Boltzmann Machine

Peter He

December 2020

1 Introduction

This project is an exploration on the image denoising (neat cat vs noisy cat) with Gibbs Sampler in the Senior Seminar in Applied Probability. We take a dive into restricted Boltzmann machines (RBM). RBMs can be used to approximate a complex unknown probability distribution (e.g. the joint probability distribution of each pixel in an image) based on samples from this distribution. Due to the hidden units, the process of approximating the target distribution induces some generalized features of the target distribution, which proves useful in classification and dimension reduction tasks. This paper introduces a maximum likelihood approach to estimate the target distribution. A Gibbs sampler is used to approximate the gradients of the likelihood function.

2 Definition of a RBM

Similar to a Markov chain, a RBM is a collection of random variables that satisfy a set of conditional relationships between them, which allow them to be visually represented using graphs. A RBM is a Markov random field (MRF) and satisfies the definition of a MRF, which specifies that any three disjoint subsets of the random variables satisfy a conditionally independent relationship.

Definition 1.
$\boldsymbol{X} = \{X_v \mid v \in V\}$ is a Markov random field (MRF) if for all disjunct subsets $\mathcal{A}, \mathcal{B}, \mathcal{S} \subset V$, all nodes in $\mathcal{A}, \mathcal{B}$ are separated by $\mathcal{S}$. For all nodes $X_a, X_b, X_s \in \mathcal{A}, \mathcal{B}, \mathcal{S}$, X_a and X_b are conditionally independent given X_s.

Definition 2. *Two sets of random variables X_1 X_2 are conditionally independent given a set of random variables X_3 if*

$$p(X_1, X_2, |X_3) = p(X_1|X_3)P(X_2|X_3)$$

As we saw in solving the Hidden Markov Model, conditional independence allows nice factorization of join probability distribution, which is going to play a role in a RBM as well. In addition to being a MRF, a RBM can be represented using a bipartite graph. The random variables inside a RBM can be divided into the visible variables, $\boldsymbol{V}$, and hidden variables, $\boldsymbol{H}$ such that all of the variables in $\boldsymbol{H}$ are separated by a variable in $\boldsymbol{V}$. Similarly, all of the variables in $\boldsymbol{V}$ are separated by a variable in $\boldsymbol{H}$. By the definition of MRF, for all $V_i, V_j \in \boldsymbol{V}$, V_i is conditionally independent

from V_j given that that $\boldsymbol{H} = \boldsymbol{h}$. The same can be said for all the variables in $\boldsymbol{H}$. In probability, we will have,

$$\begin{aligned} p(\boldsymbol{H}|\boldsymbol{V}) &= p(H_1, H_2, \dots H_n|\boldsymbol{V}) \\ &= p(H_1|\boldsymbol{V})P(H_2, \dots H_n|\boldsymbol{V}) \\ &= p(H_1|\boldsymbol{V})P(H_2|\boldsymbol{V})p(H_3, \dots H_n|\boldsymbol{V}) \\ &= \prod_{i=1}^{n} p(H_i|\boldsymbol{V}). \end{aligned}$$

We can also write

$$p(\boldsymbol{h}|\boldsymbol{v}) = \prod_{i=1}^{n} p(h_i|\boldsymbol{v}) \tag{1}$$

We also have

$$p(\boldsymbol{v}|\boldsymbol{h}) = \prod_{i=1}^{n} p(v_i|\boldsymbol{h}) \tag{2}$$

Now that you have seen some notations, I believe that it's helpful to introduce what they mean precisely. In this paper, we will use $p(\boldsymbol{v})$ to denote $p(\boldsymbol{V} = \boldsymbol{v})$, where the lower case $\boldsymbol{v}$ denotes the set of real numbers that the set of random variables $\boldsymbol{V}$ can have. Additionally, $p(v_j)$ denotes $p(V_j = v_j)$. In this case, V_j denotes the j^{th} random variable inside V_j and v_j denotes the value of j^{th} random variable $\boldsymbol{V}_j$ in $\boldsymbol{V}$.

Besides the restricted bipartite graphical structure, a RBM also has a specific energy function. Suppose that we have n hidden random variables and m visible random variables. Let w_{ij} be the weight associated with the edge between H_i and V_j. Let c_i be the bias terms associated with the H_i, b_j with V_j. Then for an arbitrary state of the RBM, the energy function is calculated as

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i. \tag{3}$$

Figure 1 is a visual representation of the weights and biases.

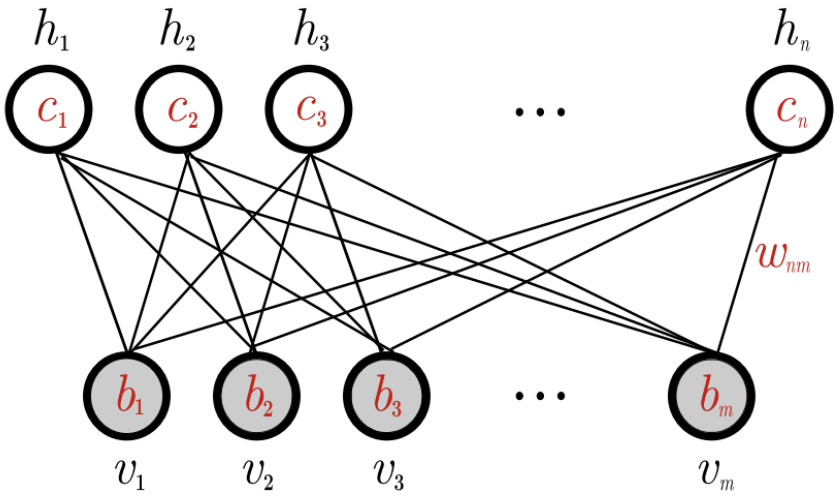

Figure 1: A graph of Boltzmann machine

The wights and biases parametrize a RBM under the structure defined by the definition of a MRF and a bipartite graph. This perspective of the weights and biases proves to be helpful in using a RBM to approximate an arbitrary distribution. Additionally, as we saw in the Metropolis-Hastings algorithm, we can map the energy function to a probability density function using Gibbs distribution. That is

$$p(\boldsymbol{v}, \boldsymbol{h}) \, \alpha \, e^{-E(\boldsymbol{v}, \boldsymbol{h})}.$$

