Journal of Machine Learning Research 1 (2000) 1-48 Submitted 4/00; Published 10/00

Two Constraints for Scalable Online Recurrent Learning

Editor: Kevin Murphy and Bernhard Schölkopf

Abstract

Online scalable recurrent learning is challenging. Two popular gradient-based methods for recurrent learning are BPTT, and RTRL. BPTT looks at complete sequence before computing gradients, and is unsuitable for online updates. RTRL can do online updates, but scales poorly with the number of parameters. In this paper, we propose two constraints that make RTRL scalable. We show that by either decomposing the network into independent modules, or learning a recurrent network incrementally, we can make RTRL scale linearly with the number of parameters. Both approaches result in different algorithms, that can be combined. We show the strengths and weaknesses of the proposed algorithms on online prediction learning benchmarks.

Keywords: Agent-state construction, online learning, scalable learning, recurrent learning, representation learning

1. Introduction

Structural credit-assignment — identifying how to change network parameters to improve predictions — is an essential component of learning in neural networks. Effective structural credit-assignment requires tracking the influence of parameters on future predictions. In recurrent networks, a parameter can influence the internal state of the network which, in turn, can affect a prediction made many steps in the future.

Back-Propagation Through Time (BPTT) (Werbos, 1988; Robinson and Fallside, 1987) is a popular algorithm for gradient-based structural credit-assignment in RNNs. BPTT extends the back-propagation algorithm for feed-forward networks — independently proposed by Werbos (1974) and Rumelhart *et al.* (1986) — to RNNs by storing network activations from prior steps, and repeatedly applying the chain-rule starting from the output of the network and ending at the activations at the beginning of the sequence. BPTT is unsuitable for online learning as it requires memory proportional to the length of the sequence. Moreover, it delays gradient computation until the end of the sequence. For online learning, this sequence can be never-ending or arbitrarily long.

RTRL—an alternative to BPTT—was proposed by Williams and Zipser (1989). RTRL relies on forward-mode differentiation—using chain-rule to compute gradients in the direction of time—to compute gradients recursively. Unlike BPTT, RTRL does not delay gradient-computation until the final step. The memory requirement of RTRL also does not depend on the sequence length. As a result, it is a better candidate for real-time online learning. Unfortunately, RTRL requires maintaining the Jacobian $\frac{\partial h(t)}{\partial \theta}$ at every step, which requires $O(|h||\theta|)$ memory, where $|h|$ is the size of state of the network and $|\theta|$ is the number

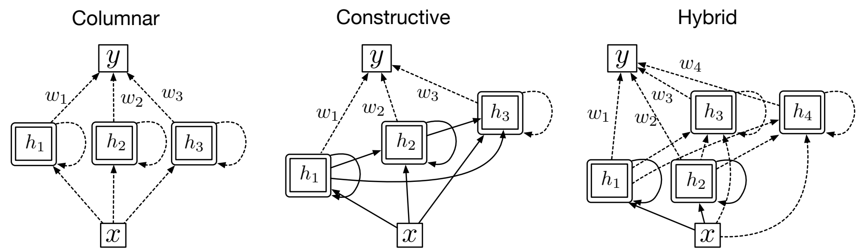

Figure 1: TODO: Make the caption concise. Three structures of recurrent neural networks that can be trained without truncation. Recurent networks with a columnar structure can be trained end-to-end using gradients without any truncation, only requiring $O(n)$ operations and memory per step. However, columnar networks do not have hierarchical recurrent features—recurrent features made out of other recurrent features. Constructive networks have hierarchical recurrent features, however must be trained incrementally to prevent bias. Incremental learning is achieved by initializing all w_i to zero, and learning h_1, h_2, and h_3 in order. Finally, columnar and constructive networks can be combined to get a hybrid network. The pairs (h_1, h_2) and (h_3, h_4) do not depend on each other, and can learn in parallel. Hoever, (h_3, h_4) must be learned after (h_1, h_2) have been learned and fixed.

of total parameters. The Jacobian is recursively updated by applying chain rule as:

$$\frac{\partial h(t+1)}{\partial \theta} = \frac{\partial h(t+1)}{\partial \theta(t+1)} + \frac{\partial h(t+1)}{\partial h(t)} \frac{\partial h(t)}{\partial \theta},$$

which requires $O(|h|^2|\theta|)$ operations and scales poorly to large networks.

