Reinforcing Feature Distributions of Hidden Units of Boltzmann Machine using Correlations

Peixu Cai[1†*], Wangze Shen[2†], Ruohan Yang[3†], and Qixian Zhou[4†]

[1] Department of Computer Science, Boston University, MA,02215, United States

[2] College of Literature, Science, and the Arts, University of Michigan, MI, 48109, United States

[3] Alibaba Group, Shanghai, 201199, China

[4] Arts and Science, University of Toronto, Toronto, Ontario, M5S 1A1, Canada

*Corresponding author's e-mail: peixu789@bu.edu

†These authors contributed equally to this work.

Abstract. This paper introduces and analyses the method of applying Neuroscience methods to Boltzmann Machine, involving a combination of cognitive psychology, information theory, and dynamical systems. We utilized the emergent property of the probability of hidden layers to find the pattern of how units are behaving when stimulated by the visual layer and research into enhancing the predictive encoding capability of the encoding layer. We measure the connections and links between the units of the encoding layer by approximating it with the probability distribution of two units' activation behaviours. For example, the portion of the Auditory cortex responsible for processing auditory information, such as music, differs from the sections responsible for processing visual information, although they can still be linked and active concurrently. Besides, Neurons can modify their connections by learning new information and reinforcing the connections that have been utilized more frequently, and forgetting the connections if the probability distributions of two units diverge much. The Boltzmann machine is the probabilistic inference machine for ground truth using the free energy principle. The latter has stepped further from the concept to interpret cortical responses as a fundamental of intelligent agency. With simple and random interactions of each neuron, this 'intelligent agency' could achieve sophisticated functions in a specific area of a brain. Randomness is also a vital aspect of learning since it may achieve balance and embrace regularities according to Ramsey's Theory.

1.Introduction

1.1. Importance and Objective

With Predictive coding theories becoming the basis of the current neural network, it has limitations to predict complex behavior due to the basic prior knowledge that our brain works as a Bayesian inference system 错误!未找到引用源。.

The objective of this paper is to step out of the limits of Bayesian inference systems and apply new methods of combining the probability of encoding units with the neural basis to form the network not only learning the exterior information but understand them better in a self-organized way [2] with emergent properties and stochastic forces [3-5]. Through measuring the correlations of hidden units, the probability distribution could be changed through internalization [6].

1.2. Analyze RBM and Semi-BM

RBM, Restricted Boltzmann machines have no connection between the hidden units, i.e., there are no correlations between them, which means they are independent and easier to be trained, but the hidden units still need to interact with each other to perform an internalization. When the neuron was active, not only was the surrounding area activated but so was the area that processed the related information. Instead of modifying the energy function, the probability of hidden units can better fit the meaning of correlations, so we use a loss function restriction on a semi-Boltzmann machine with hidden units connected with each other to update the weights.

1.3. Independent Feature Extraction

If a hidden unit can interpret each feature of the real data distribution more independently, it indicates that its eigenvector in the latent space of hidden units can be linearly independent, we assume that will allow them to better fit the data distribution. Since minimizing the free energy can be transformed into a problem of maximizing the information principle [7], we exploit hidden units' encoding capability to make predictions that satisfy all known conditions and make no subjective assumptions about the unknown.

This paper proposes a method of better distinguishing a hidden feature interpreter by imposing a constraint on the connections between two units. First, we measure the distances and correlations between the two hidden units. During the training, the hidden units learn their distribution $p(v)$,the factors between hidden units that are activated by the same kind of feature will be strengthened or inhibited when receiving relevant signals.

It is reasonable to predict that learning the $p(v)$ and learning to separate distinct variables will yield more valuable information. Reinforcement of feature extracting ability can also be applied to deeper networks, such as deep belief networks, and for image processing convolutional restricted Boltzmann machine, etc.

2.Background and related work

2.1. Review of Boltzmann Machines

As a stochastic generative deep learning models, RBMs are capable of learning a probability distribution from inputs [8]. It offers a wide range of applications, including dimension reduction, classification, and collaborative filtering.

