Gated Boltzmann Machine in Texture Modeling

Tele Hao, Tapani Raiko, Alexander Ilin, and Juha Karhunen

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

Abstract. In this paper, we consider the problem of modeling complex texture information using undirected probabilistic graphical models. Texture is a special type of data that one can better understand by considering its local structure. For that purpose, we propose a convolutional variant of the Gaussian gated Boltzmann machine (GGBM) [12], inspired by the co-occurrence matrix in traditional texture analysis. We also link the proposed model to a much simpler Gaussian restricted Boltzmann machine where convolutional features are computed as a preprocessing step. The usefulness of the model is illustrated in texture classification and reconstruction experiments.

Keywords: Gated Boltzmann Machine, Texture Analysis, Deep Learning, Gaussian Restricted Boltzmann Machine

1 Introduction

Deep learning [7] has resulted in a renaissance of neural networks research. It has been applied to various machine learning problem successfully: for instance, hand-written digit recognition [4], document classification [7], and non-linear dimensionality reduction [8].

Texture information modeling has been studied for decades, see, e.g., [6]. It can be understood by considering combinations of several repetitive local features. In this manner, various authors proposed hand-tuned feature extractors. Instead of understanding the generative models for textures, those extractors try to consider the problem discriminatingly. An old model called co-occurrence matrix was proposed in [6], where it was used to measure how often a pair of pixels with a certain offset gets particular values, thus tackling the structure of the textures. Despite the good performances of these extractors, they suffer from the fact that they contain only little information about the generative model for textures. Also, these extractors can only be applied to certain type of data, and it is fairly hard to adopt them to other tasks if needed. Conversely, generative models of textures can be applied to various texture modeling applications. In this direction, some statistical approaches for modeling textures have been introduced in [14] and [11]. A pioneering work of texture modeling using deep network is proposed in [9].

Texture modeling is a very important task in real-world computer vision applications. An object can have any shape, size, and illumination condition.

However, the texture pattern within the objects can be rather consistent. By understanding that, one can improve the understanding of objects in complex real-world recognition tasks.

In this paper, a new type of building block for deep network is explored to understand texture modeling. The new model is used to model the local relationship within the texture in a biologically plausible manner. Instead of searching exhaustively over the whole image patch, we propose to search for local structures in a smaller region of interest. Also, due to the complexity of the model, a novel learning scheme for such model is proposed.

2 Background

2.1 Co-occurrence Matrices in Texture Classification

Co-occurrence matrix [6] measures the frequencies a pair of pixels with a certain offset gets particular values. Modeling co-occurrence matrices instead of pixels brings the analysis to a more abstract level immediately, and it has therefore been used in texture modeling.

The co-occurrence matrix $\mathbf{C}$ is defined over $\{m \times n\}$ size image I, where $\{1 \ldots N_g\}$ levels of gray scales are used to model pixel intensities. Under this assumption, the size of $\mathbf{C}$ is $\{N_g \times N_g\}$. Each entry in $\mathbf{C}$ is defined by

$$c_{ij} = \sum_{m=1}^{M} \sum_{n=1}^{N} \begin{cases} 1 \text{ if } I(m,n) = i \ \& \ I(m+\delta_x, n+\delta_y) = j \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

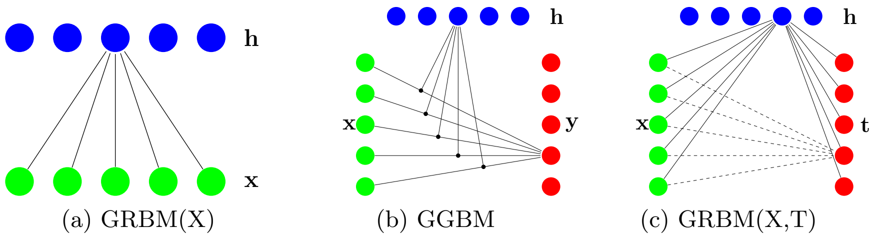
Different offset schemes for $\{\delta_x, \delta_y\}$ result in different co-occurrence matrices. For instance, one can look for textural pattern over an image with offset $\{-1, 0\}$ or $\{0, 1\}$. These different co-occurrence matrices typically have information about the texture from different orientations. Therefore, a set of invariant features can be obtained by having several different co-occurrence matrices together.

