ON THE IMPORTANCE OF HIDDEN BIAS AND HIDDEN ENTROPY IN REPRESENTATIONAL EFFICIENCY OF THE GAUSSIAN-BIPOLAR RESTRICTED BOLTZMANN MACHINES

by

Altynbek Isabekov

A Thesis Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Electrical & Electronics Engineering

Koç University

March, 2018

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a Ph.D. thesis by

Altynbek Isabekov

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assoc. Prof. Dr. Engin Erzin

Prof. Dr. Alper Erdoğan

Prof. Dr. Yücel Yemez

Prof. Dr. Murat Saraçlar

Assoc. Prof. Dr. Burak Acar

Date: _____

To my family

ABSTRACT

With development of machine learning and deep learning fields, the importance of unsupervised learning algorithms also increases. One of these algorithms is Gaussian-Bernoulli Restricted Boltzmann Machines (GBLRBMs), which are capable of modelling real-valued data. Moreover, GBLRBMs are used to pretrain weights in artificial neural networks, which improves performance of these networks. In this work, we analyze the role of hidden bias in representational efficiency of the Gaussian-Bipolar Restricted Boltzmann Machines (GBPRBMs), which are similar to the widely used Gaussian-Bernoulli RBMs. Our experiments show that hidden bias plays an important role in shaping of the probability density function of the visible units. Correspondingly, we define hidden entropy and propose it as a measure of representational efficiency of the model. By using this measure, we investigate the effect of hidden bias on the hidden entropy and provide a full analysis of the hidden entropy as function of the hidden bias for small models with up to three hidden units. We also provide an insight into understanding of the representational efficiency of the larger scale models. Furthermore, we introduce Normalized Empirical Hidden Entropy (NEHE) as an alternative to hidden entropy that can be computed for large models. Experiments on the MNIST, CIFAR-10 and Faces data sets show that NEHE can serve as measure of representational efficiency and gives an insight on minimum number of hidden units required to represent the data.

ÖZETÇE

Son zamanlarda yapay öğrenme ve derin öğrenme alanlarının yaygınlaşmasıyla eğiticisiz öğrenme yöntemlerinin önemi artmaktadır. Gauss-Bernoulli Kısıtlı Boltzmann Makineleri (KBM) gerçel veriyi modelleyebilme özelliğine sahip bu yöntemlerden birisi olarak yapay sinir ağlarının ağırlıklarının ön eğitiminde kullanılmaktadır. Bu tezde Gauss-Bernoulli KBM'lerine benzer Gauss-İki Kutuplu KBM'lerinin betimleyici verimliliğinde gizli yanlılıkların rolünü inceledik. Çalışmalarımızda gizli yanlılıkların görünür birimlerin olasılık yoğunluk işlevinin şekillendirilmesinde önemli rol oynadığı gördük. Bu doğrultuda, modelin betimleyici verimliliği için yeni bir gizli entropi ölçütü tanımladık. Bu ölçütü kullanarak, gizli yanlılıkların gizli entropiye olan etkisini inceledik. Ayrıca en fazla üç gizli birim içeren küçük modellerin gizli entropinin gizli yanlılıkların cinsinden tam analizini sunduk. Daha büyük modeller için betimleyici verimliliğin nasıl davrandığını incelemek için gizli entropiyi yaklaşıklayan Normalize Edilmiş Görgül Entropi (NEGE) ölçütünü tanımladık. MNIST, CIFAR-10 ve Yüzler veri kümesi üzerinde yapılan deneyler, bu yaklaştırımın betimleyici verimliliğin ölçüsü olarak kullanılabileceğini ve veri kümesini tasvir edebilmek için gereken asgari gizli birim sayısı hakkında fikir verebileceğini göstermektedir.

ACKNOWLEDGMENTS

I wish to express my genuine gratitude to my advisor, Assoc. Prof. Dr. Engin Erzin. His insight, experience, guidance and encouragement were very helpful in this study. I consider myself to be truly fortunate to have an opportunity to work with him. I am very thankful to members of my Ph.D. thesis monitoring committee, Prof. Dr. Alper T. Erdoğan and Prof. Dr. Yücel Yemez for the provided criticism and help. Besides, I would like to show my appreciation to the thesis jury members Prof. Dr. Murat Saraçlar and Assoc. Prof. Dr. Burak Acar for their revision and commentary. Also I would like to thank all my friends and colleagues from offices ENG 143 and MVGL for supporting me during my graduate years in the university.

On a personal note, I express my gratitude to my family for their lifetime support, encouragement and love.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

Recently, the subject of Restricted Boltzmann Machines (RBMs) and deep learning became the focus of attention in machine learning research. Application of deep learning in different areas such as image processing, computer vision, and natural language processing has proved its efficiency [1, 2, 3]. RBMs are probabilistic generative models which are used to obtain new (usually compressed) representation of the data. Different types of RBMs are used as building blocks for deep neural architecture by means of unsupervised layer-wise pre-training [4]. However, RBMs with real-valued inputs are of primarily importance as most of the analyzed data is real-valued. Conventional Bernoulli-Bernoulli RBMs have been studied in [5, 6] and [7] where they are referred as universal approximators of any binary distribution. One of the first Gaussian-Bernoulli RBM (GBLRBM) models with real-valued inputs was proposed in [4] and [8], and was explicitly analyzed in [1]. Another version of a GBLRBM with a more intuitive energy function was proposed in [9]. Moreover, a more simplified sub-type of the latter model was analyzed in [10] and [11].

Despite the ongoing research in this field, still not much is known about the principle of operation of GBLRBMs. Combinatorial nature of the model makes the analysis even harder. Nevertheless, conceptual understanding of GBLRBMs is given in [10]. The thesis has a well-described comparison to a Gaussian mixture model and a good visualization of the modelled distribution that gives an insight into the principle of operation of GBPRBMs. However, the thesis lacks analysis of hidden bias and its effect on the modelled probability density function. The effect of the biases and the mean of the data on the learning process was investigated in [12] and [13]. Visible and hidden offsets are used to center the RBM model and make learning more stable.

Another interesting visualization of RBMs is given in [14]. Debugging of the RBMs is done by visualizing weight parameters as a tensor in a cube filled with small cells. Evaluation

of histograms of the parameters on the mini-batch helps finding optimal stopping point for the training process. Disappearance of Gaussian-like shapes of the histograms indicates that training has converged to a stationary phase. This phenomenon was analyzed in [15]. A measure of non-Gaussianity based on negentropy and excess kurtosis was proposed as a stopping criterion for the training.

The problem of measuring usefulness of the hidden neurons was investigated in [7, 6] and [16]. The first two papers describe the effect of augmenting hidden layer on the representational efficiency of Bernoulli-Bernoulli RBMs. In the latter paper, mutual information between visible and hidden units is suggested as a measure of relevant activity of the hidden units. Usefulness of the hidden neurons is also tested by pruning neurons after training and by adding neurons during training. The results show that models initialized with a large number of hidden units can be simplified by pruning neurons without decreasing classification performance.

Nowadays, most of the research in deep learning is concentrated on application of RBMs and speeding up the training process. Fundamental questions remain still unanswered. What is a good measure for usefulness of the hidden neurons? How does the hidden bias affect the representational efficiency of the RBM model? What is the number of hidden neurons needed to represent the data? We try to answer these questions by introducing a new Gaussian-Bipolar RBM (GBPRBM) model, in which we investigate representational efficiency of hidden units in defining distribution of the visible units. This model is very similar to Gaussian-Bernoulli RBM except that it has a more symmetrical geometry which facilitates hidden entropy analysis described in Section 3.

Our contributions are summarized as follows:

- In Section 3, we define hidden entropy function and propose it as a measure of representational efficiency of GBPRBM models. We demonstrate how hidden bias shapes probability distribution of visible units. Moreover, we present a list of conditions needed to attain maximum hidden entropy. Also we provide a full analysis of the hidden entropy function for models with up to three hidden units. In this analysis, regions of high hidden entropy are given analytically in terms of other model parameters. This analysis provides an intuition to visualize hidden entropy space in higher dimensions.

- In Section 4, we propose a technique to measure activations of hidden units by defining Normalized Empirical Hidden Entropy (NEHE) function as an upper bound to the hidden entropy. This function allows to analyze models with higher number of hidden units. By measuring NEHE on the GBPRBM models trained using MNIST, CIFAR-10 and Faces data sets, we illustrate how number of hidden units affects representational efficiency of the GBPRBM models. This experiment gives an insight on the minimum number of hidden units needed to represent the data.

Findings and derivations in the paper are presented using examples. The reference GBPRBM model given in Section 2.1 and its derivative models with smaller number of hidden and visible units are used in visualization of the probability of visible units and the hidden entropy function.

Chapter 2

GAUSSIAN-BIPOLAR RESTRICTED BOLTZMANN MACHINES

Gaussian-Bipolar Restricted Boltzmann Machine (GBPRBM) is an undirected graphical model which is used to model relation between *visible* and *hidden* units in a probabilistic way. GBPRBMs have real-valued inputs in the visible layer and binary units in the hidden layer.

Let the input vector with real-valued visible units be of size V such that $\mathbf{v} = [v_1\ v_2\ \dots\ v_V]^T$. Binary hidden units are constrained to have antipode values $\{-1, 1\}$ and grouped into a column vector $\mathbf{h} = [h_1\ h_2\ \dots\ h_H]^T$ with H being the number of hidden units. For notational consistency, visible units are represented by vectors $\mathbf{v}, \mathbf{u}$, hidden units - by vectors $\mathbf{f}, \mathbf{h}, \mathbf{g}$ throughout the paper. Subscripts i, j are reserved for visible and hidden units, respectively.

Two more parameters are associated with visible units. The first one is visible bias term b_i^v and the second one is visible variance term σ_i where $i \in \{1, \dots, V\}$. Bias terms are also present in the hidden units as: $b_j^h,\ j \in \{1, \dots, H\}$. Visible and hidden units are connected using weights $w_{ij},\ i \in \{1, \dots, V\},\ j \in \{1, \dots, H\}$. The relationship between visible and hidden units is described by the energy function

$$E(\mathbf{v}, \mathbf{h}) = \frac{1}{2}(\mathbf{v} - \mathbf{b}_v)^T \mathbf{\Sigma}^{-1}(\mathbf{v} - \mathbf{b}_v) - \mathbf{v}^T \mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{h} - \mathbf{b}_h^T \mathbf{h}, \tag{2.1}$$

which is defined similarly for the Gaussian-Bernoulli RBM model in [9] with parameters

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,H} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,H} \\ \vdots & \vdots & \ddots & \vdots \\ w_{V,1} & w_{V,2} & \cdots & w_{V,H} \end{bmatrix}, \ \mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_V^2 \end{bmatrix},$$

$$\mathbf{b}_v = \left[b_1^v, b_2^v, \dots, b_V^v\right]^T, \qquad \mathbf{b}_h = \left[b_1^h, b_2^h, \dots, b_H^h\right]^T. \tag{2.2}$$

The energy function is used to define joint probability density function (pdf) of $\mathbf{v}$ and $\mathbf{h}$

$$p(\mathbf{v}, \mathbf{h}) = \frac{\exp(-E(\mathbf{v}, \mathbf{h}))}{\int_{\mathbf{u}} \sum_{\mathbf{g}} \exp(-E(\mathbf{u}, \mathbf{g}))d\mathbf{u}}, \tag{2.3}$$

where $\int_{\mathbf{u}}(\dots)d\mathbf{u}$ is integration over all space of visible units and $\sum_{\mathbf{g}}$ is summation over all 2^H configurations of hidden vector $\mathbf{g}$. Likewise, conditional probability of the visible vector $\mathbf{v}$ given hidden vector $\mathbf{h}$ is defined as

$$\begin{aligned} p(\mathbf{v}|\mathbf{h}) &= \frac{p(\mathbf{v},\mathbf{h})}{p(\mathbf{h})} = \frac{\exp(-E(\mathbf{v},\mathbf{h}))}{\int_{\mathbf{u}}\exp(-E(\mathbf{u},\mathbf{h}))d\mathbf{u}} \\ &= \mathcal{N}\left(\mathbf{v};\ [\mathbf{b}_v + \mathbf{W}\mathbf{h}],\ \mathbf{\Sigma}\right), \end{aligned} \tag{2.4}$$

where $\mathcal{N}\left(\mathbf{v};\ \boldsymbol{\mu},\mathbf{\Sigma}\right)$ is a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$. Since $\mathbf{\Sigma}$ is diagonal, the conditional pdf can be represented as product of marginal conditional pdfs of each visible unit:

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{V}\mathcal{N}\left(v_i;\ [b_i^v + \mathbf{W}_{(i,:)}\mathbf{h}],\ \sigma_i^2\right) = \prod_{i=1}^{V} p(v_i|\mathbf{h}). \tag{2.5}$$

Detailed derivations of $p(\mathbf{v}|\mathbf{h})$ can be found in A.1.

2.1 Data Modeling Using Probability of Visible Units

Restricted Boltzmann machines have been used as unsupervised learning algorithms to extract latent features and to model the data distribution. This corresponds to clustering in the space of visible units and encoding each cluster using hidden units. Nevertheless, RBMs are probabilistic models, and a more straightforward interpretation of the data modeling is representing the data distribution as probability of visible units $p(\mathbf{v})$. The proposed GBPRBM has a probability of visible units given as

$$\begin{aligned} p(\mathbf{v}) &= \sum_{\mathbf{h}} p(\mathbf{h})p(\mathbf{v}|\mathbf{h}) \\ &= \sum_{\mathbf{h}} p(\mathbf{h}) \times \mathcal{N}\left(\mathbf{v};\ [\mathbf{b}_v + \mathbf{W}\mathbf{h}],\ \mathbf{\Sigma}\right). \end{aligned} \tag{2.6}$$

This implies that a GBPRBM models the probability of observing $\mathbf{v}$ as a Gaussian Mixture Model (GMM). Every Gaussian component with covariance matrix $\mathbf{\Sigma}$ is scaled by mixture weight $p(\mathbf{h})$ and located at $[\mathbf{b}_v + \mathbf{W}\mathbf{h}]$.

Visualizations of $p(\mathbf{v})$ of submodels with dimension V equal to 1, 2 and 3 are shown in Figure 2.1. In order to exemplify and visualize the underlying spaces, a sample geometry is

set with model parameters of three visible units as:

$$\mathbf{W}_R = \begin{bmatrix} 10 & 6 & 2 \\ -6 & 4 & -2 \\ 1 & 3 & 5 \end{bmatrix}, \qquad \mathbf{b}_h^R = \begin{bmatrix} -57.3333 \\ -25.3333 \\ 15.5556 \end{bmatrix}, \tag{2.7}$$
$$\mathbf{\Sigma}_R = 1.5^2 \mathbf{I}_V, \qquad \mathbf{b}_v^R = \begin{bmatrix} 8 & 5 & 3 \end{bmatrix}^T,$$

where $\mathbf{I}_V$ is an identity matrix of size $(V \times V)$. Sub- and superscripts R denote reference model, whose derivatives will be used throughout the paper. Weight matrix $\mathbf{W}$ and visible bias $\mathbf{b}_v$ define geometry for the model. Covariance matrix $\mathbf{\Sigma}$ determines shape of the Gaussian components, which can be also considered as a geometrical parameter. On the other hand, hidden bias $\mathbf{b}_h$ controls expression of the Gaussian components. Figure 2.1(c) shows 5000 samples drawn according to $p(\mathbf{v})$ given in (2.7). The model's geometry is also outlined on the same plot. Position of Gaussian components is labeled by values of the hidden vector $[h_1, h_2, h_3]$. Planes perpendicular to the weights represent decision boundaries for $p(h_j|\mathbf{v})$ and will be discussed in the subsequent sections. The main observation here is that for the given value of $\mathbf{b}_h$, only four Gaussian components are expressed. This value of $\mathbf{b}_h$ is chosen in a way that the maximum number of the components are activated for a given geometry and covariance matrix. Magnitudes of the other components are negligibly small.

If we reduce number of visible units to two $(V = 2)$, then the new $p(\mathbf{v})$ will resemble a projection of the old $p(\mathbf{v})$ with three visible units into the space of the first two visible units (see Figure 2.1(b)). In this case, new model parameters are defined by

$$\mathbf{W} = \mathbf{W}_R(1{:}2, :),\ \mathbf{b}_v = \mathbf{b}_v^R(1{:}2),\ \mathbf{\Sigma} = \mathbf{\Sigma}_R(1{:}2, 1{:}2), \tag{2.8}$$

where "1:2" ("1 to 2") and ":" ("all") denote indices of the matrices and vectors. Since geometry has changed, the value of hidden bias $\mathbf{b}_h$ should be changed as well so that the same Gaussian components are activated. In such a scenario, the recomputed value of $\mathbf{b}_h$ is given as

$$\mathbf{b}_h^C = [-52.4444,\ -16.0000,\ 13.3333]^T. \tag{2.9}$$

This value should be taken as is for now, because equations according to which it was computed is given in Section 3.5. Superscript C will be useful to differentiate this point in the space of hidden bias in the following sections. If the old value of $\mathbf{b}_h$ were used, just only

one Gaussian component would be active. This shows how $p(\mathbf{v})$ is sensitive to perturbations in the hidden bias. Similarly to the three-dimensional case, the model's geometry, Gaussian components' labels and decision boundaries for $p(h_j|\mathbf{v})$ are shown on the same plot.

Further reducing number of visible units to one, yields $p(\mathbf{v})$ show in Figure 2.1(a). Likewise, new model parameters are defined by

$$\mathbf{W} = \mathbf{W}_R(1,:),\ b_v = \mathbf{b}_v^R(1),\ \sigma^2 = \mathbf{\Sigma}_R(1,1), \tag{2.10}$$

and hidden bias is set to $\mathbf{b}_h = [-71.1111,\ 0,\ 7.1111]^T$ such that the same Gaussian components are expressed. Hidden vectors, which determine position of these components, are shown under leafs of the binary tree and has the following format: $[h_1, h_2, h_3]^T$. Widths of the leafs represent columns of the matrix $\mathbf{W}$. Note that Gaussian components encoded by $[-1,-1,-1]^T$ and $[-1,-1,+1]^T$ are subject to merging due to superposition. Such components which are close to each other and which have large enough σ_v^2 tend to combine and look like a single Gaussian distribution.

2.2 Interpretation of Hidden Units

Suppose that for a given training data set of visible vectors $\{\mathbf{v}_s,\ s \in D\}$ some suitable GBPRBM model parameters $\{\mathbf{W}, \mathbf{\Sigma}, \mathbf{b}_v, \mathbf{b}_h\}$ were estimated. In this case, another interpretation of the GBPRBM operation is vector quantization. Centroid coordinates are given by $[\mathbf{b}_v + \mathbf{W}\mathbf{h}]$ where $\mathbf{h}$ encodes the path to the centroid. Given visible vector $\mathbf{v}$, the codeword of the centroid, which is most likely to model the vector $\mathbf{v}$, can be found by using conditional pdf $p(\mathbf{h}|\mathbf{v})$ which is defined as

$$\begin{aligned} p(\mathbf{h}|\mathbf{v}) = \frac{p(\mathbf{v},\mathbf{h})}{p(\mathbf{v})} &= \prod_{j=1}^{H} p(h_j|\mathbf{v}) \\ &= \prod_{j=1}^{H} \frac{\exp\left(h_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij} + b_j^h\right)\right)}{2\cosh\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij} + b_j^h\right)}. \end{aligned} \tag{2.11}$$

It follows that every hidden unit of the codeword can be encoded separately using these two equations

$$\begin{aligned} p(h_j = -1|\mathbf{v}) &= \text{sigm}\left(-2\left(\mathbf{v}^T\mathbf{\Sigma}^{-1}\mathbf{W}(:,j) + b_j^h\right)\right), \\ p(h_j = +1|\mathbf{v}) &= \text{sigm}\left(+2\left(\mathbf{v}^T\mathbf{\Sigma}^{-1}\mathbf{W}(:,j) + b_j^h\right)\right), \end{aligned} \tag{2.12}$$

where "sigm" is the sigmoid function. Argument of this function is a hyperplane depicted by $\mathbf{v}^T\mathbf{\Sigma}^{-1}\mathbf{W}(:,j) + b_j^h = 0$. It is a decision boundary for two V-dimensional half-spaces pertaining to $h_j = +1$ and $h_j = -1$. Let us consider a case with $V = 3$ and $\mathbf{\Sigma} = \sigma^2\mathbf{I}$, then $\mathbf{v}^T\mathbf{W}(:,j) + \sigma^2 b_j^h = 0$ is a plane perpendicular to weight $\mathbf{W}(:,j)$. Orthogonality of this decision boundary is shown in Figure 2.1(c). For $V = 2$, decision boundary is a line perpendicular to weight $\mathbf{W}(:,j)$ as seen in Figure 2.1(b). For the smallest dimension ($V = 1$), decision boundary becomes a point as shown in Figure 2.1(a). Hidden bias plays an important role by setting position of the hyperplane in this V-dimensional space. A more detailed derivation of $p(\mathbf{h}|\mathbf{v})$ can be found in A.2.

2.3 Relationship Between the GBPRBM and the GBLRBM Models

The intoduced GBPRBM architecture is similar to the GBLRBM model. The only difference in definition of these models is that a hidden unit can take values $\{-1, +1\}$ in GBPRBM and $\{0, 1\}$ in GBLRBM. Parameters of the GBPRBM can be converted to geometrically equivalent GBLRBM parameters using linear transformation given as

$$\begin{aligned} \mathbf{b}_v^{GBL} &= \mathbf{b}_v^{GBP} + \mathbf{W}^{GBP}(-\mathbf{1}) \\ \mathbf{W}^{GBL} &= 2\mathbf{W}^{GBP}, \end{aligned} \tag{2.13}$$

where superscripts denote the model and $(-\mathbf{1})$ is a negated all-ones vector of size $(H \times 1)$. An example of geometrically equivalent GBPRBM and GBLRBM models is shown in Figure 2.2. Although positions of the Gaussian components in the probability of visible units for both models are same, the probability of visible units of these models will be different but can share some similar characteristic.

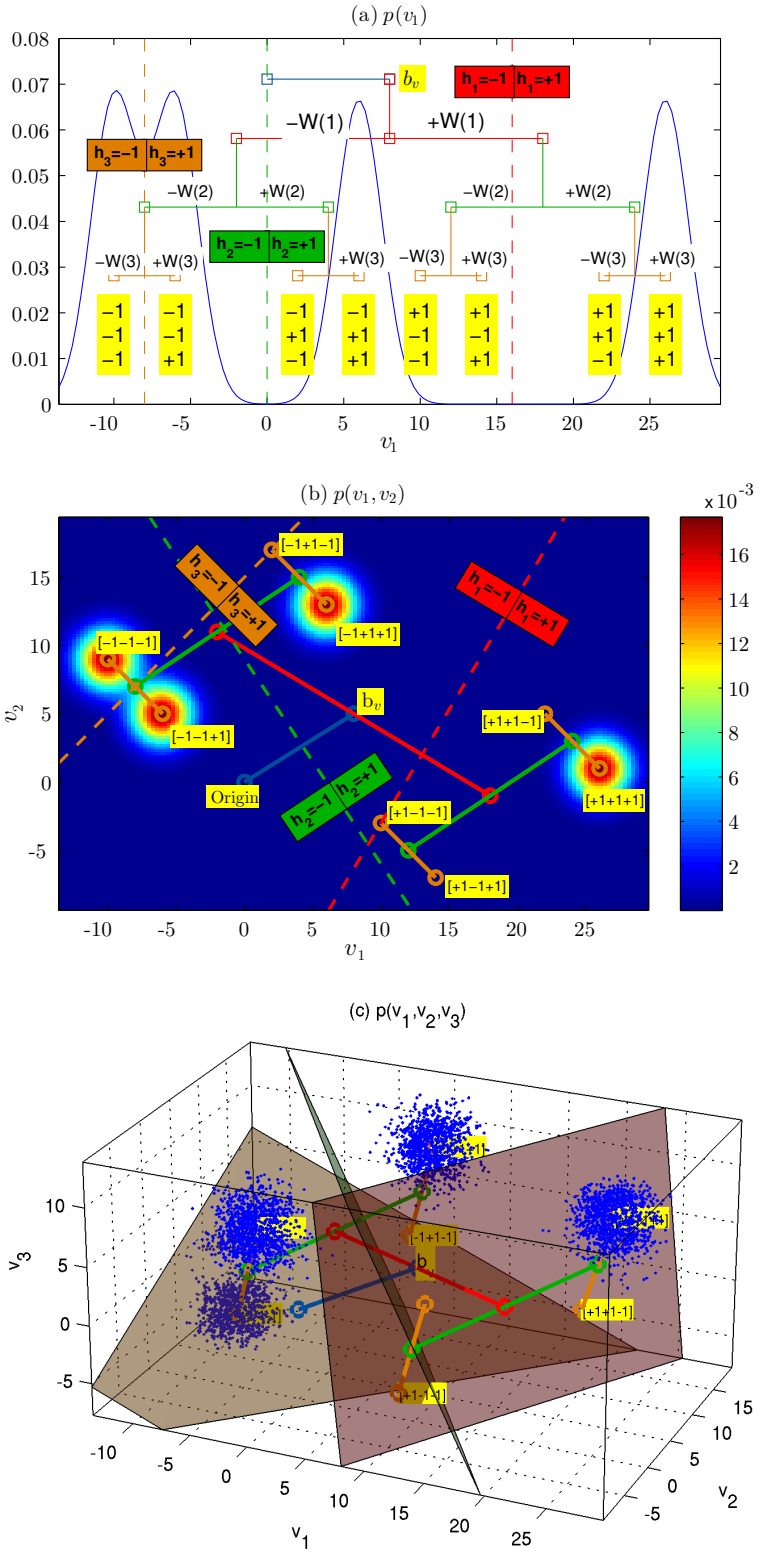

Figure 2.1: Probability of visible units $p(\mathbf{v})$ for a model with (a) $V = 1$, (b) $V = 2$ and (c) $V = 3$. Model geometry and decision boundaries for $p(h_j|\mathbf{v})$ are also shown.

