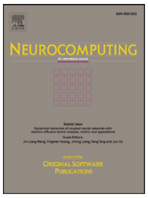
Neurocomputing 000 (2018) 1–9

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Prototypical recurrent unit

Dingkun Long[a], Richong Zhang[a,*], Yongyi Mao[b]

[a] BDBC and SKLSDE, School of Computer Science and Engineering, Beihang University, 37 Xueyuan Road, Beijing 100191, China
[b] School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Avenue, K1N 6N5 ON, Canada

ARTICLE INFO

Article history:
Received 26 May 2017
Revised 23 March 2018
Accepted 17 May 2018
Available online xxx

Communicated by Dr. Deng Cheng

Keywords:
Recurrent unit
Recurrent network
State space

ABSTRACT

Despite the great successes of deep learning, the effectiveness of deep neural networks, such as LSTM/GRU-like recurrent networks, has not been well understood. Not only attributed to their nonlinear dynamics, the difficulty in understanding LSTM/GRU-like recurrent networks also resides in the highly complex recurrence structure in these networks. This work aims at constructing an alternative recurrent unit that is as simple as possible and yet also captures the key components of LSTM/GRU recurrent units. Such a unit, if available, can then be used as a prototype for the study of LSTM/GRU-like networks and potentially enable easier analysis. Towards that goal, we take a system-theoretic perspective to design a new recurrent unit, which we call the prototypical recurrent unit (PRU). Not only having minimal complexity, PRU is demonstrated experimentally to have comparable performance to GRU and LSTM over a range of modelling tasks. This establishes PRU networks as a prototypical example for future study of LSTM/GRU-like recurrent networks. The complexity advantage of PRU may also make it a favourable alternative to LSTM and GRU in practice.

1. Introduction

Deep learning has demonstrated great power in the recent years and appears to have prevailed in a broad spectrum of application domains (see, e.g., [12,17]). Despite its successes, the effectiveness of deep neural networks has not been understood at a theoretical depth. Thus developing novel analytic tools and theoretical frameworks for studying deep neural networks is of the greatest importance at the present time, and is anticipated to be a central subject of machine learning research in the years to come.

This work is motivated by the thrust of understanding recurrent neural networks, particularly LSTM/GRU-like networks [4,8,9,13,20]. These networks are demonstrated as the state-of-the-art models for time series or sequence data [1,10,22]. Recently LSTM/GRU recurrent units have also been successfully adopted for modelling other forms of data (e.g., [3,23]). Despite these successes, the design of LSTM and GRU recurrent units was in fact heuristical; to date there is little theoretical analysis justifying their effectiveness. A particularly interesting observation regarding these networks is that they appear to possess “long-term memory”, namely, being able to selectively “remember” the information from many time steps ago [7]. As one may naturally expect such memorization capability to have played an important role in the working of these networks, this aspect has not been well studied, analytically or experimentally.

The difficulty in analyzing recurrent networks resides in the complex structure of the recurrent unit, which induces highly complex nonlinear dynamics. To understand LSTM-like recurrent networks, the methodology explored in this research is to maximally simplify the structure of the recurrent unit. That is, we wish to construct an alternative recurrent unit that captures the key components LSTM and GRU but stays as simple as possible. Such a unit can then be used for the study of recurrent networks and its structural simplicity may allow easier analysis in future research.

Towards that goal, the main objective of this present paper is to design such a recurrent unit and verify that this unit performs comparably to LSTM and GRU. To that end, we develop a new recurrent unit, which we call the *Prototypical Recurrent Unit* (PRU). We rationalize our design methodology from a system-theoretic perspective where a recurrent unit is understood as a causal time-invariant system in state-space representations. Insights from previous research suggest that additive evolution appear essential for LSTM-like networks to avoid the “gradient-vanishing” problem under back-propagation [5,14,18]. This understanding is also exploited in our design of PRU.

The performance of PRU is verified and compared against LSTM and GRU via extensive experiments. Using these three kinds of recurrent unit, we not only experiment on constructing a standard language model for character prediction [19], but also test the recurrent units for two controlled learning tasks, the Adding Problem

[*] Corresponding author.
E-mail address: zhangrc@act.buaa.edu.cn (R. Zhang).

https://doi.org/10.1016/j.neucom.2018.05.048
0925-2312/© 2018 Elsevier B.V. All rights reserved.

[13], and the Memorization Problem. The latter problem is what we propose in this work specifically for studying the memorization capability of the recurrent networks. All experimental results confirm that PRU performs comparably to LSTM and GRU, achieving the purpose of this paper.

As another contribution, our experiments in this work demonstrate that the intrinsic memorization capability of the recurrent units depends critically on the dimension of the state space. The amount of targeted information (for memorization), the duration of memory, and the intensity of the interfering signal also directly impact the memorization performance.

Finally it is perhaps worth noting that although PRU is designed to be a prototype which hopefully allows for easier analysis in future research, our experiments suggest that it can also be used as a practical alternative to LSTM and GRU. A particular advantage of PRU is its time complexity. In this metric, PRU is arguably superior to both LSTM and GRU.

2. State-space representations

In system theory [15], a (discrete-time) system can be understood as any physical or conceptual device that responds to an *input sequence* $x_1, x_2, \ldots$ and generates an *output sequence* $y_1, y_2, \ldots,$ where the indices of the sequences are discrete time. In general, each x_t and each y_t at any time t may be a vector of arbitrary dimensions. We will then use $\mathcal{X}$ and $\mathcal{Y}$ to denote the vector spaces from which x_t and y_t take value respectively. We will call $\mathcal{X}$ the *input space* and $\mathcal{Y}$ the *output space*. The behaviour of the system is characterized by a function J that maps the space of all input sequences to the space of all output sequences. Then two systems J and J' are *equivalent* if J and J' are identical as functions.

