Knowledge-Based Systems 108 (2016) 144–154

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Learning word dependencies in text by means of a deep recurrent belief network

Iti Chaturvedi[a], Yew-Soon Ong[a], Ivor W. Tsang[b], Roy E. Welsch[c], Erik Cambria[a,*]

[a] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[b] Center for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Australia
[c] Sloan School of Management, MIT, USA

ARTICLE INFO

Article history:
Received 16 November 2015
Revised 8 July 2016
Accepted 12 July 2016
Available online 13 July 2016

Keywords:
Deep belief networks
Time-delays
Variable-order
Gaussian networks
Markov Chain Monte Carlo

ABSTRACT

We propose a deep recurrent belief network with distributed time delays for learning multivariate Gaussians. Learning long time delays in deep belief networks is difficult due to the problem of vanishing or exploding gradients with increase in delay. To mitigate this problem and improve the transparency of learning time-delays, we introduce the use of Gaussian networks with time-delays to initialize the weights of each hidden neuron. From our knowledge of time delays, it is possible to learn the long delays from short delays in a hierarchical manner. In contrast to previous works, here dynamic Gaussian Bayesian networks over training samples are evolved using Markov Chain Monte Carlo to determine the initial weights of each hidden layer of neurons. In this way, the time-delayed network motifs of increasing Markov order across layers can be modeled hierarchically using a deep model. To validate the proposed Variable-order Belief Network (VBN) framework, it is applied for modeling word dependencies in text. To explore the generality of VBN, it is further considered for a real-world scenario where the dynamic movements of basketball players are modeled. Experimental results obtained showed that the proposed VBN could achieve over 30% improvement in accuracy on real-world scenarios compared to the state-of-the-art baselines.

1. Introduction

Dynamic Gaussian networks (GN) have shown success in capturing temporal characteristics of data in sports and text processing [1]. They assume an underlying hidden state of a dynamic system evolving over time. For example, in basketball, the structure of the GN can be learned from the temporal movement of players over time and determine the differences in offensive formations between expert and beginners. Hence, the directed edges represent causal dependencies among the players and the time-delays associated with the edges define the dynamics. Dynamic GN are stochastic models where learning involves enumerating different local connectivity patterns consisting of child nodes given parent nodes at the previous two or three time points that may re-occur in one or more classes [2]. Such a directed sub-graph is referred to as a 'network motif'.

Although, GN outperform state-of-the-art classifiers including Bayesian Networks (BN) and Differential equations [3], training them requires large number of time samples. Hence, in this paper we consider a framework similar to feed-forward neural networks and extreme learning machines that can automatically learn temporal features with long term memory in a fast and easy manner with minimal human intervention and limited time samples [4,5].

Further, GN need to use additional memory nodes to learn time delays. Hence, the total number of nodes and motifs increases exponentially at very long delays, making it computationally intractable. Conditional random fields alleviate this problem by approximating a very long delay as a cascade of short delays to summarize text [6]. However, in long documents or reviews this can lead to formation of long and overlapping loops. Our goal is to efficiently predict the next class label in a sequence for high-dimensional networks and this can be done using the hierarchical structure of deep neural networks.

Recently, recurrent neural networks (RNN) have exhibited good performance for modeling temporal structures with few training samples [7]. This is because bi-directional RNN can model not only

Abbreviations: *BN* =, Bayesian Network; *CD* =, Contrastive Divergence; *DBN* =, Deep Belief Network; *GN* =, Gaussian Network; *RNN* =, Recurrent Neural Network; *VBN* =, Variable-order Belief Network; *MCMC* =, Markov Chain Monte Carlo; *ML* =, Maximum Likelihood; *MVAR* =, Multivariate Autoregression; *RBM* =, Restricted Boltzmann Machine.

* Corresponding author.
E-mail address: cambria@ntu.edu.sg (E. Cambria).

http://dx.doi.org/10.1016/j.knosys.2016.07.019
0950-7051/

Notations

$x_i(\tau)$	Expression level of node i at time instant τ
$\boldsymbol{\theta}_i$	Parameters for node i in the Bayesian network
$\boldsymbol{a}_i$	Parent set of a node i
N	Number of variables in the system
v_i	Node i in the visible layer
h_j	Node j in the hidden layer
T	Number of data samples
r	index for order of delay
R	The upper-bound of delay
n_l	The number of nodes in the layer l
l	Index for a hidden layer
L	Total number of layers
f	Activation function of each hidden neuron
$g(x(\tau))$	Joint probability over all nodes at time instant τ
W_l	Weights of the hidden layer l
W_r	Weights of r-order edges among visible nodes
α	Learning rate of a DBN
S	Gaussian network

the past but also the future that is useful in sentence completion tasks [8]. However, RNN require additional memory neurons to model each time delay and training becomes difficult as the gradient declines sharply with increasing delay. This can also result in unstable convergence, as the Hessian matrix of second-order derivatives does not exist for many real datasets.