2.2 Gaussian Restricted Boltzmann Machines

Gaussian restricted Boltzmann machine (GRBM) [7] is a basic building block for deep networks. It tries to capture binary hidden features (hidden neurons) from a continuous valued data vector (visible neurons), where hidden neurons and visible neurons are fully connected by an undirected graph. Even though an efficient learning algorithm was proposed for GRBM [7], training is still very sensitive to initialization and choice of learning parameters. Cho et al. proposed an enhanced gradient learning algorithm for GRBM in [2]. Throughout the paper, a modified version of GRBM [3] is adopted, where the energy function is defined as

$$\mathbf{E}(\mathbf{x}, \mathbf{h}) = -\sum_{ik} \frac{x_i}{\sigma_i^2} h_k w_{ik} - \sum_k h_k c_k + \sum_i \frac{(x_i - b_i)^2}{2\sigma_i^2} \tag{2}$$

where $x_i, i = 1, \dots, N$ and $h_k, k = 1, \dots, K$ refer to the visible neurons and hidden neurons, respectively. w_{ik} characterizes the weight of the connection between x_i and h_k, and c_k is the bias term for hidden neuron h_k. The mean and variance of x_i are denoted by b_i and σ_i. Accordingly, the joint distribution of different Boltzmann machine can be computed as $P(\mathbf{x}, \mathbf{h}) = Z^{-1} \exp(-E(\mathbf{x}, \mathbf{h}))$, where Z is the normalization constant. A schematic illustration of GRBM is shown in Figure 1a. The input neurons $\mathbf{x}$ connect to the hidden neurons $\mathbf{h}$, where each connection is characterized by w_{ik}, A weight matrix and two bias vectors are used to characterize all the connections in the network.



(a) GRBM(X) (b) GGBM (c) GRBM(X,T)

Fig. 1: The schematic Illustration of the structures of different Boltzmann machines.

2.3 Gaussian Gated Boltzmann Machine

Gaussian gated Boltzmann machine(GGBM) [12] is a higher order Boltzmann machine where there are two sets of visible neurons and one set of hidden neurons. It is developed to model the complex image transformation in paired images [12], and the internal structures of a single image [13]. The energy function of GGBM is defined as

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = -\sum_{ijk} \frac{x_i}{\sigma_i} \frac{y_j}{\sigma_j} h_k w_{ijk} + \sum_i \frac{(x_i - b_i^x)^2}{2\sigma_i^2} + \sum_j \frac{(y_j - b_j^y)^2}{2\sigma_j^2} - \sum_k c_k h_k \quad (3)$$

A graphical illustration of GGBM is shown in Figure 1b. GGBM tries to model the relationship between visible neurons $\mathbf{x}$ and $\mathbf{y}$ by a set of hidden variables $\mathbf{h}$. A dot on the crossing of two lines in the figure represents one weight scalar w_{ijk}. The biases are omitted in the figure for simplicity.

The weight tensor w_{ijk} can be rather large if there are lots of visible neurons and hidden neurons. For instance, two data vectors of size 100 and 200 and a hidden vector of size 500 increase the number of parameter in the w_{ijk} up to $100 \times 200 \times 500$. In order to overcome this, a low rank factorization of the weight tensor is done as $w_{ijk} \rightarrow \sum_f w_{if}^x w_{jf}^y w_{kf}^h$ [12]. A new different simplification approach based on convolutional operation on local structure of texture is considered in this paper.

3 Proposed Method

Combining the nature of texture information and GGBM, a modified GGBM especially suitable for texture modeling is proposed. To start with, we consider a slightly modified general gated Boltzmann machine where there are pair-wise connections between all sets of nodes. This model has the most comprehensive information about the input vectors. Accordingly, the energy function of the model is written as

$$\begin{aligned} E(\mathbf{x},\mathbf{y},\mathbf{h}) = &-\sum_{ijk}\frac{x_i}{\sigma_i}\frac{y_j}{\sigma_j}h_k w_{ijk} - \sum_{ij}\frac{x_i}{\sigma_i}\frac{y_j}{\sigma_j}u_{ij} + \sum_i \frac{(x_i - b_i^x)^2}{4\sigma_i^2} \\ &+ \sum_j \frac{(y_j - b_j^y)^2}{4\sigma_j^2} - \sum_k h_k c_k - \sum_{ik}\frac{x_i}{2\sigma_i^2}h_k v_{ik}^{(1)} - \sum_{jk}\frac{y_j}{2\sigma_j^2}h_k v_{jk}^{(2)} \end{aligned} \tag{4}$$