They can be viewed as bipartite graphs, with visible units forming one group and hidden units forming the other group. This means that there is no interaction between visible/hidden units. In standard RBMs, the units (both visible and hidden) are binary. Below, The figures denote all visible units by $\boldsymbol{v} = \{v_1, \ldots v_m\} \in \{0,1\}^M$ and hidden units by $\boldsymbol{h} = \{h_1, \ldots h_i\} \in \{0,1\}^I$ respectively.

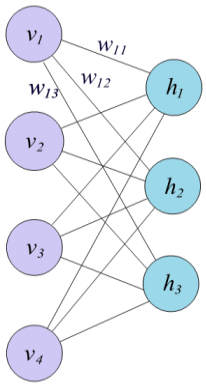

Figure 1. Restricted Boltzmann Machines

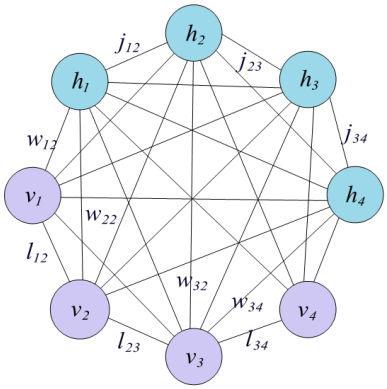

Figure 2. Boltzmann Machines

Boltzmann Machine (BM) is formed with visible units and hidden units [9] as well. A BM differs from an RBM in that there's a connection between each pair of units, meaning that it is no longer a bipartite graph. Instead, it is a complete graph. The learning algorithm could be rather slow if the network has too many layers of feature detectors with the MCMC sampling method, but it is possible to accelerate it by learning one layer at one time.

Training a Boltzmann machine takes the data from the data distribution and encodes the data to determine the probability of hidden units. The Boltzmann machine learns the true data distribution by using Maximize-log likelihood.

BM as a unidirectional graph that is formed by nodes and cliques. a factor ψ called cliques potential measures the affinity of the nodes' states. The whole graph, therefore, formed an unnormalized probability distribution [10].

$$\tilde{p}(\boldsymbol{v}) = \Pi_{C \in G(C)} \tag{1}$$

To find the actual $p(v)$,the $p(v)$needs to be divided by the 'whole energy' Z,

which is:

$$Z = \int \tilde{p}(\mathrm{v}) dv \tag{2}$$

The formal representation of $p(v)$ is:

$$p(\boldsymbol{v}) = \frac{1}{Z}\prod_{C\in\mathrm{C}} \psi_C(v_\mathrm{C}) = \frac{1}{Z} e \sum_{C\in\mathrm{C}} ln\psi_C(\mathrm{v}_C) = \frac{1}{Z} e^{-E(\mathrm{v})} \tag{3}$$

$E := \sum_{C\in C}\ ln\psi_C(x_C)$ is the energy function. For standard BM, there are weights between each unit, and the energy function is:

$$E(\boldsymbol{v},\boldsymbol{h};\boldsymbol{\theta}) = -(\boldsymbol{v}^T\boldsymbol{a}+\boldsymbol{v}^T\boldsymbol{L}\boldsymbol{v}+\boldsymbol{v}^T\boldsymbol{W}\boldsymbol{h}+\boldsymbol{h}^T\boldsymbol{J}\boldsymbol{h}+\boldsymbol{h}^T\boldsymbol{b}) \tag{4}$$

and for SBM, L vanishes, representing that they no longer have effects on each other.

$$E(\boldsymbol{v},\boldsymbol{h};\boldsymbol{\theta}) = -(\boldsymbol{v}^T\boldsymbol{a}+\boldsymbol{v}^T\boldsymbol{W}\boldsymbol{h}+\boldsymbol{h}^T\boldsymbol{J}\boldsymbol{h}+\boldsymbol{h}^T\boldsymbol{b}) \tag{5}$$

Semi-RBM doesn't have connections between visual layers but still connects two hidden units. j_{12} indicates the weights between two hidden units and both adjust by maximum-log likelihood and the constraint term.

