Recurrent Neural Network & Long Short-Term Memory

Sangwoong Yoon

Connectionist Model Seminar
Distributable Version
Nov 13, 2014

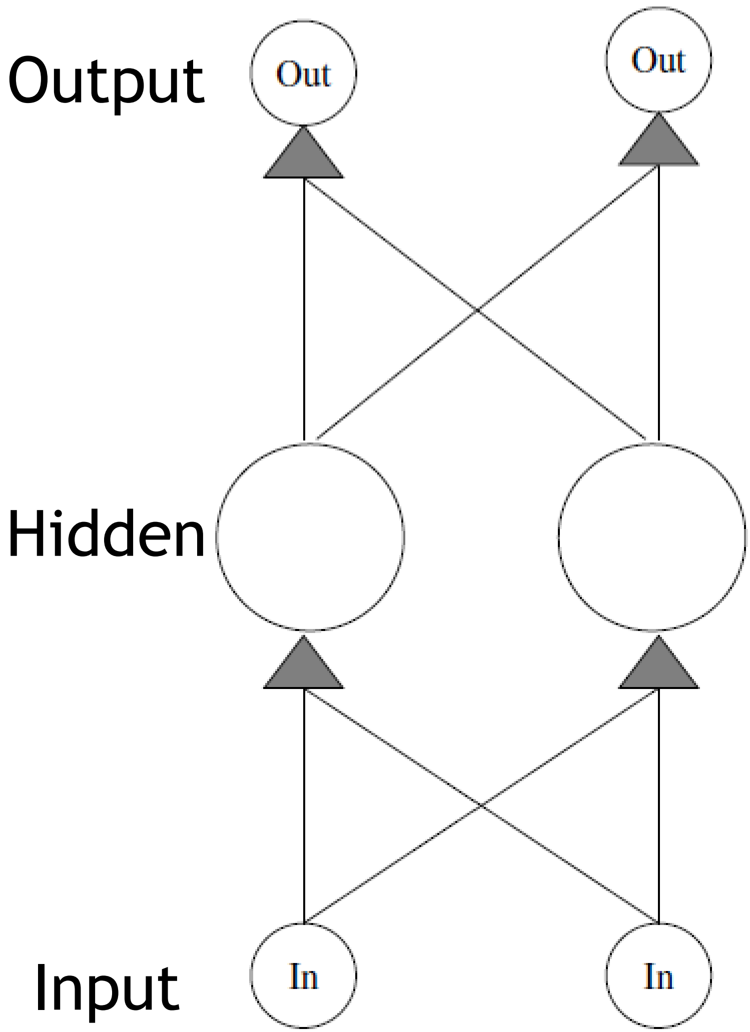
Biointelligence Laboratory
Seoul National University

Contents

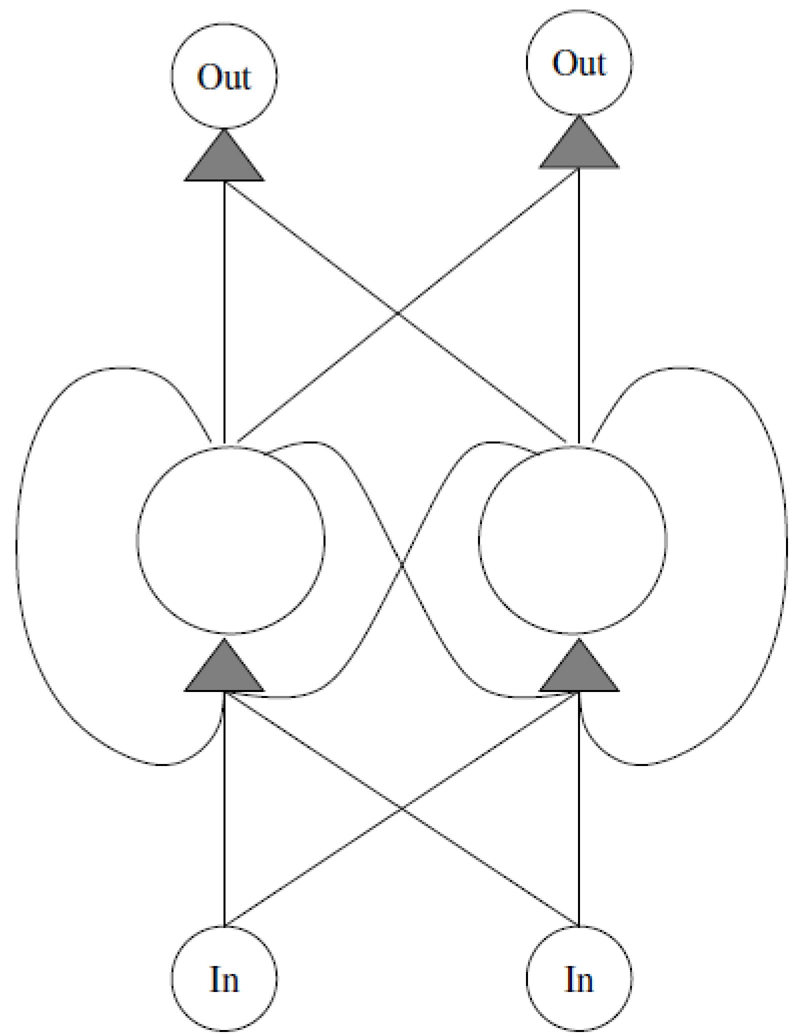
Part I

A Review on Recurrent Neural Networks

Recurrent Neural Networks (RNNs)

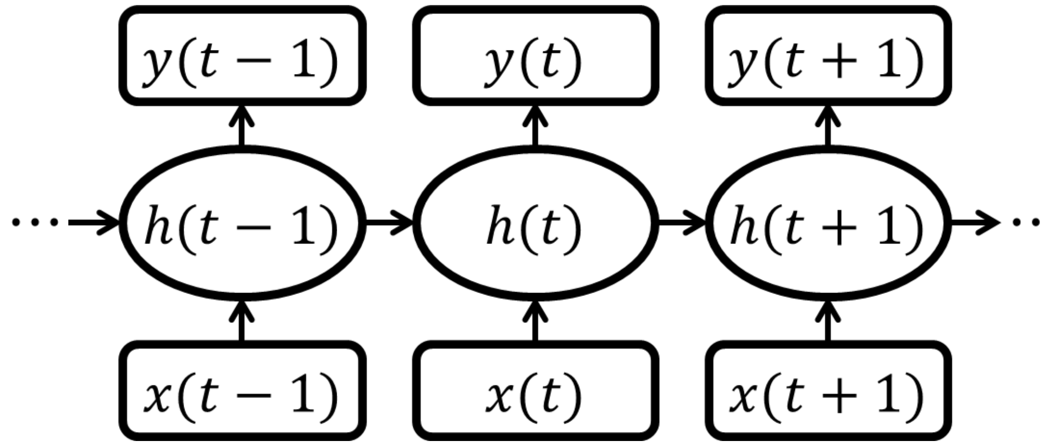


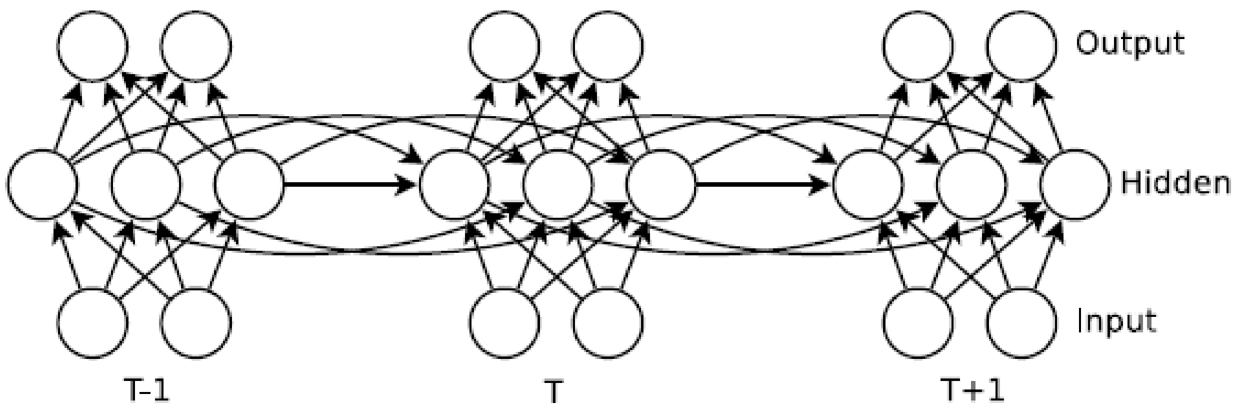
Multi-Layer Perceptron (MLP)