where u_{ij},$v_{ik}^{(1)}$ and $v_{ik}^{(2)}$ are additional parameters to model the pair-wise connections between two sets of visible neurons $\{\mathbf{x},\mathbf{y}\}$ and hidden neurons $\mathbf{h}$. Instead of looking for the image transformation, we seek for the internal structure of texture information. Therefore, the same patch of image is fed to the two sets of visible neurons, that is $\mathbf{x} = \mathbf{y}$. Accordingly, the weights $\mathbf{v}$ and bias $\mathbf{b}$ for the two sets of visible neurons are tied, which is $\mathbf{V} = \mathbf{V}^{(1)} = \mathbf{V}^{(2)}$; $\mathbf{b} = \mathbf{b}^x = \mathbf{b}^y$. Also, a unified variance $\sigma^2 = \sigma_i^2 = \sigma_j^2$ is learned to reduce the complexity of the model further

The complexity of the model remains as the weight tensor w_{ijk} still needs huge learning efforts. As $\mathbf{x} = \mathbf{y}$, x_i and y_j can be considered a pair of pixels, and h_k is learned to model this interaction. Given an image patch, the traditional GGBM will go through all the combinations of such pairs. This is highly redundant as the texture is repetitive within a very small region. Recalling that co-occurrence matrix tries to summarize the interaction of pairs of pixels over a certain area, this structure can be introduced to GGBM. In order to do that, we will assume $w_{ijk} = w_{dk}$, such that the weight w_{ijk} depends only on the displacement d and the hidden neuron h_k. d represents the offest from i to j. Similarly, $u_{ij} = u_d$. One can think of w_{dk} and u_d as a convolutional model only over the local regions in image patches. Convolutional approximation has been argued to be rather successful in other applications such as image recognition tasks [10]. It is further assumed tthat $w_{dk} = 0$ for large displacement d.

After these simplifications, the energy function (4) becomes

$$\begin{aligned} E(\mathbf{x},\mathbf{y},\mathbf{h}) = &-\frac{1}{\sigma^2}\sum_{ijk} x_i y_j h_k w_{d_{ij}k} - \frac{1}{\sigma^2}\sum_d x_i y_j u_{d_{ij}} + \frac{1}{2\sigma^2}\sum_i (x_i - b_i)^2 \\ &- \frac{1}{\sigma^2}\sum_{ik} x_i h_k v_{ik} - \sum_k h_k c_k \end{aligned} \tag{5}$$

Ignoring the restriction $\mathbf{x} = \mathbf{y}$, learning and inference of GGBM can be based on sequentially sampling from the conditional distributions $p(\mathbf{x}|\mathbf{y},\mathbf{h})$, $p(\mathbf{y}|\mathbf{x},\mathbf{h})$

and $p(\mathbf{h}|\mathbf{x},\mathbf{y})$. These conditional forms can all be written in a close form as

$$p(\mathbf{x}|\mathbf{y},\mathbf{h}) = \prod_i \mathcal{N}\left(b_i + \sum_{jk} y_j h_k w_{ijk} + \sum_j y_j u_{ij} + \sum_k h_k v_{ik}, \sigma^2\right) \tag{6}$$

$$p(\mathbf{h}|\mathbf{x},\mathbf{y}) = \prod_k \frac{1}{1+\exp\left(-\frac{1}{\sigma^2}\sum_i x_i v_{ik} - \frac{1}{\sigma^2}\sum_j y_j v_{jk} - \frac{1}{\sigma^2}\sum_{ij} x_i y_j w_{ijk} - c^h_k\right)}. \tag{7}$$

3.1 GRBM with Preprocessing

We also define a related but much simpler model as follows. Firstly, we define auxiliary variables $t_d = \sum_i x_i y_{i+d}$ where d is the offset between pixels i and j as before. This formulation stems from the principle of the co-occurrence matrix where each feature is only related to particular pairs of pixels in the image. These computations can be done as a preprocessing step. Secondly, we learn a GRBM using the concatenation of vectors $[\mathbf{x},\mathbf{t}]$ as data. We call this model the GRBM(X,T) and illustrate it in Figure 1c. In the figure, the dashed line represents $\mathbf{t}$ being computed from $\mathbf{x}$.