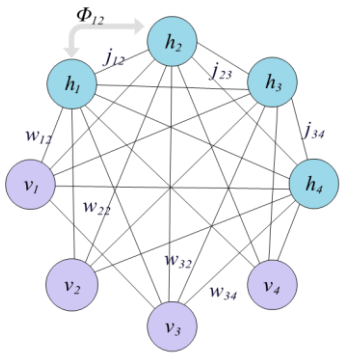

Figure 3. Semi- Boltzmann Machines with constraint

3. Definition of $\boldsymbol{\Phi}$

To quantify the correlations, there should be a factor between two hidden units. It is preferable to specify the factor using a meaningful value such as probability rather than a manually defined factor value. The factor between two hidden units acts as a regularization term and is used to constrain the Loss function while updating gradients to apply a drag force. Since the expectation of this correlation is simply the sum of hidden units' probabilities of having their value as one, the factor can be deemed zero if two hidden units are both zero or one is on and another is off. Table 1 reference from [12].

Table 1. The factor between hidden units

	$h_j = 0$	$h_j = 1$
$h_i = 0$	0	0
$h_i = 1$	0	Φ_{ij}

For the measurement**, the definition is:**

$$\Phi_{ij} = JSD\left(P(h_i = 1|\boldsymbol{v}, \boldsymbol{h}_{-i}; \boldsymbol{\theta}) \mid |P\left(h_j = 1|\boldsymbol{v}, \boldsymbol{h}_{-j}; \boldsymbol{\theta}\right)\right) \tag{6}$$

3.1. Objective function

The Objective function for this constrained Boltzmann machine is[13]:

$$argmin_\theta - logP(\boldsymbol{v}; \boldsymbol{\theta}) \Rightarrow argmax_\theta logP(\boldsymbol{v}; \boldsymbol{\theta}) + \lambda \boldsymbol{\Phi} \tag{7}$$

while in terms of this method is to maximize the distance between two distributions.

3.2. JSD

Using the simple multiplication of the probability of two hidden units can cause two problems.

The first is the failure of is simple multiplication is linear. Consider multiplication of two probability as the 'rate' of two hidden units meeting each other at the same time. The correlation s is proportional to the 'rate'. But a linear description has limits. Another failure of multiplication of two probability fails to converge with the training time becoming longer. It still has the same amount of 'acceleration' when the accuracy reaches a certain percentage which will disturb the ability to find $p(v)$ more precisely.

So, we use an approach to better interpret the relations - Jensen–Shannon divergence to solve these two problems[14].

Given two probability distributions P and Q, the JSD between them is:

$$JSD(P \mid\mid Q) = \frac{1}{2} D(P \mid\mid M) + \frac{1}{2} D(Q \mid\mid M) \tag{8}$$

where $M = \frac{1}{2}(P + Q)$ and D refers to the Kullback-Leibler divergence [15].

We also present the formula of Kullback-Leibler divergence for reference.

If distributions P and Q are discrete and defined within the same probability space, then $D(P \mid\mid Q)$ can be expressed as

$$D(P \,||\, Q) = \sum_{x \in X} P(x)\, log\left(\frac{P(x)}{Q(x)}\right). \tag{9}$$

In respect of physical significance, Kullback-Leibler divergence measures the extra bits to code samples from $p(x)$ when using a code based on $q(x)$.

Considering $P(h_i = 1|\boldsymbol{v}, \boldsymbol{h}_{-i})$ as the probability distribution of h_i and is the collection of $P(h_i = 1|\boldsymbol{v}, \boldsymbol{h}_{-i})$ of all of the data , and $P(h_j = 1|\boldsymbol{v}, \boldsymbol{h}_{-j})$ as the probability distribution of another hidden unit that is activated, the relative entropy can measure the bits the need to encode the probability distribution of hidden unit j being activated with the probability of hidden unit i being activated, and vice versa.

Maximizing the distribution's distance thus lowing the information overlapping and causing the features to distinguish from each other since Boltzmann machines use binary encoding to represent a feature on or off. Also, reducing the redundancy of information that a hidden layer can interpret means using the same amount of bits can contain more information and enhance the hidden layers' capability.

