ABSTRACT

Title of thesis: MUTUAL INFORMATION-BASED RBM NEURAL NETWORKS

Kang-Hao Peng, Master of Science, 2016

Thesis directed by: Professor Rama Chellappa
Department of Electrical
and Computer Engineering

(Deep) neural networks are increasingly being used for various computer vision and pattern recognition tasks due to their strong ability to learn highly discriminative features. However, quantitative analysis of their classification ability and design philosophies are still nebulous. In this work, we use information theory to analyze the concatenated restricted Boltzmann machines (RBMs) and propose a mutual information-based RBM neural networks (MI-RBM). We develop a novel pre-training algorithm to maximize the mutual information between RBMs. Extensive experimental results on various classification tasks show the effectiveness of the proposed approach.

MUTUAL INFORMATION-BASED RBM NEURAL NETWORKS

by

Kang-Hao Peng

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2016

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Joseph JaJa
Professor Prakash Narayan
Professor Richard La

Acknowledgments

I owe my gratitude to my advisor, Dr. Rama Chellappa. His vision on theoretical analysis for deep neural networks has inspired me to discover knowledge and conduct experiments. This dissertation could not have be completed without his supports and advise. In addition, I sincerely thank him for giving me the opportunity to work on challenging projects with the most intelligent and amazing people. It has been a pleasure working in this group.

Moreover, I gratefully acknowledge the assistance from Mr. Heng Zhang. His identification of the usefulness of RBM in non-vision datasets (such as touch data) is crucial in this work. I would also like to give special gratitude to Mr. Jun-Cheng Chen, for his experimental supervisions.

Finally, I sincerely acknowledge the support of NVIDIA Corporation for donating the Tesla K40 GPU used in this research.

Table of Contents

List of Figures

List of Abbreviations

FFNN	Feed Forward Neural Network
CNN	Convolutional Neural Network
RBM	Restricted Boltzmann Machine
CD	Contrastive Divergence
PCD	Persistent Contrastive Divergence

Chapter 1: Introduction

Deep neural networks (DNNs) are artificial neural networks (ANNs) with a deeper architecture, better activation function and appropriate pre-training algorithms. Different network architectures have been proposed including convolutional neural networks (CNNs) [1], the Autoencoder [19], and restricted Boltzmann machines (RBMs) [21]. For visual data that exhibit spatial correlation, the combination of CNN, rectified linear units (ReLUs) [7], max-pooling and fully connected layers has been the dominant architecture for feature extraction. For non-visual data, RBMs and autoencoder have been applied.

Conventionally, ANNs use sigmoid function as the activation function and are trained using back propagation. However, the sigmoid function suffers from the problem that weight gradients vanish when back-propagated to the input layer. Unsupervised pre-training algorithms can initialize its weights and avoid the gradient vanishing problem. Popular pre-training methods include contrastive divergence (CD) [20] that adjusts the RBM parameters according to the maximum likelihood (ML) of visible nodes and persistent contrastive divergence (PCD) [6] that essentially improves CD by an improved Gibbs' sampling procedure. Under ML pre-training, the hidden nodes are treated as latent variables to represent the probability distri-

bution of visible nodes.

Despite the success of DNNs for feature learning, there are few theoretical studies for DNN. Fundamental questions such as architecture design philosophy remain unanswered. For example, [26, 27] concentrate on DNNs' ability to universally approximate marginal distributions of visible nodes. Specifically, they put more emphasis on the representative power of DNNs and show that DNNs can approximate any distribution over binary vectors to arbitrary accuracy.

In this work, we analyze RBMs from the information theoretic point of view to obtain insights on the design philosophy of DNNs. Specifically, in one layer of neural networks, by assuming that visible and hidden nodes follow Boltzmann distribution, the mutual information between visible and hidden nodes is a function of parameters. The supremum value of mutual information is termed channel capacity, and it measures the maximum information (in terms of bits) that can be reliably transmitted through conditional distribution between input and output.

Furthermore, we propose a novel pre-training algorithm to maximize mutual information in RBM neural networks. Experiments on three image datasets (COIL20, COIL100, CIFAR10) and two touch datasets (touchalytics, Active Authentication touch) demonstrate that RBM neural networks using the proposed pre-training algorithm outperform networks using other popular pre-training algorithms based on contrastive divergence and persistent contrastive divergence. Finally, we analyze neural networks via information theory with various architectures.

Note that there are some recent works that apply mutual information to neural networks. In [5], mutual information is used to measure the usefulness of RBM

hidden nodes. They discover that classification performance of neural networks is robust to deletion of hidden nodes that have lower mutual information measures.

This thesis is organized as follows. Chapter 2 describes (persistent) contrastive divergence that is the fundamental approach for training RBM deep neural networks. Chapter 3 derives the mutual information between RBMs and its gradients. A novel training algorithms to maximize the mutual information for RBM neural networks is proposed. Chapter 4 presents experimental results on various datasets. Chapter 5 concludes the thesis with a brief summary. Finally, the derivation of mutual information is presented in Appendix A.

Chapter 2: (Persistent) Contrastive Divergence

Consider an RBM that has binary visible nodes $\mathbf{V} \in \{0,1\}^n$ and hidden nodes $\mathbf{Y} \in \{0,1\}^m$. Their joint probability mass function (pmf) follows the Boltzmann distribution that is a parametric model with parameters $\theta := (W, \mathbf{b}, \mathbf{c})$,

$$P_\theta(\mathbf{V} = \mathbf{v}, \mathbf{Y} = \mathbf{y}) = \frac{1}{Z(\theta)} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y})$$

where $W \in \mathbb{R}^{n \times m}$ is the weight matrix, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^m$ are the biases. $Z(\theta) := \sum_{\mathbf{v},\mathbf{y}} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y})$ is the partition function that normalizes the probability mass function. Note that the following Markov relationships hold,

$$V_i \to \mathbf{Y} \to V_j, Y_i \to \mathbf{V} \to Y_j, \forall i \neq j$$

Next, consider a training dataset $\mathcal{D} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N\}$, where each $\mathbf{v}_s \in \{0,1\}^n$ is a realization of $\mathbf{V}$ for index $s = 1, 2, \cdots, N$, and N is the size of the training dataset. Given $\mathcal{D}$, we define the empirical distribution as

$$P^0(\mathbf{V} = \mathbf{v}) := \frac{1}{N} \sum_{s=1}^{N} \mathbf{1}\{\mathbf{v}_s = \mathbf{v}\}$$

where $\mathbf{1}\{x = y\}$ is the indicator function. In addition, define the marginal distribution after K-step Gibbs sampling as,

$$P_\theta^K(\mathbf{V} = \mathbf{v}) := \frac{1}{N} \sum_{s=1}^{N} \mathbf{1}\{\mathbf{v}_s^K = \mathbf{v}\}$$

$\mathbf{v}_s^K$ is obtained by iterating the following Gibbs sampler K times.

$$\mathbf{V}_s^{K+1} \sim Bernoulli\left(\sigma\left(W\mathbf{y}_s^K + \mathbf{b}\right)\right)$$

$$\mathbf{Y}_s^{K+1} \sim Bernoulli\left(\sigma\left(W^T\mathbf{v}_s^K + \mathbf{c}\right)\right)$$

where $\sigma(x) := 1/(1+\exp(-x))$ is the sigmoid function, and we set $\mathbf{v}_s^0 = \mathbf{v}_s \in \mathcal{D}$ as initial condition. Theoretically, $P_\theta^K(\mathbf{v})$ converges to $P_\theta(\mathbf{v})$ for $N, K \to \infty$, or approximately $(\mathbf{V}_s^K, \mathbf{Y}_s^K) \sim P_\theta$.

Now, CD adopts the maximum likelihood criterion that adjusts θ to maximize the log-likelihood of $\mathcal{D}$, or equivalently to minimize the Kullback-Leibler (KL) divergence between the empirical distribution $P^0(\mathbf{v})$ and the final distribution $P_\theta^\infty(\mathbf{v})$.

$$D(P^0||P_\theta^\infty) := -H(P^0) - \langle \log P_\theta^\infty(\mathbf{v}) \rangle_{P^0}$$

where $\langle . \rangle_P$ denotes expectation over distribution P, therefore $\langle . \rangle_{P^0}$ is simply the average over the training dataset. Since the entropy $H(P^0)$ is fixed, minimizing the KL divergence is equivalent to maximizing $\langle \log P_\theta^\infty(\mathbf{v}) \rangle_{P^0}$. We can derive

$$\begin{aligned}\left\langle \frac{\partial \log P_\theta^\infty(\mathbf{V})}{\partial w_{ik}} \right\rangle_{P^0} &= \langle v_i y_k \rangle_{P^0} - \langle v_i y_k \rangle_{P_\theta^\infty} \\ &\approx \langle v_i y_k \rangle_{P^0} - \langle v_i y_k \rangle_{P_\theta^1}\end{aligned}$$

Although Gibbs sampling has the well known drawback that its convergence to stationary distribution takes considerable time, [20] has shown that, empirically one-step Gibbs sampling is good enough. Therefore, the weight matrix is updated according to

$$w_{ik}^{(t)} = w_{ik}^{(t-1)} + \eta\left(\langle v_i y_k \rangle_{P^0} - \langle v_i y_k \rangle_{P_{\theta^{(t-1)}}^1}\right)$$

where $\eta \geq 0$ is the learning rate and $\theta^{(t)} = (W^{(t)}, \mathbf{b}^{(t)}, \mathbf{c}^{(t)})$ is the parameters at iteration t. The updating rule for biases follow the similarity.