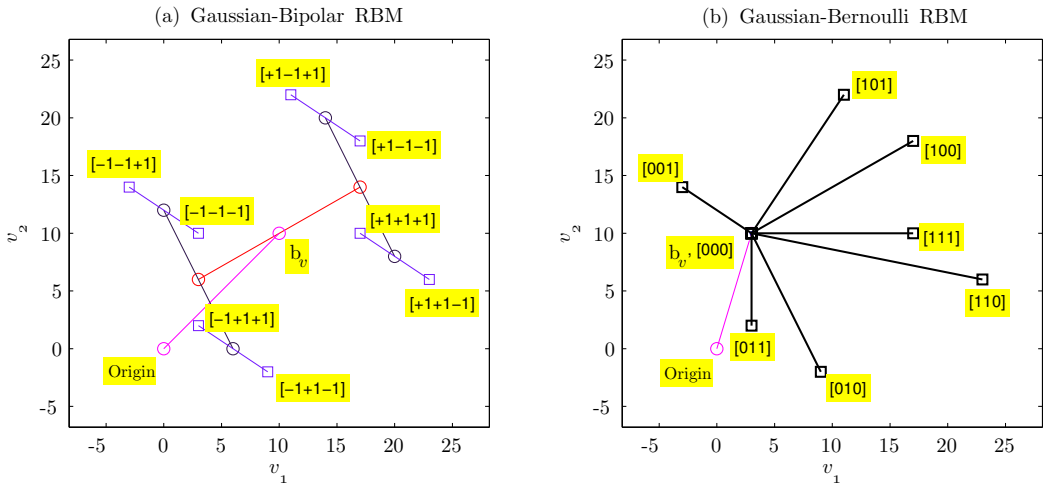

Figure 2.2: Position of the mean vectors of the visible Gaussian components for (a) Gaussian-Bipolar RBM and (b) Gaussian-Bernoulli RBM models.

Chapter 3

HIDDEN ENTROPY AS A MEASURE OF REPRESENTATIONAL EFFICIENCY

As was stated in Section 2.1, probability of hidden units $p(\mathbf{h})$ controls the expression of Gaussian components in the probability of visible units $p(\mathbf{v})$, which is used to model the data. Numerical evaluations show that $p(\mathbf{h})$ is usually far away from being uniform and only few out of 2^H configurations of the hidden vector $\mathbf{h}$ are active, which results in low hidden unit entropy.

Certainly, each cluster in the distribution can be encoded by a single hidden unit, which activates only one the 2^H hidden vector configurations. This is extremely inefficient and is not desired due to excessive complexity of the model. Moreover, training large models is computationally expensive, especially if the number of visible and hidden units is in the order of thousands. Ideally, it would be optimal to model the given data using a small number of hidden units by wisely choosing the model geometry $\{\mathbf{W}, \mathbf{\Sigma}, \mathbf{b}_v\}$ and setting proper hidden bias terms. Therefore, activation of as many hidden vector configurations as possible is desired and makes it feasible to model the data using a smaller number of hidden units. As a consequence, in order to activate as many hidden vector configurations as possible, we need to maximize entropy of the hidden units. Correspondingly, entropy of the hidden units, which will be called "hidden entropy" thereafter, is defined as

$$\mathcal{H}(\mathbf{h}) = -\sum_{\mathbf{h}} p(\mathbf{h}) \log_2 p(\mathbf{h}), \tag{3.1}$$

where probability mass function of hidden units is given as

$$\begin{aligned} p(\mathbf{h}) &= \int_{\mathbf{v}} p(\mathbf{v}, \mathbf{h}) d\mathbf{v} = \frac{\int_{\mathbf{v}} \exp(-E(\mathbf{v}, \mathbf{h})) d\mathbf{v}}{\int_{\mathbf{u}} \sum_{\mathbf{g}} \exp(-E(\mathbf{u}, \mathbf{g})) d\mathbf{u}} = \frac{\exp(A(\mathbf{h}))}{\sum_{\mathbf{g}} \exp(A(\mathbf{g}))} \\ &= \frac{\exp\left(\mathbf{b}_v^T \mathbf{\Sigma}^{-1} \mathbf{W} \mathbf{h} + \frac{1}{2} \mathbf{h}^T \mathbf{W}^T \mathbf{\Sigma}^{-1} \mathbf{W} \mathbf{h} + \mathbf{b}_h^T \mathbf{h}\right)}{\sum_{\mathbf{g}} \exp\left(\mathbf{b}_v^T \mathbf{\Sigma}^{-1} \mathbf{W} \mathbf{g} + \frac{1}{2} \mathbf{g}^T \mathbf{W}^T \mathbf{\Sigma}^{-1} \mathbf{W} \mathbf{g} + \mathbf{b}_h^T \mathbf{g}\right)}, \end{aligned} \tag{3.2}$$

whose derivation is given in A.3.

3.1 Hidden Bias Analysis

Having defined the hidden entropy function in (3.1), it would be interesting to see how the hidden bias term affects probability of hidden units and hence the hidden entropy. In order to analyze the effect of the hidden bias, the first step would be to eliminate the term which depends on the visible bias in (3.2) by setting hidden bias to $\mathbf{b}_h = -\mathbf{W}^T\mathbf{\Sigma}^{-1}\mathbf{b}_v$. In this scenario, the probability mass function of hidden units reduces to

$$\begin{aligned} p(\mathbf{h}) &= \frac{\exp\left(\frac{1}{2}\mathbf{h}^T\mathbf{W}^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{h}\right)}{\sum_{\mathbf{g}}\exp\left(\frac{1}{2}\mathbf{g}^T\mathbf{W}^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{g}\right)} \\ &= \frac{\exp\left(\frac{1}{2}\|\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{W}\mathbf{h}\|_2^2\right)}{\sum_{\mathbf{g}}\exp\left(\frac{1}{2}\|\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{W}\mathbf{g}\|_2^2\right)}. \end{aligned} \tag{3.3}$$

The term $\|\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{W}\mathbf{h}\|_2$ is the Mahalanobis distance between vector $\mathbf{W}\mathbf{h}$ and the origin. Due to the exponential nature of the numerator and denominator functions in $p(\mathbf{h})$, the values with large distances get boosted and values with smaller distances get suppressed yielding a pmf with only most distant components which are active. Usually, depending on the distance $\|\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{W}\mathbf{h}\|_2$ for various hidden vectors $\mathbf{h}$'s, two most distant components with configurations $\mathbf{h}_D$ and $-\mathbf{h}_D$ are activated:

$$\mathbf{h}_D = \arg\max_{\mathbf{h}} \|\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{W}\mathbf{h}\|_2 \tag{3.4}$$

However, depending on $\mathbf{W}$, other components with the same or similar distances can be activated as well.

An example illustrating activation of two most distant components is shown in Figure 3.3(c). The figure portrays probability of visible units $p(\mathbf{v})$ for a model with two visible units where parameters were set as

$$\mathbf{W} = \mathbf{W}_R(1{:}2, 1{:}2), \mathbf{b}_v = \mathbf{b}_v^R(1{:}2), \mathbf{\Sigma} = \mathbf{\Sigma}_R(1{:}2, 1{:}2). \tag{3.5}$$

Here Mahalanobis distance corresponds to Euclidean distance because covariance matrix $\mathbf{\Sigma}$ is a scaled version of an identity matrix, i.e. $\mathbf{\Sigma} = \sigma^2\mathbf{I}$. The hidden bias was set to $\mathbf{b}_h = -\mathbf{W}^T\mathbf{\Sigma}^{-1}\mathbf{b}_v$ so that two most distant components are activated. Note that all decision boundaries for $p(h_j|\mathbf{v})$ pass through point $\mathbf{b}_v$.

From now on, the point $\mathbf{b}_h^* = -\mathbf{W}^T\mathbf{\Sigma}^{-1}\mathbf{b}_v$ in the hidden bias space will be denoted by superscript "*" since it possesses a property of being the center of symmetry of the hidden

bias vs. hidden entropy plot. This can be seen from Figure 3.3(a) and Figure 3.4(a) and will be discussed in Section 3.3.

3.2 Conditions to Attain Maximum Hidden Entropy

Suppose that the number of visible units is greater or equal to the number of hidden units ($V \geq H$) and the hidden bias is set to the center of symmetry point $\mathbf{b}_h = \mathbf{b}_h^*$ described above. Furthermore, let the weight matrix $\mathbf{W}$ have a form of

$$\mathbf{W} = \boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{U} = \boldsymbol{\Sigma}^{\frac{1}{2}}[\mathbf{u}_1 \ \ldots \ \mathbf{u}_H], \tag{3.6}$$

where $\mathbf{U}$ is a matrix with orthogonal columns, such that $\mathbf{u}_j \in \mathbb{R}^V$ for $j \in \{1, \ldots, H\}$, and for all distinct pairs of $j, k \in \{1, \ldots, H\}$, the inner product $\mathbf{u}_j^T\mathbf{u}_k = 0$. Plugging given $\mathbf{W}$ into (3.3) yields

$$\begin{aligned} p(\mathbf{h}) &= \frac{\exp\left(\frac{1}{2}\|\boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{U})\mathbf{h}\|_2^2\right)}{\sum_{\mathbf{g}}\exp\left(\frac{1}{2}\|(\boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{U})\mathbf{g}\|_2^2\right)} \\ &= \frac{\exp\left(\frac{1}{2}\sum_{j=1}^{H}\|\mathbf{u}_j\|_2^2\right)}{\sum_{\mathbf{g}}\exp\left(\frac{1}{2}\sum_{j=1}^{H}\|\mathbf{u}_j\|_2^2\right)} = \frac{1}{2^H}. \end{aligned} \tag{3.7}$$

In this scenario, all of the 2^H hidden configurations become equiprobable and the hidden entropy reaches its maximum value of H bits. To visualize this phenomenon, a GBPRBM model with orthogonal weights $\mathbf{W}(:,1)$ and $\mathbf{W}(:,2)$, which satisfy the condition given in (3.6) with parameters

$$\begin{aligned} \mathbf{W} &= \begin{bmatrix} -9 & 2 \\ 2 & -9 \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} 2^2 & 0 \\ 0 & 2^2 \end{bmatrix}, \mathbf{b}_v = \begin{bmatrix} 11 \\ 12 \end{bmatrix}, \\ \mathbf{b}_h^* &= -\mathbf{W}^T\boldsymbol{\Sigma}^{-1}\mathbf{b}_v = \begin{bmatrix} -30.75, & 21.50 \end{bmatrix}^T \end{aligned} \tag{3.8}$$

is analyzed. Probability of visible units for this model is plotted in Figure 3.1(b). As seen from the figure, four Gaussian components are active. In Figure 3.1(a), empirical evaluation of hidden entropy $\mathcal{H}(\mathbf{h})$ as a function of hidden bias is shown. The hidden bias value listed in (3.8) is a red point located at the intersection of two lines. It has a hidden entropy value of 2 bits and activates four Gaussian components. A more detailed analysis of hidden entropy as a function of hidden bias will be given in the next section.

In summary, maximum hidden entropy of H bits can be obtained if $V \geq H$ and $\mathbf{W}$ has an orthogonal geometry as in (3.6). Such a setup is very restrictive and cannot model data in real world. Moreover, for $V < H$ or any other non-orthogonal geometry, the underlying pmf will be more far away from uniform distribution and hidden entropy will be always less than H. However, maximally useful utilization of hidden units is achievable if a necessary number of Gaussian components, which are needed to model the data, are parametrized by a convenient geometry, in which minimum number of hidden units is used. This is possible only if the hidden entropy is high.

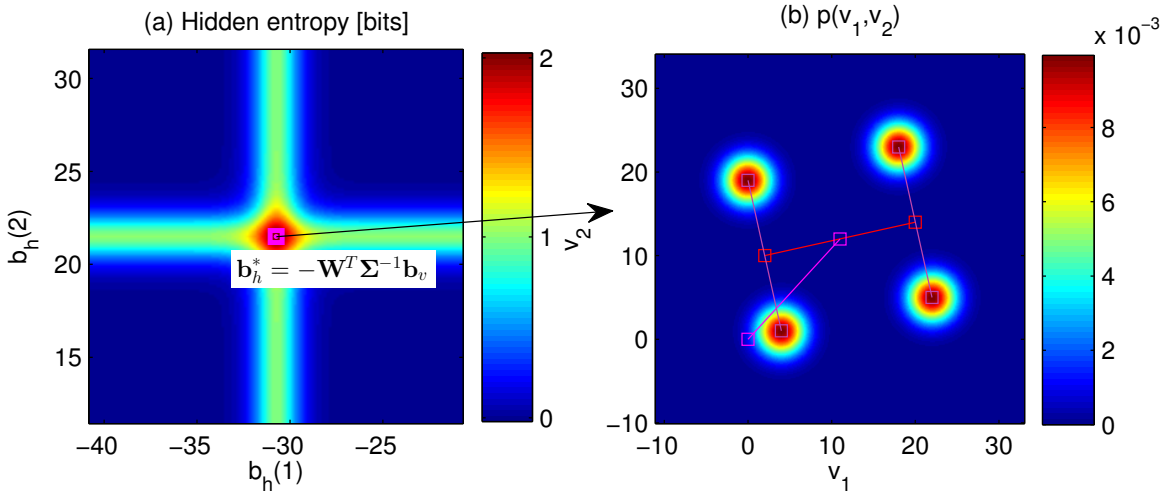

Figure 3.1: (a) Hidden entropy as a function of hidden bias $\mathbf{b}_h$ for a model with parameters $\{\mathbf{W}, \mathbf{\Sigma}, \mathbf{b}_v\}$ listed in (3.8). Intersection of two 1-bit regions (lines) is a 2-bit point $\mathbf{b}_h^* = -\mathbf{W}^T\mathbf{\Sigma}^{-1}\mathbf{b}_v$. (b) Probability of visible units for the same model geometry and a hidden bias set to point $\mathbf{b}_h^*$, shown on the left. In this case, hidden entropy attains its maximum value of $H = 2$ bits and all four (2^H) Gaussian components are activated. All parameters of the model are listed in (3.8).

3.3 Numerical Evaluation and Analytical Description of Hidden Entropy as a Function of Hidden Bias

In order to obtain a complete understanding of how the hidden bias affects the hidden entropy, a visualization of the hidden entropy as a function of the hidden bias is needed.

For this purpose, all other model parameters $\{\mathbf{W}, \mathbf{\Sigma}, \mathbf{b}_v\}$ are fixed and the hidden entropy defined in (3.1) is evaluated as a function of the hidden bias $\mathbf{b}_h$. In all analyses provided here, the target space of $\mathbf{b}_h$ is centered around point $\mathbf{b}_h^*$ and divided into a grid with 150×151 samples.

The analysis of the effect of hidden bias on hidden entropy is conducted for models with 1, 2 and 3 hidden units. In the simplest case, numerical evaluation of the hidden entropy as a function of the hidden bias term for a GBPRBM model with parameters

$$\mathbf{W} = \mathbf{W}_R(:, 1),\ \mathbf{b}_v = \mathbf{b}_v^R,\ \mathbf{\Sigma} = \mathbf{\Sigma}_R, \tag{3.9}$$

is shown in Figure 3.2(a). Here hidden entropy can be found analytically. Setting $c := b_1^h + \mathbf{w}^T \mathbf{\Sigma}^{-1} \mathbf{b}_v$, where b_1^h is hidden bias, yields probability mass function

$$\begin{aligned} p(h) &= \frac{\exp\left(\frac{1}{2} h \mathbf{w}^T \mathbf{\Sigma}^{-1} \mathbf{w} h + ch\right)}{\sum_{\mathbf{g}} \exp\left(\frac{1}{2} g \mathbf{w}^T \mathbf{\Sigma}^{-1} \mathbf{w} g + cg\right)} \\ &= \frac{\exp(ch)}{\sum_{g \in \pm 1} \exp(cg)} = \frac{\exp(ch)}{2\cosh(c)}, \end{aligned} \tag{3.10}$$

which is independent of the model parameters. Plugging calculated values of $p(h)$ into (3.1) results in hidden entropy equal to

$$\begin{aligned} \mathcal{H}(h) &= -\sum_{h \in \pm 1} p(h) \log_2 p(h) \\ &= \frac{c \cdot \tanh(-c) + \ln(2\cosh(c))}{\ln 2}. \end{aligned} \tag{3.11}$$

The maximum value of the hidden entropy ($\mathcal{H}(h) = 1$ bit) is achieved when $c = 0$, which complies with conditions listed in Section 3.2. Analytical evaluation of (3.11) for model parameters listed in (3.9) is shown in Figure 3.2(b) and it matches empirical evaluation of the same function shown in Figure 3.2(a).

For a model with two hidden units ($H = 2$) and parameters $\{\mathbf{W}, \mathbf{\Sigma}, \mathbf{b}_v\}$ listed in (3.5), a plot of the hidden entropy as a function of the hidden bias is shown in Figure 3.3(a). Dark blue regions activate only one configuration of hidden vector and have entropy close to zero. Yellow and crimson regions on the plot activate two and three Gaussian components in $p(\mathbf{v})$, respectively. Points located at these two regions are $\mathbf{b}_h^*$ and $\mathbf{b}_h^S$. Setting hidden bias to these two points yields $p(\mathbf{v})$ as shown in Figure 3.3(c) for $\mathbf{b}_h^*$ and Figure 3.3(d) for $\mathbf{b}_h^S$. As can be seen from Figure 3.3(a), regions of $\mathbf{b}_h$ with high hidden entropy comprise of vertical,

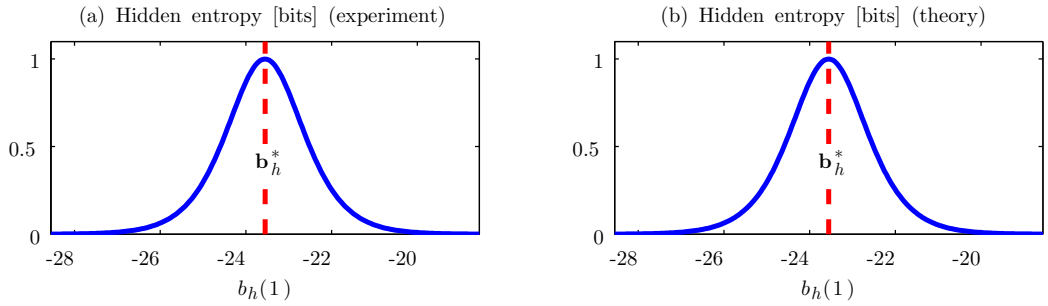

Figure 3.2: (a) Empirical evaluation, and (b) theoretical model of hidden entropy $\mathcal{H}(h)$ as a function of hidden biases b_1^h for a model with parameters listed in (3.9). Empirical evaluation consists in computation of the hidden entropy using (3.1) at every point of the hidden bias space. Theoretical model is based on plotting (3.11) as a function of b_1^h mapped through c.

horizontal and diagonal lines intersecting at some points. Explanation of this phenomenon is given in the next sections. Note that hidden entropy plot has a symmetry with respect to point $\mathbf{b}_h^*$.

The largest visualizable model of hidden entropy as a function of $\mathbf{b}_h$ has three hidden units ($H = 3$). The easiest way to visualize this function is to draw an isosurface at certain hidden entropy level. In Figure 3.4(c), a 0.85-bit isosurface of hidden entropy as a function of hidden bias is shown for a model with parameters $\{\mathbf{W}, \mathbf{\Sigma}, \mathbf{b}_v\}$ listed in (2.8). Similarly to the model with two hidden units, high entropy regions consist of planes with one-bit hidden entropy which intersect at some points. The plot is also symmetrical around the point $\mathbf{b}_h^*$. Further experiments with decreasing the isosurface level show that the rate of change in value of $\mathbf{b}_h$ with respect to the isosurface level is very fast. This means that a vast region of the hidden bias space has hidden entropy almost equal to zero. In the space of $p(\mathbf{v})$ this corresponds to activation of a single Gaussian component only. Such a sparse distribution of high hidden entropy regions emphasizes the importance of hidden bias in representational efficiency of GBPRBMs. For this reason, modeling $\mathbf{b}_h$ analytically in terms of other model parameters $\{\mathbf{W}, \mathbf{\Sigma}, \mathbf{b}_v\}$ is crucial. In the next section, a strategy to model high hidden entropy regions is presented.

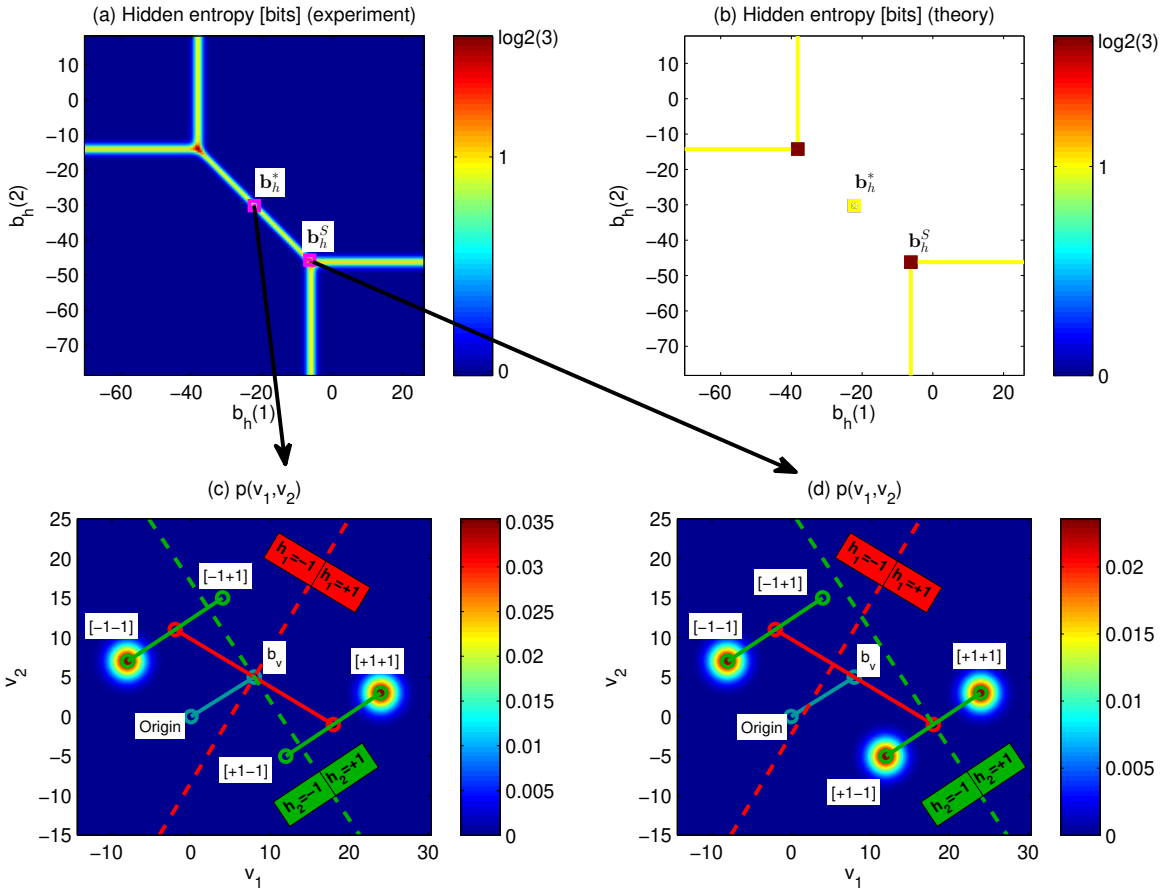

Figure 3.3: (a) Empirical evaluation, and (b) theoretical model of hidden entropy $\mathcal{H}(\mathbf{h})$ as a function of hidden biases b_1^h and b_2^h for a model with parameters listed in (3.5). Empirical evaluation consists in computation of the hidden entropy using its definition in (3.1) at every point of the hidden bias space. Theoretical model is based on plotting one-bit hidden entropy regions using derived inequalities in (3.20) and intersections of these regions. (c) and (d) Probability of visible units $p(\mathbf{v})$ for a model with parameters listed in (3.5) and different hidden biases set to the origins $\mathbf{b}_h^*$ and $\mathbf{b}_h^S$ of the arrows, shown between the plots. $\mathbf{b}_h^*$ has hidden entropy of 1 bit and hence activates 2^1 Gaussian components in (c). $\mathbf{b}_h^S$ has hidden entropy of $\log_2(3)$ bits and activates $2^{\log_2(3)}$ Gaussian components in (d).

