Neural Networks 103 (2018) 108–117

Contents lists available at ScienceDirect

Neural Networks

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

The Growing Curvilinear Component Analysis (GCCA) neural network

Giansalvo Cirrincione [a,b], Vincenzo Randazzo [c,*], Eros Pasero [c]

[a] University of Picardie Jules Verne, Amiens, France
[b] University of South Pacific, Suva, Fiji
[c] Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy

ARTICLE INFO

Article history:
Received 24 December 2017
Received in revised form 2 March 2018
Accepted 27 March 2018
Available online 6 April 2018

Keywords:
Curvilinear component analysis
Dimensionality reduction
Neural network
Non-stationary data
Soft competitive learning
Bridge

ABSTRACT

Big high dimensional data is becoming a challenging field of research. There exist a lot of techniques which infer information. However, because of the curse of dimensionality, a necessary step is the dimensionality reduction (DR) of the information. DR can be performed by linear and nonlinear algorithms. In general, linear algorithms are faster because of less computational burden. A related problem is dealing with time-varying high dimensional data, where the time dependence is due to nonstationary data distribution. Data stream algorithms are not able to project in lower dimensional spaces. Indeed, only linear projections, like principal component analysis (PCA), are used in real time while nonlinear techniques need the whole database (offline). The Growing Curvilinear Component Analysis (GCCA) neural network addresses this problem; it has a self-organized incremental architecture adapting to the changing data distribution and performs simultaneously the data quantization and projection by using CCA, a nonlinear distance-preserving reduction technique. This is achieved by introducing the idea of "seed", pair of neurons which colonize the input domain, and "bridge", a novel kind of edge in the manifold graph, which signals the data non-stationarity. Some artificial examples and a real application are given, with a comparison with other existing techniques.

1. Introduction

DATA mining is more and more facing the extraction of meaningful information from big data (e.g. from internet), which is often very high dimensional. For both visualization and automatic purposes, their dimensionality has to be reduced. This is also important in order to learn the data manifold, which, in general, is lower dimensional than the original data. Dimensionality reduction (DR) also mitigates the curse of dimensionality: e.g., it eases classification, analysis and compression of high-dimensional data.

Most DR techniques work offline, i.e. they require a static database (batch) of data, whose dimensionality is reduced. They can be divided into linear and nonlinear techniques, the latter being in general slower, but more accurate in real world scenarios. See for an overview (Van der Maaten, Postma, & Van der Herik, 2009).

However, the possibility of using a DR technique working in real time is very important, because it allows not only having a projection after only the presentation of few data (i.e. a very fast projection response), but also tracking non-stationary data distributions (e.g. time-varying data manifolds). This can be applied, for example, to all applications of real time pattern recognition, where the data reduction step plays a very important role: fault diagnosis, novelty detection, intrusion detection for alarm systems, speech, face and text recognition, computer vision and scene analysis and so on.

Working in real time requires a data stream, a continuous input for the DR algorithms, which are defined as on-line or, sometimes, incremental (synonym of non-batch). They require, in general, data drawn from a stationary distribution. The fastest algorithms are linear and use the Principal Component Analysis (PCA) by means of linear neural networks, like the Generalized Hebbian Algorithm (GHA, Sanger, 1989), the Adaptive Principal-component Extractor (APEX, Diamantaras & Kung, 1996) and the incremental PCA (candid covariance-free CCIPCA, Weng, Zhang, & Hwang, 2003).

Nonlinear DR techniques are not suitable for online applications. Many efforts have been tried in order to speed-up these algorithms: updating the structure information (graph), new data prediction, embedding updating. However, these incremental versions (e.g. iterative LLE, Kouropteva, Okun, & Pietikainen, 2005, incremental Laplacian eigenmaps, Jia, Yin, Huang, & Hu, 2009, incremental Hessian LLE, Li et al., 2011) require too a cumbersome computational burden and are useless in real time applications.

Neural networks can also be used for data projection. In general, they are trained offline and used in real time (recall phase). In this

* Corresponding author.
E-mail addresses: giansalvo.cirrincione@u-picardie.fr (G. Cirrincione), vincenzo.randazzo@polito.it (V. Randazzo), eros.pasero@polito.it (E. Pasero).

https://doi.org/10.1016/j.neunet.2018.03.017
0893-6080/

case, they work only for stationary data and can be better considered as implicit models of the embedding. Radial basis functions and multilayer perceptrons work well for this purpose (out-of-sample techniques). However, their adaptivity can be exploited either by creating ad hoc architectures and error functions (de Ridder & Duin, 1997) or by using self-organizing maps (SOM) and variants (Qiang, Cheng, & Li, 2010). The former comprises multilayer perceptrons trained on a precomputed Sammon's mapping or with a backpropagation rule based on the Sammon's technique and an unsupervised architecture (SAMANN, Mao & Jain, 1995). These techniques require the stationarity of their training set. The same problem affects the deep neural autoencoders (Hinton & Salakhutdinov, 2006), which are trained for modeling the data reduction. They are multilayer feed-forward neural networks with an odd number of hidden layers and shared weights between the top and bottom layers; sigmoid activation functions are generally used (except in the middle layer, where a linear activation function is usually employed). The main weakness of autoencoders is that their training may converge very slowly, especially in cases where the input and target dimensionality are very high (since this yields a high number of weights in the network). In addition, they are limited by the presence of local optima in the objective function.

