Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Current Opinion in Neurobiology

Backpropagation through time and the brain

Timothy P Lillicrap[1,2,3] and Adam Santoro[1,3]

It has long been speculated that the backpropagation-of-error algorithm (backprop) may be a model of how the brain learns. Backpropagation-through-time (BPTT) is the canonical temporal-analogue to backprop used to assign credit in recurrent neural networks in machine learning, but there's even less conviction about whether BPTT has anything to do with the brain. Even in machine learning the use of BPTT in classic neural network architectures has proven insufficient for some challenging temporal credit assignment (TCA) problems that we know the brain is capable of solving. Nonetheless, recent work in machine learning has made progress in solving difficult TCA problems by employing novel memory-based and attention-based architectures and algorithms, some of which are brain inspired. Importantly, these recent machine learning methods have been developed in the context of, and with reference to BPTT, and thus serve to strengthen BPTT's position as a useful normative guide for thinking about temporal credit assignment in artificial and biological systems alike.

Addresses
[1] DeepMind, London, UK
[2] UCL, UK

Corresponding author: Santoro, Adam (adamsantoro@google.com)
[3] Equal contributions.

Current Opinion in Neurobiology 2019, **55**:82–89

This review comes from a themed issue on **Machine learning, big data, and neuroscience**

Edited by **Jonathan Pillow** and **Maneesh Sahani**

For a complete overview see the Issue and the Editorial

Available online 7th March 2019

https://doi.org/10.1016/j.conb.2019.01.011

Introduction

Synaptic physiology helps to explain the rules and processes underlying individual synaptic changes, but it does not explain how these changes coordinate to achieve a network's goal [1–3]. The backpropagation (backprop) algorithm was introduced as a solution to this coordination problem in artificial deep neural networks [4–7]. Backprop efficiently computes the effect of slight changes to each synapse on a network's deviation from its goal (also known as its *error*), taking into account the effects of these changes on all downstream neurons. It then uses the results of the computations to make small synaptic modifications to reduce the network's error. Thus, backprop solves the *credit assignment problem* by determining the role of each synapse in contributing to the network's overall performance. Backprop has a temporal analogue known as backpropagation-through-time (BPTT), which solves the *temporal* credit assignment (TCA) problem in recurrent neural networks (RNNs) [8,4,9,10].

Backprop and BPTT's enormous success in artificial neural networks has led many to consider their potential role in explaining learning in the brain [11,12,4]. While the precise connections between backprop and the brain remain unclear, recent results in neuroscience and machine learning (ML) have renewed researchers' enthusiasm for using it to help explain learning in biological networks [13–16]. The role of BPTT in explaining learning through time in the brain, however, has particular problems not faced by feedforward backprop, and its relationship to the brain is less well studied in general.

Nonetheless, BPTT-based approaches have solved an expanding set of difficult problems that require sophisticated temporal credit assignment. BPTT underlies everything from text-to-speech [17], translation [18], and learning to solve control problems that demand memory [19••,20]. Successful approaches in these temporal domains are sometimes inspired by biological considerations and simultaneously hint at formal ways to understand temporal credit assignment in the brain.

Recurrent neural networks and backpropagation through time

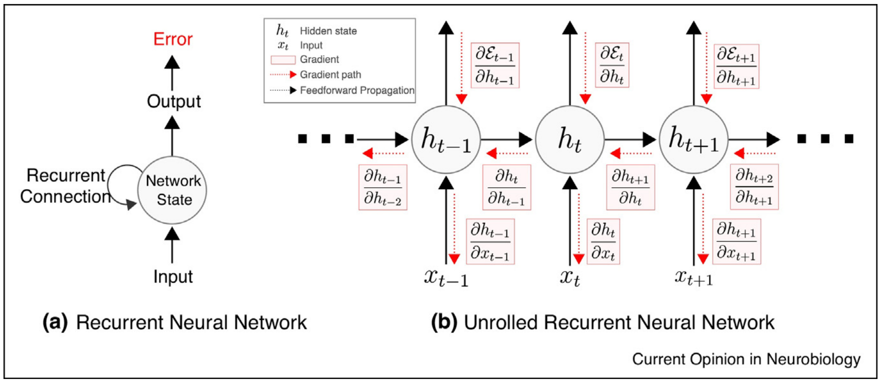
An RNN's self-connections cause neuron activities (the RNN's *state*) to reverberate as time passes. In machine learning we typically discretize the state changes, computing them using the activities at the previous discrete time-step and the synaptic weights (this is known as 'unrolling' the network — see Figure 1):

$$h_t = f(x_t, h_{t-1}; \theta). \tag{1}$$

Here h denotes a vector of activity states (indexed by time in the subscript, where T is the final time-point in a sequence), x is the network input, and θ are the synaptic weights, also known as the learnable parameters.[4]

[4] We use θ for convenience to include the set of recurrent weights, $\boldsymbol{W_r}$, the set of input weights $\boldsymbol{W_i}$, the set of output weights $\boldsymbol{W_o}$, and all other learnable parameters such as biases.

