



BIOLOGICALLY INSPIRED MEMORY ENHANCEMENT OF DEEP RECURRENT NEURAL NETWORKS

T. Anderson Keller, Sharath Nittur Sridhar & Xin Wang

Artificial Intelligence Products Group, Intel Corporation

2018 NICE Workshop

FAST WEIGHT LONG SHORT-TERM MEMORY

T. Anderson Keller, Sharath Nittur Sridhar, Xin Wang

Intel AI Lab, Artificial Intelligence Products Group, Intel Corporation

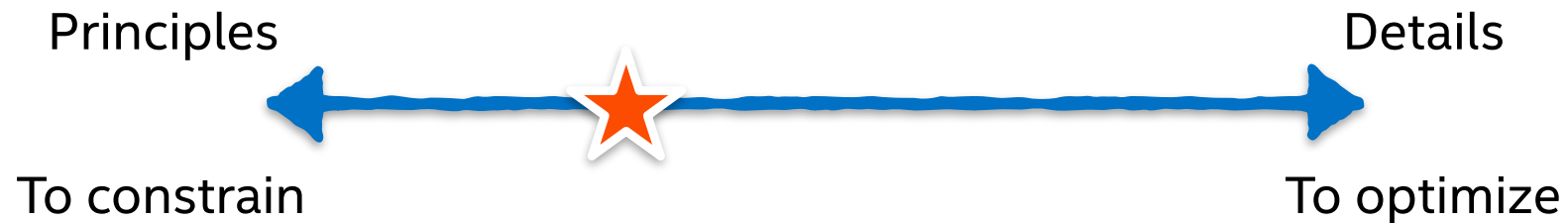
{andy.a.keller, sharath.nittur.sridhar, xin3.wang}@intel.com

ABSTRACT

Associative memory using fast weights is a short-term memory mechanism that substantially improves the memory capacity and time scale of recurrent neural networks (RNNs). As recent studies introduced fast weights only to regular RNNs, it is unknown whether fast weight memory is beneficial to gated RNNs. In this work, we report a significant synergy between long short-term memory (LSTM) networks and fast weight associative memories. We show that this combination, in learning associative retrieval tasks, results in much faster training and lower test error, a performance boost most prominent at high memory task difficulties.

Short paper under review, full work in progress

What should the role of biological inspiration be?



- Algorithm researchers to take the responsibility of guiding HW/SW tool design.
- Tools to support flexible algorithm research needs. 👍 Loihi!

Is there a philosophical rift between neuromorphic engineering and deep learning (or ML in general)?

- Neuromorphic algorithm researchers to design based on first principle. 🙌
- Deep learning researchers to be inspired by biological designs.

Deep learning: empirical laws in the realm of magic

First Law: For any network structure, optimizer or application, there exist at least one paper on the subject.

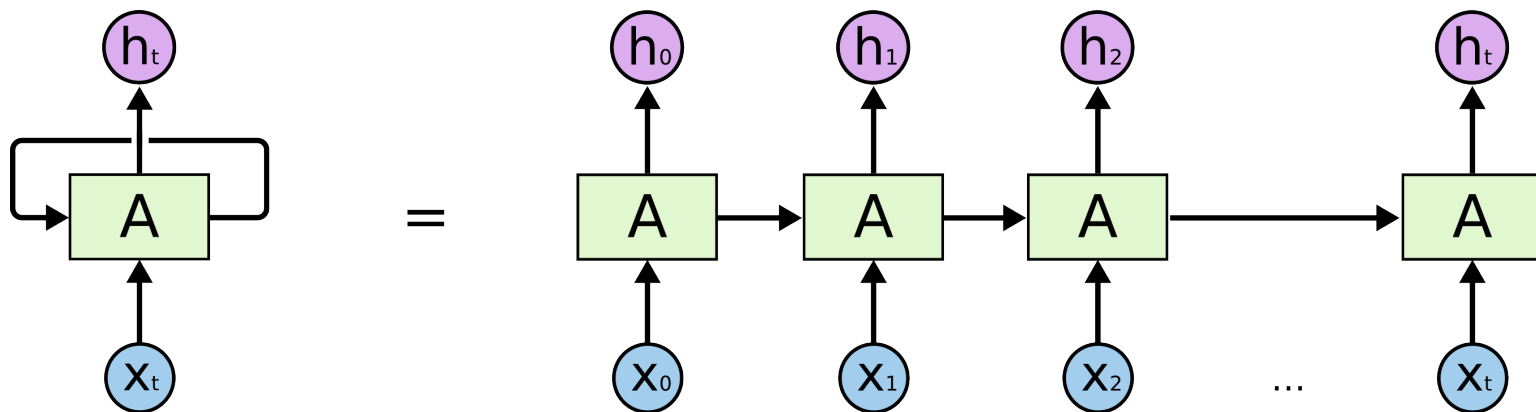
Second Law: For any pair-wise combination of network structure, optimizer and application, there exist at least one paper on the subject.

