Boltzmann Machines for Image Denoising

KyungHyun Cho

Department of Information and Computer Science
Aalto University School of Science, Finland
{firstname.lastname@aalto.fi}

Abstract. Image denoising based on a probabilistic model of local image patches has been employed by various researchers, and recently a deep denoising autoencoder has been proposed in [2] and [17] as a good model for this. In this paper, we propose that another popular family of models in the field of *deep learning*, called Boltzmann machines, can perform image denoising as well as, or in certain cases of high level of noise, better than denoising autoencoders. We empirically evaluate these two models on three different sets of images with different types and levels of noise. The experiments confirmed our claim and revealed that the denoising performance can be improved by adding more hidden layers, especially when the level of noise is high.

Keywords: Image Denoising, Deep Learning, Restricted Boltzmann Machine, Deep Boltzmann Machine

1 Introduction

Numerous approaches based on machine learning have been proposed for image denoising tasks over time. A dominant approach has been to perform denoising based on local statistics of image patches. Under this approach, small image patches from a larger noisy image are denoised and combined afterward to form a clean image.

For instance, in [9] independent component analysis (ICA) was used to estimate a dictionary of sparse elements and compute the sparse code of image patches. Subsequently, a shrinkage nonlinear function is applied to the estimated sparse code elements to suppress those elements with small absolute magnitude. These shrunk sparse codes are then used to reconstruct a noise-free image patch. More recently, it was shown in [5] that sparse overcomplete representation may be more useful in denoising images.

In essence, these approaches build a probabilistic model of natural image patches using a single layer of sparse latent variables. The posterior distribution of each noisy patch is either exactly computed or estimated, and the noise-free patch is reconstructed as an expectation of a conditional distribution over the posterior distribution.

Some researchers have proposed very recently to utilize a model that has more than one layers of latent variables for image denoising. It was shown in [2] that a deep denoising autoencoder [16] that learns a mapping from a noisy image patch to its corresponding clean version, can perform as good as the state-of-the-art denoising methods. Similarly, a variant of a stacked sparse denoising autoencoder that is more effective in image denoising was recently proposed in [17].

Along this line of research, we propose yet another type of deep neural networks for image denoising, in this paper. A Gaussian-Bernoulli restricted Boltzmann machines (GRBM) [6] and deep Boltzmann machines (GDBM) [13] are empirically shown to perform well in image denoising, compared to stacked denoising autoencoders. Furthermore, we evaluate the effect of the number of hidden layers of both Boltzmann machines and denoising autoencoders. The empirical evaluation is conducted using different noise types and levels on three different sets of images.

2 Deep Neural Networks

We start by briefly describing Boltzmann machines and denoising autoencoders which have become increasingly popular in the field of machine learning.

2.1 Boltzmann Machines

Originally proposed in 1980s, a Boltzmann machine (BM) [1] and especially its structural constrained version, a restricted Boltzmann machine (RBM) [14] have become increasingly important in machine learning since it was shown that a deep multi-layer perceptron can be trained easily by stacking RBMs on top of each other [6]. More recently, another variant of a BM, called a deep Boltzmann machine (DBM), has been proposed and shown to outperform other conventional machine learning methods in many tasks (see, e.g., [12]).

We first describe a Gaussian-Bernoulli DBM (GDBM) that has L layers of binary hidden units and a single layer of Gaussian visible units. A GDBM is defined by its energy function

$$
\begin{aligned}
-E(\mathbf{v}, \mathbf{h} \mid \boldsymbol{\theta}) = & \sum_i -\frac{(v_i - b_i)^2}{2\sigma^2} + \sum_{i,j} \frac{v_i}{\sigma^2} h_j^{(1)} w_{i,j} + \sum_j h_j^{(1)} c_j^{(1)} \\
& + \sum_{l=2}^{L} \left(\sum_j h_j^{(l)} c_j^{(l)} + \sum_{j,k} h_j^{(l)} h_k^{(l+1)} u_{j,k}^{(l)} \right), \qquad (1)
\end{aligned}
$$

where $\mathbf{v} = [v_i]_{i=1 \dots N_v}$ and $\mathbf{h}^{(l)} = \left[h_j^{(l)}\right]_{j=1 \dots N_l}$ are N_v Gaussian visible units and N_l binary hidden units in the l-th hidden layer. $\mathbf{W} = [w_{i,j}]$ is the set of weights between the visible neurons and the first layer hidden neurons, while $\mathbf{U}^{(l)} = \left[u_{j,k}^{(l)}\right]$ is the set of weights between the l-th and $l+1$-th hidden neurons. σ^2 is the shared variance of the conditional distribution of v_i given the hidden units.