Figure 1


Error
Output
Recurrent Connection
Network State
Input
h_t Hidden state
x_t Input
Gradient
Gradient path
Feedforward Propagation
h_{t-1}
h_t
h_{t+1}
$\frac{\partial \mathcal{E}_{t-1}}{\partial h_{t-1}}$
$\frac{\partial \mathcal{E}_t}{\partial h_t}$
$\frac{\partial \mathcal{E}_{t+1}}{\partial h_{t+1}}$
$\frac{\partial h_{t-1}}{\partial h_{t-2}}$
$\frac{\partial h_t}{\partial h_{t-1}}$
$\frac{\partial h_{t+1}}{\partial h_t}$
$\frac{\partial h_{t+2}}{\partial h_{t+1}}$
$\frac{\partial h_{t-1}}{\partial x_{t-1}}$
$\frac{\partial h_t}{\partial x_t}$
$\frac{\partial h_{t+1}}{\partial x_{t+1}}$
x_{t-1}
x_t
x_{t+1}
Current Opinion in Neurobiology

(a) Recurrent Neural Network

(b) Unrolled Recurrent Neural Network

Recurrent neural networks and BPTT. **(a)** depicts a simple recurrent network. Self-connections, or recurrent-connections cause input activity to feed back into the network, producing evolving activity dynamics. The network's output can be compared to a target output to compute the error (here shown at every timestep). **(b)** depicts an unrolled network and the gradients that arise from computing the effect of various network components on the output error. These gradients flow backwards through the network and help inform synaptic updates that reduce the network's error. An important focus of BPTT involves the hidden state gradients, $\frac{\partial h_{t+1}}{\partial h_t}$, which multiplicatively compound to produce vanishing or exploding signals.

The goal of BPTT is to compute the partial derivatives of the error with respect to the synaptic weights, known as the 'gradients', $\frac{\partial E}{\partial \theta}$. The network improves its performance by learning through 'gradient descent'; nudging the synaptic weights in the negative direction of the gradient reduces the network's error. We will not derive BPTT, but rather, will highlight a specific aspect that has drawn the focus of much the research into training RNNs. When using the chain rule to compute the gradients of the recurrent parameters for a standard RNN (e.g. $h_t = \sigma(\boldsymbol{W}_r h_{t-1} + \boldsymbol{W}_i x_t)$), we have an intermediate term $\frac{\partial E}{\partial h_t}$ that computes the gradient of the error with respect to the activity states:

$$\frac{\partial E}{\partial h_t} = \frac{\partial E}{\partial h_T}\frac{\partial h_T}{\partial h_t} = \frac{\partial E}{\partial h_T}\prod_{k=t}^{T-1}\frac{\partial h_{k+1}}{\partial h_k} \tag{2}$$

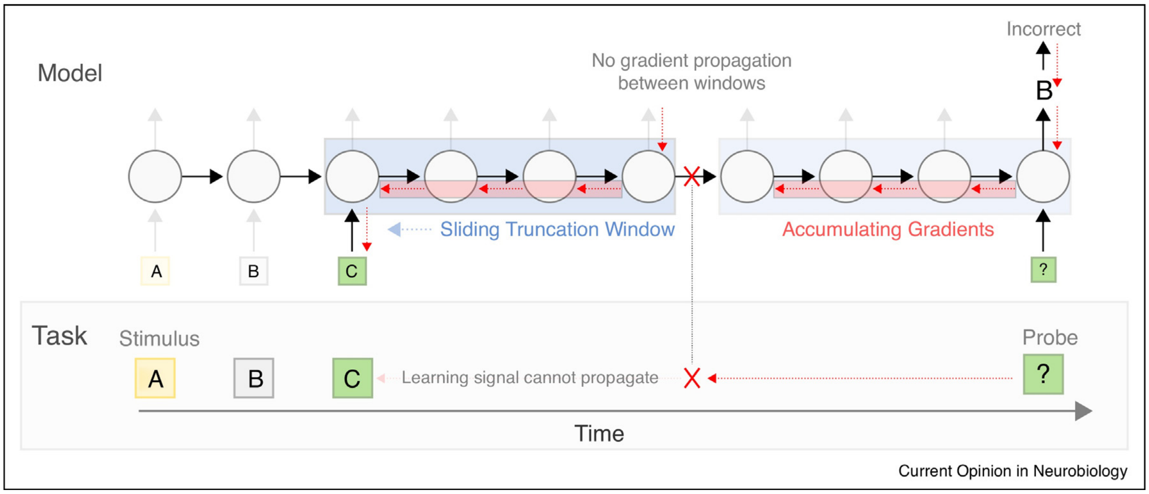
$$= \frac{\partial E}{\partial h_T}\prod_{k=t}^{T-1}\mathrm{diag}(\sigma'(h_{t+1}))\boldsymbol{W}_r^T, \tag{3}$$

The important part to note is in Eqn. (3), where we have an iterated product of matrices, including the matrix denoting the network's recurrent synaptic weights $\boldsymbol{W}_r$. As noted in [21], similar to how iterated products of real numbers can *explode* to infinity or *vanish* to zero, iterated products of matrices can explode or vanish along some vector direction (in particular, the directions corresponding to the eigenvectors with the leading eigenvalues of the recurrent weight matrix). Thus, the learning signal for the network (the gradient) either becomes less and less useful over time due to vanishing, or causes massive instability due to explosion. In the case of vanishing, while the gradient with respect to the recurrent weights may be non-zero, the gradient of one recurrent state with respect to another decreases exponentially with the number of intervening timesteps, implying exponentially more training time to assign credit to events that occurred long in the past. While RNNs are powerful models, exploding and vanishing gradients can make them difficult to optimize for useful memory formation and retrieval, which in turn can prevent them from being practically useful. A solution to this 'accumulating gradient' problem is critical for developing models that can assign credit across minutes, hours, or even days, which we know is possible in humans and animals.