3.4 One-Bit Hidden Entropy Regions

In this section, a methodology to model high hidden entropy regions in the space of hidden bias is presented. It is based on deducing one-bit hidden entropy regions from assumption

that two configurations of the hidden vector are equiprobable. The hint can be found in Figure 3.3(a) and Figure 3.4(c). Yellow lines in Figure 3.3(a), which correspond to hidden entropy value of 1 bit, activate two configurations of the hidden units. Hence, they separate dark blue regions where only a single configuration of hidden vector is active. For a model with any number of hidden units, this should be true as well. In this case, we are interested in expressing one-bit hidden entropy region of the hidden bias in terms of the other model parameters. Let us reindex elements of hidden vector and hidden bias:

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_{(1:j-1)} \\ h_j \\ \mathbf{h}_{(j+1:H)} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_P \\ h_j \\ \mathbf{h}_N \end{bmatrix}, \mathbf{b}_h = \begin{bmatrix} \mathbf{b}_{h,(1:j-1)} \\ b_j^h \\ \mathbf{b}_{h,(j+1:H)} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{h,P} \\ b_j^h \\ \mathbf{b}_{h,N} \end{bmatrix}. \tag{3.12}$$

A reasonable assumption is that

$$p(h_j, \mathbf{h}_P, \mathbf{h}_N) = p(-h_j, \mathbf{h}_P, \mathbf{h}_N) \approx 0.5 \tag{3.13}$$

and hence there are two active configurations which yield 1 bit of the hidden entropy. The question is, for which value of b_j^h is this true? What are the constraints on $\mathbf{b}_{h,P}$ and $\mathbf{b}_{h,N}$ imposed by this equation? It should be noted that the assumption above is not the only one possible, because any pair of antipode configurations of hidden vector with some fixed hidden units can be equiprobable, such as

$$\begin{aligned} p(h_j, \mathbf{h}_P, \mathbf{h}_N) &= p(-h_j, -\mathbf{h}_P, \mathbf{h}_N) && \approx 0.5, \\ p(h_j, \mathbf{h}_P, \mathbf{h}_N) &= p(h_j, -\mathbf{h}_P, \mathbf{h}_N) && \approx 0.5, \\ p(h_j, \mathbf{h}_P, \mathbf{h}_N) &= p(h_j, \mathbf{h}_P, -\mathbf{h}_N) && \approx 0.5, \end{aligned} \tag{3.14}$$

and so on for any number of fixed hidden units beginning from 1 till $(H-1)$. In the next two subsections, general solutions to one-bit hidden entropy regions with 1 antipode unit shown in (3.13) and for regions with $(H-1)$ antipode units,

$$p(h_j, \mathbf{h}_P, \mathbf{h}_N) = p(h_j, -\mathbf{h}_P, -\mathbf{h}_N) \approx 0.5, \tag{3.15}$$

are given. Explicit solutions for special cases of the models with two $(H=2)$ and three hidden units $(H=3)$ are provided.

3.4.1 One-Bit Hidden Entropy Region With a Single Antipode Hidden Unit

The assumption here is that two configurations of the hidden vector, $[h_j = +1, \mathbf{h}_P, \mathbf{h}_N]$ and $[h_j = -1, \mathbf{h}_P, \mathbf{h}_N]$, are equiprobable, in which $\mathbf{h}_P$ and $\mathbf{h}_N$ are kept constant and h_j's two antipode values differentiate configurations. Solving (3.13) involves equating energy terms given in (3.2) for both configurations of the hidden vector. The solution for b_j^h is given as

$$b_j^h = -\mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + \mathbf{W} \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix} \right), \tag{3.16}$$

and its detailed derivation can be found in A.4. Plugging the obtained value of b_j^h into (3.13) yields

$$\begin{aligned} p(h_j = +1, \mathbf{h}_P, \mathbf{h}_N) &= p(h_j = -1, \mathbf{h}_P, \mathbf{h}_N) \\ &= \frac{1}{2 + F(\mathbf{h}_P, \mathbf{h}_N)}, \end{aligned} \tag{3.17}$$

where $F(\mathbf{h}_P, \mathbf{h}_N)$ is defined as

$$\begin{aligned} F(\mathbf{h}_P, \mathbf{h}_N) = \sum_{\forall \mathbf{g}_P, \mathbf{g}_N}^{\setminus \mathbf{h}_P, \mathbf{h}_N} \sum_{g_j = \pm 1} \exp \Bigg\{ \begin{bmatrix} \mathbf{g}_P - \mathbf{h}_P \\ 0 \\ \mathbf{g}_N - \mathbf{h}_N \end{bmatrix}^T \times \\ \times \left(\mathbf{b}_h + \mathbf{W}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + g_j \mathbf{W}_{(:,j)} \right) \right) + \\ + \frac{1}{2} \left(\begin{bmatrix} \mathbf{g}_P \\ 0 \\ \mathbf{g}_N \end{bmatrix}^T \mathbf{W}^T \mathbf{\Sigma}^{-1} \mathbf{W} \begin{bmatrix} \mathbf{g}_P \\ 0 \\ \mathbf{g}_N \end{bmatrix} - \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix}^T \mathbf{W}^T \mathbf{\Sigma}^{-1} \mathbf{W} \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix} \right) \Bigg\}. \end{aligned} \tag{3.18}$$

For hidden entropy to be one bit, $F(\mathbf{h}_P, \mathbf{h}_N)$ should be a very small number close to zero. Empirical evaluations shows that this constraint can be mitigated by setting upper bound for $F(\mathbf{h}_P, \mathbf{h}_N)$ as 1:

$$F(\mathbf{h}_P, \mathbf{h}_N) < 1. \tag{3.19}$$

This upper bound corresponds to $\log_2(3)$ bits of hidden entropy value.

Consider a simple model with two hidden units ($H = 2$), which are indexed as $[\mathbf{h}_P, h_j, \mathbf{h}_N]^T \equiv [h_j, h_a]^T$. The solution to (3.13) and inequality (3.19) induced by the same equation is given

as

$$\begin{aligned} b_j^h &= -\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_a \mathbf{W}_{(:,a)}\right), \\ b_a^h &\underset{h_a=-1}{\overset{h_a=+1}{\gtrless}} -\mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)}\right), \quad \text{where} \\ h_j &= -\text{sgn}\left(h_a \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right). \end{aligned} \tag{3.20}$$

When hidden biases b_j^h and b_a^h are set to the values listed above, hidden entropy is approximately equal to 1 bit. In Figure 3.3(b), regions with one-bit hidden entropy plotted using (3.20) are shown as yellow lines. Points where these lines intersect have hidden entropy equal to $\log_2(3)$ bits. If we compare this theoretical model of hidden entropy as a function of hidden bias with the empirical evaluation of the same function in Figure 3.3(a), we can see a perfect match between the modeled function and its empirical evaluation.

Similarly, hidden vector in a model with three hidden units ($H = 3$) can be indexed as $[\mathbf{h}_P, h_j, \mathbf{h}_N]^T \equiv [h_a, h_j, h_b]^T$. For three hidden units the solution is more complex due to combinatorial nature of the problem. In summary, conditions needed to attain hidden entropy of 1 bit are listed below:

$$\begin{aligned} b_j^h &= -\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)}\right), \\ b_a^h &\underset{h_a=-1}{\overset{h_a=+1}{\gtrless}} -\mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j^a \mathbf{W}_{(:,j)} + h_b \mathbf{W}_{(:,b)}\right), \\ &\text{where} \quad h_j^a = -\text{sgn}\left(h_a \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right), \\ b_b^h &\underset{h_b=-1}{\overset{h_b=+1}{\gtrless}} -\mathbf{W}_{(:,b)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j^b \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)}\right), \\ &\text{where} \quad h_j^b = -\text{sgn}\left(h_b \mathbf{W}_{(:,b)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right), \\ h_a b_a^h + h_b b_b^h > &- \left(h_b \mathbf{W}_{(:,b)} + h_a \mathbf{W}_{(:,a)}\right)^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j^{ab} \mathbf{W}_{(:,j)}\right), \\ &\text{where} \quad h_j^{ab} = -\text{sgn}\left(\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left(h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)}\right)\right). \end{aligned} \tag{3.21}$$

In order to check the above hypothesis, hidden entropy analysis given in Figure 3.4(c) was remade by setting the second hidden bias b_2^h to a fixed value

$$\begin{aligned} &b_2^h = -\mathbf{W}_{(:,2)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_1 \mathbf{W}_{(:,1)} + h_3 \mathbf{W}_{(:,3)}\right) = -48, \\ &\text{where} \quad h_1 = h_3 = +1, \end{aligned} \tag{3.22}$$

which is the first condition listed in (3.21) necessary to attain hidden entropy of one bit. In Figure 3.4(a) empirical evaluation of hidden entropy as a function of the remaining hidden biases, b_1^h and b_3^h, is shown. Other conditions necessary to obtain hidden entropy of one bit are actually restrictions on b_1^h and b_3^h. They are visualized in Figure 3.4(b). If we compare this to the empirical evaluation of the same function in Figure 3.4(a), we can see a perfect match between the modeled function and its empirical evaluation.

3.4.2 One-Bit Hidden Entropy Region With $(H-1)$ Antipode Hidden Units

Another region in the space of hidden bias with one-bit hidden entropy can be derived from the assumption that two configurations of the hidden vector with $(H-1)$ antipode hidden units are equiprobable:

$$\begin{aligned} p(h_j, \mathbf{h}_P, \mathbf{h}_N) &= p(h_j, -\mathbf{h}_P, -\mathbf{h}_N) \\ &= \frac{1}{2 + F(h_j, \mathbf{h}_P, \mathbf{h}_N)} \approx 0.5. \end{aligned} \tag{3.23}$$

Solving (3.23) by equating energy terms given in (3.2) for both configurations of the hidden vector yields a constraint for $\mathbf{b}_{h,P}$ and $\mathbf{b}_{h,N}$ as

$$\left(\mathbf{b}_h^T + (\mathbf{b}_v + h_j \mathbf{W}_{(:,j)})^T \mathbf{\Sigma}^{-1} \mathbf{W}\right) \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix} = 0. \tag{3.24}$$

For a model with three hidden units ($H = 3$), hidden vector $\mathbf{h}$ can be indexed as $[\mathbf{h}_P, h_j, \mathbf{h}_N]^T \equiv [h_a, h_j, h_b]^T$and $F(h_j, \mathbf{h}_P, \mathbf{h}_N)$ becomes $F(h_j, h_a, h_b)$, in which residual $(2^H - 2)$ exponential terms are summed up as

$$F(h_j, h_a, h_b) = \sum_{\forall (g_j, g_a, g_b)}^{\setminus (h_j, h_a, h_b), (h_j, -h_a, -h_b)} \exp(B(g_j, g_a, g_b)), \tag{3.25}$$

where $B(g_j, g_a, g_b)$ is defined as:

$$\begin{aligned} B(g_j, g_a, g_b) = &\left(g_b - \frac{g_a h_b}{h_a}\right)\left(b_b^h + \mathbf{b}_v^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,b)}\right) + \\ &+ (g_j - h_j) b_j^h + (g_a g_b - h_a h_b) \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,b)} + \\ &+ \mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1} (g_j - h_j)\left(\mathbf{b}_v + g_a \mathbf{W}_{(:,a)}\right) + \\ &+ \mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1} \left(g_j g_b - \frac{h_j g_a h_b}{h_a}\right) \mathbf{W}_{(:,b)}. \end{aligned} \tag{3.26}$$

For the hidden entropy to be one bit, $F(h_j, h_a, h_b)$ should be a very small number close to zero. Similarly to the derivation of the single antipode hidden unit case, this constraint can be mitigated by setting upper bound for $F(h_j, h_a, h_b)$ as 1:

$$F(h_j, h_a, h_b) < 1. \tag{3.27}$$

This upper bound corresponds to $\log_2(3)$ bits of hidden entropy value.

In summary, conditions needed to attain one-bit hidden entropy are listed below:

$$\begin{aligned}
&b_b^h > -\mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + h_b h_s \mathbf{W}_{(:,a)} + h_j \mathbf{W}_{(:,j)}\right), \\
&b_b^h < -\mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v - h_b h_s \mathbf{W}_{(:,a)} + h_j \mathbf{W}_{(:,j)}\right), \\
&\text{where} \quad h_s = \text{sgn}(\mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,b)}), \\
&b_a^h = -\frac{h_b}{h_a} b_b^h - \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)}\right)^T \mathbf{\Sigma}^{-1} \left(\frac{h_b}{h_a} \mathbf{W}_{(:,b)} + \mathbf{W}_{(:,a)}\right), \\
&b_j^h \underset{h_j=-1}{\overset{h_j=+1}{\gtrless}} -\mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v - h_j h_s^{ab} (h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)})\right), \\
&\text{where} \quad h_s^{ab} = \text{sgn}(\mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1} (h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)})), \\
&h_b b_b^h - h_j b_j^h < + \left(h_j \mathbf{W}_{(:,j)} - h_b \mathbf{W}_{(:,b)}\right)^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v - h_a \mathbf{W}_{(:,a)}\right), \\
&h_b b_b^h + h_j b_j^h > - \left(h_j \mathbf{W}_{(:,j)} + h_b \mathbf{W}_{(:,b)}\right)^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + h_a \mathbf{W}_{(:,a)}\right).
\end{aligned} \tag{3.28}$$

In Figure 3.4(d), one of the one-bit hidden entropy regions with $(H-1)$ antipode hidden units is plotted in magenta using (3.28) and marked with "$p(h_j, h_a, h_b) = p(h_j, -h_a, -h_b)$" label.

3.5 Intersections of One-Bit Hidden Entropy Regions

Depending on the dimension of the hidden entropy space (H), different types of one-bit hidden entropy regions intersect/connect with each other at their boundaries, forming $\log_2(3)$, 2, $\log_2(5)$-bit and higher hidden entropy regions. These regions can be lines, planes or hyperplanes and are of primary interest because they provide highest representational efficiency of the model.

More detailed look at Figures 3.3(b) and 3.4(d) reveals that regions of hidden bias with high hidden entropy are concentrated in a grid, formed by intersection of one-bit hidden entropy regions with a single antipode hidden unit, i.e. having $p(\mathbf{h})$ with the following

property:

$$p(h_j, \mathbf{h}_P, \mathbf{h}_N) = p(-h_j, \mathbf{h}_P, \mathbf{h}_N) \approx 0.5. \tag{3.29}$$

The solution to these hyperplanes is given in (3.16). The vector $[\mathbf{h_P}; \mathbf{h}_N]$ has $(H-1)$ hidden units, so there are $2^{(H-1)}$ different combinations for hidden unit h_j, which corresponds to $2^{(H-1)}$ hyperplanes for each hidden bias dimension j. Hence, the number of intersections of these hyperplanes is $(2^{(H-1)})^H$.

However, not all of these intersections have high hidden entropy. For models with 2 hidden units, only two points out of 4 (i.e. $2^{(2-1)2}$) have high hidden entropy. One of such points was used with the model given in (3.5) to activate three Gaussian components in $p(\mathbf{v})$ shown in Figure 3.3(d). From Figure 3.3(a) it can be seen that $\mathbf{b}_h^S$ is an intersection of two lines. Point $\mathbf{b}_h^S$ was computed using boundary values from (3.21) adapted for a model with $V = 2$ and $H = 2$. Explicitly, $\mathbf{b}_h^S$ was calculated as follows:

$$\begin{aligned} b_1^{h,S} &= -\mathbf{W}_{(:,1)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + \mathbf{W}\mathbf{H}_{(:,1)}^S\right) = \ -6.2222, \\ b_2^{h,S} &= -\mathbf{W}_{(:,2)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + \mathbf{W}\mathbf{H}_{(:,2)}^S\right) = -46.2222, \end{aligned} \qquad \mathbf{H}^S = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} \tag{3.30}$$

where $\{\mathbf{W}, \mathbf{\Sigma}, \mathbf{b}_v\}$ are taken from (3.5) and $\mathbf{H}^S$ is configuration matrix which represents vectors $\mathbf{h}_P$ and $\mathbf{h}_N$ in a packed form.

In a model with 3 hidden units, only six points out of 64 (i.e. $2^{3(3-1)}$) have high hidden entropy. To demonstrate this, all possible one-bit hidden entropy regions and their intersections with non-zero hidden entropy are shown in Figure 3.4(d). These regions and their boundary values were calculated using (3.21) and (3.28). If we compare them to the empirical evaluation of the same function in Figure 3.4(c), we can see a perfect match between the modeled function and its empirical evaluation. Additionally, a sample point $\mathbf{b}_h^C$ with high hidden entropy was calculated for the model geometry $\{\mathbf{W}, \mathbf{\Sigma}, \mathbf{b}_v\}$ given in (2.8) using the following equation:

$$\begin{aligned} b_1^{h,C} &= -\mathbf{W}_{(:,1)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + \mathbf{W}\mathbf{H}_{(:,1)}^C\right) = -52.4444, \\ b_2^{h,C} &= -\mathbf{W}_{(:,2)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + \mathbf{W}\mathbf{H}_{(:,2)}^C\right) = -16.0000, \\ b_3^{h,C} &= -\mathbf{W}_{(:,3)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + \mathbf{W}\mathbf{H}_{(:,3)}^C\right) = +13.3333, \end{aligned} \qquad \mathbf{H}^C = \begin{bmatrix} 0 & +1 & +1 \\ -1 & 0 & +1 \\ -1 & -1 & 0 \end{bmatrix}. \tag{3.31}$$

where configuration matrix $\mathbf{H}^C$ is of size 3×3 since the model has three hidden units. From Figure 3.4(d), it can be noted that point $\mathbf{b}_h^C$ has hidden entropy value of 2 bits. This

corresponds to activation of four configurations of hidden units which, in turn, switch four Gaussian components on in $p(\mathbf{v})$ in Figure 2.1(b).

In Figure 3.4(d), all intersections of one-bit hidden entropy regions, which have high hidden entropy values, lie on a diagonal plane which passes through the center $\mathbf{b}_h^*$ and form a hexagon-like structure. Observing such a complex and interesting shape even in three dimensions makes it harder to predict, how many of the $2^{(H-1)H}$ hyperplane intersections will have high entropy and what shape will they take in higher dimensions, but their number tends to be small.

3.6 One-Bit Hidden Entropy Regions as Decision Boundaries for Hidden Units Activations

As was shown in Figure 2.1 and (2.12), hidden bias acts like a parameter which specifies the position of the hyperplane in the space of visible units. This hyperplane is a decision boundary which separates antipode configurations of the hidden unit.

Geometrical interpretation of $p(h_j = +1|\mathbf{v})$ with hidden bias set to one-bit hidden entropy region with a single antipode hidden unit is shown in Figure 3.5. In this case, conditional probability $p(h_j = +1|\mathbf{v})$ with b_j^h set to (3.16) is given as

$$\begin{aligned} p(h_j = +1|\mathbf{v}) &= \text{sigm}\left(2\left(\mathbf{v}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,j)} + b_j^h\right)\right) \\ &= \text{sigm}\left(2\mathbf{W}_{(:,j)}^T\boldsymbol{\Sigma}^{-1}\left((\mathbf{v} - \mathbf{b}_v) - \mathbf{W}\begin{bmatrix}\mathbf{h}_{(1:j-1)}\\ 0 \\ \mathbf{h}_{(j+1:H)}\end{bmatrix}\right)\right). \end{aligned} \tag{3.32}$$

From Figure 3.5, it can be seen that decision boundary passes perpendicularly to the jth weight right in the middle between active hidden configurations.

Similarly, for a model with three hidden units, decision boundaries for $p(h_1 = +1|\mathbf{v})$ and $p(h_3 = +1|\mathbf{v})$ can be plotted by setting hidden bias to one-bit hidden entropy region with two, i.e. $(H-1)$ antipode hidden units are given as

$$\begin{aligned} b_1^h &= -\mathbf{W}_{(:,1)}\boldsymbol{\Sigma}^{-1}(\mathbf{b}_v + h_2\mathbf{W}_{(:,2)}), \\ b_3^h &= -\mathbf{W}_{(:,3)}\boldsymbol{\Sigma}^{-1}(\mathbf{b}_v + h_2\mathbf{W}_{(:,2)}). \end{aligned} \tag{3.33}$$

Note that these two values are not a unique solution to (3.24). Plugging these values into

(2.12) yields:

$$p(h_1{=}{+}1|\mathbf{v}){=}\text{sigm}\Big(2\mathbf{W}_{(:,1)}^T\boldsymbol{\Sigma}^{-1}(\mathbf{v}{-}\big(\mathbf{b}_v{+}h_2\mathbf{W}_{(:,2)}\big)\Big),$$
$$p(h_3{=}{+}1|\mathbf{v}){=}\text{sigm}\Big(2\mathbf{W}_{(:,3)}^T\boldsymbol{\Sigma}^{-1}(\mathbf{v}{-}\big(\mathbf{b}_v{+}h_2\mathbf{W}_{(:,2)}\big)\Big). \tag{3.34}$$

From Figure 3.6, it can be seen that decision boundaries pass perpendicularly to the first and third weights right in the middle between active hidden configurations.

Such geometrical interpretation is crucial for understanding principle of operation of GBPRBM models and gives an insight into the data modeling from the clustering perspective.