Since the weights and biases parametrize the energy function and the probability distribution, we can write the probability distribution in this form to emphasize the parametrization relationship. Let $\boldsymbol{\theta}$ denote the weights and biases,

$$p(\boldsymbol{v}, \boldsymbol{h}|\boldsymbol{\theta}) \, \alpha \, e^{-E(\boldsymbol{v}, \boldsymbol{h})}.$$

Finally, all random variables in a RBM are Bernoulli variables.

3 Application

Suppose that we want to estimate an unknown joint probability distribution q of m random variables given l i.i.d sample of q, $S = \{\boldsymbol{v}_i \in \mathbb{R}^m \mid 1 \leq i \leq l\}$. By treating $\boldsymbol{v}_i$ a sample of the visible random variables, we can approximate q by finding the most likely joint probability distribution of the visible random variables that would yield the samples S. Taking the maximum likelihood estimate approach, our goal is to find the parameters $\boldsymbol{\theta}$ (the weights and biases in the RBM) that maximizes

the log-likelihood function $\ln(\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{v}))$. Since the these samples are independent, we have

$$\begin{aligned}\ln(\mathcal{L}(\boldsymbol{\theta}|S)) &= \ln \prod_{i=1}^{l} p(\boldsymbol{v_i}|\theta) \\ &= \sum_{i=1}^{l} \ln p(\boldsymbol{v_i}|\theta)\end{aligned}$$

By the law of total probabilities, the marginal distribution of $\boldsymbol{v}$ is calculated by summing over the joint probability $p(\boldsymbol{v}, \boldsymbol{h})$ over all possible values of $\boldsymbol{h}$,

$$\begin{aligned}p(\boldsymbol{v}) &= \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \\ &\alpha \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v}, \boldsymbol{h})}.\end{aligned}$$

If you haven't noticed so far, calculating $p(\boldsymbol{v})$ is very problematic. Since $\boldsymbol{v} \in \{0,1\}^m$, the runtime of this calculation grows exponentially in 2^m. We shall see why exactly this is problematic and how it is resolved. To continue, the log-likelihood function is,

$$\begin{aligned}\sum_{i=1}^{l} \ln p(\boldsymbol{v_i}|\theta) &= \sum_{i=1}^{l} \ln \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \\ &= \sum_{i=1}^{l} \ln \left[\frac{\sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h})}{1} \right].\end{aligned}$$

By the law of total probabilities, the sum the probabilities of all possible values of $\boldsymbol{v}$ and $\boldsymbol{h}$ should be 1. Then, it follows that

$$\begin{aligned}\sum_{i=1}^{l} \ln \left[\frac{\sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h})}{1} \right] &= \sum_{i=1}^{l} \ln \left[\frac{\sum_{\boldsymbol{h}} p(\boldsymbol{v}_i, \boldsymbol{h})}{\sum_{\boldsymbol{v}, \boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h})} \right] \\ &= \sum_{i=1}^{l} \left[\ln \sum_{\boldsymbol{h}} p(\boldsymbol{v}_i, \boldsymbol{h}) - \ln \sum_{\boldsymbol{v}, \boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \right]\end{aligned}$$

At this point, we should differentiate the $\boldsymbol{v}_i$ that are samples and arbitrary $\boldsymbol{v}$ in the sum. We have

$$\ln(\mathcal{L}(\boldsymbol{\theta}|S)) = \sum_{i=1}^{l} \left[\ln \sum_{\boldsymbol{h}} p(\boldsymbol{v}_i^o, \boldsymbol{h}) - \ln \sum_{\boldsymbol{v}, \boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \right]$$

For now, we can calculate the gradient of the log-likelihood with respect to one arbitrary sample and sum the gradients over all samples later. We have

