Report on Project 4

Zhunxuan Wang, 13300180086
School of Mathematical Sciences

December 18, 2017

1 Restricted Boltzmann Machine

1.1 RBM Layer

RBM is an efficient NN structure to deal with tasks regarding dimensionality reduction, classification, regression, etc. It contains a visible layer and a hidden layer, with 2 vectors $\mathbf{b}_\mathrm{v}, \mathbf{b}_\mathrm{h}$ as visible bias and hidden bias, and a transition matrix $\mathbf{M}$, all of which are the parameters of the model [1]. The brief structure of the encoding process is shown as follows

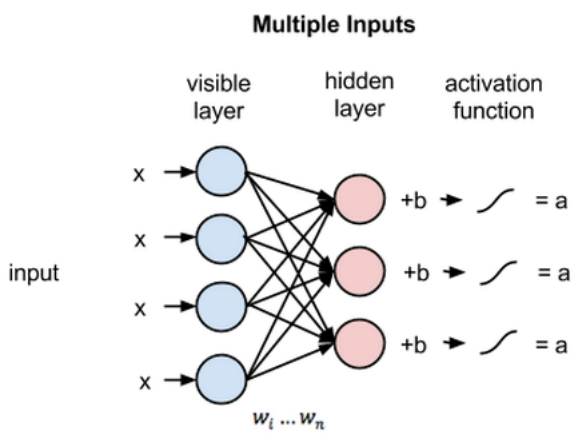

Figure 1: Restricted Boltzmann Machine

The above diagram shows the encoding process from visible states to hidden states, by matrix multiplication with biases followed by activations. The formulas dominating the calculation are expressed as follows

- $\mathbf{x}$, $\mathbf{a}$ is the probabilities of sampling of visible state and hidden state, of which the encoding process is dominated by

$$\mathbf{a} = \sigma\left(\mathbf{x}\mathbf{W} + \mathbf{b}_\mathrm{h}\right)$$

- $\mathbf{v}$, $\mathbf{h}$ is the sampling results of visible state and hidden state, which are distributed as

$$h_i \sim \mathcal{B}\left(a_i\right)$$

where $\mathcal{B}$ is the binomial distribution.

1.2 Reconstruction in RBM

The reconstruction of data in RBM is based on a backward propagation in an unsupervised setting, during which the activation vector is the input and the transition weight matrix $\mathbf{W}$ remains the same (only do transpose), the bias term will be $\mathbf{b}_\mathrm{v}$ which is different from $\mathbf{b}_\mathrm{h}$. The backward process is shown as follows

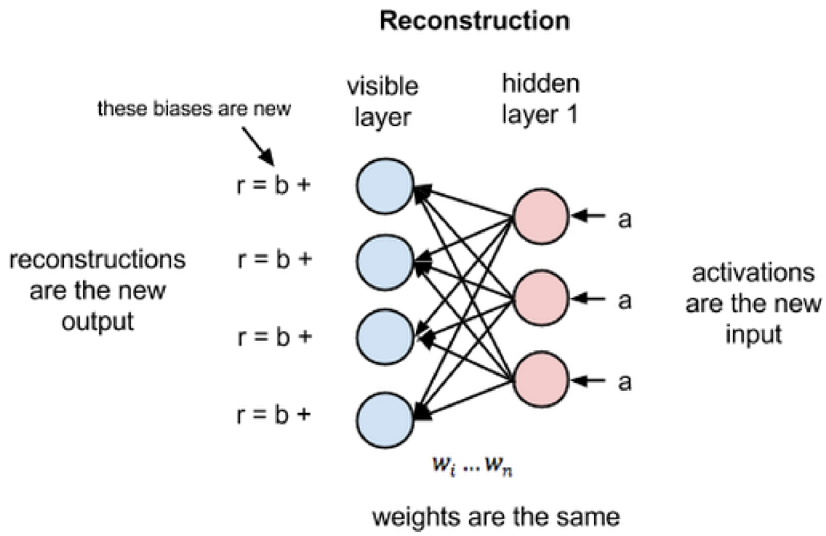

Figure 2: Reconstruction in RBM

As the diagram shows, the reconstruction process from hidden states to visible states, by matrix multiplication with biases followed by activations. The formulas dominating the calculation are expressed as follows

- $\mathbf{r}$ is the probabilities of sampling of reconstruction state, of which the encoding process is dominated by

$$\mathbf{r} = \sigma\left(\mathbf{a}\mathbf{W}^\intercal + \mathbf{b}_\mathrm{v}\right)$$

- The sampling is similar to the last section

$$v_i \sim \mathcal{B}\left(r_i\right).$$

1.3 Unsupervised Parameters Learning

For training the $\mathbf{W}$, $\mathbf{b}_\mathrm{v}$ and $\mathbf{b}_\mathrm{h}$, we decide to do these on the existed model

- Given visible state $\mathbf{x}$, generate $\mathbf{h}$ by $\Pr\left(\mathbf{h} \mid \mathbf{x}\right)$ (the vector of sampling probabilities is $\mathbf{a}$).
- Reconstruct to vector $\mathbf{v}$ by $\Pr\left(\mathbf{v} \mid \mathbf{h}\right)$ (the vector of sampling probabilities is $\mathbf{r}$).
- generate hidden state again $\mathbf{h}'$ by $\Pr\left(\mathbf{h}' \mid \mathbf{v}\right)$ (the vector of sampling probabilities is $\mathbf{a}'$).