A promising direction to scale gradient-based credit-assignment to large networks is to approximate the gradient. Elman (1990) proposed to ignore the influence of parameters on future predictions completely for training RNNs. This resulted in a scalable but biased algorithm. Williams and Peng (1990) proposed a more general algorithm called Truncated BPTT (T-BPTT). T-BPTT tracks the influence of all parameters on predictions made up to k steps in the future. T-BPTT limits the BPTT computation to last k activation, and works well for a range of problems (Mikolov *et al.*, 2009, 2010; Sutskever, 2013 and Kapturowski *et al.*, 2018). Its main limitation is that the resultant gradient is blind to long-range dependencies. Mujika *et al.* (2018) showed that on a simple copy task, T-BPTT failed to learn dependencies beyond the truncation window. Tallec *et al.* (2017) demonstrated T-BPTT can even diverge when a parameter has a negative long-term effect on a target and a positive short-term effect.

RTRL can also be approximated to reduce its computational overhead. Ollivier *et al.* (2015) and Tallec *et al.* (2017) proposed NoBacktrack and UORO. Both of these algorithms provide stochastic unbiased estimates of the gradient and scale well. However, their estimates have high variance and require extremely small step sizes for effective learning.

Cooijmans and Martens (2019) and Menick *et al.* (2021) showed that, for practical problems, UORO does not perform well due to its high variance compared to other biased approximations. Menick *et al.* (2021) proposed an approximation to RTRL called SnAp-k. SnAp-k tracks the influence of a parameter on a state only if the parameter can influence the state within k steps. It first identifies parameters whose influence on a state is zero for k steps and then assumes the future influence to be zero as well. For the remaining parameters, it tracks their influence on all future predictions. The bias introduced by SnAp-k is fundamentally different than the bias introduced by T-BPTT. SnAp-1 can be computationally efficient but introduces significant bias. SnAp-k for $k > 1$ reduces bias but can be as expensive as RTRL for dense RNNs. Menick *et al.* (2021) further proposed using sparse connections as a way to make SnAp more scalable. Connection sparsity reduces the number of parameters that can influence a state within k steps
Remainder omitted in this sample. See http://www.jmlr.org/papers/ for full paper.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF grant IIS-9988642) and the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637).

Appendix A.

A. Forward-mode gradient computation of an LSTM cell

Here we derive the update equations for recursively computing the gradients of a single LSTM based recurrent column. Each column has a single hidden unit. Because all columns are identical, the same update equations can be used to learn the complete Columnar Network. The state of an LSTM column is updated using following equations:

$$i(t) = \sigma(W_i^T x_k(t) + u_i h(t-1) + b_i) \tag{1}$$
$$f(t) = \sigma(W_f^T x_k(t) + u_f h(t-1) + b_f) \tag{2}$$
$$o(t) = \sigma(W_o^T x_k(t) + u_o h(t-1) + b_o) \tag{3}$$
$$g(t) = \phi(W_g^T x_k(t) + u_g h(t-1) + b_g) \tag{4}$$
$$c(t) = f(t)c(t-1) + i(t)g(t) \tag{5}$$
$$h(t) = o(t)\phi(c(t)) \tag{6}$$

where σ and ϕ are the sigmoid and tanh activation functions, $h(t)$ is the state of the column at time t and $W_i^T x_k(t) = \sum_{k=1}^{m} W_{i_k} x_k(t)$. The derivative of $\sigma(x)$ and $\phi(x)$ w.r.t to x are $\sigma(x)(1-\sigma(x))$ and $(1-\phi^2(x))$ respectively.

