Neural Networks 103 (2018) 29–43

Contents lists available at ScienceDirect

Neural Networks

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

# Multilayer bootstrap networks

Xiao-Lei Zhang

*Center for Intelligent Acoustics and Immersive Communications, School of Marine Science and Technology, Northwestern Polytechnical University, China*

ARTICLE INFO

*Article history:*
Received 26 June 2017
Received in revised form 4 January 2018
Accepted 6 March 2018
Available online 20 March 2018

*Keywords:*
Resampling
Ensemble
Nearest neighbor
Tree

ABSTRACT

Multilayer bootstrap network builds a gradually narrowed multilayer nonlinear network from bottom up for unsupervised nonlinear dimensionality reduction. Each layer of the network is a nonparametric density estimator. It consists of a group of k-centroids clusterings. Each clustering randomly selects data points with randomly selected features as its centroids, and learns a one-hot encoder by one-nearest-neighbor optimization. Geometrically, the nonparametric density estimator at each layer projects the input data space to a uniformly-distributed discrete feature space, where the similarity of two data points in the discrete feature space is measured by the number of the nearest centroids they share in common. The multilayer network gradually reduces the nonlinear variations of data from bottom up by building a vast number of hierarchical trees implicitly on the original data space. Theoretically, the estimation error caused by the nonparametric density estimator is proportional to the correlation between the clusterings, both of which are reduced by the randomization steps.



## 1. Introduction

Principal component analysis (PCA) (Pearson, 1901) is a simple and widely used unsupervised dimensionality reduction method, which finds a coordinate system in the original Euclidean space that the linearly uncorrelated coordinate axes (called principal components) describe the most variances of data. Because PCA is insufficient to capture highly-nonlinear data distributions, many dimensionality reduction methods are explored.

Dimensionality reduction has two core steps. The first step finds a suitable feature space where the density of data with the new feature representation can be well discovered, i.e. a density estimation problem. The second step discards the noise components or small variations of the data with the new feature representation, i.e. a principal component reduction problem in the new feature space.

Dimensionality reduction methods are either linear (He & Niyogi, 2004) or nonlinear based on the connection between the data space and the feature space. This paper focuses on nonlinear methods, which can be categorized to three classes. The first class is kernel methods. It first projects data to a kernel-induced feature space, and then conducts PCA or its variants in the new space. Examples include kernel PCA (Schölkopf, Smola, & Müller, 1998), Isomap (Tenenbaum, De Silva, & Langford, 2000), locally linear embedding (LLE) (Roweis & Saul, 2000), Laplacian eigenmaps (Belkin & Niyogi, 2003; Ng, Jordan, & Weiss, 2002; Shi & Malik, 2000), $t$-distributed stochastic neighbor embedding (t-SNE) (Van der Maaten & Hinton, 2008), and their generalizations (Nie, Zeng, Tsang, Xu, & Zhang, 2011; Yan et al., 2007). The second class is probabilistic models. It assumes that data are generated from an underlying probability function, and takes the posterior parameters as the feature representation. Examples include Gaussian mixture model and latent Dirichlet allocation (Blei, Ng, & Jordan, 2003). The third class is autoassociative neural networks (Hinton & Salakhutdinov, 2006). It learns a piecewise-linear coordinate system explicitly by backpropagation, and uses the output of the bottleneck layer as the new representation.

However, the feature representations produced by the aforementioned methods are defined in continuous spaces. A fundamental weakness of using a continuous space is that it is hard to find a simple mathematical form that transforms the data space to an ideal continuous feature space, since a real-world data distribution may be non-uniform and irregular. To overcome this difficulty, a large number of machine learning methods have been proposed, such as distance metric learning (Xing, Jordan, Russell, & Ng, 2002) and kernel learning (Lanckriet, Cristianini, Bartlett, Ghaoui, & Jordan, 2004) for kernel methods, and Dirichlet process prior for Bayesian probabilistic models (Teh, Jordan, Beal, & Blei, 2005), in which advanced optimization methods have to be applied. Recently, learning multiple layers of nonlinear transforms, named deep learning, is a trend (Hinton & Salakhutdinov, 2006). A deep network contains more than one nonlinear layers. Each layer consists of a group of nonlinear computational units in parallel. Due to the hierarchical structure and distributed representation at each layer, the representation learning ability of a deep network is exponentially more powerful than that of a shallow network when

*E-mail addresses:* xiaolei.zhang@nwpu.edu.cn, xiaolei.zhang9@gmail.com.
*URL:* http://www.xiaolei-zhang.net.

https://doi.org/10.1016/j.neunet.2018.03.005

given the same number of nonlinear units. However, the development of deep learning was mostly supervised, e.g. He, Zhang, Ren, and Sun (2016), Hinton et al. (2012), Schmidhuber (2015), Wang and Chen (2017), Wang, Qin, Nie, and Yuan (2017) and Zhou and Feng (2017). To our knowledge, deep learning for unsupervised dimensionality reduction seems far from explored (Hinton & Salakhutdinov, 2006).

To overcome the aforementioned weakness in a simple way, we revisit the definition of frequentist probability for the density estimation subproblem of dimensionality reduction. *Frequentist probability defines an event's probability as the limit of its relative frequency in a large number of trials* (Wikipedia, 2017). In other words, the density of a local region of a probability distribution can be approximated by counting the events that fall into the local region. This paper focuses on exploring this idea. To generate the events, we resort to *random resampling* in statistics (Efron, 1979; Efron & Tibshirani, 1993). To count the events, we resort to one-nearest-neighbor optimization and binarize the feature space to a discrete space.