$$
\begin{aligned}
\frac{\partial \ln(\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{v}^o))}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} \left[\ln \sum_{\boldsymbol{h}} p(\boldsymbol{v^o}, \boldsymbol{h}) \right] - \frac{\partial}{\partial \boldsymbol{\theta}} \left[\ln \sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \right] \\
&= \frac{\partial}{\partial \boldsymbol{\theta}} \left[\ln \sum_{\boldsymbol{h}} exp(-E(\boldsymbol{v^o}, \boldsymbol{h})) \right] - \frac{\partial}{\partial \boldsymbol{\theta}} \left[\ln \sum_{\boldsymbol{v},\boldsymbol{h}} exp(-E(\boldsymbol{v}, \boldsymbol{h})) \right] \\
&= -\frac{1}{\sum_{\boldsymbol{h}} exp(-E(\boldsymbol{v^o}, \boldsymbol{h}))} \left[\sum_{\boldsymbol{h}} exp(-E(\boldsymbol{v^o}, \boldsymbol{h})) \frac{\partial E(\boldsymbol{v^o}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \right] \\
&+ \frac{1}{\sum_{\boldsymbol{v},\boldsymbol{h}} exp(-E(\boldsymbol{v}, \boldsymbol{h}))} \left[\sum_{\boldsymbol{h}} exp(-E(\boldsymbol{v}, \boldsymbol{h})) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \right] \\
&= -\frac{1}{\sum_{\boldsymbol{h}} p(\boldsymbol{v^o}, \boldsymbol{h})} \left[\sum_{\boldsymbol{h}} p(\boldsymbol{v^o}, \boldsymbol{h}) \frac{\partial E(\boldsymbol{v^o}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \right] \\
&+ \frac{1}{\sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h})} \left[\sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \right] \\
&= -\left[\sum_{\boldsymbol{h}} \frac{p(\boldsymbol{v^o}, \boldsymbol{h})}{\sum_{\boldsymbol{h}} p(\boldsymbol{v^o}, \boldsymbol{h})} \frac{\partial E(\boldsymbol{v^o}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \right] \\
&+ \left[\sum_{\boldsymbol{h}} \frac{p(\boldsymbol{v}, \boldsymbol{h})}{\sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h})} \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \right] \\
&= -\left[\sum_{\boldsymbol{h}} \frac{p(\boldsymbol{v^o}, \boldsymbol{h})}{p(\boldsymbol{v}^o)} \frac{\partial E(\boldsymbol{v^o}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \right] \\
&+ \left[\sum_{\boldsymbol{h}} \frac{p(\boldsymbol{v}, \boldsymbol{h})}{1} \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial \boldsymbol{\theta}} \right] \\
&= -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^o) \frac{\partial E(\boldsymbol{v}^o, \boldsymbol{h})}{\partial \boldsymbol{\theta}} + \sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial \boldsymbol{\theta}}.
\end{aligned}
$$

Two observations on the resulting formula for gradients should be noted. The first is that the calculation for the second term of the gradient grows exponentially with the number of hidden and visible variables. Secondly, the gradient can be viewed as the difference between two expected

values with distribution $p(\boldsymbol{h}|\boldsymbol{v}^o)$ and $p(\boldsymbol{v},\boldsymbol{h})$. For parameter w_{ij}, the first term of the gradient is

$$\begin{aligned}
-\sum_h p(\boldsymbol{h}|\boldsymbol{v^o})\frac{\partial E(\boldsymbol{v^o},\boldsymbol{h})}{\partial w_{ij}} &= \sum_h p(\boldsymbol{h}|\boldsymbol{v^o})h_i v_j^o \\
&= \sum_h \prod_{k=1}^{n} p(h_k|\boldsymbol{v^o})h_i v_j^o \\
&= \sum_{h_i} p(h_i|\boldsymbol{v^o})h_i v_j^o \\
&= p(H_i=1|\boldsymbol{v^o})v_j^o.
\end{aligned}$$

Thus, the gradient of the log-likelihood function is

$$\begin{aligned}
\frac{\partial \ln \mathcal{L}}{w_{ij}} &= -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v^o})\frac{\partial E(\boldsymbol{v^o},\boldsymbol{h})}{\partial \boldsymbol{\theta}} + \sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h})\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} \\
&= p(H_i=1|\boldsymbol{v^o})v_j^o - \sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v})p(\boldsymbol{h}|\boldsymbol{v})\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} \\
&= p(H_i=1|\boldsymbol{v}^o)v_j^o - \sum_{\boldsymbol{v}} p(\boldsymbol{v})\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v})\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} \\
&= p(H_i=1|\boldsymbol{v}^o)v_j^o - \sum_{\boldsymbol{v}} p(\boldsymbol{v})p(H_i=1|\boldsymbol{v})v_j.
\end{aligned}$$

The gradients with respect to b_j, c_i are

$$\begin{aligned}
\frac{\partial \ln \mathcal{L}}{b_j} &= v_j^o - \sum_{\boldsymbol{v}} p(\boldsymbol{v})v_j, \\
\frac{\partial \ln \mathcal{L}}{c_i} &= p(H_i=1|\boldsymbol{v}^o) - \sum_{\boldsymbol{v}} p(\boldsymbol{v})p(H_i=1|\boldsymbol{v}).
\end{aligned}$$

The term $p(H_i = 1|\boldsymbol{v})$ is calculable given that the hidden random variables are conditionally independent from each other. Let $\boldsymbol{h}_{-i}$ denotes the values of the set of hidden variables beside the i^{th} hidden variable. Using the conditionally independent property, for any values of $\boldsymbol{h}_{-i}, \boldsymbol{v}$

$$\begin{aligned}
p(H_i=1|\boldsymbol{v}) &= p(H_i=1|\boldsymbol{v},\boldsymbol{h}_{-i}) \\
&= \frac{p(H_i=1,\boldsymbol{v},\boldsymbol{h}_{-i})}{p(\boldsymbol{v},\boldsymbol{h}_{-i},h_i=1)+p(\boldsymbol{v},\boldsymbol{h}_{-i},h_i=0)} \\
&= \frac{exp(-E(h_i=1,\boldsymbol{v},\boldsymbol{h}_{-i}))}{exp(-E(\boldsymbol{v},\boldsymbol{h}_{-i},h_i=0))+exp(-E(\boldsymbol{v},\boldsymbol{h}_{-i},h_i=1))} \\
&= \frac{exp(\sum_{k=1,k\neq i}^{n}\sum_{j=1}^{m} w_{kj}h_k v_j + \sum_{j=1}^{m} b_j v_j + \sum_{k=1,k\neq i}^{n} c_k h_k + \sum_{j=1}^{m} w_{ij}v_j + c_i)}{exp(\sum_{k=1,k\neq i}^{n}\sum_{j=1}^{m} w_{kj}h_k v_j + \sum_{j=1}^{m} b_j v_j + \sum_{k=1,k\neq i}^{n} c_k h_k + \sum_{j=1}^{m} w_{ij}v_j + c_i) + exp(\sum_{k=1,k\neq i}^{n}\sum_{j=1}^{m} w_{kj}h_k v_j + \sum_{j=1}^{m} b_j v_j + \sum_{k=1,k\neq i}^{n} c_k h_k))} \\
&= \frac{exp(\sum_{k=1,k\neq i}^{n}\sum_{j=1}^{m} w_{kj}h_k v_j + \sum_{j=1}^{m} b_j v_j + \sum_{k=1,k\neq i}^{n} c_k h_k)exp(\sum_{j=1}^{m} w_{ij}v_j + c_i)}{exp(\sum_{k=1,k\neq i}^{n}\sum_{j=1}^{m} w_{kj}h_k v_j + \sum_{j=1}^{m} b_j v_j + \sum_{k=1,k\neq i}^{n} c_k h_k)(1+exp(\sum_{j=1}^{m} w_{ij}v_j + c_i))} \\
&= \frac{exp(\sum_{j=1}^{m} w_{ij}v_j + c_i)}{(1+exp(\sum_{j=1}^{m} w_{ij}v_j + c_i))} \\
&= \frac{1}{1+exp(-\sum_{j=1}^{m} w_{ij}v_j + c_i)}
\end{aligned}$$