Let the length of input vector x be m. Then, W_i, W_f, W_o and W_g are vectors of length m whereas $u_i, b_i, u_f, b_f, u_o, b_o, u_g$ and b_g are scalars. We want to compute gradient of $h(t)$ with respect to all the parameters. We derive the update equations for $\frac{\partial h(t)}{\partial W_i}, \frac{\partial h(t)}{\partial u_i}, \frac{\partial h(t)}{\partial b_i}, \frac{\partial h(t)}{\partial W_f}, \frac{\partial h(t)}{\partial u_f}, \frac{\partial h(t)}{\partial b_f}, \frac{\partial h(t)}{\partial W_o}, \frac{\partial h(t)}{\partial u_o}, \frac{\partial h(t)}{\partial b_o}, \frac{\partial h(t)}{\partial W_g}, \frac{\partial h(t)}{\partial u_g}$, and $\frac{\partial h(t)}{\partial b_g}$ in the following sections.

A.1 $\frac{\partial h(t)}{\partial W_i}$

$W_i = (W_{i_1}, W_{i_2}, \cdots, W_{i_m})$ is a vector of length m. Since all elements of W_i are symmetric, we show gradient derivation for W_{i_j} without loss of generality. Let

$$TH_{W_{i_j}}(t) := \frac{\partial h(t)}{\partial W_{i_j}} \qquad \textit{(By definition)} \tag{7}$$
$$TH_{W_{i_j}}(0) := 0 \qquad \textit{(By definition)} \tag{8}$$
$$TC_{W_{i_j}}(t) := \frac{\partial c(t)}{\partial W_{i_j}} \qquad \textit{(By definition)} \tag{9}$$
$$TC_{W_{i_j}}(0) := 0 \qquad \textit{(By definition)} \tag{10}$$

Then:

$$
\begin{aligned}
TH_{W_{i_j}}(t) &= \frac{\partial}{\partial W_{i_j}}\left(o(t)\phi(c(t))\right) && \textit{From equation 6 and definition 7}\\
&= o(t)\frac{\partial \phi(c(t))}{\partial W_{i_j}} + \phi(c(t))\frac{\partial o(t)}{\partial W_{i_j}} && \textit{Product rule of differentiation}\\
&= o(t)(1-\phi^2(c(t)))\frac{\partial c(t)}{\partial W_{i_j}} + \phi(c(t))\frac{\partial o(t)}{\partial W_{i_j}} && \textit{Derivative of } \phi(x) \textit{ is (1-}\phi^2(x))\\
&= o(t)(1-\phi^2(c(t)))TC_{W_{i_j}}(t) + \phi(c(t))\frac{\partial o(t)}{\partial W_{i_j}} && \textit{From definition 9}\\
\frac{\partial o(t)}{\partial W_{i_j}} &= \frac{\partial}{\partial W_{i_j}}\sigma(W_o^T x(t) + u_o h(t-1) + b_o) && \textit{From equation 3}\\
&= \sigma(y)(1-\sigma(y))u_o TH_{W_{i_j}}(t-1) && \textit{Where } y \textit{ equals } W_o^T x(t) + u_o h(t-1) +\\
TC_{W_{i_j}}(t) &= \frac{\partial c(t)}{\partial W_{i_j}} && \textit{From definition 9}\\
&= \frac{\partial}{\partial W_{i_j}}(f(t)c(t-1) + i(t)g(t)) && \textit{From equation 5}\\
&= f(t)TC_{W_{i_j}}(t-1) + c(t-1)\frac{\partial f(t)}{\partial W_{i_j}} + \frac{\partial}{\partial W_{i_j}}(i(t)g(t)) && \textit{Product rule and definition 9}\\
&= f(t)TC_{W_{i_j}}(t-1) + c(t-1)\frac{\partial f(t)}{\partial W_{i_j}} + i(t)\frac{\partial g(t)}{\partial W_{i_j}} + g(t)\frac{\partial i(t)}{\partial W_{i_j}} && \textit{Product rule}
\end{aligned}
$$

Where gradient of $g(t)$ w.r.t W_{i_j} is:

$$
\begin{aligned}
\frac{\partial g(t)}{\partial W_{i_j}} &= \frac{\partial}{\partial W_{i_j}}\phi(W_g^T x(t) + u_g h(t-1) + b_g) && \textit{From equation 4}\\
&= (1-\phi^2(y))u_g TH_{W_{i_j}}(t-1) && \textit{Where } y \textit{ equals } W_g^T x(t) + u_g h(t-1) + b_g
\end{aligned}
$$