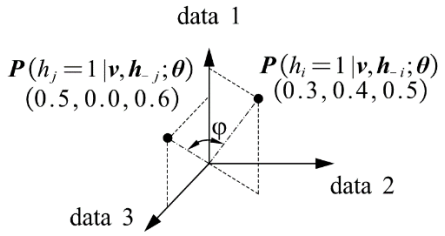

Figure 4. The geometric meaning of $\boldsymbol{P}(h_i = 1|\boldsymbol{v}, \boldsymbol{h}_{-i})$

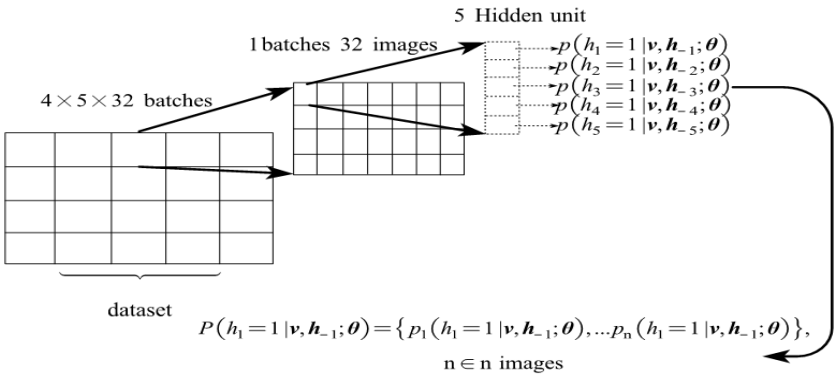

Figure 5. Probability distribution of $h_i = 1$

3.2.1. Mathematic property

1. JSD is symmetric, and the weights in J matrices are also symmetric.

2. JSD can stop updating gradients when the distributions are separate. This gives us the convenience that when the model has reached certain accuracy, it tends to update the original derivative of $p(v)$.

3.2.2. Derivative

When doing the derivatives of J, we consider J_{ij} is different from J_{ji}, but they still are equal, thus only calculating partial derivatives of Φ with respect to J_{ij} and ignoring J_{ji} can give more convenience.

Here is the method to calculate derivatives of J with respect to J_{ij}not J_{ji}. Because $\frac{\partial \Phi_{\mathrm{ij}}}{\partial J_{ij}} = \frac{\partial \Phi_{ji}}{\partial J_{ji}}$, only calculating $\frac{\partial \Phi_{ij}}{\partial J_{ij}}$ would be possible. So:

$$\frac{\partial p\left(h_j = 1|\boldsymbol{v}, \boldsymbol{h}_{-j}; \boldsymbol{\theta}\right)}{\partial J_{ij}} = 0 \tag{11.0}$$

And also:

$$\frac{\partial log p\left(h_j = 1|\boldsymbol{v}, \boldsymbol{h}_{-j}; \boldsymbol{\theta}\right)}{\partial J_{ij}} = 0 \tag{11.1}$$

$$\begin{aligned}\frac{\partial \Phi_{ij}}{\partial J_{ij}} = &\frac{1}{2}\frac{\partial p(h_i = 1|\boldsymbol{v},\boldsymbol{h}_{-i}; \boldsymbol{\theta}) \cdot log\dfrac{p(h_i = 1|\boldsymbol{v}, \boldsymbol{h}_{-i}; \boldsymbol{\theta})}{p(h_i = 1|\boldsymbol{v}, \boldsymbol{h}_{-i}; \boldsymbol{\theta}) + p\left(h_j = 1|\boldsymbol{v}, \boldsymbol{h}_{-j}; \boldsymbol{\theta}\right)}}{\partial J_{ij}} \\ &+\frac{1}{2}\frac{\partial p\left(h_j = 1|\boldsymbol{v},\boldsymbol{h}_{-j}; \boldsymbol{\theta}\right) \cdot log\dfrac{p\left(h_j = 1|\boldsymbol{v},\boldsymbol{h}_{-j}; \boldsymbol{\theta}\right)}{p(h_i = 1|\boldsymbol{v}, \boldsymbol{h}_{-i}; \boldsymbol{\theta}) + p\left(h_j = 1|\boldsymbol{v}, \boldsymbol{h}_{-j}; \boldsymbol{\theta}\right)}}{\partial J_{ij}}\end{aligned}$$

$$= \frac{1}{2} h_j \cdot \left(log\frac{\sigma(h_i)}{\sigma(h_i) + \sigma\left(h_j\right)}\right)\sigma(h_i)\left(1-\sigma(h_i)\right) \tag{11.2}$$

Then first write down the convenient notation of the *JSD,* it can use $\sigma(h_i)$ to represent $p(h_i = 1|\boldsymbol{v},\boldsymbol{h}_{-i}; \boldsymbol{\theta})$.