Third Law: Any exception of the previous two laws is an (almost) guaranteed paper.

Remark 1: Combinatorial alchemy is fruitful (caveat: for practice, not theory).

Remark 2: Why it is often unreasonably effective should be a subject of future theoretic investigations.

Recurrent neural nets (RNNs)



Trained by back propagation through time (BPTT)

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

“Memory” in RNNs

Hidden states

- Fast
- Capacity scales linearly with number of hidden units

Recurrent weights

- Slow
- Capacity scales quadratically with number of hidden units

Limitations of vanilla RNNs

- The memory capacity problem
 - Short-term memory maintained by activations scales linearly with number of hidden units
- The memory time scale problem
 - Difficult to support memory at long and/or diverse time scales
- The training problem
 - Vanishing/exploding gradients

Biological inspiration: multiple time scales

Fast



Slow

Neural activities

- “Activations”

A myriad of cellular and circuit mechanisms

- Cellular mechanisms, e.g. calcium dynamics, intracellular signaling, ...
- Circuit mechanisms, e.g. temporally diverse connections, delays, oscillations, ...
- “???”

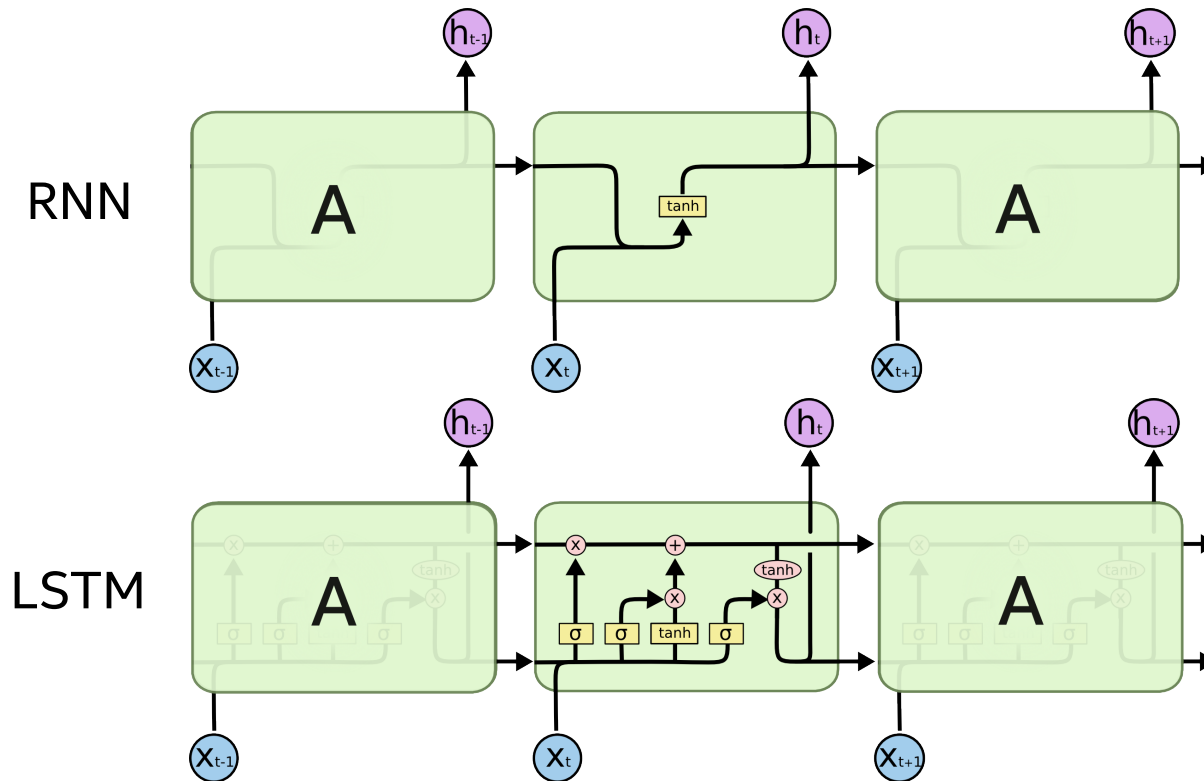
Synaptic weights

- “Recurrent weights”

Two approaches toward overcoming RNNs' limitations

- “Circuit”: clever design of recurrent network topologies → e.g. gated recurrent memory cells
- “Cellular”: enhancement with differentiable memory mechanisms → e.g. fast weights, NTM, ...

Gated RNNs: e.g. long short-term memory (LSTM)



Note: today there are ever more complex recurrent memory cells: multi-scale hierarchical, nested, skip connections, ...

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

The origin of fast weight

*In Proceedings of the Ninth Annual Conference of the
Cognitive Science Society. Seattle, WA.*

Using Fast Weights to Deblur Old Memories

Geoffrey E. Hinton and David C. Plaut

