

## I. Pencil-and-Paper

Suppose  $(V, H)$  denote the visible and hidden random variable which takes values  $(v \in \{0, 1\}^m, h \in \{0, 1\}^n)$ . And the joint probability is  $p(v, h; \theta) = \frac{1}{Z} e^{-E(v, h; \theta)}$ , where  $E$  is the energy function:

$$\begin{aligned} E(v, h; \theta) &= - \sum_{i=1}^n \sum_{j=1}^m w_{ij} h_i v_j - \sum_{j=1}^m b_j v_j - \sum_{i=1}^n c_i h_i \\ &= -(\mathbf{v}^T \mathbf{W} \mathbf{h} + \mathbf{v}^T \mathbf{b} + \mathbf{h}^T \mathbf{c}) \end{aligned} \quad (1)$$

where  $Z = \sum_v \sum_h e^{-E(v, h; \theta)}$  and  $\theta = (W, b, c)$ .

1. Find  $p(v|h, \theta)$  and  $\mathbb{E}(v|h, \theta)$ .

From the structure of RBM, it means that the hidden variables are independent given the state of the visible variables and the visible variables are independent given the state of the hidden variables. In this way,

$$\begin{aligned} \underline{p(v|h, \theta)} &= \frac{p(v, h|\theta)}{p(h|\theta)} \\ &= \frac{\prod_i^n \prod_j^m p(v_j, h_i|\theta)}{\sum_v \prod_i^n \prod_j^m p(v_j, h_i|\theta)} \end{aligned} \quad (2)$$

So, we also have:

$$\mathbb{E}(v|h, \theta) = \text{sigmoid}(W^T h + b) \quad (3)$$

(The proof will be shown later)

For  $p(v_j|h)$ ,

$$p(v_j|h) = \frac{p(v_j, h)}{p(h)} = \frac{p(v_j, h)}{p(v_j = 0, h) + p(v_j = 1, h)} \quad (4)$$

Since that  $p(v_j = 0, h) = \frac{1}{Z} e^{-E(v_j=0, h; \theta)}$  and  $p(v_j = 1, h) = \frac{1}{Z} e^{-E(v_j=1, h; \theta)}$ . From equation (1), we can think  $E(v, h; \theta) = v_j \alpha_j + \beta$ , where  $\beta$  doesn't contain  $v_j$ .

In this way, we have:

$$\begin{aligned} p(v_j|h) &= \frac{p(v_j, h)}{p(v_j = 0, h) + p(v_j = 1, h)} \\ &= \frac{\exp(-v_j \alpha_j - \beta)}{\exp(-\beta) + \exp(-\alpha_j - \beta)} \\ &= \frac{\exp(-v_j \alpha_j)}{1 + \exp(-\alpha_j)} \end{aligned} \quad (5)$$

So, we have

$$p(v_j = 1|h) = \frac{\exp(-\alpha_j)}{1 + \exp(-\alpha_j)} = \frac{1}{1 + \exp(\alpha_j)} \quad (6)$$

and

$$p(v_j = 0|h) = \frac{1}{1 + \exp(-\alpha_j)} = 1 - \frac{1}{1 + \exp(\alpha_j)} \quad (7)$$

From equation (1), we can see that  $\alpha_j = -\sum_{i=1}^n w_{ij}h_i - b_j$ , so we can see that:

$$\underline{p(v_j = 1|h) = \text{sigmoid}(-\alpha_j) = \text{sigmoid}(\sum_{i=1}^n w_{ij}h_i + b_j)} \quad (8)$$

and

$$\underline{p(v_j = 0|h) = 1 - \text{sigmoid}(-\alpha_j) = 1 - \text{sigmoid}(\sum_{i=1}^n w_{ij}h_i + b_j)} \quad (9)$$

From equation (8) and (9), we can see that:

$$\mathbb{E}(v_j|h, \theta) = 1 \cdot p(v_j = 1|h, \theta) + 0 \cdot p(v_j = 0|h, \theta) = \text{sigmoid}\left(\sum_{i=1}^n w_{ij}h_i + b_j\right) \quad (10)$$