, gradient of $f(t)$ w.r.t W_{i_j} is:

$$
\begin{aligned}
\frac{\partial f(t)}{\partial W_{i_j}} &= \frac{\partial}{\partial W_{i_j}}\sigma(W_f^T x(t) + u_f h(t-1) + b_f) && \textit{From equation 2}\\
&= \sigma(y)(1-\sigma(y))u_f TH_{W_{i_j}}(t-1) && \textit{Where } y \textit{ equals } W_f^T x(t) + u_f h(t-1) + b_f
\end{aligned}
$$

and gradient of $i(t)$ w.r.t W_{i_j} is:

$$
\begin{aligned}
\frac{\partial i(t)}{\partial W_{i_j}} &= \frac{\partial}{\partial W_{i_j}}\sigma(W_i^T x(t) + u_i h(t-1) + i_f) && \textit{From equation 1}\\
&= \sigma(y)(1-\sigma(y))\left(x_j(t) + u_i TH_{W_{i_j}}(t-1)\right) && \textit{Where } y \textit{ equals } W_i^T x(t) + u_i h(t-1) + b_i
\end{aligned}
$$

The derivation shows that using two traces per parameter of W_i, it is possible to compute the gradient of $h(t)$ w.r.t W_i recursively. We provide the derivations for parameters u_i and b_i below. We skip the step-by-step derivations for the remaining parameters as they are similar.

A.2 $\frac{\partial h(t)}{\partial u_i}$

$$TH_{u_i}(t) := \frac{\partial h(t)}{\partial u_i} \qquad \textit{(By definition)} \tag{11}$$

$$TH_{u_i}(0) := 0 \qquad \textit{(By definition)} \tag{12}$$

$$TC_{u_i}(t) := \frac{\partial c(t)}{\partial u_i} \qquad \textit{(By definition)} \tag{13}$$

$$TC_{u_i}(0) := 0 \qquad \textit{(By definition)} \tag{14}$$

$$\begin{aligned}
TH_{u_i}(t) &= \frac{\partial}{\partial u_i}\left(o(t)\phi(c(t))\right) && \textit{From equation 6} \\
&= o(t)\frac{\partial \phi(c(t))}{\partial u_i} + \phi(c(t))\frac{\partial o(t)}{\partial u_i} && \textit{Product rule} \\
&= o(t)(1-\phi^2(c(t)))\frac{\partial c(t)}{\partial u_i} + \phi(c(t))\frac{\partial o(t)}{\partial u_i} && \textit{Derivative of } \phi(x) \textit{ is } 1-\phi^2(x) \\
&= o(t)(1-\phi^2(c(t)))TC_{u_i}(t) + \phi(c(t))\frac{\partial o(t)}{\partial u_i} && \textit{Using definition 13} \\
\frac{\partial o(t)}{\partial u_i} &= \frac{\partial}{\partial u_i}\sigma(W_o^T x(t) + u_o h(t-1) + b_o) && \textit{Using equations 3} \\
&= \sigma(x)(1-\sigma(x))u_o TH_{u_i}(t-1) && \textit{Where } x \textit{ equal } W_o^T x(t) + u_o h(t-1) + b_o \\
TC_{u_i}(t) &= \frac{\partial c(t)}{\partial u_i} && \textit{Definition 13} \\
&= \frac{\partial}{\partial u_i}(f(t)c(t-1) + i(t)g(t)) && \textit{From equation 6} \\
&= f(t)TC_{u_i}(t-1) + c(t-1)\frac{\partial f(t)}{\partial u_i} + \frac{\partial}{\partial u_i}(i(t)g(t))) && \textit{Product rule} \\
&= f(t)TC_{u_i}(t-1) + c(t-1)\frac{\partial f(t)}{\partial u_i} + i(t)\frac{\partial g(t)}{\partial u_i} + g(t)\frac{\partial i(t)}{\partial u_i} && \textit{Product rule}
\end{aligned}$$