Computer Science Department
Carnegie-Mellon University

Abstract

Connectionist models usually have a single weight on each connection. Some interesting new properties emerge if each connection has two weights: A slowly changing, plastic weight which stores long-term knowledge and a fast-changing, elastic weight which stores temporary knowledge and spontaneously decays towards zero. If a network learns a set of associations and then these associations are "blurred" by subsequent learning, *all* the original associations can be "deblurred" by rehearsing on just a few of them. The rehearsal allows the fast weights to take on values that temporarily cancel out the changes in the slow weights caused by the subsequent learning.

Hinton & Plaut 1987

The origin of fast weight

Communicated by Fernando Pineda

Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks

Jürgen Schmidhuber*

Institut für Informatik, Technische Universität München,
Arcisstr. 21, 8000 München 2, Germany

Previous algorithms for supervised sequence learning are based on dynamic recurrent networks. This paper describes an alternative of gradient-based systems consisting of two feedforward nets that deal with temporal sequences using fast weights: The first net produces context-dependent weight changes for the second net, which weights may vary very quickly. The method offers the potential for high STM storage efficiency: A single weight (instead of a full-fledged recurrent net) may be sufficient for storing temporal information. Various learning methods are derived. Two experiments with unknown time delays illustrate the approach. One experiment shows how the system can be used for adaptive temporary variable binding.

REDUCING THE RATIO BETWEEN LEARNING COMPLEXITY AND NUMBER OF TIME VARYING VARIABLES IN FULLY RECURRENT NETS

In Proceedings of the International Conference on Artificial Neural Networks ICANN'93, Amsterdam, pages 460-463. Springer, 1993.