Thus, fundamental research on learning in recurrent networks has focused on taming exploding or vanishing gradients, whose effects are especially problematic over long time scales [22]. Two categories of approaches are those based on network design, and optimization.

Among network-based approaches, echo state networks have fixed recurrent weights that do not undergo learning, and hence avoid this problem at the expense of network expressivity [23]. Unitary RNNs constrain the recurrent weight matrices to be unitary, wherein the eigenvalues are equal to 1, so that the matrix products do not cause explosion or vanishing [24]. Initialization schemes that, for example, set the initial weight matrix to be orthogonal produce better behaved gradients, at least at the beginning of training [25,26]. Long short-term memory networks (LSTMs) and Gated Recurrent Units generally [27–30] use 'gates' — special neurons that control the flow of information into recurrent 'memory' neurons. These memory units can hold information for long periods,

Figure 2


Model
Incorrect
No gradient propagation between windows
B
Sliding Truncation Window
Accumulating Gradients
A
B
C
?
Task
Stimulus
Probe
A
B
C
Learning signal cannot propagate
?
Time

Current Opinion in Neurobiology

Truncated backpropagation through time (TBPTT). In TBPTT gradients are computed within a small window of time so as to reduce memory requirements and limit the explosion or vanishing effects of accumulating hidden state gradients. A consequence of TBPTT is that gradient propagation is halted between truncation windows. Consider a task wherein some probe input needs to be pattern-completed at some later point in time such that it matches a previous input (here it is the green square containing the 'C'). If the network produces the incorrect letter (depicted as 'B' here), then there is no way for the gradients to reach back to the point of encoding of the green-square-C input, and hence no way for the network to learn to better encode and store this information for later use.

allow the gradient of the error with respect to the held value to remain independent of computations in intervening timesteps, and may have analogs in cortical microciruitry [31].

Among the optimization-based approaches, gradient clipping upper bounds the gradients by clipping them to some threshold value [32,21]. Among the most commonly employed mechanisms, Truncated BPTT (TBPTT) (Figure 2) implements BPTT on small time windows so that the number of products in Eqn. (3) is kept small, and hence is less prone to exploding or vanishing gradients, at the expense of not being able to assign credit to events outside the truncation window [9,33,34]. One consequence of TBPTT is an inability to assign credit outside of the truncation window, even in situations where the gradients would not vanish.

Although these approaches have proved fruitful in machine learning, neuroscientists may not be satisfied. Perhaps the biggest problem is that the network must still store and retrieve, with perfect accuracy, the values of its activities from all points in past. In a feedforward network learning with backprop this storage problem is slightly more plausible, since the activity states live in unique sets of neurons. However, for RNNs learning with BPTT the *same* neurons must store and retrieve their entire activation history. TBPTT ameliorates this by reducing the size of the history, but even still, practically the history must be many tens of steps.

In the brain there does seem to be evidence that sequential bouts of activity are stored and replayed (both forwards and backwards) in the medial temporal lobe, and in some cases the cortex [35,36,37,38,39,35–39,65•]. These replay events are often proposed to be useful for systems consolidation — a memory encoding process wherein the medial temporal lobe transforms and transfers stored memories to the cortex [40,41] — and for planning and reinforcement learning more generally. It's not clear whether the kind of memory replay events observed empirically are sufficient to support BPTT[5], though compressed, backwards replay of sequences might suggest that BPTT-like TCA is not unreasonable.

A less known class of TCA methods uses *forward-mode differentiation* (as opposed to the backwards-mode differentiation used in BPTT) to forego the storage of hidden states entirely [42]. As networks run forward in time the sensitivity of their state on the parameters, $\partial h_t/\partial \theta$, is computed and maintained *online*, often with synaptic weight updates being applied at each time step in which there is a non-zero error. The canonical forward-mode algorithm is called real-time recurrent learning (RTRL). While RTRL solves one problem by obviating the need for hidden state storage and replay, it introduces another: the propagated sensitivities can be extremely large: a

[5] Since the inputs aren't preserved, there is no way to train the network's input synaptic weights. Also, the retrieved activities must be noiseless, which is not the case in the brain. Even further, the number of replayed events is much fewer than those expected by even TBPTT

network with N recurrent units requires $O(N^3)$ storage and $O(N^4)$ computation at each time-step to maintain accurate sensitivities, which appears unfeasible. A number of other algorithms, such as Unbiased Online Recurrent Optimization (UORO) [43,44•,45], have since built off RTRL's scaffolding, using approximate but unbiased gradient estimates to reduce computation and storage requirements. The ideas in these algorithms are connected to the notion of learning via eligibility trace maintenance [46], and may help indeed inform our views of how the brain does TCA, at least at shorter time-scales.