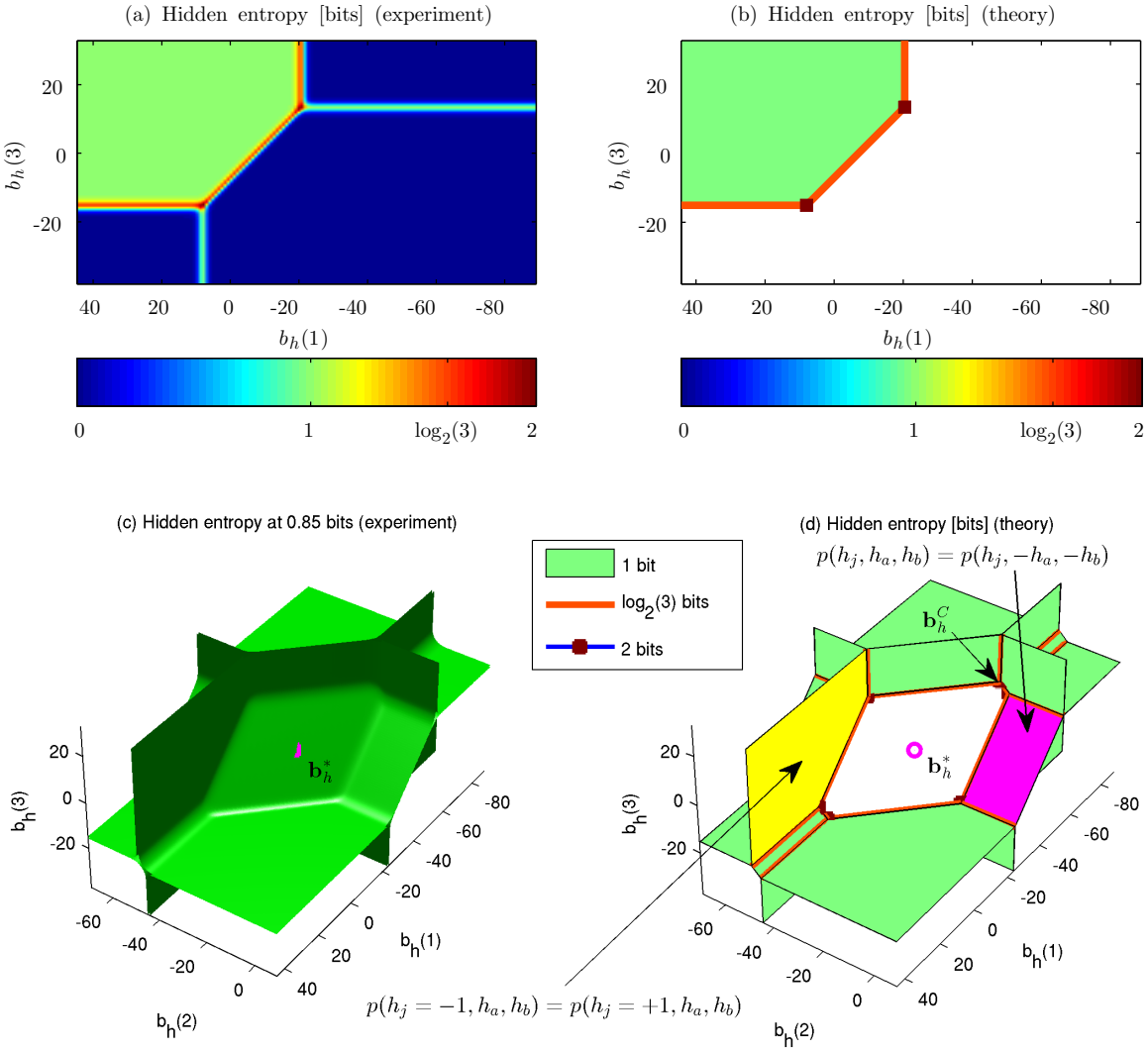

Figure 3.4: (a) Empirical evaluation, and (b) theoretical model of hidden entropy $\mathcal{H}(\mathbf{h})$ as a function of hidden biases b_1^h and b_3^h for a model with parameters listed in (2.8) with second bias b_2^h is set to -48. (c) Empirical evaluation, and (d) theoretical model of hidden entropy $\mathcal{H}(\mathbf{h})$ as a function of all hidden biases b_1^h, b_2^h, b_3^h. In (a) and (c), the hidden entropy was calculated using its definition in (3.1) at every point of the hidden bias space spanned by b_1^h, b_3^h in (a) and b_1^h, b_2^h, b_3^h in (c). Theoretical model is based on plotting one-bit hidden entropy regions, $p(h_j, h_a, h_b) = p(-h_j, h_a, h_b)$ (yellow) and $p(h_j, h_a, h_b) = p(h_j, -h_a, -h_b)$ (magenta), using derived inequalities in (3.21) and (3.28), respectively. Intersections of such regions produce hidden entropy equal to $\log_2(3)$ and 2 bits. One such point $\mathbf{b}_h^C$ is shown in (d). It was calculated using (3.31). Plots (a) and (b) correspond to a slice taken from (c) and (d) by setting $b_2^h = -48$.

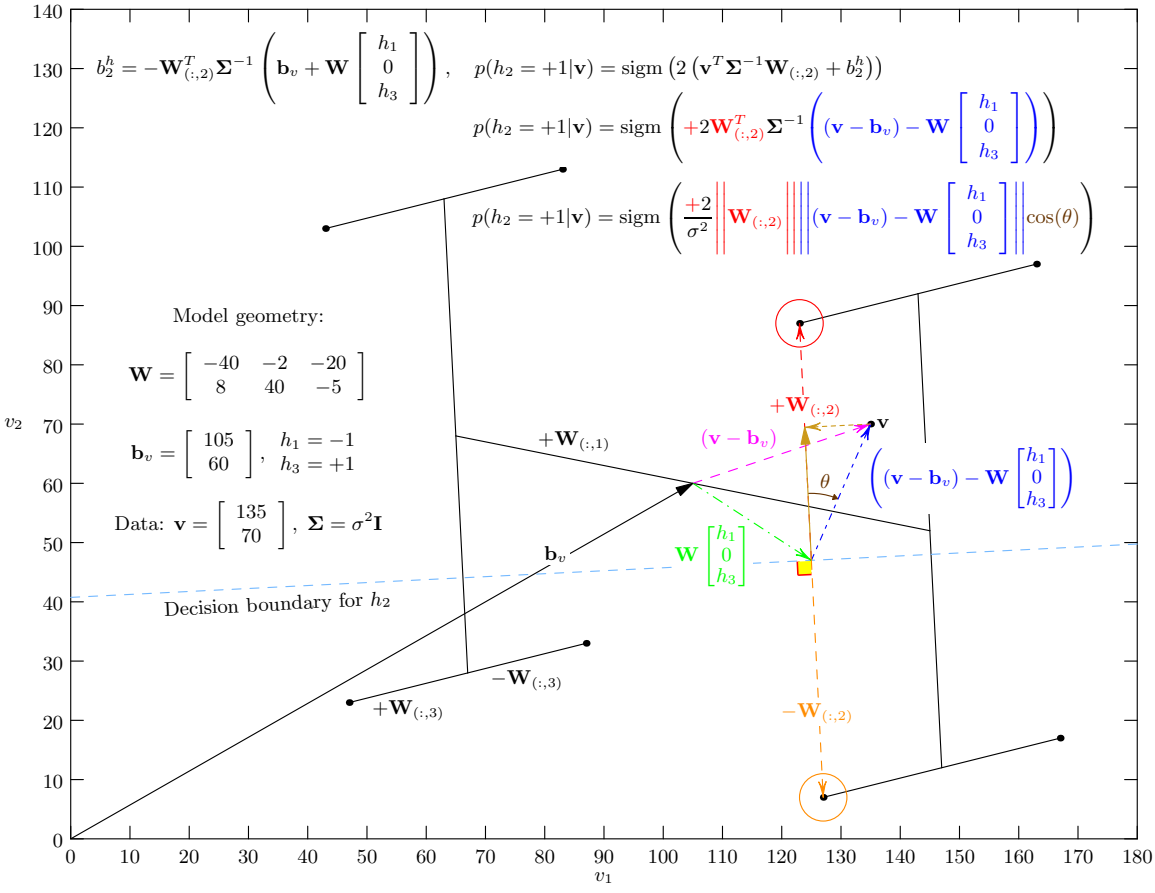

Figure 3.5: Decision boundaries for $p(h_j|\mathbf{v})$ with b_j^h set to one-bit hidden entropy region with a single antipode hidden unit, i.e. b_j^h satisfies equality $p(h_j, h_a, h_b) = p(-h_j, h_a, h_b)$, where indices $j = 2, a = 1$ and $b = 3$.

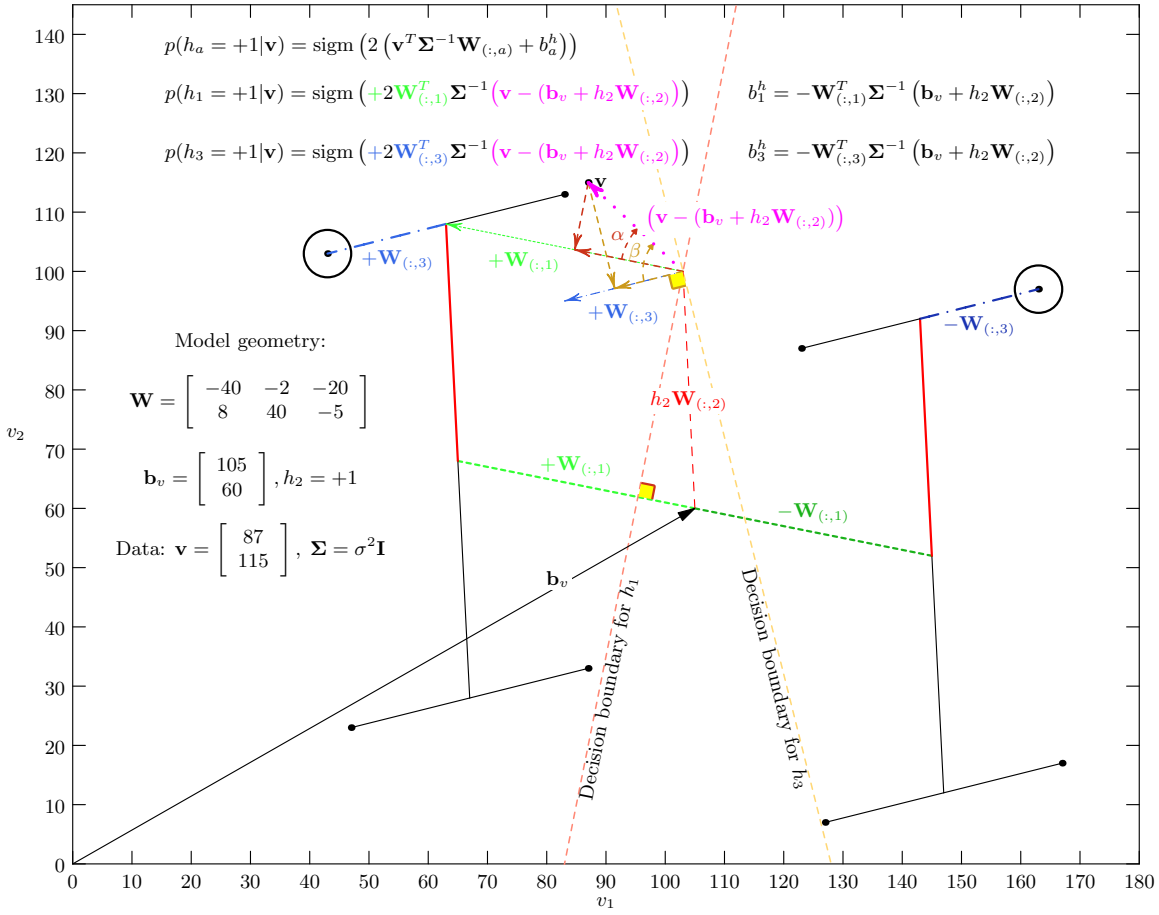

Figure 3.6: Decision boundaries for $p(h_1|\mathbf{v})$ and $p(h_3|\mathbf{v})$ with b_1^h and b_3^h set to one-bit hidden entropy region with two antipode hidden units, i.e. b_j^h satisfies equality $p(h_j, h_a, h_b) = p(h_j, -h_a, -h_b)$ where indices $j = 2, a = 1$ and $b = 3$.

Chapter 4

EMPIRICAL ANALYSIS OF REPRESENTATIONAL EFFICIENCY

In Section 3, we presented challenges of analysis of the hidden entropy as a function of hidden bias. Computing hidden entropy defined in (3.1) implies calculating $p(\mathbf{h})$ and expectations over all 2^H elements in the range of the random vector $\mathbf{h}$. This is not feasible for models with large number of hidden units.

4.1 Normalized Empirical Hidden Entropy

As a workaround, we propose a new measure of usefulness of hidden units which can be estimated from the statistics of their activations. It relies on an assumption that hidden units are independent. If a hidden unit is always "on" for all data samples, it means that it shifts $\mathbf{b}_v$ by $+\mathbf{W}_{(:,j)}$ in all samples. So why not just replace $\mathbf{b}_v$ with $[\mathbf{b}_v + \mathbf{W}_{(:,j)}]$ and get rid of this useless hidden unit? It will not affect other hidden units, because their activations are based on conditional pdf $p(\mathbf{h}|\mathbf{v})$, in which individual hidden units are independent as shown in (**??**). Hence, to compute the individual hidden unit activations, we first sample hidden vector $\widehat{\mathbf{h}}$ from $p(\mathbf{h}|\mathbf{v})$ for each visible vector $\mathbf{v}$ in the data set S. Then we obtain matrix $\widehat{\mathbf{A}}$ with hidden vector activations by concatenating all vectors $\widehat{\mathbf{h}}^s, s \in \{1, \ldots, |S|\}$ side by side where $|S|$ is the number of samples in the data set:

$$\widehat{\mathbf{A}} = [\widehat{\mathbf{h}}^1, \ldots, \widehat{\mathbf{h}}^{|S|}]. \tag{4.1}$$

Provided that, an estimate of the probability of hidden unit's value h_j is calculated as a frequency of occurrence of h_j in the jth row of matrix $\widehat{\mathbf{A}}$:

$$\begin{aligned} p(\widehat{h}_j = +1) &= \frac{1}{|S|} \sum_{s=1}^{|S|} I_{\widehat{\mathbf{A}}(j,s)=+1}, \\ p(\widehat{h}_j = -1) &= 1 - p(\widehat{h}_j = +1) \quad \text{for} \quad j \in \{1, \ldots, H\}, \end{aligned} \tag{4.2}$$

where $I_{\{cond.\}}$ is an indicator function which returns "1" if condition is satisfied and "0" otherwise. Given pmf estimates of H individual hidden units, a measure of hidden units'

activations can be devised based on entropy. Consequently, we define Normalized Empirical Hidden Entropy (NEHE) as an average of estimated hidden unit entropies,

$$\widehat{\mathbf{H}} = \frac{1}{H}\sum_{j=1}^{H}\mathcal{H}(\widehat{h}_j) = -\frac{1}{H}\sum_{j=1}^{H}\sum_{\widehat{h}_j\in\pm 1} p(\widehat{h}_j)\log_2 p(\widehat{h}_j). \tag{4.3}$$

NEHE can take a maximum value of 1 bit, which will indicate uniform distribution of all hidden units. It should also be noted that NEHE and hidden entropy are two different measures, which share the idea of measuring usefulness of hidden units. If the hidden units are independent, then NEHE approximates the true hidden entropy,

$$\mathcal{H}(\mathbf{h}) = H\cdot\widehat{\mathbf{H}} \quad \text{if} \quad p(\mathbf{h}) = \prod_{i=j}^{H} p(h_j). \tag{4.4}$$

On the other hand, if hidden units are correlated, the NEHE can be taken as an upper bound for the true hidden entropy. We further discuss independence assumption and correlatedness of hidden units in the next section.

4.2 Experiments with Normalized Empirical Hidden Entropy

In order to analyze how number of hidden units affects representational efficiency, we conducted experiments using the MNIST, CIFAR-10 and Faces data sets. We trained GBPRBM models with different numbers of hidden units and observed how Root-Mean Square Error (RMSE) and normalized empirical hidden entropy (NEHE) change in each case.

Particularly, we used vanilla Contrastive Divergence (CD) algorithm described in [17] to train GBPRBM models, in which model parameters are estimated by maximizing log-likelihood function

$$\mathcal{L}(\boldsymbol{\theta}|\mathbf{v}) = \ln p(\mathbf{v};\boldsymbol{\theta}). \tag{4.5}$$

Given a training data set S with realizations of visible vector $\mathbf{v}$, the optimal estimates of model parameters $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h\}$ are found as

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \frac{1}{|S|}\sum_{s\in S}\mathcal{L}(\boldsymbol{\theta}|\mathbf{v}_s), \tag{4.6}$$

where $|S|$ is the number of samples in data set S. It is not feasible to find global optimum for (4.6) directly, instead a modified gradient ascent algorithm is used for this purpose. An

update rule for every parameter θ of the parameter set $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h, \}$ is given as

$$\theta^{(t+1)} = \theta^{(t)} + \Delta\theta^{(t)} \tag{4.7}$$

where index t represents iteration number, and the gradient is computed according the following equation:

$$\Delta\theta^{(t)} = (1-\mu)\nu \left. \frac{1}{|S|} \sum_{s \in S} \frac{\partial}{\partial\theta} \mathcal{L}(\theta|\mathbf{v}_s) \right|_{\theta^{(t)}} + \mu\Delta\theta^{(t-1)}. \tag{4.8}$$

In addition to the derivative term $\mathcal{L}'(\theta|\mathbf{v}_s)$, the gradient $\Delta\theta^{(t)}$ contains value $\Delta\theta^{(t-1)}$ from the previous iteration, scaled by momentum factor $\mu \in (0, 1)$ which prevents possible oscillation and stabilizes convergence.

We used the MNIST data set, which contains 60000 gray-scale images of size 28×28 pixels, stored in unsigned 8-bit integer format. All images in the data set were scaled into the range of $[0, 1]$. A predefined setup was used, in which the data are splitted into a training and a test sets with 50000 and 10000 images respectively. Afterwards, the conventional contrastive divergence algorithm was run with the following parameters: fixed variances of the visible units $\sigma_v^2 = 6.7 \cdot 10^{-3}$, CD order $k = 1$, learning rate $\nu = 10^{-5}$ and momentum $\mu = 0.5$. Model weights and hidden biases are initialized randomly with the maximum magnitude of 0.01. The visible bias is set to the mean of the data, which makes it a good guess considering symmetrical geometry of the GBPRBM model. In the beginning of the first epoch the whole data set is partitioned into $|S|/|S_m|$ disjoint mini-batches of size $|S_m| = 20$, and the learning algorithm starts. At every iteration, an average of the gradient is taken over a mini-batch and the update rule is applied according to (4.7). After every epoch, the samples in the data set are shuffled and repartitioned into mini-batches of the same size and learning continues. In total three epochs were used, which is quite enough for the convergence of the training algorithm. An example model with 1500 hidden units has a per-pixel Root-Mean Square Error (RMSE) performance of 0.0832 on the test data set. In Figure 4.1, original and reconstructed test images are shown. Images were reconstructed by sampling hidden units and using them to sample visible units (Gibbs sampling).

We also conducted experiments on the CIFAR-10 data set, which consists of 60000 color images of size 32×32 pixels pertaining to 10 classes (airplane, automobile, ship, truck, bird, cat, deer, dog, frog, and horse). The images were converted into a gray scale and the same

procedure of data normalization and splitting was done as in the MNIST case. Similarly, GBPRBM models with different numbers of hidden units were trained. The mini-batch size $|S_m|$ was set to 50 and the learning rate was decreased to $5 \cdot 10^{-6}$. Number of epochs was augmented to 5. An example model with 1024 hidden units has a per-pixel Root-Mean Square Error (RMSE) performance of 0.1053 on the test data set. In Figure 4.3, original and reconstructed test images are shown. CIFAR-10 is a very complex data with huge variability in every pixel. For this reason, reconstructed images have worse quality than those in MNIST. In Figure 4.4, some of the learned weights (reshaped columns of $\mathbf{W}$) are shown. There is no particular structure in the filters learned.

The third data set used for the experiment is the human faces data set[1]. It contains 3993 gray-scale images of faces belonging to different people of diverse ethnicities and both genders. Original images of size 128×128 pixels were downsampled to a size of 52×52 pixels and normalized to a $[0, 1]$ range. Similarly, GBPRBM models with different numbers of hidden units were trained. The mini-batch size $|S_m|$ was set to 100 and the constant variances of the visible units were set to $9.1188 \cdot 10^{-4}$. The learning rate for models with tens and hudereds hidden units was set to $2 \cdot 10^{-6}$ and decreased to a value of $5 \cdot 10^{-7}$ for models with thousands of hidden units. A slower convergence was compensated by increasing number of epochs to 20. Prior information about the data was used to wisely initialize weight matrix $\mathbf{W}$. Since the face is almost always centered and the background is black, corners of the image do not carry any useful information. Weight matrix $\mathbf{W}$ was initialized as an average face scaled by a random number in the range $[-5{\cdot}10^{-4}, +5{\cdot}10^{-4}]$. An example model with 4056 hidden units has a per-pixel Root-Mean Square Error (RMSE) performance of 0.0595 on the test data set. In Figure 4.5, original and reconstructed test images are shown. Faces data set is a structured data set with less variability in every pixel. For this reason, reconstructed images have better quality than those in MNIST. In Figure 4.6, some of the learned weights are shown. It looks like every hidden unit encodes a distinct human face, because the number of hidden units is close to the number of samples in the data set.

Since we investigate the effect of increasing the number of hidden units on the model performance and representational efficiency, we trained several models with H ranging from

[1]The data set was downloaded from:
http://courses.media.mit.edu/2002fall/mas622j/proj/faces/rawdata.zip

order of tens to thousands. Per-pixel RMSE and NEHE were measured for the trained models using the test data set. In Figure 4.7, these quantities are plotted as a function of the ratio of hidden units to visible units (H/V). NEHE tends to attain its maximum value of $[0.88, 0.96]$ when H/V is in the range of $[0.25, 1]$. After this point, NEHE starts to decline. This may indicate redundant complexity of the model. Meanwhile, normalized RMSE decreases as H/V increases and after some point it reaches its constant value.

From Section 3.2, we know that in order to attain maximum hidden entropy, before diving into the orthogonality condition of the weights, the first most important requirement $V \geq H$ should be satisfied. As a consequence, $H/V = 1$ is the point after which increasing number of hidden units will only decrease hidden entropy. Note that NEHE is an approximation of the hidden entropy only if hidden activations are independent, otherwise it defines an upper bound for the hidden entropy. In Figure 4.7, we observe that NEHE, indeed, decreases as H/V exceeds value of 1. On the other hand, augmenting the number of hidden units does not help and RMSE slowly decreases, attaining its constant value after $H = V$.

For a small number of hidden units, the model will try to make use of as many hidden vector configurations as possible. Not every hidden unit can be activated because norm (length) of the vector $\mathbf{W}_{(:,j)}$ is large. In the domain of images, this phenomenon corresponds to thick large strokes, whose superimposition reconstructs original image. In Figure 4.2(a), weights $\mathbf{W}_{(:,j)}$s learned by a model with 512 hidden units are shown. The model was trained with the MNIST data set and accordingly, the filters learned resemble thick strokes and half-digits. Since shapes of the digits are different, not all hidden units will be superimposed to reconstruct original image. This may explain the under-utilization of hidden units and a higher RMSE when $H/V < [0.25, 1]$. After increasing the number of hidden units there will be an "optimal" point $H/V < [0.25, 1]$, where NEHE attains its maximal value. In Figure 4.2(b), filters learned at this point are shown. They no longer bear resemblance to strokes, rather they have more cloud-like structure. Surprisingly, combination of these filters yield shapes of digits with quite sharp edges. Further augmentation of the number of hidden units results in generation of excessive filters with noise, as shown in Figure 4.2(c). These noisy filters are not used in the image reconstruction and decrease the value of NEHE.

Independence assumption of hidden units in NEHE is a very strong assumption. Some of the redundant hidden units may be correlated with other hidden units. They have to be

somehow incorporated into the model. This results in smaller lengths (norms) of vectors $\mathbf{W}(:, j)$s. In case of 100% correlation, the values of hidden units will be the same, i.e. $h_j = h_k$, such that the distance "travelled" in the space of visible units is $h_j\mathbf{W}(:, j) + h_k\mathbf{W}(:, k) = h_j[\mathbf{W}(:, j)+\mathbf{W}(:, k)]$. Moreover, two completely correlated vectors $\mathbf{W}(:, j)$ and $\mathbf{W}(:, k)$ should point in the same direction, because running Gibbs sampling using $p(h_j|\mathbf{v})$ and $p(h_k|\mathbf{v})$ will not generate the same value of hidden units otherwise. Intuitively, it would be logical to remove superfluous hidden unit h_k and replace $\mathbf{W}(:, j)$ with $[\mathbf{W}(:, j) + \mathbf{W}(:, k)]$, if both models perform the same. Thus, complexity of the model can be decreased. Two correlated hidden units will have the same contribution to NEHE, but it is hard to predict the ratio of the correlated hidden units as the number of hidden units gets augmented. Perhaps, correlation of hidden units should also be incorporated into a NEHE-like measure, as well as the norms of the weights $\mathbf{W}(:, j)$s, since geometry of the model also has a huge effect on usefulness of the hidden units. It is a subject for future work and more profound analysis.

