Knowledge-Based Systems 96 (2016) 104–114

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

On the importance of sluggish state memory for learning long term dependency

Jonathan A. Tepper*, Mahmud S. Shertil, Heather M. Powell

School of Science and Technology, Nottingham Trent University, Burton Street, Nottingham NG1 4BU, UK

ARTICLE INFO

Article history:
Received 7 May 2015
Revised 27 October 2015
Accepted 27 December 2015
Available online 5 January 2016

Keywords:
Simple Recurrent Networks
Vanishing gradient problem
Echo State Network
Grammar prediction task
Sluggish state space
Internal representation

ABSTRACT

The vanishing gradients problem inherent in Simple Recurrent Networks (SRN) trained with back-propagation, has led to a significant shift towards the use of Long Short-Term Memory (LSTM) and Echo State Networks (ESN), which overcome this problem through either second order error-carousel schemes or different learning algorithms, respectively.

This paper re-opens the case for SRN-based approaches, by considering a variant, the Multi-recurrent Network (MRN). We show that memory units embedded within its architecture can ameliorate against the vanishing gradient problem, by providing variable sensitivity to recent and more historic information through layer- and self-recurrent links with varied weights, to form a so-called sluggish state-based memory.

We demonstrate that an MRN, optimised with noise injection, is able to learn the long term dependency within a complex grammar induction task, significantly outperforming the SRN, NARX and ESN. Analysis of the internal representations of the networks, reveals that sluggish state-based representations of the MRN are best able to latch on to critical temporal dependencies spanning variable time delays, to maintain distinct and stable representations of *all* underlying grammar states. Surprisingly, the ESN was unable to fully learn the dependency problem, suggesting the major shift towards this class of models may be premature.

1. Introduction

Recent studies have demonstrated that the ability to learn non-linear temporal dynamic behaviour is a significant factor in the solution of numerous complex problem-solving tasks, such as those found in many practical problem domains e.g. natural language processing [1,2,41,42,43], speech processing [3] and financial modelling [4,5]. Recurrent neural networks (RNNs) are a class of connectionist network whose computational neurons produce activations based on the activation history of the network [6]. RNNs have nonlinear dynamics, allowing them to perform in a highly complex manner; network activations from previous time steps are fed back as input into the RNN at future time steps. In theory, the states of the hidden units can store data through time in the form of a distributed representation, and this can be used many time steps later to predict subsequent input vectors [7]. This particular characteristic distinguishes them from their feedforward counterpart (the Multi-layer Perceptron, MLP) and enables them to act as vector-sequence transducers [6].

Although there are numerous connectionist techniques for processing temporal information, historically, the most widely used RNN has been the Simple Recurrent Network (SRN) [1]. The SRN is a state-based model, similar in complexity to a Hidden Markov Model, and represents sequences of information as internal states of neural activation [8]. The SRN has proven remarkably useful for temporal problem domains such as natural language processing, and in particular, learning to model regular and simple context-free languages. Moreover, much research has been conducted to investigate the temporal processing ability of SRNs [8–15,35]. It has been shown that in some cases, the SRN and its variants are unable to learn time lagged information (dependencies) exceeding as few as 5 to 10 discrete time steps between relevant input events and target signals [18]. This is most likely due to their use of gradient descent learning where the gradient of the total output error, from previous input observations, vanishes quickly as the time lag between relevant inputs and errors increases [17]. SRNs have also been severely criticised for their lack of ability to model the combinatorial systematicity of human language [16,36]. This, however, has cogently been refuted by Christiansen & MacDonald [47] who demonstrate that the SRN is able to make non-local generalisations based on the structural regularities found in the training corpus [3] and appropriate constituent-based generalisations,

* Corresponding author. Tel.: +44 115 8488363.
E-mail address: jonathan.tepper@ntu.ac.uk (J.A. Tepper).

http://dx.doi.org/10.1016/j.knosys.2015.12.024
0950-7051/

providing further support for non-parametric usage-based models of language [48].

Researchers have investigated various architectural configurations to enhance the memory capacity of SRNs. For example, Ulbricht [19] introduced the Multi-Recurrent Network (MRN), which utilises variable-length memory banks with variable strength recurrent and self-recurrent links, to form a so-called sluggish state-based memory mechanism. The integration of variable state activations with the replication of state nodes, to form memory banks for representing temporal dependency, is a particularly distinctive feature, relative to other models in the SRN family. Dorffner [20] states that these sluggish state spaces can exploit the information from both recent time steps and the distant past to form a longer averaged history, and that this can help to solve long term dependency problems. The MRN has been successfully used to solve a number of complex prediction problems, yielding very competitive results over traditional SRNs, and has also faired competitively with Kernel methods [5,19]. These published results justify the additional connections and resulting complexity of the MRN. Another approach which utilises additional memory units within the architecture to overcome the vanishing gradient issue, is the Nonlinear AutoRegressive model process with eXogenous input (NARX) network, introduced by Lin et al. [34]. Unlike the MRN, the NARX does not utilise past state information; it was introduced to process temporal dependencies, primarily within the continuous domain. The NARX network is essentially an RNN. It contains feedback from the output layer and the input layer to create a context layer. This context layer represents temporal information explicitly, in the form of a shift-register of the previous activations. NARX networks therefore overcome the limitations of SRNs for encoding temporal dependency, by associating nodes with temporal information rather than state activations.