Altogether, some combination of the aforementioned methods — TBPTT, gated RNNs, etc. — can induce successful learning over short time-scales in artificial networks. However, they struggle with long time-scale learning because of scaling issues, or inevitably problematic gradients. While online methods such as RTRL and its approximations are interesting, they also succumb to issues related to accumulating gradients and typically require *ad hoc* approaches which forget sensitivities from far in the past in order to work. A new wave of research addressing learning over *long* time-scales in artificial networks has turned to brain-inspired mechanisms, such as content addressable memory and attention, for inspiration. In turn, this research casts new light on these mechanisms in the brain, and their potential role in implementing TCA.

Long-term temporal credit assignment using memory and attention

To assign credit to states from long in the past, networks need a mechanism to propagate gradient information from the present with high-fidelity. Gated RNNs, such as LSTMs and GRUs, proposed 'memory cells', which are hidden states that can remain unchanged for long periods of time, and hence render the gradient of the error with respect to their value to be independent of the computations in intervening timesteps. However, Gated RNNs are tasked with both storing information in their memory cells and using this stored information to compute relevant information for the current output. In practice, this dual purposing of the hidden state (*computation* vs. *storage*) can greatly hinder learning over long time periods.

Attention-based models offload the storage problem by supposing that the hidden states are stored somewhere outside the network, but are nonetheless readily accessible at any point in time [47••,48••]. These networks then use current network activity to attend to one or more of these stored previous hidden states, and the attended states are used to update the current state. More importantly, from a learning perspective this mechanism establishes a 'skip-connection' from the current state to the attended state; gradients can propagate along this skip-connection instead of through every intervening hidden state (and the intervening non-linearities), and hence can bypass any problems related to accumulating gradients. In straightforward applications of this method the skip-connected gradients are considered in addition to the intervening accumulated gradients; however, some recent work has explored avoiding these accumulated gradients altogether [49].

Attention-based models are of course easy to implement in a computer, wherein hidden states are simply stored in the computer's memory. Taking inspiration from the brain's memory systems, recent models propose *augmenting* Gated RNNs with their own large external memory storage that can be read from and written to [50,51••,52], thus, treating the external memory as a component of the network itself. Rather than detail a particular model, we introduce the essential concepts that are common across many models that employ external memory. In a simple version of such an idea the external memory encodes the RNN's hidden state, growing linearly with the number of time-steps, and hence with the number of states realized by the network:

$$M_t = \{M_{t-1}, h_t\} \tag{4}$$

where curly braces denote an append operation. The memory matrix is then included in the RNN state update: $h_t = f(x_t, h_{t-1}, M_{t-1}; \theta)$. See Figure 3, which demonstrates how this helps propagate gradient information back through time. This kind of simple writing mechanism demands more memory and compute as time goes on [53]. Other augmented memory architectures use a fixed number of slots that are *updated* rather than appended to [50,51••,54], or are explicitly designed for distributed and compressive writing [55].

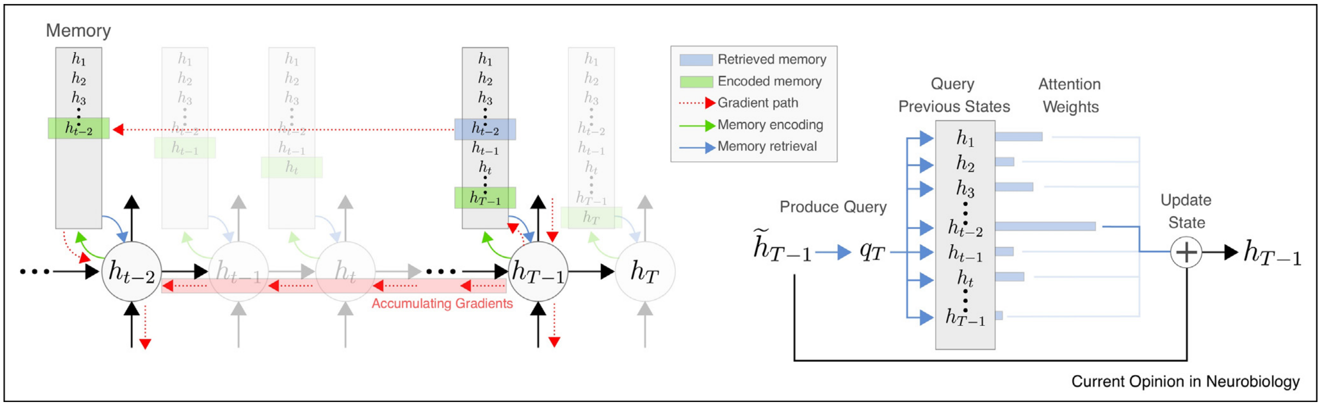
Once we augment a network with a large external memory it needs to be able to query the memory to retrieve relevant data. Given the simple writing approach described above, a retrieval mechanism can work as follows. Suppose the current hidden state produces a query vector q_t, for example, via a linear transformation of the current hidden state, $q_t = W_q h_t$. One can then compute the similarity of this query to each memory using a distance metric to produce 'attention weights', which are then used to 'read' memories,

$$\text{Compute weights}: w_t[i] = d(q_t, M_{t-1}[i,:])$$

$$\text{Read memories}: r_t = \sum_i w_t[i] M_{t-1}[i,:], \tag{5}$$

which update the current hidden state and the output of the network at that timestep. Note that in Figure 3 we picture the retrieval of memories as a discrete choice, but

