Neurocomputing 298 (2018) 9–20

Contents lists available at ScienceDirect

Neurocomputing

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

Randomized neural networks for preference learning with physiological data

Davide Bacciu[a,*], Michele Colombo[a,b], Davide Morelli[a,b,c], David Plans[b,c]

[a] Dipartimento di Informatica, Università di Pisa, L.go B. Pontecorvo 3, 56127, Italy
[b] Biobeats Group Ltd, London, UK
[c] Center for Digital Economy, University of Surrey, Guildford GU2 7XH, UK

ARTICLE INFO

Article history:
Received 5 July 2017
Revised 20 October 2017
Accepted 11 November 2017

Keywords:
Randomized networks
Physiological data
Preference learning
Extreme learning machine
Echo state network

ABSTRACT

The paper discusses the use of randomized neural networks to learn a complete ordering between samples of heart-rate variability data by relying solely on partial and subject-dependent information concerning pairwise relations between samples. We confront two approaches, i.e. Extreme Learning Machines and Echo State Networks, assessing the effectiveness in exploiting hand-engineered heart-rate variability features versus using raw beat-to-beat sequential data. Additionally, we introduce a weight sharing architecture and a preference learning error function whose performance is compared with a standard architecture realizing pairwise ranking as a binary-classification task. The models are evaluated on real-world data from a mobile application realizing a guided breathing exercise, using a dataset of over 54 K exercising sessions. Results show how a randomized neural model processing information in its raw sequential form can outperform its vectorial counterpart, increasing accuracy in predicting the correct sample ordering by about 20%. Further, the experiments highlight the importance of using weight sharing architectures to learn smooth and generalizable complete orders induced by the preference relation.

1. Introduction

Randomization has recently gained attention as a guiding principle in the design of neural models as well as in learning algorithms, mostly motivated by considerations pertaining excellent trade-offs between computational complexity and predictive performance. Randomiz ation methods, in this respect, are used as an approach to network design, where parts of the neural model are randomly initialized (often subject to some theory-induced constraint) and left untrained, while the rest of the parameters can be conveniently fit by some efficient iterative algorithm or through a closed-form solution to the optimization problem. An excellent overview of the randomized neural network paradigm is given by two recent surveys [1,2]. Other forms of randomization can be found in the learning algorithms, for instance under the form of random noise injection in the input data or in the model parameters during training.

In this paper, we focus on the use of randomization as a network design method within a real-world application exploiting the constrained parameter space and architecture of such models to address a problem characterized by data which is large in sample size, very noisy and fast changing. Our application targets learning the correlation between Heart Rate Variability (HRV) information and the activity of the Autonomic Nervous System (ANS) [3]. HRV can be useful in understanding the psychophysiology of anxiety and emotion, and can thus help us investigate ANS disturbances [4] in humans, such as the effect and correlation between HRV changes and particularly psychosocial traits such as rumination (repetitive thought) [5] as well as its effect on cardiovascular health. However, discerning a direct correlation between HRV activity and ANS activation is a difficult problem, given the very subjective nature of HRV changes. As a result, typical supervised learning approaches fail to capture such a correlation [6].

An alternative way to tackle the problem is to address it as a preference learning task [6], where the learning model is provided with partial information according to some subjective preferential ordering between samples, e.g. the fact that two HRV samples from the same subject are taken under different ANS conditions. Such partial, pairwise information is then used to build a total ordering of the samples from all subjects according to the unknown total ordering function, i.e. the relaxation level. In this work, we discuss how the problem of learning preferential rankings can be effectively addressed by randomized architectures, by introducing a specialized preference learning objective function coupled with

* Corresponding author.
E-mail addresses: bacciu@di.unipi.it (D. Bacciu), michele@biobeats.com (M. Colombo), davide@biobeats.com (D. Morelli), d.plans@surrey.ac.uk (D. Plans).

https://doi.org/10.1016/j.neucom.2017.11.070
0925-2312/

a weight sharing architecture. We tackle the problem using two different representations for the input information, which require randomized neural models with different processing capabilities. One representation uses informative hand-engineered HRV features, resulting in data of vectorial form; the second is based on raw heart rate data of a sequential nature.