We need to make $\mathbf{x}$ and $\mathbf{v}$, $\mathbf{a}$ and $\mathbf{a}'$ closer respectively, so do the transition processes of both. Therefore the gradient of those parameters for minimizing the difference would be

$$\begin{aligned}
\Delta\mathbf{W} &= \mathbf{x} \cdot \mathbf{a} - \mathbf{v} \cdot \mathbf{a}' \\
\Delta\mathbf{b}_\mathrm{v} &= \mathbf{x} - \mathbf{v} \\
\Delta\mathbf{b}_\mathrm{h} &= \mathbf{a} - \mathbf{a}'
\end{aligned}$$

Fixing a considerable learning rate, the training would show good effects.

2 Experimental Results

We take the handwritten digit recognition task `MNIST` to train the RBM model. Fixing learning rate at 0.1, and batch size 1000, we have the following reconstruction results after 100 iterations of training

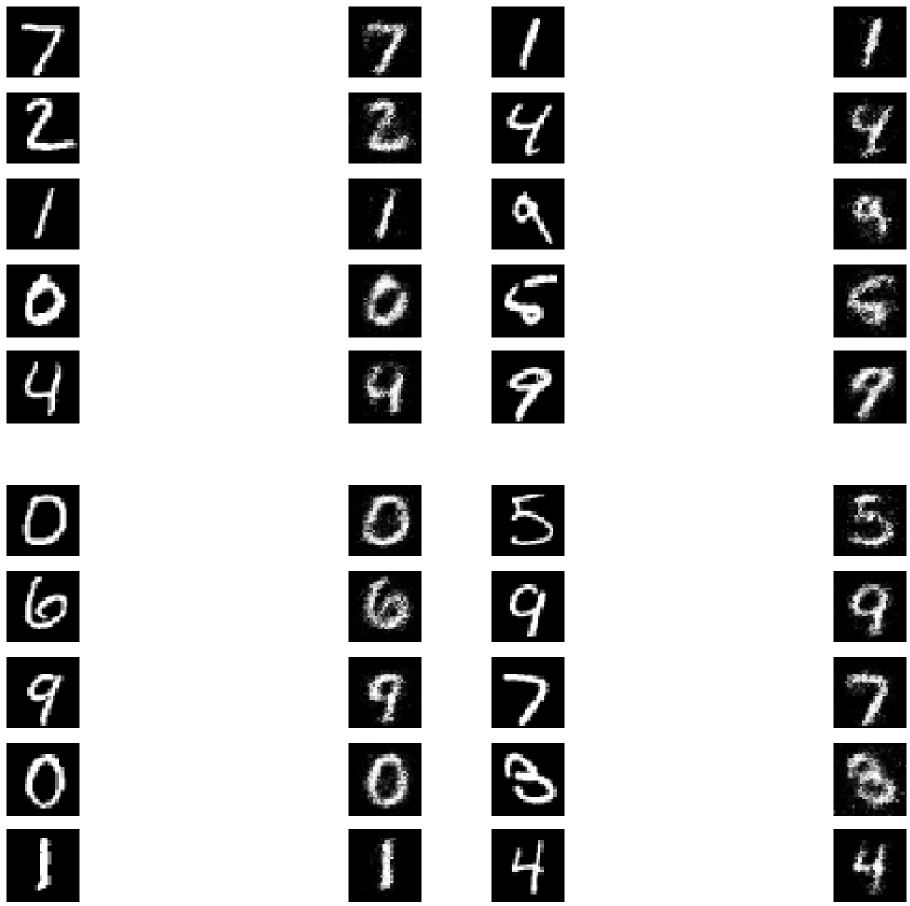

Figure 3: The Reconstruction Result (original left reconstruction right)

As the results show, the shapes and contours of the image after reconstruction are basically the same as that of the originals, which means that the information (feature) extraction might be efficient. However, we can also see that the image after reconstruction becomes blurred, which means there may be loss of information.

We take the mean square

$$C = \frac{1}{N} \sum_{i=1}^{Nd} \|\mathbf{x}_i - \mathbf{r}_i\|_2^2$$

as the observed cost function. The final cost is 0.0262130.

References

[1] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.