Thus, equation (6)

$$= \frac{1}{2}\sum_{h_i,h_j}\sigma(h_i)log\frac{\sigma(h_i)}{\sigma(h_i)+\sigma(h_j)}+\frac{1}{2}\sum_{h_i,h_j}\sigma(h_j)log\frac{\sigma(h_j)}{\sigma(h_i)+\sigma(h_j)}+log2 \tag{11.3}$$

For a naive version of this probability distribution, it can simply take the mean of the first dimension as the value of the probability matrix, so the naive derivative of Φ_{ij} is:

$$\frac{\partial\Phi_i}{\partial W_{mi}} = \frac{1}{2}\, v_m \cdot \sigma(h_i)\big(1-\sigma(h_i)\big)log\frac{\sigma(h_i)}{\sigma(h_i)+\sigma(h_j)} \tag{12}$$

$$\frac{\partial\Phi_{ij}}{\partial b_i} = \frac{1}{2}\sigma(h_i)\big(1-\sigma(h_i)\big)log\frac{\sigma(h_i)}{\sigma(h_i)+\sigma(h_j)} \tag{13}$$

Besides, for saving the computation resources, the derivative will be rewritten in form of the matrix.

Here is an example: if an SBM has three visible units and five hidden units, then the matrix of probabilities of hidden units has the $N \times I$ shape matrix (N represents data's number, and I represents hidden units' number). It should be written in form of below:

$$\boldsymbol{P_h} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix} \in \mathbb{R}^{I\times 1}, \boldsymbol{P_h^T} = [p_1\ \ p_2\ \ p_3\ \ p_4\ \ p_5], \tag{14.0}$$

To obtain the same shape as the weights' matrix, we did the following tiling operation:
$T\,(Vectors, 1)$ represents the Matrix was repeated in row dimension as tile function and $mean(Vectors, 1)$ represents the Matrix's mean value along row dimension.

$$T(\boldsymbol{P_h}, 1) = \boldsymbol{P_H} = \begin{bmatrix} - \boldsymbol{p_1} - \\ \vdots \\ - \boldsymbol{p_5} - \end{bmatrix} \in \mathbb{R}^{I\times I} \tag{14.1}$$

$$T(\boldsymbol{P_h}, 0) = \boldsymbol{P_H^T} = \begin{bmatrix} \boldsymbol{p_1^T}\ \ \boldsymbol{p_2^T}\ \boldsymbol{p_3^T}\ \ \boldsymbol{p_4^T}\ \boldsymbol{p_5^T} \end{bmatrix} \in \mathbb{R}^{I\times I} \tag{14.2}$$

$$T(\boldsymbol{P_h}, 1) + T(\boldsymbol{P_h}, 0) = \boldsymbol{P_H} + \boldsymbol{P_H^T} \tag{14.3}$$

$$\boldsymbol{J} = \begin{bmatrix} J_{11} & \cdots & J_{15} \\ \vdots & \ddots & \vdots \\ J_{51} & \cdots & J_{55} \end{bmatrix}, J_{ii} = 0, \boldsymbol{J} \in \mathbb{R}^{I\times I} \tag{14.4}$$

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, T(\boldsymbol{v}, 1) = \boldsymbol{V} = \begin{bmatrix} v_1 & \cdots & v_1 \\ v_2 & \cdots & v_2 \\ v_3 & \cdots & v_3 \end{bmatrix}, \boldsymbol{V} \in \mathbb{R}^{M\times I}, \tag{14.5}$$

$$\boldsymbol{h} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{bmatrix}, T(\boldsymbol{h}, 1) = \boldsymbol{H} = \begin{bmatrix} - \boldsymbol{h_1} - \\ \vdots \\ - \boldsymbol{h_5} - \end{bmatrix}, \tag{14.6}$$

