Matrix Product Operator Restricted Boltzmann Machines

Anonymous Author(s)
Affiliation
Address
email

Abstract

A restricted Boltzmann machine (RBM) learns a probabilistic distribution over its input samples and has numerous uses like dimensionality reduction, classification and generative modeling. Conventional RBMs accept vectorized data that dismisses potentially important structural information in the original tensor (multi-way) input. Matrix-variate and tensor-variate RBMs, named MvRBM and TvRBM, have been proposed but are all restrictive by construction. This work presents the matrix product operator RBM (MPORBM) that utilizes a tensor network generalization of Mv/TvRBM, preserves input formats in both the visible and hidden layers, and results in higher expressive power. A novel training algorithm integrating contrastive divergence and an alternating optimization procedure is also developed.

1 Introduction

A restricted Boltzmann machine (RBM) [1] is a probabilistic model that employs a layer of hidden variables to achieve highly expressive marginal distributions. RBMs are an unsupervised learning technique and have been extensively explored and applied in various fields [2–4]. However, conventional RBMs are designed for vector data and cannot directly deal with matrices and higher-dimensional data, which are common in real-life. The traditional approach to apply an RBM on high-dimensional data is through vectorization of the data which leads to two drawbacks. First, the spatial information in the original data is lost, thus weakening the model's capability to represent these structural correlations. Second, the fully connected visible and hidden units may cause overfitting since the intrinsic low-rank property of many real-life data is disregarded.

Researchers have been motivated to develop corresponding multiway RBMs [5, 3]. However, those works are both aiming to capture the interaction among different vector inputs and are hence not directly applicable to matrix and tensor data. The first RBM designed for tensor inputs is described in [6], where the visible layer is represented as a tensor but the hidden layer is still a vector. Furthermore, the connection between the visible and hidden layers is described by a canonical polyadic (CP) tensor decomposition [7], which constrains the model representation capability [8]. Another RBM related model that utilizes tensor input is the matrix-variate RBM (MvRBM) [8]. The visible and hidden layers in an MvRBM are both matrices. Nonetheless, to limit the number of parameters, an MvRBM models the connection between the visible and hidden layers through two separate matrices, which restricts the ability of the model to capture correlations between different data modes.

All these issues have motivated this work. Specifically, we propose a matrix product operator (MPO) restricted Boltzmann machine (MPORBM) where both the visible and hidden layers are in tensor forms. Moreover, MPORBM utilizes a general and powerful tensor network, namely an MPO, to connect the tensorial visible and hidden layers. By doing so, an MPORBM achieves a more powerful model representation capacity than MvRBM and at the same time greatly reduces the number of model parameters compared to a standard RBM.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

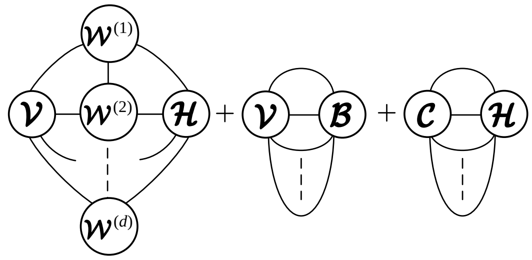


Figure 1: Negative energy functions ($-E$) of the MPORBM.

2 Method

In an MPORBM, both the visible layer $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$ and the hidden layer $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{J_1 \times \cdots \times J_d}$ are d-way tensors. As a result, the weight matrix $\boldsymbol{W}$ is now a $2d$-way tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{I_1 \times \cdots \times I_d \times J_1 \times \cdots \times J_d}$, which is represented by an MPO instead in order to lift the curse of dimensionality. Per definition, the corresponding MPO decomposition represents each entry of $\boldsymbol{\mathcal{W}}$ as

$$\boldsymbol{\mathcal{W}}(i_1, \ldots, i_d, j_1, \ldots, j_d) = \sum_{r_1, r_2, \ldots, r_d}^{R_1, R_2, \ldots, R_d} \boldsymbol{\mathcal{W}}^{(1)}(r_1, i_1, j_1, r_2) \cdots \boldsymbol{\mathcal{W}}^{(d)}(r_d, i_d, j_d, r_1). \quad (1)$$