To further reduce the small variations and noise components of data, i.e. the second step of dimensionality reduction, we extend the density estimator to a gradually narrowed deep architecture, which essentially builds a vast number of hierarchical trees on the discrete feature space. The overall simple algorithm is named *multilayer bootstrap networks* (MBN).

To our knowledge, although ensemble learning (Breiman, 2001; Dietterich, 2000; Freund & Schapire, 1995; Friedman, Hastie, Tibshirani, et al., 2000; Tao, Tang, Li, & Wu, 2006), which was triggered by random resampling, is a large family of machine learning, it is not very prevalent in unsupervised dimensionality reduction. Furthermore, we did not find methods that estimate the density of data in discrete spaces by random resampling, nor their extensions to deep learning.

This paper is organized as follows. In Section 2, we describe MBN. In Section 3, we give a geometric interpretation of MBN. In Section 4, we justify MBN theoretically. In Section 5, we study MBN empirically. In Section 6, we introduce some related work. In Section 7, we summarize our contributions.

## 2. Multilayer bootstrap networks

### 2.1. Network structure

MBN contains multiple hidden layers and an output layer (Fig. 1). Each hidden layer consists of a group of mutually independent $k$-centroids clusterings; each $k$-centroids clustering has $k$ output units, each of which indicates one cluster; the output units of all $k$-centroids clusterings are concatenated as the input of their upper layer. The output layer is PCA.

The network is gradually narrowed from bottom up, which is implemented by setting parameter $k$ as large as possible at the bottom layer and be smaller and smaller along with the increase of the number of layers until a predefined smallest $k$ is reached.

### 2.2. Training method

MBN is trained layer-by-layer from bottom up.

For training each layer given a $d$-dimensional input data set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ either from the lower layer or from the original data space, we simply need to focus on training each $k$-centroids clustering, which consists of the following steps:

- **Random sampling of features.** The first step randomly selects $\hat{d}$ dimensions of $\mathcal{X}$ ($\hat{d} \leq d$) to form a subset of $\mathcal{X}$, denoted as $\hat{\mathcal{X}} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_n\}$.

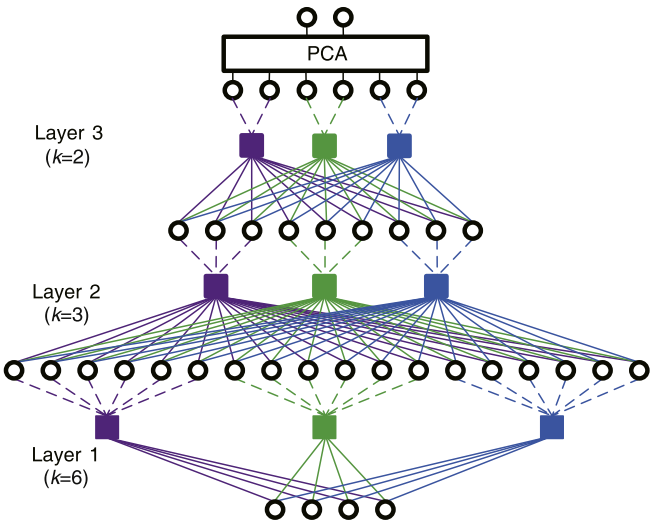


**Fig. 1.** Network structure. The dimension of the input data for this demo network is 4. Each colored square represents a $k$-centroids clustering. Each layer contains 3 clusterings. Parameters $k$ at layers 1, 2, and 3 are set to 6, 3, and 2 respectively. The outputs of all clusterings in a layer are concatenated as the input of their upper layer. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- **Random sampling of data.** The second step randomly selects $k$ data points from $\hat{\mathcal{X}}$ as the $k$ centroids of the clustering, denoted as $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$.
- **One-nearest-neighbor learning.** The new representation of an input $\hat{\mathbf{x}}$ produced by the current clustering is an indicator vector $\mathbf{h}$ which indicates the nearest centroid of $\hat{\mathbf{x}}$. For example, if the second centroid is the nearest one to $\hat{\mathbf{x}}$, then $\mathbf{h} = [0, 1, 0, \ldots, 0]^T$. The similarity metric between the centroids and $\hat{\mathbf{x}}$ at the bottom layer is customized, e.g. the squared Euclidean distance $\arg\min_{i=1}^{k} \|\mathbf{w}_i - \hat{\mathbf{x}}\|^2$, and set to $\arg\max_{i=1}^{k} \mathbf{w}_i^T \hat{\mathbf{x}}$ at all other hidden layers.

### 2.3. Novelty and advantages

Two novel components of MBN distinguish it from other dimensionality reduction methods.

The first component is that each layer is a nonparametric density estimator based on resampling, which has the following major merits:

- It estimates the density of data correctly without any predefined model assumptions. As a corollary, it is insensitive to outliers.
- The representation ability of a group of $k$-centroids clusterings is exponentially more powerful than that of a single $k$-centroids clustering.
- The estimation error introduced by binarizing the feature space can be controlled to a small value by simply increasing the number of the clusterings.

The second component is that MBN reduces the small variations and noise components of data by an unsupervised deep ensemble architecture, which has the following main merits:

- It reduces larger and larger local variations of data gradually from bottom up by building as many as $O(k_L 2^V)$ hierarchical trees on the data space (instead of on data points) implicitly, where $L$ is the total number of layers, $k_L$ is parameter $k$ at the $L$th layer, $V$ is the number of the clusterings at the layer, and function $O(\cdot)$ is the *order* in mathematics.