Pattern Recognition 47 (2014) 25–39

Contents lists available at ScienceDirect

Pattern Recognition

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

Training restricted Boltzmann machines: An introduction

Asja Fischer [a,b,1], Christian Igel [b,*]

[a] Institut für Neuroinformatik, Ruhr-Universität Bochum, Universitätsstraße 150, 44780 Bochum, Germany
[b] Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark

ARTICLE INFO

Available online 7 June 2013

Keywords:
Restricted Boltzmann machines
Markov random fields
Markov chains
Gibbs sampling
Neural networks
Contrastive divergence learning
Parallel tempering

ABSTRACT

Restricted Boltzmann machines (RBMs) are probabilistic graphical models that can be interpreted as stochastic neural networks. They have attracted much attention as building blocks for the multi-layer learning systems called deep belief networks, and variants and extensions of RBMs have found application in a wide range of pattern recognition tasks. This tutorial introduces RBMs from the viewpoint of Markov random fields, starting with the required concepts of undirected graphical models. Different learning algorithms for RBMs, including contrastive divergence learning and parallel tempering, are discussed. As sampling from RBMs, and therefore also most of their learning algorithms, are based on Markov chain Monte Carlo (MCMC) methods, an introduction to Markov chains and MCMC techniques is provided. Experiments demonstrate relevant aspects of RBM training.

1. Introduction

In the last years, models extending or borrowing concepts from *restricted Boltzmann machines* (RBMs, [49]) have enjoyed much popularity for pattern analysis and generation, with applications including image classification, processing, and generation [25,51,34,28,48,31]; learning movement patterns [52,53]; collaborative filtering for movie recommendations [47]; extraction of semantic document representations [46,17,60]; and acoustic modeling [40]. As the name implies, RBMs are a special case of general Boltzmann machines. The latter were introduced as bidirectionally connected networks of stochastic processing units, which can be interpreted as neural network models [1,22]. A Boltzmann machine is a parameterized model representing a probability distribution, and it can be used to learn important aspects of an unknown *target distribution* based on samples from this target distribution. These samples, or observations, are referred to as the training data. Learning or training a Boltzmann machine means adjusting its parameters such that the probability distribution the machine represents fits the training data as well as possible.

In general, learning a Boltzmann machine is computationally demanding. However, the learning problem can be simplified by imposing restrictions on the network topology, which leads us to RBMs, the topic of this tutorial. In Boltzmann machines two types of units can be distinguished. They have *visible neurons* and potentially *hidden neurons*. Restricted Boltzmann machines always have both types of units, and these can be thought of as being arranged in two layers, see Fig. 1 for an illustration. The visible units constitute the first layer and correspond to the components of an observation (e.g., one visible unit for each pixel of a digital input image). The hidden units model dependencies between the components of observations (e.g., dependencies between the pixels in the images) and can be viewed as non-linear feature detectors [22]. In the RBMs network graph, each neuron is connected to all the neurons in the other layer. However, there are no connections between neurons in the same layer, and this restriction gives the RBM its name.

Now, what is learning RBMs good for? After successful learning, an RBM provides a closed-form representation of the distribution underlying the training data. It is a generative model that allows sampling from the learned distribution (e.g., to generate image textures [34,28]), in particular from the marginal distributions of interest, see right plot of Fig. 1. For example, we can fix some visible units corresponding to a partial observation (i.e., we set the corresponding visible variables to the observed values and treat them as constants) and sample the remaining visible units to complete the observation, for example, to solve an image inpainting task [28,51], see Fig. 7 in Section 7. In this way, RBMs can also be used as classifiers: the RBM is trained to model the joint probability distribution of inputs (explanatory variables) and the corresponding labels (response/output variables), both represented by the visible units of the RBM. This is illustrated in the left plot of Fig. 2. Afterwards, a new input pattern can be clamped to the corresponding visible variables and the label can be predicted by sampling, as shown in the right plot of Fig. 2.

Compared to the 1980s when RBMs were first introduced [49], they can now be applied to more interesting problems due to the increase in computational power and the development of new learning strategies [21]. Restricted Boltzmann machines have received a lot of attention recently after being proposed as the building blocks for the multi-layer learning architectures called deep

* Corresponding author. Tel.: +45 21849673.
E-mail addresses: asja.fischer@rub.de (A. Fischer), c.igel@ieee.org, igel@diku.dk (C. Igel).
[1] Tel.: +49 234 32 27987; fax: +49 234 32 14210.

0031-3203/$ - see front matter
http://dx.doi.org/10.1016/j.patcog.2013.05.025

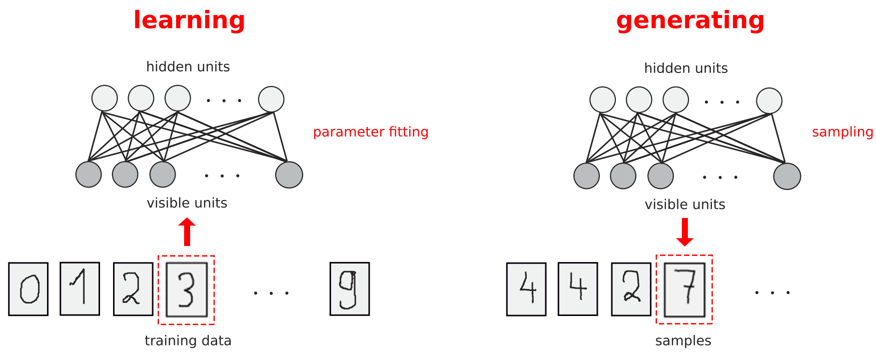

Fig. 1. Left: learning an RBM corresponds to fitting its parameters such that the distribution represented by the RBM models the distribution underlying the training data, here handwritten digits. Right: after learning, the trained RBM can be used to generate samples from the learned distribution.