The “building blocks" of the MPO are the 4-way tensors $\boldsymbol{\mathcal{W}}^{(1)}, \ldots, \boldsymbol{\mathcal{W}}^{(d)}$, also called the MPO-cores. The dimensions $R_1, \ldots, R_d$ of the summation indices $r_1, \ldots, r_d$ are called the MPO-ranks. With both the visible and hidden layers being tensors, it is therefore also required that the bias vectors $\boldsymbol{b}, \boldsymbol{c}$ are tensors $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times \cdots \times I_d}, \boldsymbol{\mathcal{C}} \in \mathbb{R}^{J_1 \times \cdots \times J_d}$, respectively. A tensor network diagram representation of the negative energy function of the MPORBM is shown in Figure 1, where each tensor is represented by a node in the network and the number of edges connected to a node represents the order of the corresponding tensor. The vertical edges between the different MPO-cores $\boldsymbol{\mathcal{W}}^{(1)}, \ldots, \boldsymbol{\mathcal{W}}^{(d)}$ represent the summations in (1) and are the key ingredients in being able to express generic weight tensors $\boldsymbol{\mathcal{W}}$. The storage complexity of an MPORBM with uniform ranks and dimensions is $O(dIJR^2)$, which is linear on the order d and therefore removes the curse of dimensionality. The MvRBM model can be interpreted as a very specific case of an MPORBM where there are only 2 MPO-cores without any vertical edge, which limits the expressive power. The corresponding conditional distribution over the hidden or visible layer involves the summation of the weight MPO with either the hidden or visible layer tensors into a d-way tensor, which is then added elementwise with the corresponding bias tensor. The final step in the computation of the conditional probability is an elementwise application of the logistic sigmoid function on the resulting tensor.

Let $\Theta = \{\boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{W}}^{(1)}, \boldsymbol{\mathcal{W}}^{(2)}, \ldots, \boldsymbol{\mathcal{W}}^{(d)}\}$ denote the model parameters. The model learning task is then formulated into maximizing the training data likelihood:

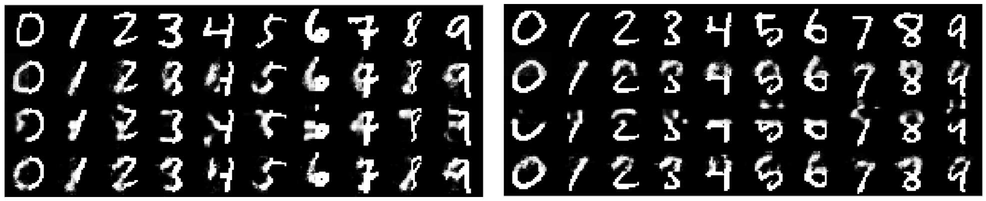
$$\boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{V}}; \Theta) = p(\boldsymbol{\mathcal{V}}; \Theta) = \sum_{\boldsymbol{\mathcal{H}}} p(\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{H}}; \Theta) \quad (2)$$

with respect to the model parameter Θ. Similar to the standard RBM [1], the expression of the gradient of the log-likelihood is:

$$\frac{\partial}{\partial \Theta} \log \boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{V}}; \Theta) = -\mathbb{E}_{\boldsymbol{\mathcal{H}}|\boldsymbol{\mathcal{V}}}\left[\frac{\partial E(\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{H}})}{\partial \Theta}\right] + \mathbb{E}_{\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{H}}}\left[\frac{\partial E(\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{H}})}{\partial \Theta}\right] \quad (3)$$

We mainly use the contrastive divergence (CD) procedure to train the MPORBM model. First, a Gibbs chain is initialized with one particular training sample $\boldsymbol{\mathcal{V}}_{(0)} = \boldsymbol{\mathcal{X}}_{train}$, followed by K times Gibbs sampling which results in the chain $\{(\boldsymbol{\mathcal{V}}_{(0)}, \boldsymbol{\mathcal{H}}_{(0)}), (\boldsymbol{\mathcal{V}}_{(1)}, \boldsymbol{\mathcal{H}}_{(1)}), \ldots, (\boldsymbol{\mathcal{V}}_{(K)}, \boldsymbol{\mathcal{H}}_{(K)})\}$. The model expectation is then approximated by $\{\boldsymbol{\mathcal{V}}_{(K)}\}$. The derivative of $\log \boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{V}}; \Theta)$ with respect to the k-th MPO-core $\boldsymbol{\mathcal{W}}^{(k)}$ can be computed by removing $\boldsymbol{\mathcal{W}}^{(k)}$ from two tensor network diagrams (one diagram with $\boldsymbol{\mathcal{V}}_{(0)}, \boldsymbol{\mathcal{H}}_{(0)}$ and one with $\boldsymbol{\mathcal{V}}_{(K)}, \boldsymbol{\mathcal{H}}_{(K)}$), taking the elementwise difference and summing over all edges. The derivatives of the log-likelihood with respect to the bias tensors $\boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}$ are

$$\frac{\partial}{\partial \boldsymbol{\mathcal{B}}} \log \boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{V}}; \Theta) = \boldsymbol{\mathcal{V}}_{(0)} - \boldsymbol{\mathcal{V}}_{(K)}, \quad \frac{\partial}{\partial \boldsymbol{\mathcal{C}}} \log \boldsymbol{\mathcal{L}}(\boldsymbol{\mathcal{V}}; \Theta) = \boldsymbol{\mathcal{H}}_{(0)} - \boldsymbol{\mathcal{H}}_{(K)}.$$