With the energy function, a GDBM can assign a probability to each state vector $\mathbf{x} = [\mathbf{v}; \mathbf{h}^{(1)}; \cdots; \mathbf{h}^{(L)}]$ using a Boltzmann distribution: $p(\mathbf{x} \mid \boldsymbol{\theta}) = \exp\left\{-E(\mathbf{x} \mid \boldsymbol{\theta})\right\} / Z(\boldsymbol{\theta})$. Then, the parameters can be learned by maximizing the log-likelihood

$$
\mathcal{L} = \sum_{n=1}^{N} \log \sum_{\mathbf{h}} p(\mathbf{v}^{(n)}, \mathbf{h} \mid \boldsymbol{\theta})
$$

given N training samples $\{\mathbf{v}^{(n)}\}_{n=1,\dots,N}$, where $\mathbf{h} = \left[\mathbf{h}^{(1)}; \cdots ; \mathbf{h}^{(L)}\right]$.

Although the update rules based on the gradients of the log-likelihood function are well defined, it is intractable to exactly compute them. Hence, an approach that uses variational approximation together with Markov chain Monte Carlo (MCMC) sampling was proposed in [13]. This approach is often used together with a pretraining algorithm [13, 4]. In this paper, we initialized GDBMs using the two-stage pretraining algorithm recently proposed in [4].

A Gaussian-Bernoulli RBM (GRBM) is a special case of a GDBM, where the number of hidden layers is restricted to one, $L = 1$. Unlike GDBMs, it is possible to compute the posterior distribution over the hidden units exactly and tractably by

$$p(h_j = 1 \mid \mathbf{v}, \boldsymbol{\theta}) = f\left(\sum_i w_{ij} \frac{v_i}{\sigma^2} + c_j\right), \tag{2}$$

where f is a logistic sigmoid function, and we dropped the superscript (1) from h_j for simplicity.

This removes the need for the variational approximation in computing the gradient and allows an efficient block Gibbs sampling. Based on this property many fast approximate algorithms for training GRBMs, such as contrastive divergence [7], have been proposed.

2.2 Denoising Autoencoders

A denoising autoencoder (DAE) [16] is a special form of multi-layer perceptron network with $2L - 1$ hidden layers and $L - 1$ sets of (tied) weights. A DAE tries to learn a network that denoises an explicitly corrupted input vector by minimizing the following cost function:

$$\sum_{n=1}^{N} \left\| \mathbf{W} g^{(1)} \circ \cdots \circ g^{(L-1)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)} \left(\eta(\mathbf{v}^{(n)})\right) - \mathbf{v}^{(n)} \right\|^2, \tag{3}$$

where $f^{(l)} = \phi(\mathbf{W}^{(l)\top} \mathbf{h}^{(l-1)})$ and $g^{(l)} = \phi(\mathbf{W}^{(l)} \mathbf{h}^{(2L-l)})$ are, respectively, encoding and decoding functions for l-th layer with a component-wise nonlinearity function ϕ. η stochastically adds noise to an input vector $\mathbf{x}$ at each update step. $\mathbf{W}^{(l)}$ is the weights between the l-th and $(l+1)$-th layers and is shared by the encoder and decoder.

3 Image Denoising

Let us define a set of N binary matrices $\mathbf{D}_n \in \mathbb{R}^{p \times d}$ that extract a set of small image patches given a large, whole image $\mathbf{x} \in \mathbb{R}^d$, where $d = wh$ is the product of the width w and the height h of the image and p is the size of image patches. Then, an image can be denoised by

$$\mathbf{x} = \left(\sum_{n=1}^{N} \mathbf{D}_n^\top r_{\boldsymbol{\theta}}(\mathbf{D}_n \tilde{\mathbf{x}})\right) \oslash \left(\sum_{n=1}^{N} \mathbf{D}_n^\top \mathbf{D}_n \mathbf{1}\right), \tag{4}$$

where $\oslash$ is an element-wise division and $\mathbf{1}$ is a vector of ones. $r_{\boldsymbol{\theta}}(\cdot)$ is an image denoising function, parameterized by $\boldsymbol{\theta}$, that denoises each image patch $\mathbf{D}_n\mathbf{x}$.

In short, Eq. (4) extracts and denoises image patches from the input image. Then, it combines them by taking an average of those overlapping pixels.