Figure 3



Current Opinion in Neurobiology

Gradient skip-connections through memory. RNNs augmented with external memories allow for high fidelity recall and gradient propagation. Suppose a memory at h_{t-2} is encoded and later used at h_{T-1}. Since the content of the memory is unchanged during this time, the gradient with respect to this memory is not dependent on any computations that occur between memory writing and memory retrieval. This high-fidelity gradient can then be used to inform learning at the time of encoding. Conversely, if the gradient needed to pass through the intermediary hidden states, then it would suffer the consequences of accumulating gradients. Depicted on the right is a potential mechanism for using attention to read from memory: (1) queries attend to each memory using a distance function $d(q_T, h_t)$; (2) the distance is used as an attention weight to compute a weighted sum of memories; (3) the weighted sum of memories update the original hidden state.

in practice it is a ‘soft’ blend to allow gradients to flow nicely.

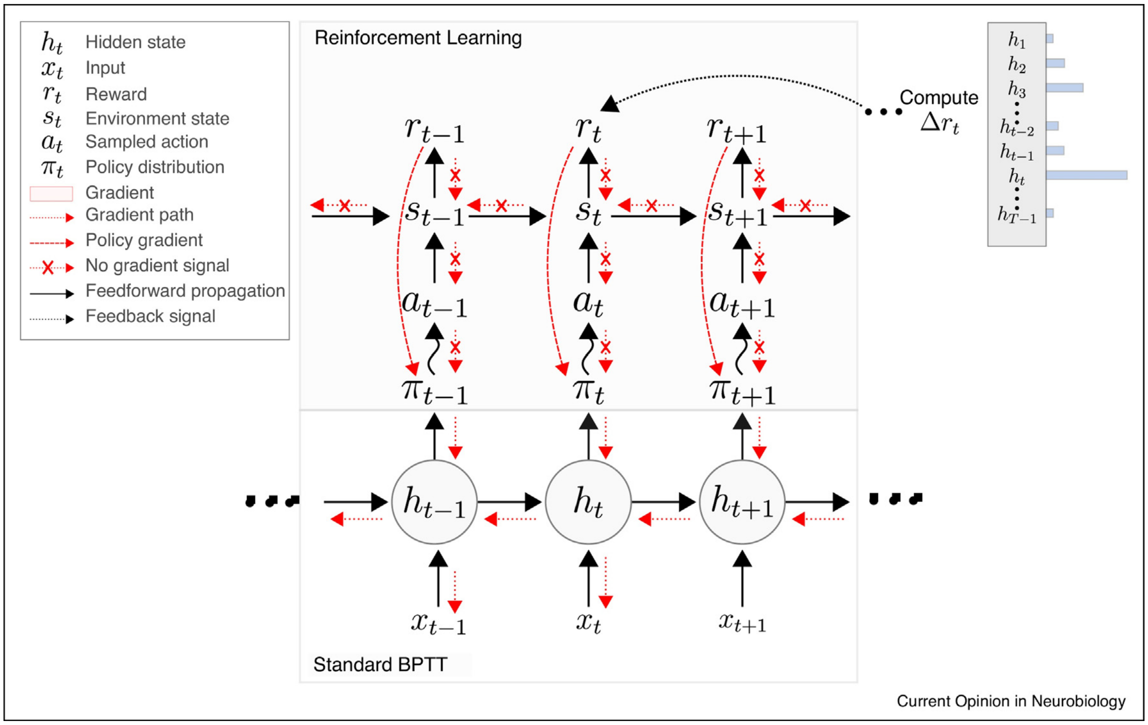
A non-parametric memory that simply appends new hidden state activations as they arrive has a constant number of learnable synaptic weights. The size of the memory only factors into the attention-based ‘reading’, which is a parameter-free computation since the computation of the query q_t takes place before reading and requires the same number of parameters irrespective of the number of memories. In Gated RNNs, on the other hand, the number of learnable parameters scales with the size of memory. Offloading storage requirements to an external memory has benefits beyond the separation of computation and storage: if the network can guarantee that stored information remains untouched, then gradients passing from the time of memory retrieval back to the time of memory encoding will be of high quality, and will not have succumbed to any accumulation issues. Just how large memories and temporal credit assignment mechanisms should interact is an area of active research. As memory size grows full BPTT becomes increasingly problematic [53]. TBPTT is still straightforward, but in this case credit assignment will not reach those memories that were encoded and written in the distant past. In this case, some approaches learn to encode input data using local-in-time credit assignment and trust that it will be useful to retrieve and use this data in the future [56,53,20]. Unsupervised objectives may also help shape encoded memories when task-oriented feedback is only infrequently available [20] (e.g. outside a models truncation window).