Recurrent Neural Network

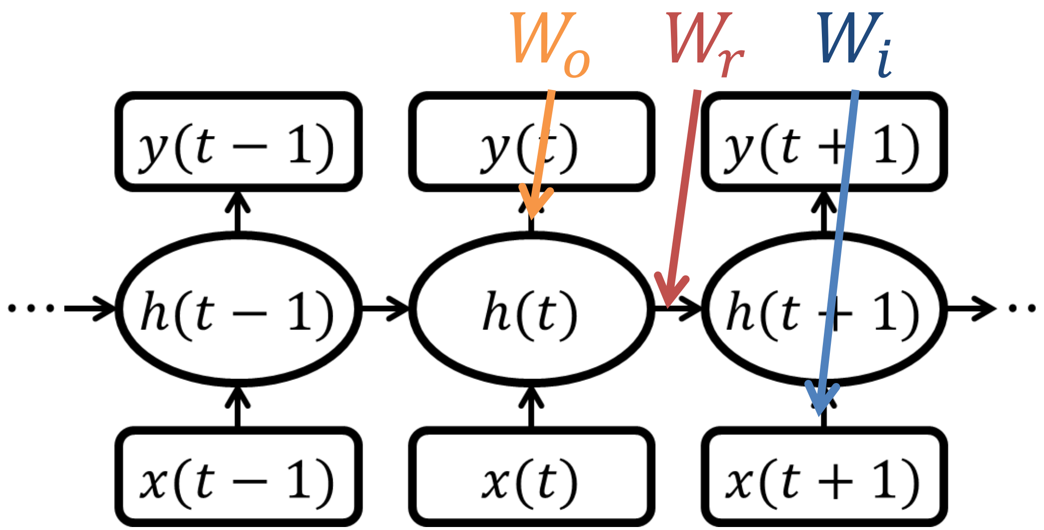
Recurrent Neural Networks (RNNs)





The figure from (Sutskever et al., 2011 [1])

Recurrent Neural Networks (RNNs)

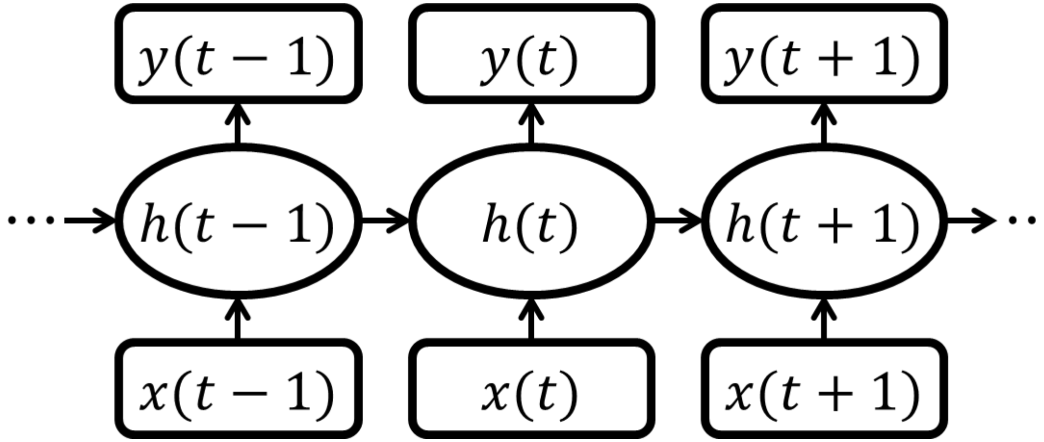


$$f: \big(x(1), \dots, x(t)\big) \rightarrow y(t)$$

$$h(t) = \sigma(W_i x(t) + W_r h(t-1))$$

$$y(t) = \sigma(W_o h(t))$$

Remarks



- Temporal, sequential model
- Big degree of freedom on structure
  - Many, many models has been proposed
  - Standard form : Elman network (Elman, 1990 [2])
- Training : Any optimization method

Why RNNs?

1. **Natural and powerful**
   - Natural : Sliding window
   - Powerful : Hidden Markov Model

2. **“Something”**
   - Most close to real neural networks
   - Many other interpretations

1. Natural and Powerful

VS Time Window Approach

- Hand craft window size
- Dependency longer than window size
- Multiple time scale dependency
- Changing dependency

(Gers, 2001)

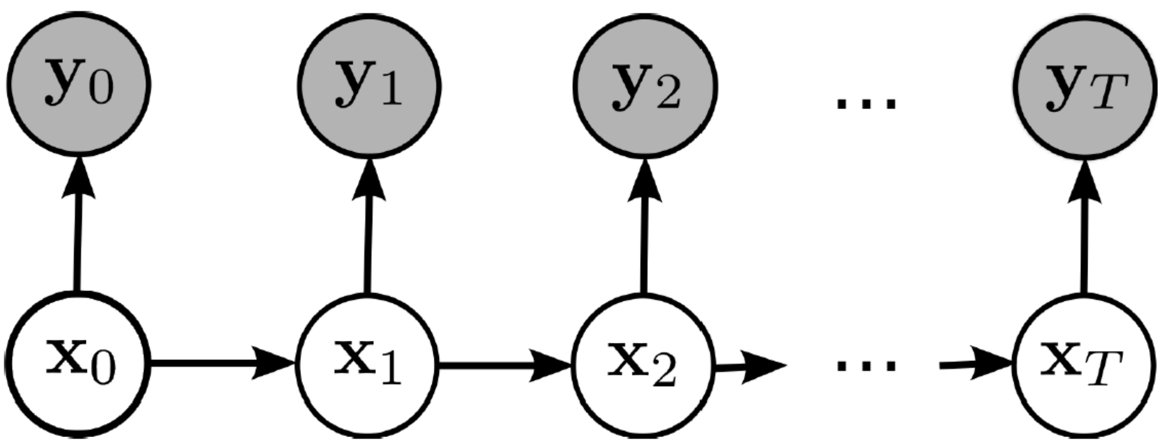
1	2	3	4	5	6
1	2	3	4	5	6
1	2	3	4	5	6

Window size: 3

1. Natural and Powerful

VS Hidden Markov Model

- Continuous & combinatorial hidden
- Bigger memory (ex) counting task
- PGMs have their own strength

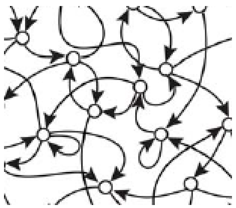


(Image: http://iacs-courses.seas.harvard.edu/courses/am207/blog/lecture-18.html)

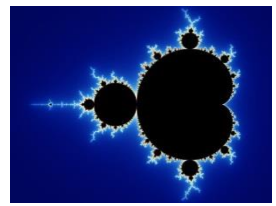
2. "Something"

- Other possible views on RNNs

Neuroscience



Complexity



Dynamic Systems

$$\frac{\partial y}{\partial t} = f(y(t))$$

Statistical Physics

$$Z = \sum e^{-\beta E}$$

Theory of Computation

0	1	1	1	1	0	1	1	1	0

Reinforcement Learning

Big Questions for RNNs

Q1. Learning long-term dependency

- Short-term RNN is meaningless
- Vanishing gradient (Pascanu et al., 2013 [3])

Q2. Expressive Power & Structure

- Is a standard RNN strong enough?
- If not, what do we need more?

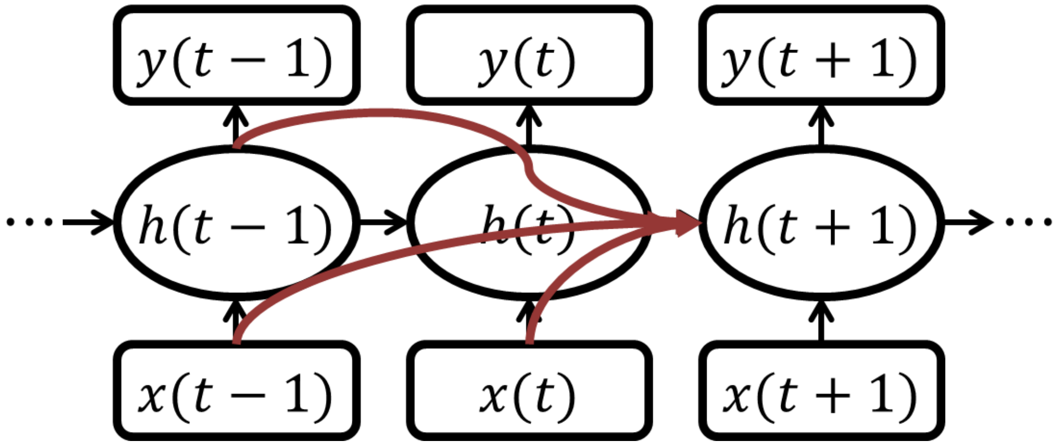
Q1 Long-Term Dependency

NARX RNN (Lin et al., 1996 [5])