$$T(\boldsymbol{h}, 1) = \begin{bmatrix} \boldsymbol{h_1^T} & \boldsymbol{h_2^T} & \boldsymbol{h_3^T} & \boldsymbol{h_4^T} & \boldsymbol{h_5^T} \end{bmatrix} = \boldsymbol{H^T} \tag{14.7}$$

$$\boldsymbol{mean}\left(log\left(\frac{\boldsymbol{P_H}}{\boldsymbol{P_H} + \boldsymbol{P_H^T}} \right), 1 \right) = \boldsymbol{mean} \begin{bmatrix} \log\left(\frac{\sigma(h_1)}{\sigma(h_1) + \sigma(h_1)} \right) & \cdots & log\left(\frac{\sigma(h_1)}{\sigma(h_1) + \sigma(h_5)} \right) \\ \vdots & \ddots & \vdots \\ \log\left(\frac{\sigma(h_5)}{\sigma(h_5) + \sigma(h_1)} \right) & \cdots & log\left(\frac{\sigma(h_5)}{\sigma(h_5) + \sigma(h_5)} \right) \end{bmatrix} \tag{14.8}$$

So, the final gradient updating constrain term is:

$$\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{J}} = \frac{1}{2} \cdot \mathbf{H}^T \odot \boldsymbol{P_H} \odot (1 - \boldsymbol{P_H}) \odot log\left(\frac{\boldsymbol{P_H}}{\boldsymbol{P_H} + \boldsymbol{P_H}^T} \right) \tag{15}$$

$$\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{W}} = \frac{1}{2} \boldsymbol{V} \circ \boldsymbol{P_H} \circ (\boldsymbol{1} - \boldsymbol{P_H})^{\boldsymbol{T}} \circ \boldsymbol{mean}\left(log\left(\frac{\boldsymbol{P_H}}{\boldsymbol{P_H} + \boldsymbol{P_H^T}} \right), 1 \right)^T \tag{16}$$

$$\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{b}} = \frac{1}{2} \boldsymbol{P_h} \circ (\boldsymbol{1} - \boldsymbol{P_h}) \circ \boldsymbol{mean}\left(log\left(\frac{\boldsymbol{P_H}}{\boldsymbol{P_H} + \boldsymbol{P_H^T}} \right), 1 \right)^T \tag{17}$$

3.3. Algorithm

Training an SBM is much the same as training a conventional RBM or BM. It is possible to use the same approach of constructing a loss function and updating the gradients of a deep belief network that contains two or more RBMs.

The Gradient will be updated in the following formula [16-17]:

$$\Delta \boldsymbol{W} \leftarrow \varepsilon \left(E_{p_{data}}[\boldsymbol{v}^T \boldsymbol{h}] - E_{p_{model}}[\boldsymbol{v}^T \boldsymbol{h}] + \lambda \cdot \frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{W}} \right) \tag{18}$$

$$\Delta \boldsymbol{b} \leftarrow \varepsilon \left(E_{p_{data}}[\boldsymbol{h}^T] - E_{p_{model}}[\boldsymbol{h}^T] + \lambda \cdot \frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{b}} \right) \tag{19}$$

$$\Delta \boldsymbol{J} \leftarrow \varepsilon \left(E_{p_{data}}[\boldsymbol{h}^T \boldsymbol{h}] - E_{p_{model}}[\boldsymbol{h}^T \boldsymbol{h}] + \lambda \cdot \frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{J}} \right) \tag{20}$$

where ε is the learning rate and the $E_{p_{data}}$ means the expectation of units from data probability distribution, $E_{p_{model}}$ is the expectation of model probability distribution respectively.

3.4. Improving accuracy

To achieve a full accuracy of JSD measurement, it can be deprived of the mean operator when calculating the probability matrix $\boldsymbol{P_h}$. Thus, the sum of total divergence of the probability of hidden unit I being activated and the probability distribution of hidden unit j being activated could be maintained, but the price of accuracy is the storage and computing cost to perform more operations of multidimensional matrix's transposition and multiplication.

We proposed the calculation progress as follows:

1. Figure 5 has showed how the probability of hidden units' activation form a probability distribution. The probability of hidden when unit i is activated and the matrix $\boldsymbol{H}$ are $N \times I$ shape matrixes. The object shape of $\boldsymbol{\Phi}$ is $N \times I \times I$ so at final state is to summate the first dimension of $\boldsymbol{P}(h_i = 1|\boldsymbol{v}, \boldsymbol{h}_{-i})$.