In the vector format:

$$\underline{\mathbb{E}(v|h, \theta) = \text{sigmoid}(W^T h + b)} \quad (11)$$

## 2. Find $p(h|v, \theta)$ and $\mathbb{E}(h|v, \theta)$ .

Similar to the procedures in 1, since that the hidden variables are independent given the state of the visible variables. In this way,

$$\begin{aligned} \underline{p(h|v, \theta)} &= \frac{p(v, h|\theta)}{p(v|\theta)} \\ &= \frac{\prod_i^n \prod_j^m p(v_j, h_i|\theta)}{\sum_h \prod_i^n \prod_j^m p(v_j, h_i|\theta)} \end{aligned} \quad (12)$$

So, we also have:

$$\mathbb{E}(h|v, \theta) = \text{sigmoid}(Wv + c) \quad (13)$$

For  $p(h_i|v)$ ,

$$p(h_i|v) = \frac{p(h_i, v)}{p(v)} = \frac{p(h_i, v)}{p(h_i = 0, v) + p(h_i = 1, v)} \quad (14)$$

Since that  $p(h_i = 0, v) = \frac{1}{Z} e^{-E(v, h_i=0; \theta)}$  and  $p(h_i = 1, v) = \frac{1}{Z} e^{-E(v, h_i=1; \theta)}$ . From equation (1), we can think  $E(v, h; \theta) = h_i \gamma_i + \beta'$ , where  $\beta'$  doesn't contain  $h_i$ . In this way, we have:

$$\begin{aligned} p(h_i|v) &= \frac{p(h_i, v)}{p(h_i = 0, v) + p(h_i = 1, v)} \\ &= \frac{\exp(-h_i \gamma_i - \beta')}{\exp(-\beta') + \exp(-\gamma_i - \beta')} \\ &= \frac{\exp(-h_i \gamma_i)}{1 + \exp(-\gamma_i)} \end{aligned} \quad (15)$$

So, we have

$$p(h_i = 1|v) = \frac{\exp(-\gamma_i)}{1 + \exp(-\gamma_i)} = \frac{1}{1 + \exp(\gamma_i)} \quad (16)$$

and

$$p(h_i = 0|v) = \frac{1}{1 + \exp(-\gamma_i)} = 1 - \frac{1}{1 + \exp(\gamma_i)} \quad (17)$$

From equation (1), we can see that  $\gamma_i = -\sum_{j=1}^m w_{ij}v_j - c_i$ , so we can see that:

$$\underline{p(h_i = 1|v) = \text{sigmoid}(-\gamma_i) = \text{sigmoid}(\sum_{j=1}^m w_{ij}v_j + c_i)} \quad (18)$$

and

$$\underline{p(h_i = 0|v) = 1 - \text{sigmoid}(-\gamma_i) = 1 - \text{sigmoid}(\sum_{j=1}^m w_{ij}v_j + c_i)} \quad (19)$$

From equation (18) and (19), we can see that:

$$\mathbb{E}(h_i|v, \theta) = 1 \cdot p(h_i = 1|v, \theta) + 0 \cdot p(h_i = 0|v, \theta) = \text{sigmoid}\left(\sum_{j=1}^m w_{ij}v_j + c_i\right) \quad (20)$$

In the vector format:

$$\underline{\mathbb{E}(h|v, \theta) = \text{sigmoid}(Wv + c)} \quad (21)$$

3. Compute  $\frac{\partial \mathcal{L}(D|\theta)}{\partial W_{ij}}$ ,  $\frac{\partial \mathcal{L}(D|\theta)}{\partial b_j}$  and  $\frac{\partial \mathcal{L}(D|\theta)}{\partial c_i}$

