Neural Networks 52 (2014) 33–42

Contents lists available at ScienceDirect

Neural Networks

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

# Hybrid extreme rotation forest

Borja Ayerdi, Manuel Graña *

*Computational Intelligence Group, UPV/EHU, Spain*

ARTICLE INFO

*Article history:*
Received 14 July 2013
Received in revised form 16 December 2013
Accepted 3 January 2014

*Keywords:*
Extreme learning machines
Ensembles of classifiers
Rotation forest

ABSTRACT

This paper proposes the Hybrid Extreme Rotation Forest (HERF), an innovative ensemble learning algorithm for classification problems, combining classical Decision Trees with the recently proposed Extreme Learning Machines (ELM) training of Neural Networks. In the HERF algorithm, training of each individual classifier involves two steps: first computing a randomized data rotation transformation of the training data, second, training the individual classifier on the rotated data. The testing data is subjected to the same transformation as the training data, which is specific for each classifier in the ensemble. Experimental design in this paper involves (a) the comparison of factorization approaches to compute the randomized rotation matrix: the Principal Component Analysis (PCA) and the Quartimax, (b) assessing the effect of data normalization and bootstrapping training data selection, (c) all variants of single and combined ELM and decision trees, including Regularized ELM. This experimental design effectively includes other state-of-the-art ensemble approaches in the comparison, such as Voting ELM and Random Forest. We report extensive results over a collection of machine learning benchmark databases. Ranking the cross-validation results *per* experimental dataset and classifier tested concludes that HERF significantly improves over the other state-of-the-art ensemble classifier. Besides, we find some other results such as that the data rotation with Quartimax improves over PCA, and the relative insensitivity of the approach to regularization which may be attributable to the *de facto* regularization performed by the ensemble approach.



## 1. Introduction

*Basic classifiers*. Extreme Learning Machines (ELM) (Huang, Zhu, & Siew, 2006) are a quite successful proposition for the fast training of single layer feedforward neural networks (SFLN), providing good quality classification and regression results with orders of magnitude less computation time than the conventional backpropagation training. Since their proposal, ELM have been applied to a large number of problems such as image deblurring (Wang, Huang, Luo, Wang, & Luo, 2011), location estimation from wifi information (Liu, Chen, Liu, & Zhao, 2011), handwritten character recognition (Chacko, Vimal Krishnan, Raju, & Anto, 2012), and face recognition (Marques & Graña, 2012). However, a major criticism to ELM is their sensitivity to the random generation of the hidden layer weights. To overcome it, some regularization approaches have been proposed, such as pruning of an overparameterized ELM (Rong, Ong, Tan, & Zhu, 2008) based on the weight statistical significance. The incremental ELM (Huang & Chen, 2007) follows the opposite approach, adding new hidden units until learning converges to its optima. Combining both points of view, Zhao, Wang, and Park (2012) performs incremental learning, but with a forgetting mechanism that allows to prune redundant hidden units. Decision trees (Breiman, Friedman, Olshen, & Stone, 1984; Quinlan, 1993) are a classical classifier building approach. They are very fast to train and can provide good results. However, their training processes are very unstable, meaning that small variations in the presentation of the data may produce drastically different classifiers.

*Ensembles of classifiers*. Ensembles of classifiers aim to obtain improved classification results by the combination of weak and diversified classifiers (Oza & Tumer, 2008). The Random Forest combines by majority voting the outputs of a collection of Decision Trees (DT) (Breiman, 1996, 2001) built from bootstrapped training data over random variable selections. The intuition that diversified collections would lead to improved collective performance has driven research in approaches performing data preprocessing, such as the randomized rotation matrices applied in Rotation Forests (Rodriguez, Kuncheva, & Alonso, 2006), and the computation of supervised projections (García-Pedrajas & García-Osorio, 2011). The rationale behind these approaches is the conjecture that classifiers with quite different fields of expertise may complement each other, giving enhanced results of the ensemble as a whole. Bootstrapping and random rotations/projections of the data are the source of diversity among the elementary classifiers. Construction of ensembles have used many other basic classifiers besides decision trees, such as Support Vector Machines (SVM) (Kim, Pang,

* Corresponding author. Tel.: +34 943018000.
*E-mail address:* manuel.grana@ehu.es (M. Graña).

0893-6080/$ – see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.neunet.2014.01.003

Je, Kim, & Bang, 2003). Recent works report ensembles whose elementary classifiers are trained by some variant of ELM, which will be reviewed in the Related Works section, include the Voting ELM (V-ELM) (Cao, Lin, Huang, & Liu, 2012), the ensembles of Online Sequential ELMs (EOS-ELM) (Lan, Soh, & Huang, 2009; Zhao et al., 2012), and the Multiple ELM approach (Lin, Chang, & Hsu, 2013).

