

Deep Belief Networks for Dimensionality Reduction

Athanasios K. Noulas ^a

^a *University of Amsterdam, 1098 SJ Amsterdam, The Netherlands*

Abstract

Deep Belief Networks are probabilistic generative models which are composed by multiple layers of latent stochastic variables. The top two layers have symmetric undirected connections, while the lower layers receive directed top-down connections from the layer above. The current state-of-the-art training method for DBNs is contrastive divergence, an efficient learning technique that can approximate and follow the gradient of the data likelihood with respect to the model parameters. In this work we explore the quality of the non-linear dimensionality reduction achieved through a DBN on face images. We compare the results achieved to the well know Principal Component Analysis as well as with a Harmonium model, which is the top layer of a DBN.

1 Introduction

Dimensionality reduction in statistics refers to the process of reducing the number of random variables at hand. In machine learning, these random variables correspond usually to features of our domain, and the process of reducing their number can be seen as feature selection or extraction. Dimensionality reduction has been the subject of numerous studies, since real-world problems are very high-dimensional and reducing the number of dimensions is essential to boost the efficiency of machine learning algorithms.

There are many different criteria on the basis of which one can reduce the dimensionality of a dataset. A very common technique is Principal Component Analysis (PCA), where we select the linear projections of the data which will result in maximum variance, in the hope that the lower dimensionality space will produce easily separable classes for classification. In classic artificial intelligence, can be seen as a neural network of multiple layers, visible in Figure 1, which is called autoencoder or autoassociator. The network is trained to discover a lower representation of the data, lying in the middle layer, that will allow optimal reconstruction. If there is only one linear hidden layer with k nodes, the autoencoder will project the data in the span of the k first principal components of the dataset. However if the hidden layer is non-linear then different kind of multi-modal abstraction with possibly better results are feasible [3].

In modern artificial intelligence, the most popular framework is undoubtedly probabilistic models. The problem of dimensionality reduction can be seen as a two layer model: the bottom layer comprises of observable variables and corresponds to the input vector, while the top layer comprises from hidden variables and corresponds to the reduced-dimensionality space. The more recent probabilistic PCA and multinomial PCA can be seen as realizations of such a directed graphical model. The undirected version of such two layer models was first introduced in [5] and is called harmonium. In this work we consider harmoniums with multinomial visible variables and continuous, Gaussian hidden variables, introduced in [6], as well as the Restricted Boltzmann Machines (RBMs) which have binary hidden and visible nodes [1]. Harmoniums and RBMs comprise the building blocks of a DBN [2]. We discuss the advantages and disadvantages of directed and undirected models in section 4.

A DBN is a multiple layer generative model. Roughly speaking, the bottom layer is observable, and the multiple hidden layers are created by stacking multiple RBMs on top of each other. The final layer is a Harmonium with Gaussian hidden nodes, which in our case correspond to the reduced dimensionality. We discuss briefly the theoretical advantages of deep architectures in 4, but note here that in general there is no straightforward learning technique. The parameters are learned approximately using Contrastive Divergence learning [1], and the final deep architecture is fine-tuned based on a problem-specific objective function.

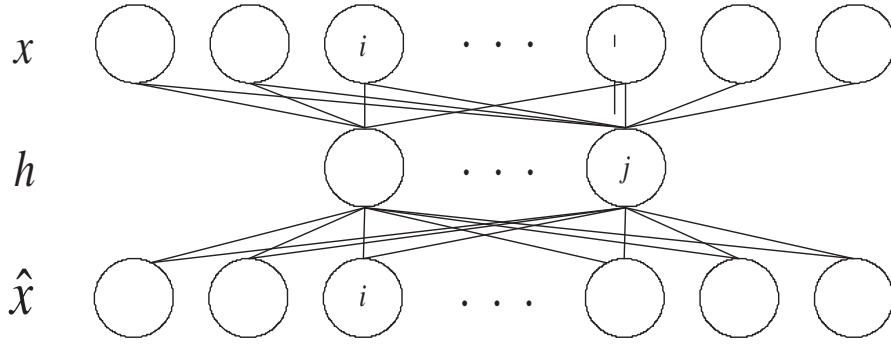


Figure 1: Graphical representation of an autoencoder. The input is the same as the output during training, while the middle layer corresponds to the lower dimensional representation.