Gradient of $g(t)$ w.r.t u_i is:

$$\begin{aligned}
\frac{\partial g(t)}{\partial u_i} &= \frac{\partial}{\partial u_i}\phi(W_g^T x(t) + u_g h(t-1) + b_g) && \textit{From equations 6} \\
&= (1-\phi^2(y))u_g TH_{u_i}(t-1) && \textit{Where } y \textit{ equals } W_g^T x(t) + u_g h(t-1) + b_g
\end{aligned}$$

, gradient of $f(t)$ w.r.t u_i is:

$$\begin{aligned}\frac{\partial f(t)}{\partial u_i} &= \frac{\partial}{\partial u_i}\sigma(W_f^T x(t) + u_f h(t-1) + b_f) && \textit{From equations 1}\\ &= \sigma(y)(1-\sigma(y))u_f TH_{u_i}(t-1) && \textit{Where } y \textit{ equals } W_f^T x(t) + u_f h(t-1) + b_f\end{aligned}$$

and the gradient of $i(t)$ w.r.t u_i is

$$\begin{aligned}\frac{\partial i(t)}{\partial u_i} &= \frac{\partial}{\partial u_i}\sigma(W_i^T x(t) + u_i h(t-1) + b_i) && \textit{Using equations 1}\\ &= \sigma(y)(1-\sigma(y))\left(h(t-1) + u_i TH_{u_i}(t-1)\right) && \textit{Where } y \textit{ equals } W_i^T x(t) + u_i h(t-1) + b_i\end{aligned}$$

A.3 $\frac{\partial h(t)}{\partial b_i}$

$$\begin{aligned}TH_{b_i}(t) &:= \frac{\partial h(t)}{\partial b_i} && \textit{(By definition)} && (15)\\ TH_{b_i}(0) &:= 0 && \textit{(By definition)} && (16)\\ TC_{b_i}(t) &:= \frac{\partial c(t)}{\partial b_i} && \textit{(By definition)} && (17)\\ TC_{b_i}(0) &:= 0 && \textit{(By definition)} && (18)\end{aligned}$$

$$\begin{aligned}TH_{b_i}(t) &= \frac{\partial}{\partial b_i}\left(o(t)\phi(c(t))\right) && \textit{From equation 6}\\ &= o(t)\frac{\partial \phi(c(t))}{\partial b_i} + \phi(c(t))\frac{\partial o(t)}{\partial b_i} && \textit{Product rule}\\ &= o(t)(1-\phi^2(c(t)))\frac{\partial c(t)}{\partial b_i} + \phi(c(t))\frac{\partial o(t)}{\partial b_i} && \textit{Derivative of of } \phi(x) \textit{ is } 1-\phi^2(x)\\ &= o(t)(1-\phi^2(c(t)))TC_{b_i}(t) + \phi(c(t))\frac{\partial o(t)}{\partial b_i} && \textit{From definition 17}\\ \frac{\partial o(t)}{\partial b_i} &= \frac{\partial}{\partial b_i}\sigma(W_o^T x(t) + u_o h(t-1) + b_o) && \textit{From equations 3}\\ &= \sigma(y)(1-\sigma(y))u_o TH_{b_i}(t-1) && \textit{Where } y \textit{ equal } W_o^T x(t) + u_o h(t-1) + b_o\\ TC_{b_i}(t) &= \frac{\partial c(t)}{\partial b_i} && \textit{From definition 17}\\ &= \frac{\partial}{\partial b_i}(f(t)c(t-1) + i(t)g(t)) && \textit{From equation 5}\\ &= f(t)TC_{b_i}(t-1) + c(t-1)\frac{\partial f(t)}{\partial b_i} + \frac{\partial}{\partial b_i}i(t)g(t) && \textit{Product rule}\\ &= f(t)TC_{b_i}(t-1) + c(t-1)\frac{\partial f(t)}{\partial b_i} + i(t)\frac{\partial g(t)}{\partial b_i} + g(t)\frac{\partial i(t)}{\partial b_i} && \textit{Product rule}\end{aligned}$$

Where gadient of $g(t)$ w.r.t b_i is:

$$\begin{aligned}\frac{\partial g(t)}{\partial b_i} &= \frac{\partial}{\partial b_i}\phi(W_g^T x(t) + u_g h(t-1) + b_g) && \textit{From equation 4}\\ &= (1-\phi^2(y))u_g TH_{b_i}(t-1) && \textit{Where } y \textit{ equal } W_g^T x(t) + u_g h(t-1) + b_g\end{aligned}$$