Note that, Gibbs sampling operates in batch mode, and PCD differs from CD by initializing the state of each Gibbs sampling by the sample outcomes from the last batch.

Chapter 3: Mutual Information for RBM

For communication channels, we can often claim independence of source distribution from channel distribution. In RBM, we need to consider the source and channel distribution jointly since θ controls both the conditional $P_\theta(\mathbf{y}|\mathbf{v})$ and marginal $P_\theta(\mathbf{v})$ distribution. In this section, we first derive the conditional and marginal distribution of RBM, then compute the mutual information and finally propose a training algorithm to maximize it in RBM neural network.

3.1 Conditional and Marginal Distribution

First note that $P_\theta(Y_k = 1|\mathbf{V} = \mathbf{v}) = \sigma\left(\sum_{i=1}^n v_i w_{ik} + c_k\right)$. We can simplify the conditional probability as

$$\begin{aligned}
&P_\theta(\mathbf{Y} = \mathbf{y}|\mathbf{V} = \mathbf{v}) \\
&= \prod_{k=1}^m \sigma\left(\sum_{i=1}^n v_i w_{ik} + c_k\right)^{y_k} \sigma\left(-\sum_{i=1}^n v_i w_{ik} - c_k\right)^{1-y_k} \\
&= \prod_{k=1}^m \sigma\left(-\sum_{i=1}^n v_i w_{ik} - c_k\right) \left(\frac{\sigma\left(\sum_{i=1}^n v_i w_{ik} + c_k\right)}{\sigma\left(-\sum_{i=1}^n v_i w_{ik} - c_k\right)}\right)^{y_k} \\
&= \prod_{k=1}^m \sigma\left(-\sum_{i=1}^n v_i w_{ik} - c_k\right) \exp\left(\sum_{i=1}^n v_i w_{ik} y_k + c_k y_k\right) \\
&= \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{c}^T \mathbf{y}) \prod_{k=1}^m \sigma\left(-\sum_{i=1}^n v_i w_{ik} - c_k\right) \\
&= P_\theta(\mathbf{v}, \mathbf{y}) \exp\left(-\mathbf{b}^T \mathbf{v}\right) Z(\theta) \prod_{k=1}^m \sigma\left(-\sum_{i=1}^n v_i w_{ik} - c_k\right)
\end{aligned} \tag{3.1}$$

By Bayes' theorem, we obtain the following marginal,

$$\begin{aligned}
P_\theta(\mathbf{V} = \mathbf{v}) &= \frac{P_\theta(\mathbf{y}, \mathbf{v})}{P_\theta(\mathbf{y}|\mathbf{v})} \\
&= \exp\left(\mathbf{b}^T \mathbf{v}\right) \left(Z(\theta) \prod_{k=1}^m \sigma\left(-\sum_{i=1}^n v_i w_{ik} - c_k\right)\right)^{-1}
\end{aligned} \tag{3.2}$$

Similarly, we can also obtain the marginal of hidden nodes,

$$\begin{aligned}
P_\theta(\mathbf{Y} = \mathbf{y}) &= \frac{P_\theta(\mathbf{y}, \mathbf{v})}{P_\theta(\mathbf{v}|\mathbf{y})} \\
&= \exp(\mathbf{c}^T \mathbf{y}) \left(Z(\theta) \prod_{i=1}^n \sigma\left(-\sum_{k=1}^m w_{ik} y_k - b_i\right)\right)^{-1}
\end{aligned} \tag{3.3}$$

3.2 Mutual Information

By the definition of the mutual information, we have

$$\begin{aligned} I(\mathbf{V};\mathbf{Y}) &:= \sum_{\mathbf{v},\mathbf{y}} P_\theta(\mathbf{v},\mathbf{y}) \left[\log\left(\frac{P_\theta(\mathbf{v},\mathbf{y})}{P_\theta(\mathbf{v})P_\theta(\mathbf{y})}\right)\right] \\ &= \log Z(\theta) + E\left[\mathbf{V}^T W \mathbf{Y}\right] \\ &+ \sum_{l=1}^{m} E\left[\log\sigma\left(-\left(\sum_{j=1}^{n} V_j w_{jl} + c_l\right)\right)\right] \\ &+ \sum_{j=1}^{n} E\left[\log\sigma\left(-\left(\sum_{l=1}^{m} w_{jl} Y_l + b_j\right)\right)\right] \end{aligned} \tag{3.4}$$

In this thesis, our objective is to adjust the parameters $\theta = (W, \mathbf{b}, \mathbf{c})$ to maximize the mutual information. First, we calculate the gradient of mutual information, and then adjust the weights using the stochastic gradient ascent.

We calculate the gradient in (3.4) one term at a time, then combine them to have the gradient of mutual information. Without loss of generality, in the following we will only show the gradient with respect to weight w_{ik}.

- Gradient of $\log Z(\theta)$ with respect to w_{ik}

$$\frac{\partial}{\partial w_{ik}} \log Z(\theta) = E[V_i Y_k]$$

Furthermore, the following equalities hold:

$$\begin{aligned} E[V_i Y_k] &= E\left[V_i \sigma\left(\sum_{j=1}^{n} V_j w_{jk} + c_k\right)\right] \\ &= E\left[\sigma\left(\sum_{l=1}^{m} w_{il} Y_l + b_i\right) Y_k\right] \end{aligned}$$

- Gradient of $E\left[\mathbf{V}^T W \mathbf{Y}\right]$ with respect to w_{ik}

$$\frac{\partial}{\partial w_{ik}} E\left[\mathbf{V}^T W \mathbf{Y}\right] = E\left[\mathbf{V}^T W \mathbf{Y}(V_i Y_k)\right] - E\left[\mathbf{V}^T W \mathbf{Y}\right] E\left[V_i Y_k\right] + E\left[V_i Y_k\right]$$

- Gradient of $E\left[\log\sigma\left(-\left(\sum_{j=1}^{n} V_j w_{jl} + c_l\right)\right)\right]$ with respect to w_{ik}

$$\begin{aligned}
&\frac{\partial}{\partial w_{ik}} \sum_{l=1}^{m} E\left[\log\sigma\left(-\left(\sum_{j=1}^{n} V_j w_{jl} + c_l\right)\right)\right] \\
&= -E\left[V_i \sigma\left(\sum_{j=1}^{n} V_j w_{jk} + c_k\right)\right] \\
&+ \sum_{l=1}^{m} E\left[\log\sigma\left(-\sum_{j=1}^{n} V_j w_{jl} + c_l\right)\left(V_i Y_k - E[V_i Y_k]\right)\right]
\end{aligned}$$

- Gradient of $E\Big[\log\sigma\Big(-\Big(\sum_{l=1}^{m} w_{jl} Y_l + b_j\Big)\Big)\Big]$ with respect to w_{ik}

$$\begin{aligned}
&\frac{\partial}{\partial w_{ik}} \sum_{j=1}^{n} E\left[\log\sigma\left(-\left(\sum_{l=1}^{m} w_{jl} Y_l + b_j\right)\right)\right] \\
&= -E\left[\sigma\left(\sum_{l=1}^{m} w_{il} Y_l + b_i\right) Y_k\right] \\
&+ \sum_{j=1}^{n} E\left[\log\sigma\left(-\sum_{l=1}^{m} w_{jl} Y_l + b_j\right)\left(V_i Y_k - E[V_i Y_k]\right)\right]
\end{aligned}$$

Combining the above equations, the gradient of mutual information can be calculated as

$$\begin{aligned}
\frac{\partial I}{\partial w_{ik}} &= E\left[\Big(V_i Y_k - E[V_i Y_k]\Big)\mathbf{V}^T W \mathbf{Y}\right] \\
&+ E\left[\Big(V_i Y_k - E[V_i Y_k]\Big)\sum_{l=1}^{m} \log\sigma\left(-\sum_{j=1}^{n} V_j w_{jl} - c_l\right)\right] \\
&+ E\left[\Big(V_i Y_k - E[V_i Y_k]\Big)\sum_{j=1}^{n} \log\sigma\left(-\sum_{l=1}^{m} w_{jl} Y_l - b_j\right)\right]
\end{aligned} \tag{3.5}$$

$$\frac{\partial I}{\partial b_i} = E\big[(V_i - E[V_i])\mathbf{V}^T W\mathbf{Y}\big] + E\left[\Big(V_i - E[V_i]\Big)\sum_{j=1}^{n}\log\sigma\left(-\sum_{l=1}^{m} w_{jl}Y_l - b_j\right)\right] \tag{3.6}$$

$$\frac{\partial I}{\partial c_k} = E\big[(Y_k - E[Y_k])\mathbf{V}^T W\mathbf{Y}\big] + E\left[\Big(Y_k - E[Y_k]\Big)\sum_{l=1}^{m}\log\sigma\left(-\sum_{j=1}^{n} V_j w_{jl} - c_l\right)\right] \tag{3.7}$$

As $W, \mathbf{b}, \mathbf{c}$ belong to Euclidean space (an open set with non-empty interior), the necessary condition for supremum mutual information is that the gradients in (3.5-3.7) equal to zero. However, the close form solution is not tractable. Therefore, we apply stochastic gradient ascent. By first initializing $\theta^{(0)}$ randomly and iteratively adjusting the parameters $\theta^{(t)}$ according to its gradients, we maximize the mutual information between visible and hidden nodes.