(a) (b)

Figure 2: Image completion results when given only the (a) right half; and (b) bottom half. Top row: original binarized images; 2nd row: RBM completion; 3rd row: MvRBM completion; 4th row: MPORBM completion.

Instead of updating all MPO-cores simultaneously with one batch of input training data, we employ the alternating optimization procedure (AOP). This involves updating only one MPO-core at a time while keeping the others unchanged using the same batch of input training data. We name this parameter learning algorithm CD-AOP. The superiority of AOP over simultaneously updating all MPO-cores, which we call CD-SU henceforth, will be demonstrated through numerical experiments.

3 Experiments

In the first experiment, we demonstrate the superior data classification accuracy of MPORBM using standard datasets, namely, the Binary Alphadigits, normalized DrivFace, Arcene and COIL-100 datasets. The vectorized sample sizes of these datasets vary from 320 to 49152. We assume a binary input in our RBM setting, so for non-binary datasets a multi-bit vector is used to represent each value in the original data. The trained RBM models were employed to extract features from the hidden layer. These features were then utilized to train a K Nearest Neighbor (K-NN) classifier with $K = 1$ for all experiments. Table 1 lists the resulting classification errors. The restrictive expressive power of the weight matrix in MvRBM explains why it has the worst classification performance for all datasets. The worse performance of the standard RBM may be caused by overfitting due to the small training sample size. For COIL-100 dataset, the standard RBM fails to learn the large number of parameters in the full weight matrix due to out-of-memory errors. We need to mention that CD-AOP algorithm achieves a surprisingly 0% test error because of the small test sample number. Moreover, the CD-AOP algorithm shows a higher classification accuracy than CD-SU, which indicates that the alternating updating scheme is more suitable for the proposed MPORBM model.

Table 1: Classification errors of different RBM models.

Datasets	RBM	MvRBM	MPORBM CD-SU	MPORBM CD-AOP
Alphadigits	28.10%	31.20%	28.10%	26.90%
DrivFace	24.20%	15.48%	9.68%	8.06%
Arcene	45.00%	34.00%	32.00%	27.00%
COIL-100	–	6.82%	6.82%	0.00%

Finally, we show that an MPORBM is good at generative modeling exemplified by image completion. We tested this generative task on the binarized MNIST dataset: one half of the image was provided to the trained RBM models to complete the other half. Figure 2 shows the completed images of different RBM models when given the same randomly selected right and bottom halves, respectively. It is clear that MvRBM is not able to complete the image well, which further confirms the efficacy of the MPO generalization.

4 Conclusion

The MPORBM is proposed, which preserves the tensorial nature of the input data and utilizes a matrix product operator (MPO) to represent the weight matrix. The MPORBM generalizes all existing RBM models to tensor inputs and has better storage complexity since the number of parameters grows only linearly with the order of the tensor. Experiments have verified the superiority of MPORBM over traditional counterparts.

References

[1] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[2] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pages 536–543. ACM, 2008.

[3] Alex Krizhevsky, Geoffrey Hinton, et al. Factored 3-way restricted Boltzmann machines for modeling natural images. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 621–628, 2010.

[4] Abdel-rahman Mohamed and Geoffrey Hinton. Phone recognition using restricted Boltzmann machines. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 4354–4357. IEEE, 2010.

[5] Graham W Taylor and Geoffrey E Hinton. Factored conditional restricted Boltzmann machines for modeling motion style. In *Proceedings of the 26th annual international conference on machine learning*, pages 1025–1032. ACM, 2009.

[6] Tu Dinh Nguyen, Truyen Tran, Dinh Q Phung, et al. Tensor-Variate Restricted Boltzmann Machines. In *AAAI*, pages 2887–2893, 2015.

[7] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[8] Guanglei Qi, Yanfeng Sun, Junbin Gao, et al. Matrix variate restricted Boltzmann machine. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 389–395. IEEE, 2016.