, gradient of $f(t)$ w.r.t b_i is:

$$\begin{aligned}\frac{\partial f(t)}{\partial b_i} &= \frac{\partial}{\partial b_i}\sigma(W_f^T x(t) + u_f h(t-1) + b_f) && \textit{From equation 2}\\ &= \sigma(y)(1-\sigma(y))u_f TH_{b_i}(t-1) && \textit{Where } y \textit{ equal } W_f^T x(t) + u_f h(t-1) + b_f\end{aligned}$$

and gradient of $i(t)$ w.r.t b_i is:

$$\begin{aligned}\frac{\partial i(t)}{\partial b_i} &= \frac{\partial}{\partial b_i}\sigma(W_i^T x(t) + u_i h(t-1) + b_i) && \textit{From equation 1}\\ &= \sigma(y)(1-\sigma(y))\left(u_i TH_{b_i}(t-1) + 1\right) && \textit{Where } y \textit{ equal } W_i^T x(t) + b_i h(t-1) + b_i\end{aligned}$$

A.4 $\frac{\partial h(t)}{\partial W_{f_j}}$

The derivations for the remaining parameters is analogous to what previous derivations. The final equations are as follows.

$$\begin{aligned}\frac{\partial g(t)}{\partial W_{f_j}} &= (1-\phi^2(y))(u_g TH_{W_{f_j}}(t-1))\\ \frac{\partial f(t)}{\partial W_{f_j}} &= \sigma(y)(1-\sigma(y))(x_j + u_f TH_{W_{f_j}}(t-1))\\ \frac{\partial i(t)}{\partial W_{f_j}} &= \sigma(y)(1-\sigma(y))(u_i TH_{W_{f_j}}(t-1))\\ \frac{\partial o(t)}{\partial W_{f_j}} &= \sigma(y)(1-\sigma(y))(u_o TH_{W_{f_j}}(t-1))\\ TC_{W_{f_j}} &= f(t)TC_{f_j}(t-1) + c(t-1)\frac{\partial f(t)}{\partial b_i} + i(t)\frac{\partial g(t)}{\partial b_i} + g(t)\frac{\partial i(t)}{\partial b_i}\\ TH_{W_{f_j}} &= o(t)(1-\phi^2(c(t)))TC_{W_{f_j}}(t) + \phi(c(t))\frac{\partial o(t)}{\partial W_{ij}}\end{aligned} \tag{19}$$

A.5 $\frac{\partial h(t)}{\partial W_{o_j}}$

$$
\begin{aligned}
\frac{\partial g(t)}{\partial W_{o_j}} &= (1-\phi^2(y))(u_g TH_{W_{o_j}}(t-1)) \\
\frac{\partial f(t)}{\partial W_{o_j}} &= \sigma(y)(1-\sigma(y))(u_f TH_{W_{o_j}}(t-1)) \\
\frac{\partial i(t)}{\partial W_{o_j}} &= \sigma(y)(1-\sigma(y))u_i TH_{W_{o_j}}(t-1) \\
\frac{\partial o(t)}{\partial W_{o_j}} &= \sigma(x)(1-\sigma(x))(x_j + u_o TH_{W_{o_j}}(t-1)) \\
TC_{W_{o_j}} &= f(t)TC_{o_j}(t-1) + c(t-1)\frac{\partial f(t)}{\partial b_i} + i(t)\frac{\partial g(t)}{\partial b_i} + g(t)\frac{\partial i(t)}{\partial b_i} \\
TH_{W_{o_j}} &= o(t)(1-\phi^2(c(t)))TC_{W_{o_j}}(t) + \phi(c(t))\frac{\partial o(t)}{\partial W_{ij}}
\end{aligned}
\tag{20}
$$