Despite the promise shown by the MRN and NARX networks over SRNs, there has been a strong shift away from the traditional SRN family of networks and towards more complex second order RNNs and those with different learning regimes. One such innovation is the Long Short-Term Memory (LSTM) introduced by Hochreiter and Schmidhuber [37] and developed further by Gers et al. [21]. This is a second-order RNN that consists of multiple recurrently connected subnets, called memory blocks. Each block contains a set of internal units having activations controlled by three multiplicative units (input gate, forget gate and output gate). This block-based mechanism enables an error carousel to be formed which enables the LSTM to latch on to appropriate error information. The LSTM is considered to provide state-of-the-art performance in numerous temporal modelling tasks, however without appropriate external resets, internal unit values can grow uncontrollably, creating instability. Ad hoc reset methods may therefore be required which adds to the complication of the design process.

Another type of RNN aimed at resolving the long temporal dependency problem is the Echo-state Network (ESN). The ESN has again exhibited state-of-the-art performance. It is similar in architecture to the SRN but it has an entirely different learning mechanism and so does not suffer from the vanishing or exploding gradient problem [22]. Training is reduced to a one-shot simple linear regression task applied to the output layer weights. ESNs have been applied with varying success to numerous problem domains such as behaviour classification, natural language processing [23,42,43,44], and speech recognition [24]. ESNs are gaining particular popularity with those researchers seeking biologically plausible models of language and memory. For example, Dominey [42] used ESN-like models to better capture the principles of neurophysiology and address the issue identified by Friederici [45] to explore the role of subcortical structures in language processing. In particular, Dominey suggested '*corticostriatal plasticity plays a role in implementing the structural mapping processes required for assignment of open-class elements to their appropriate thematic role*' and both Dominey [42] and Hinaut and Dominey [43] therefore sought to apply ESNs to implement a mechanism for the real-time parallel processing of conceptual and grammatical structures. Indeed, Pascanu and Jaeger [44] recognise that Dominey's earlier work in cognitive neuroscience with the development of the Temporal Recurrent Network (TRN) [41], independently discovered the reservoir principle that underpins ESNs. Although state-of-the-art performance has been reported with ESNs for the iterated prediction of noiseless time series data, the usefulness of this for discrete problem domains such as grammar induction (and state representation) is questionable. Moreover, studies with ESNs for realistic problem domains have revealed the difficulty of creating the reservoir of interconnections (connections between hidden units) in a systematic way for a given problem. It can take the exploration of many reservoir configurations before a solution is found [22,25]. Clearly, there is scope for advancing knowledge concerning the strengths and weaknesses of ESNs for different types of problem and a need for a principled approach to ESN application, appropriate to the problem domain in order to increase their utility. In particular, it will be interesting to determine whether the ESN learning algorithm is better able to discover the optimum solution for a complex grammar induction task than gradient descent-based learning methods used in traditional SRNs. This is important as RNNs, including SRNs and ESNs, are theoretically capable of representing universal Turing machines [46].

In this paper, we seek to ascertain whether the strong shift away from the SRN family of models trained with gradient descent for language acquisition tasks is premature. In particular, we provide further exploration of the MRN variant of the SRN, which appears to have gone largely unnoticed in the literature since 1996. In particular, we are interested in whether the unique MRN approach to associating temporal features with both nodes and state values, is sufficient to endow SRNs with superior power over ESNs enabling them to implicitly capture the sort of temporal dependency over variable time delays that may be associated with a complex grammatical structure. If this is shown to be the case, then this will have significant implications for current models of human memory and sentence comprehension that are dependent on ESN-like approaches, including those posed by [23,42,43,44].

2. Network architectures

2.1. Elman's Simple Recurrent Network

The SRN architecture employed in this study is depicted in Fig. 1. The activations of the hidden units of the network from time t are used as input to the network at time $t+1$. Recurrent

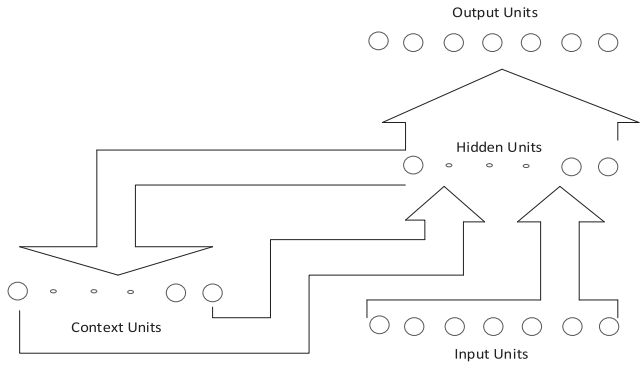

Fig 1. Simple Recurrent Network.