Let $\sigma(x) = \frac{1}{1+e^{-x}}$ be the sigmoid function, then

$$p(H_i = 1|\boldsymbol{v}) = \sigma(\sum_{j=1}^{m} w_{ij}v_j + c_i).$$

4 Solution

In the gradient formula from the last section, we are capable of calculating every terms except the summation over all possible values of $\boldsymbol{v}$. The number of terms inside the summation grows exponentially with the number of visible random variables m. However, we can use the MCMC to approximate $\sum_{\boldsymbol{v}} p(\boldsymbol{v})p(H_i = 1|\boldsymbol{v})v_j$ and $\sum_{\boldsymbol{v}} p(\boldsymbol{v})v_j$. Although we are not going to get the precise value of the gradient, we are at least getting an estimated value of the gradient.

4.1 Gibbs Sampler

Instead of calculating the second term of the gradient directly, we are going to use a Gibbs sampler to estimate it. The target distribution is the marginal distribution $p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h})$, which cannot be evaluated in reasonable time. Since the joint probability distribution $p(\boldsymbol{v}, \boldsymbol{h}) = exp(-E(\boldsymbol{v}, \boldsymbol{h}))$ is calculable, we can sample $\boldsymbol{v}, \boldsymbol{h}$ at the same time from the joint distribution.

The Gibbs sampler from class allows us to sample a one random variable while holding all other random variables constant. Sampling one random variable changes its value and changes the conditional distribution from which other random variables are sampled. Hence, sampling is done one at a time. In the case of a RBM, since the visible random variables are conditionally independent from each other given values of the hidden variables, the condition distribution from which any visible random variables are sampled from stays the same if any other random variables change their values. For any visible random variable V_j and any values of other visible random variables

$$p(V_j|\boldsymbol{v}_{-j}, \boldsymbol{h}) = p(V_j|\boldsymbol{h}).$$

We can sample all the hidden variables at once and all the visible variables at once, rather than sampling each one sequentially. Instead of choosing a random variable to sample randomly like the Gibbs sampler from class, we fix the order of the random variables to be sampled. As we sample more and more, the properties of Markov Chain guarantees that as $t \to \infty$, the distribution of the sample $\boldsymbol{v}^{(t)}, \boldsymbol{h}^{(t)}$ will converge to the target distribution $p(\boldsymbol{v}, \boldsymbol{h})$. The marginal distribution of the samples $p(\boldsymbol{v}^{(t)})$ also converges to $p(\boldsymbol{v})$. Calculating the expected values in the second term of the gradients is just averaging the value of the second term over all samples, $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)} \dots \boldsymbol{v}^{(T)}$. In the case of $\frac{\partial \ln \mathcal{L}}{w_{ij}}$, the approximation for

$$\sum_{\boldsymbol{v}} p(\boldsymbol{v})p(H_i = 1|\boldsymbol{v})v_j$$

is

$$\frac{1}{T}\sum_{t}^{T} p(H_i = 1|\boldsymbol{v^{(t)}})v_j^{(t)}.$$

Algorithm 1 Training BM with Gradient Descent and Gibbs Sampling

- Inputs: learning rate α, $\boldsymbol{v}^{observation}$
- **for** $(1,1)\dots(i,j)\dots(n,m)$ **do**:
  - Initialize w_{ij}, b_j, c_i
- **while** true **do:**
  - Randomly initialize $\boldsymbol{v}^{(1)}$
  - **for** $t = 1, \dots T$ **do:**
    - ∗ **for** $j = 1, \dots m$ **do:**
      - · Generate $h_j^{(t)} \sim p(H_j|\boldsymbol{v}^{(t-1)})$ to get $\boldsymbol{h}^{(t)}$
    - ∗ **for** $i = 1, \dots n$ **do:**
      - · Generate $v_i^{(t+1)} \sim p(V_i|\boldsymbol{h}^{(t)})$ to get $\boldsymbol{v}^{(t+1)}$
  - $w_{ij} := w_{ij} + \alpha \cdot \left[\sigma(\sum_{j=1}^{m} w_{ij} v_j^{(observation)}) - \frac{1}{T}\sum_t^T \sigma(\sum_{j=1}^{m} w_{ij} v_j^{(t)}) v_j^{(t)}\right]$
  - $b_j := b_j + \alpha \cdot \left[v_j^{(t)} - \frac{1}{T}\sum_t^T v_j^{(t)}\right]$
  - $c_i := c_i + \alpha \cdot \left[\sigma(\sum_{j=1}^{m} w_{ij} v_j^{(observation)}) - \frac{1}{T}\sum_t^T \sigma(\sum_{j=1}^{m} w_{ij} v_j^{(t)})\right]$