3.3 Approximate Expectation Using Gibbs Sampling

We see that the exact calculation of expectations in (3.4-3.7) requires summing over exponentially many elements, which is intractable. Therefore, we use Gibbs sampling to approximate the expectations. Using (3.1) we can sample the random variables $(\mathbf{V}, \mathbf{Y}) \sim P_\theta$. Now, for any function $g(\mathbf{V}, \mathbf{Y})$ we can approximate its expectation by the following

$$E\left[g(\mathbf{V}, \mathbf{Y})\right] \approx \left\langle g(\mathbf{v}_s^{(K)}, \mathbf{y}_s^{(K)})\right\rangle_{P^0} \tag{3.8}$$

Nevertheless, we see that (3.5-3.7) contain an expectation inside an expectation. This may result in large variation of the final approximation. Therefore we need

large batch and high K value so that the gradients in (3.5-3.7) can be approximated reasonably well. Alternatively, we can use tools that can symbolically calculate the gradient of (3.4) with respect to parameters. GPU-based languages such as theano can do this task easily. However, in this case we have to approximate the partition function. In our experiments, we use annealed importance sampling.

3.4 Annealed Importance Sampling

Annealed Importance Sampling (AIS) is an algorithm that enables fast approximation of partition function $Z(\theta)$. Here we briefly introduce the M-stage AIS. More details can be found in [16]. First, we assume outcome $(\mathbf{V}, \mathbf{Y}) \sim P_{\frac{r\theta}{M}}$ can be sampled for $r = 0, 1, \cdots, M-1$. This can be trivially achieved by (3.1) by scaling down parameters from θ to $\frac{r\theta}{M}$. Note that

$$\frac{Z(\theta)}{Z(0)} = \frac{Z\left(\frac{1}{M}\theta\right)}{Z(0)} \frac{Z\left(\frac{2}{M}\theta\right)}{Z\left(\frac{1}{M}\theta\right)} \cdots \frac{Z(\theta)}{Z\left(\frac{M-1}{M}\theta\right)}$$

Provided M is large enough, we can approximate the ratio

$$\frac{Z(\frac{r+1}{M}\theta)}{Z(\frac{r}{M}\theta)} \approx \frac{1}{N} \sum_{s=1}^{N} \frac{P^*_{r+1}(\mathbf{v}_s^{(r)}, \mathbf{y}_s^{(r)})}{P^*_r(\mathbf{v}_s^{(r)}, \mathbf{y}_s^{(r)})} \tag{3.9}$$

where $(\mathbf{V}_s^{(r)}, \mathbf{Y}_s^{(r)}) \sim P_{\frac{r\theta}{M}}$, and P^*_r is the unnormalized pmf,

$$P^*_r(\mathbf{v}, \mathbf{y}) = \exp\left(\frac{r}{M}\left(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}\right)\right)$$

Finally, we can trivially calculate $Z(0) = 2^{n+m}$.

3.5 Training Mutual Information-based RBM Neural Networks

As shown in figure 3.1, typically an RBM neural network is greedily pre-trained layer by layer using unsupervised methods, then is fine-tuned by minimizing the logistic regression error through back-propagation. Note that, a deep network can have many hidden layers.

Different from previous works, we pre-train the neural networks by greedily maximizing the mutual information layer-wisely. Note that, given a training dataset, the source (visible node) entropy is fixed. Therefore maximizing the mutual information is equivalent to minimizing the conditional entropy $H(\mathbf{V}|\mathbf{Y})$. This increases the dependency of $\mathbf{V}$ and $\mathbf{Y}$. Since we usually use the outcome of the hidden nodes for further classification tasks, maximizing the mutual information helps to improve the classification performance for neural networks. We name the resulting neural network as MI-RBM. Our pre-training algorithm is summarized in **Algorithm 1**.

Algorithm 1 Pre-training *MI-RBM*

Require: Training data $\mathcal{D}$

Initialization: Randomly initialize $\theta^{(0)}$

Main loop: Update $(\theta^{(t)}, \theta^{(t+1)})$ for $t = 1, \cdots, T$

1: **for** $r = 0$ to M **do**

2: Sample $\left(\mathbf{v}_s^{(r)}, \mathbf{y}_s^{(r)}\right) \sim P_{\frac{r}{M}\theta^{(t)}}$ by iterating (3.1) K times for all indices s in $\mathcal{D}$

3: **end for**

4: **for** $r = 0$ to $M - 1$ **do**

5: Construct $ratio(r)$ as (3.9) using $\left(\mathbf{v}_s^{(r)}, \mathbf{y}_s^{(r)}\right)$

6: **end for**

7: $\log Z(\theta^{(t)}) = (n + m) \log 2 + \sum_{r=0}^{M-1} \log\left(ratio(r)\right)$

8: Construct $I(\mathbf{V}; \mathbf{Y})$ as (3.4), where expectations in (3.4) are approximated by (3.8) using $\left(\mathbf{v}_s^{(M)}, \mathbf{y}_s^{(M)}\right)$, and log-partition function is approximated by $\log Z(\theta^{(t)})$

9: Construct gradient $\nabla_\theta I$ as in (3.5-3.7) or by symbolic programming language using constructed $I(\mathbf{V}; \mathbf{Y})$.

10: Update $\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta I$.

Ensure: $\theta^{(T)}$

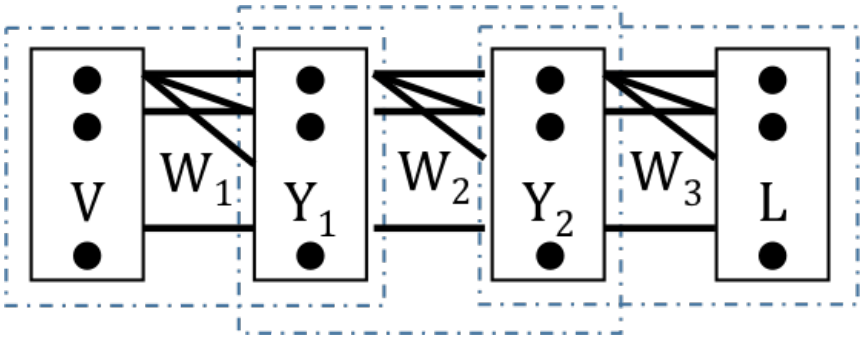

Figure 3.1: RBM neural networks with two hidden layers. W_i are weights of the neural networks and logistic regression layer. V is the input layer, Y_i is the i^{th} hidden layer, and L is the output label layer. W_1, W_2 are first pre-trained, then W_1, W_2, W_3 are fine-tuned. Biases are not shown in this figure.

Chapter 4: Experiments

4.1 Experiments

We report the resulting experiments on three image datasets: the Columbia University Image Library (COIL-20, COIL-100) and the Canadian Institute for Advanced Research (CIFAR-10) datasets; and on two screen touch datasets (touchalytics, Active Authentication touch). We compare the proposed MI-RBM with RBM deep neural networks pre-trained using CD and PCD. We name them CD-RBM and PCD-RBM for simplicity.

In the following experiments we use theano [29] to calculate the mutual information, its gradient and approximate the partition function using annealed importance sampling. We also use momentum [25] to train RBMs and dropout [17] to fine-tune the final neural networks.

4.1.1 COIL-20 and COIL-100

COIL-20 and COIL-100 [14] consist of images of 20 and 100 objects repectively. Each object is represented by 72 images taken sequentially. Each image has 5° rotation apart from each other, and has 32×32 pixels. For each object, we select

12 images as test data, 12 images as validation data and the remaining 48 images as training data. We train a neural network for COIL-20 with hidden layer sizes [100 100 100] and for COIL-100 with [300 300 300]. Pre-training with CD, PCD and MI-RBM over 300 epochs, and fine-tune using logistic regression over 2,000 epochs. From Tables 4.1 and 4.2, we see that MI-RBM outperforms the others.

Table 4.1: Probability of Error - COIL-20

	COIL-20		
	CD-RBM	PCD-RBM	MI-RBM
Original	1.67%	2.50%	1.25%
Dropout	1.25%	0.83%	0.83%

Table 4.2: Probability of Error - COIL-100

	COIL-100		
	CD-RBM	PCD-RBM	MI-RBM
Original	4.08%	3.25%	2.58%
Dropout	7.42%	4.42%	2.58%

Table 4.3 shows the mutual information before and after back propagation. We observe the following:

- Large network size corresponds to large mutual information value after pre-training. For instance, large mutual information value at the first layer is due to the input size (32×32) being larger than hidden layers sizes.

- Back propagation and dropout reduce the mutual information value.