The main difference from the work done in [9] is that instead of resorting to Hessian free optimization to learn RNN with time-delays, we take cue from the fact that RNN are deep neural networks with weight sharing across time. Hence, we consider deep learning where hierarchies of modules can provide a compact representation to temporal features in the form of input-output pairs. Contrary to previous approaches, we propose a variable-order deep Belief Network (VBN) that uses a dynamic Gaussian Bayesian network and Markov Chain Monte Carlo (MCMC) sampling to reduce the dimensionality of temporal problems without any loss of information. This is achieved from our knowledge of time delays; we can learn short delays independent of long delays in a hierarchical manner, since the former is a part of the latter. Here, we train each additional hidden layer of neurons with recurrent motifs extracted from the time series data of increasing Markov order using dynamic Gaussian Bayesian networks.

In order to reduce the complexity of the model each hidden layer is a restricted Boltzmann machine (RBM) that is learned independent of the others. In particular, to train the proposed VBN, we extract time-delayed features in the form of dynamic network motifs from the original time series using MCMC. Fig. 1(a) illustrates a deep belief network where the input nodes are a connectivity matrix of width N; the maximum order of delay is R and there are L RBM layers. Given input vector **x**, each hidden neuron in the lth RBM layer is designed to learn weights W_{ij}^{l}. Fig. 1(b) illustrates l-order hidden neurons that learns from the inputs with up-to l-order time delays to arrive at the weights W_j^l of hidden neuron j. In contrast to previous methods of duplicating neurons to model time-delays, here dynamic Gaussian Bayesian networks over training motifs are evolved using Markov Chain Monte Carlo to perturb the initial weights of each hidden layer of neurons to include time-delays. VBN does not require any additional memory neurons as delays are cascaded. In this way they do not need to pass through the non-linearity at each time point and the loss in gradient is much lower than in the case of RNN.

For instance, the number of possible first-order time delayed features is exponential; hence if we initialize the weights of the hidden neurons of first RBM layer using high probability first-order motifs, it is then much easier to train using contrastive divergence. Similarly, for the second RBM layer we initialize the weights using high probability second-order motifs. Since, both the first and second-order network motifs are learned from the same training data, they will belong to the same distribution.

Further, as explained in [10,11], in such a layered model, the features or network motifs learned in the first layer become input to the second layer and so on. For example, the first RBM layer will learn network motif representations with only first-order time delays by minimizing the error between each initial motif and the corresponding motif predicted by the hidden neurons in the training data. Next, the training motifs are evolved to include second-order time delayed edges prior to training the second RBM layer. The first RBM layer will now try to emit each second-order time delayed edge as a cascade of two first-order time delayed edges occurring in different features that will become the input to the second RBM layer.

In such a hierarchy of predictors, the input at any given time at one level is coming from the previous level. Hence, it is sufficient to know those elements of the input data that were not correctly predicted. The error function of each module forces it to emit a learned target representation in the input data. If the module makes an error, the unpredicted input will be transformed to a unique representation and send to the next higher module. To our knowledge, such a framework that can combine small delays to model long delays via deep MCMC sampling has previously not been proposed.

2. Related work and contributions

Depending on the problem, learning in a neural network may require long causal chains of computational stages. To reduce the redundancy in the data and consequently depth of the model, unsupervised learning methods such as Boltzmann machines are used that maximize the entropy related information [12]. Unsupervised learning can automatically generate sparse representations of inputs using well-known feature detectors such as edge detectors or Gabor filters. From then on, only unexpected inputs (errors) convey new information and are fed to the next higher layer. For each individual input sequence, we get a series of less and less redundant encoding in deeper and deeper levels also known as history compression. In convolution NN, a filter of shared weights is shifted step by step over a 2D array of inputs resulting in massive weight sharing. Each convolution layer is inter-leaved with a max-pooling layer [13]. In max pooling, each convolution layer is replaced by a down sampling layer by the activation of its maximally active unit. By eliminating non-maximal values, it can reduce redundancies due to convolution.

Finally, Deep Belief Network (DBN) is a stack of Restricted Boltzmann Machines [14]. Each RBM takes as input the pattern representations from the level below and learns to encode them in an unsupervised fashion. Occam's razor suggests that a NN with low weight complexity corresponds to high NN accuracy without over fitting to training data. Each RBM layer results in a reduction in dimensionality of the input as long as the number of hidden neurons is lower than previous layer. Hence, the corresponding minimum description length of the data or the negative log probability of the data will keep improving [15]. Lastly, the DBN can be fine-tuned using back-propagation.

Computing the probability of an RBM is difficult, as the normalization constant requires summation over all possible configurations of the hidden neurons. In [16] the authors proposed a heuristic approach called Contrastive Divergence(CD) that tries to minimize the Kullback–Leibler divergence between the input samples and target distribution. Here, we use Gibbs sampling over each