4.2 Contrastive Divergence

In practice, it is still quite expensive to run the Markov Chain till convergence. In Algorithm 1, we start sampling from a random point $\boldsymbol{v}^{(0)}$. From there it is going to take a very long time to reach the stationary distribution p. Instead, we can let $\boldsymbol{v}^{(0)} = \boldsymbol{v}^o$, which is much closer to the stationary distribution than a random point in the state space. Additionally, instead of using the average over many samples, the second term in the gradient is estimated by a single sample. Suppose that there are k samples from Gibbs Sampler, then the second term of the gradient is estimated by

$$p(H_i = 1|\boldsymbol{v}^{(\boldsymbol{k})})v_j.$$

Concretely, $\frac{\partial \ln \mathcal{L}}{w_{ij}}$ is approximated by

$$p(H_i = 1|\boldsymbol{v}^{(o)})v_j - p(H_i = 1|\boldsymbol{v}^{(\boldsymbol{k})})v_j.$$

Additionally, $\frac{\partial \ln \mathcal{L}}{b_j}$ and $\frac{\partial \ln \mathcal{L}}{c_i}$ are approximated by

$$v_j^{(o)} - v_j^{(k)},$$
$$p(H_i = 1|\boldsymbol{v}^{(o)}) - p(H_i = 1|\boldsymbol{v}^{(\boldsymbol{k})}),$$

respectively.

Algorithm 2 Contrastive Divergence

- Inputs: learning rate α, $\boldsymbol{v}^o$
- **for** $(1,1)\dots(i,j)\dots(n,m)$ **do**:
  - Initialize w_{ij}, b_j, c_i
- **while** true **do:**
  - Let $\boldsymbol{v}^{(1)} = \boldsymbol{v}^o$
  - **for** $t = 1, \dots k$ **do:**
    - ∗ **for** $j = 1, \dots m$ **do:**
      - · Generate $h_j^{(t)} \sim p(H_j|\boldsymbol{v}^{(t-1)})$ to get $\boldsymbol{h}^{(t)}$
    - ∗ **for** $i = 1, \dots n$ **do:**
      - · Generate $v_i^{(t+1)} \sim p(V_i|\boldsymbol{h}^{(t)})$ to get $\boldsymbol{v}^{(t+1)}$
  - $w_{ij} := w_{ij} + \alpha \cdot \left[\sigma(\sum_{j=1}^m w_{ij} v_j^{(o)}) - \sigma(\sum_{j=1}^m w_{ij} v_j^{(k)}) v_j^{(k)}\right]$
  - $b_j := b_j + \alpha \cdot \left[v_j^{(t)} - v_j^{(k)}\right]$
  - $c_i := c_i + \alpha \cdot \left[\sigma(\sum_{j=1}^m w_{ij} v_j^{(o)}) - \sigma(\sum_{j=1}^m w_{ij} v_j^{(k)})\right]$

5 Result & Conclusion

Using contrastive divergence, I trained a restricted Boltzmann machine with 10 hidden variables on the MNIST dataset. Each observation in the MNIST is a picture of a handwritten digit with 728 pixels. Note that although the image has very low resolution, the number of operations involved in directly calculating the second term of the likelihood gradient $\sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1|\boldsymbol{v}) v_j$ is at least 2^{728}. After performing 100 rounds iterations of gradient descent, I reached a decent set of weights and biases. One application of the RBM is dimension reduction. I used an image of a handwritten 7 (Figure 2) as a sample of the visible random variables, $\boldsymbol{v}^o$. The conditional probability $p(\boldsymbol{h}|\boldsymbol{v}^o)$

is calculated (Figure 3), which has only 10 variables. The conditional probability of the hidden variables $p(\boldsymbol{h}|\boldsymbol{v}^o)$ can be thought of as a compressed representation of $\boldsymbol{v}^o$ that captures some features of the marginal distribution of the visible variables. To verify that the hidden variables capture some structure of the handwritten digit 7, I sample the value of the hidden variables according to the conditional probability and calculate the conditional probability of the visible variables given the sampled hidden variables, $p(\boldsymbol{v}|\boldsymbol{h})$. The result is an image that captures an impression of 7 (Figure 4). In plain words, I give an image of 7, $p(\boldsymbol{v}^o)$, to the RBM; the RBM spits out the relevant features of the input image $p(\boldsymbol{h}|\boldsymbol{v}^o)$; the RBM can reconstruct an image that has these features $p(\boldsymbol{v}|\boldsymbol{h})$.

