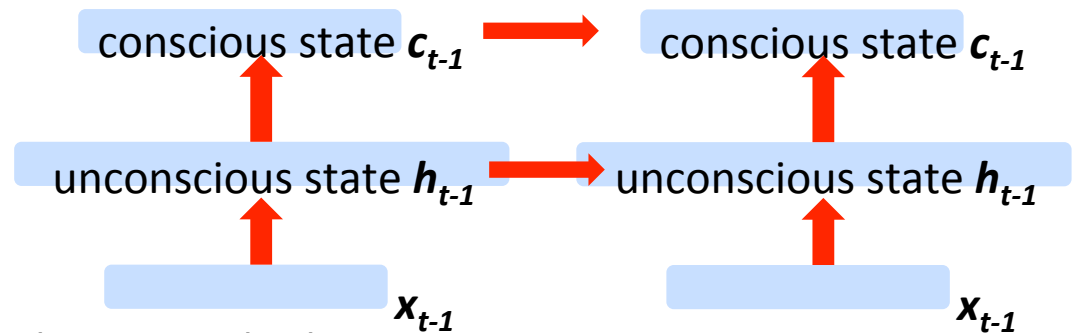
Factor  
name

Predicted  
value

Earlier conscious  
state



## What Training Objective?



- How to train the attention mechanism which selects which variables to predict?
  - Representation learning without reconstruction:
    - Maximize entropy of code
    - Maximize mutual information between past and future
  - *Objective function completely in abstract space, higher-level parameters model dependencies in abstract space*
  - *Usefulness of thoughts: as conditioning information for action, i.e., a particular form of planning for RL, i.e., the estimated gradient of rewards could also be used to drive learning of abstract representations*



# Montreal Institute for Learning Algorithms

Université   
de Montréal