The class of systems that are of primary interest are causal systems, namely those in which the output y_t at each time t is independent of all future inputs $x_{t+1}, x_{t+2}, \ldots$. The grand idea in system theory is arguably the introduction of the notion of *state* to causal systems [15]. This makes state-space models the central topic in system theory, resulting in wide and profound impact on system analysis and design. In a nutshell, a *state* is an quantity internal to the system, serving as a complete summary of all past inputs so that *given the current state, the current and future outputs are independent of all past inputs*.

In this perspective, a recurrent unit can be regarded precisely as a causal time-invariant system in a state-space representation. We now formalize such a state-space representations.

At each time instant t, in addition to the input variable x_t and output variable y_t, the representation of a recurrent unit also contains a state variable s_t, taking values in a vector space $\mathcal{S}$, which will be referred to as the *state space*. Before the system is excited by the input, or at time $t = 0$, it is assumed that the state variable s_0 takes certain initial configuration, which is assumed customarily to be the origin $0 \in \mathcal{S}$.

The behaviour of the recurrent unit is governed by two functions $F : \mathcal{X} \times \mathcal{S} \to \mathcal{S}$ and $G : \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$ as follows. At each time instant t, function F maps the current input x_t and the previous state s_{t-1} to the current state s_t, namely, via

$$s_t = F(x_t, s_{t-1}), \tag{1}$$

and function G maps the current input x_t and the current state s_t to the current output y_t, namely, via

$$y_t = G(x_t, s_t). \tag{2}$$

That is, in general a recurrent unit can be specified by the tuple $(\mathcal{X}, \mathcal{Y}, \mathcal{S}, F, G)$ according to (1) and (2). We call such specification of the recurrent unit *Type-I state-space representation* of the unit, and denote it by $(\mathcal{X}, \mathcal{Y}, \mathcal{S}, F, G)_{\text{I}}$.

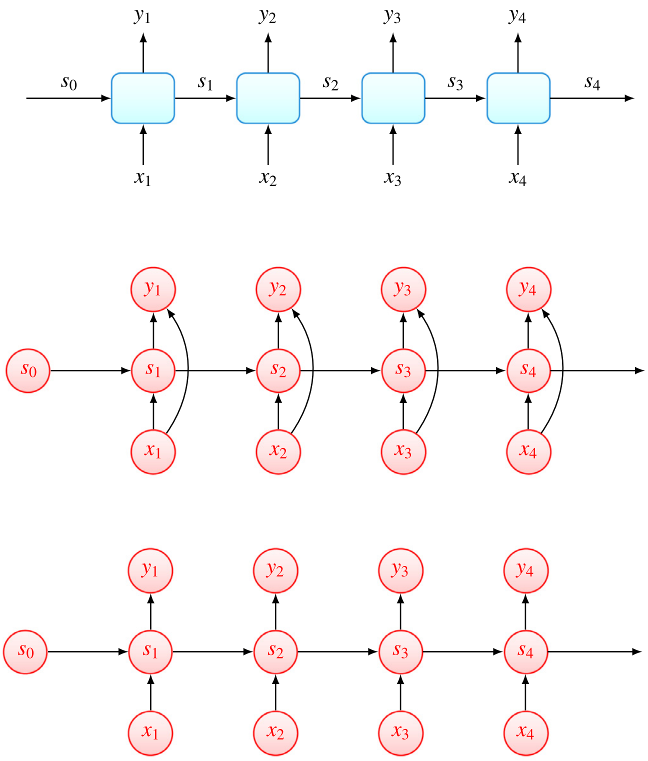


Fig. 1. A recurrent network(top) and the dependency structure of variables in Type-I representation (middle) and Type-II representation (bottom).

It is remarkable that Type-I state-space representation is generic for any causal time-invariant system and hence generic for any recurrent unit. To illustrate this, we take the LSTM network as an example.

The standard formulation of the LSTM network is given by the following equations:

$$i_t = \sigma(W_i[c_{t-1}, h_{t-1}, x_t] + b_i) \tag{3}$$

$$f_t = \sigma(W_f[c_{t-1}, h_{t-1}, x_t] + b_f) \tag{4}$$

$$o_t = \sigma(W_o[c_{t-1}, h_{t-1}, x_t] + b_o) \tag{5}$$

$$\tilde{c}_t = tanh(W_c[h_{t-1}, x_t] + b_g) \tag{6}$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \tag{7}$$

$$h_t = o_t \odot tanh(c_t) \tag{8}$$

where $\odot$ is the element-wise product. In these equations, if we take (c_t, h_t) as state s_t, and h_t as y_t, Eqs. (3–7) can be expressed as Eq. (1), and Eqs. (5) and (8) can be expressed as Eq. (2). We then arrive at a Type-I representation. It is also easy to verify that the recurrent unit in RNN [6] and GRU networks can all be expressed this way.

As a clarification which might be necessary for the remainder of this paper, we pause to remark that in this paper (and under a system-theoretic perspective), the notion of a recurrent unit and that of a recurrent (neural) network are synonyms. In particular, a recurrent unit that operates over n time instances may be viewed as n copies of the same recurrent unit connected in a chain-structured network as shown in Fig. 1 (top). In this "time-unfolded" view, the dependency structure between the variables in Type-I representation is shown in Fig. 1 (middle).

Since we aim at designing a *simpler* recurrent unit, we now introduce another simpler representation, which we call *Type-II*

Please cite this article as: D. Long et al., Prototypical recurrent unit, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.05.048