In summary, the ratio of the number of hidden units to the number of visible units H/V may need be in the range of $[0.25, 1]$ or greater to efficiently represent the data. Increasing this ratio induces sparsity in hidden units and NEHE starts to decline while RMSE decreases very slowly and reaches its constant value. It is a sign that augmenting the number of hidden units does not enhance the model's performance. This behavior depends on the nature of the data and may differ from data set to data set.

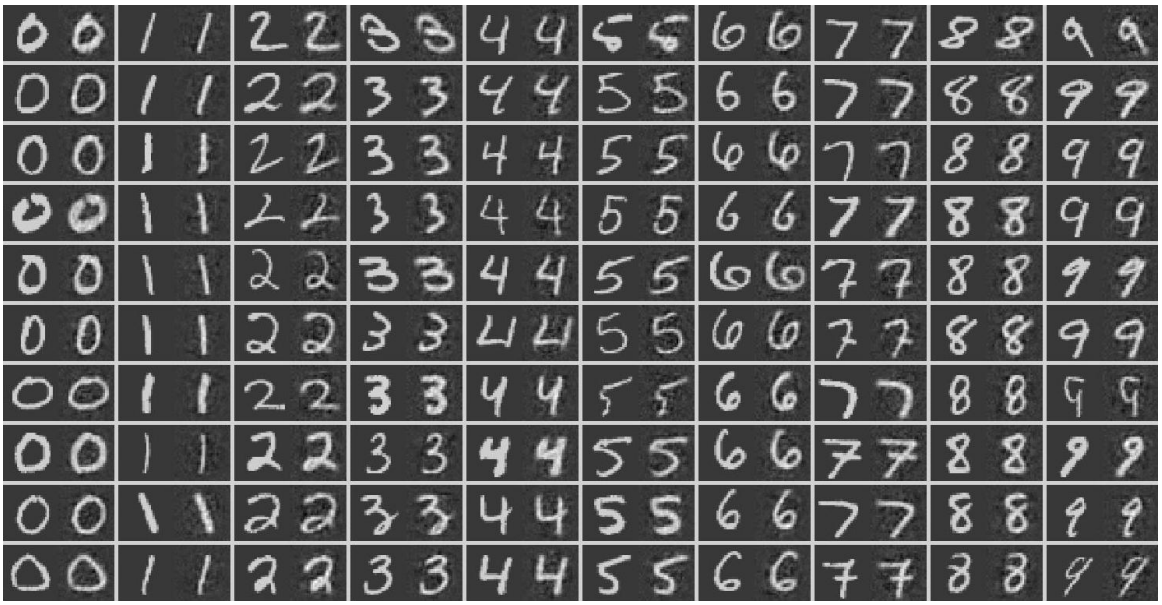

Figure 4.1: Original (left) and reconstructed (right) images from the MNIST data set. Images were reconstructed by using a GBPRBM model with 1500 hidden units.

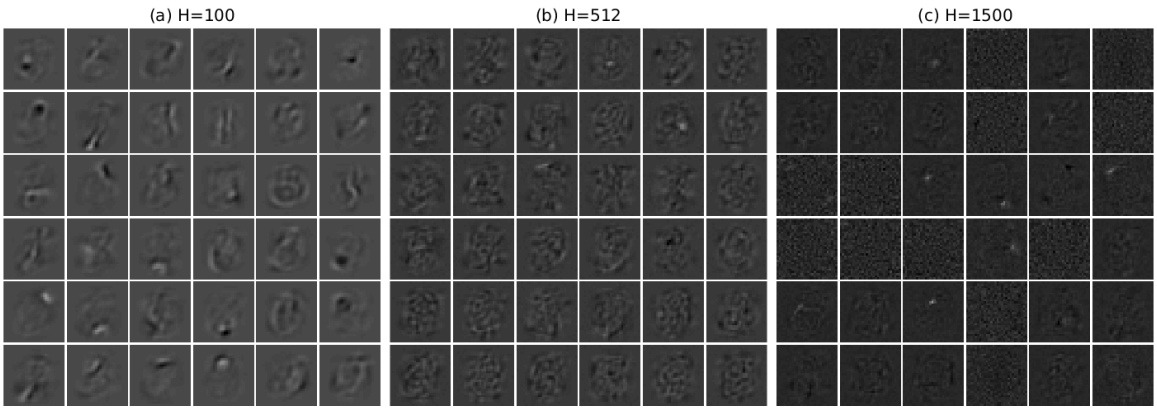

Figure 4.2: Some of the filters (reshaped columns of $\mathbf{W}$) learned from the MNIST data set for GBPRBM models with (a) 100, (b) 512 and (c) 1500 hidden units.

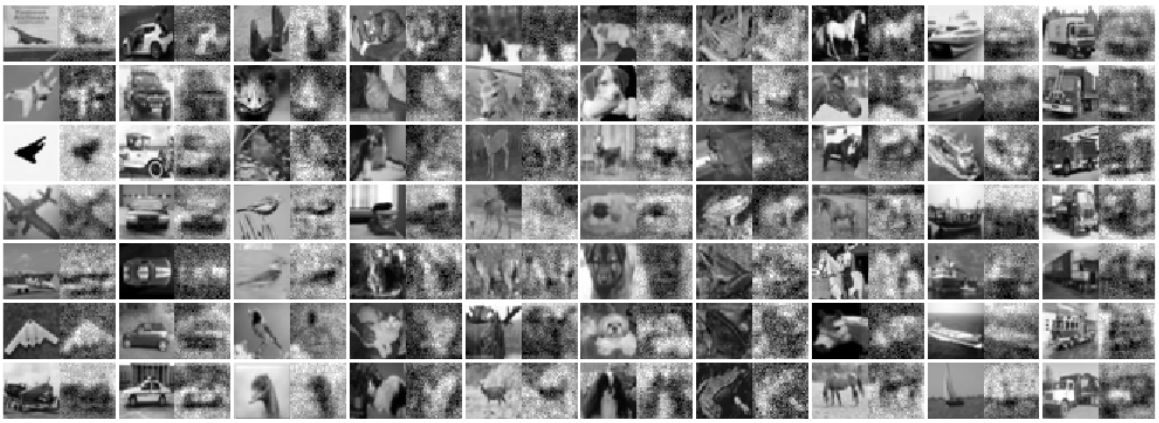

Figure 4.3: Original (left) and reconstructed (right) images from the CIFAR-10 data set. Images were reconstructed by using a GBPRBM model with 1024 hidden units.

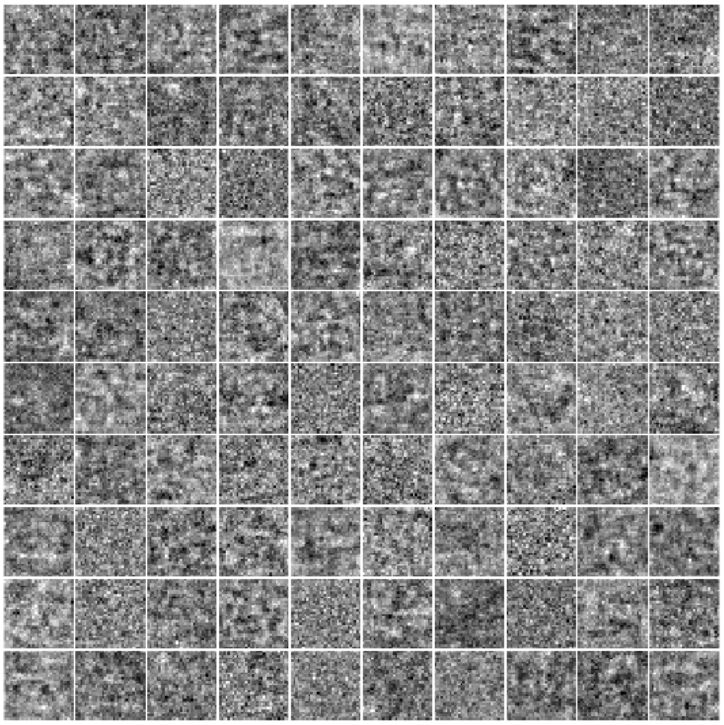

Figure 4.4: Some of the filters (reshaped columns of $\mathbf{W}$) learned from the CIFAR-10 data set for the GBPRBM model with 1024 hidden units.

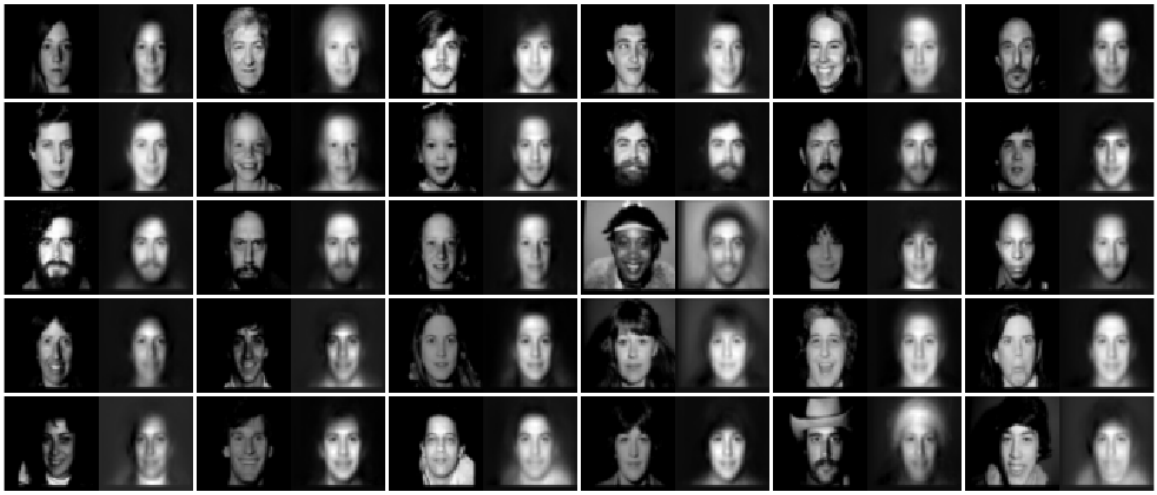

Figure 4.5: Original (left) and reconstructed (right) images from the Faces data set. Images were reconstructed by using a GBPRBM model with 4056 hidden units.

Figure 4.6: Some of the filters (reshaped columns of $\mathbf{W}$) learned from the Faces data set for the GBPRBM model with 4056 hidden units.

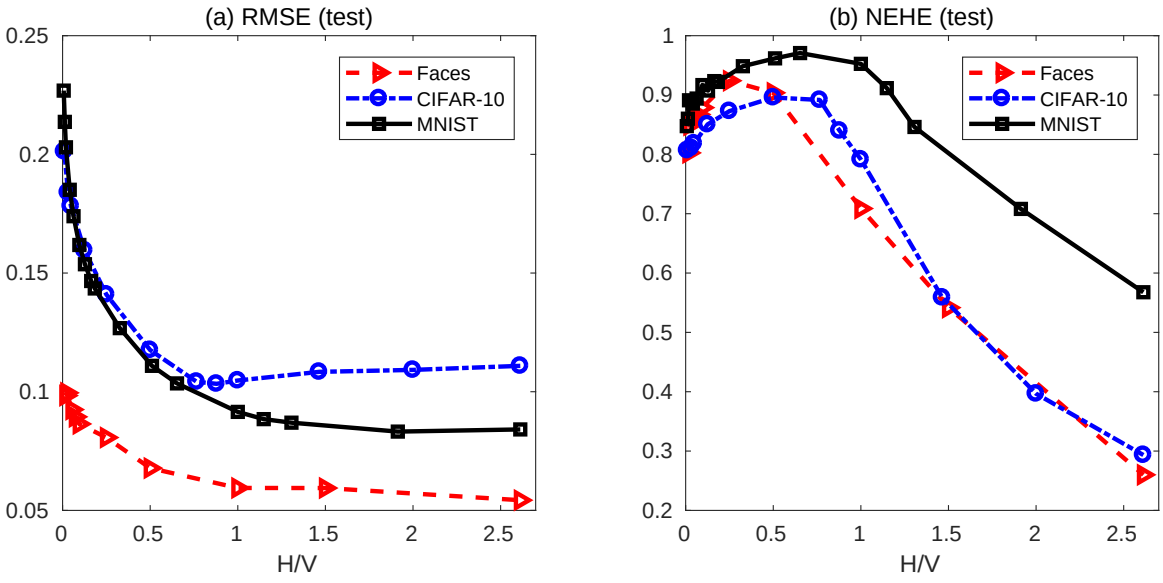

Figure 4.7: (a) Per-pixel root-mean square error and (b) normalized empirical hidden entropy as a function of H/V ratio for different GBPRBM models trained using the MNIST, CIFAR-10, and Faces data sets.

Chapter 5

CONCLUSION AND FUTURE WORK

We introduced a GBPRBM model as an alternative to the widely used GBLRBM model and visualized its similarity to a Gaussian mixture model. Also, we proposed to use hidden entropy as a measure of the representational efficiency of the GBPRBM. Within this scope, numerical evaluations show that the hidden bias term plays crucial role in representational efficiency of the model. Hence we presented a methodology for analysis of hidden entropy as a function of hidden bias. In this analysis, point $\mathbf{b}_h^*$ in the hidden bias space which activates most distant components in $p(\mathbf{v})$, was deduced. Besides, conditions necessary to attain maximum hidden entropy were also stated. These conditions put constraints on the geometry of the model, requiring weight matrix $\mathbf{W}$ to have orthogonal columns when $\mathbf{\Sigma}$ is a scalar matrix. In the space of visible units, this corresponds to the weights' span of a hypercube with 2^H active centroids, where H is the number of hidden units. Obviously, such geometry is not suitable for data clustering in real life. Modeling data requires updating the geometry $\{\mathbf{W}, \mathbf{b}_v\}$ to match centroid locations. This causes decreasing of the number of active centroids for the the same H.

Findings given above provide an insight on the number of hidden units needed to be chosen. If the number of clusters C_N in the data is roughly known a priori, then H should be greater than $\log_2(C_N)$. Incrementing H just by one doubles the number of the centroids covered by $\mathbf{Wh}$ for all hidden vectors. This exponential property shows the representational power of the model.

Our observations show that hidden bias controls the expression of Gaussian components in $p(\mathbf{v})$. Sparsity of the hidden entropy vs. hidden bias space indicates that only a few regions possess high hidden entropy values. Improperly set hidden bias yields $p(\mathbf{v})$ with a single active Gaussian component making modeling useless. Also we demonstrated how different values of the hidden bias suppress or activate different Gaussian components in $p(\mathbf{v})$. Taking this phenomenon into account, we derived high entropy regions with 1, $\log_2(3)$, 2 bits for

models with 1, 2 and 3 hidden units analytically.

Furthermore, we introduced Normalized Empirical Hidden Entropy (NEHE) as an alternative to hidden entropy to measure usefulness of hidden units. It also serves as an upper bound for the hidden entropy. Experiments with MNIST, CIFAR-10 and Faces data sets indicate that based on the value of NEHE, the ratio of the number of hidden units to the number of visible units H/V may need to be in the range of $[0.25, 1]$ or greater to efficiently represent the data. At this point, adding extra hidden units does not improve performance of the model and redundant hidden units are barely active. Although this behavior depends on the nature of the data, this may be a good guess to estimate minimum number of hidden units required.

Moreover, the experiments also show that the visible bias should be set to the mean of the data and weights should be initialized with small magnitudes at the beginning of the contrastive divergence training. This will stabilize and accelerate convergence of the algorithm.

As a future study, it would be interesting to analyze hidden entropy for models with large number of hidden units. The motivation behind the use of hidden bias with high hidden entropy regions is to eliminate redundant hidden units whilst preserving similar geometry. In addition, a few interesting questions arise. How does pruning hidden units affect performance of the model? In the same fashion, the inverse problem is whether it is possible to add hidden units during training and speed up convergence. What should be the minimum number of the hidden units to be added? We partially answered this question based on the NEHE measure, however, it would be interesting to use original hidden entropy measure for this anlaysis. Inevitably, dealing with models with large number of hidden units is challenging, since computations related to the hidden entropy require a large number of mathematical operations due to the combinatorial nature of the problem. Another important task is to check whether supervised initialization of hidden biases is feasible for large models and if so, does it speed up the conventional CD, PCD and PT algorithms [17, 18, 19]. Provided analysis also shows that the hidden bias space has very few high hidden entropy regions. In order to overcome this inefficiency, the energy function used to define the joint probability of the visible and hidden units can be modified.

Appendix A

A.1 Probability of Visible Vector Given Hidden Vector

Conditional probability of observing $\mathbf{v}$ given $\mathbf{h}$ is given as:

$$
\begin{aligned}
p(\mathbf{v}|\mathbf{h}) &= \frac{p(\mathbf{v},\mathbf{h})}{p(\mathbf{h})} \\
&= \frac{\exp(-E(\mathbf{v},\mathbf{h}))}{\int_{\mathbf{u}} \exp(-E(\mathbf{u},\mathbf{h}))d\mathbf{u}} \\
&= \frac{\exp\left(-\sum_{i=1}^{V}\frac{(v_i-b_i^v)^2}{2\sigma_i^2}+\sum_{i=1}^{V}\sum_{j=1}^{H}\frac{v_i}{\sigma_i^2}h_j w_{ij}+\sum_{j=1}^{H}h_j b_j^h\right)}{\int_{\mathbf{u}}\exp\left(-\sum_{i=1}^{V}\frac{(u_i-b_i^v)^2}{2\sigma_i^2}+\sum_{i=1}^{V}\sum_{j=1}^{H}\frac{u_i}{\sigma_i^2}h_j w_{ij}+\sum_{j=1}^{H}h_j b_j^h\right)d\mathbf{u}} \\
&= \frac{\exp\left(-\sum_{i=1}^{V}\frac{(v_i-b_i^v)^2}{2\sigma_i^2}+\sum_{i=1}^{V}\sum_{j=1}^{H}\frac{v_i}{\sigma_i^2}h_j w_{ij}\right)}{\int_{\mathbf{u}}\exp\left(-\sum_{i=1}^{V}\frac{(u_i-b_i^v)^2}{2\sigma_i^2}+\sum_{i=1}^{V}\sum_{j=1}^{H}\frac{u_i}{\sigma_i^2}h_j w_{ij}\right)d\mathbf{u}} \\
&= \frac{\exp\left(-\sum_{i=1}^{V}\left(\frac{(v_i-b_i^v)^2}{2\sigma_i^2}-\sum_{j=1}^{H}\frac{v_i}{\sigma_i^2}h_j w_{ij}\right)\right)}{\int_{\mathbf{u}}\exp\left(-\sum_{i=1}^{V}\left(\frac{(u_i-b_i^v)^2}{2\sigma_i^2}-\sum_{j=1}^{H}\frac{u_i}{\sigma_i^2}h_j w_{ij}\right)\right)d\mathbf{u}} \\
&= \frac{\prod_{i=1}^{V}\exp\left(-\frac{(v_i-b_i^v)^2}{2\sigma_i^2}+\sum_{j=1}^{H}\frac{v_i}{\sigma_i^2}h_j w_{ij}\right)}{\prod_{i=1}^{V}\left[\int_{u_i=-\infty}^{u_i=\infty}\exp\left(-\frac{(u_i-b_i^v)^2}{2\sigma_i^2}+\sum_{j=1}^{H}\frac{u_i}{\sigma_i^2}h_j w_{ij}\right)du_i\right]} \\
&= \prod_{i=1}^{V}\frac{\exp\left(-\frac{(v_i-b_i^v)^2}{2\sigma_i^2}+\sum_{j=1}^{H}\frac{v_i}{\sigma_i^2}h_j w_{ij}\right)}{\int_{u_i=-\infty}^{u_i=\infty}\exp\left(-\frac{(u_i-b_i^v)^2}{2\sigma_i^2}+\sum_{j=1}^{H}\frac{u_i}{\sigma_i^2}h_j w_{ij}\right)du_i}
\end{aligned}
\tag{A.1}
$$

The denominator term can be expressed as:

$$
\int_{u_i=-\infty}^{u_i=\infty}\exp\left(-\frac{(u_i-b_i^v)^2}{2\sigma_i^2}+\sum_{j=1}^{H}\frac{u_i}{\sigma_i^2}h_j w_{ij}\right)du_i = \tag{A.2}
$$

$$
\int_{u_i=-\infty}^{u_i=\infty}\exp\left(-\frac{u_i^2}{2\sigma_i^2}+\frac{u_i b_i^v}{\sigma_i^2}-\frac{(b_i^v)^2}{2\sigma_i^2}+\sum_{j=1}^{H}\frac{u_i}{\sigma_i^2}h_j w_{ij}\right)du_i \tag{A.3}
$$

Using Gaussian integral expression:

$$
\int_{-\infty}^{\infty}e^{-ax^2+bx+c}\,dx=\sqrt{\frac{\pi}{a}}\,e^{\frac{b^2}{4a}+c} \tag{A.4}
$$

where parameters are given by

$$\begin{aligned} a &= \frac{1}{2\sigma_i^2} \\ b &= \frac{1}{\sigma_i^2}\left[b_i^v + \sum_{j=1}^{H} h_j w_{ij}\right] \\ c &= -\frac{(b_i^v)^2}{2\sigma_i^2} \end{aligned} \tag{A.5}$$

yields

$$\begin{aligned} &\int_{u_i=-\infty}^{u_i=\infty} \exp\left(-\frac{(u_i - b_i^v)^2}{2\sigma_i^2} + \sum_{j=1}^{H} \frac{u_i}{\sigma_i^2} h_j w_{ij}\right) du_i = \\ &= \sqrt{2\sigma_i^2 \pi} \exp\left(\frac{\left[\frac{1}{\sigma_i^2}\left[b_i^v + \sum_{j=1}^{H} h_j w_{ij}\right]\right]^2}{4/(2\sigma_i^2)} - \frac{(b_i^v)^2}{2\sigma_i^2}\right) \\ &= \sigma_i \sqrt{2\pi} \exp\left(\frac{\left[b_i^v + \sum_{j=1}^{H} h_j w_{ij}\right]^2}{2\sigma_i^2} - \frac{(b_i^v)^2}{2\sigma_i^2}\right) \end{aligned} \tag{A.6}$$

$$
\begin{aligned}
p(\mathbf{v}|\mathbf{h}) &= \prod_{i=1}^{V} \frac{\exp\left(-\frac{(v_i-b_i^v)^2}{2\sigma_i^2} + \sum_{j=1}^{H} \frac{v_i}{\sigma_i^2} h_j w_{ij}\right)}{\sigma_i\sqrt{2\pi}\exp\left(\frac{\left[b_i^v+\sum_{j=1}^{H} h_j w_{ij}\right]^2}{2\sigma_i^2} - \frac{(b_i^v)^2}{2\sigma_i^2}\right)} \\
&= \prod_{i=1}^{V} \frac{\exp\left(-\frac{(v_i-b_i^v)^2}{2\sigma_i^2} + \sum_{j=1}^{H} \frac{v_i}{\sigma_i^2} h_j w_{ij}\right)}{\sigma_i\sqrt{2\pi}\exp\left(\frac{1}{2\sigma_i^2}\left[(b_i^v)^2 + 2b_i^v\sum_{j=1}^{H} h_j w_{ij} + \left(\sum_{j=1}^{H} h_j w_{ij}\right)^2 - (b_i^v)^2\right]\right)} \\
&= \prod_{i=1}^{V} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(\frac{1}{2\sigma_i^2}\left(-(v_i - b_i^v)^2 + 2v_i\sum_{j=1}^{H} h_j w_{ij} - 2b_i^v\sum_{j=1}^{H} h_j w_{ij} - \left(\sum_{j=1}^{H} h_j w_{ij}\right)^2\right)\right) \\
&= \prod_{i=1}^{V} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(\frac{1}{2\sigma_i^2}\left(-v_i^2 + 2v_i b_i^v - (b_i^v)^2 + 2v_i\sum_{j=1}^{H} h_j w_{ij} - 2b_i^v\sum_{j=1}^{H} h_j w_{ij} - \left(\sum_{j=1}^{H} h_j w_{ij}\right)^2\right)\right) \\
&= \prod_{i=1}^{V} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(\frac{1}{2\sigma_i^2}\left(-v_i^2 + 2v_i\left[b_i^v + \sum_{j=1}^{H} h_j w_{ij}\right] - \left[b_i^v + \sum_{j=1}^{H} h_j w_{ij}\right]^2\right)\right) \\
&= \prod_{i=1}^{V} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{\left(v_i - \left[b_i^v + \sum_{j=1}^{H} h_j w_{ij}\right]\right)^2}{2\sigma_i^2}\right) \\
&= \prod_{i=1}^{V} \mathcal{N}\left(v_i;\ [b_i^v + \sum_{j=1}^{H} h_j w_{ij}],\ \sigma_i^2\right) = \prod_{i=1}^{V} p(v_i|\mathbf{h}) \\
&= \frac{1}{\sqrt{(2\pi)^V \det(\mathbf{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{v} - (\mathbf{b}_v + \mathbf{W}\mathbf{h}))^T \mathbf{\Sigma}^{-1}(\mathbf{v} - (\mathbf{b}_v + \mathbf{W}\mathbf{h}))\right) \qquad \text{(A.7)}
\end{aligned}
$$

where

$$
p(v_i|\mathbf{h}) = \mathcal{N}\left(v_i;\ [b_i^v + \sum_{j=1}^{H} h_j w_{ij}],\ \sigma_i^2\right) \qquad \text{(A.8)}
$$

