Information Sciences 000 (2015) 1–12

Contents lists available at ScienceDirect

Information Sciences

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

A comprehensive evaluation of random vector functional link networks

Le Zhang, P.N. Suganthan*

School of Electric and Electronic Engineering, NanYang Technological University, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 14 May 2015
Revised 4 September 2015
Accepted 11 September 2015
Available online xxx

Keywords:
Random vector functional link networks
Ridge regression
Moore–Penrose pseudoinverse
Data classification

ABSTRACT

With randomly generated weights between input and hidden layers, a random vector functional link network is a universal approximator for continuous functions on compact sets with fast learning property. Though it was proposed two decades ago, the classification ability of this family of networks has not been fully investigated yet. Through a very comprehensive evaluation by using 121 UCI datasets, the effect of bias in the output layer, direct links from the input layer to the output layer and type of activation functions in the hidden layer, scaling of parameter randomization as well as the solution procedure for the output weights are investigated in this work. Surprisingly, we found that the direct link plays an important performance enhancing role in RVFL, while the bias term in the output neuron had no significant effect. The ridge regression based closed-form solution was better than those with Moore–Penrose pseudoinverse. Instead of using a uniform randomization in [−1,+1] for all datasets, tuning the scaling of the uniform randomization range for each dataset enhances the overall performance. Six commonly used activation functions were investigated in this work and we found that *hardlim* and *sign* activation functions degenerate the overall performance. These basic conclusions can serve as general guidelines for designing RVFL networks based classifiers.

1. Introduction

Single layer feedforward neural networks (SLFN) have been widely applied to solve problems such as classification and regression because of their universal approximation capability [14,17,20,31]. Conventional methods for training SLFN are back-propagation based learning algorithms [7,10]. These iterative methods suffer from slow convergence, getting trapped in a local minimum and being sensitivity to learning rate setting. Random Vector Functional Link Networks (RVFL), shown in Fig. 1, which is a randomized version of the functional link neural network network [8,25], shows that actual values of the weights from the input layer to hidden layer can be randomly generated in a suitable domain and kept fixed in the learning stage. Independently developed method in [35] also belongs to the family of randomized methods for training artificial neural networks with randomized input layer weights. This method [35] does not have direct links between the inputs and the outputs whereas RVFL has highly beneficial direct links.

RVFL was proposed in [28]. Learning and generalization characteristics of RVFL were discussed in [26]. In [17], Igelnik and Pao proved that the RVFL network is a universal approximator for a continuous function on a bounded finite dimensional set

* Corresponding author. Tel.: +65 670 5404; fax: +65 6793 3318.
E-mail address: epnsugan@ntu.edu.sg (P.N. Suganthan).

http://dx.doi.org/10.1016/j.ins.2015.09.025
0020-0255/

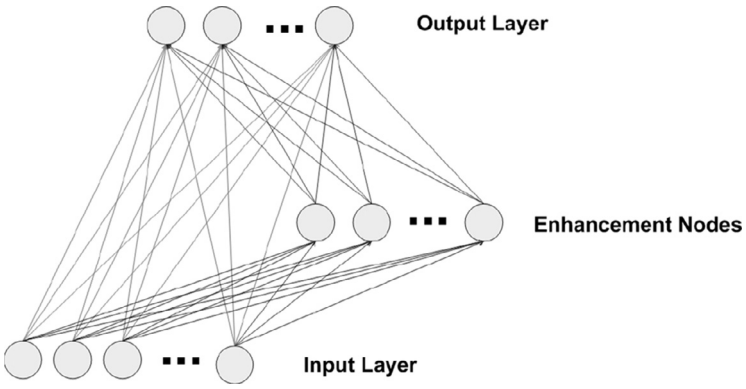

Fig. 1. The structure of RVFL. The input features are firstly transformed into the enhanced features by the enhancement nodes. Input weights and biases of the enhancement nodes are randomly generated. At the output layer, all the enhanced and original features are concatenated and fed into output neurons.

with a closed-form solution. From then on, RVFL has been employed to solve problems in diverse domains. A dynamic step-wise updating algorithm was proposed to update the output weights of the RVFL on-the-fly in [5] for both a new added pattern and a new added enhancement node. The RVFL network was investigated in [37] in the context of modeling and control. They [37] suggested to combine unsupervised placement of network nodes to the input data density with subsequent supervised or reinforcement learning of the linear parameters of the approximator. Modeling conditional probabilities with RVFL was reported in [15].