One of the popular choices for $r_{\boldsymbol{\theta}}(\cdot)$ has been to construct a probabilistic model with a set of latent variables that describe natural image patches (see, e.g., [8, 5]). Under this approach denoising can be considered as a two-step reconstruction. First, the posterior distribution over the latent variables is computed given an image patch. Based on that, the conditional distribution, or its mean, over the visible units is computed and used as a denoised image patch.

3.1 Boltzmann machines

We consider a BM with a set of Gaussian visible units $\mathbf{v}$ that correspond to the pixels of an image patch and a set of binary hidden units $\mathbf{h}$. Then, the goal of denoising can be written as

$$p(\mathbf{v} \mid \tilde{\mathbf{v}}) = \sum_{\mathbf{h}} p(\mathbf{v} \mid \mathbf{h}) p(\mathbf{h} \mid \tilde{\mathbf{v}}) = \mathbb{E}_{\mathbf{h}|\tilde{\mathbf{v}}} \left[p(\mathbf{v} \mid \mathbf{h}) \right], \tag{5}$$

where $\tilde{\mathbf{v}}$ is a noisy input patch. In other words, we find a mean of the conditional distribution of the visible units with respect to the posterior distribution over the hidden units given the visible units fixed to the corrupted input image patch.

However, since taking the expectation over the posterior distribution is usually not tractable, it is often easier and faster to approximate it. We approximate the marginal conditional distribution in (5) with $p(\mathbf{v} \mid \tilde{\mathbf{v}}) \approx p(\mathbf{v} \mid \mathbf{h}) Q(\mathbf{h})$, where $Q(\mathbf{h})$, parameterized by $\boldsymbol{\mu} = \mathbb{E}_{Q(\mathbf{h})}[\mathbf{h}]$, is a fully-factorial (approximate) posterior distribution $p(\mathbf{h} \mid \tilde{\mathbf{v}})$.

Given a noisy image patch $\tilde{\mathbf{v}}$, following this approach, a GRBM reconstructs a noise-free patch by

$$\hat{v}_i = \sum_{j=1}^{N_h} w_{ij} \mathbb{E}\left[\mathbf{h} \mid \tilde{\mathbf{v}}\right] + b_i.$$

The conditional distribution over the hidden units can be computed exactly from Eq. (2).

Unlike a GRBM, the posterior distribution of the hidden units of a GDBM is neither tractably computable nor has an analytical form. Instead, we use a fully-factorial *approximate* posterior $Q(\mathbf{h}) = \prod_{l=1}^{L} \prod_j \mu_j^l$, where the parameters $\mu_j^{(l)}$'s can be estimated by maximizing the variational lower-bound [13].

Once the variational parameters are converged, a GDBM reconstructs a noise-free patch by

$$\hat{v}_i = \sum_{j=1}^{N_l} w_{ij} \mu_j^{(1)} + b_i.$$

The convergence of the variational parameters may take too much time in practice. Hence, in the experiments, we initialize the variational parameters by the feed-forward propagation using the doubled weights [13] and performing the fixed-point update for at most five iterations only.

Set	# of all images	# of color images	Min. Size	Max. Size
Textures	64	0	512×512	1024×1024
Aerials	38	37	512×512	2250×2250
Miscellaneous	44	16	256×256	1024×1024

Table 1. Descriptions of the test image sets.

3.2 Denoising autoencoders

An encoder part of a DAE can be considered as performing an approximate inference of a fully-factorial posterior distribution of top-layer hidden units, i.e. a bottleneck, given an input image patch [16]. Hence, a similar approach can be applied to DAEs.

Firstly, the variational parameters $\boldsymbol{\mu}^{(L)}$ of the fully-factorial posterior distribution $Q(\mathbf{h}^{(L)}) = \prod_j \mu_j^{(L)}$ are computed by

$$\boldsymbol{\mu}^{(L)} = f^{(L-1)} \circ \cdots \circ f^{(1)} \left(\tilde{\mathbf{v}}\right).$$

Then, the denoised image patch can be reconstructed simply by propagating the variational parameters through the decoding nonlinearity functions $g^{(l)}$ such that

$$\hat{\mathbf{v}} = g^{(1)} \circ \cdots \circ g^{(L-1)} \left(\boldsymbol{\mu}^{(L)}\right).$$

4 Experiments

In the experiment, we test both GRBMs and GDBMs together with DAEs having varying numbers of hidden layers. These models are compared to each other on a *blind* image denoising task, where no prior knowledge about target images nor the type or level of noise is known when the models are trained. In other words, no separate training was done for different types or levels of noise injected to the test images. Unlike this, for instance, in [17] each DAE was trained specifically for the target noise level and type by changing $\eta(\cdot)$ accordingly.