A.6 $\frac{\partial h(t)}{\partial W_{g_j}}$

$$
\begin{aligned}
\frac{\partial g(t)}{\partial W_{g_j}} &= (1-\phi^2(y))(x_j + u_g TH_{W_{g_j}}(t-1)) \\
\frac{\partial f(t)}{\partial W_{g_j}} &= \sigma(y)(1-\sigma(y))(u_f TH_{W_{g_j}}(t-1)) \\
\frac{\partial i(t)}{\partial W_{g_j}} &= \sigma(y)(1-\sigma(y))(u_i TH_{W_{g_j}}(t-1)) \\
\frac{\partial o(t)}{\partial W_{g_j}} &= \sigma(x)(1-\sigma(x))(u_o TH_{W_{g_j}}(t-1)) \\
TC_{W_{g_j}} &= f(t)TC_{g_j}(t-1) + c(t-1)\frac{\partial f(t)}{\partial b_i} + i(t)\frac{\partial g(t)}{\partial b_i} + g(t)\frac{\partial i(t)}{\partial b_i} \\
TH_{W_{g_j}} &= o(t)(1-\phi^2(c(t)))TC_{W_{g_j}}(t) + \phi(c(t))\frac{\partial o(t)}{\partial W_{ij}}
\end{aligned}
\tag{21}
$$

A.7 $\frac{\partial h(t)}{\partial u_o}$

$$
\begin{aligned}
\frac{\partial g(t)}{\partial u_o} &= (1-\phi^2(y))(u_g TH_{u_o}(t-1)) \\
\frac{\partial f(t)}{\partial u_o} &= \sigma(y)(1-\sigma(y))(u_f TH_{u_o}(t-1)) \\
\frac{\partial i(t)}{\partial u_o} &= \sigma(y)(1-\sigma(y))(u_i TH_{u_o}(t-1)) \\
\frac{\partial o(t)}{\partial u_o} &= \sigma(x)(1-\sigma(x))(u_o TH_{u_o}(t-1)+h(t-1)) \\
TC_{u_o} &= f(t)TC_{i_j}(t-1)+c(t-1)\frac{\partial f(t)}{\partial b_i}+i(t)\frac{\partial g(t)}{\partial b_i}+g(t)\frac{\partial i(t)}{\partial b_i} \\
TH_{u_o} &= o(t)(1-\phi^2(c(t)))TC_{u_o}(t)+\phi(c(t))\frac{\partial o(t)}{\partial W_{ij}}
\end{aligned}
\tag{22}
$$

A.8 $\frac{\partial h(t)}{\partial u_f}$

$$
\begin{aligned}
\frac{\partial g(t)}{\partial u_f} &= (1-\phi^2(y))(u_g TH_{u_f}(t-1)) \\
\frac{\partial f(t)}{\partial u_f} &= \sigma(y)(1-\sigma(y))(u_f TH_{u_f}(t-1)+h(t-1)) \\
\frac{\partial i(t)}{\partial u_f} &= \sigma(y)(1-\sigma(y))(u_i TH_{u_f}(t-1)) \\
\frac{\partial o(t)}{\partial u_f} &= \sigma(x)(1-\sigma(x))(u_o TH_{u_f}(t-1)) \\
TC_{u_f} &= f(t)TC_{i_j}(t-1)+c(t-1)\frac{\partial f(t)}{\partial b_i}+i(t)\frac{\partial g(t)}{\partial b_i}+g(t)\frac{\partial i(t)}{\partial b_i} \\
TH_{u_f} &= o(t)(1-\phi^2(c(t)))TC_{u_f}(t)+\phi(c(t))\frac{\partial o(t)}{\partial W_{ij}}
\end{aligned}
\tag{23}
$$

A.9 $\frac{\partial h(t)}{\partial u_g}$

$$\begin{aligned}
\frac{\partial g(t)}{\partial u_g} &= (1-\phi^2(y))(u_g TH_{u_g}(t-1) + h(t-1)) \\
\frac{\partial f(t)}{\partial u_g} &= \sigma(y)(1-\sigma(y))(u_f TH_{u_g}(t-1)) \\
\frac{\partial i(t)}{\partial u_g} &= \sigma(y)(1-\sigma(y))(u_i TH_{u_g}(t-1)) \\
\frac{\partial o(t)}{\partial u_g} &= \sigma(x)(1-\sigma(x))(u_o TH_{u_g}(t-1)) \\
TC_{u_g} &= f(t)TC_{i_j}(t-1) + c(t-1)\frac{\partial f(t)}{\partial b_i} + i(t)\frac{\partial g(t)}{\partial b_i} + g(t)\frac{\partial i(t)}{\partial b_i} \\
TH_{u_g} &= o(t)(1-\phi^2(c(t)))TC_{u_g}(t) + \phi(c(t))\frac{\partial o(t)}{\partial W_{ij}}
\end{aligned} \tag{24}$$

