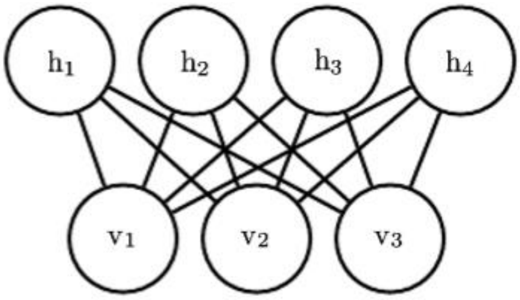
Report: RBM

1. Introduction to Restricted Boltzmann Machines(RBM)

Invented under the name harmonium, restricted Boltzmann machines are some of the most common building blocks of deep probabilistic models.

a) What is RBM

RBMs are undirected probabilistic graphical models containing a layer of observable variables and a single layer of latent variables. RBMs may be stacked (one on top of the other) to form deeper models. It is a bipartite graph, with no connections permitted between any variables in the observed layer or between any units in the latent layer. The following figure shows the structure of the RBM:



b) Binary RBM

Let the observed layer consist of a set of n_v binary random variables which we refer to collectively with the vector **v**. We refer to the latent or hidden layer of n_h binary random variables as **h**.

The restricted Boltzmann machine is an energy-based model with the joint probability distribution specified by its energy function:

$$P(\mathbf{v}=\boldsymbol{v}, \mathbf{h}=\boldsymbol{h}) = \frac{1}{Z}\exp\left(-E(\boldsymbol{v}, \boldsymbol{h})\right).$$

The energy function for an RBM is given by

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h} - \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h},$$

and Z is the normalizing constant known as the partition function:

$$Z = \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \exp\left\{-E(\boldsymbol{v}, \boldsymbol{h})\right\}$$

The partition function Z is intractable which implies that the normalized joint probability distribution P (v) is also intractable to evaluate.

Though P (v) is intractable, the bipartite graph structure of the RBM has the very special property that its conditional distributions P(h | v) and P(v | h) are relatively simple to compute and to sample from.

Regardless of the deriving process, We can express the conditional probability over the hidden layer:

$$p(\mathbf{h} = \underline{1}|\mathbf{v}) = \sigma\left(\mathbf{b} + W^{\mathrm{T}} v_i\right)$$

Also:

$$p(\mathbf{v} = \underline{1}|\mathbf{h}) = \sigma\left(\mathbf{a} + W h_j\right)$$

c) Sampling

Given a visible variable, hidden variables are mutually and conditionally independent. Similarly, given the hidden variables, visible variables are also independent conditionally. Therefore, RBM can sample all visible variables (or all hidden variables) at the same time, by which it can reach a thermal equilibrium fast.

The RBM finds the optimal parameter w (a, b) by maximizing the like-hood function. To get the parameters, we need to approximately calculate p(v) by Gibbs sampling, but it’s not efficient. Thankfully, due to the special structure of the RBM, we have a more efficient and fast way called contrastive divergence (Algorithm CD-k). Using a training sample as the initial value of the visible variable, algorithm CD-k implement Gibbs-sampling by turns without reaching convergence.

2. Explanation of program

a) Training model

I used CD-k algorithm to sample variables, calculate and update parameters. I have tried BBRBM and GBRBM, but they did not make experimental results any better. Therefore, I made the reconstructed variables equal to their conditional probabilities regarding that they are binary variables.

To compute loss, I chose the MSE method which is simple and functional sufficient.

I used tensorflow saver mainly to save and restore the parameters in order to reconstruct images.

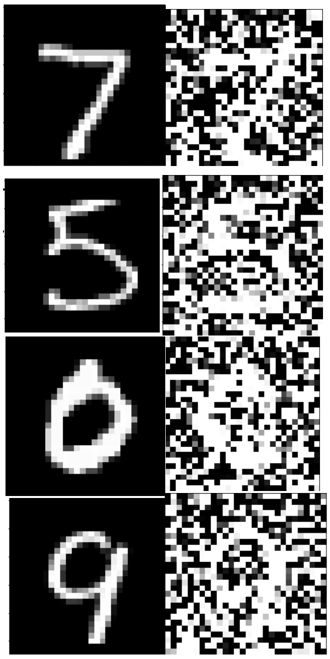
b) Generate images

I used matplotlib.pyplot to reconstruct and show images with reshaping them in size 28*28.

3. Results

a) Results

Some of my experimental results as below: (original image on the left with reconstructed image on the right)



b) Summary

According to the results, we can see that reconstructed images can express the feature of original images by RBM. However, there are still lots of noises in the reconstructed images. I tried Bernoulli sampling to protrude the features, but it did not work. Increasing the epochs can slightly reduce the noises.

We cannot avoid the noises because the differences between images from different labels are obvious. Training and reconstructing images from a single label can reduce noises but cannot eliminate noises because the images though which are from the same label are different. Alternatively, we may need a deep RBM and DBN to gain better-reconstructed images.

.