A.2 Probability of Hidden Vector Given Visible Vector

Similarly, by using definition of the energy function, conditional probability $p(\mathbf{h}|\mathbf{v})$ can be derived:

$$\begin{aligned} p(\mathbf{h}|\mathbf{v}) &= \frac{p(\mathbf{v},\mathbf{h})}{p(\mathbf{v})} = \frac{\exp(-E(\mathbf{v},\mathbf{h}))}{\sum_{\mathbf{g}}\exp(-E(\mathbf{v},\mathbf{g}))} \\ &= \frac{\exp\left(-\sum_{i=1}^{V}\frac{(v_i-b_i^v)^2}{2\sigma_i^2}+\sum_{i=1}^{V}\sum_{j=1}^{H}\frac{v_i}{\sigma_i^2}h_j w_{ij}+\sum_{j=1}^{H}h_j b_j^h\right)}{\sum_{\mathbf{g}}\exp\left(-\sum_{i=1}^{V}\frac{(v_i-b_i^v)^2}{2\sigma_i^2}+\sum_{i=1}^{V}\sum_{j=1}^{H}\frac{v_i}{\sigma_i^2}g_j w_{ij}+\sum_{j=1}^{H}g_j b_j^h\right)} \\ &= \frac{\exp\left(\sum_{i=1}^{V}\sum_{j=1}^{H}\frac{v_i}{\sigma_i^2}h_j w_{ij}+\sum_{j=1}^{H}h_j b_j^h\right)}{\sum_{\mathbf{g}}\exp\left(\sum_{i=1}^{V}\sum_{j=1}^{H}\frac{v_i}{\sigma_i^2}g_j w_{ij}+\sum_{j=1}^{H}g_j b_j^h\right)} \\ &= \frac{\exp\left(\sum_{j=1}^{H}h_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)}{\sum_{\mathbf{g}}\exp\left(\sum_{j=1}^{H}g_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)} \\ &= \frac{\prod_{j=1}^{H}\exp\left(h_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)}{\sum_{\mathbf{g}}\prod_{j=1}^{H}\exp\left(g_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)} \end{aligned} \tag{A.9}$$

$$\begin{aligned} p(\mathbf{h}|\mathbf{v}) &= \frac{\prod_{j=1}^{H}\exp\left(h_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)}{\sum_{g_1\in\pm1}\cdots\sum_{g_H\in\pm1}\exp\left(g_1\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{i,1}+b_1^h\right)\right)\ldots\exp\left(g_H\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{i,H}+b_H^h\right)\right)} \\ &= \frac{\prod_{j=1}^{H}\exp\left(h_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)}{\prod_{j=1}^{H}\left[\sum_{g_j\in\pm1}\exp\left(g_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)\right]} \\ &= \prod_{j=1}^{H}\frac{\exp\left(h_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)}{\sum_{g_j\in\pm1}\exp\left(g_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)} \\ &= \prod_{j=1}^{H}\frac{\exp\left(h_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)}{\exp\left(-\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)+\exp\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)} \\ &= \prod_{j=1}^{H}\frac{\exp\left(h_j\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)\right)}{2\cosh\left(\sum_{i=1}^{V}\frac{v_i}{\sigma_i^2}w_{ij}+b_j^h\right)} = \prod_{j=1}^{H}p(h_j|\mathbf{v}). \end{aligned} \tag{A.10}$$

It follows that

$$p(h_j|\mathbf{v}) = \text{sigm}\left(2h_j\left(\mathbf{v}^T\mathbf{\Sigma}^{-1}\mathbf{W}(:,j)+b_j^h\right)\right). \tag{A.11}$$

A.3 Probability of Hidden Vector

Probability of hidden vector is defined as:

$$
\begin{aligned}
p(\mathbf{h}) &= \int_{\mathbf{v}} p(\mathbf{v},\mathbf{h})d\mathbf{v} = \frac{\int_{\mathbf{v}} \exp(-E(\mathbf{v},\mathbf{h}))d\mathbf{v}}{\int_{\mathbf{u}}\sum_{\mathbf{g}} \exp(-E(\mathbf{u},\mathbf{g}))d\mathbf{u}} = \frac{1}{Z}\int_{\mathbf{v}} \exp(-E(\mathbf{v},\mathbf{h}))d\mathbf{v} \\
&= \frac{\exp\left(\sum_{j=1}^{H} h_j b_j^h\right)}{Z}\prod_{i=1}^{V}\sigma_i\sqrt{2\pi}\exp\left(\frac{[b_i^v+\sum_{j=1}^{H}h_j w_{ij}]^2}{2\sigma_i^2} - \frac{(b_i^v)^2}{2\sigma_i^2}\right) \\
&= \frac{\exp\left(\sum_{j=1}^{H} h_j b_j^h\right)(2\pi)^{V/2}}{Z}\prod_{i=1}^{V}\sigma_i\exp\left(\frac{1}{2\sigma_i^2}\left[(b_i^v)^2 + 2b_i^v\sum_{j=1}^{H}h_j w_{ij} + \left(\sum_{j=1}^{H}h_j w_{ij}\right)^2 - (b_i^v)^2\right]\right) \\
&= \frac{\exp\left(\sum_{j=1}^{H} h_j b_j^h\right)(2\pi)^{V/2}}{Z}\prod_{i=1}^{V}\sigma_i\exp\left(\frac{1}{2\sigma_i^2}\left[2b_i^v\sum_{j=1}^{H}h_j w_{ij} + \left(\sum_{j=1}^{H}h_j w_{ij}\right)^2\right]\right) \\
&= \frac{\exp\left(\sum_{j=1}^{H} h_j b_j^h\right)(2\pi)^{V/2}\prod_{i=1}^{V}\sigma_i}{Z}\exp\left(\sum_{i=1}^{V}\frac{1}{2\sigma_i^2}\left[2b_i^v+\sum_{j=1}^{H}h_j w_{ij}\right]\sum_{j=1}^{H}h_j w_{ij}\right) \\
&= \frac{(2\pi)^{V/2}\sqrt{\det(\mathbf{\Sigma})}\exp\left(\mathbf{b}_h^T\mathbf{h}\right)}{Z}\exp\left(\frac{1}{2}(2\mathbf{b}_v+\mathbf{W}\mathbf{h})^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{h}\right) \\
&= \frac{\sqrt{(2\pi)^V\det(\mathbf{\Sigma})}}{Z}\exp\left(\mathbf{b}_v^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{h} + \frac{1}{2}\mathbf{h}^T\mathbf{W}^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{h} + \mathbf{b}_h^T\mathbf{h}\right) \qquad \text{(A.12)}
\end{aligned}
$$

Computing normalizing constant Z:

$$
\begin{aligned}
Z &= \sum_{\mathbf{h}}\int_{\mathbf{v}}\exp(-E(\mathbf{v},\mathbf{h}))d\mathbf{v} \\
&= \sqrt{(2\pi)^V\det(\mathbf{\Sigma})}\sum_{\mathbf{h}}\exp\left(\frac{1}{2}(2\mathbf{b}_v+\mathbf{W}\mathbf{h})^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{h} + \mathbf{b}_h^T\mathbf{h}\right) \qquad \text{(A.13)}
\end{aligned}
$$

Simplification of the $\sqrt{(2\pi)^V\det(\mathbf{\Sigma})}$ term gives:

$$
p(\mathbf{h}) = \frac{\exp\left(\mathbf{b}_v^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{h} + \frac{1}{2}\mathbf{h}^T\mathbf{W}^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{h} + \mathbf{b}_h^T\mathbf{h}\right)}{\sum_{\mathbf{g}}\exp\left(\mathbf{b}_v^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{g} + \frac{1}{2}\mathbf{g}^T\mathbf{W}^T\mathbf{\Sigma}^{-1}\mathbf{W}\mathbf{g} + \mathbf{b}_h^T\mathbf{g}\right)}. \qquad \text{(A.14)}
$$

A.4 One-Bit Hidden Entropy Region With a Single Antipode Hidden Unit

Let us reindex model parameters and the vector $\mathbf{h}$ to separate hidden unit h_j from the rest of the hidden units:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{(:,1:j-1)} & \mathbf{W}_{(:,j)} & \mathbf{W}_{(:,j+1:H)} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_P & \mathbf{W}_{(:,j)} & \mathbf{W}_N \end{bmatrix} \quad \text{and}$$
$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_{(1:j-1)} \\ h_j \\ \mathbf{h}_{(j+1:H)} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_P \\ h_j \\ \mathbf{h}_N \end{bmatrix}, \ \mathbf{b}_h = \begin{bmatrix} \mathbf{b}_{h,(1:j-1)} \\ b_j^h \\ \mathbf{b}_{h,(j+1:H)} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{h,P} \\ b_j^h \\ \mathbf{b}_{h,N} \end{bmatrix}. \tag{A.15}$$

Now recall the energy term $A(\mathbf{h})$ defined in $p(\mathbf{h})$:

$$A(\mathbf{h}) = A(h_j, \mathbf{h}_P, \mathbf{h}_N) = \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}\mathbf{h} + \frac{1}{2}\mathbf{h}^T \mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}\mathbf{h} + \mathbf{b}_h^T \mathbf{h}. \tag{A.16}$$

It is logically to assume that $A(-1, \mathbf{h}_P, \mathbf{h}_N)$ is equal to $A(+1, \mathbf{h}_P, \mathbf{h}_N)$ since both configurations $[\mathbf{h}_P, -1, \mathbf{h}_N]$ and $[\mathbf{h}_P, +1, \mathbf{h}_N]$ are equiprobable and all other configurations have probability close to zero. Then the first term of $A(\mathbf{h})$ is equal to:

$$\mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}\mathbf{h} = \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_P \mathbf{h}_P + h_j \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)} + \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_N \mathbf{h}_N. \tag{A.17}$$

Similarly, the second term of $A(\mathbf{h})$ can be expressed as:

$$\begin{aligned} \mathbf{h}^T \mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}\mathbf{h} = 2h_j \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} (\mathbf{W}_P \mathbf{h}_P + \mathbf{W}_N \mathbf{h}_N) + (h_j)^2 \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)} + \\ + \mathbf{h}_P^T \mathbf{W}_P^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_P \mathbf{h}_P + \mathbf{h}_N^T \mathbf{W}_N^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_N \mathbf{h}_N + 2\mathbf{h}_P^T \mathbf{W}_P^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_N \mathbf{h}_N. \end{aligned} \tag{A.18}$$

Note that $(h_j)^2$ is always equal to 1. Analogously, the third term of $A(\mathbf{h})$ is equal to:

$$\mathbf{b}_h^T \mathbf{h} = \mathbf{b}_{h,P}^T \mathbf{h}_P + h_j b_j^h + \mathbf{b}_{h,N}^T \mathbf{h}_N. \tag{A.19}$$

To summarize, the energy term has a form of:

$$\begin{aligned} A(h_j, \mathbf{h}_P, \mathbf{h}_N) &= h_j \left(\mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)} + \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} (\mathbf{W}_P \mathbf{h}_P + \mathbf{W}_N \mathbf{h}_N) + b_j^h \right) \\ &\quad + C(\mathbf{h}_P, \mathbf{h}_N) \\ &= h_j \left(b_j^h + \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} (\mathbf{b}_v + \mathbf{W}_P \mathbf{h}_P + \mathbf{W}_N \mathbf{h}_N) \right) + C(\mathbf{h}_P, \mathbf{h}_N) \\ &= h_j \left(b_j^h + B(\mathbf{h}_P, \mathbf{h}_N) \right) + C(\mathbf{h}_P, \mathbf{h}_N) \end{aligned} \tag{A.20}$$

where $B(\mathbf{h}_P, \mathbf{h}_N)$ and $C(\mathbf{h}_P, \mathbf{h}_N)$ are terms independent of h_j:

$$B(\mathbf{h}_P, \mathbf{h}_N) = \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1}(\mathbf{b}_v + \mathbf{W}_P \mathbf{h}_P + \mathbf{W}_N \mathbf{h}_N) \tag{A.21}$$

and

$$\begin{aligned} C(\mathbf{h}_P, \mathbf{h}_N) &= \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1}(\mathbf{W}_P \mathbf{h}_P + \mathbf{W}_N \mathbf{h}_N) + \mathbf{b}_{h,P}^T \mathbf{h}_P + \mathbf{b}_{h,N}^T \mathbf{h}_N + & \text{(A.22)} \\ &+ \frac{1}{2}\Big(\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)} + \mathbf{h}_P^T \mathbf{W}_P^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_P \mathbf{h}_P + \\ &+ \mathbf{h}_N^T \mathbf{W}_N^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_N \mathbf{h}_N + 2\mathbf{h}_P^T \mathbf{W}_P^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_N \mathbf{h}_N\Big) & \text{(A.23)} \\ &= \left(\mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W} + \mathbf{b}_h^T\right) \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix} + \frac{1}{2} \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)} + & \text{(A.24)} \\ &+ \frac{1}{2} \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix}^T \mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W} \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix}. & \text{(A.25)} \end{aligned}$$

The value of b_j^h which satisfies the following condition

$$A(-1, \mathbf{h}_P, \mathbf{h}_N) = A(+1, \mathbf{h}_P, \mathbf{h}_N) \tag{A.26}$$

can be found by solving the equation:

$$\begin{aligned} &+ \left(b_j^h + \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1}(\mathbf{b}_v + \mathbf{W}_P \mathbf{h}_P + \mathbf{W}_N \mathbf{h}_N)\right) + C = \\ &- \left(b_j^h + \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1}(\mathbf{b}_v + \mathbf{W}_P \mathbf{h}_P + \mathbf{W}_N \mathbf{h}_N)\right) + C. \end{aligned} \tag{A.27}$$

The solution is:

$$b_j^h = -\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1}(\mathbf{b}_v + \mathbf{W}_P \mathbf{h}_P + \mathbf{W}_N \mathbf{h}_N) = -\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + \mathbf{W} \begin{bmatrix} \mathbf{h}_{(1:j-1)} \\ 0 \\ \mathbf{h}_{(j+1:H)} \end{bmatrix}\right). \tag{A.28}$$

Now let us find constraints for $\mathbf{b}_{h,P}$ and $\mathbf{b}_{h,N}$. Recall that $p(\mathbf{h})$ is defined as:

$$p(\mathbf{h}) = \frac{\exp(A(\mathbf{h}))}{\sum_{\mathbf{g}} \exp(A(\mathbf{g}))} \tag{A.29}$$

Plugging the obtained value of b_j^h given in (A.28) into (A.29) yields:

$$\begin{aligned} &p(\mathbf{h}_P, h_j = +1, \mathbf{h}_N) = p(\mathbf{h}_P, h_j = -1, \mathbf{h}_N) \\ &= \frac{\exp\left(C(\mathbf{h}_P, \mathbf{h}_N)\right)}{2\exp\left(C(\mathbf{h}_P, \mathbf{h}_N)\right) + \sum\limits_{\forall \mathbf{g}_P, \mathbf{g}_N}^{\setminus \mathbf{h}_P, \mathbf{h}_N} \sum\limits_{g_j = \pm 1} \exp(g_j(b_j^h + B(\mathbf{g}_P, \mathbf{g}_N))) \exp(C(\mathbf{g}_P, \mathbf{g}_N))} \end{aligned} \tag{A.30}$$

Dividing both numerator and denominator by $\exp\left(C(\mathbf{h}_P, \mathbf{h}_N)\right)$ produces:

$$\begin{aligned} & p(\mathbf{h}_P, h_j = +1, \mathbf{h}_N) = p(\mathbf{h}_P, h_j = -1, \mathbf{h}_N) \\ &= \frac{1}{2 + \sum\limits_{\forall \mathbf{g}_P, \mathbf{g}_N}^{\setminus \mathbf{h}_P, \mathbf{h}_N} \sum\limits_{g_j = \pm 1} \exp(g_j(b_j^h + B(\mathbf{g}_P, \mathbf{g}_N))) \exp(C(\mathbf{g}_P, \mathbf{g}_N) - C(\mathbf{h}_P, \mathbf{h}_N))} \\ &= \frac{1}{2 + F(\mathbf{h}_P, \mathbf{h}_N)} \end{aligned} \tag{A.31}$$

where $F(\mathbf{h}_P, \mathbf{h}_N)$ is defined as:

$$F(\mathbf{h}_P, \mathbf{h}_N) = \sum_{\forall \mathbf{g}_P, \mathbf{g}_N}^{\setminus \mathbf{h}_P, \mathbf{h}_N} D(\mathbf{g}_P, \mathbf{g}_N) \exp(E(\mathbf{g}_P, \mathbf{g}_N)) \tag{A.32}$$

where $D(\mathbf{g}_P, \mathbf{g}_N)$ and $E(\mathbf{g}_P, \mathbf{g}_N)$ are defined as:

$$\begin{aligned} D(\mathbf{g}_P, \mathbf{g}_N) &= \sum_{g_j = \pm 1} \exp(g_j(b_j^h + B(\mathbf{g}_P, \mathbf{g}_N))) \\ &= \frac{2\left[\exp(b_j^h + B(\mathbf{g}_P, \mathbf{g}_N)) + \exp(-(b_j^h + B(\mathbf{g}_P, \mathbf{g}_N)))\right]}{2} \\ &= 2\cosh(b_j^h + B(\mathbf{g}_P, \mathbf{g}_N)) \\ &= 2\cosh\left(\mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1} \mathbf{W} \left(\begin{bmatrix} \mathbf{g}_P \\ 0 \\ \mathbf{g}_N \end{bmatrix} - \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix}\right)\right) \\ &= 2\cosh\left(\mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1} \mathbf{W} \left(\begin{bmatrix} \mathbf{g}_{(1:j-1)} \\ 0 \\ \mathbf{g}_{(j+1:H)} \end{bmatrix} - \begin{bmatrix} \mathbf{h}_{(1:j-1)} \\ 0 \\ \mathbf{h}_{(j+1:H)} \end{bmatrix}\right)\right) \end{aligned} \tag{A.33}$$

$$
\begin{aligned}
E(\mathbf{g}_P, \mathbf{g}_N) &= C(\mathbf{g}_P, \mathbf{g}_N) - C(\mathbf{h}_P, \mathbf{h}_N) \qquad \text{(A.34)}\\
&= \mathbf{b}_h^T \left(\begin{bmatrix} \mathbf{g}_P \\ 0 \\ \mathbf{g}_N \end{bmatrix} - \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix} \right) + \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W} \left(\begin{bmatrix} \mathbf{g}_P \\ 0 \\ \mathbf{g}_N \end{bmatrix} - \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix} \right) + \\
&+ \frac{1}{2} \left(\begin{bmatrix} \mathbf{g}_P \\ 0 \\ \mathbf{g}_N \end{bmatrix}^T \mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W} \begin{bmatrix} \mathbf{g}_P \\ 0 \\ \mathbf{g}_N \end{bmatrix} - \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix}^T \mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W} \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix} \right) \\
&= (\mathbf{b}_h^T + \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}) \left(\begin{bmatrix} \mathbf{g}_{(1:j-1)} \\ 0 \\ \mathbf{g}_{(j+1:H)} \end{bmatrix} - \begin{bmatrix} \mathbf{h}_{(1:j-1)} \\ 0 \\ \mathbf{h}_{(j+1:H)} \end{bmatrix} \right) + \\
&+ \frac{1}{2} \left(\left\| \boldsymbol{\Sigma}^{-1/2} \mathbf{W} \begin{bmatrix} \mathbf{g}_{(1:j-1)} \\ 0 \\ \mathbf{g}_{(j+1:H)} \end{bmatrix} \right\|_2^2 - \left\| \boldsymbol{\Sigma}^{-1/2} \mathbf{W} \begin{bmatrix} \mathbf{h}_{(1:j-1)} \\ 0 \\ \mathbf{h}_{(j+1:H)} \end{bmatrix} \right\|_2^2 \right)
\end{aligned}
$$

For hidden entropy to be one bit, $F(\mathbf{h}_P, \mathbf{h}_N)$ should be a very small number close to zero:

$$p(h_j = +1, \mathbf{h}_P, \mathbf{h}_N) = \frac{1}{2 + F(\mathbf{h}_P, \mathbf{h}_N)} \approx \frac{1}{2}, \quad \text{if} \qquad \text{(A.35)}$$

$$F(\mathbf{h}_P, \mathbf{h}_N) \text{ is close to zero.} \qquad \text{(A.36)}$$

Empirical evaluations shows that this constraint can be mitigated by setting upper bound for $F(\mathbf{h}_P, \mathbf{h}_N)$ as 1:

$$F(\mathbf{h}_P, \mathbf{h}_N) < 1. \qquad \text{(A.37)}$$

This upper bound corresponds to $\log_2(3)$ bits of hidden entropy value.

A.4.1 Constraints on the Remaining Hidden Bias for Models With $H = 2$

Consider a simple model with two hidden units ($H = 2$), which are indexed as $[\mathbf{h}_P, h_j, \mathbf{h}_N]^T \equiv [h_j, h_a]^T$. The first condition given in (A.28) reduces to:

$$b_j^h = -\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_a \mathbf{W}_{(:,a)} \right). \qquad \text{(A.38)}$$

In this case $F(\mathbf{h}_P, \mathbf{h}_N)$ reduces to:

$$F(h_a) = \sum_{\forall g_a}^{\setminus h_a} D(g_a) \exp(E(g_a)), \tag{A.39}$$

where

$$\begin{aligned} D(g_a) &= 2\cosh\left((g_a - h_a)\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,a)}\right) \\ E(g_a) &= b_a^h (g_a - h_a) + (g_a^2 - h_a^2)\mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,a)} + (g_a - h_a)\mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,a)} \\ &= (g_a - h_a)\left(b_a^h + \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,a)}\right), \quad \text{since } (g_a^2 - h_a^2) = 0. \end{aligned} \tag{A.40}$$

The summation limits of $F(h_a)$ have only one summand term corresponding to $g_a = -h_a$:

$$\begin{aligned} F(h_a) &= D(-h_a)\exp(E(-h_a)) \\ &= 2\cosh\left(-2h_a \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,a)}\right) \exp\left(-2h_a\left(b_a^h + \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,a)}\right)\right) \\ &= \sum_{g_j = \pm 1} \exp\left(-2h_a\left[b_a^h + \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + g_j \mathbf{W}_{(:,j)}\right)\right]\right). \end{aligned} \tag{A.41}$$

For hidden entropy to be 1 bit (two active configurations (h_j, h_a) and $(-h_j, h_a)$), $F(h_a)$ should be smaller than 1. In summary, conditions needed to attain one-bit hidden entropy are listed below:

$$\begin{aligned} b_j^h &= -\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_a \mathbf{W}_{(:,a)}\right), \\ b_a^h &\underset{h_a=-1}{\overset{h_a=+1}{\gtrless}} -\mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)}\right), \quad \text{where} \\ h_j &= -\text{sgn}\left(h_a \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right). \end{aligned} \tag{A.42}$$

It should be noted that upper ($h_a = -1$) or lower ($h_a = +1$) bound for b_a^h is actually a first condition listed in (A.28) for one-bit hidden entropy region of ath hidden bias term. In other words, setting

$$b_a^h = -\mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)}\right) \tag{A.43}$$

yields

$$F(h_a) = 1 + \exp\left(+4h_a h_j \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right). \tag{A.44}$$

For $F(h_a)$ to be close to 1, the exponential term should be a very small number. This is possible only if

$$h_a h_j \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,j)} < 0 \tag{A.45}$$

which gives a solution:

$$h_j = -\text{sgn}\left(h_a \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right). \tag{A.46}$$

Intersection of one-bit hidden entropy regions in ath and jth hidden bias terms is a point where hidden entropy of $\log_2(3)$ bits is achieved:

$$\begin{aligned} b_j^h &= -\mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + h_a \mathbf{W}_{(:,a)}\right), \\ b_a^h &= -\mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)}\right), \quad \text{where} \\ h_j &= -\text{sgn}\left(h_a \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right). \end{aligned} \tag{A.47}$$

Configurations $(-h_j, h_a)$, (h_j, h_a) and $(h_j, -h_a)$ are active where h_j is defined in (A.47).