When we write the energy function of GRBM(X,T)

$$\begin{aligned}\mathbf{E}(\mathbf{x},\mathbf{t},\mathbf{h}) = &-\frac{1}{\sigma^2}\left(\sum_{ik} x_i h_k v_{ik} + \sum_{dk} t_i h_k w_{dk}\right) - \sum_k h_k c_k \\ &+ \frac{1}{2\sigma^2}\left(\sum_i (x_i - b_i)^2 + \sum_d (t_d - u_d)^2\right),\end{aligned} \tag{8}$$

we notice the similarities to the GGBM energy function in Equation (5). Each parameter has its corresponding counterpart. The only remaining difference is

$$\mathbf{E}(\mathbf{x},\mathbf{t},\mathbf{h}) - \mathbf{E}(\mathbf{x},\mathbf{y},\mathbf{h}) = \frac{1}{2\sigma^2}\sum_d t_d^2 + \text{const} \tag{9}$$

It turns out $p(\mathbf{h}|\mathbf{x},\mathbf{y})$ can be written in the exact same form as in Equation (7).

Since learning higher order Boltzmann machines is known to be quite difficult, we propose to use this related model as a way for learning them. So in practice we first train a GRBM(X,T), and then convert the parameters to the GGBM model. Actually, in texture classification, the converted model produces exactly the same hidden activations $\mathbf{h}$ and thus the same classification results. On the other hand, in the texture reconstruction problem, the GRBM(X,T) model cannot be used directly, since $\mathbf{t}$ cannot be computed from partial observations.

We noticed experimentally, that the converted GGBM model needs to be further regularized, since the regularizing terms t_d^2 in the energy function of GRBM(X,T) are dropped off as seen in Equation (9). We simply converted w_{dk} and u_d by scaling them with a constant factor smaller than 1, and chose that constant by the smallest validation reconstruction error.

Settings	Training	Testing
X	25.0%	16.2%
T	54.2%	50.4%
XT	61.8%	52.8%
FX	87.6%	63.0%
FT	91.7%	65.3%
FXT	**94.8%**	**67.0%**

(a) Brodatz 24 data set

Settings	Training	Testing
X	29.2%	19.0%
T	46.7%	43.8%
XT	57.3%	49.2%
FX	68.2%	60.4%
FT	72.0%	62.2%
FXT	**77.4%**	**66.2%**

(b) KTH data set

Table 1: The texture classification result on various benchmark data sets.

4 Experiments

We test our methods with texture classification and reconstruction experiments. The proposed method is first run to extract a set of meaningful features from different datasets, and these features are then used for the classification and reconstruction.

4.1 Texture Classification

Two publicly available texture data sets are tested. The liblinear library [5] is used to build a classifier. In all classification experiments, a L1-regularized logistic regression (L1LR) is trained. For the feature extraction experiments, one step contrastive divergence and some regularization parameters[1] are used. In all experiment, 1000 hidden neurons are chosen, and $w_{dk} = 0$ for all $||d||_\infty > 5$. For comparison, we conducted six different classification experiments:

raw image patches (X) L1LR on X
transforms of X (T) L1LR on T
joint X and T (XT) L1LR on XT
features from X (FX) First run GRBM on X, and then L1LR on FX
features from T (FT) First run GRBM on T, and then L1LR on FT
features from XT (FXT) First run GRBM on XT, and then L1LR on FXT

The classification results in our experiments cannot be directly compared to other texture classification experiments as they typical extract a highly complex feature set from the whole image, while we directly extract features from small patches of textures. In other words, our model is capable of performing classification even though there is only little information about the texture, while it is typically hard to extract features if the images are too small in other conventional texture classification experiments.

[1] weight decay = 0.0002, momentum = 0.2

Brodatz 24 Data Set A subset of 24 different textures is manually selected from a large collection of 112 different textures. Only one large image is available for each class [11]. Each image in each class is divided into 25 $\{128 \times 128\}$ small images, 13 of them are used to generate the training patches, and rest of them are used to generate the testing patches. The patch size in the learning and testing is manually selected as $\{20 \times 20\}$. 240000 image patches are used in extracting the features. 24000 samples are used for training a classifier and 2400 samples are used for testing. The classification results are shown in Table 1a. Among all the experiments, the proposed method performs the best.