This paper is organized as follows. In section 2 we describe the details of the models used to perform dimensionality reduction. In section 3 we describe the dataset used and the experimental results acquired from these models. We conclude this paper with section 4 where not only the experimental results but also theoretical perspectives of the proposed models are discussed.

2 Model Specifications

In this section we describe the models evaluated on dimensionality reduction on our dataset. We start in 2.1 with a description of RBMs which are the simplest two layered model we may have. Their discrete nature makes them an ideal platform to present contrastive divergence learning which is described in section 2.2. In section 2.3 we describe the structure and application of contrastive learning in a harmonium with continuous Gaussian hidden variables. Finally in section 2.4 we present the DBN framework, learning, and fine tuning for dimensionality reduction.

2.1 Restricted Boltzman Machines

A RBM is an energy-based model, which means that the probability distribution over the variables of interest is defined through an energy function. It is composed from a set of observable variables $x = \{x(i)\}$ and a set of hidden variables $h = \{h(i)\}$, as we can see in figure 2. The energy of a given configuration is estimated as:

$$Energy(x, h) = -b'x - x'h - h'Wx \quad (1)$$

while the probability distribution over the configuration of the variables is

$$P(x, h) \propto e^{-Energy(x, h)} \quad (2)$$

note that we did not write the normalization term $Z = \sum_{x, h} e^{-Energy(x, h)}$ in order to express our inability to compute it in general. The parameters W , b and h of the Energy function are learned using a problem specific criteria. In the case of dimensionality reduction we would like to set them to those values that will allow optimal reconstruction of the input vector given it's low-dimensional representation.

In order to see how we can use an RBM for dimensionality reduction and reconstruction, consider an RBM with fewer nodes at the hidden layer than in the visible one. An important property of the RBM is that there are no connections among nodes of the same layer. Thus, we sampling from the conditional distributions $p(h|x)$ and $p(x|h)$, which factorize as:

$$P(x|h) = \prod_i P(x(i)|h) \quad (3)$$

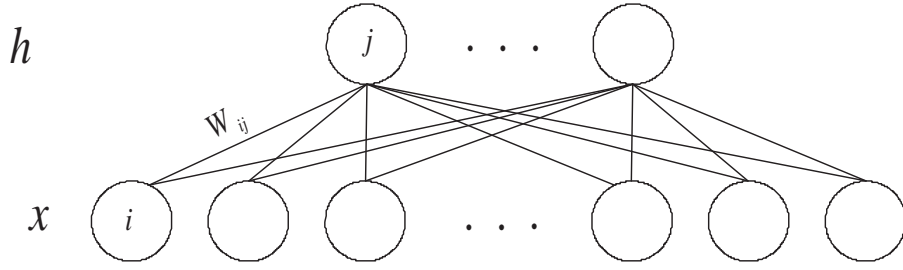


Figure 2: A Restricted Boltzmann machine. Each node of the hidden layer is connected to all the nodes of the visible layer, while there are no connections among nodes of the same layer.

$$P(h|x) = \prod_j P(h(j)|x) \quad (4)$$

Therefore, given an observation vector x we can sample a lower-dimension representation h . We can then use this representation to reconstruct \hat{x} .

In the case of the RBM, all the nodes are binary and the individual node probabilities are given as:

$$P(x(i)|h) = \text{sigm}(b_i + W_{\cdot i} \cdot h) \quad (5)$$

$$P(h(j)|x) = \text{sigm}(c_j + W_j \cdot x) \quad (6)$$

where $W_{\cdot i}$ and W_j refer to the corresponding vectors and rows of matrix W .

In the learning face, we want to discover the parameters that maximize the probability of the vector x given it's encoding in the low dimensional space. That is maximize $p(x|\hat{h})$ with $\hat{h} \sim p(h|x)$, which corresponds to setting the parameters of the RBM to the values that maximize the data likelihood. We can rewrite the data likelihood as:

$$\begin{aligned} p(x, h) &= \frac{e^{-\text{Energy}(x, h)}}{Z} \\ p(x) &= \sum_h \frac{e^{-\text{Energy}(x, h)}}{Z} \\ &= \frac{e^{-\text{FreeEnergy}(x)}}{\sum_x e^{-\text{FreeEnergy}(x)}} \end{aligned}$$

where $\text{FreeEnergy}(x) = -\log \sum_h e^{-\text{Energy}(x, h)}$, with the term *Free Energy* coming from physics.