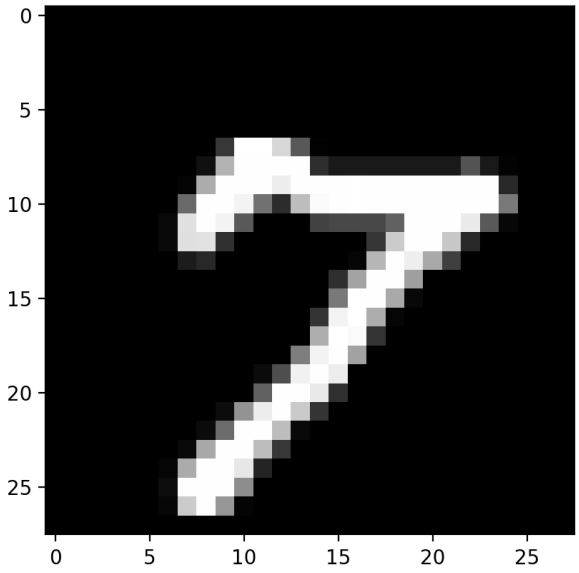

Figure 2: $\boldsymbol{v}^o$

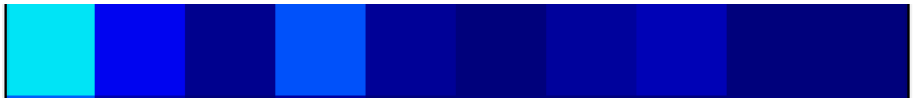

Figure 3: $\boldsymbol{p}(h|v^o)$

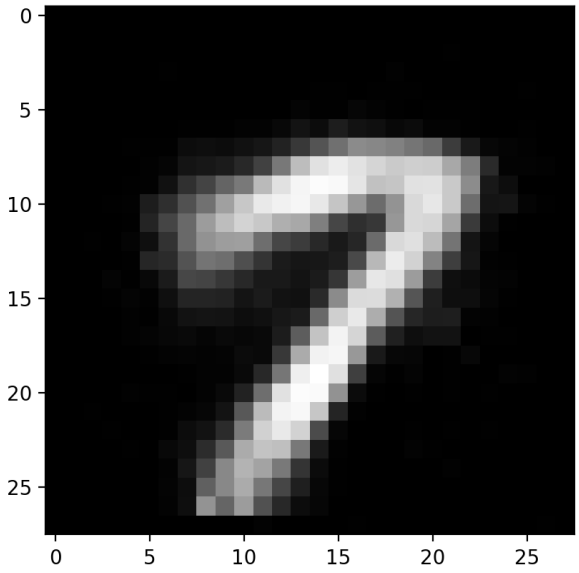

Figure 4: $\boldsymbol{p}(v|h)$

The ability to extract feature can be further verified by comparing the hidden variables given images of different digits. In Figure 5, each row represents the average probabilities of the hidden variables $p(\boldsymbol{h}|\boldsymbol{v}^o)$ given 100 images of each digit. With the exception of digit 3, 5 and 8, the probability distribution in each row is distinct from each other. Considering that the RBM is only allowed to extract 10 features out of 728 variables, the level of the distinction is quite impressive.

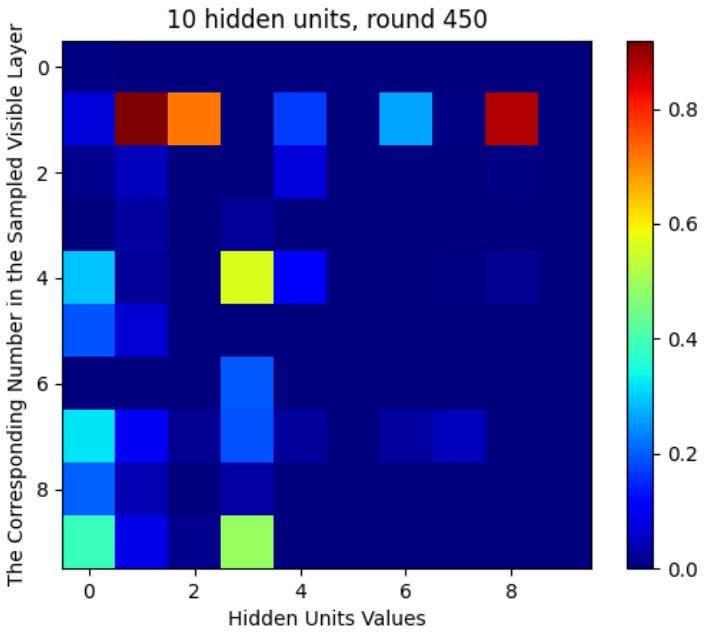

Figure 5: $p(\boldsymbol{h}|\boldsymbol{v}^o)$

The appendix shows the conditional probability of the visible variables conditioned on that one hidden unit is set to 1 and others are set to 0. From Figure 5, we can see that digit 4,7, and 9 are associated with higher value of hidden unit 0. The activation of the hidden variable 0 corresponds to a circle on the top left and a vertical stroke attached to the to right. From Figure 5, we can see that digit 4 and 9 are associated with activation of the hidden unit 3, which seems to extract the feature of a 4-like shape. Additionally, the hidden variable 4 is associated with a diagonal stroke from right to left, which is strongly associated with the digit 4. Other hidden units don't display features as salient as the above mentioned. Their corresponding visible unit representations are in the appendix.

The ability to extract features is useful in classification tasks. The values of the hidden variables corresponding to each image can be fed into a logistic regression model to classify the digit that corresponds to the image. This construct is a basic architecture of a feed-forward neural network. Perhaps the RBM can give us some intuition as to why neural networks have been much more successful in image classification task than previous benchmarks. Each layer of the neural network captures some features of the results from previous layers. As features get extracted from features, more abstract features emerge. The logistic regression at the output layer of a neural network use these abstract features to separate different groups of observations.

6 Appendix

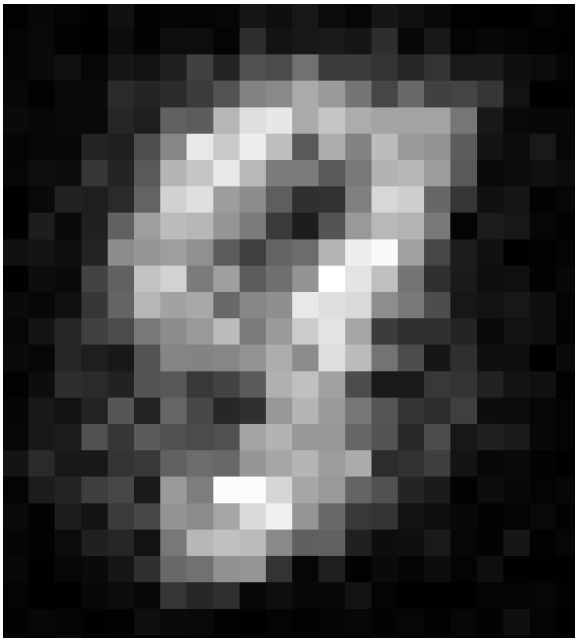

Figure 6: $p(\boldsymbol{v}|\boldsymbol{h} = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0))$