Table 4.3: Mutual Information (measured in nat)

COIL-20, architecture = [100 100 100]				
	1^{st}	2^{nd}	3^{rd}	P_e
Pre-trained	394.56	70.21	70.21	-
Finetuned	394.98	69.69	69.76	1.25%
COIL-100, architecture = [300 300 300]				
Pre-trained	799.31	362.23	362.23	-
Finetuned	799.51	360.98	361.43	2.83%
Dropout rate 0.1	760.98	320.24	264.61	1.66%
Dropout rate 0.2	711.85	267.46	307.92	3.00%

4.1.2 CIFAR-10

CIFAR-10 [15] consists of tiny images (32×32 pixels) of 10 objects (airplane, automobile, etc.) Each object has 40,000 training, 10,000 validation and 10,000 test images. We train gray-scaled CIFAR-10 on two networks with hidden layer sizes [$1k$ $1k$ $1k$] and [$1k$ $1k$ $1k$ $1k$], with batchsize 400 and momentum. From Table 4.4, it is seen that MI-RBM is better than CD-RBM under the same architecture and training epochs. In addition, the entropy for gray scaled CIFAR-10 is 574.55 *nat* and

the entropy of labels is 2.30 *nat*. After finetuning, the minimum mutual information among all layers is 1296.71 *nat*.

Table 4.4: Probability of Error - CIFAR-10 (k is 1,000)

Hidden layer size	CC-RBM	CD-RBM
$[1k\ 1k\ 1k]$	44.41%	48.71%
$[1k\ 1k\ 1k\ 1k]$	44.85%	50.24%

Finally, for image datasets such as CIFAR-10, convolutional features can be more discriminative, and we look forward to extending our approach to CNN in future works.

4.1.3 Touch Data

There has been a growing interest in applying screen touch data to authenticate users on smartphones [3] and [2]. Every swipe is a sequence of touch events recorded when the finger is in touch with the screen of the smartphone. Each swipe $\mathbf{s}$ is encoded as a sequence of vectors

$$\mathbf{s}_i = (x_i, y_i, t_i, A_i, o_i^{ph}),$$

$i \in \{1, \cdots, N_c\}$ where x_i, y_i are the location points, t_i is the time stamp, A_i is the area occluded by the finger and o_i^{ph} is the orientation of the phone (e.g. landscape or portrait). Since the number of touch events in every swipe is different, hand-crafted feature vector of low dimension [3] is first extracted and followed by some traditional classifiers like SVM and dictionary learning. Performance can be significantly

boosted by incorporating kernels to the classifier.

The problem with kernelized classifiers is that it does not scale well to large dataset and computation time is often prohibitive. These problems make the kernelized classifiers not attractive especially on mobile platforms. We apply RBM neural networks to the raw features (hand-crafted feature) and try to learn more discriminative representation which is the output of the last hidden layer.

We use two publicly available touch datasets: Touchalytics dataset [3] consisting of 41 users' touch data collected using Android smartphones and Active Authentication (AA) touch dataset [2] consisting of 50 users' touch data collected over 3 sessions using iPhone 5s. For each of these datasets, we randomly split the dataset with ratio $6:2:2$ for training, cross-validation and testing respectively. We repeat the random partition for 5 times. We feed the original touch features to the MI-RBM as well as CD-RBM and PCD-RBM and use the output of the last hidden layer of the networks as the new features. The hidden layers of the networks are set to be $[80,\ 60]$. We apply a linear SVM classifier to the original features as well as learned representations, tune the parameters using the cross validation set and report classification error averaged over five trials in Table 4.5.

Table 4.5: Averaged Classification Errors on Touch Data.

Datasets	Raw features	CD-RBM	PCD-RBM	MI-RBM
Touchalytics	59.65 %	31.33 %	30.14 %	28.88 %
AA touch	81.85 %	63.78 %	59.31 %	55.13 %

From Table 4.5 we observe that all the RBM neural networks learn more discriminative representations and show significant improvement in terms of classification performance compared with the raw touch features. MI-RBM performs the best.

Furthermore, we study the influence of mutual information on the performance of the neural networks. According to information theory, if data with entropy H is to be transmitted reliably through a channel, then the channel capacity C should be greater than H.

For the AA touch dataset, we calculate the upper bound of the entropy of data $H_D \approx 8.9022$ *nats* by the following approximation

$$H_D = \frac{1}{N} \sum_{i=1}^{N} H(\mathbf{p}_s)$$

$$H(\mathbf{p}_s) = \sum_i -p_{si} \log(p_{si}) - (1 - p_{si}) \log(1 - p_{si})$$

where p_{si} is the i^{th} element of $\mathbf{p}_s$ and it denotes the probability of the visible node v_{si} equals 1. Usually $\mathbf{p}_s$ is the input feature after normalization.

From Figure 4.1, we observe that the mutual information of each layer after pre-training is affine with respect to network size, and it always drops after fine-tuning. Therefore, to reliably transmit the data over the network, the network size k needs to be greater than 10 after pre-training, and k needs to be greater than 25 after fine-tuning.

We also compare the error rate of MI-RBM, CD-RBM and PCD-RBM under the same network architecture shown in Figure 4.2. This show that mutual information serves as a design guide for the DNN architectures.

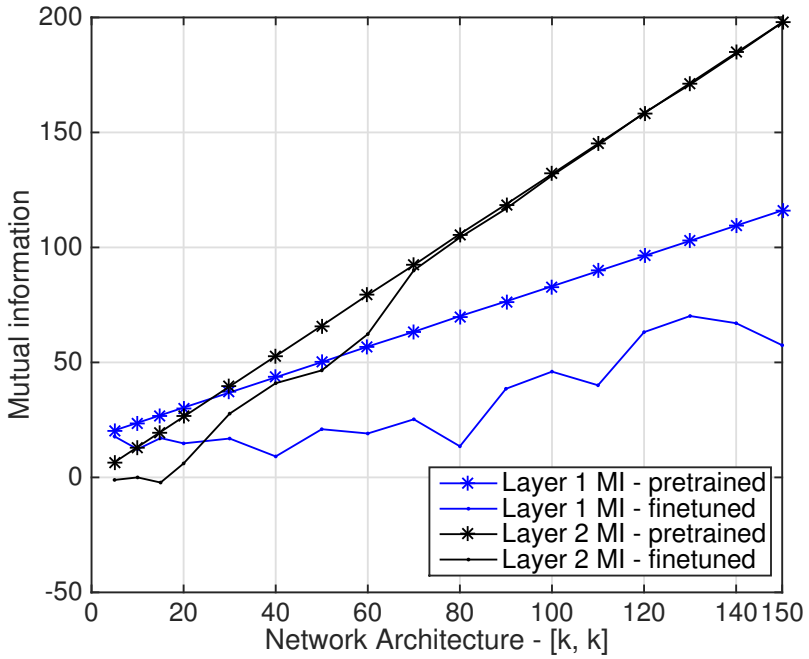

Figure 4.1: Mutual information versus network architecture with hidden layer sizes $[k,\ k]$ on AA touch dataset.

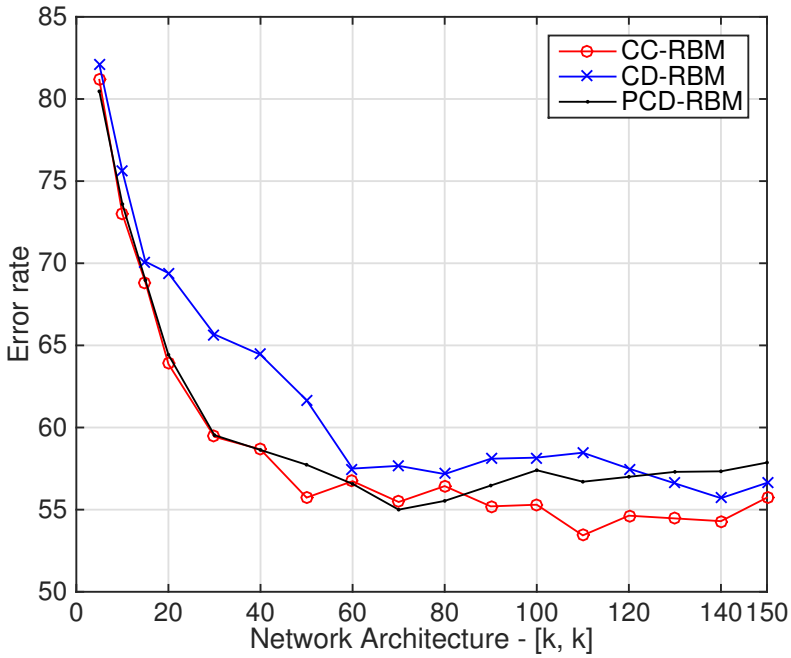

Figure 4.2: Error rate versus network architecture on AA touch dataset.

Chapter 5: Conclusion

In this work we analyze the RBM neural networks via information theoretic point of view by deriving the expression of mutual information and its gradient with respect to parameters. We propose a RBM pre-training method based on maximizing mutual information. Experiments on various datasets show that the proposed neural networks outperforms other RBMs neural networks. The future work will explore the application of mutual information to convolutional RBM.