Nonlinear Autoregressive Models with Exogenous Inputs

$$y(t) = f[u(t - D_u), \cdots, u(t-1), u(t), \\ y(t - D_y), \cdots, y(t-1)]$$

u : input, y: hidden



x : input, h: hidden, y: output

NARX

ESN

RNNLM

HF

Q1 Long-Term Dependency

NARX RNN (Lin et al., 1996 [5])

Nonlinear Autoregressive Models with Exogenous Inputs

Remarks

- "Skip connection"
- Manually setting D
- Similar : Time Delay Neural Networks

  (TDNN, Haffner and Waibel, 1992 [6])

NARX

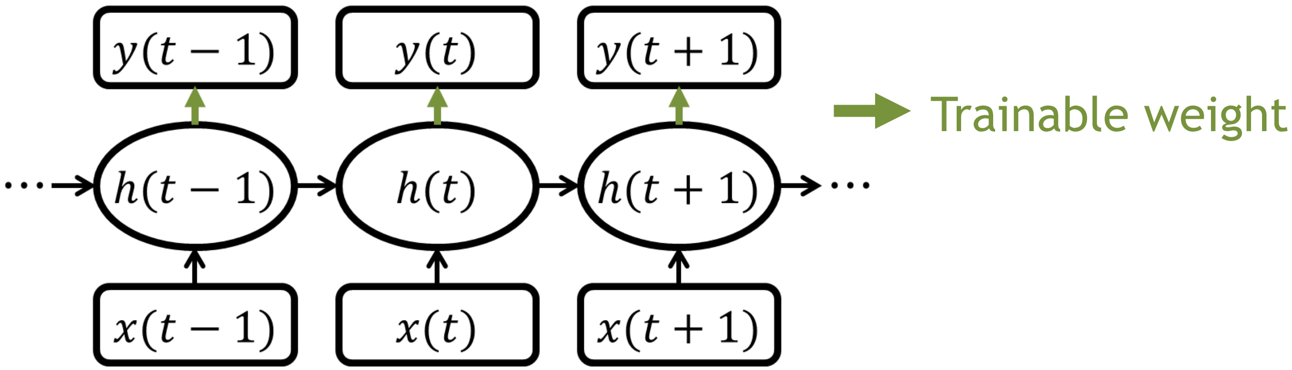
ESN

RNNLM

HF

Q1 Long-Term Dependency

Echo State Network (Jaeger, 2001 [7])



- **"Echo state property"**
  - W_r: Sparse random, spectral radius < 1
- Very robust & long term memory
- Implicit constraint: $Dim(h) \gg Dim(x)$
  - h acts like a kernel
- Why not optimize?

NARX

ESN

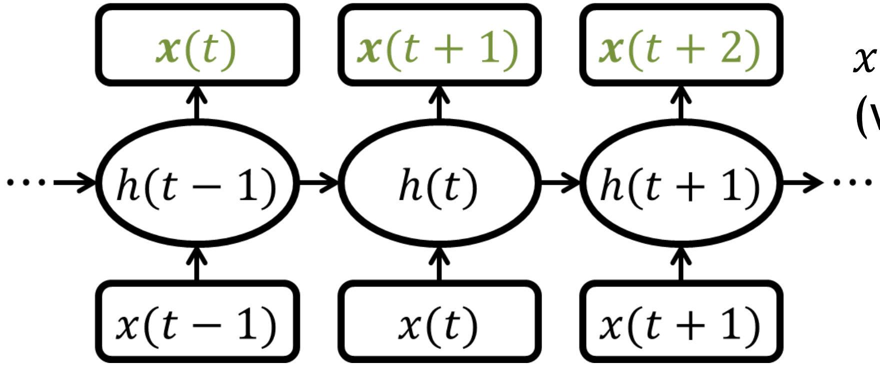
RNNLM

HF

Q1 Long-Term Dependency

RNN Language Model (Mikolov, 2010 [8])

http://rnnlm.org/



x(t) : Language token (word, char)

- State-of-the-art performance
- **SURPRISING !**

  - The first practical application (non-LSTM)
  - Simple, ordinary Elman RNN
  - Simple, ordinary back-propagation

NARX

ESN

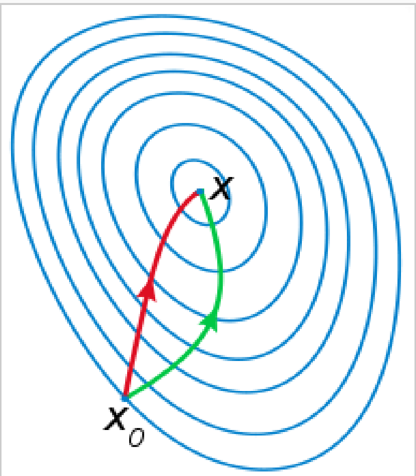
RNNLM

HF

Q1 Long-Term Dependency

Hessian-Free Optimization

(Martens and Sutskever, 2011 [9])



A comparison of gradient descent (green) and Newton's method (red) for minimizing a function (with small step sizes). Newton's method uses curvature information to take a more direct route.

(Wikipedia)

- Second-order optimization
- May not suffer from vanishing gradients
- Beat LSTM, ESN in long-term memory task

Q2 Structure

2nd-Order RNN

(Goudreau, Giles, et al., 1994 [10])

- **Product term**
- Many kinds of possible product exists

$$h_i(t) = \sigma\left(\sum_{j,k} w_{ijk} z_j z_k\right)$$

$$z_j \in \{\, h_l(t-1) \,\} \cup \{\, x_l(t) \}$$

Comparison : 1st-order RNN

$$h(t) = \sigma(W_i x(t) + W_r h(t-1))$$

2nd-Order

Universal?

MRNN

DRNN

DRNN(LISA)

Q2 Structure

Is a RNN Powerful Enough?

On the Computational Power of Neural Nets*

HAVA T. SIEGELMANN†

Department of Information Systems Engineering, Technion, Haifa 32000, Israel

AND

EDUARDO D. SONTAG‡

Department of Mathematics, Rutgers University, New Brunswick, New Jersey 08903

Received February 4, 1992; revised May 24, 1993

(Siegelmann and Sontag, 1993 [11])

- First-order RNN can simulate all Turing machines
- Products terms are not needed
- However…

2nd-Order

Universal?

MRNN

DRNN

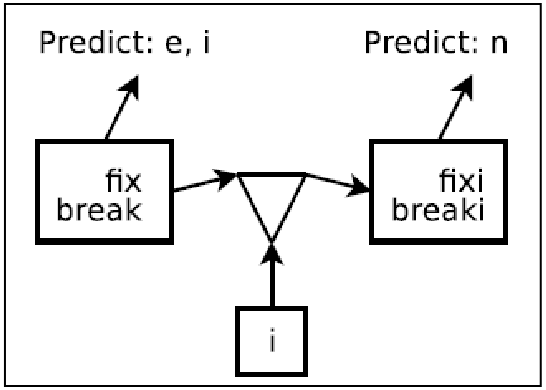
DRNN(LISA)

Q2 Structure

Multiplicative RNN

(Sutskever et al., 2011 [1])

- Character-level LM
- Characters seems to have a multiplicative connection
- Tensor factorization + HF



$$h_t = \tanh\left(W_{hx}x_t + \underline{W_{hh}^{(x_t)}}h_{t-1} + b_h\right)$$

$$o_t = W_{oh}h_t + b_o$$

$$\underline{W_{hh}^{(x_t)}} = \sum_{m=1}^{M} x_t^{(m)} W_{hh}^{(m)}$$

Structure contains prior, and it has to be consistent with the data

2nd-Order

Universal?

MRNN

DRNN

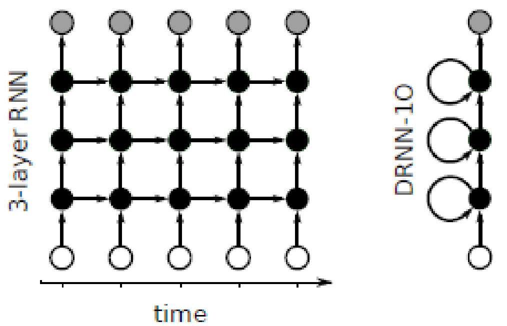
DRNN(LISA)

5 days with 8 GPUs…

Q2 Structure

Deep Recurrent Neural Network

(Hermans and Schrauwen, 2013 [12])



Model	BPC test
RNN	1.610
DRNN-AO	1.557
DRNN-1O	1.541
MRNN	1.55
PAQ	1.51
Hutter Prize (current record) [12]	1.276
Human level (estimated) [18]	0.6 – 1.3

5 days with 8 GPUs…

- Intuitive, but naïve
- Can be reduced to a shallow one
- Not clear what kind of prior the structure contains

2nd-Order

Universal?

