Information Fusion 39 (2018) 41–52

Contents lists available at ScienceDirect

Information Fusion

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

On building ensembles of stacked denoising auto-encoding classifiers and their further improvement

Ricardo F. Alvear-Sandoval*, Aníbal R. Figueiras-Vidal

GAMMA-L+/DTSC, Universidad Carlos III de Madrid, Spain

ARTICLE INFO

Article history:
Received 31 January 2017
Revised 15 March 2017
Accepted 26 March 2017
Available online 28 March 2017

Keywords:
Augmentation
Classification
Deep
Diversity
Learning
Pre-emphasis

ABSTRACT

To aggregate diverse learners and to train deep architectures are the two principal avenues towards increasing the expressive capabilities of neural networks. Therefore, their combinations merit attention. In this contribution, we study how to apply some conventional diversity methods –bagging and label switching– to a general deep machine, the stacked denoising auto-encoding classifier, in order to solve a number of appropriately selected image recognition problems. The main conclusion of our work is that binarizing multi-class problems is the key to obtain benefit from those diversity methods.

Additionally, we check that adding other kinds of performance improvement procedures, such as pre-emphasizing training samples and elastic distortion mechanisms, further increases the quality of the results. In particular, an appropriate combination of all the above methods leads us to reach a new absolute record in classifying MNIST handwritten digits.

These facts reveal that there are clear opportunities for designing more powerful classifiers by means of combining different improvement techniques.

1. Introduction

The number of available training samples limits the expressive capability of traditional one-hidden layer perceptrons, or shallow Multi-Layer Perceptrons (MLPs), for practical applications, in spite of their theoretically unbounded approximation capacities [1,2]. Consequently, a lot of attention is being paid to architecture and parameterization procedures that allow to improve their performance when solving practical problems. The most relevant procedures increase the number of the trainable weights following two main avenues: Building ensembles of learning machines, or constructing Deep Neural Networks (DNNs).

Most of the advances in DNN design correspond to the last decade. In fact, prior to 2006 the only successfully used DNNs were the Convolutional Neural Network (CNN) classifiers [3], whose significantly simplified structure makes possible their training by means of conventional algorithms such as Back Propagation (BP). This architecture is appropriate for some kinds of applications, those in which the samples show translation-invariant characteristics, for example, image processing. But direct training of general form DNNs remained without solution because the appearance of vanishing or exploding derivatives [4,5]. In 2006 Hinton et al. [6] proposed a deep classifier indirect design that involved the stacking of Reduced Boltzmann Machines (RBMs) [7]. A top layer task-oriented training and overall refining complete these classifiers, that have come to be called Deep Belief Machines (DBMs). Contrastive divergence algorithms allow for the training of the DBMs without a huge computational effort [8].

Some years later, Vincent et al. [9] introduced a similar procedure consisting of an expansive, denoising auto-encoder layer-wise training plus the final top classification and refining. These are the Stacked Denoising Auto-Encoder (SDAE) classifiers. It is worth mentioning that both DBMs and SDAEs are representation machines [10], i.e., their hidden layers provide more and more sophisticated high-level feature representations of the input vectors. These representations can be useful for analysis purposes [11], and, even more, the representation process induces a disentangling of the sub-spaces in which the samples appear [12]. On the other hand, another sequentially trainable deep architecture, the Deep Stacking Networks (DSNs), was introduced in [13,14], following the idea of training shallow MLPs and adding their outputs to the input vector for training further units. Finally, a number of modifications that reduce the difficulties with the derivatives have been proposed for training directly DNNs, such as data conscious initializations [15], Hessian-free search [16], mini-batch iterations [17], non-sigmoidal activations [18], and adding scale and location trainable parameters [19].

There are also proofs of universal approximation capabilities for DNNs [20,21], as well as of some interesting characteristics of them

* Corresponding author.
E-mail address: ralvear@tsc.uc3m.es (R.F. Alvear-Sandoval).

http://dx.doi.org/10.1016/j.inffus.2017.03.008
1566-2535/

[22]. The analyses in [23,24] show that by adding layers to a network, it is possible to reduce the effort to establish input-output correspondences. In practice, DNNs have offered excellent performance results in many applications, therefore, one can conclude they are important in spite of the large number of parameters that need to be learned. There is not room here to give more details, so, the interested reader is referred to excellent tutorials [4,5,25] for more extensive reviews and bibliography, as well as to [26] for a bibliography of applications.

Ensembles are the second option to effectively increase the expressive capability of learning machines, including MLPs. They are built by means of training learners that consider the problem to be solved from different perspectives, i.e., under a principle of diversity, and aggregating their outputs to obtain an improved solution. We present a very concise overview of ensembles, emphasizing just the design methods that we will use in our experiments, in Section 2, for the sake of continuity in this Introduction.