The latter family of neural networks comprises the self-organizing feature maps (SOM, Kohonen, Schroeder & Huang, 2001) and its incremental variants. SOM is inherently a feature mapper with fixed topology (which is also its limit). Its variants have no topology (neural gas, NG, Martinetz & Schulten, 1991) or a variable topology and pave the way to pure incremental networks like growing neural gas (GNG, Fritzke, 1995). These networks, in conjunction with the Competitive Hebbian Rule (CHR, White, 1992), create a graph representing the manifold, which is the first step for most DR techniques. NG plus CHR is called Topology representing network (TRN, Martinetz & Schulten, 1994). The approach is called TRNMap (Vathy-Fogarassy, Kiss, & Abonyi, 2008) if the DR technique is a multidimensional scaling (MDS); it is called RBF-NDR (Tomenko, 2011) if the projection is modeled by an RBF with an error function based on geodesic and Euclidean distances: in both cases, the projection follows the graph estimation, which results in the impossibility to track changes in real time. If the graph is computed by GNG, then the DR can be computed by OVI-NG (Estevez & Figueroa, 2006), if Euclidean distances are used, and GNLG-NG (Estevez, Chong, Held, & Perez, 2006) if geodesic distances replace Euclidean distances. However, from the point of view of real time applications, only the former is interesting, because it estimates, in the same time, the graph updating and its projection.

For data drawn from a nonstationary distribution (nonstationary data stream), as it is the case for fault and pre-fault diagnosis and system modeling, the above-cited techniques basically fail. For instance, the methods based on geodesic distances always require a connected graph. If the distribution changes abruptly (jump), they cannot track anymore. Apart the linear techniques, like PCA, which, for their speed, can be adapted to this problem, but are forcedly approximations in case of nonlinearities, only the variant of SOM, called DSOM (Rougier & Boniface, 2011), can be used. It exploits some changes of the SOM learning law in order to avoid a quantization proportional to the data distribution density. However, what is more interesting is the use of constant parameters (learning rate, elasticity) instead of time-decreasing ones. As a consequence, DSOM is able to promptly react to changing inputs. Unfortunately, it is a forgetting network, in the sense that it forgets the past information and only tracks the last changes. This is a serious problem, especially in case the past inputs would carry useful information. There are also neural networks, like SOINN and its variants (Shen, Tomotaka, & Hasegawa, 2007), which record the whole life of the process to be modeled (life-long learning), but are not able to project the information into a lower dimensional space. Nonetheless, they can be considered as a preprocessing clustering stage before the dimensionality reduction step. As a consequence, DR tools can be chosen according to the interest in either a network tailored only on the last data (forgetting network) or the whole (life-long) story, if, for instance, the data can repeat in the future. However, this preprocessing step is outside the scope of this paper, which is centered on the dimensionality reduction step. The same observation can be repeated for the data stream clustering methods (Ghesmoune, Lebbah, & Azzag, 2016). There exist techniques which can be categorized according to the nature of their underlying clustering approach, like: GNG based methods, which are incremental versions (e.g., G-Stream, Ghesmoune, Azzag, & Lebbah, 2014) of the Growing Neural Gas neural network, hierarchical stream methods, like BIRCH (Zhang, Ramakrishnan, & Livny, 1996) and ClusTree (Kranen, Assent, Baldauf, & Seidl, 2016), partitioning stream methods, like CluStream (Aggarwal et al., 2003), and density-based stream methods, like DenStream (Cao, Ester, Qian, & Zhou, 2006) and SOStream (Isaksson, Dunham, & Hahsler, 2012), which is inspired by SOM. No technique takes the dimensionality reduction step into account, which is necessary in case of high dimensional streams. Only in Hontabat and Rising (2016) this problem is considered by applying a variant of the deep autoencoder, called the Variational Autoencoder (VAE, Kingma & Welling, 2014) for DR. There is a significant improvement in the clustering accuracy of high dimensional datasets, but, not all clustering algorithms benefit in the same way from DR. Additionally, regardless of the clustering algorithm, no relevant improvement in the purity of the clusters can be obtained. However, the theory underlying this DR technique is completely different from the classical techniques and is not taken into account in this paper.

Recently, an ad hoc architecture, somewhat in the GNG based category, has been proposed (onCCA, Cirrincione, Hérault, & Randazzo, 2015), which addresses this problem by using an incremental quantization synchronously with a fast projection based on the Curvilinear Component Analysis (CCA, Demartines & Hérault, 1997; Sun, Crowe, & Fyfe, 2010). This neural network requires an initial architecture provided by a fast offline CCA.

The growing CCA (GCCA, Cirrincione, Randazzo, & Pasero, 2018; Kumar, Randazzo, Cirrincione, Cirrincione, & Pasero, 2017) neural network is an improved version of onCCA, which, by using the new idea of seed, does not need an initial CCA architecture. It also uses the principle of bridges in order to detect changes in the data stream.

This paper is an extensive description and analysis of GCCA. After the presentation of the traditional (offline) CCA in Section 2, Section 3 presents some online algorithms such as OVI-NG and DSOM. Section 4 introduces the new algorithm and discusses both its basic ideas and the influence of its user-dependent parameters. Section 5 shows the results of a few simulations on artificial problems and a real application. Section 6 yields the conclusions.

2. The curvilinear component analysis

One of the most important non-linear techniques for dimensionality reduction is the Curvilinear Component Analysis (CCA, Demartines & Hérault, 1997; Sun et al., 2010), which is a non-convex technique based on weighted distances. It derives from the Sammon mapping (Van der Maaten et al., 2009), but improves it because of its properties of unfolding data and extrapolation. CCA is a self-organizing neural network (see Fig. 1), which performs the quantization of a data training set (input space, say X) for estimating the corresponding non-linear projection into a lower dimensional space (latent space, say Y). Two weights are attached to each neuron. The first one has the dimensionality of the X space and is here called X-weight: it quantizes the input data. The