We can now maximize the data likelihood by gradient ascent, since the average log-likelihood gradient with respect to the parameters $\theta = \{W, b, h\}$ is:

$$E_{\hat{P}} \frac{\partial \log p(x)}{\partial \theta} = -E_{\hat{P}} \frac{\partial \text{FreeEnergy}(x)}{\partial \theta} + E_{\hat{P}} \frac{\partial \text{FreeEnergy}(x)}{\partial \theta} \quad (7)$$

where \hat{P} is the training set empirical distribution, and E_P denotes expected value under the models distribution. We refer the interested reader to [] for formal proof of the gradient formula.

2.2 Contrastive Divergence

Even with equation 7 at hand, it is not easy to train an RBM. As we can see in figure 3, we need to run a sampling algorithm multiple times for each training example. If sampling is repeated for a sufficient amount of iterations, we can acquire the value of the gradient for the specific model parameters. Following the gradient in this manner is extremely expensive computationally, and in multi-dimensional problems practically infeasible.

Instead, we can use contrastive divergence, a technique which has given very promising results. The technique is based on two approximations. The first approximation is replace the average over all possible

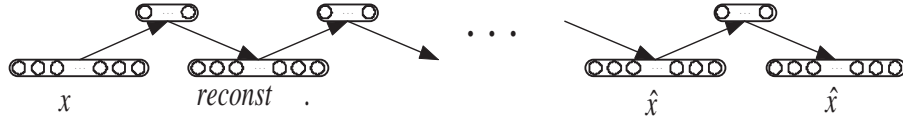


Figure 3: A sampling chain from the original vector x up to convergence to \hat{x} . The first reconstruction (*reconst.*) is used for contrastive divergence learning.

inputs, as seen in equation 7, with a single sample. We update the parameters very often, after one or a few samples, and therefore indirectly we introduce some averaging which we expect to be sufficient. The second and most important approximation is to run the sampler for a single iteration, instead until the chain converges. In this way we expect the parameters to learn the values of the parameters that produce the minimum reconstruction error. From a macroscopic point of view, if the data distribution remains intact for a single reconstruction, it will remain intact for the rest of the iterations, and thus we have converged to the final distribution from the first reconstruction. Once more we refer the interested reader to [1] for a broader discussion.

2.3 Harmonium

The harmonium is a RBM with continuous hidden nodes. Welling et. al in [6] introduced a harmonium with multinomial visible nodes, which proved to be extremely efficient in latent semantic indexing, and explored the theoretical possibilities and restrictions of this structure. It can also be trained using contrastive divergence, with $P(x(i)|h) = \text{sigm}(b_i + W_{.i} \cdot h)$ and $P(h(j)|x) = \mathcal{N}(c_j + W_{j.} \cdot x, 1)$ where $\mathcal{N}(\mu, \sigma)$ is the normal distribution with mean μ and standard deviation σ . In case x is multinomial, we keep a separate weight for each possible $x(i)$, and we sample $x(i)$ from a soft-max function over the possible states [6].

2.4 Deep Belief Networks

A DBN with l layers, models the joint distribution of the observable layer x and the hidden layers h^k for $k = 1 : l$ as

$$p(x, h^1, \dots, h^l) = p(x|h^1) \prod_{k=1:l-2} p(h^k|h^{k+1}) p(h^{l-1}, h^l) \quad (8)$$

where each of the conditional probabilities is modelled as an RBM, while the probability over the two top layers is modelled as a harmonium. A graphical representation of a DBN is visible in figure 2.4.