The notion of using external memory storage to encode memories for later retrieval is not unfathomable from a biological perspective. In the medial temporal lobe the hippocampus is thought to encode memories by establishing attractor states in the CA3 (akin to the aforementioned ‘writing’) [57,51••]. In turn, partial reactivation (akin to the aforementioned ‘querying’) of these CA3 attractors induces re-activation of the neocortical neurons responsible for the initial encoding [58]. Thus, one can imagine a situation wherein gradient information in neocortical circuits is ‘transported’ back to the neocortical states at the time of memory encoding using a process mediated by hippocampal-based memory retrieval.

BPTT, temporal credit assignment, and reinforcement learning

Reinforcement learning (RL) scenarios — wherein an agent interacts with an environment and learns better behavioral policies by correlating actions with environment rewards — have different TCA considerations compared to many non-RL regimes. In particular, the model-environment ‘loop’ in RL typically has a number of non-differentiable components that usually do not permit gradient signal propagation, such as the environment itself and the stochastic process of choosing an action from a policy distribution (though see [59,60] for differentiable stochastic categorical sampling, and [61,62] for control approaches using differentiable environment models) (see Figure 4). Since there is only a ‘gradient path’ through a model’s hidden states and not through its sampled actions and the environment, an agent may have difficulty computing the effect of its actions on the environment, and hence with assigning credit to actions

Figure 4



BPTT and reinforcement learning. Reinforcement learning (RL) frameworks can be built on top of standard BPTT frameworks. Importantly, many RL components are non-differentiable (e.g. one cannot pass gradients through the environment or action sampling), making it impossible for a model to *compute* the effect of its actions on future rewards using BPTT. Instead, RL algorithms *correlate* actions with future rewards, which is a noisy process with high variance when actions and consequences are separated by long delays. Some new algorithms, such as Temporal Value Transport (TVT) [63•], can ameliorate these difficulties by delivering value information along temporal skip-connections (using, for example, the strength of memory read events to determine the connections); this value information is then used to increase the signal-to-noise ratio of normal BPTT-based policy-gradient learning.

taken far in the past even if they are directly responsible for eventual reward.

RL algorithms often make use of Monte-Carlo rollouts and bootstrapping to overcome this difficulty. For example, past states and actions can be assigned value because at some future point reward will be received. This value can then be used to inform certain actions over others. However, this process is based on *correlations* between current states and actions and future rewards, as opposed to the explicit *computations* afforded should the process be amenable to full BPTT. These correlations can have high variance, especially if the future horizon is long and filled with unrelated intermediary reward signals, making this approach unfeasible in certain circumstances.

New methods have also begun to explore the use of attention-based and memory-based systems to overcome downsides of RL's correlative algorithms. The fundamental idea underlying them is the 'transport' of reward signals to within a close temporal window of *relevant* past states so as to allow BPTT to more accurately compute the credit to assign to the actions chosen in these states [63•,64]. Memory systems can be used to form links between the present and the distant past, similar to the mechanism described in the previous section and in Figure 3. However, instead of forming a skip-connection along which gradients can propagate, these memory links are used to signal avenues along which *reward* can be transported so as to make local-in-time credit assignment more effective.

Conclusion

By tackling increasingly difficult problems, practical innovations in ML have used biology-inspired solutions to TCA. These solutions may in turn help guide our understanding of credit assignment in the brain. Ultimately we expect that agents and animals alike will not adhere to strict formulations of BPTT. This does not imply that BPTT should not remain a canonical guide to TCA; even when full differentiation through time isn't possible, innovation should be guided towards its approximation, and progress should be gauged with the bar set by its hypothetical possibility.

Conflict of interest statement

Nothing declared.

Acknowledgements

We thank Greg Wayne, Daan Wierstra, Alden Hung, Josh Abramson, Blake Richards, Yoshua Bengio and Geoffrey Hinton for discussions that shaped our thinking on the subject. As well, we thank David Barrett for comments on the manuscript.

References and recommended reading

Papers of particular interest, published within the period of review, have been highlighted as:

- • of special interest
- •• of outstanding interest

1. Hebb DO: *The Organization of Behavior: A Neuropsychological Theory*. Psychology Press; 2005.
2. Markram H, Lübke J, Frotscher M, Sakmann B: **Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs**. *Science* 1997, **275**:213-215.
3. Bliss TVP, Lømo T: **Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path**. *J Physiol* 1973, **232**:331-356.
4. Rumelhart DE, Hinton GE, Williams RJ: *Learning Internal Representations by Error Propagation. Technical Report*. California Univ San Diego La Jolla Inst for Cognitive Science; 1985.
5. Werbos P: *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences (Ph.D. dissertation)*. Harvard University; 1974.
6. Parker DB: *Learning Logic*. 1985.
7. LeCun Y: **Une procedure d'apprentissage ponr reseau a seuil asymetrique**. *Proc Cognit* 1985, **85**:599-604.
8. Werbos PJ: **Generalization of backpropagation with application to a recurrent gas market model**. *Neural Netw* 1988, **1**:339-356.
9. Elman JL: **Finding structure in time**. *Cognit Sci* 1990, **14**:179-211.
10. Werbos PJ: **Backpropagation through time: what it does and how to do it**. *Proc IEEE* 1990, **78**:1550-1560.
11. Grossberg S: **Competitive learning: from interactive activation to adaptive resonance**. *Cognit Sci* 1987, **11**:23-63.
12. Crick F: **The recent excitement about neural networks**. *Nature* 1989, **337**:129-132.
13. Lillicrap TP, Cownden D, Tweed DB, Akerman CJ: *Random feedback weights support learning in deep neural networks*. 2014. arXiv preprint. arXiv:1411.0247.
14. Cadieu CF, Hong H, Yamins DLK, Pinto N, Ardila D, Solomon EA, Majaj NJ, DiCarlo JJ: **Deep neural networks rival the representation of primate it cortex for core visual object recognition**. *PLoS Comput Biol* 2014, **10**:e1003963.
15. Yamins DLK, Hong H, Cadieu CF, Solomon EA, Seibert D, DiCarlo JJ: **Performance-optimized hierarchical models predict neural responses in higher visual cortex**. *Proc Natl Acad Sci* 2014, **111**:8619-8624.
16. Guerguiev J, Lillicrap TP, Richards BA: **Towards deep learning with segregated dendrites**. *ELife* 2017, **6**:e22901.
17. Van Den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior AW, Kavukcuoglu K: **Wavenet: a generative model for raw audio.**. *SSW* 2016:125.
18. Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K *et al.*: *Google's neural machine translation system: bridging the gap between human and machine translation*. 2016. arXiv preprint. arXiv:1609.08144.
19. •• Oh J, Chockalingam V, Singh S, Lee H: **Uses reinforcement learning in conjunction with BPTT in recurrent networks to solve challenging memory tasks.**. *Control of memory, active perception, and action in minecraft*. 2016. arXiv preprint. arXiv:1605.09128.
20. Wayne G, Hung C-C, Amos D, Mirza M, Ahuja A, Grabska-Barwinska A, Rae J, Mirowski P, Leibo JZ, Santoro A *et al.*: *Unsupervised predictive memory in a goal-directed agent*. 2018. arXiv preprint. arXiv:1803.10760.
21. Pascanu R, Mikolov T, Bengio Y: **On the difficulty of training recurrent neural networks**. *International Conference on Machine Learning* 2013:1310-1318.
22. Bengio Y, Simard P, Frasconi P: **Learning long-term dependencies with gradient descent is difficult**. *IEEE Trans Neural Netw* 1994, **5**:157-166.
23. Maass W, Natschläger T, Markram H: **Real-time computing without stable states: a new framework for neural computation based on perturbations**. *Neural Comput* 2002, **14**:2531-2560.
24. Arjovsky M, Shah A, Bengio Y: **Unitary evolution recurrent neural networks**. *International Conference on Machine Learning* 2016:1120-1128.
25. Saxe AM, McClelland JL, Ganguli S: *Exact solutions to the nonlinear dynamics of learning in deep linear neural networks*. 2013. arXiv preprint. arXiv:1312.6120.
26. Le QV, Jaitly N, Hinton GE: *A simple way to initialize recurrent networks of rectified linear units*. 2015. arXiv preprint. arXiv:1504.00941.
27. Hochreiter S, Schmidhuber J: **Long short-term memory**. *Neural Comput* 1997, **9**:1735-1780.
28. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y: *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. 2014. arXiv preprint. arXiv:1406.1078.
29. Jozefowicz R, Zaremba W, Sutskever I: **An empirical exploration of recurrent network architectures**. *International Conference on Machine Learning* 2015:2342-2350.
30. Danihelka I, Wayne G, Uria B, Kalchbrenner N, Graves A: *Associative long short-term memory*. 2016. arXiv preprint. arXiv:1602.03032.
31. Costa R, Alexandros Assael I, Shillingford B, de Freitas N, Vogels T: **Cortical microcircuits as gated-recurrent neural networks**. *Advances in Neural Information Processing Systems*. 2017:272-283.
32. Mikolov T: **Statistical Language Models based on Neural Networks**.
33. Williams RJ, Peng J: **An efficient gradient-based algorithm for on-line training of recurrent network trajectories**. *Neural Comput* 1990, **2**:490-501.
34. Gruslys A, Munos R, Danihelka I, Lanctot M, Graves A: **Memory-efficient backpropagation through time**. *Advances in Neural Information Processing Systems* 2016:4125-4133.
35. Skaggs WE, McNaughton BL, Gothard KM: **An information-theoretic approach to deciphering the hippocampal code**. *Advances in Neural Information Processing Systems* 1993:1030-1037.
36. Wilson MA, McNaughton BL: **Reactivation of hippocampal ensemble memories during sleep**. *Science* 1994, **265**:676-679.
37. Foster DJ, Wilson MA: **Reverse replay of behavioural sequences in hippocampal place cells during the awake state**. *Nature* 2006, **440**:680.
38. Ji D, Wilson MA: **Coordinated memory replay in the visual cortex and hippocampus during sleep**. *Nat Neurosci* 2007, **10**:100.

39. Ambrose RE, Pfeiffer BE, Foster DJ: **Reverse replay of hippocampal place cells is uniquely modulated by changing reward**. *Neuron* 2016, **91**:1124-1136.