2. For example, a dataset contains 60000 images, 144 hidden units and 784 visual units, so $N = 60000, M = 784$ and $I = 144$.Formula 11.2 is the result of the summation of all the i , j and the whole data's probability. When calculate the final $\boldsymbol{\Phi}$, it needs to preserve every single data's activation information.

There is an illustration of the data's dimension in Figure 6:

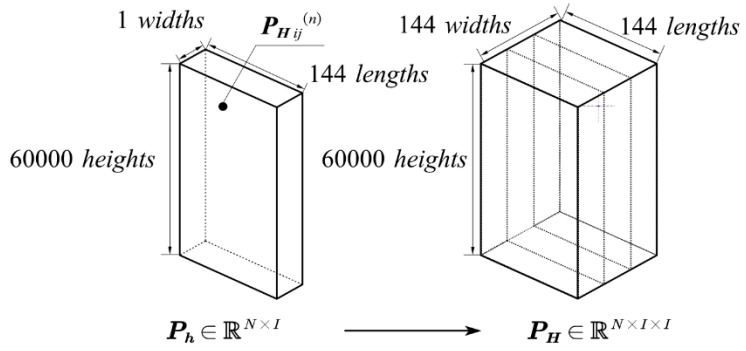

Figure 6. illustration of expanded $\boldsymbol{P_h}$ with respect to J

3.To calculate each pair of hi and hj, the multi-dimension probability matrix could be transposed by alternating coordinate axes along the second and third dimensions but still maintain the first dimensions unchanged. The position of the element on n data, i rows and j columns which denoted as $\boldsymbol{P_{H}}_{ij}^{(n)}$could be transposed to $\boldsymbol{P_{H}}_{ji}^{(n)}$, samely with $\boldsymbol{H}$.

4. Experiment and comparison

When using JSD update, SBM tends to converge to better accuracy than the original SBM. With only three layers and without too many hidden units in each layer, JSD constrain can enhance a model's accuracy and capability to interpret features.

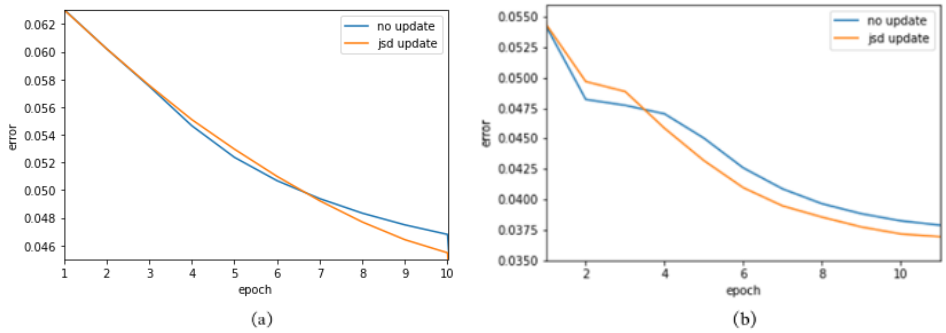

Figure 7. error of SBM trained on CIFAR-10(a) and MNIST(b) dataset

Table 2. SBM on reconstruction

Model/Task on reconstruction	Loss on MNIST	Loss on FASHION- MNIST
SBM	0.047	0.06
SBM with JSD	0.045	**0.05**