J. Schmidhuber
Institut für Informatik
Technische Universität München
Arcisstr. 21, 8000 München 40, Germany

ABSTRACT. Let m be the number of time-varying variables for storing temporal events in a fully recurrent sequence processing network. Let R_{time} be the ratio between the number of operations per time step (for an exact gradient based supervised sequence learning algorithm), and m . Let R_{space} be the ratio between the maximum number of storage cells necessary for learning arbitrary sequences, and m . With conventional recurrent nets, m equals the number of units. With the popular 'real time recurrent learning algorithm' (RTRL), $R_{time} = O(m^3)$ and $R_{space} = O(m^2)$. With 'back-propagation through time' (BPTT), $R_{time} = O(m)$ (much better than with RTRL) and R_{space} is infinite (much worse than with RTRL). The contribution of this paper is a novel fully recurrent network and a corresponding exact gradient based learning algorithm with $R_{time} = O(m)$ (as good as with BPTT) and $R_{space} = O(m^2)$ (as good as with RTRL).

Schmidhuber 1992, 1993

Fast weight RNNs

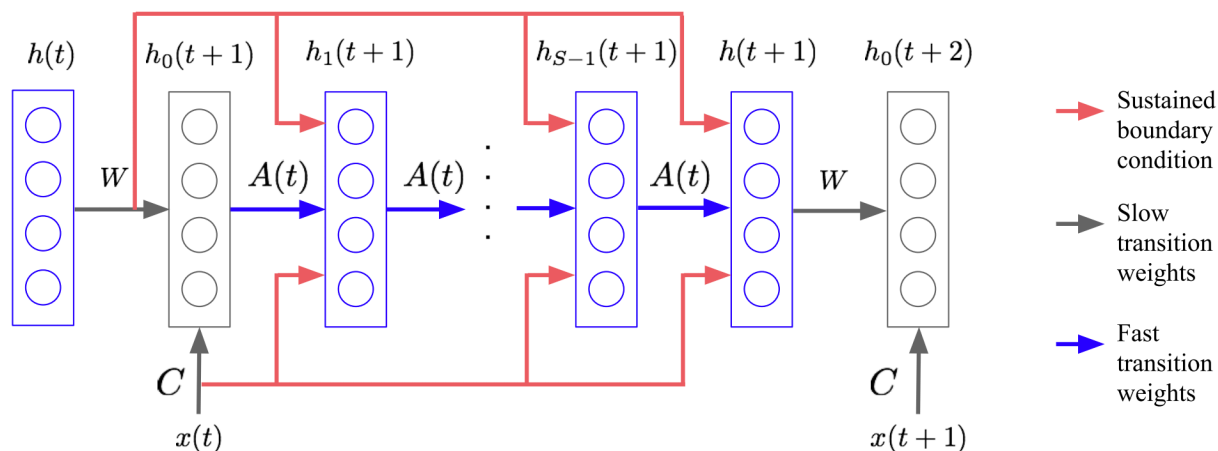


Figure 1: The fast associative memory model.

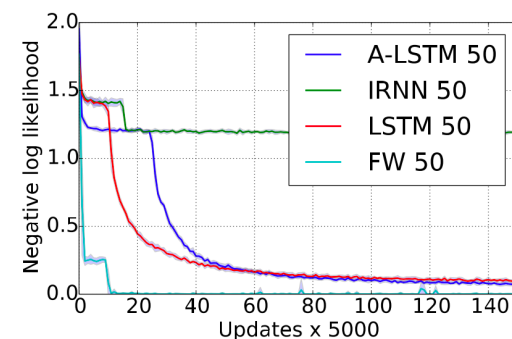


Figure 2: Comparison of the test log likelihood on the associative retrieval task with 50 recurrent hidden units.

Note: there is a rich literature on many different types of differentiable memory enhancement of DNNs before/after Ba et al. 2016: attention nets, memory nets, NTM, variations on fast weights...