MRNN

DRNN

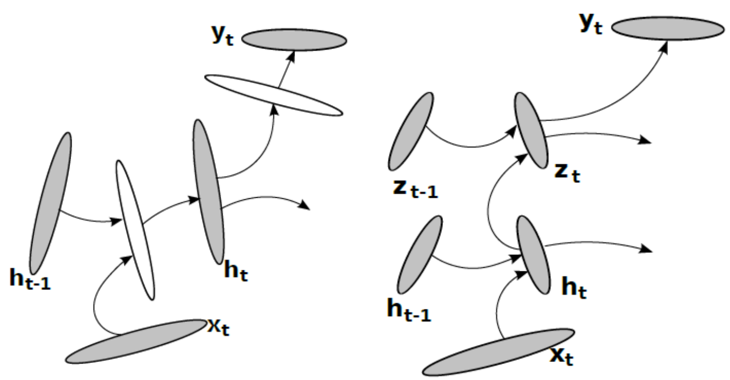
DRNN(LISA)

Q2 Structure

Deep Recurrent Neural Network

(Pascanu et al., 2014 [13])

The "deep connection"
is multiple non-linear transformation



(c) DOT-RNN (d) Stacked RNN

- MLP (thus arbitrary transformation) between each layer

2nd-Order

Universal?

MRNN

DRNN

DRNN(LISA)

Summary of the History

Q1 Can a RNN learn a long-range correlation?

Q2 Is the structure capable enough?

Incredible improvements have been made

Maybe now we can really do something with RNNs

The Long Short-Term Memory solves

Q1 and Q2 simultaneously

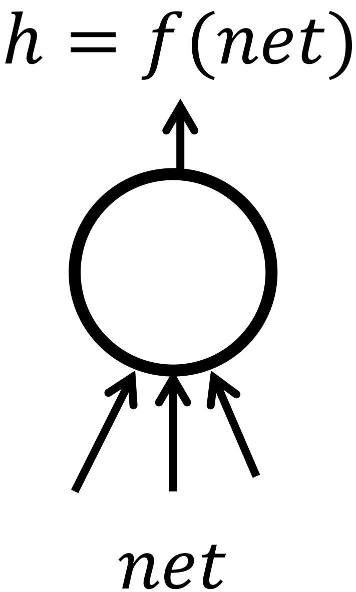
Part II

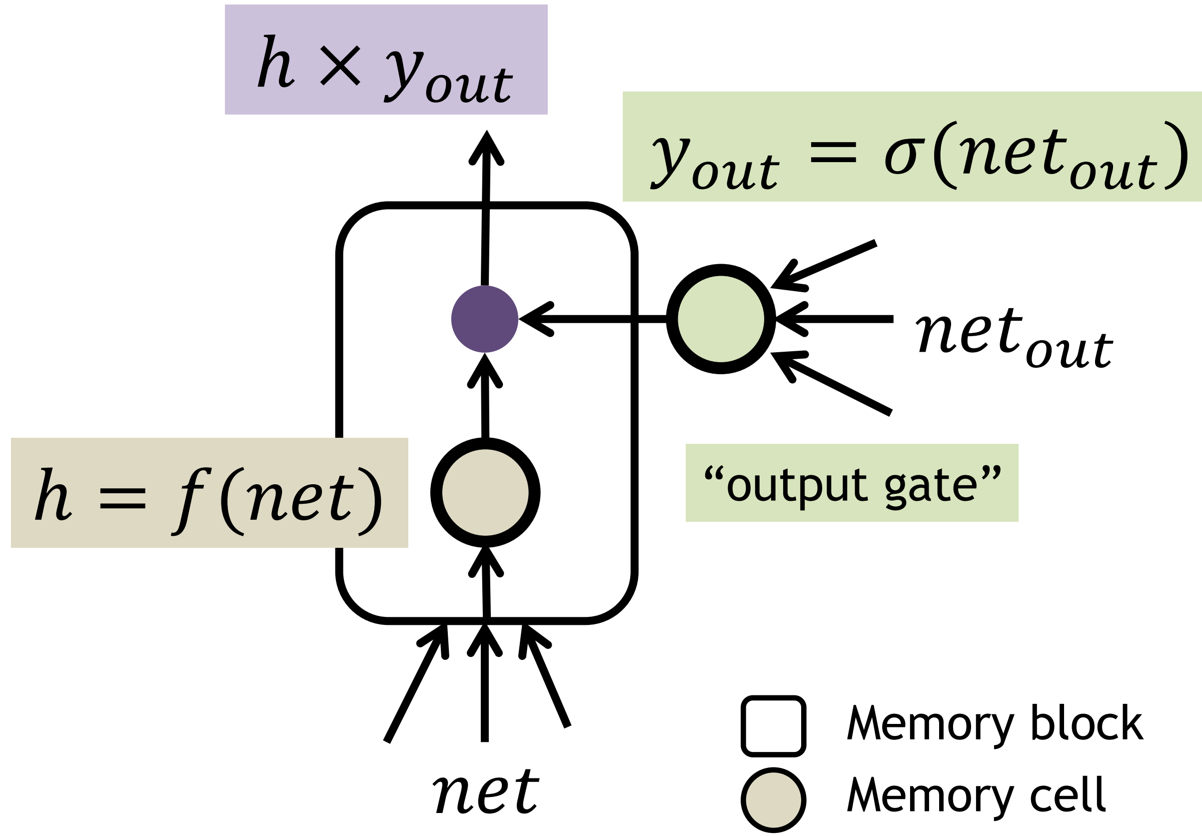
Long Short-Term Memory

(Hochreiter and Schmidhuber, 1997 [14])
(Gers, 2001 [4])