Appendix A: Detailed Derivation of RBM Mutual Information

Consider an RBM that has $\mathbf{V} \in \{0,1\}^n$ visible nodes and $\mathbf{Y} \in \{0,1\}^m$ hidden nodes. The joint distribution of them is given by

$$P(\mathbf{V} = \mathbf{v}, \mathbf{Y} = \mathbf{y}) = \frac{1}{Z(W, \mathbf{b}, \mathbf{c})} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) \tag{A.1}$$

where $W \in \mathbb{R}^{n \times m}$ is the weight matrix, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^m$ are biases, and $Z(W, \mathbf{b}, \mathbf{c})$ is the partition function that normalizes the pmf.

$$Z(W, \mathbf{b}, \mathbf{c}) = \sum_{\mathbf{v} \in \{0,1\}^n} \sum_{\mathbf{y} \in \{0,1\}^m} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y})$$

A.1 Markov Chains

The following Markov chain holds for any $i \neq j$.

$$V_i \to \mathbf{Y} \to V_j$$

$$Y_i \to \mathbf{V} \to Y_j$$

A.2 Conditional Probability

It is easy to verify that

$$\begin{aligned} P(Y_l = 1|\mathbf{V} = \mathbf{v}, \mathbf{Y}_{\setminus l} = \mathbf{y}_{\setminus l}) &= P(Y_l = 1|\mathbf{V} = \mathbf{v}) = \sigma\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right) \\ P(V_j = 1|\mathbf{Y} = \mathbf{y}, \mathbf{V}_{\setminus j} = \mathbf{v}_{\setminus j}) &= P(V_j = 1|\mathbf{Y} = \mathbf{y}) = \sigma\left(\sum_{l=1}^{m} W_{jl} y_l + b_j\right) \end{aligned} \tag{A.2}$$

where the function $\sigma : \mathbb{R} \to (0, 1)$ is the sigmoid function,

$$\sigma(x) = \frac{1}{1 + e^{-x}}, x \in \mathbb{R}$$

Sigmoid function has the properties that

- $1 - \sigma(x) = \sigma(-x)$
- $\sigma(-x) = e^{-x}\sigma(x)$.
- $\log \frac{\sigma(x)}{\sigma(-x)} = x$ for natural logarithm $\log(.)$

A.3 Marginal Distribution

Because $\mathbf{Y}$ are conditionally independent of each other given $\mathbf{V} = \mathbf{v}$. The conditional probability can be further simplified by the following

$$\begin{aligned}
P(\mathbf{Y} = \mathbf{y}|\mathbf{V} = \mathbf{v}) &= \prod_{k=1}^{m} \sigma\left(\sum_{i=1}^{n} v_i W_{ik} + c_k\right)^{y_k} \sigma\left(-\left(\sum_{i=1}^{n} v_i W_{ik} + c_k\right)\right)^{1-y_k} \\
&= \prod_{k=1}^{m} \sigma\left(-\left(\sum_{i=1}^{n} v_i W_{ik} + c_k\right)\right) \left(\frac{\sigma\left(\sum_{i=1}^{n} v_i W_{ik} + c_k\right)}{\sigma\left(-\left(\sum_{i=1}^{n} v_i W_{ik} + c_k\right)\right)}\right)^{y_k} \\
&= \prod_{k=1}^{m} \sigma\left(-\left(\sum_{i=1}^{n} v_i W_{ik} + c_k\right)\right) \exp\left(\sum_{i=1}^{n} v_i W_{ik} y_k + c_k y_k\right) \\
&= \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{c}^T \mathbf{y}) \prod_{k=1}^{m} \sigma\left(-\left(\sum_{i=1}^{n} v_i W_{ik} + c_k\right)\right) \\
&= P(\mathbf{Y} = \mathbf{y}, \mathbf{V} = \mathbf{v}) Z(W, \mathbf{b}, \mathbf{c}) \exp\left(-\mathbf{b}^T \mathbf{v}\right) \prod_{k=1}^{m} \sigma\left(-\left(\sum_{i=1}^{n} v_i W_{ik} + c_k\right)\right)
\end{aligned}$$

Therefore, by Bayes' theorem,

$$\begin{aligned}
P(\mathbf{V} = \mathbf{v}) &= \frac{P(\mathbf{Y} = \mathbf{y}, \mathbf{V} = \mathbf{v})}{P(\mathbf{Y} = \mathbf{y}|\mathbf{V} = \mathbf{v})} \\
&= \left(Z(W, \mathbf{b}, \mathbf{c}) \prod_{l=1}^{m} \sigma\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right)\right)^{-1} \exp\left(\mathbf{b}^T \mathbf{v}\right)
\end{aligned} \tag{A.3}$$

Similarly,

$$\begin{aligned}
P(\mathbf{Y} = \mathbf{y}) &= \frac{P(\mathbf{Y} = \mathbf{y}, \mathbf{V} = \mathbf{v})}{P(\mathbf{V} = \mathbf{v}|\mathbf{Y} = \mathbf{y})} \\
&= \left(Z(W, \mathbf{b}, \mathbf{c}) \prod_{j=1}^{n} \sigma\left(-\left(\sum_{l=1}^{m} W_{jl} y_l + b_j\right)\right)\right)^{-1} \exp\left(\mathbf{c}^T \mathbf{y}\right)
\end{aligned} \tag{A.4}$$

Therefore, since $\sum_{\mathbf{v} \in \{0,1\}^n} P(\mathbf{V} = \mathbf{v}) = \sum_{\mathbf{y} \in \{0,1\}^m} P(\mathbf{Y} = \mathbf{y}) = 1$, we have

$$Z(W, \mathbf{b}, \mathbf{c}) = \sum_{\mathbf{v} \in \{0,1\}^n} \left(\prod_{l=1}^{m} \sigma\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right)\right)^{-1} \exp\left(\mathbf{b}^T \mathbf{v}\right)$$

$$Z(W, \mathbf{b}, \mathbf{c}) = \sum_{\mathbf{y} \in \{0,1\}^m} \left(\prod_{j=1}^{n} \sigma\left(-\left(\sum_{l=1}^{m} W_{jl} y_l + b_j\right)\right)\right)^{-1} \exp\left(\mathbf{c}^T \mathbf{y}\right)$$

We can also obtain the following essential formulae

$$\exp\left(\mathbf{b}^T\mathbf{v}\right)\prod_{l=1}^{m}\sigma^{-1}\left(-\left(\sum_{j=1}^{n}v_jW_{jl}+c_l\right)\right)=\sum_{\mathbf{y}\in\{0,1\}^m}\exp\left(\mathbf{v}^TW\mathbf{y}+\mathbf{b}^T\mathbf{v}+\mathbf{c}^T\mathbf{y}\right) \tag{A.5}$$

$$\exp\left(\mathbf{c}^T\mathbf{y}\right)\prod_{j=1}^{n}\sigma^{-1}\left(-\left(\sum_{l=1}^{m}W_{jl}y_l+b_j\right)\right)=\sum_{\mathbf{v}\in\{0,1\}^n}\exp\left(\mathbf{v}^TW\mathbf{y}+\mathbf{b}^T\mathbf{v}+\mathbf{c}^T\mathbf{y}\right) \tag{A.6}$$

A.4 Mutual Information

We are interested in the mutual information of RBM for the following reasons:

- Boltzmann machines (as well as RBMs) are generative models, *i.e.*, there is a true $P_V(\mathbf{v})$ that we want to learn. In other words, we simply use latent variables $\mathbf{Y}\in\{0,1\}^m$ and weights W to approximate $P_V(\mathbf{v})$ by $\sum_{\mathbf{y}\in\{0,1\}^m}P(\mathbf{v},\mathbf{y})$.
- Practically we treat the latent variables $\mathbf{Y}$ as the feature of visible nodes $\mathbf{V}$, and we further feed $\mathbf{Y}$ into another classifier (*e.g.* SVM, NN, etc.)
- Furthermore, although convolution neural network (CNN) is not a Boltzmann machine, practically the CNN is also treated as a feature extractor, and generally performs better than other feature extractors (e.g. SIFT) in computer vision, the capacity of RBM serves as the fundamental knowledge of why deep learning network works so well.