Training a DBN has two phases. In the first phase, we start by training the RBM of layers x and h^1 . We keep adding consecutive layers treating the reconstructions acquired in the previous layer as data of the visible layer. For instance, we get the projection at h^1 and train an RBM between h^1 and h^2 using the reconstructions of h^1 as data. We train the top two layers model as an harmonium, using the reconstruction of h^{l-1} as data, and hidden Gaussian nodes in h^l . The first phase is performed using contrastive divergence, and we expect to get the model parameters near a good local maximum of the data likelihood function. In the second phase, we tune the parameters of the whole DBN based on a problem specific criteria. In the case of dimensionality reduction this is performed with back propagation, exactly the way it is performed for neural networks. The hope is that this phase will tune the parameters to the local maximum approached by the first phase.

3 Experiments

In this section we present the dimensionality reduction experiments conducted for this work. In section 3.1 we describe briefly the dataset we used, and in section 3.2 the objective and setup of our experiments. The numerical results acquired can be found in section 3.3

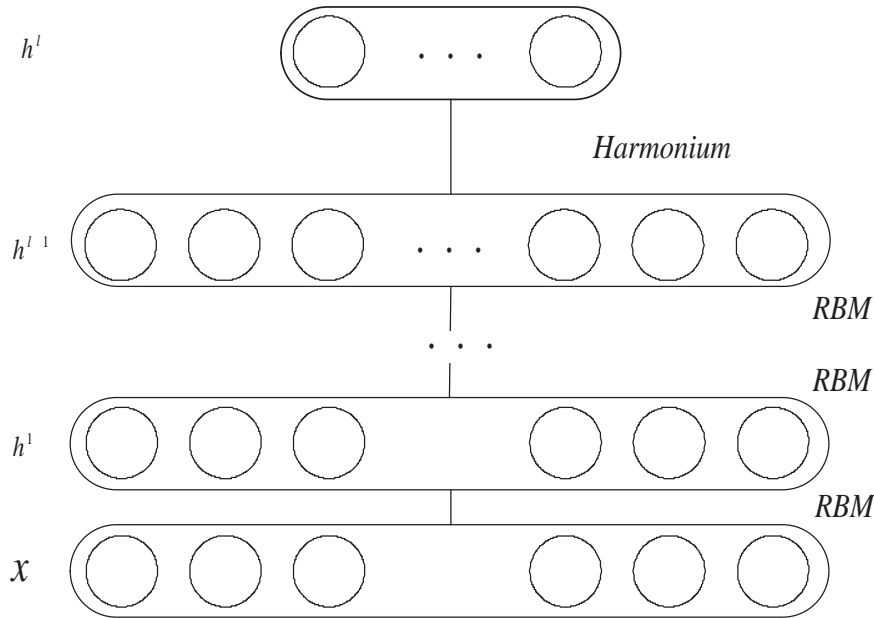


Figure 4: A Deep Belief Network with l layers of hidden nodes. The first $l - 1$ layers are added as RBMs, using the projection on the last layer as data. The top two layers are a Harmonium with continuous Gaussian hidden variables, and corresponds to the reduced dimensional space.

3.1 Dataset

We used 400 faces coming from the AR Face Database [4]. The face images are 36×48 pixels, creating 1728 dimensional datapoints. The original grey valued images were turned into they binary counterparts in order to be usable as input for the RBM.

3.2 Experimental Setup

We extracted the principal components of the dataset and performed projection and reconstruction for a grid of 40 values between 2 and 300 dimensions. The same applied for harmoniums, where we trained different harmoniums for each lower space dimensionality. Finally, we trained different DBNs with 4 hidden layers. The first three hidden layers had 1000, 500 and 250 nodes, and the top layer took values in the low dimensional space. We evaluated each reconstruction using the sum of squared differences between the original value and the reconstructed one.

3.3 Results

In figure 6, we can see the reconstruction error for different dimensionality reductions. The experiments grid had the values 2 : 20, 25 : 100 and 110 : 300. Furthermore, in figure , we can see how the reconstructed face images look like. We see that the harmonium outperforms PCA, while the DBN produces the best results for low number of retained dimensions.

This was expected for two reasons. Firstly, the harmonium and the DBN perform non-linear dimensionality reduction which proves more efficient for this kind of data. Secondly, they are trained with main objective the optimal reconstruction of a vector from the hidden space, while PCA's main objective is to maximize the variance of the data in the lower dimensional space. The DBN outperforms the Harmonium when a few dimensions are retained because it learns the data distribution as a product of multiple layers. In this way, we can learn much more varying distributions. On the other hand, as more flexibility is available because we keep a greater number of dimensions, the Harmonium uses it optimally while the DBN is trapped in a sub-optimal local minima. The image vectors require sharp distributions in order to discriminate face pixels from the background, and that's why this flexibility produces better results.