40. Winocur G, Moscovitch M: **Memory transformation and systems consolidation**. *J Int Neuropsychol Soc* 2011, **17**:766-780.

41. Squire LR: **Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans**. *Psychol Rev* 1992, **99**:195.

42. Williams RJ, Zipser D: **A learning algorithm for continually running fully recurrent neural networks**. *Neural Comput* 1989, **1**:270-280.

43. Ollivier Y, Tallec C, Charpiat G: *Training recurrent networks online without backtracking*. 2015. arXiv preprint. arXiv:1507.07680.

44. • Tallec C, Ollivier Y: *Unbiased online recurrent optimization*. 2017. arXiv preprint. arXiv:1702.05043.

Introduces Unbiased Online Recurrent Optimization (UORO), an approximation to real-time recurrent learning (RTRL) which requires no BPTT.

45. Mujika A, Meier F, Steger A: *Approximating real-time recurrent learning with random Kronecker factors*. 2018. arXiv preprint. arXiv:1805.10842.

46. Seung HS: **Learning in spiking neural networks by reinforcement of stochastic synaptic transmission**. *Neuron* 2003, **40**:1063-1073.

47. •• Bahdanau D, Cho K, Bengio Y: *Neural machine translation by jointly learning to align and translate*. 2014. arXiv preprint. arXiv:1409.0473.

Introduces an attention mechanism in a recurrent neural network architecture to better solve machine translation.

48. •• Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I: **Attention is all you need**. *Advances in Neural Information Processing Systems* 2017:5998-6008.

Demonstrates the power of attention-based models for modeling temporal data, using attention-based connections to allow gradient information to flow between states that are separated by a large amount of time.

49. Rosemary Ke N, Goyal A, Bilaniuk O, Binas J, Mozer MC, Pal C, Bengio Y: *Sparse attentive backtracking: temporal credit assignment through reminding*. 2018. arXiv preprint. arXiv:1809.03702.

50. Graves A, Wayne G, Danihelka I: *Neural turing machines*. 2014. arXiv preprint. arXiv:1410.5401.

51. •• Graves A, Wayne G, Reynolds M, Harley T, Danihelka I, Grabska-Barwińska A, Gómez Colmenarejo S, Grefenstette E, Ramalho T, Agapiou J *et al.*: **Hybrid computing using a neural network with dynamic external memory**. *Nature* 2016, **538**:471.

Proposes a differentiable neural memory, demonstrating the importance of segregating storage and computation in recurrent models.

52. Sukhbaatar S, Weston J, Fergus R *et al.*: **End-to-end memory networks**. *Advances in Neural Information Processing Systems* 2015:2440-2448.

53. Rae J, Hunt JJ, Danihelka I, Harley T, Senior AW, Wayne G, Graves A, Lillicrap T: **Scaling memory-augmented neural networks with sparse reads and writes**. *Advances in Neural Information Processing Systems* 2016:3621-3629.

54. Santoro A, Bartunov S, Botvinick M, Wierstra D, Lillicrap T: **Meta-learning with memory-augmented neural networks**. *International Conference on Machine Learning* 2016:1842-1850.

55. Wu Y, Wayne G, Graves A, Lillicrap T: *The Kanerva machine: a generative distributed memory*. 2018. arXiv preprint. arXiv:1804.01756.

56. Grave E, Joulin A, Usunier N: *Improving neural language models with a continuous cache*. 2016. arXiv preprint. arXiv:1612.04426.

57. Neunuebel JP, Knierim JJ: **CA3 retrieves coherent representations from degraded input: direct evidence for CA3 pattern completion and dentate gyrus pattern separation**. *Neuron* 2014, **81**:416-427.

58. Frankland PW, Bontempi B: **The organization of recent and remote memories**. *Nat Rev Neurosci* 2005, **6**:119.

59. Jang E, Gu S, Poole B: *Categorical reparameterization with Gumbel-Softmax*. 2016. arXiv preprint. arXiv:1611.01144.

60. Maddison CJ, Mnih A, Whye Teh Y: *The concrete distribution: a continuous relaxation of discrete random variables*. 2016. arXiv preprint. arXiv:1611.00712.

61. Jordan MI, Rumelhart DE: **Forward models: supervised learning with a distal teacher**. *Cognit Sci* 1992, **16**:307-354.

62. Todorov E, Li W: **A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems**. *American Control Conference, 2005. Proceedings of the 2005. IEEE* 2005:300-306.

63. • Hung C-C, Lillicrap T, Abramson J, Wu Y, Mirza M, Carnevale F, Ahuja A, Wayne G: *Optimizing agent behavior over long time scales by transporting value*. 2018. arXiv preprint. arXiv:1810.06721.

Introduces Temporal Value Transport (TVT), an algorithm that uses specific recall of memories to credit actions from the distant past.

64. Arjona-Medina JA, Gillhofer M, Widrich M, Unterthiner T, Hochreiter S: *Rudder: return decomposition for delayed rewards*. 2018. arXiv preprint. arXiv:1806.07857.

65. • Foster DJ: **Replay comes of age**. *Annu Rev Neurosci* 2017, **40**:581-602.

Provides a comprehensive overview of hippocampal-based memory replay, including a theoretical perspective linking it to reinforcement learning.