The objective of our problem can now be defined as follows

$$\sup_{W} I(\mathbf{V};\mathbf{Y}) \tag{A.7}$$

A.4.1 From KL Divergence

The mutual information is defined as $I(\mathbf{V};Y) = D(P(\mathbf{V};\mathbf{Y})||P(\mathbf{V})P(\mathbf{Y}))$, where the marginal distributions are given by (A.3) and (A.4). The above KL divergence is written as

$$D(P(\mathbf{V},\mathbf{Y})||P(\mathbf{V})P(\mathbf{Y})) = \sum_{\mathbf{v},\mathbf{y}} P(\mathbf{v},\mathbf{y}) \left[-\log P(\mathbf{v}) - \log P(\mathbf{y}) + \log P(\mathbf{v},\mathbf{y})\right]$$

where

$$-\log P(\mathbf{V}=\mathbf{v}) = \log Z(W,\mathbf{b},\mathbf{c}) - \mathbf{b}^T\mathbf{v} + \sum_{l=1}^{m} \log\sigma\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right)$$

$$-\log P(\mathbf{Y}=\mathbf{y}) = \log Z(W,\mathbf{b},\mathbf{c}) - \mathbf{c}^T\mathbf{y} + \sum_{j=1}^{n} \log\sigma\left(-\left(\sum_{l=1}^{m} W_{jl} y_l + b_j\right)\right)$$

$$\log P(\mathbf{V}=\mathbf{v},\mathbf{Y}=\mathbf{y}) = -\log Z(W,\mathbf{b},\mathbf{c}) + \mathbf{v}^T W\mathbf{y} + \mathbf{b}^T\mathbf{v} + \mathbf{c}^T\mathbf{y}$$

Therefore,

$$\begin{aligned}
&-\log P(\mathbf{v}) - \log P(\mathbf{y}) + \log P(\mathbf{v},\mathbf{y}) \\
&\quad = \log Z(W,\mathbf{b},\mathbf{c}) + \mathbf{v}^T W\mathbf{y} \\
&\qquad + \sum_{l=1}^{m} \log\sigma\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right) + \sum_{j=1}^{n} \log\sigma\left(-\left(\sum_{l=1}^{m} W_{jl} y_l + b_j\right)\right)
\end{aligned}$$

Finally we have the mutual information,

$$\begin{aligned}
&I(\mathbf{V};\mathbf{Y}) = D(P(\mathbf{V},\mathbf{Y})||P(\mathbf{V})P(\mathbf{Y})) \\
&\quad = \sum_{\mathbf{v},\mathbf{y}} P(\mathbf{v},\mathbf{y}) \left[-\log P(\mathbf{v}) - \log P(\mathbf{y}) + \log P(\mathbf{v},\mathbf{y})\right] \\
&\quad = \log Z(W,\mathbf{b},\mathbf{c}) + E\left[\mathbf{V}^T W\mathbf{Y}\right] \\
&\qquad + \sum_{l=1}^{m} E\left[\log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\right] + \sum_{j=1}^{n} E\left[\log\sigma\left(-\left(\sum_{l=1}^{m} W_{jl} Y_l + b_j\right)\right)\right]
\end{aligned} \tag{A.8}$$

A.4.2 Mutual Information Gradient w.r.t. Weights

Since $W \in \mathbb{R}^{n \times m}$ is an open set, and $\log(.)$, $\sigma(.)$ functions are continuously differentiable, necessarily the maximizing weights satisfy

$$\frac{\partial}{\partial W_{ik}} I(\mathbf{V}; \mathbf{Y}) = 0, \forall i \in \{1, \cdots, n\}, \forall k \in \{1, \cdots, m\}$$

First we prepare some derivatives.

- Partition function

$$\begin{aligned}
\frac{\partial}{\partial W_{ik}} Z(W, \mathbf{b}, \mathbf{c}) &= \sum_{(\mathbf{v},\mathbf{y}) \in \{0,1\}^{n+m}} \frac{\partial}{\partial W_{ik}} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) \\
&= \sum_{(\mathbf{v},\mathbf{y}) \in \{0,1\}^{n+m}} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) v_i y_k
\end{aligned}$$

  Therefore,

$$\frac{\partial}{\partial W_{ik}} \log Z(W, \mathbf{b}, \mathbf{c}) = \frac{1}{Z(W, \mathbf{b}, \mathbf{c})} \frac{\partial}{\partial W_{ik}} Z(W, \mathbf{b}, \mathbf{c}) = E\left[V_i Y_k\right]$$

  For biases

$$\begin{aligned}
\frac{\partial}{\partial b_i} Z(W, \mathbf{b}, \mathbf{c}) &= \sum_{(\mathbf{v},\mathbf{y}) \in \{0,1\}^{n+m}} \frac{\partial}{\partial b_i} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) \\
&= \sum_{(\mathbf{v},\mathbf{y}) \in \{0,1\}^{n+m}} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) v_i
\end{aligned}$$

  Therefore $\frac{\partial}{\partial b_j} \log Z(W, \mathbf{b}, \mathbf{c}) = E[V_i]$ and similarly, $\frac{\partial}{\partial c_k} \log Z(W, \mathbf{b}, \mathbf{c}) = E[Y_k]$.

  Now, from the marginal distribution point of view, we have

$$\frac{\partial}{\partial W_{ik}} Z(W, \mathbf{b}, \mathbf{c}) = \frac{\partial}{\partial W_{ik}} \sum_{\mathbf{v} \in \{0,1\}^n} \prod_{l=1}^{m} \sigma^{-1}\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right) \exp\left(\mathbf{b}^T \mathbf{v}\right)$$

$$= \sum_{\mathbf{v} \in \{0,1\}^n} \exp\left(\mathbf{b}^T \mathbf{v}\right) \left[\prod_{l \neq k} \sigma^{-1}\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right)\right] \frac{\partial}{\partial W_{ik}} \sigma^{-1}\left(-\left(\sum_{j=1}^{n} v_j W_{jk} + c_k\right)\right)$$

$$= \sum_{\mathbf{v} \in \{0,1\}^n} \exp\left(\mathbf{b}^T \mathbf{v}\right) \left[\prod_{l \neq k} \sigma^{-1}\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right)\right] \frac{\sigma\left(\sum_{j=1}^{n} v_j W_{jk} + c_k\right)}{\sigma\left(-\left(\sum_{j=1}^{n} v_j W_{jk} + c_k\right)\right)} v_i$$

$$= \sum_{\mathbf{v} \in \{0,1\}^n} \exp\left(\mathbf{b}^T \mathbf{v}\right) \left[\prod_{l=1}^{m} \sigma^{-1}\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right)\right] \sigma\left(\sum_{j=1}^{n} v_j W_{jk} + c_k\right) v_i$$

where, for any function $f : \mathbb{R} \to \mathbb{R}$, we have

$$\begin{aligned}
\frac{\partial}{\partial x} \sigma\left(f(x)\right) &= \frac{\partial}{\partial x} \left(1 + \exp\left(-f(x)\right)\right)^{-1} \\
&= -\left(1 + \exp\left(-f(x)\right)\right)^{-2} \exp\left(-f(x)\right) \left(-\frac{\partial}{\partial x} f(x)\right) \\
&= \sigma(f(x)) \sigma(-f(x)) \left(\frac{\partial}{\partial x} f(x)\right)
\end{aligned}$$

Therefore,

$$\frac{\partial}{\partial W_{ik}} \log Z(W, \mathbf{b}, \mathbf{c}) = \frac{1}{Z(W, \mathbf{b}, \mathbf{c})} \frac{\partial}{\partial W_{ik}} Z(W, \mathbf{b}, \mathbf{c}) = E\left[V_i \sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right)\right]$$

Finally, follow the similarity, we have the following equality

$$E[V_i Y_k] = E\left[V_i \sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right)\right] = E\left[\sigma\left(\sum_{l=1}^{m} W_{il} Y_l + b_i\right) Y_k\right] \quad \text{(A.9)}$$

We can also differentiate the partition function with respect to biases and have

$$E[V_i] = E\left[\sigma\left(\sum_{l=1}^{m} W_{il} Y_l + b_i\right)\right], \text{ and } E[Y_k] = E\left[\sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right)\right]$$

- Joint distribution $P(\mathbf{V} = \mathbf{v}, \mathbf{Y} = \mathbf{y})$

$$\begin{aligned}
\frac{\partial}{\partial W_{ik}} P(\mathbf{V} = \mathbf{v}, \mathbf{Y} = \mathbf{y}) &= \frac{\partial}{\partial W_{ik}} \left(\frac{1}{Z(W, \mathbf{b}, \mathbf{c})} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) \right) \\
&= -\frac{\frac{\partial}{\partial W_{ik}} Z(W, \mathbf{b}, \mathbf{c})}{Z(W, \mathbf{b}, \mathbf{c})^2} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) \\
&+ \frac{1}{Z(W, \mathbf{b}, \mathbf{c})} \frac{\partial}{\partial W_{ik}} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) \\
&= P(\mathbf{V} = \mathbf{v}, \mathbf{Y} = \mathbf{y}) \left(v_i y_k - E\left[V_i Y_k \right] \right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial b_i} P(\mathbf{V} = \mathbf{v}, \mathbf{Y} = \mathbf{y}) &= \frac{\partial}{\partial b_i} \left(\frac{1}{Z(W, \mathbf{b}, \mathbf{c})} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) \right) \\
&= -\frac{\frac{\partial}{\partial b_i} Z(W, \mathbf{b}, \mathbf{c})}{Z(W, \mathbf{b}, \mathbf{c})^2} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) \\
&+ \frac{1}{Z(W, \mathbf{b}, \mathbf{c})} \frac{\partial}{\partial b_i} \exp(\mathbf{v}^T W \mathbf{y} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{y}) \\
&= P(\mathbf{V} = \mathbf{v}, \mathbf{Y} = \mathbf{y}) \left(v_i - E\left[V_i \right] \right)
\end{aligned}$$

and similarly,

$$\frac{\partial}{\partial c_k} P(\mathbf{V} = \mathbf{v}, \mathbf{Y} = \mathbf{y}) = P(\mathbf{V} = \mathbf{v}, \mathbf{Y} = \mathbf{y})(y_k - E[Y_k])$$