Let's Modify a Hidden Neuron a Little Bit…

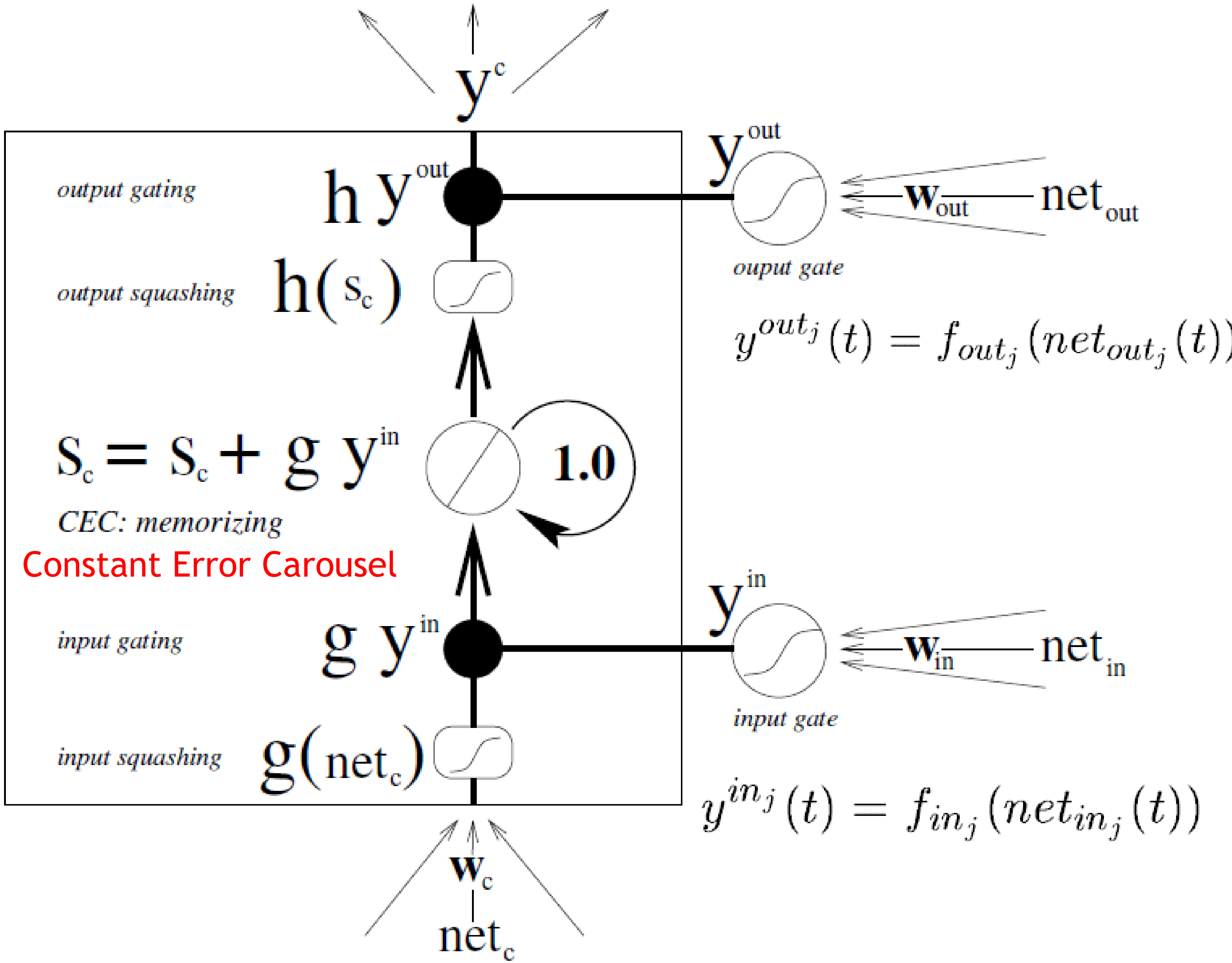
Standard RNN

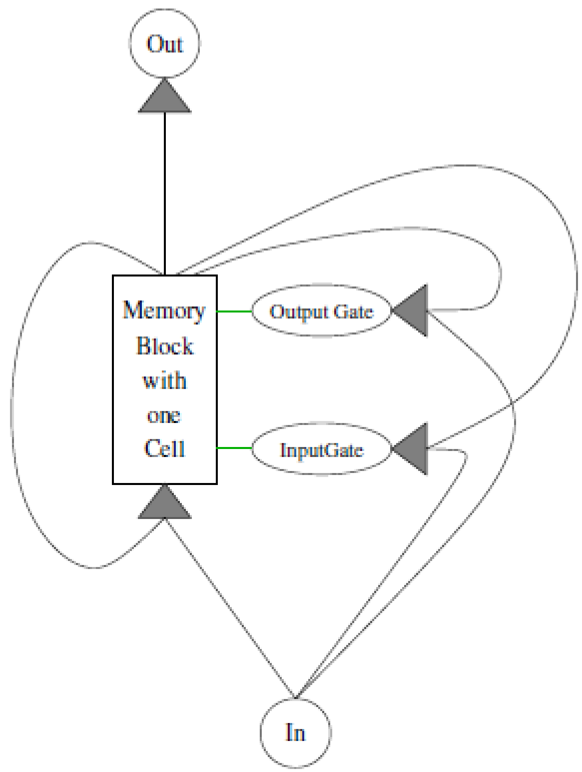




$net = W_{in}x(t) + W_r h(t)$
f: any non-linearity
σ: sigmoid

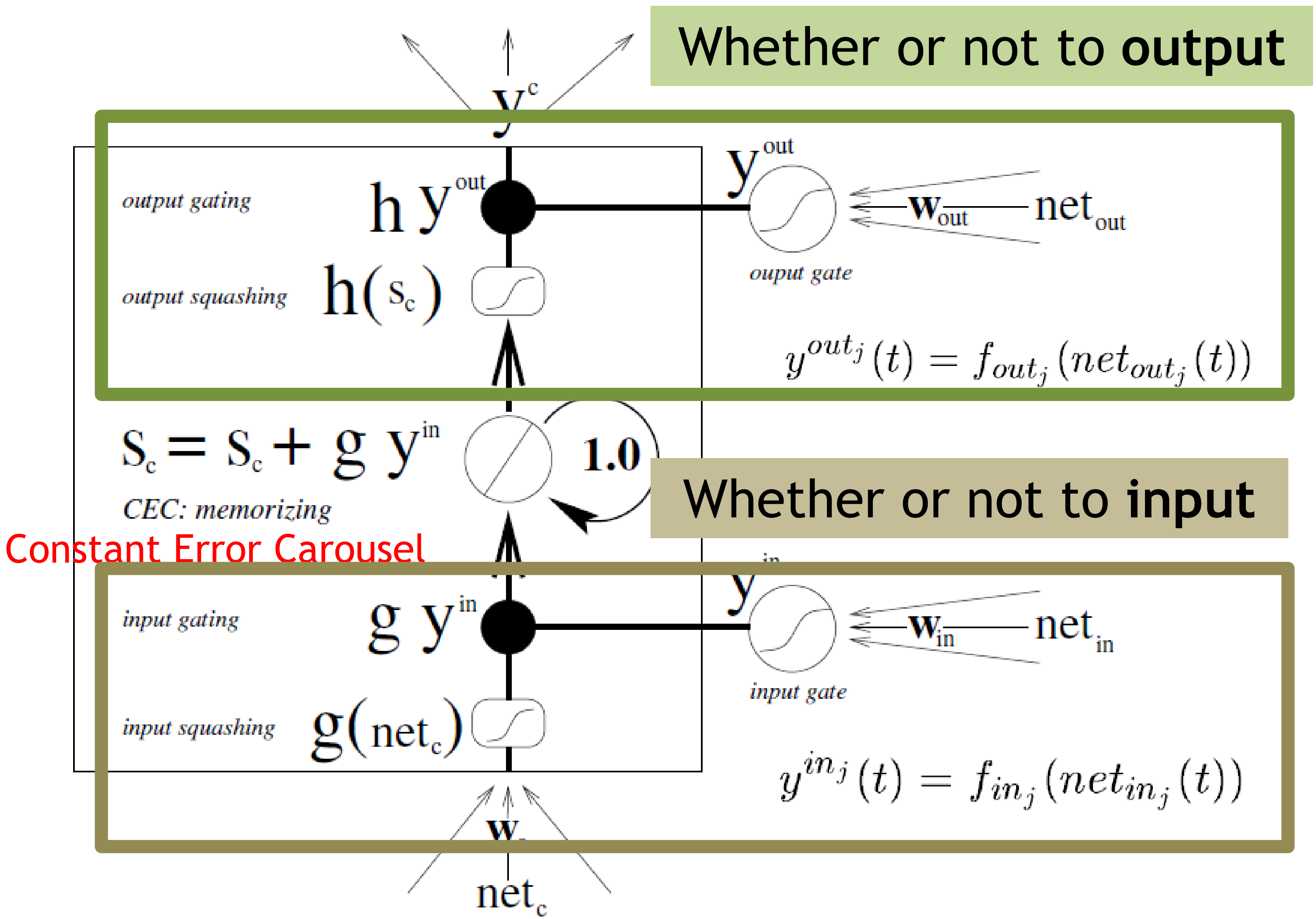
Make an "input gate" like this


y^c
output gating
h y^out
y^out
w_out
net_out
ouput gate
output squashing
h(s_c)
y^{out_j}(t) = f_{out_j}(net_{out_j}(t))
s_c = s_c + g y^in
1.0
CEC: memorizing
Constant Error Carousel
input gating
g y^in
y^in
w_in
net_in
input gate
input squashing
g(net_c)
y^{in_j}(t) = f_{in_j}(net_{in_j}(t))
w_c
net_c


Out
Memory Block with one Cell
Output Gate
InputGate
In

WHY?????????????

1. Stronger Expressive Power

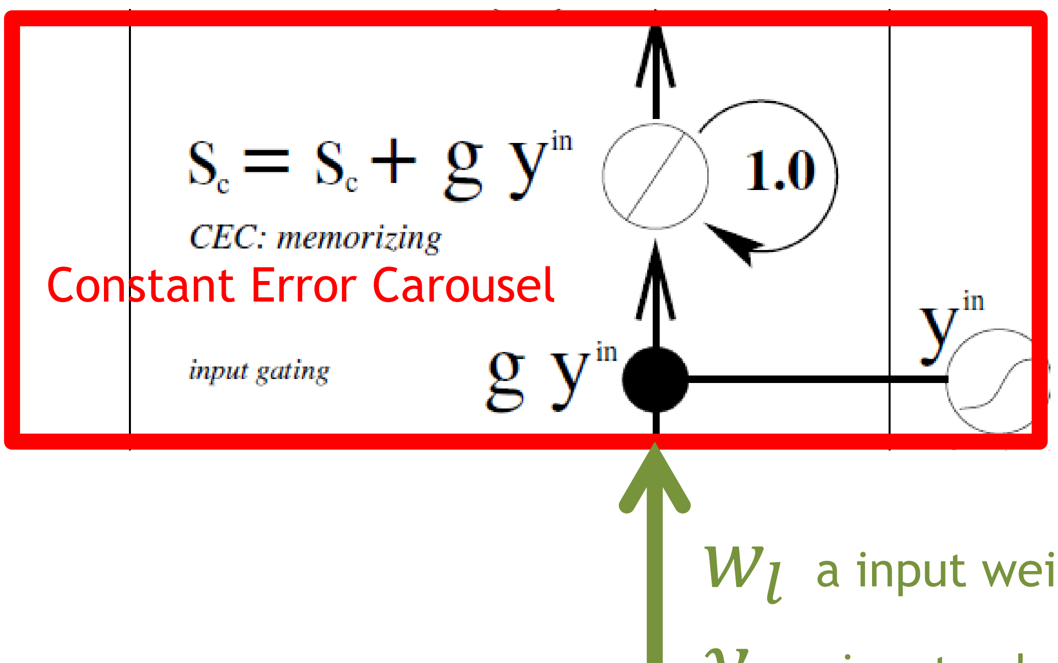


2. Non-Vanishing Gradient

$$\frac{\partial E}{\partial w_l} = \frac{\partial E}{\partial s_t}\frac{\partial s_t}{\partial w_l} = \frac{\partial E}{\partial s_t}\frac{\partial s_t}{\partial s_{t-1}} \cdots \frac{\partial s_{l+1}}{\partial s_l}\frac{\partial s_l}{\partial w_l}$$