Now, suppose the given dataset is  $D = \{v_1, v_2, \dots, v_N\}$ , then the log-likelihood can be calculated as:

$$\mathcal{L}(D|\theta) = \sum_{t=1}^N \log(p(v_t|\theta)) \quad (22)$$

Since that

$$p(v_t|\theta) = \sum_{h_t} p(v_t, h_t|\theta) = \frac{1}{Z} \sum_{h_t} \exp(-E(v_t, h_t; \theta)) \quad (23)$$

where  $Z = \sum_{v_t} \sum_{h_t} \exp(-E(v_t, h_t; \theta))$

we can get:

$$\begin{aligned} \log(p(v_t|\theta)) &= \log\left(\frac{1}{Z} \sum_{h_t} \exp(-E(v_t, h_t; \theta))\right) \\ &= \log\left(\sum_{h_t} \exp(-E(v_t, h_t; \theta))\right) - \log\left(\sum_{v_t} \sum_{h_t} \exp(-E(v_t, h_t; \theta))\right) \end{aligned} \quad (24)$$

So, to calculate the derivation, we can get:

$$\begin{aligned}
\frac{\partial \log(p(v_t|\theta))}{\partial \theta} &= \frac{\partial}{\partial \theta} \log\left(\sum_{h_t} \exp(-E(v_t, h_t; \theta))\right) - \frac{\partial}{\partial \theta} \log\left(\sum_{v_t} \sum_{h_t} \exp(-E(v_t, h_t; \theta))\right) \\
&= -\frac{1}{\sum_{h_t} e^{-E}} \sum_{h_t} e^{-E} \cdot \frac{\partial E}{\partial \theta} + \frac{1}{\sum_{v_t} \sum_{h_t} e^{-E}} \sum_{v_t} \sum_{h_t} e^{-E} \cdot \frac{\partial E}{\partial \theta} \\
&= -\frac{\sum_{h_t} e^{-E} \cdot \frac{\partial E}{\partial \theta}}{Z \cdot p(v_t|\theta)} + \sum_{v_t} \sum_{h_t} \frac{e^{-E}}{Z} \cdot \frac{\partial E}{\partial \theta} \\
&= -\sum_{h_t} p(h_t|v_t, \theta) \cdot \frac{\partial E}{\partial \theta} + \sum_{v_t} \sum_{h_t} p(v_t, h_t|\theta) \cdot \frac{\partial E}{\partial \theta} \\
&= -\sum_h p(h|v) \cdot \frac{\partial E(v, h)}{\partial \theta} + \sum_v \sum_h p(v, h) \cdot \frac{\partial E(v, h)}{\partial \theta}
\end{aligned} \tag{25}$$

From equation (25), we can get:

$$\begin{aligned}
\frac{\partial \mathcal{L}(D|\theta)}{\partial W_{ij}} &= -\sum_h p(h|v) \cdot \frac{\partial E(v, h)}{\partial w_{ij}} + \sum_v \sum_h p(v, h) \cdot \frac{\partial E(v, h)}{\partial w_{ij}} \\
&= \sum_h h_i v_j \cdot p(h|v) - \sum_h \sum_v h_i v_j \cdot p(v, h) \\
&= \underline{p(h_i = 1|v) \cdot v_j - \sum_v p(v) \cdot p(h_i = 1|v) \cdot v_j} \\
&= \underline{p(h_i = 1|v) \cdot v_j - \mathbf{E}[v_j h_i]}
\end{aligned} \tag{26}$$

In the similar way, we can have:

$$\begin{aligned}
\frac{\partial \mathcal{L}(D|\theta)}{\partial b_j} &= \sum_h v_j \cdot p(h|v) - \sum_h \sum_v v_j \cdot p(v, h) \\
&= \underline{v_j - \sum_v p(v) \cdot v_j} \\
&= \underline{v_j - \mathbf{E}[v_j]}
\end{aligned} \tag{27}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}(D|\theta)}{\partial c_i} &= \sum_h h_i \cdot p(h|v) - \sum_h \sum_v h_i \cdot p(v, h) \\
&= \underline{p(h_i = 1|v) - \sum_v p(v) \cdot p(h_i = 1|v)} \\
&= \underline{p(h_i = 1|v) - \mathbf{E}[h_i]}
\end{aligned} \tag{28}$$

#### 4. Contrastive divergence

From equation (26), (27) and (28), we can find that  $p(v, h)$  is actually a computationally intractable term. To solve this problem, we should consider 1-step contrastive divergence to solve this problem.