A.4.2 Constraints on the Remaining Hidden Biases for Models With $H = 3$

Similarly, hidden vector in a model with three hidden units ($H = 3$) can be indexed as $[\mathbf{h}_P, h_j, \mathbf{h}_N]^T \equiv [h_a, h_j, h_b]^T$. For three hidden units the solution is more complex due to combinatorial nature of the problem. The first condition on b_j^h is given in (A.28):

$$b_j^h = -\mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1} \left(\mathbf{b}_v + h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)}\right). \tag{A.48}$$

In this case

$$F(\mathbf{h}_P, \mathbf{h}_N) = \sum_{\forall \mathbf{g}_P, \mathbf{g}_N}^{\setminus \mathbf{h}_P, \mathbf{h}_N} D(\mathbf{g}_P, \mathbf{g}_N) \exp(E(\mathbf{g}_P, \mathbf{g}_N))$$

reduces to

$$F(h_a, h_b) = \sum_{\forall g_a, g_b}^{\setminus h_a, h_b} D(g_a, g_b) \exp(E(g_a, g_b)) \tag{A.49}$$

where

$$D(g_a, g_b) = 2 \cosh\left(\mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1} \left((g_a - h_a)\mathbf{W}_{(:,a)} + (g_b - h_b)\mathbf{W}_{(:,b)}\right)\right) \tag{A.50}$$

$$\begin{aligned} E(g_a, g_b) = {} & b_a^h (g_a - h_a) + b_b^h (g_b - h_b) + (g_a g_b - h_a h_b) \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,b)} + \\ & + \mathbf{b}_v^T \mathbf{\Sigma}^{-1} \left((g_a - h_a)\mathbf{W}_{(:,a)} + (g_b - h_b)\mathbf{W}_{(:,b)}\right) \end{aligned} \tag{A.51}$$

and

$$F(h_a,h_b)=\sum_{\begin{bmatrix}g_a\\g_b\end{bmatrix}=\left\{\begin{bmatrix}-h_a\\h_b\end{bmatrix},\begin{bmatrix}h_a\\-h_b\end{bmatrix},\begin{bmatrix}-h_a\\-h_b\end{bmatrix}\right\}} 2\cosh\left(\mathbf{W}_{(:,j)}^T\boldsymbol{\Sigma}^{-1}\left((g_a-h_a)\mathbf{W}_{(:,a)}+(g_b-h_b)\mathbf{W}_{(:,b)}\right)\right)\times$$

$$\times\exp\left(\left(\left(b_a^h+\mathbf{b}_v\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,a)}\right)(g_a-h_a)+\left(b_b^h+\mathbf{b}_v\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,b)}\right)(g_b-h_b)\right)+(g_ag_b-h_ah_b)\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,b)}\right)$$

$$=2\cosh\left(-2h_a\mathbf{W}_{(:,j)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,a)}\right)\exp\left(-2h_a\left(b_a^h+\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v+h_b\mathbf{W}_{(:,b)}\right)\right)\right)+$$

$$+2\cosh\left(-2h_b\mathbf{W}_{(:,j)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,b)}\right)\exp\left(-2h_b\left(b_b^h+\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v+h_a\mathbf{W}_{(:,a)}\right)\right)\right)+$$

$$+2\cosh\left(-2\mathbf{W}_{(:,j)}^T\boldsymbol{\Sigma}^{-1}\left(h_a\mathbf{W}_{(:,a)}+h_b\mathbf{W}_{(:,b)}\right)\right)\exp\left(-2[h_a\ h_b]\begin{bmatrix}b_a^h+\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\mathbf{b}_v\\b_b^h+\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\mathbf{b}_v\end{bmatrix}\right)$$

$$=\sum_{g_j=\pm1}\exp\left(-2h_a\left(b_a^h+\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v+g_j\mathbf{W}_{(:,j)}+h_b\mathbf{W}_{(:,b)}\right)\right)\right)+$$

$$+\sum_{g_j=\pm1}\exp\left(-2h_b\left(b_b^h+\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v+g_j\mathbf{W}_{(:,j)}+h_a\mathbf{W}_{(:,a)}\right)\right)\right)+$$

$$+\sum_{g_j=\pm1}\exp\left(-2h_a\left(b_a^h+\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v+g_j\mathbf{W}_{(:,j)}\right)\right)-\right.$$

$$\left.-2h_b\left(b_b^h+\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v+g_j\mathbf{W}_{(:,j)}\right)\right)\right) \tag{A.52}$$

$\log_2(3)$-*bit Hidden Entropy Regions*

Now the goal is to find a value of b_a^h which makes hidden entropy equal to $\log_2(3)$ bits. This is possible only if $F(h_a,h_b)$ is close to 1. Setting

$$b_a^h=-\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v+h_j\mathbf{W}_{(:,j)}+h_b\mathbf{W}_{(:,b)}\right) \tag{A.53}$$

yields

$$F(h_a,h_b)=1+\exp\left(+4h_ah_j\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,j)}\right)+$$

$$+\exp\left(-2h_b\left(b_b^h+\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v+h_j\mathbf{W}_{(:,j)}+h_a\mathbf{W}_{(:,a)}\right)\right)\right)+$$

$$+\exp\left(-2h_b\left(b_b^h+\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v-h_j\mathbf{W}_{(:,j)}+h_a\mathbf{W}_{(:,a)}\right)\right)\right)+$$

$$+\exp\left(+2h_ah_b\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,b)}-\right.$$

$$\left.-2h_b\left(b_b^h+\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v+h_j\mathbf{W}_{(:,j)}\right)\right)\right)$$

$$+\exp\left(+2h_a\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(2h_j\mathbf{W}_{(:,j)}+h_b\mathbf{W}_{(:,b)}\right)-\right.$$

$$\left.-2h_b\left(b_b^h+\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v-h_j\mathbf{W}_{(:,j)}\right)\right)\right) \tag{A.54}$$

Regrouping the terms results in:

$$\begin{aligned} F(h_a, h_b) = 1 + \exp\left(+4h_a h_j \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right) + \\ + \exp\left(-2h_b\left(b_b^h + \mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)}\right)\right)\right) + \\ + \exp\left(-2h_b\left(b_b^h + \mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v - h_j \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)}\right)\right)\right) + \\ + \exp\left(-2h_b\left(b_b^h + \mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} - h_a \mathbf{W}_{(:,a)}\right)\right)\right) + \\ + \exp\left(-2h_b\left(b_b^h + \mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v - h_j \mathbf{W}_{(:,j)} - h_a \mathbf{W}_{(:,a)}\right) - \right.\right. \\ \left.\left. -2\frac{h_a h_j}{h_b} \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right)\right). \end{aligned} \tag{A.55}$$

For $F(h_a, h_b)$ to be close to 1, all exponential terms should be very close to zero. This can be achieved if

$$h_j = -\text{sgn}\left(h_a \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right), \tag{A.56}$$

and b_b^h is set to the following values depending on h_b:

$$\begin{aligned} b_b^h < \min(\mathcal{B}_h) \quad \text{if} \quad h_b = -1, \\ b_b^h > \max(\mathcal{B}_h) \quad \text{if} \quad h_b = +1, \end{aligned} \tag{A.57}$$

where

$$\begin{aligned} \mathcal{B}_h = \Big\{ & -\mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)}\right), \\ & -\mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v - h_j \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)}\right), \\ & -\mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} - h_a \mathbf{W}_{(:,a)}\right), \\ & -\mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v - h_j \mathbf{W}_{(:,j)} - h_a \mathbf{W}_{(:,a)}\right) + 2\frac{h_a h_j}{h_b} \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,j)} \Big\}. \end{aligned} \tag{A.58}$$

Configurations $(-h_j, h_a, h_b)$, (h_j, h_a, h_b) and $(h_j, -h_a, h_b)$ are active where h_j is defined in (A.56).

Similarly, for the second term in (A.52), the goal is to find a value of b_b^h which makes hidden entropy equal to $\log_2(3)$ bits. This is possible only if $F(h_a, h_b)$ is close to 1. Setting

$$b_b^h = -\mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)}\right) \tag{A.59}$$

yields

$$\begin{aligned}
F(h_a, h_b) = &\exp\left(-2h_a\left(b_a^h + \mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_j\mathbf{W}_{(:,j)} + h_b\mathbf{W}_{(:,b)}\right)\right)\right) + \\
&+ \exp\left(-2h_a\left(b_a^h + \mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v - h_j\mathbf{W}_{(:,j)} + h_b\mathbf{W}_{(:,b)}\right)\right)\right) + \\
&+ 1 + \exp\left(+4h_bh_j\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,j)}\right) + \\
&+ \exp\left(-2h_a\left(b_a^h + \mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_j\mathbf{W}_{(:,j)}\right)\right) + \right. \\
&\qquad \left. +2h_ah_b\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,b)}\right) \\
&+ \exp\left(-2h_a\left(b_a^h + \mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v - h_j\mathbf{W}_{(:,j)}\right)\right) + \right. \\
&\qquad \left. +2h_b\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(2h_j\mathbf{W}_{(:,j)} + h_a\mathbf{W}_{(:,a)}\right)\right)
\end{aligned} \tag{A.60}$$

Regrouping the terms results in:

$$\begin{aligned}
F(h_a, h_b) = &1 + \exp\left(+4h_bh_j\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,j)}\right) + \\
&+ \exp\left(-2h_a\left(b_a^h + \mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_j\mathbf{W}_{(:,j)} + h_b\mathbf{W}_{(:,b)}\right)\right)\right) + \\
&+ \exp\left(-2h_a\left(b_a^h + \mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v - h_j\mathbf{W}_{(:,j)} + h_b\mathbf{W}_{(:,b)}\right)\right)\right) + \\
&+ \exp\left(-2h_a\left(b_a^h + \mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_j\mathbf{W}_{(:,j)} - h_b\mathbf{W}_{(:,b)}\right)\right)\right) + \\
&+ \exp\left(-2h_a\left(b_a^h + \mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v - h_j\mathbf{W}_{(:,j)} - h_b\mathbf{W}_{(:,b)}\right) - \right.\right. \\
&\qquad \left.\left. -2\frac{h_bh_j}{h_a}\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,j)}\right)\right)
\end{aligned} \tag{A.61}$$

For $F(h_a, h_b)$ to be close to 1, all exponential terms should be very close to zero. This can be achieved if

$$h_j = -\text{sgn}\left(h_b\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,j)}\right), \tag{A.62}$$

and b_a^h is set to the following values depending on h_a:

$$\begin{aligned}
b_a^h &< \min(\mathcal{A}_h) \quad \text{if} \quad h_a = -1, \\
b_a^h &> \max(\mathcal{A}_h) \quad \text{if} \quad h_a = +1,
\end{aligned} \tag{A.63}$$

where

$$\mathcal{A}_h = \Bigg\{ \begin{aligned} &- \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} + h_b \mathbf{W}_{(:,b)} \right), \\ &- \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v - h_j \mathbf{W}_{(:,j)} + h_b \mathbf{W}_{(:,b)} \right), \\ &- \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} - h_b \mathbf{W}_{(:,b)} \right), \\ &- \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v - h_j \mathbf{W}_{(:,j)} - h_b \mathbf{W}_{(:,b)} \right) + 2 \frac{h_b h_j}{h_a} \mathbf{W}_{(:,b)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)} \end{aligned} \Bigg\}. \tag{A.64}$$

Configurations $(-h_j, h_a, h_b)$, (h_j, h_a, h_b) and $(h_j, h_a, -h_b)$ are active where h_j is defined in (A.62).

Similarly, for the last term in (A.52), the goal is to find values of b_a^h and b_b^h which make hidden entropy equal to $\log_2(3)$ bits. This is possible only if $F(h_a, h_b)$ is close to 1. Equating the last term to zero

$$\begin{aligned} &- 2h_a \left(b_a^h + \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} \right) \right) - \\ &- 2h_b \left(b_b^h + \mathbf{W}_{(:,b)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} \right) \right) = 0 \end{aligned} \tag{A.65}$$

is equivalent to:

$$\begin{aligned} & h_a b_a^h + h_b b_b^h + \left(h_b \mathbf{W}_{(:,b)} + h_a \mathbf{W}_{(:,a)} \right)^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} \right) = 0 \\ & b_a^h + \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} \right) = - \frac{h_b}{h_a} \left(b_b^h + \mathbf{W}_{(:,b)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} \right) \right) \\ & b_a^h = - \frac{h_b}{h_a} b_b^h - \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} \right)^T \boldsymbol{\Sigma}^{-1} \left(\frac{h_b}{h_a} \mathbf{W}_{(:,b)} + \mathbf{W}_{(:,a)} \right). \end{aligned} \tag{A.66}$$

This is actually the (A.73) for $h_P = h_a$ and $h_N = h_b$. It describes a plane which separates configurations (h_j, h_a, h_b) and $(h_j, -h_a, -h_b)$. Plugging obtained b_a^h value into (A.52) yields:

$$\begin{aligned} F(h_a, h_b) = & \exp \left(+2h_b \left(b_b^h + \mathbf{W}_{(:,b)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} - h_a \mathbf{W}_{(:,a)} \right) \right) \right) + \\ & + \exp \left(+2h_b \left(b_b^h + \mathbf{W}_{(:,b)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} - h_a \mathbf{W}_{(:,a)} \right) + \right. \right. \\ & \qquad \left. \left. + 2 \frac{h_a h_j}{h_b} \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)} \right) \right) + \\ & + \exp \left(-2h_b \left(b_b^h + \mathbf{W}_{(:,b)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)} \right) \right) \right) + \\ & + \exp \left(-2h_b \left(b_b^h + \mathbf{W}_{(:,b)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v - h_j \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)} \right) \right) \right) + \\ & + 1 + \exp \left(+4h_j \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left(h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)} \right) \right). \end{aligned} \tag{A.67}$$

For $F(h_a, h_b)$ to be close to 1, all exponential terms should be very close to zero. This can be achieved if

$$h_j = -\text{sgn}\left(\mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1}\left(h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)}\right)\right), \tag{A.68}$$

and b_b^h is set to the following values depending on h_b:

$$\begin{aligned} \max(\mathcal{B}_h^n) < b_b^h < \min(\mathcal{B}_h^p) \quad \text{if} \quad h_b = +1, \\ \max(\mathcal{B}_h^p) < b_b^h < \min(\mathcal{B}_h^n) \quad \text{if} \quad h_b = -1, \end{aligned} \tag{A.69}$$

where

$$\mathcal{B}_h^p = \Big\{ \begin{aligned} &- \mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} - h_a \mathbf{W}_{(:,a)}\right), \\ &-\mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v - h_j \mathbf{W}_{(:,j)} - h_a \mathbf{W}_{(:,a)}\right) - 2\frac{h_a h_j}{h_b} \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,j)} \Big\}. \end{aligned}$$

is a set of values corresponding to the first two terms of (A.67) and

$$\mathcal{B}_h^n = \Big\{ \begin{aligned} &- \mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)}\right), \\ &-\mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v - h_j \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)}\right) \Big\}. \end{aligned}$$

is a set of values corresponding to the third and fourth term of (A.67). Configurations $(-h_j, h_a, h_b)$, (h_j, h_a, h_b) and $(h_j, -h_a, -h_b)$ are active where h_j is defined in (A.68).

$\log_2(2)$-*bit Hidden Entropy Regions*

For hidden entropy to be 1 bit (two active configurations (h_j, h_a, h_b) and $(-h_j, h_a, h_b)$), $F(h_a, h_b)$ should be a very small number close to zero. However, the upper bound for $F(h_a, h_b)$ can be set to 1. In summary, conditions needed to attain one-bit hidden entropy are listed below:

$$\begin{aligned} b_a^h \underset{h_a=-1}{\overset{h_a=+1}{\gtrless}} -\mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v + h_j^a \mathbf{W}_{(:,j)} + h_b \mathbf{W}_{(:,b)}\right) \\ \text{where} \quad h_j^a = -\text{sgn}\left(h_a \mathbf{W}_{(:,a)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right), \\ b_b^h \underset{h_b=-1}{\overset{h_b=+1}{\gtrless}} -\mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v + h_j^b \mathbf{W}_{(:,j)} + h_a \mathbf{W}_{(:,a)}\right) \\ \text{where} \quad h_j^b = -\text{sgn}\left(h_b \mathbf{W}_{(:,b)}^T \mathbf{\Sigma}^{-1} \mathbf{W}_{(:,j)}\right), \\ h_a b_a^h + h_b b_b^h > -\left(h_b \mathbf{W}_{(:,b)} + h_a \mathbf{W}_{(:,a)}\right)^T \mathbf{\Sigma}^{-1}\left(\mathbf{b}_v + h_j^{ab} \mathbf{W}_{(:,j)}\right) \\ \text{where} \quad h_j^{ab} = -\text{sgn}\left(\mathbf{W}_{(:,j)}^T \mathbf{\Sigma}^{-1}\left(h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)}\right)\right). \end{aligned} \tag{A.70}$$

A.5 One-Bit Hidden Entropy Region With $(H-1)$ *Antipode Hidden Units*

Another region in the space of hidden bias with one-bit hidden entropy can be derived from the assumption that two configurations of the hidden vector with $(H-1)$ antipode hidden units are equiprobable:

$$\begin{aligned} p(h_j, \mathbf{h}_P, \mathbf{h}_N) &= p(h_j, -\mathbf{h}_P, -\mathbf{h}_N), \\ A(h_j, \mathbf{h}_P, \mathbf{h}_N) &= A(h_j, -\mathbf{h}_P, -\mathbf{h}_N). \end{aligned} \tag{A.71}$$

Plugging definition of $A(h_j, \mathbf{h}_P, \mathbf{h}_N)$, given in (A.20), produces:

$$\begin{aligned} &h_j \left(b_j^h + B(\mathbf{h}_P, \mathbf{h}_N)\right) + C(\mathbf{h}_P, \mathbf{h}_N) = \\ &h_j \left(b_j^h + B(-\mathbf{h}_P, -\mathbf{h}_N)\right) + C(-\mathbf{h}_P, -\mathbf{h}_N), \\ &h_j B(\mathbf{h}_P, \mathbf{h}_N) + C(\mathbf{h}_P, \mathbf{h}_N) = h_j B(-\mathbf{h}_P, -\mathbf{h}_N) + C(-\mathbf{h}_P, -\mathbf{h}_N). \end{aligned} \tag{A.72}$$

Regrouping yields necessary constraints on $\mathbf{b}_{h,P}$ and $\mathbf{b}_{h,N}$ is:

$$\begin{aligned} \left(\mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W} + \mathbf{b}_h^T\right) \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix} + h_j \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W} \begin{bmatrix} \mathbf{h}_P \\ 0 \\ \mathbf{h}_N \end{bmatrix} = 0, \\ \mathbf{b}_{h,P}^T \mathbf{h}_P + \mathbf{b}_{h,N}^T \mathbf{h}_N + \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)}\right)^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{W}_P \mathbf{h}_P + \mathbf{W}_N \mathbf{h}_N\right) = 0. \end{aligned} \tag{A.73}$$

For GBPRBM models with three hidden units $(H = 3)$ and two equiprobable configurations of the hidden vector

$$p(h_j, h_a, h_b) = p(h_j, -h_a, -h_b), \tag{A.74}$$

the necessary conditions and constraints simplify to:

$$\begin{aligned} &h_a b_a^h + h_b b_b^h + \left(h_b \mathbf{W}_{(:,b)} + h_a \mathbf{W}_{(:,a)}\right)^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)}\right) = 0 \\ &b_a^h + \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)}\right) = -\frac{h_b}{h_a} \left(b_b^h + \mathbf{W}_{(:,b)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)}\right)\right) \\ &b_a^h = -\frac{h_b}{h_a} b_b^h - \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)}\right)^T \boldsymbol{\Sigma}^{-1} \left(\frac{h_b}{h_a} \mathbf{W}_{(:,b)} + \mathbf{W}_{(:,a)}\right) \end{aligned} \tag{A.75}$$

Now recall the energy term $A(\mathbf{g})$ defined in $p(\mathbf{g})$:

$$\begin{aligned}
A(\mathbf{g}) = A(g_j, g_a, g_b) &= \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}\mathbf{g} + \frac{1}{2}\mathbf{g}^T \mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}\mathbf{g} + \mathbf{b}_h^T \mathbf{g} \\
&= \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \left(g_a \mathbf{W}_{(:,a)} + g_b \mathbf{W}_{(:,b)} + g_j \mathbf{W}_{(:,j)} \right) + b_j^h g_j + b_b^h g_b - \\
&- \frac{g_a h_b}{h_a} b_b^h - \left(\mathbf{b}_v + h_j \mathbf{W}_{(:,j)} \right)^T \boldsymbol{\Sigma}^{-1} \left(\frac{g_a h_b}{h_a} \mathbf{W}_{(:,b)} + g_a \mathbf{W}_{(:,a)} \right) + \\
&+ \frac{1}{2} \Bigg\{ 2 g_j \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left(g_a \mathbf{W}_{(:,b)} + g_b \mathbf{W}_{(:,a)} \right) + \sum_{k=a,b,j} \mathbf{W}_{(:,k)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,k)} + \\
&\qquad + 2 g_a g_b \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,b)} \Bigg\} \\
&= \left(g_b - \frac{g_a h_b}{h_a} \right) \left(b_b^h + \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,b)} \right) + b_j^h g_j + g_a g_b \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,b)} \\
&+ \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left(g_j \mathbf{b}_v + \left(g_j g_b - \frac{h_j g_a h_b}{h_a} \right) \mathbf{W}_{(:,b)} + g_a (g_j - h_j) \mathbf{W}_{(:,a)} \right) + \\
&+ \sum_{k=a,b,j} \mathbf{W}_{(:,k)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,k)}
\end{aligned} \tag{A.76}$$

To normalize the numerator and the denominator in $p(\mathbf{h})$ we need to subtract constant terms and the term which activate configurations (h_j, h_a, h_b) and $(h_j, -h_a, -h_b)$ of the hidden vector $\mathbf{h}$:

$$\begin{aligned}
B(g_j, g_a, g_b) = A(g_j, g_a, g_b) &- \sum_{k=a,b,j} \mathbf{W}_{(:,k)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,k)} - \\
&- h_j (b_j^h + \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,j)}) - h_a h_b \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,b)} \\
&= \left(g_b - \frac{g_a h_b}{h_a} \right) \left(b_b^h + \mathbf{b}_v^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,b)} \right) + (g_j - h_j) b_j^h + \\
&+ (g_a g_b - h_a h_b) \mathbf{W}_{(:,a)}^T \boldsymbol{\Sigma}^{-1} \mathbf{W}_{(:,b)} + \\
&+ \mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left((g_j - h_j) \left(\mathbf{b}_v + g_a \mathbf{W}_{(:,a)} \right) + \left(g_j g_b - \frac{h_j g_a h_b}{h_a} \right) \mathbf{W}_{(:,b)} \right)
\end{aligned} \tag{A.77}$$