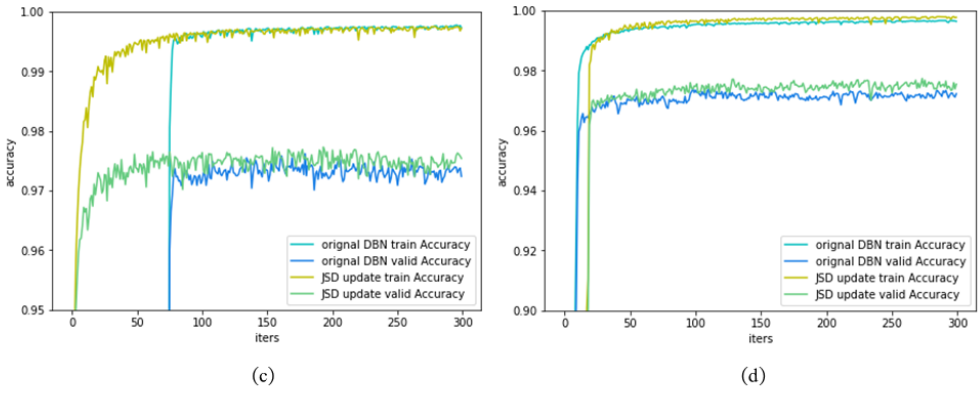

Figure 8. modified-DBN on classification on mnist (c) and fashion-mnist(d)

JSD constrain can better separate the main feature structure (a)with the original SBM (b)which tends to mix the encoding information together. ⋆The figure 2 references from [18-19].

Table 3. modified-DBN on classification

Model/Task on classification	Accuracy on MNIST	Accuracy on FASHION-MNIST
DBN (3 semi-BM)	97.24	97.15
DBN with JSD	97.41	97.59

5. Conclusion

With the experiment result, we have proved that the learning process of finding a probability distribution of the Boltzmann Machine to some degree was connected to the ability to encode the information of hidden units. Using the constraint methods on hidden units, we can manually adjust the probability distributions on hidden layers and separate the probability distribution of each activated unit to maximize the informational entropy.

Further study can be achieved by exploring the latent space of hidden units and using vectors to better define a hidden unit's position in multidimensional space. This can measure the geometry distance thus the connections between hidden units can be reinforced using clustering of a single area's unit or dispersing units in a different area. Also, instead of Euclidean Distance, the methods can be expanded with tangent distance in the manifold to construct a bridge between different areas without losing the information of the distance in the hidden units' tangent space.

Furthermore, these experiments are done on the original SBM and DBN, but image processing can achieve a higher level with structured features represented using CRBM and CDBN.

Reference

[1] Huang, Y., & Rao, R. P. (2011). Predictive coding. Wiley Interdisciplinary Reviews: Cognitive Science, 2(5):580-593.

[2] Haken, H. (1983). An introduction nonequilibrium phase transitions and self-organization in physics. *Chemistry, and Biology*, 206-216.

[3] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences, 79*(8):2554-2558.

[4] Jirsa, V. , & Sheheitli, H. . Entropy, free energy, symmetry and dynamics in the brain. *Journal of Physics*: Complexity.

[5] Ramsey, F. P. (2009). On a problem of formal logic. *In Classic Papers in Combinatorics* (pp. 1-24). Birkhäuser Boston.

[6] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39*(1):1-22.

[7] Pinkard, H., & Waller, L. (2022). A visual introduction to information theory. *arXiv preprint*

arXiv:2206.07867.

[8] Hinton, G. (2010, August 2). *A practical guide to training restricted boltzmann machines*. Retrieved from https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf

[9] Hinton, G. E. (2007). *Boltzmann machine.* Scholarpedia, 2(5):1668.

[10] Fischer, A., & Igel, C. (2012, September). An introduction to restricted Boltzmann machines. In: *Iberoamerican congress on pattern recognition.* Springer, Berlin, Heidelberg.pp.14-36.

[11] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning. MIT Press*. Retrieved from http://www.deeplearningbook.org

[12] Li, M. (2018). Generalized Lagrange multiplier method and KKT conditions with an application to distributed optimization. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(2):252-256.

[13] Everett III, H. (1963). Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations research*, *11*(3): 399-417.

[14] Fuglede, B., & Topsoe, F. (2004, June). Jensen-Shannon divergence and Hilbert space embedding. In: *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.* p.31.

[15] Van Erven, T., & Harremos, P. (2014). Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, *60*(7):3797-3820.

[16] Salakhutdinov, R., Mnih, A., & Hinton, G. (2007, June). Restricted Boltzmann machines for collaborative filtering. *In:Proceedings of the 24th international conference on Machine learning* pp. 791-798.

[17] Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). *A learning algorithm for Boltzmann machines. Cognitive science*, 9(1):147-169.

[18] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786): 504-507.

[19] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):27-1554.