KTH texture dataset This dataset [1] has 11 different textures, 4 different samples for each texture, and 108 different images are available for each sample. Each image is of size $\{200 \times 200\}$, and the patch size is still selected as 20×20. Only the 108 images from sample a[2] in each texture are used: 54 for generating training samples and 54 for generating testing samples. 118800 patches are used for extracting the features. 11000 patches are used for training a classifier and 1100 sample are used for testing. The best result is obtained with the proposed method. Please note a poorer overall performance is expected as the variations within the training samples make the problem harder. The detailed results are shown in Table 1b.

4.2 Texture Reconstruction

We also made a demonstration of texture reconstruction for showing the connections between the proposed model and its approximation. In this experiment, 6 random image patches are chosen from the Brodatz 24 dataset testing samples, and a $\{10 \times 10\}$ square center of the patches are removed for reconstruction. The reconstruction result can be seen in Figure 2. For comparison, the reconstruction result from GRBM(X) model is provided. From this experiment, we can see that the learned model is capable of learning a generative model for the texture successfully. Despite the regularization, the reconstructions still seem to have blockiness by over-emphasizing low frequencies. One way to improve the result would be to use the GRBM(X,T) as an initialization for the GGBM, and train it further.

5 Conclusions

In this paper, we tackled the problem of modeling texture information. We proposed a modified version of GGBM and a simpler learning algorithm for that. From the experimental results, we can argue that the proposed model is beneficial in terms of modeling the structured information such as textures. Among all the results, the highest accuracies are obtained by the features learned from the proposed model. Although these accuracies are not the state-of-the-art, the proposed model opened up a possibility where the texture information can be successfully modeled using the higher order Boltzmann machine.

[2] Available at http://www.nada.kth.se/cvap/databases/kth-tips/

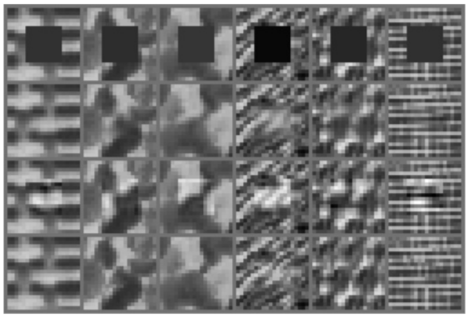


Fig. 2: The texture reconstruction experiment. The first row shows the random samples with missing centers. The second row shows the reconstruction from GRBM model, and the reconstruction from the proposed model is shown in the third row. The original samples are shown at the last row.

References

1. Caputo, B., Hayman, E., Mallikarjuna, P.: Class-Specific Material Categorisation. Int. Conf. on Computer Vision pp. 1597–1604 (2005)
2. Cho, K., Raiko, T., Ilin, A.: Gaussian-Bernoulli Deep Boltzmann Machine. NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning (2011)
3. Cho, K., Ilin, A., Raiko, T.: Improved Learning of Gaussian-Bernoulli Restricted Boltzmann Machines. Int. Conf. on Artifical Neural Networks pp. 10–17 (2011)
4. Cireşan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Deep, Big, Simple Neural Nets for Handwritten Digit Recognition. Neural Comput. 22(12), 3207–3220
5. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A Library for Large Linear Classification. JMLR pp. 1871–1874 (2008)
6. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural Features for Image Classification. IEEE trans. Syst., Man, Cybern. 3(6), 610–621 (1973)
7. Hinton, G., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
8. Hinton, G., Salakhutdinov, R.: Discovering Binary Codes for Documents by Learning Deep Generative Models. Topics in Cognitive Science 3(1), 74–91 (Aug 2010)
9. Kivinen, J., Williams, C.: Multiple Texture Boltzmann Machines. Int. Conf. Artificial Intelligence and Statistics 22, 638–646 (2012)
10. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. Int. Conf. Machine Learning p. 77 (2009)
11. Liu, L., Fieguth, P.: Texture Classification from Random Features. IEEE Trans. Pattern Anal. Mach. Intell. 34(3), 574–586 (2012)
12. Memisevic, R., Hinton, G.E.: Learning to Represent Spatial Transformations with Factored Higher-Order Boltzmann Machines. Neural Comput. 22(6), 1473–1492
13. Ranzato, M., Krizhevsky, A., Hinton, G.E.: Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images. Int. Conf. Artificial Intelligence and Statistics 9, 621–628 (2010)
14. Varma, M., Zisserman, A.: A Statistical Approach to Material Classification Using Image Patch Exemplars. IEEE Trans. Pattern Anal. Mach. Intell. 31(11), 2032–2047 (2009)