Newly derived $B(g_j, g_a, g_b)$ term is used in defining function $F(h_j, h_a, h_b)$ in the denominator term of $p(\mathbf{h})$:

$$\begin{aligned}
p(h_j, h_a, h_b) &= p(h_j, -h_a, -h_b) \\
&= \frac{1}{2 + F(h_j, h_a, h_b)} \approx \frac{1}{2}, \quad \text{if}
\end{aligned} \tag{A.78}$$

$$F(h_j, h_a, h_b) \text{ is close to zero.} \tag{A.79}$$

where $F(h_j, h_a, h_b)$ is given as:

$$\begin{aligned}
F(h_j, h_a, h_b) = & \sum_{\forall(g_j,g_a,g_b)}^{\setminus(h_j,h_a,h_b),(h_j,-h_a,-h_b)} \exp(B(g_j, g_a, g_b)) \\
& = \exp\left(-2h_b\left(b_b^h + \mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_a\mathbf{W}_{(:,a)} + h_j\mathbf{W}_{(:,j)}\right)\right)\right) \\
& + \exp\left(+2h_b\left(b_b^h + \mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v - h_a\mathbf{W}_{(:,a)} + h_j\mathbf{W}_{(:,j)}\right)\right)\right) \\
& + \exp\left(-2h_j\left(b_j^h + \mathbf{W}_{(:,j)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_a\mathbf{W}_{(:,a)} + h_b\mathbf{W}_{(:,b)}\right)\right)\right) \\
& + \exp\left(-2h_j\left(b_j^h + \mathbf{W}_{(:,j)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v - h_a\mathbf{W}_{(:,a)} - h_b\mathbf{W}_{(:,b)}\right)\right)\right) \\
& + \exp\left(+2h_b\left(b_b^h + \mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v - h_a\mathbf{W}_{(:,a)}\right)\right) - \right. \\
& \qquad \left. - 2h_j\left(b_j^h + \mathbf{W}_{(:,j)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v - h_a\mathbf{W}_{(:,a)}\right)\right)\right) \\
& + \exp\left(-2h_b\left(b_b^h + \mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_a\mathbf{W}_{(:,a)}\right)\right) - \right. \\
& \qquad \left. - 2h_j\left(b_j^h + \mathbf{W}_{(:,j)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_a\mathbf{W}_{(:,a)}\right)\right)\right)
\end{aligned} \tag{A.80}$$

First two terms in (A.80) bring a condition for h_a:

$$h_a = h_b\,\mathrm{sgn}(\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,b)}) \tag{A.81}$$

and a constraint for b_b^h:

$$-\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v - h_a\mathbf{W}_{(:,a)} + h_j\mathbf{W}_{(:,j)}\right) \underset{h_b=-1}{\overset{h_b=+1}{\gtrless}} b_b^h \underset{h_b=-1}{\overset{h_b=+1}{\gtrless}} -\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_a\mathbf{W}_{(:,a)} + h_j\mathbf{W}_{(:,j)}\right), \tag{A.82}$$

which can be simplified to:

$$-\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v + h_bh_s\mathbf{W}_{(:,a)} + h_j\mathbf{W}_{(:,j)}\right) < b_b^h < -\mathbf{W}_{(:,b)}^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{b}_v - h_bh_s\mathbf{W}_{(:,a)} + h_j\mathbf{W}_{(:,j)}\right) \tag{A.83}$$

where supplementary h_s term is defined as:

$$h_s = \mathrm{sgn}(\mathbf{W}_{(:,a)}^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_{(:,b)}). \tag{A.84}$$

The third and the fourth term in (A.80) bring a constraint for b_j^h:

$$\begin{aligned}
b_j^h &< \min(\mathcal{J}_h) \quad \text{if} \quad h_j = -1, \\
b_j^h &> \max(\mathcal{J}_h) \quad \text{if} \quad h_j = +1,
\end{aligned} \tag{A.85}$$

where $\mathcal{J}_h$ is defined as:

$$\mathcal{J}_h = \Big\{ -\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)}\right), \\ -\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v - h_a \mathbf{W}_{(:,a)} - h_b \mathbf{W}_{(:,b)}\right) \Big\}. \tag{A.86}$$

Rewriting it without min and max statements can be done by adding an extra variable h_s^{ab}:

$$b_j^h \underset{h_j=-1}{\overset{h_j=+1}{\gtrless}} -\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} \Big(\mathbf{b}_v - h_j h_s^{ab} (h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)}\Big) \\ \text{where} \quad h_s^{ab} = \operatorname{sgn}(\mathbf{W}_{(:,j)}^T \boldsymbol{\Sigma}^{-1} (h_a \mathbf{W}_{(:,a)} + h_b \mathbf{W}_{(:,b)})). \tag{A.87}$$

The fifth and the sixth terms in (A.80) require that

$$h_b b_b^h - h_j b_j^h < + \left(h_j \mathbf{W}_{(:,j)} - h_b \mathbf{W}_{(:,b)}\right)^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v - h_a \mathbf{W}_{(:,a)}\right), \\ h_b b_b^h + h_j b_j^h > - \left(h_j \mathbf{W}_{(:,j)} + h_b \mathbf{W}_{(:,b)}\right)^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{b}_v + h_a \mathbf{W}_{(:,a)}\right). \tag{A.88}$$

A.6 Contrastive Divergence Learning

Contrastive divergence (CD) algorithm is based on maximum-likelihood estimation, in which parameters of the RBM model are estimated by maximizing log-likelihood function

$$\mathcal{L}(\boldsymbol{\theta}|\mathbf{v}) = \ln p(\mathbf{v};\boldsymbol{\theta}). \tag{A.89}$$

Since a modified gradient ascent algorithm is used in CD, explicit form of the derivatives of the model parameters should be given in the first place.

Let us define two auxiliary terms $Y = \sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h}))$ and $Z = \int_{\mathbf{u}} \sum_{\mathbf{g}} \exp(-E(\mathbf{u},\mathbf{g}))d\mathbf{u}$, then

$$p(\mathbf{v};\boldsymbol{\theta}) = \frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h}))}{\int_{\mathbf{u}} \sum_{\mathbf{g}} \exp(-E(\mathbf{u},\mathbf{g}))d\mathbf{u}} = \frac{Y}{Z}. \tag{A.90}$$

Differentiating the logarithm of $p(\mathbf{v})$ with respect to parameter $\theta \in \{\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h\}$ yields:

$$\begin{aligned} \frac{\partial}{\partial\theta} \ln p(\mathbf{v};\theta) &= \frac{1}{p(\mathbf{v};\theta)} \frac{\partial p(\mathbf{v};\theta)}{\partial\theta} \\ &= \frac{1}{Y/Z} \frac{\left[\frac{\partial Y}{\partial\theta} Z - \frac{\partial Z}{\partial\theta} Y\right]}{Z^2} \\ &= \frac{1}{Y} \frac{\partial Y}{\partial\theta} - \frac{1}{Z} \frac{\partial Z}{\partial\theta}. \end{aligned} \tag{A.91}$$

Nominator term Y and denominator term Z are given as:

$$\frac{\partial Y}{\partial \theta} = \frac{\partial}{\partial \theta} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h})) = -\sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h})) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \theta}, \tag{A.92}$$
$$\frac{\partial Z}{\partial \theta} = \frac{\partial}{\partial \theta} \int_{\mathbf{u}} \sum_{\mathbf{g}} \exp(-E(\mathbf{u}, \mathbf{g})) d\mathbf{u} = -\int_{\mathbf{u}} \sum_{\mathbf{g}} \exp(-E(\mathbf{u}, \mathbf{g})) \frac{\partial E(\mathbf{u}, \mathbf{g})}{\partial \theta} d\mathbf{u}.$$

Substituting derivative terms produces:

$$\begin{aligned}
\frac{\partial}{\partial \theta} \ln p(\mathbf{v}; \theta) &= -\sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{v}, \mathbf{h}))}{\sum_{\mathbf{g}} \exp(-E(\mathbf{v}, \mathbf{g}))} \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \theta} + \\
&\quad + \int_{\mathbf{u}} \sum_{\mathbf{g}} \frac{\exp(-E(\mathbf{u}, \mathbf{g}))}{\int_{\mathbf{p}} \sum_{\mathbf{q}} \exp(-E(\mathbf{p}, \mathbf{q})) d\mathbf{p}} \frac{\partial E(\mathbf{u}, \mathbf{g})}{\partial \theta} d\mathbf{u} \\
&= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \theta} + \int_{\mathbf{u}} \sum_{\mathbf{g}} p(\mathbf{u}, \mathbf{g}) \frac{\partial E(\mathbf{u}, \mathbf{g})}{\partial \theta} d\mathbf{u} \\
&= -\mathbf{E}_{p(\mathbf{h}|\mathbf{v})} \left[\frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \theta} \right] + \mathbf{E}_{p(\mathbf{u}, \mathbf{g})} \left[\frac{\partial E(\mathbf{u}, \mathbf{g})}{\partial \theta} \right] \\
&= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \theta} + \int_{\mathbf{u}} p(\mathbf{u}) \sum_{\mathbf{g}} p(\mathbf{g}|\mathbf{u}) \frac{\partial E(\mathbf{u}, \mathbf{g})}{\partial \theta} d\mathbf{u} \\
&= -\mathbf{E}_{p(\mathbf{h}|\mathbf{v})} \left[\frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \theta} \right] + \mathbf{E}_{p(\mathbf{u})} \mathbf{E}_{p(\mathbf{g}|\mathbf{u})} \left[\frac{\partial E(\mathbf{u}, \mathbf{g})}{\partial \theta} \right].
\end{aligned} \tag{A.93}$$

The derivative of the log-likelihood function with respect to weight w_{ij} is given as:

$$\begin{aligned}
\frac{\partial}{\partial w_{ij}} \ln p(\mathbf{v}) &= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \left[\frac{-v_i h_j}{\sigma_i^2} \right] + \int_{\mathbf{u}} p(\mathbf{u}) \sum_{\mathbf{g}} p(\mathbf{g}|\mathbf{u}) \left[\frac{-u_i g_j}{\sigma_i^2} \right] d\mathbf{u} \\
&= \frac{1}{\sigma_i^2} \left(v_i \sum_{h_j \in \pm 1} p(h_j|\mathbf{v}) h_j \prod_{\substack{k=1 \\ k \neq j}}^{H} \sum_{h_k \in \pm 1} p(h_k|\mathbf{v}) - \int_{\mathbf{u}} p(\mathbf{u}) u_i \sum_{g_j \in \pm 1} p(g_j|\mathbf{u}) g_j \prod_{\substack{k=1 \\ k \neq j}}^{H} \sum_{g_k \in \pm 1} p(g_k|\mathbf{u}) d\mathbf{u} \right) \\
&= \frac{1}{\sigma_i^2} \left(v_i \sum_{h_j \in \pm 1} p(h_j|\mathbf{v}) h_j - \int_{\mathbf{u}} p(\mathbf{u}) u_i \sum_{g_j \in \pm 1} p(g_j|\mathbf{u}) g_j d\mathbf{u} \right) \\
&= \frac{1}{\sigma_i^2} \left(v_i \cdot \tanh \left(\sum_{l=1}^{V} \frac{v_l}{\sigma_l^2} w_{lj} + b_j^h \right) - \int_{\mathbf{u}} p(\mathbf{u}) u_i \cdot \tanh \left(\sum_{l=1}^{V} \frac{u_l}{\sigma_l^2} w_{lj} + b_j^h \right) d\mathbf{u} \right).
\end{aligned} \tag{A.94}$$

The derivative of the log-likelihood function with respect to visible bias term b_i^v is given as:

$$\begin{aligned}\frac{\partial}{\partial b_i^v}\ln p(\mathbf{v}) &= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v})\left[-\frac{v_i - b_i^v}{\sigma_i^2}\right] + \int_{\mathbf{u}} p(\mathbf{u})\sum_{\mathbf{g}} p(\mathbf{g}|\mathbf{u})\left[-\frac{u_i - b_i^v}{\sigma_i^2}\right] d\mathbf{u} \\ &= \frac{1}{\sigma_i^2}\left(v_i - b_i^v - \int_{\mathbf{u}} p(\mathbf{u})\left[u_i - b_i^v\right] d\mathbf{u}\right) \\ &= \frac{1}{\sigma_i^2}\left(v_i - b_i^v - \int_{\mathbf{u}} p(\mathbf{u}) u_i d\mathbf{u} + b_i^v \int_{\mathbf{u}} p(\mathbf{u}) d\mathbf{u}\right) \\ &= \frac{1}{\sigma_i^2}\left(v_i - \int_{\mathbf{u}} p(\mathbf{u}) u_i d\mathbf{u}\right) \\ &= \frac{1}{\sigma_i^2}\left(v_i - \int_{u_i} p(u_i) u_i du_i\right). \end{aligned} \tag{A.95}$$

The derivative of the log-likelihood function with respect to hidden bias term b_j^h is given as:

$$\begin{aligned}\frac{\partial}{\partial b_j^h}\ln p(\mathbf{v}) &= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v})\left[-h_j\right] + \int_{\mathbf{u}} p(\mathbf{u})\sum_{\mathbf{g}} p(\mathbf{g}|\mathbf{u})\left[-g_j\right] d\mathbf{u} \\ &= \sum_{h_j\in\pm 1} p(h_j|\mathbf{v}) h_j \prod_{\substack{k=1\\k\neq j}}^{H}\sum_{h_k\in\pm 1} p(h_k|\mathbf{v}) - \int_{\mathbf{u}} p(\mathbf{u})\sum_{g_j\in\pm 1} p(g_j|\mathbf{u}) g_j \prod_{\substack{k=1\\k\neq j}}^{H}\sum_{g_k\in\pm 1} p(g_k|\mathbf{u}) d\mathbf{u} \\ &= \sum_{h_j\in\pm 1} p(h_j|\mathbf{v}) h_j - \int_{\mathbf{u}} p(\mathbf{u})\sum_{g_j\in\pm 1} p(g_j|\mathbf{u}) g_j d\mathbf{u} \\ &= \tanh\left(\sum_{l=1}^{V}\frac{v_l}{\sigma_l^2} w_{lj} + b_j^h\right) - \int_{\mathbf{u}} p(\mathbf{u})\cdot\tanh\left(\sum_{l=1}^{V}\frac{u_l}{\sigma_l^2} w_{lj} + b_j^h\right) d\mathbf{u}. \end{aligned} \tag{A.96}$$

Since $p(\mathbf{v})$ is unknown calculating $\mathbf{E}_{p(\mathbf{u})}\mathbf{E}_{p(\mathbf{g}|\mathbf{u})}\left[\frac{\partial E(\mathbf{u},\mathbf{g})}{\partial\theta}\right]$ of the derivative term is infeasible. To deal with this intractable expression, an approximation based on Gibbs sampling is used:

$$\mathrm{CD}_\theta(\mathbf{v}^{(0)}, \mathbf{v}^{(k)}) = -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}^{(0)})\frac{\partial E(\mathbf{v}^{(0)}, \mathbf{h})}{\partial\theta} + \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}^{(k)})\frac{\partial E(\mathbf{v}^{(k)}, \mathbf{h})}{\partial\theta}. \tag{A.97}$$

$\mathrm{CD}_\theta(\mathbf{v}^{(0)}, \mathbf{v}^{(k)})$ is a k-order approximation of the $\frac{\partial}{\partial\theta}\ln p(\mathbf{v})$. Two values of the visible layer, $\mathbf{v}^{(0)}$, the input data, and $\mathbf{v}^{(k)}$, the sample drawn using Gibbs sampling are used to approximate the derivative term given in (A.93).

In contrastive divergence algorithm the first sample $\mathbf{h}^{(0)}$ is drawn from $p(\mathbf{h}|\mathbf{v}^{(0)})$. Obtained sample $\mathbf{h}^{(0)}$ is used to generate visible vector $\mathbf{v}^{(1)}$ by taking a sample from $p(\mathbf{v}|\mathbf{h}^{(0)})$. This process continues for k steps yielding $\mathbf{v}^{(k)}$ in the end:

$$\mathbf{v}^{(0)} \Rightarrow \mathbf{h}^{(0)} \Rightarrow \mathbf{v}^{(1)} \Rightarrow \mathbf{h}^{(1)} \Rightarrow \cdots \Rightarrow \mathbf{h}^{(k-1)} \Rightarrow \mathbf{v}^{(k)}. \tag{A.98}$$

The complete algorithm of the training based on the contrastive is given in Algorithm 1. Gibbs sampling approximation of the derivatives of the model parameters is given below:

$$\begin{aligned}
\mathrm{CD}_w(\mathbf{v}^{(0)}, \mathbf{v}^{(k)}, i, j) &= \frac{1}{\sigma_i^2}\left[v_i^{(0)} \tanh\left(\sum_{l=1}^{V} \frac{v_l^{(0)}}{\sigma_l^2} w_{lj} + b_j^h\right) - v_i^{(k)} \tanh\left(\sum_{l=1}^{V} \frac{v_l^{(k)}}{\sigma_l^2} w_{lj} + b_j^h\right)\right] \\
\mathrm{CD}_{b^v}(\mathbf{v}^{(0)}, \mathbf{v}^{(k)}, i) &= \frac{1}{\sigma_i^2}\left(v_i^{(0)} - v_i^{(k)}\right) \\
\mathrm{CD}_{b^h}(\mathbf{v}^{(0)}, \mathbf{v}^{(k)}, j) &= \tanh\left(\sum_{l=1}^{V} \frac{v_l^{(0)}}{\sigma_l^2} w_{lj} + b_j^h\right) - \tanh\left(\sum_{l=1}^{V} \frac{v_l^{(k)}}{\sigma_l^2} w_{lj} + b_j^h\right).
\end{aligned} \tag{A.99}$$

Algorithm 1: Training GBPRBMs using k-step contrastive divergence.

Input : Number of visible units V; number of hidden units H; number of epochs N; a batch S of visible vectors $\mathbf{v}_s \in S$; number of mini-batches M; Gibbs sampling order k; learning rate ν; regularization constant λ; momentum μ.

Output: For all $i \in \{1, \ldots, V\},\ j \in \{1, \ldots, H\}$ GBPRBM weights w_{ij}, biases b_i^v, b_j^h.

begin

for $i \in \{1, \ldots, V\}$ and $j \in \{1, \ldots, H\}$ **do**

Initialize weights w_{ij}, biases b_i^v, b_j^h and corresponding Δ-updates:

$w_{ij} \sim \mathcal{N}(0, 0.1)$, $b_i^v = \text{mean}(S)$, $b_j^h \sim \mathcal{U}(0, 0.1)$ (for modified algorithm $\mathbf{b}_h = \mathbf{b}_h^*$),

$\Delta w_{ij} = 0$, $\Delta b_i^v = 0$, $\Delta b_j^h = 0$

for epoch $n \in \{1, \ldots, N\}$ **do**

Permute samples in batch S.

Partition batch S into M disjoint mini-batches $S_m,\ i \in \{1, \ldots, M\}$ such that

$S_p \cap S_q = \emptyset,\ \forall p, q \in \{1, \ldots, M\},\ p \neq q$, and $S_1 \cup S_2 \cup \cdots \cup S_M = S$.

for $m \in \{1, \ldots, M\}$ */* For every mini-batch S_m */* **do**

for $i \in \{1, \ldots, V\}$ and $j \in \{1, \ldots, H\}$ **do**

Clear derivatives of the log-likelihood function:

$\delta w_{ij} = 0$; $\delta b_i^v = 0$; $\delta b_j^h = 0$.

for $\mathbf{v} \in S_m$ */* For every sample in mini-batch S_m */* **do**

$\mathbf{v}^{(0)} \leftarrow \mathbf{v}$ */* Set sample as a visible vector */*

for $t \in \{0, \ldots, k-1\}$ */* k-step Gibbs sampling */* **do**

for $j \in \{1, \ldots, H\}$ **do**

Sample hidden units: $h_j^{(t)} \sim p(h_j | \mathbf{v}^{(t)})$.

for $i \in \{1, \ldots, V\}$ **do**

Sample visible units: $v_i^{(t+1)} \sim p(v_i | \mathbf{h}^{(t)})$.

for $i \in \{1, \ldots, V\}$ and $j \in \{1, \ldots, H\}$ **do**

Contrastive divergence update:

$\delta w_{ij} \leftarrow \delta w_{ij} + \text{CD}_w(\mathbf{v}^{(0)}, \mathbf{v}^{(k)}, i, j)$

$\delta b_i^v \leftarrow \delta b_i^v + \text{CD}_{b^v}(\mathbf{v}^{(0)}, \mathbf{v}^{(k)}, i)$

$\delta b_j^h \leftarrow \delta b_j^h + \text{CD}_{b^h}(\mathbf{v}^{(0)}, \mathbf{v}^{(k)}, j)$

for $i \in \{1, \ldots, V\}$ and $j \in \{1, \ldots, H\}$ **do**

Normalize by mini-batch size:

$\delta w_{ij} \leftarrow \frac{\delta w_{ij}}{|S_m|}$, $\delta b_i^v \leftarrow \frac{\delta b_i^v}{|S_m|}$, $\delta b_j^h \leftarrow \frac{\delta b_j^h}{|S_m|}$

for $i \in \{1, \ldots, V\}$ and $j \in \{1, \ldots, H\}$ **do**

Update all parameters:

$\Delta w_{ij} \leftarrow (1 - \mu)\nu \cdot \delta w_{ij} + \mu \Delta w_{ij}$

$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$

$\Delta b_i^v \leftarrow (1 - \mu)\nu \cdot \delta b_i^v + \mu \Delta b_i^v$

$b_i^v \leftarrow b_i^v + \Delta b_i^v$

$\Delta b_j^h \leftarrow (1 - \mu)\nu \cdot \delta b_j^h + \mu \Delta b_j^h$

$b_j^h \leftarrow b_j^h + \Delta b_j^h$

Bibliography

[1] A. Krizhevsky, Learning multiple layers of features from tiny images, Master's thesis, University of Toronto (2009).

[2] M. Ranzato, C. Poultney, S. Chopra, Y. L. Cun, Efficient Learning of Sparse Representations with an Energy-Based Model, MIT Press, 2007, pp. 1137–1144.

[3] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proc. of the 25th International Conference on Machine Learning, ICML 2008, ACM, New York, NY, USA, 2008, pp. 160–167.

[4] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: Advances in Neural Information Processing Systems 19 (NIPS 2006), Curran Associates, Inc., 2006, pp. 153–160.

[5] G. F. M. Cuartas, On the expressive power of discrete mixture models, Restricted Boltzmann Machines, and Deep Belief Networks - a unified mathematical treatment, Ph.D. thesis, University of Leipzig (2012).

[6] N. Le Roux, Y. Bengio, Representational power of Restricted Boltzmann Machines and Deep Belief Networks, Neural Computation 20 (6) (2008) 1631–1649. doi:10.1162/neco.2008.04-07-510.

[7] J. Martens, A. Chattopadhya, T. Pitassi, R. Zemel, On the representational efficiency of Restricted Boltzmann Machines, in: Advances in Neural Information Processing Systems 26 (NIPS 2013), Curran Associates, Inc., 2013, pp. 2877–2885.

[8] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.

[9] K. Cho, A. Ilin, T. Raiko, Improved learning of Gaussian-Bernoulli Restricted Boltzmann Machines, in: Artificial Neural Networks and Machine Learning, ICANN 2011, Springer-Verlag, 2011, pp. 10–17.

[10] J. Melchior, Learning Natural Image Statistics with Gaussian-Binary Restricted Boltzmann Machines, Master's thesis, Ruhr-Universitat Bochum (2012).

[11] N. Wang, J. Melchior, L. Wiskott, An analysis of Gaussian-Binary Restricted Boltzmann Machines for natural images, European Symposium on Artificial Neural Networks (2012) 287–292.

[12] G. Montavon, K.-R. Müller, Deep Boltzmann Machines and the Centering Trick, 2nd Edition, Vol. 7700 of Lecture Notes in Computer Science, Springer, 2012, pp. 621–637.

[13] J. Melchior, A. Fischer, L. Wiskott, How to center Deep Boltzmann Machines, Journal of Machine Learning Research 17 (99) (2016) 1–61.

[14] J. Yosinski, H. Lipson, Visually debugging Restricted Boltzmann Machine training with a 3D example, in: Representation Learning Workshop, 29th International Conference on Machine Learning, 2012.

[15] S. Dieleman, B. Schrauwen, Accelerating sparse Restricted Boltzmann Machine training using non-Gaussianity measures, in: Deep Learning and Unsupervised Feature Learning, Proc., 2012, p. 9.

[16] M. Berglund, T. Raiko, K. Cho, Measuring the usefulness of hidden units in Boltzmann Machines with mutual information, Neural Networks (2014) 12–18doi:10.1016/j.neunet.2014.09.004.

[17] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Computation 14 (8) (2002) 1771–1800. doi:10.1162/089976602760128018.

[18] T. Tieleman, Training Restricted Boltzmann Machines using approximations to the likelihood gradient, in: Proc. of the 25th International Conference on Machine Learning, ICML 2008, ACM, 2008, p. 1064–1071. doi:10.1145/1390156.1390290.

[19] G. Desjardins, A. Courville, Y. Bengio, P. Vincent, O. Delalleau, Parallel tempering for training of Restricted Boltzmann Machines, in: JMLR Workshop and Conference Proc., Vol. 9, MIT Press, 2010, p. 145–152.