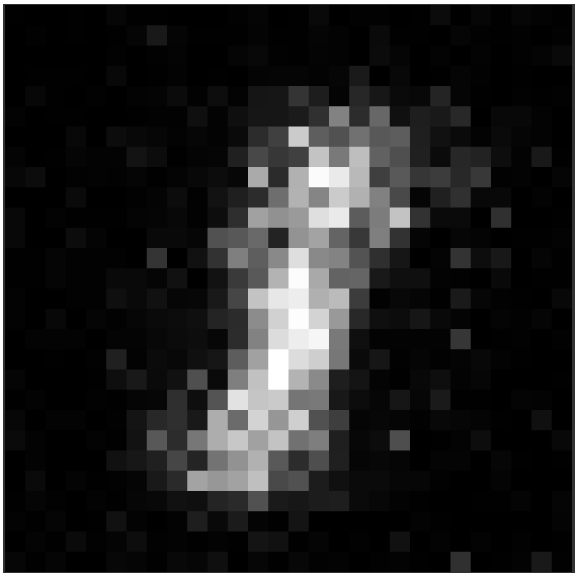

Figure 7: $p(\boldsymbol{v}|\boldsymbol{h} = (0, 1, 0, 0, 0, 0, 0, 0, 0, 0))$

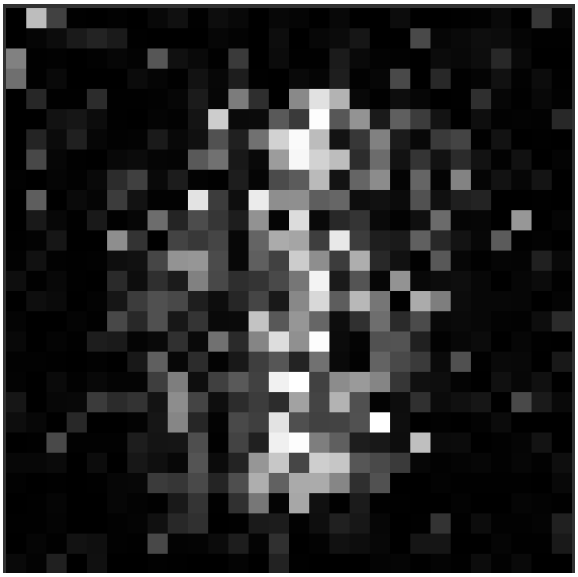

Figure 8: $p(\boldsymbol{v}|\boldsymbol{h} = (0, 0, 1, 0, 0, 0, 0, 0, 0, 0))$

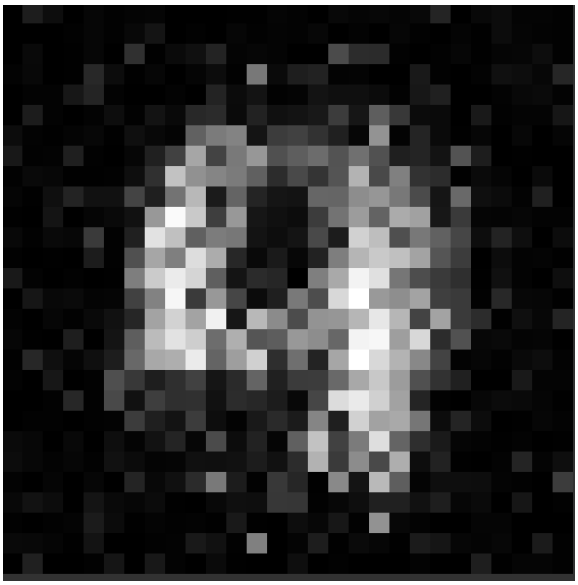

Figure 9: $p(\boldsymbol{v}|\boldsymbol{h} = (0, 0, 0, 1, 0, 0, 0, 0, 0, 0))$

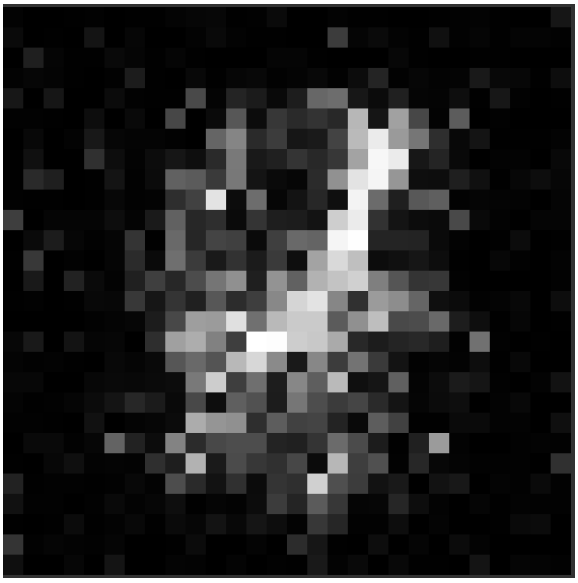

Figure 10: $p(\boldsymbol{v}|\boldsymbol{h} = (0, 0, 0, 0, 1, 0, 0, 0, 0, 0))$

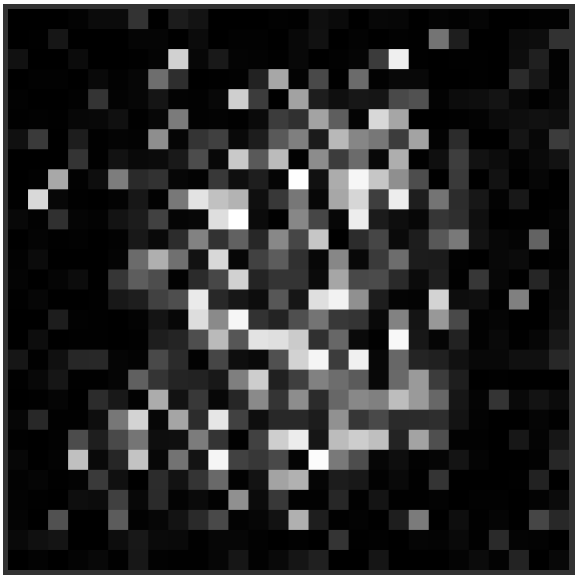

Figure 11: $p(\boldsymbol{v}|\boldsymbol{h} = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0))$