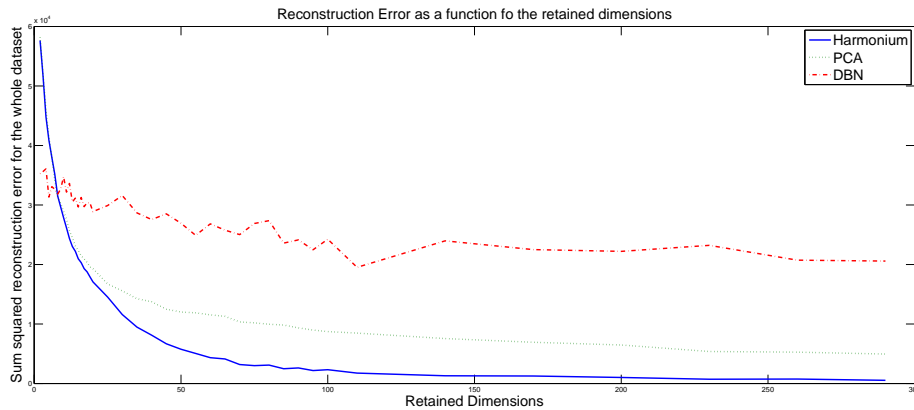


Figure 5: Sum of the squared error between the reconstructions and the original vectors as a function of the retained dimensions

4 Discussion & Conclusions

In this paper, an undirected deep architecture provided optimal results in dimensionality reduction. This comes in favor of some theoretical considerations that support the use of similar architectures for learning in artificial intelligence. The arguments regard the flexibility these architectures provide in terms of inference, mapping in the latent space and representational power.

Deep architectures have been avoided in the past because they are much harder to train. More specifically, if one tries to train directly a deep architecture, the two top layers tend to get meaningful, criteria-related weights, while the rest just perform useless perturbation of the data. Training with contrastive divergence overcomes this difficulty. In the first phase we create multiple perturbations of the data, retaining the information contained in it. Thus, we hope that when optimizing our problem-specific objective, we will be able to utilize the layer containing the optimal data reconstruction. The question that arises naturally is how many layers do we need for a given problem, and it cannot be answered directly. However, we should keep in mind that a DBN can use large quantities of unlabelled data to learn the structure of a given dataset. We can then use a few labeled examples to perform high accuracy classification in this well structured space.

This view is very similar to human learning. Babies observe the world around them in an unsupervised manner, and afterward they use a few labeled examples to perform classification in the space they have structured. For example, someone can use thousands of face images in an unsupervised way, and then based on the task at hand use labeled examples of male or female subjects, to classify genders or different ages to perform regression in the age space.

Another important quality of undirected graphs in comparison to the directed ones is that we can sample the configuration of the hidden nodes directly, since they are independent of each other given the visible ones. This can be very useful in applications that require fast lower-space representations like for instance document retrieval, with an example of a harmonium performing document retrieval and discovering word structure given in [6]. Finally, we can use the trained DBN to generate data from our training distribution.

References

- [1] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, August 2002.
- [2] Geoffrey E. Hinton. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, July 2006.
- [3] Nathalie Japkowicz, Stephen Jose Hanson, and Mark A. Gluck. Nonlinear autoassociation is not equivalent to PCA. *Neural Computation*, 12(3):531–545, 2000.
- [4] A.M. Martinez and R. Benavente. The ar face database.



Figure 6: From top-down we have different datapoint visualizations for PCA, Harmonium and DBN. Their reconstructions from the different models for 3, 5, 10 and 20 retained dimensions from left to right

- [5] P. Smolensky. Information processing in dynamical systems: foundations of harmony theory. *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*, pages 194–281, 1986.
- [6] Max Welling, Michal Rosen-Zvi, and Geoffrey Hinton. Exponential family harmoniums with an application to information retrieval. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1481–1488. MIT Press, Cambridge, MA, 2005.