Since both diversity and depth increase the expressive power of MLPs but through very different mechanisms, it seems reasonable to expect that a combination of them would lead to an even better performance. However, there are a moderate number of contributions along this research direction. We will briefly revise them in Section 4, but we anticipate that some difficulties appear when trying to apply the usual ensemble building methods to multi-class problems, and, consequently, most of the DNN ensembles are constructed by means of "ad hoc" procedures.

In this paper, we explore and discuss in detail how and why diversification can be applied to DNNs, as well as if including other improvement techniques gives additional advantage. The objective is to evaluate if it is possible to get significant advantages by combining diversification and deep learning, as well as other techniques.

Of course, we have to select both DNN architectures and classification problems for our experiments and subsequent analysis. Although most of the previous studies with the databases we will use have considered CNNs, we have decided to work with a less specific architecture to exclude the possibility of obtaining conclusions only valid for this particular form of DNN and the kind of problems that are appropriate for it. So, we select SDAE classifiers, and in particular the SDAE-3 design that is introduced in [9]. However, at the same time and just to show the potential of combining diversity and depth, we will address some traditional image classification tasks –also included in [9],– that are more appropriate for CNN architectures. The selected problems for our experiments will be the well-known 10-class handwritten digit MNIST database [3], its version with a smaller training set MNIST-BASIC [9], in order to analyze the relevance of the weak or strong character of the SDAE-3 classifiers, and also the binary database RECTANGLES [9], with the objective of studying the origin of the difficulties for creating ensembles of multi-class DNNs. We emphasize that these selections are not arbitrary: There are many published results for MNIST, for example in [27,28], and clearly established records for representation DNNs, a 0.86% error rate [10], and for CNN ensembles [29], a 0.21% error rate. We anticipate from now that, with the help of a boosting-type training reinforcement or pre-emphasis, and a simple data augmentation besides of the binarization and training diversification we will apply, we arrive to a new absolute performance record, a 0.19% error rate. We repeat that this record was not our objective, but we looked for a better understanding of how to combine diversity and depth, and to avoid conclusions only valid for particular situations –using CNNs for image problems,– we select both SDAEs and the databases. Thus, in our opinion, there is no reason to think that our conclusions are problem- or architecture-dependent.

The rest of the paper is structured as follows. In Section 2, we present brief overviews of machine ensembles, both general forms and those that come from binarizing multi-class problems. We dedicate Section 4 to list and comment previously published works in designing DNN ensembles. Section 5 describes the additional techniques –pre-emphasis and data augmentation– we will use in the second part of our experiments. The experimental framework is detailed in Section 6: Databases, deep learning units, diversification and binarization techniques, and pre-emphasis and data augmentation forms. The results of the experiments appear in Section 7 following a sequential order, plus the corresponding discussions. Finally, the main conclusions of this work and some directions for further research close the paper.

2. Ensembles

To build an ensemble of diverse machines and aggregate their outputs is a way to increase expressive power. Although the first ideas on it were published half a century ago [30], they have been mainly developed along the last two decades. In the following, we briefly review some ensemble techniques, including those we will use to diversify SDAEs. We dedicate separate sub-sections to designs that introduce diversification by means of architecture or training differences, that we will call conventional ensembles, and to ensembles that come from transforming a multi-class problem in a number of binary classifications from which the resulting class can be obtained. Since a complete review of ensembles is beyond the scope of this paper, the reader is referred to monographs [31–34], as well as to tutorial article [35], which includes interesting perspectives on ensemble applications.

2.1. Conventional ensembles

Conventional diversification methods may be broadly classified into two categories. The first are those approaches that independently train a number of machines, usually with different training sets. These machines, or learners, can also have different structures. After it, learners' outputs are aggregated –typically with simple, non-trainable procedures– to come up with the final classification. These ensembles are called committees.

Among committees, Random Forests (RFs) [36] are very popular because they offer a remarkable performance. They diversify a number of tree classifiers by means of probabilistic branching, which can be combined with sub-space projections. There are other committees that can be applied to general types of learners, requiring only that they are unstable: Bagging [37] and label switching [38,39]. We will include both of them in our experiments because they are simple to implement and provide high expressive power, clearly improving the performance of a single machine. Yet we announce that the first experimental results will lead us to focus on the second.

Bagging ("Bootstrap and aggregating") produces diversity by training the ensemble learners with bootstraping re-sampled versions of the original training set and, then, aggregating these learners' outputs, usually by averaging them or with a majority vote. Bootstrap is a random sampling mechanism which includes replacement to permit arbitrary sizes of the re-sampled population. Although its primitive form used bootstrapped sets of the same size as the true training set, to explore the size of these bootstrapped sets is important to find a good balance between computational effort and number of learners and ensemble performance, because in some cases the reduction of the true samples that each learner sees can provoke losses. On the other hand, label switching changes the labels of a given portion of the training samples according to some stochastic mechanism. We will employ the simplest version, for which these changes appear purely at random. The switching rate must be explored when designing these committees.