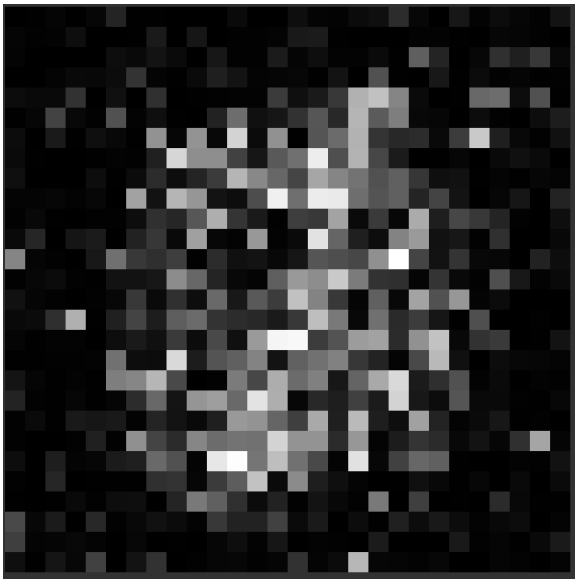

Figure 12: $p(\boldsymbol{v}|\boldsymbol{h} = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0))$

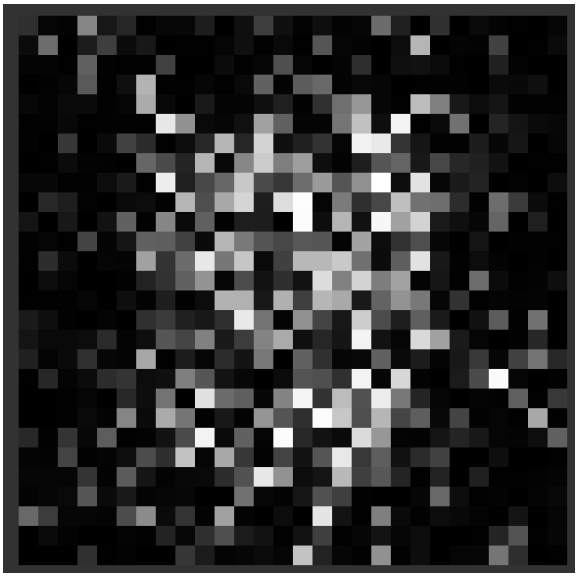

Figure 13: $p(\boldsymbol{v}|\boldsymbol{h} = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0))$

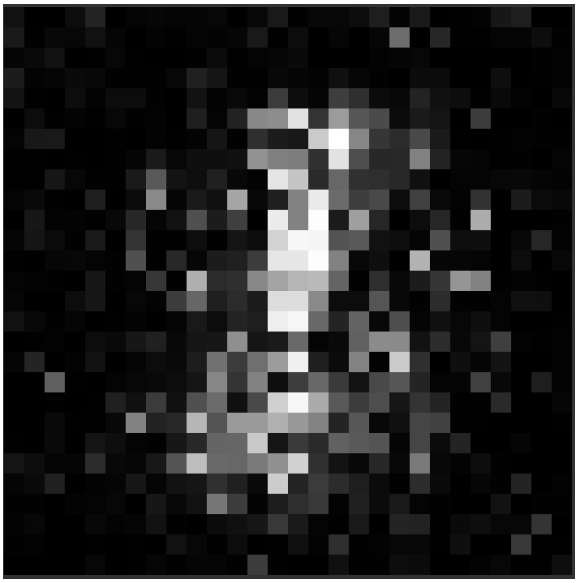

Figure 14: $p(\boldsymbol{v}|\boldsymbol{h} = (0, 0, 0, 0, 0, 0, 0, 0, 1, 0))$

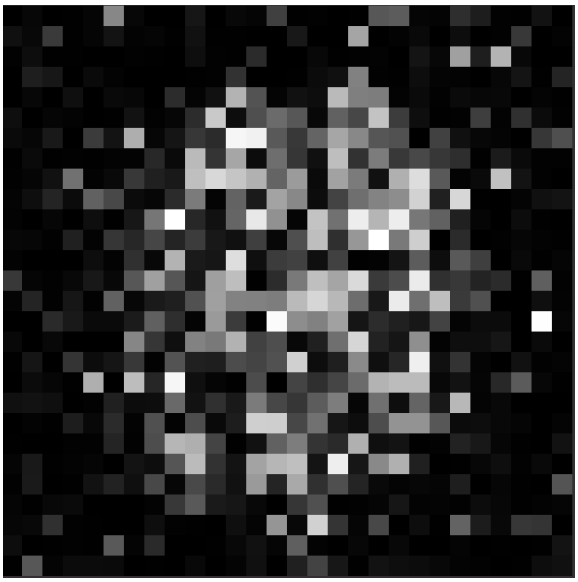

Figure 15: $p(\boldsymbol{v}|\boldsymbol{h} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1))$