Considering Hinton approximation:

$$\mathbf{E}[v_j h_i] \simeq \mathbf{E}[v_j|h] \mathbf{E}[h_i|v] \tag{29}$$

And the k-step contrastive divergence is:

$$CD_k(\theta, v^{(0)}) = - \sum_h p(h|v^{(0)}) \frac{\partial E(v^{(0)}, h)}{\partial \theta} + \sum_h p(h|v^{(k)}) \frac{\partial E(v^{(k)}, h)}{\partial \theta} \quad (30)$$

First, choose one visible data  $v^{(0)}$  from the given training set, sample hidden nodes  $h^{(0)}$  according to  $p(h|v^{(0)})$ . Then, calculate the probability  $p(v|h^{(0)})$  using the sampled hidden nodes, and then reconstruct the visible nodes  $v^{(1)}$  according to  $p(v|h^{(0)})$ . Finally, calculate the probability of  $p(h^{(1)}|v^{(1)})$ . Using these result to approximate and calculate the update rules for  $W$ ,  $b$  and  $c$ .

**Note:** The above method is for 1-step CD sampling, if needed, this procedure can be repeated k times.

## II. Code-from-Scratch

### 1. Methods

The algorithm for RBM is relatively simple: First initialize all the parameters including  $W$ ,  $b$  and  $c$ . Then given any visible sample  $v_0$ , first sample  $h_0$  according to  $p(h_0|v_0)$ . Then, reconstruct  $v_1$  according to  $p(v_1|h_0)$ . Finally, re-sample  $h_1$  according to  $p(h_1|v_1)$ . Finally, update all the parameters including  $W$ ,  $b$  and  $c$ . Repeat this procedure enough times, then you can get the desired result. The detailed structure of the code is shown in Figure 1.

In the class `RBM()`, there are several functions such as `train()`, `_initialize()`, `update_h()`, `update_v()`, `gradient()` and `sigmoid()`. Function `_initialize()` is used to initialize all parameters. Then function `train()` is used to train the RBM model, it will call the functions `update_h()`, `update_v()`, `gradient()` and `sigmoid()` to update all parameters. In this code, RBM can be trained through single input data, or through mini-batch learning.