RVFL can also be combined with other learning methods. In [6], RVFL was combined with statistical hypothesis testing and self-organization of a number of enhancement nodes to generate a new learning system called a statistical self-organizing learning system (SSOLS) for remote sensing applications. In [16], expectation maximization was combined with RVFL to improve its performance. RVFL has also been investigated in ensemble learning framework. In [1], decorrelated RVFL ensemble was introduced based on the negative correlation learning. RVFL based multi-source data ensemble for clinker free lime content estimation in rotary kiln sintering processes can be found [21]. RVFL has also been widely applied to solve real-life problems. In [30], the authors reported the performance of a holistic-styled word-based approach to off-line recognition of English language script. Radial basis function neural net and RVFL were combined. Their approach, named as density-based random-vector functional-link net (DBRVFLN), was helpful in improving the performance of the word recognition. In [29], RVFL was used in MPEG-4 coder. In [38] RVFL was applied for pedestrian detection based on combination of multi-feature. In [39], RVFL was combined with Adaboost in the pedestrian detection system. In [23], the authors investigated the performance of hardware implementation methods for RVFL. In [34], distributed learning of RVFL was proposed where training data is distributed under a decentralized information structure.

Consider an RVFL as demonstrated in Fig. 1. As mentioned before, the weights a_{ij} from the input to the enhancement nodes are randomly generated such that the activation functions $g(a_j^T x + b_j)$ are not all saturated. Following the approach in [1], all the weights are generated with the a uniform distribution within $[-S, +S]$ in this work, where S is a scale factor to be determined during the parameter tuning stage for each dataset. For RVFL, only the output weights β need to the determined by solving the following problem:

$$t_i = d_i^T \beta, \quad i = 1, 2, \ldots, P \tag{1}$$

where P is the number of data samples, t is the target and d is the vector version of the concatenation of the original features as well as the random features.[1] Directly solving the problem in Eq. (1) may lead to over-fitting. In practice, a regularization on the solution such as regularized least square or preference of the solution with smaller norm [3] can be adopted to obtain the solution. RVFL can be roughly divided into 2 classes based on the algorithm to obtain the output weights. One is iterative RVFL, which obtains the output weights in an iterative manner based on the gradient of the error function. The other one is closed-form based RVFL, which obtains the output weights in a single-step. The present work focuses on the closed-form based RVFL because of its efficiency. A straightforward solution within a single learning step can be achieved by the pseudo-inverse [17,27], among which Moore–Penrose pseudoinverse, $\beta = D^+T$, where D and T are the matrix versions of the features and targets by stacking the features and targets of all data samples, is most commonly used. Another alternative is the $L2$ norm regularized least square (or ridge regression), which solves the following problem:

$$\sum_i (t_i - d_i^T \beta)^2 + \lambda \|\beta\|^2; i = 1, 2, \ldots P. \tag{2}$$

The solution is given by $\beta = D(D^T D + \lambda I)^{-1} T$, where λ is the regularization parameter to be tuned.

[1] For notational simplicity, we use the same formulation for all cases no matter whether there are biases in the output neurons since representing the features for the output neurons with $d = [d, 1]$ is equivalent to having a bias term in the output neurons.

Please cite this article as: L. Zhang, P.N. Suganthan, A comprehensive evaluation of random vector functional link networks, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.09.025