$$\frac{\partial s_t}{\partial s_{t-1}} = 1$$

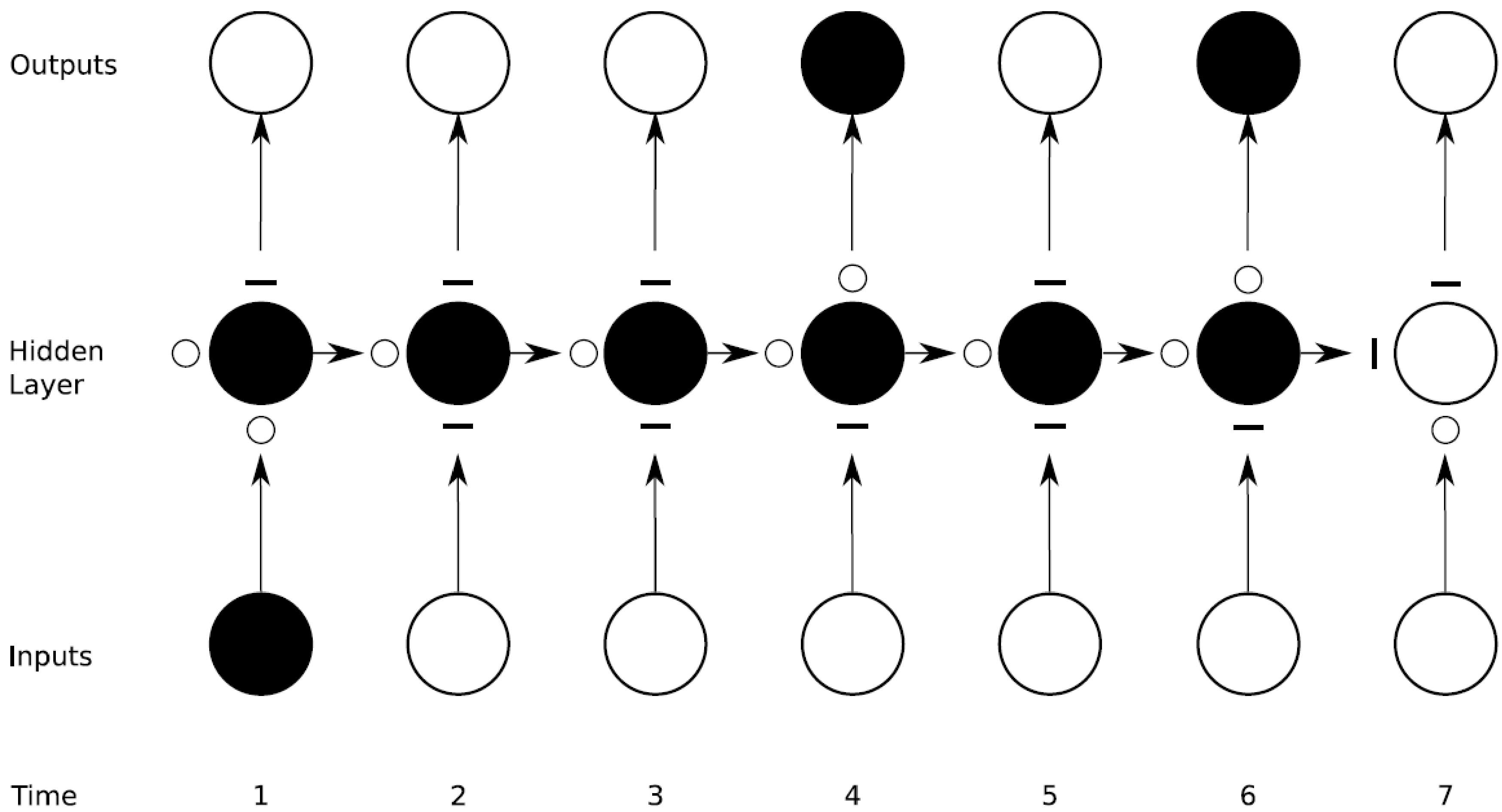
$$\frac{\partial s_l}{\partial w_l} = y_l g'(net_l) y^{in}$$



w_l a input weight at time l

y_l a input value at time l

2. Non-Vanishing Gradient

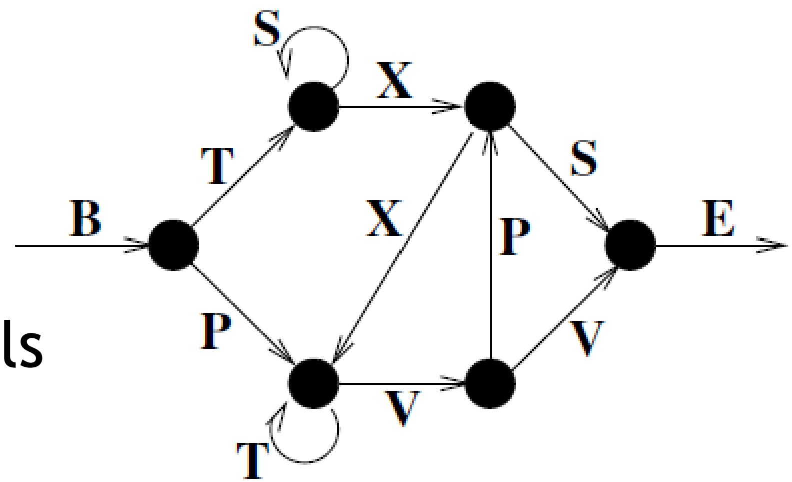


(Graves, 2012 [19])

O : open gate , — : closed gate

Reber Grammar

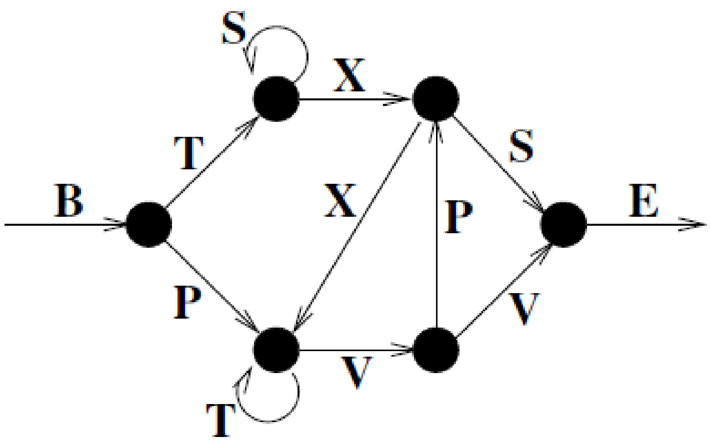
- Input, output dim : 7
- To predict next possible symbols

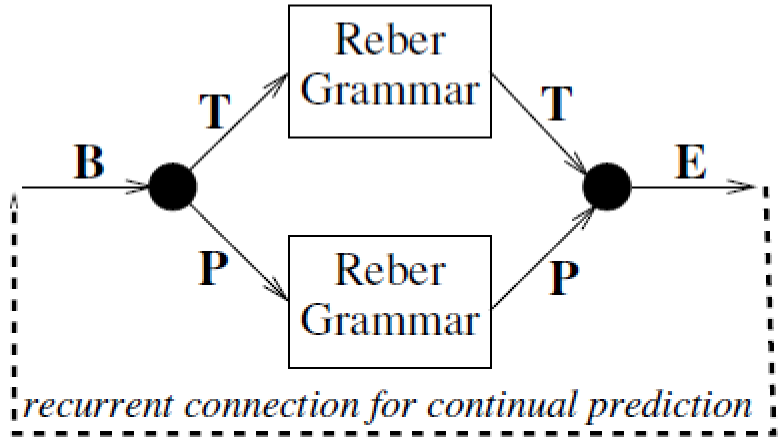


Algo-rithm	# hidden units	#weights	learning rate	% of success	success after
RTRL	3	≈ 170	0.05	"some fraction"	173,000
RTRL	12	≈ 494	0.1	"some fraction"	25,000
ELM	15	≈ 435		0	>200,000
RCC	7-9	≈ 119-198		50	182,000
Tra. LSTM	3bl.,size 2	276	0.5	100	8,440

Table 3.1: Standard embedded Reber grammar (ERG): percentage of successful trials and number of sequence presentations until success for RTRL (results taken from Smith and Zipser 1989), "Elman net trained by Elman's procedure" (results taken from Cleeremans et al. 1989), "Recurrent Cascade-Correlation" (results taken from Fahlman 1991) and traditional LSTM (results taken from Hochreiter and Schmidhuber 1997). Weight numbers in the first 4 rows are estimates.