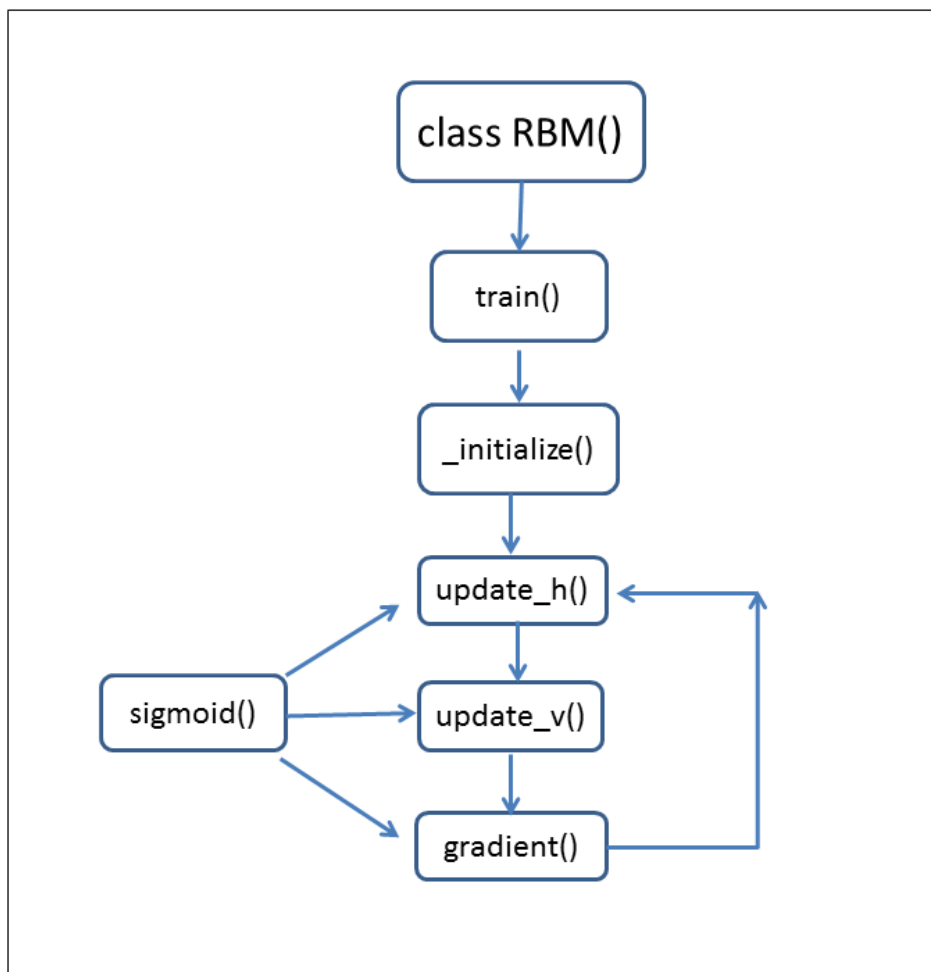


Figure 1: Overall Structure for RBM

## 2. Results

**Note:** In the following result section, due to different initialization, the final result may be different for each running.

In this part, using the code illustrated in Figure 1 and updating RBM model using single input image, the first 64 of the 200 learned filters are shown in Figure 2.

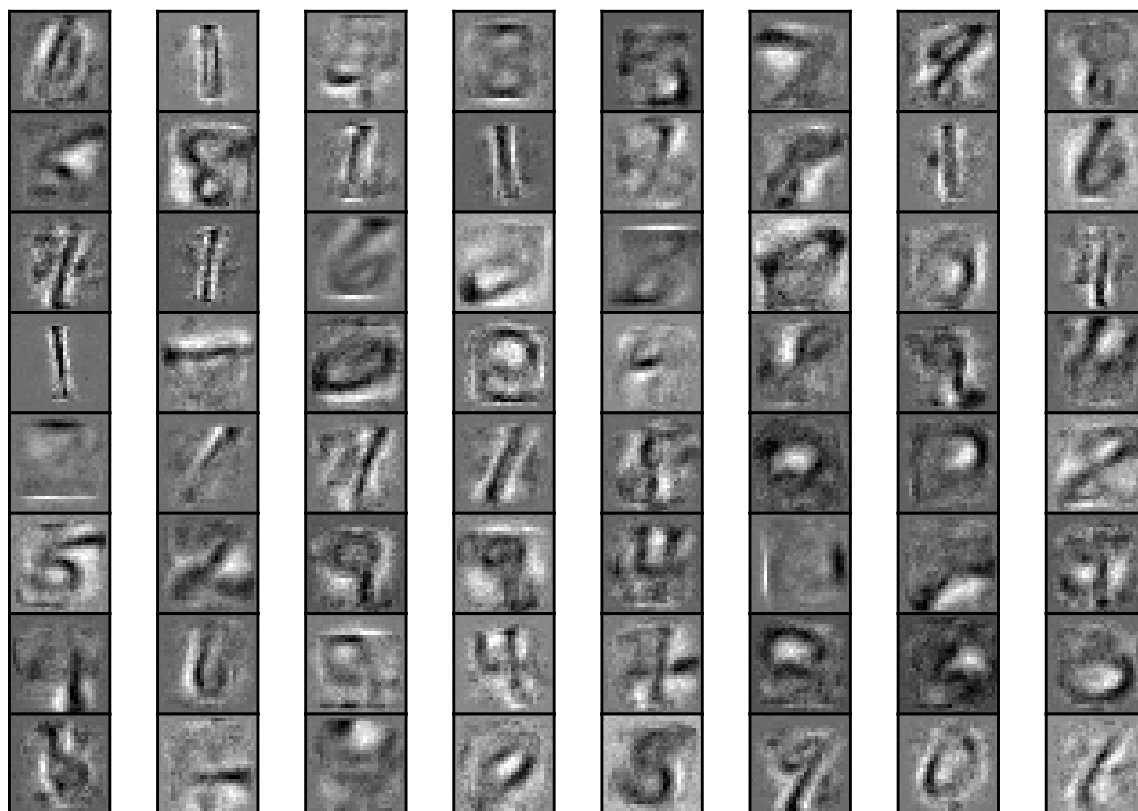


Figure 2: Learned Filter

In Figure 2, all the learned filters shown some interesting phenomenon, most of them shown some clear digits.