- Correlation: $E\left[\mathbf{V}^T W \mathbf{Y} \right]$

$$\begin{aligned}
\frac{\partial}{\partial W_{ik}} E\left[\mathbf{V}^T W \mathbf{Y} \right] &= \sum_{\mathbf{v}, \mathbf{y}} \left(\mathbf{v}^T W \mathbf{y} \frac{\partial}{\partial W_{ik}} P(\mathbf{v}, \mathbf{y}) + P(\mathbf{v}, \mathbf{y}) v_i y_k \right) \\
&= \sum_{\mathbf{v}, \mathbf{y}} \left(\mathbf{v}^T W \mathbf{y} P(\mathbf{v}, \mathbf{y}) \left(v_i y_k - E\left[V_i Y_k \right] \right) + P(\mathbf{v}, \mathbf{y}) v_i y_k \right) \\
&= E\left[\mathbf{V}^T W \mathbf{Y} \left(V_i Y_k \right) \right] - E[\mathbf{V}^T W \mathbf{Y}] E[V_i Y_k] + E\left[V_i Y_k \right]
\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial b_i}E\left[\mathbf{V}^T W\mathbf{Y}\right] &= \sum_{\mathbf{v},\mathbf{y}}\left(\mathbf{v}^T W\mathbf{y}\frac{\partial}{\partial b_i}P(\mathbf{v},\mathbf{y})\right)\\ &= \sum_{\mathbf{v},\mathbf{y}}\left(\mathbf{v}^T W\mathbf{y}P(\mathbf{v},\mathbf{y})\left(v_i - E\left[V_i\right]\right)\right)\\ &= E\left[\mathbf{V}^T W\mathbf{Y}\left(V_i\right)\right] - E[\mathbf{V}^T W\mathbf{Y}]E[V_i]\end{aligned}$$

and similarly,

$$\frac{\partial}{\partial c_k}E\left[\mathbf{V}^T W\mathbf{Y}\right] = E\left[\mathbf{V}^T W\mathbf{Y}\left(Y_k\right)\right] - E[\mathbf{V}^T W\mathbf{Y}]E[Y_k]$$

- Marginals: $P(\mathbf{V}=\mathbf{v}), P(\mathbf{Y}=\mathbf{y})$

$$\begin{aligned}&\frac{\partial}{\partial W_{ik}}P(\mathbf{V}=\mathbf{v})\\ &= \frac{\partial}{\partial W_{ik}}\left[\exp\left(\mathbf{b}^T\mathbf{v}\right)Z^{-1}(W,\mathbf{b},\mathbf{c})\prod_{l=1}^{m}\sigma^{-1}\left(-\left(\sum_{j=1}^{n}v_j W_{jl}+c_l\right)\right)\right]\\ &= \exp\left(\mathbf{b}^T\mathbf{v}\right)\left[\prod_{l=1}^{m}\sigma^{-1}\left(-\left(\sum_{j=1}^{n}v_j W_{jl}+c_l\right)\right)\left(\frac{-\frac{\partial Z(W,\mathbf{b},\mathbf{c})}{\partial W_{ik}}}{Z^2(W,\mathbf{b},\mathbf{c})}\right)\right.\\ &\left.+ Z^{-1}(W,\mathbf{b},\mathbf{c})\frac{\partial}{\partial W_{ik}}\prod_{l=1}^{m}\sigma^{-1}\left(-\left(\sum_{j=1}^{n}v_j W_{jl}+c_l\right)\right)\right]\\ &= -P(\mathbf{V}=\mathbf{v})E\left[V_i Y_k\right] + P(\mathbf{V}=\mathbf{v})\sigma\left(\sum_{j=1}^{n}v_j W_{jk}+c_k\right)v_i\\ &= P(\mathbf{V}=\mathbf{v})\left(\sigma\left(\sum_{j=1}^{n}v_j W_{jk}+c_k\right)v_i - E\left[\sigma\left(\sum_{j=1}^{n}V_j W_{jk}+c_k\right)V_i\right]\right)\end{aligned}$$

Similarly, we have

$$\begin{aligned}&\frac{\partial}{\partial W_{ik}}P(\mathbf{Y}=\mathbf{y})\\ &\quad= P(\mathbf{Y}=\mathbf{y})\left(\sigma\left(\sum_{l=1}^{m}W_{il}y_l+b_i\right)y_k - E\left[\sigma\left(\sum_{l=1}^{m}W_{il}Y_l+b_i\right)Y_k\right]\right)\end{aligned}$$

Now, turn to the biases

$$\frac{\partial}{\partial b_i}P(\mathbf{V}=\mathbf{v}) = \frac{\partial}{\partial b_i}\left[\exp\left(\mathbf{b}^T\mathbf{v}\right)Z^{-1}(W,\mathbf{b},\mathbf{c})\prod_{l=1}^{m}\sigma^{-1}\left(-\left(\sum_{j=1}^{n}v_jW_{jl}+c_l\right)\right)\right]$$
$$= \prod_{l=1}^{m}\sigma^{-1}\left(-\left(\sum_{j=1}^{n}v_jW_{jl}+c_l\right)\right)\frac{\partial}{\partial b_i}\left[\exp\left(\mathbf{b}^T\mathbf{v}\right)Z^{-1}(W,\mathbf{b},\mathbf{c})\right]$$
$$= P(\mathbf{V}=\mathbf{v})\left[v_i - E[V_i]\right]$$

$$\frac{\partial}{\partial c_k}P(\mathbf{V}=\mathbf{v}) = \frac{\partial}{\partial c_k}\left[\exp\left(\mathbf{b}^T\mathbf{v}\right)Z^{-1}(W,\mathbf{b},\mathbf{c})\prod_{l=1}^{m}\sigma^{-1}\left(-\left(\sum_{j=1}^{n}v_jW_{jl}+c_l\right)\right)\right]$$
$$= \exp\left(\mathbf{b}^T\mathbf{v}\right)\frac{\partial}{\partial c_k}\left[\prod_{l=1}^{m}\sigma^{-1}\left(-\left(\sum_{j=1}^{n}v_jW_{jl}+c_l\right)\right)Z^{-1}(W,\mathbf{b},\mathbf{c})\right]$$
$$= P(\mathbf{V}=\mathbf{v})\left[\sigma\left(\sum_{j=1}^{n}v_jW_{jk}+c_k\right) - E[Y_k]\right]$$

Similarly,

$$\frac{\partial}{\partial c_k}P(\mathbf{Y}=\mathbf{y}) = P(\mathbf{Y}=\mathbf{y})[y_k - E[Y_k]]$$

$$\frac{\partial}{\partial b_i}P(\mathbf{Y}=\mathbf{y}) = P(\mathbf{Y}=\mathbf{y})\left[\sigma\left(\sum_{l=1}^{m}W_{il}y_l+b_i\right) - E[V_i]\right]$$

- Sigmoid function: for any function $f:\mathbb{R}\to\mathbb{R}$

$$\frac{\partial}{\partial x}\log\sigma\left(f(x)\right) = \frac{1}{\sigma(f(x))}\frac{\partial}{\partial x}\sigma\left(f(x)\right) = \sigma(-f(x))\left(\frac{\partial}{\partial x}f(x)\right)$$

Therefore,

$$\frac{\partial}{\partial W_{ik}} \log \sigma \left(- \left(\sum_{j=1}^{n} v_j W_{jl} + c_l \right) \right) = \begin{cases} -\sigma \left(\sum_{j=1}^{n} v_j W_{jk} + c_k \right) v_i & \text{if } l = k \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial W_{ik}} \log \sigma \left(- \left(\sum_{l=1}^{m} W_{jl} y_l + b_j \right) \right) = \begin{cases} -\sigma \Big(\sum_{l=1}^{m} W_{il} y_l + b_i \Big) y_k & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial c_k} \log \sigma \left(- \left(\sum_{j=1}^{n} v_j W_{jl} + c_l \right) \right) = \begin{cases} -\sigma \left(\sum_{j=1}^{n} v_j W_{jk} + c_k \right) & \text{if } l = k \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial b_i} \log \sigma \left(- \left(\sum_{l=1}^{m} W_{jl} y_l + b_j \right) \right) = \begin{cases} -\sigma \Big(\sum_{l=1}^{m} W_{il} y_l + b_i \Big) & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$$