Continual Reber Grammar



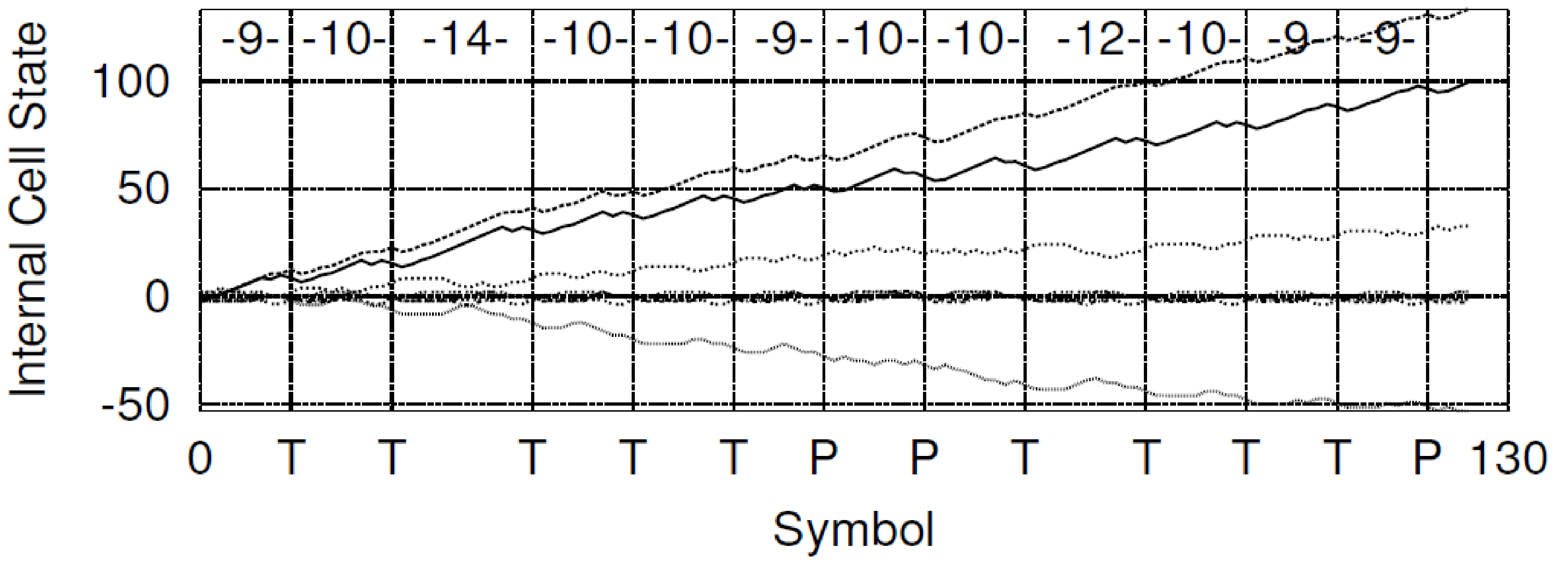


Algorithm	%Solutions	%Good Sol.	%Rest
Tra. LSTM with external reset	74 (7441)	0 ⟨–⟩	26 ⟨31⟩
Traditional LSTM	0 (-)	1 ⟨1166⟩	99 ⟨37⟩
LSTM with State Decay (0.9)	0 (-)	0 ⟨–⟩	100 ⟨56⟩

LSTM fails completely!

%Solutions : correct for a whole sequence (100,000 symbols)
%Good : correct > 1000

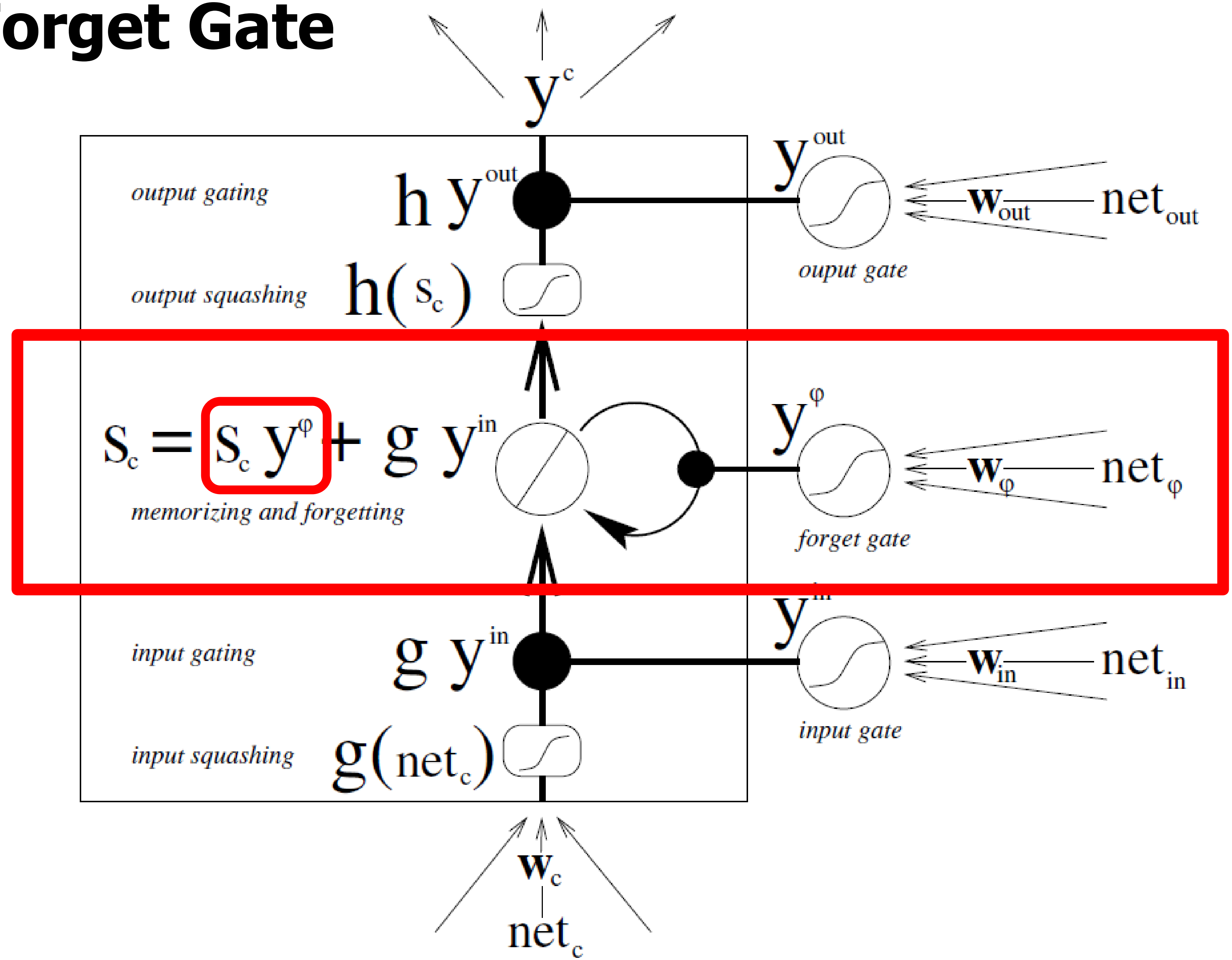
Fail, Why?



Memory cell activation diverges!

