Information Sciences 294 (2015) 139–151

Contents lists available at ScienceDirect

Information Sciences

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

Achieving a compromise between performance and complexity of structure: An incremental approach

Patrick Marques Ciarelli *, Elias Oliveira

Universidade Federal do Espírito Santo – UFES, Av. Fernando Ferrari, 514, 29075-910 Vitória, ES, Brazil

ARTICLE INFO

Article history:
Received 18 October 2013
Received in revised form 5 September 2014
Accepted 13 September 2014
Available online 7 October 2014

Keywords:
Incremental learning
Probabilistic Neural Network
Expectation Maximization

ABSTRACT

In incremental learning techniques, learning occurs continuously over time and does not cease once available data have been exhausted. Such techniques are useful in cases where problem data may be acquired in small quantities over time. This paper presents an incremental neural network called the evolving Probabilistic Neural Network. The main advantage of this technique lies in its adaptive architecture, which adjusts to data distributions. This method requires that each training sample be used only once throughout the training phase without being reprocessed. The technique is flexible and offers a simplified structure while maintaining performance levels comparable to those of other techniques. Experiments were conducted using publicly available benchmark data sets. These experiments show that overall, the proposed model achieves a quality of response that is comparable to those of the best techniques evaluated, and its structure size and classification time were as low as those of less complex techniques. These results indicate that the proposed model achieves a satisfactory balance between efficiency and efficacy.

1. Introduction

Machine learning techniques have traditionally assumed the presence of a sufficiently representative amount of training data so that future data generate the same distribution while the learning phase ceases after training data processing [17]. However, in several practical applications, when data arrive over a period of time or when storage capacities are very limited, these techniques fail to work effectively [38]. These conditions lead to at least three drawbacks [20]: technique retraining, which is often computationally expensive; determining when a learned model is not useful and must be substituted; and determining which data should be used to obtain a new model for performing classification tasks. An incremental learning approach can be used to avoid these problems.

In incremental learning techniques, learning occurs continuously over time and does not cease once available data have been exhausted. Such techniques are useful in situations where problem data are acquired in small quantities over time and when task characteristics are also subject to change over time [17]. These situations arise during many real-world tasks, and thus machine learning techniques that apply incremental learning have been applied to several applications, such as recommender systems that adapt dynamically [30], causality models for events built from data streams [1], robots that learn dynamic models incrementally [16] and adaptive mechanisms of spam filtering [13].

* Corresponding author.
E-mail addresses: pciarelli@lcad.inf.ufes.br (P.M. Ciarelli), elias@lcad.inf.ufes.br (E. Oliveira).

http://dx.doi.org/10.1016/j.ins.2014.09.013

Several incremental approaches have been presented in the literature [6,27,28,33]. However, given the large amount of data needed for long-term training, these techniques tend to produce models with complicated structures, rendering them unfeasible for accomplishing certain tasks. Hence, structure complexity grows with an increase in the amount of training data.

Three assumptions are often implicit in researches on incremental learning [23]. First, an algorithm should be able to apply acquired knowledge to perform a task at any learning stage. Second, the addition of knowledge throughout the learning process should be computationally efficient. Third, the learning process should not require an exorbitant amount of space, as memory requirements increase as a function of the number of training samples involved.

Based on these assumptions, Langley [23] defines incremental learning as a learning process in which only one sample is used for training at a time, samples used for training are not reprocessed, and in which techniques only retain knowledge of a structure such that it is not possible to directly return to previous knowledge states because they are not retained in memory. According to this definition, techniques such as Multi-Layer Perceptron (MLP), in which the samples are reprocessed numerous times, and k Nearest Neighbor (kNN), which involves storing all knowledge structures, cannot be considered incremental learning techniques.

This paper presents a model of incremental supervised learning for use in classification problems. This model exhibits a simplified structure while maintaining a level of performance comparable to those of classical classification techniques. The proposed model, called the evolving Probabilistic Neural Network (*e*PNN), possesses an evolving architecture (in other words, an architecture that changes in response to new training data) that adapts to classification problems, and training samples are used only once for training purposes, and are not reprocessed. The size of the model's architecture is continuously controlled with each training iteration; while the architecture can increase in size, it is reduced whenever possible. Therefore, the model's structure size does not typically grow significantly over time, as is the case for other incremental techniques.

The neural network presented in this paper tends to exhibit a simplified architectural structure, thus limiting computational costs and the need for memory resources, and the training phase is fast because each training sample is used only once throughout the training phase without being reprocessed. *e*PNN is thus recommended for applications in which a flexible neural network that includes incremental learning does not require large memory capacities. Moreover, *e*PNN may be used in applications for which it is not possible to store excessive data for retraining purposes (as needed for some classical techniques) and in which training and classification rates for the technique must be efficient. Furthermore, the *e*PNN algorithm presents characteristics that satisfy Langley's definition [23] of incremental learning techniques. The *e*PNN approach presented in this paper is an improved version of the method presented in [8].

Experiments performed on several data sets and using incremental and non-incremental techniques have shown that *e*PNN exhibits performance levels similar to those of the best techniques that have been evaluated as well as a similar response period and structure size to the fastest techniques used in experiments. These results indicate that the presented model reaches a satisfactory balance between performance and architectural complexity.

This paper is organized as follows. Section 2 describes the proposed technique. Section 3 describes techniques that are included for purposes of comparison and presents data sets used for the experiments. Section 4 describes how the experiments were performed and the experimental results. Finally, Section 5 presents a series of conclusions.

2. *e*PNN

The *e*PNN is a supervised incremental neural network that is based on the Probabilistic Neural Network (PNN) [35], Gaussian Mixture Model (GMM) and Expectation Maximization (EM) algorithms [9]. GMM was selected for this study because it is a powerful technique that can model a wide range of continuous probability distributions [4] or, in our case, sample distributions for each class. Furthermore, some distributions may be modeled roughly from only one Gaussian. GMM also allows for the extraction of statistical information from the generated model, such as the mean and variance of each kernel, and this may help determine the nature of a specific problem. EM was selected because it is a classical and elegant technique that can be used to adjust GMM parameters.

The *e*PNN is composed of three layers: the input layer, the pattern layer and the decision layer. During the classification phase, each input instance is presented to the input layer. The input layer, which is fully connected to the pattern layer, performs no computations and simply passes the input sample to the pattern layer. In the *e*PNN architecture, the pattern layer is divided into $|C|$ parts in which each part possesses a set of neurons, and $|C|$ is the number of categories in the training data set. Each part is thus responsible for a single category. The neuron outputs for each part are connected to the neuron in the decision layer (this layer includes only one neuron), and this neuron generates the category of the neuron in the pattern layer that is most heavily activated by the input sample. Fig. 1 illustrates the *e*PNN architecture.

Training for this neural network is predominantly concentrated in the pattern layer. The training process is quite similar to the training approach used to train each level of a deep learning algorithm [2]: the representation of knowledge is obtained via unsupervised learning (the distribution of information among kernels of a GMM is performed in an unsupervised manner), and learning is performed locally (only part of the neural network is adjusted for each new training sample). The training procedures are described in the following subsections.