We have,

$$\begin{aligned} &\frac{\partial}{\partial W_{ik}} \sum_{l=1}^{m} E \left[\log \sigma \left(- \left(\sum_{j=1}^{n} V_j W_{jl} + c_l \right) \right) \right] \\ &\quad = \sum_{l=1}^{m} \sum_{\mathbf{v},\mathbf{y}} \left[P(\mathbf{v}, \mathbf{y}) \frac{\partial}{\partial W_{ik}} \log \sigma \left(- \left(\sum_{j=1}^{n} v_j W_{jl} + c_l \right) \right) \right. \\ &\qquad \left. + \log \sigma \left(- \left(\sum_{j=1}^{n} v_j W_{jl} + c_l \right) \right) \frac{\partial}{\partial W_{ik}} P(\mathbf{v}, \mathbf{y}) \right] \\ &\quad = -E \left[V_i \sigma \left(\sum_{j=1}^{n} V_j W_{jk} + c_k \right) \right] + \\ &\quad \sum_{l=1}^{m} E \left[\log \sigma \left(- \left(\sum_{j=1}^{n} V_j W_{jl} + c_l \right) \right) (V_i Y_k - E[V_i Y_k]) \right] \end{aligned}$$

$$
\begin{aligned}
&\frac{\partial}{\partial c_k}\sum_{l=1}^{m} E\left[\log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\right] \\
&\quad = \sum_{l=1}^{m}\sum_{\mathbf{v},\mathbf{y}}\left[P(\mathbf{v},\mathbf{y})\frac{\partial}{\partial c_k}\log\sigma\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right)\right. \\
&\qquad\qquad \left. + \log\sigma\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right)\frac{\partial}{\partial c_k}P(\mathbf{v},\mathbf{y})\right] \\
&\quad = -E\left[\sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right)\right] + \sum_{l=1}^{m} E\left[\log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)(Y_k - E[Y_k])\right]
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
&\frac{\partial}{\partial W_{ik}}\sum_{j=1}^{n} E\left[\log\sigma\left(-\left(\sum_{l=1}^{m} W_{jl} Y_l + b_j\right)\right)\right] \\
&\qquad = -E\left[\sigma\left(\sum_{l=1}^{m} W_{il} Y_l + b_i\right) Y_k\right] \\
&\qquad + \sum_{j=1}^{n} E\left[\log\sigma\left(-\left(\sum_{l=1}^{m} W_{jl} Y_l + b_j\right)\right)(V_i Y_k - E[V_i Y_k])\right]
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\partial}{\partial b_i}\sum_{j=1}^{n} E\left[\log\sigma\left(-\left(\sum_{l=1}^{m} W_{jl} Y_l + b_j\right)\right)\right] \\
&\qquad = -E\left[\sigma\left(\sum_{l=1}^{m} W_{il} Y_l + b_i\right)\right] \\
&\qquad + \sum_{j=1}^{n} E\left[\log\sigma\left(-\left(\sum_{l=1}^{m} W_{jl} Y_l + b_j\right)\right)(V_i - E[V_i])\right]
\end{aligned}
$$

Now, again from the marginal distribution point of view, we have

$$\begin{aligned}
&\frac{\partial}{\partial W_{ik}} \sum_{l=1}^{m} E\left[\log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\right] \\
&= \sum_{l=1}^{m}\sum_{\mathbf{v}} P(\mathbf{v}) \frac{\partial}{\partial W_{ik}} \log\sigma\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right) \\
&+ \sum_{l=1}^{m}\sum_{\mathbf{v}} \log\sigma\left(-\left(\sum_{j=1}^{n} v_j W_{jl} + c_l\right)\right) \frac{\partial}{\partial W_{ik}} P(\mathbf{v}) \\
&= -E\left[V_i \sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right)\right] \\
&+ E\left[\sum_{l=1}^{m} \log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right) \sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right) V_i\right] \\
&- E\left[\sum_{l=1}^{m} \log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\right] E\left[\sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right) V_i\right]
\end{aligned}$$

Therefore, we have the following equality

$$\begin{aligned}
&E\left[\sum_{l=1}^{m} \log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\left(V_i Y_k - E[V_i Y_k]\right)\right] \\
&= E\left[\sum_{l=1}^{m} \log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right) \sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right) V_i\right] \\
&- E\left[\sum_{l=1}^{m} \log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\right] E\left[\sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right) V_i\right]
\end{aligned}$$

This simply implies the following equalities

$$\begin{aligned}
&E\left[\sum_{l=1}^{m} \log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\left(V_i Y_k\right)\right] \\
&= E\left[\sum_{l=1}^{m} \log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\left(\sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right)\right) V_i\right]
\end{aligned}$$

Finally, combined with the equality of (A.9), we can derive the derivative of mutual information as

$$
\begin{aligned}
&\frac{\partial}{\partial W_{ik}} D(P(\mathbf{V},\mathbf{Y})||P(\mathbf{V})P(\mathbf{Y})) \\
&\quad = E\left[V_i Y_k\right] + E\left[\mathbf{V}^T W \mathbf{Y}\left(V_i Y_k\right)\right] - E[\mathbf{V}^T W \mathbf{Y}] E[V_i Y_k] + E\left[V_i Y_k\right] \\
&\qquad - E\left[V_i \sigma\left(\sum_{j=1}^{n} V_j W_{jk} + c_k\right)\right] - E\left[\sigma\left(\sum_{l=1}^{m} W_{il} Y_l + b_i\right) Y_k\right] \\
&\qquad + \sum_{l=1}^{m} E\left[\log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\left(V_i Y_k - E[V_i Y_k]\right)\right] \\
&\qquad + \sum_{j=1}^{n} E\left[\log\sigma\left(-\left(\sum_{l=1}^{m} W_{jl} Y_l + b_j\right)\right)\left(V_i Y_k - E[V_i Y_k]\right)\right] \\
&\quad = E\left[\mathbf{V}^T W \mathbf{Y}\left(V_i Y_k - E[V_i Y_k]\right)\right] \\
&\qquad + \sum_{l=1}^{m} E\left[\log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\left(V_i Y_k - E[V_i Y_k]\right)\right] \\
&\qquad + \sum_{j=1}^{n} E\left[\log\sigma\left(-\left(\sum_{l=1}^{m} W_{jl} Y_l + b_j\right)\right)\left(V_i Y_k - E[V_i Y_k]\right)\right]
\end{aligned}
\tag{A.10}
$$

Similarly,

$$
\begin{aligned}
&\frac{\partial}{\partial b_i} D(P(\mathbf{V},\mathbf{Y})||P(\mathbf{V})P(\mathbf{Y})) \\
&\quad = E\left[\mathbf{V}^T W \mathbf{Y}\left(V_i - E[V_i]\right)\right] \\
&\qquad + \sum_{j=1}^{n} E\left[\log\sigma\left(-\left(\sum_{l=1}^{m} W_{jl} Y_l + b_j\right)\right)\left(V_i - E[V_i]\right)\right]
\end{aligned}
\tag{A.11}
$$

$$
\begin{aligned}
&\frac{\partial}{\partial c_k} D(P(\mathbf{V},\mathbf{Y})||P(\mathbf{V})P(\mathbf{Y})) \\
&\quad = E\left[\mathbf{V}^T W \mathbf{Y}\left(Y_k - E[Y_k]\right)\right] \\
&\qquad + \sum_{l=1}^{m} E\left[\log\sigma\left(-\left(\sum_{j=1}^{n} V_j W_{jl} + c_l\right)\right)\left(Y_k - E[Y_k]\right)\right]
\end{aligned}
\tag{A.12}
$$

When the above derivative equals 0 for all i, k, we say this is the necessary condition to achieve supremum value of mutual information, *i.e.*, channel capacity.

Bibliography

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, and W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989

[2] H. Zhang, V. M. Patel, M. Fathy, and R. Chellappa, "Touch Gesture-Based Active User Authentication Using Dictionaries," *WACV*, pp. 207–214, January 2015.

[3] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, January 2013.

[4] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] M. Berglund,T. Raiko, and K. Cho, "Measuring the usefulness of hidden units in Boltzmann machines with mutual information," *Neural Networks*, vol. 64, pp. 12–18, 2015

[6] T. Tieleman, "Training restricted Boltzmann machines using approximations to the likelihood gradient," *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071, 2008.

[7] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *International Conference on Machine Learning (ICML-10)*, 2010.

[8] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," *Artificial Neural Networks and Machine Learning–ICANN 2011*, pp. 52–59, 2011.

[9] Q. Qiu, V. Patel and R. Chellappa, "Information-theoretic dictionary learning for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2173–2184, 2014.

[10] M.-Y. Liu and O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 99–112, 2014.

[11] A. Krizhevsky and I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[12] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CVPR*, pp. 580–587, 2014.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *IEEE Proceedings*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] S. A Nene, S. K Nayar, H. Murase, and others, "Columbia object image library (COIL-20)."

[15] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[16] R. Salakhutdinov, "Learning and evaluating Boltzmann machines," *Technical Report UTML TR 2008-002, Department of Computer Science, University of Toronto*, 2008.

[17] S. Nitish, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research 15*, pp. 1929–1958, 2014.

[18] R. M. Neal, "Annealed Importance Sampling," *Statistics and Computing*, vol. 11, pp. 125–139, 1998.

[19] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Proceedings of the 27th International Conference on Machine Learning*, pp. 3371–3408, 2010.

[20] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[21] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[22] R. Salakhutdinov, and G. E. Hinton, "Deep boltzmann machines," *International Conference on Artificial Intelligence and Statistics*, pp. 448–455, 2009.

[23] G. E. Hinton, and R. Salakhutdinov, "A better way to pretrain deep Boltzmann machines," *Advances in Neural Information Processing Systems*, pp. 2447–2455, 2012.

[24] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 609–616, 2009.

[25] I. Sutskever, J. Martens, G. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning," *ICML-13*, pp. 1139–1147, 2013.

[26] N. Le Roux, and Y. Bengio, "Representational power of restricted Boltzmann machines and deep belief networks," *Neural computation*, vol. 20, no. 6, pp. 1631–1649, 2008.

[27] N. Le Roux, and Y. Bengio, "Deep belief networks are compact universal approximators," *Neural computation*, vol. 22, no. 8, pp. 2192–2207, 2010.

[28] J. Dai and Y.-N. Wu, "Generative Modeling of Convolutional Neural Networks," *ICLR-15*, 2015.

[29] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," *NIPS 2012 Workshop*, 2012.