A.10 $\frac{\partial h(t)}{\partial b_g}$

$$\begin{aligned}
\frac{\partial g(t)}{\partial b_g} &= (1-\phi^2(y))(u_g TH_{b_g}(t-1) + 1) \\
\frac{\partial f(t)}{\partial b_g} &= \sigma(y)(1-\sigma(y))(u_f TH_{b_g}(t-1)) \\
\frac{\partial i(t)}{\partial b_g} &= \sigma(y)(1-\sigma(y))(u_i TH_{b_g}(t-1)) \\
\frac{\partial o(t)}{\partial b_g} &= \sigma(x)(1-\sigma(x))(u_o TH_{b_g}(t-1)) \\
TC_{b_g} &= f(t)TC_{i_j}(t-1) + c(t-1)\frac{\partial f(t)}{\partial b_i} + i(t)\frac{\partial g(t)}{\partial b_i} + g(t)\frac{\partial i(t)}{\partial b_i} \\
TH_{b_g} &= o(t)(1-\phi^2(c(t)))TC_{b_g}(t) + \phi(c(t))\frac{\partial o(t)}{\partial W_{ij}}
\end{aligned} \tag{25}$$

A.11 $\frac{\partial h(t)}{\partial b_f}$

$$
\begin{aligned}
\frac{\partial g(t)}{\partial b_f} &= (1-\phi^2(y))(u_g TH_{b_f}(t-1)) \\
\frac{\partial f(t)}{\partial b_f} &= \sigma(y)(1-\sigma(y))(u_f TH_{b_f}(t-1)+1) \\
\frac{\partial i(t)}{\partial b_f} &= \sigma(y)(1-\sigma(y))(u_i TH_{b_f}(t-1)) \\
\frac{\partial o(t)}{\partial b_f} &= \sigma(x)(1-\sigma(x))(u_o TH_{b_f}(t-1)) \\
TC_{b_f} &= f(t)TC_{i_j}(t-1) + c(t-1)\frac{\partial f(t)}{\partial b_i} + i(t)\frac{\partial g(t)}{\partial b_i} + g(t)\frac{\partial i(t)}{\partial b_i} \\
TH_{b_f} &= o(t)(1-\phi^2(c(t)))TC_{b_f}(t) + \phi(c(t))\frac{\partial o(t)}{\partial W_{ij}}
\end{aligned}
\tag{26}
$$

A.12 $\frac{\partial h(t)}{\partial b_o}$

$$
\begin{aligned}
\frac{\partial g(t)}{\partial b_o} &= (1-\phi^2(y))(u_g TH_{b_o}(t-1)) \\
\frac{\partial f(t)}{\partial b_o} &= \sigma(y)(1-\sigma(y))(u_f TH_{b_o}(t-1)) \\
\frac{\partial i(t)}{\partial b_o} &= \sigma(y)(1-\sigma(y))(u_i TH_{b_o}(t-1)) \\
\frac{\partial o(t)}{\partial b_o} &= \sigma(x)(1-\sigma(x))(u_o TH_{b_o}(t-1)+1) \\
TC_{b_o} &= f(t)TC_{i_j}(t-1) + c(t-1)\frac{\partial f(t)}{\partial b_i} + i(t)\frac{\partial g(t)}{\partial b_i} + g(t)\frac{\partial i(t)}{\partial b_i} \\
TH_{b_o} &= o(t)(1-\phi^2(c(t)))TC_{b_o}(t) + \phi(c(t))\frac{\partial o(t)}{\partial W_{ij}}
\end{aligned}
\tag{27}
$$

References