*Contributions and motivation*. The contribution of this paper is the experimental validation of a new hybrid rotation ensemble composed of decision trees and ELMs, which we call Hybrid Extreme Rotation Forest (HERF). The motivation of this new proposal is to further enhance the classifier diversity (Wozniak, Graña, & Corchado, 2014), profiting from the fast training of both Decision Trees and ELM. To assess the HERF classifiers, this paper provides an extensive experimental exploration of the results of homogeneous and hybrid ensembles of Decision Trees and ELMS following several data preprocessing such as $z$-score data normalization, random rotations performed either with the Principal Component Analysis (PCA) or the Quartimax factorial analysis, and bootstrapping of the train dataset. The combinatorial design of this exploration includes the Random Forest (ensembles of bootstrapped decision trees), the Rotation Forest (ensembles of decision trees trained on randomized PCA rotated data), and V-ELM (Cao et al., 2012) (ensembles of ELMs trained on unprocessed data). The proposed HERF performs a random rotation of the data and the elementary classifiers can be either DT or SFLN trained with ELM. Statistical tests assessing the significance of the results are also reported. Overall, we have found that HERF with a Quartimax random rotation provides the best overall results.

The paper is structured as follows: Section 2 reviews recent related works. Section 3 recalls the basic definitions of the classifiers used in this work. Section 4 describes the randomized rotation approach, recalling the definition of PCA and Quartimax data rotations. Section 5 gives the description of the HERF. Section 6 discusses the experimental design followed the validation of the proposed HERF. Section 7 gives the experimental results and a discussion of their significance. Section 8 gives our conclusions.

## 2. Related works

The definition of ensemble classifiers based on ELM has a number of representatives in the literature. The Voting ELM (V-ELM) (Cao et al., 2012) is a direct composition by majority voting of a collection of ELMs trained independently. The ensemble approach is shown to enhance the stability of ELM results, by approaching the limit perfect classification performance when the number of training samples grows sufficiently. The V-ELM has been empirically proven to improve over ELM in a number of experiments, and it is one of the ensemble architectures tested in this paper.

The ensembles of Online Sequential ELMs (EOS-ELM) (Lan et al., 2009) are intended to provide stability and reproducibility of results of the single OS-ELM classifier. The approach is applied to regression problems, combining the output of the individual OS-ELM regressors by their average. The underlying justification is that the EOS-ELM is closer to the expected values of the true output than the individual OS-ELM. Model selection is performed empirically. It is claimed that it trains faster and with better accuracy. The addition of a forgetting mechanism improves the EOS-ELM in Zhao et al. (2012) in situations where the lifetime of the data is limited. Such situations happen in the financial market applications, and similar prediction tasks. The FOS-ELM discards the outdated data samples in the training process by a filtering process that weights the prediction value of each datum.

The Multiple ELM (MELM) approach (Lin et al., 2013) combines specific sampling and feature selection methods with an ensemble of ELMs to improve prediction in highly imbalanced datasets. The process of the data starts with the SMOTE algorithm for sample generation in order to correct the dataset imbalance. Then feature selection is performed on the basis of the information gain. The MELM is trained on the dataset to play the role of knowledge repository for the last element of the architecture, which is a DT trained on the MELM outputs. The DT is used as a set of rules allowing interpretation of the system reasoning. The approach has been applied to predict the company decline.

Evolutionary algorithms have been proposed to improve the stability of learning ensembles of ELMs (Wang & Alhamdoosh, 2012). The evolutionary algorithm is guided by the diversity of classifier outputs as the fitness function. The approach is computationally very heavy as it involves the training of a large number of ELM ensembles in order to produce the desired evolution, with the advantage of providing a tuned ensemble size which can be of interest in some specific applications.

Regarding efficient implementations, ensembles of ELM are easily implemented in parallel architectures of GPU processing units (van Heeswijk, Miche, Oja, & Lendasse, 2011). The approach allows to perform the training of individual ELMs in parallel, as well as the reutilization of past training on the composition of new ensembles. The parallelization is generalizable to any kind of ELM architecture. A distributed implementation of OS-ELM ensembles reported in Sun, Yuan, and Wang (2011) achieves high computational time reductions with little loss of accuracy. The distributed implementation is realized on a hierarchical P2P network, profiting from the incremental learning principle of OS-ELM, allowing two kinds of architectures: one-by-one and parallel ensemble classifications. The system design includes a novel indexing structure for peer selection.

## 3. Elementary classifiers

### 3.1. Decision trees and random forests

Decision Trees (DT) (Breiman et al., 1984; Quinlan, 1993) are built by recursive partitioning of the data space. A univariate (single attribute) split is recursively defined for each tree node, from the root to the leaves, of the tree using some criterion (e.g., mutual information, gain-ratio, gini index). The data space and data samples are then partitioned according to the univariate test. Tree leaves correspond to the probabilistic assignment of data samples to classes. Often, a pruning process is applied in order to reduce the tree complexity and overfitting to the training data.

In order to improve generalization properties of DT classifiers, ensembles of classifiers have been proposed, which include Bagging (Breiman, 1996) and Random Forests (Breiman, 2001). These approaches try to build improved classifiers combining several weak classifiers. Random Forests are DT ensembles where each individual DT is built on a bootstrapped training data subset and on a random subset of the input variables. The majority voting rule applied to the ensemble of outputs decides the input data class assignment.

The Rotation Forest (Rodriguez et al., 2006) tries to enhance the diversity of individual DT applying a randomized rotation transformation matrix to the training data before the training of each individual DT. The process starts with a random partition of the input variables. Separate PCA rotation matrices are computed for each subset of input variables, which are composed into a single rotation matrix by interleaving the separate rotation matrix columns according to the original order of the feature variables.