4.1 Datasets

We used three sets of images, *textures*, *aerials* and *miscellaneous*, from the USC-SIPI Image Database[1] as test images. Tab. 1 lists the details of the image sets.

These datasets are, in terms of contents and properties of images, very different from each other. For instance, most of the images in the texture set have highly repetitive patterns that are not present in the images in the other two sets. Most images in the aerials set have simultaneously both coarse and fine structures. Also, the sizes of the images vary quite a lot across the test sets and across the images in each set.

As we are aiming to evaluate the performance of denoising a very general image, we used a large separate data set of natural image patches to train the models. We used

[1] http://sipi.usc.edu/database/

a large number of image patches randomly of sizes 4×4, 8×8 and 16×16 extracted from CIFAR-10 dataset [10]. The same set of experiments was run with the models trained on the patches from the Berkeley Segmentation Dataset [11], and the similar results were observed.

4.2 Settings

We tried three different depth settings for both Boltzmann machines and denoising autoencoders; a single, two and four hidden layers. Each hidden layer had the same number of hidden units which was the constant factor 5 multiplied by the number of pixels in an image patch, as suggested in [17].

We denote Boltzmann machines with one, two and four hidden layers by GRBM, GDBM(2) and GDBM(4), respectively. Denoising autoencoders are denoted by DAE, DAE(2) and DAE(4), respectively.

The GRBMs were trained using the enhanced gradient [3] and persistent contrastive divergence (PCD) [15]. The GDBMs were trained by PCD after initializing the parameters with a two-stage pretraining algorithm [4]. DAEs were trained by a stochastic backpropagation, and when there were more than one hidden layers, we pretrained each pair of consecutive layers as a single-layer DAE with sparsity target set to 0.1.

Two types of noise have been tested; white Gaussian and salt-and-pepper. White Gaussian noise simply adds zero-mean normal random value with a predefined variance to each image pixel, while salt-and-pepper noise sets a randomly chosen subset of pixels to either black or white. Three different noise levels (0.1, 0.2 and 0.4) were tested. In the case of white Gaussian noise, they were used as standard deviations, and in the case of salt-and-pepper noise, as a noise probability.

After noise was injected, each image was preprocessed by pixel-wise adaptive Wiener filtering, following the approach of [9]. The width and height of the pixel neighborhood were chosen to be small enough (3×3) to avoid removing too much detail.

4.3 Results and Analysis

In Fig. 1, the performances of all the tested models trained on 8×8 image patches, measured by the peak signal-to-noise ratio (PSNR), are presented. Those trained on the patches of different sizes showed similar trend, and they are omitted here.

Interestingly, the GRBMs consistently performed comparably to much deeper DAEs in most cases regardless of the level of noise. This indirectly suggests that learning a good generative model might be important in image denoising. Considering that the number of parameters of the GRBM is, for instance, only half compared to DAE(2), training a model in a fully generative manner helps learning more compact representation of natural image patches that are more suitable for image denoising.

However, the GDBMs which are essentially deeper version of the GRBM were only able to outperform the other models in the high noise regime. In the cases of the aerials and miscellaneous sets, in the low noise regime, the GDBMs lag behind all the other models. A possible explanation for this rather poor performance of the GDBMs in the low noise regime is that the posterior distribution had to be approximated, whereas it