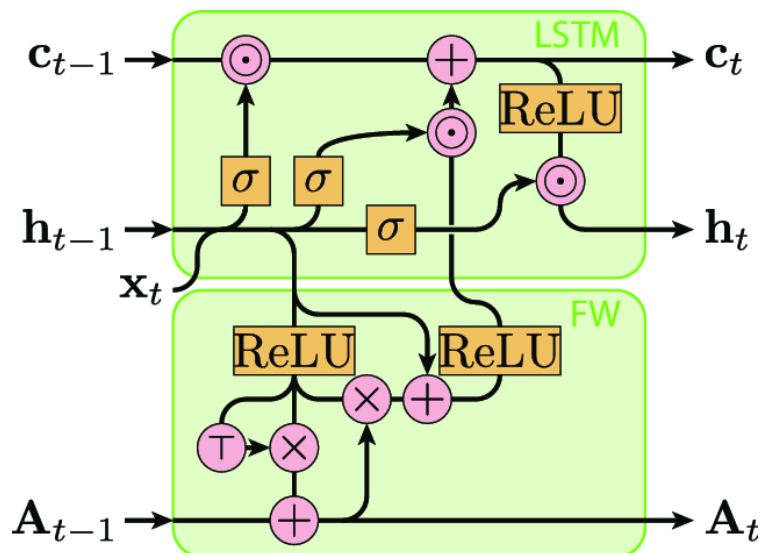
Ba et al. 2016

Question

How do **gated RNNs** such as LSTM **interact with** associative memory mechanisms like **fast weights**?

- Redundant?
- Competitive?
- Synergistic?

Fast weight LSTM



$$\begin{pmatrix} \hat{\mathbf{i}}_t \\ \hat{\mathbf{f}}_t \\ \hat{\mathbf{o}}_t \\ \hat{\mathbf{g}}_t \end{pmatrix} = \mathcal{LN} \left[\begin{pmatrix} \mathbf{W}_i & \mathbf{U}_i \\ \mathbf{W}_f & \mathbf{U}_f \\ \mathbf{W}_o & \mathbf{U}_o \\ \mathbf{W}_g & \mathbf{U}_g \end{pmatrix} \begin{pmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{pmatrix} + \begin{pmatrix} \mathbf{b}_i \\ \mathbf{b}_f \\ \mathbf{b}_o \\ \mathbf{b}_g \end{pmatrix} \right]$$

$$(\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t, \mathbf{g}_t) = (\sigma(\hat{\mathbf{i}}_t), \sigma(\hat{\mathbf{f}}_t), \sigma(\hat{\mathbf{o}}_t), \text{ReLU}(\hat{\mathbf{g}}_t))$$

$$\mathbf{A}_t = \lambda \mathbf{A}_{t-1} + \eta \mathbf{g}_t \mathbf{g}_t^\top$$

$$\mathbf{c}_t = \mathcal{LN}[\mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \text{ReLU}(\hat{\mathbf{g}}_t + \mathbf{A}_t \mathbf{g}_t)]$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \text{ReLU}(\mathbf{c}_t)$$


Major differences from fast-weight RNN (Ba et al. 2016) and regular LSTM

- No need for multiple settling iterations of fast weights
- Simultaneous layer normalization on input and hidden states
- Replaced hyperbolic tangent with rectified linear

Keller et al. 2018

Associative retrieval task (ART)

c9k8j3f1??c \mapsto 9
j0a5s5z2??a \mapsto 5 $(K = 8)$




Association Retrieval

The diagram shows two horizontal lines with vertical end caps. The top line is orange and spans from the first '5' to the first '?'. The bottom line is blue and spans from the first 'a' to the last 'c'.

Ba et al. 2016

Modified associative retrieval task (mART)

ckjf9831??c \mapsto 9
jasz0552??a \mapsto 5 $(K = 8)$



Association Retrieval

The diagram shows two horizontal lines with vertical end caps. The top line is orange and spans from the first '5' to the first '?'. The bottom line is blue and spans from the first 'a' to the last 'c'.