The literature on randomized neural models dates back to the early works on Rosenblatt's Perceptron [7], where the synthetic retinal units are randomly connected to the associator units in a many-to-many fashion. In the early 1990s, different authors [8,9] independently proposed feedforward neural networks comprising randomly initialized and untrained connections between the input layer and a hidden layer of non-linear neurons, focusing on learning only the hidden-to-output synaptic connections. Such an approach has been later re-discovered by the Extreme Learning Machine (ELM) [10] model which, in the intention of its authors, is expected to provide a unifying framework for random feedforward neural networks (single or multi-layered) as well as a proof of their universal approximation capabilities which, in fairness, builds strongly on well-known results for Radial Basis Functions networks [11]. ELMs are accompanied by non negligible criticism of part of the scientific community concerning their actual novelty and a certain tendency towards self-reference [12]. Without entering in the quarrel, in the remainder of the paper, we will consider ELM as a reference for randomized feed-forward neural networks. Nevertheless, an interested reader can find in [2] an interesting survey of randomized neural networks which presents an alternative formulation bypassing references to the ELM conceptualization. The key point of randomized feed-forward networks is that the random, untrained nature of the hidden neurons makes training computationally efficient, e.g. when dealing with Big Data [13], as well as fairly easy to implement, while also simplifying human intervention concerning network design choices. As such, there as been much interest on these models in the recent literature, with several extensions of the original model. A thorough review of all these models is outside of the scope of this paper, but [14] provides an overview of the most recent advancements. The idea underlying feedforward randomized networks is tightly connected with that of random projections [15] and kernel-based methods, as ELM create a high dimensional basis expansion of the input through the random hidden layer, followed by a simple (possibly linear) adaptive model. In [16], a solution is proposed that further merges the two approaches by defining an efficient kernel based on ELM that is then used within a Support Vector Machine (SVM).

The randomized neural models discussed so far are primarily feedforward, hence they do not have the capability of encoding the history of the input signal, making them less apt to be used with sequential information. The Reservoir Computing (RC) [17,18] paradigm is somehow the recurrent equivalent of the ELM model, providing a framework for randomized neural networks with recurrent connectivity. Similarly to its feedforward counterpart, the RC approach is founded on a conceptual separation between a randomly and sparsely connected layer of (typically) untrained recurrent neurons, i.e. the reservoir, and the output neurons with trainable connectivity, which perform the predictions. Liquid State Machines (LSM) [19] are a popular RC model using spiking activation functions for biological plausibility that are characterized by randomly fixed input and reservoir weights, complemented by output weights that can be trained by recursive least-square optimization. The Echo State Network (ESN) [20] has the same partition into random and trainable weights as the LSM, but uses sigmoid-like activations in the reservoir and simple linear models, trainable by least square minimization, in the output neurons. ESN models have found wide application in the processing of time-series and sequential data, thanks to a predictive performance which often matches (if not outperforms) that of fully adaptive neural models, coupled with the advantage of using a simple and more efficient training procedure [21]. For these reasons, they have also found applications in a number of tasks associated with sensor data processing, such as ambient assisted living [22,23]. Very recently, randomized neural models have also found application in Deep Learning architectures, including deep random projection models for image classification [24] and hierarchical extensions of the ESN with multiple reservoir layers [25].

For the remainder of this paper, we will focus on the ELM and ESN models, respectively for vectorial and sequential HRV data, assessing them in the context of learning preferential rankings from only pairwise comparison information and evaluating the effect of weight sharing architectures on the quality of the reconstructed ranking. Both models have been chosen for their adequacy for the task as well as for their popularity within the community. Recent work has proposed either hybrids of the two approaches [26] or a unifying framework capable of representing both models throughout a non-linear time-delayed neuron formulation [27]. Our experimental assessment exploits heart-rate samples acquired by Biobeats'[1] app *Hear and Now.*[2] The app offers diaphragmatic breathing coaching to its users, guiding them through the same breathing technique described in [28]. In doing so, it captures 40 s of heart-rate information before and after each exercise by implementing a simple form of photoplethysmography using the smartphone camera, and asking the user to keep the index finger over the camera during the exercise.

Part of this work has already appeared in a conference paper in [29]. Here we consistently extend such work with a completely renewed experimental assessment comprising three-times the amount of data in the original paper. More importantly, in this work, we also use information in raw sequential form, introducing what we believe to be the first ESN approach to preference learning (in [29], instead, we have only considered ELM models applied to HRV vectors). The source code of both ELM and ESN models for all the preference learning architectures discussed in the paper is freely available on an open repository.[3]

2. Randomized neural networks for preference learning

2.1. Randomized neural networks for vectorial and sequential data

Before delving into the specifics of the preference learning task, we provide a brief overview of the two randomized neural models used to analyse the vectorial and sequential data used in our application scenario. Both models are based on the concept of building a high dimensional basis expansion of input information through one (or more) hidden layers of randomly initialized and non-plastic neurons, followed by an output layer of plastic neurons which are adapted to provide predictions in accordance to the task at hand.

We adopt ELMs [10] as a reference model for processing HRV feature vectors. The simplest form of ELM network, depicted in Fig. 1, comprises a hidden layer of sigmoid-like neurons which receive the data samples $\mathbf{u}$ as input, and which are connected to an output layer of linear neurons whose number depends on the task at hand. Each output neuron realizes a weighted combination of the hidden neuron out puts to compute the network predictions $\mathbf{y}$ that, in vectorial form, write as

$$\mathbf{Y} = \mathbf{W}_{out}\mathbf{H} \tag{1}$$

where $\mathbf{W}_{out}$ is the $N_{out} \times N_h$ matrix of linear combination weights from the N_h hidden neurons to the N_{out} output neurons, while $\mathbf{H}$

[1] http://www.biobeats.com.

[2] https://itunes.apple.com/us/app/hear-and-now/id977650202.

[3] https://github.com/m-colombo/neurocom-randomized-preference-learning.