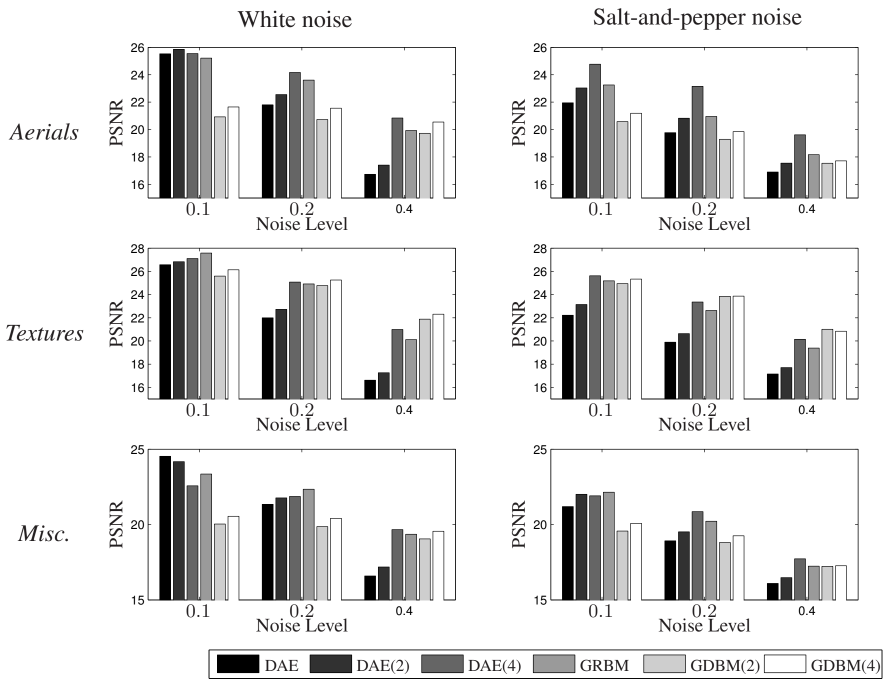

Fig. 1. The median PSNRs of grayscale images corrupted by different types and levels of noise.

was computed exactly in the case of GRBMs. A better approximation strategy might resolve this problem.

An important observation is that the deeper models significantly outperformed their corresponding shallower models as the level of injected noise grew. In other words, the power of the deep models became more evident as the difficulty of the task increased.

5 Discussion

In this paper, we proposed that, in addition to DAEs, Boltzmann machines (BM) can also be used efficiently for denoising images. Boltzmann machines and DAEs were empirically evaluated against each other in the *blind* image denoising task where no prior knowledge about target images and their level of corruption was assumed.

The experimental results showed that Boltzmann machines (BM) are good, potential alternatives to DAEs. BMs and DAEs performed comparably to each other in the low noise regime, while BMs were able to, in many cases, outperform DAEs when the level of injected noise was high. This suggests that BMs may be more robust to noise than DAEs are.

More careful look at the experimental results clearly showed that, in the case of DAEs, hidden layers do improve performance, especially when the level of noise is high. This did not always apply to BMs, where we found that the GRBMs outperformed, or performed as well as, the GDBMs in many cases. Regardlessly, in the high noise regime, it was always beneficial to have more hidden layers, even for BMs.

References

1. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzmann machines. Cognitive Science 9, 147–169 (1985)
2. Burger, H., Schuler, C., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 2392 –2399 (june 2012)
3. Cho, K., Raiko, T., Ilin, A.: Enhanced gradient for training restricted Boltzmann machines. Neural Computation 25(3), 805–831 (2013)
4. Cho, K., Raiko, T., Ilin, A., Karhunen, J.: A Two-Stage Pretraining Algorithm for Deep Boltzmann Machines. In: NIPS 2012 Workshop on Deep Learning and Unsupervised Feature Learning. Lake Tahoe (December 2012)
5. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. Image Processing, IEEE Transactions on 15(12), 3736–3745 (Dec 2006)
6. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (July 2006)
7. Hinton, G.: Training products of experts by minimizing contrastive divergence. Neural Computation 14, 1771–1800 (August 2002)
8. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Transactions on Neural Networks 10(3), 626–34 (1999)
9. Hyvärinen, A., Hoyer, P., Oja, E.: Image denoising by sparse code shrinkage. In: Intelligent Signal Processing. IEEE Press (1999)
10. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., Computer Science Department, University of Toronto (2009)
11. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision. vol. 2, pp. 416–423 (July 2001)
12. Salakhutdinov, R., Hinton, G.: An efcient learning procedure for deep Boltzmann machines. Neural Computation 24, 1967–2006 (2012)
13. Salakhutdinov, R., Hinton, G.E.: Deep Boltzmann machines. In: Proc. of the Int. Conf. on Artificial Intelligence and Statistics (AISTATS 2009). pp. 448–455 (2009)
14. Smolensky, P.: Information processing in dynamical systems: foundations of harmony theory. In: Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations, pp. 194–281. MIT Press, Cambridge, MA, USA (1986)
15. Tieleman, T.: Training restricted Boltzmann machines using approximations to the likelihood gradient. ICML '08, ACM, New York, NY, USA (2008)
16. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research 11, 3371–3408 (Dec 2010)
17. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Bartlett, P., Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 25, pp. 350–358 (2012)