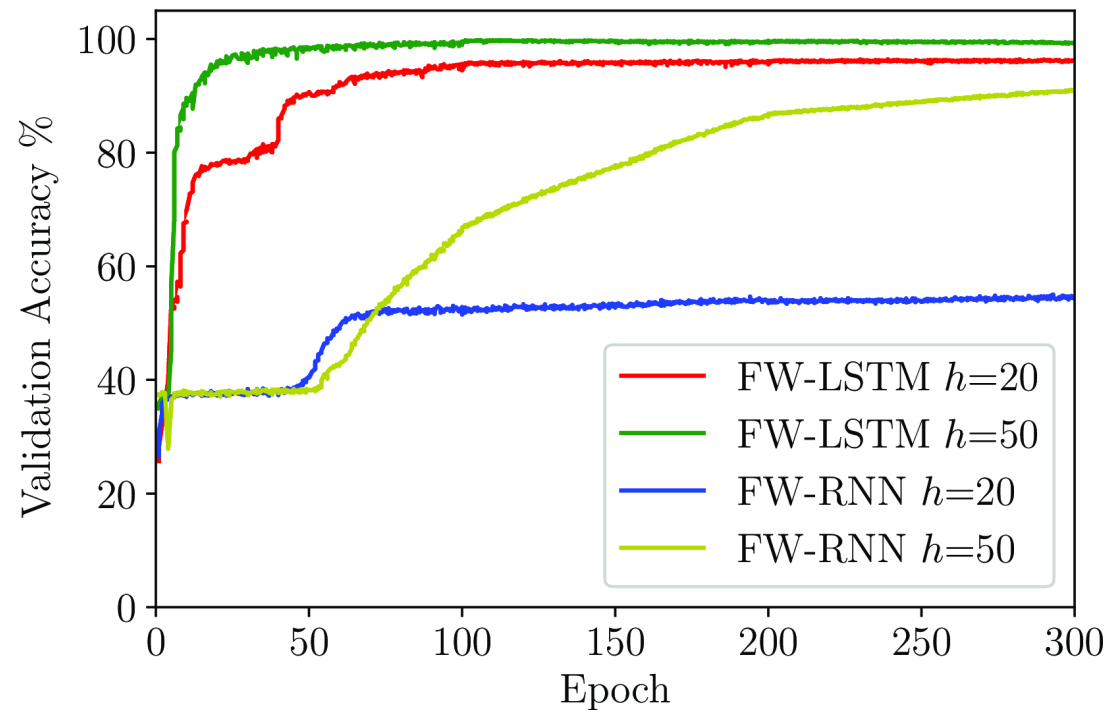
Keller et al. 2018

Results: accuracy

Task		ART		mART		# Parameters
# Hidden	Model	$K = 8$	$K = 30$	$K = 8$	$K = 16$	
$h = 20$	LN-LSTM	37.8	22.7	38.2	29.5	19k
	FW-RNN	98.7	95.7	55.5	30.3	12k
	FW-LSTM	99.6	97.5	96.3	38.9	19k
$h = 50$	LN-LSTM	95.4	21.0	34.8	25.7	43k
	FW-RNN	100.0	100.0	90.9	29.0	20k
	FW-LSTM	100.0	100.0	99.4	93.3	43k
$h = 100$	LN-LSTM	97.6	18.4	33.4	22.5	100k
	FW-RNN	100.0	100.0	91.9	30.5	38k
	FW-LSTM	100.0	100.0	99.9	92.6	100k

Keller et al. 2018

Results: speed of learning



Keller et al. 2018

Summary

We demonstrate **a strong synergy between fast weight associative memory and gated recurrent nets.**

- LSTM with fast weight associative memory trains much faster and achieves lower test error in associative retrieval tasks.
- Fast weight LSTM remains highly performant at high task memory difficulties where both LSTM and fast-weight-enhanced regular RNN utterly fail.
- This is true even for fast weight LSTMs with fewer parameters than competing models.