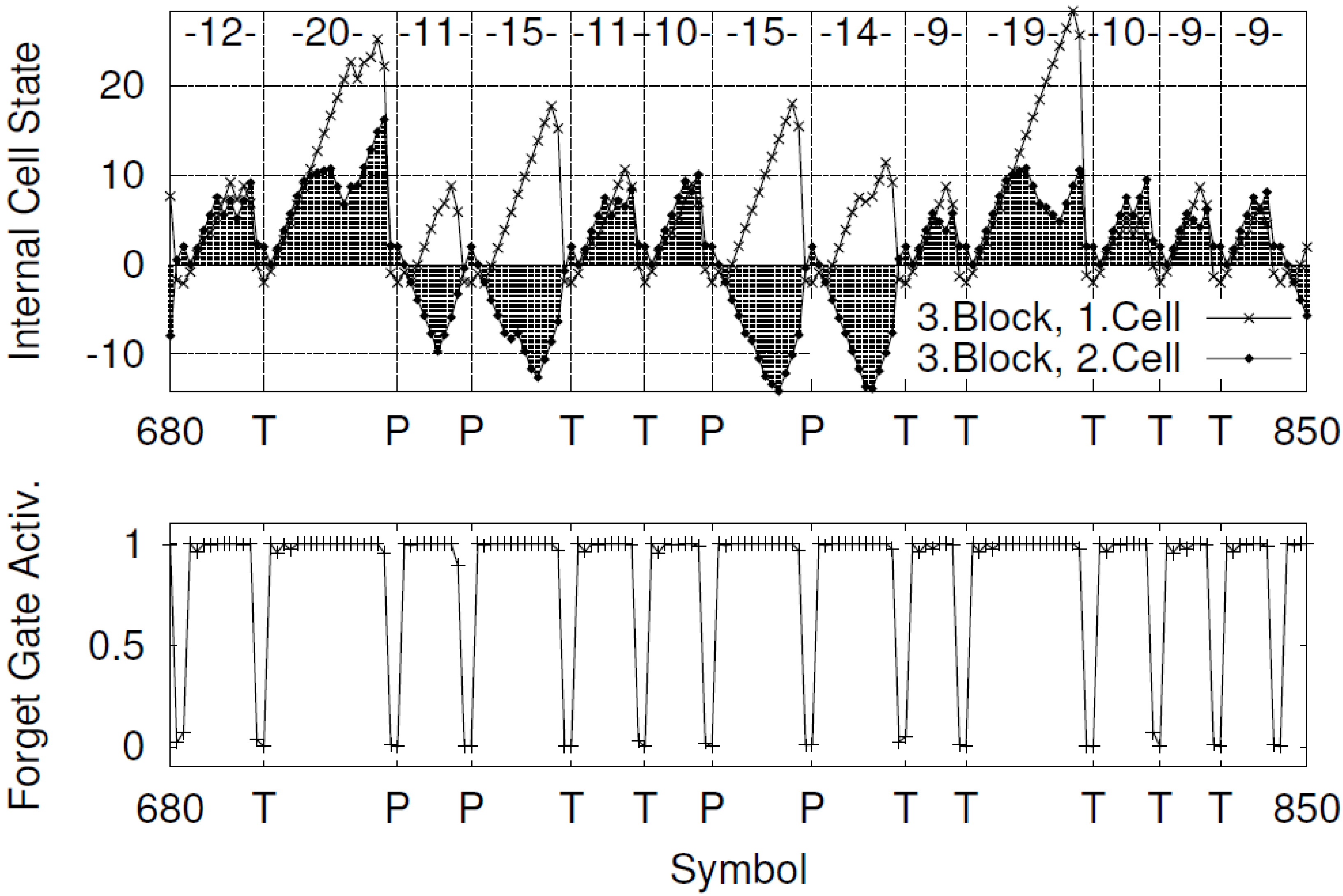
Then, let it forget!

Forget Gate



y^c
output gating
$h\, y^{out}$
output squashing
$h(s_c)$
y^{out}
w_{out}
net_{out}
ouput gate
$s_c = s_c\, y^{\varphi} + g\, y^{in}$
memorizing and forgetting
y^{φ}
w_{φ}
net_{φ}
forget gate
y^{in}
input gating
$g\, y^{in}$
w_{in}
net_{in}
input gate
input squashing
$g(net_c)$
w_c
net_c

With Forget Gates



With Forget Gates

Algorithm	%Solutions	%Good Sol.	%Rest
Tra. LSTM with external reset	74 (7441)	0 $\langle - \rangle$	26 $\langle 31 \rangle$
Traditional LSTM	0 (-)	1 $\langle 1166 \rangle$	99 $\langle 37 \rangle$
LSTM with State Decay (0.9)	0 (-)	0 $\langle - \rangle$	100 $\langle 56 \rangle$
LSTM with Forget Gates	18 (18889)	29 $\langle 39171 \rangle$	53 $\langle 145 \rangle$
LSTM with Forget Gates and sequential α decay	62 (14087)	6 $\langle 68464 \rangle$	32 $\langle 30 \rangle$

Remarks on LSTM

- Not as messy as it looks (?)
- One-step computation is expensive (?)
- A second-order RNN
- Learning is mainly GD
  - A few algorithm has been proposed
- Peephole connection is added (Gers et al., 2003 [15])
- Design issue
  - A memory block can contain multiple memory cells
  - Not all gates are necessary – Which gates to use?

Recent Trends on LSTM

Alex Graves (at Google DeepMind)

- Most recent, state-of-the-art LSTM works
- (http://www.cs.toronto.edu/~graves/)
- **Supervised sequence labeling**
  - Handwriting recognition / generation
  - Speech recognition (Graves and Jaitly, 2014 [16])
  - Connectionist Temporal Classification
  - Bidirectional RNN

✓ One more : LSTM + Dropout
(Zaremba, Sutskever and Vinyals, 2014 [17])

Part III

Future Research Direction

Important Questions

Q1 So many models. Which one is the best?

- Theoretical tool?

Q2 Big model, big data, yet limited performance. What should we do more?

- Maybe need to redefine the problem

Q3 What other tasks can we do with RNNs?

Reference

[1] I. Sutskever, J. Martens, and G. E. Hinton. "Generating text with recurrent neural networks." *ICML-11*. 2011.
[2] J. Elman. "Finding structure in time." *Cognitive science*. 1990.
[3] R. Pascanu, T. Mikolov, and Y. Bengio. "On the difficulty of training recurrent neural networks." *arXiv preprint arXiv:1211.5063*. 2012.
[4] F. Gers. “Long Short-Term Memory in Recurrent Neural Networks”. Ecole Polytechnique Federale De Lausanne, PhD thesis. 2001.
[5] Lin, Tsungnan, et al. "Learning long-term dependencies in NARX recurrent neural networks." *Neural Networks, IEEE Transactions on*. 1996.
[6] P. Haffner, and A. Waibel. "Multi-state time delay networks for continuous speech recognition“. NIPS. 1992.
[7] H. Jaeger, “The “echo state” approach to analysing and training recurrent neural networks”, GMD Technical Report, 2001.
[8] T. Mikolov, M. Karafiát, L. Burget, J. Èernocký, and S. Khudanpur. “Recurrent neural network based language model”. INTERSPEECH 2010.
[9] J. Martens, and I. Sutskever. "Learning recurrent neural networks with hessian-free optimization." *ICML-11*. 2011.
[10] M. Goudreau, et al. "First-order versus second-order single-layer recurrent neural networks." *Neural Networks, IEEE Transactions on*. 1994.

Reference

[11] H. Siegelmann, and E. Sontag. "On the computational power of neural nets." *Journal of computer and system sciences*. 1995.
[12] M. Hermans, and B. Schrauwen. "Training and analysing deep recurrent neural networks." *NIPS*. 2013.
[13] R. Pascanu, et al. "How to Construct Deep Recurrent Neural Networks." arXiv preprint arXiv:1312.6026. 2013.
[14] S. Hochreiter, and J. Schmidhuber. "Long short-term memory." Neural computation. 1997.
[15] F. Gers, N. Schraudolph, and J. Schmidhuber. "Learning precise timing with LSTM recurrent networks." The Journal of Machine Learning Research. 2003.
[16] A. Graves, and N. Jaitly. "Towards end-to-end speech recognition with recurrent neural networks." ICML-14. 2014.
[17] W. Zaremba, I. Sutskever, and O. Vinyals. "Recurrent Neural Network Regularization." *arXiv preprint arXiv:1409.2329*. 2014.
[18] Y. Yamashita, and J. Tani. "Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment." *PLoS computational biology*. 2008.
[19] A. Graves, ”*Supervised sequence labelling with recurrent neural networks*”. Vol. 385. Springer, 2012.

Acknowledgement

Sang-woo Lee

- A LOT of discussion

Geonmin Kim (KAIST)

- Discussion on LSTM

Byoung-Tak Zhang

Thank You!