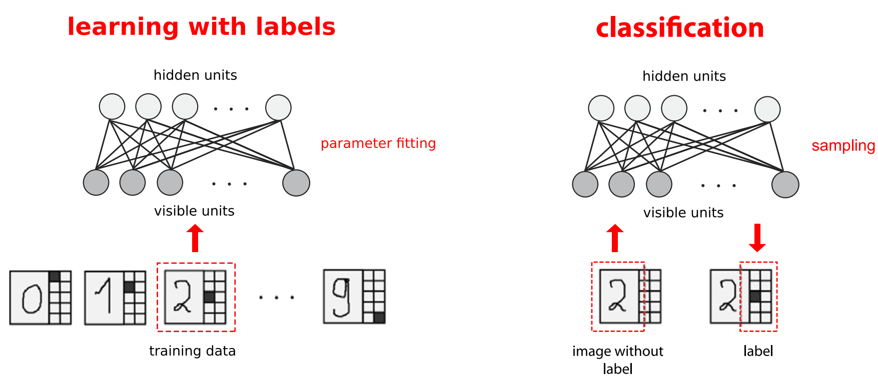

Fig. 2. Left: RBM trained on labeled data, here images of handwritten digits combined with 10 binary indicator variables, one of which is set to 1 indicating that the image shows a particular digit while the others are set to 0. Right: the label corresponding to an input image is obtained by fixing the visible variables corresponding to the image and then sampling the remaining visible variables corresponding to the labels from the (marginalized) joined probability distribution of images and labels modeled by the RBM.

belief networks (DBNs) [25,23]. The basic idea underlying these deep architectures is that the hidden neurons of a trained RBM represent relevant features of the observations, and that these features can serve as input for another RBM, see Fig. 3 for an illustration. By stacking RBMs in this way, one can learn features from features in the hope of arriving at a high-level representation.

It is an important property that single as well as stacked RBMs can be reinterpreted as deterministic feed-forward neural networks. When viewed as neural networks they are used as functions mapping the observations to the expectations of the latent variables in the top layer. These can be interpreted as the learned features, which can, for example, serve as inputs for a supervised learning system. Furthermore, the neural network corresponding to a trained RBM or DBN can be augmented by an output layer where the additional units represent labels (e.g., corresponding to classes) of the observations. Then we have a standard neural network for classification or regression that can be further trained by standard supervised learning algorithms [43]. It has been argued that this initialization (or unsupervised pretraining) of the feed-forward neural network weights based on a generative model helps to overcome some of the problems that have been observed when training multi-layer neural networks [25].

Boltzmann machines can be regarded as probabilistic graphical models, namely undirected graphical models also known as Markov random fields (MRFs) [29]. The embedding into the framework of probabilistic graphical models provides immediate access to a wealth of theoretical results and well-developed algorithms. Therefore, we introduce RBMs from this perspective after providing the required background on MRFs. This approach and the coverage of more recent learning algorithms and theoretical results distinguishes this tutorial from others. Section 2 will provide

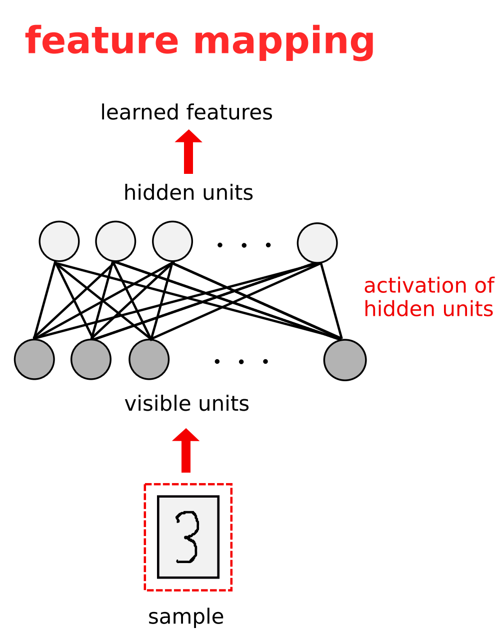

Fig. 3. The trained RBM can be used as a feature extractor. An input pattern is clamped to the visible neurons. The conditional probabilities of the hidden neurons to be 1 are interpreted as a new representation of the input. This new representation can serve as input to another RBM or to a different learning system.

the introduction to MRFs and unsupervised MRF learning. Training of RBMs (i.e., the fitting of the parameters) is usually based on gradient-based maximization of the likelihood of the RBM parameters given