As a comparison, we also tried using 10-mini-batch learning method, the first 64 learned filters are shown in Figure 3. As you can see, using mini-batch, the learned filter is not as clear as shown in Figure 2.

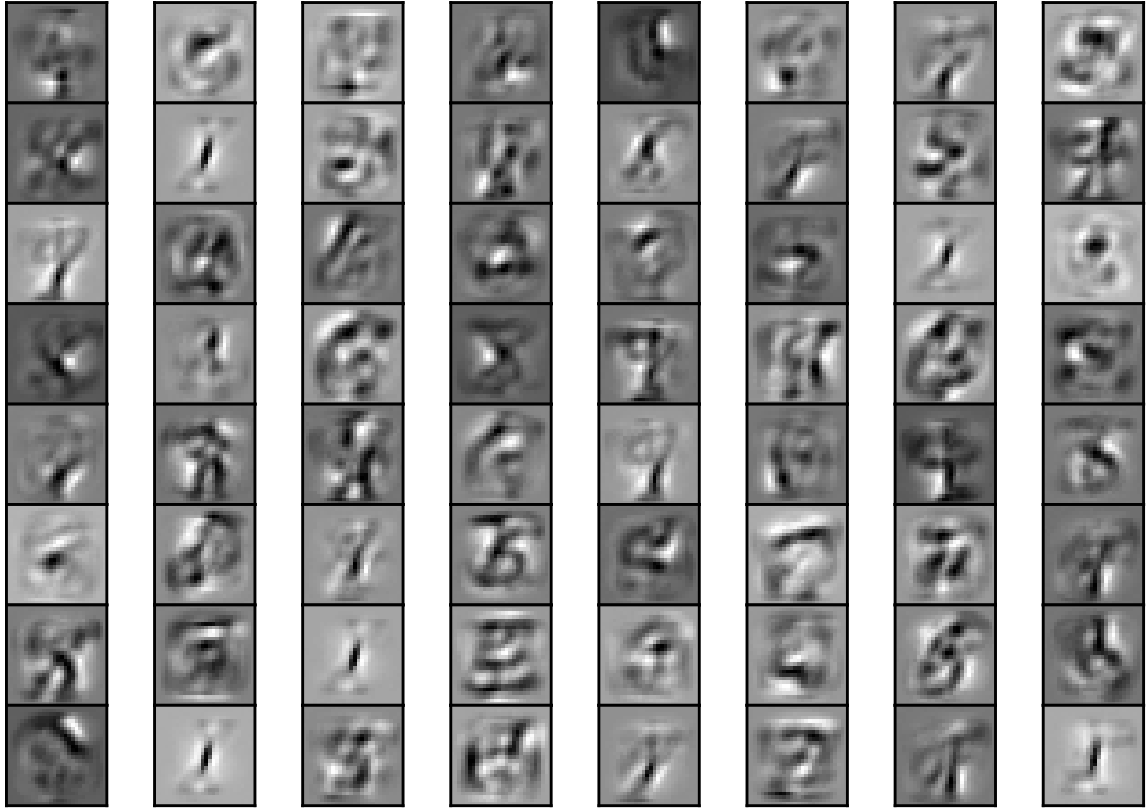


Figure 3: Learned Filter



### III. TensorFlow

#### 1. Methods

(1) In this part, the RBM model is re-written in TensorFlow. The overall structure is very similar. However, considering the characteristic of TensorFlow, there aren't so many functions as shown in Figure 1. Instead, the overall structure is shown in Figure 4.

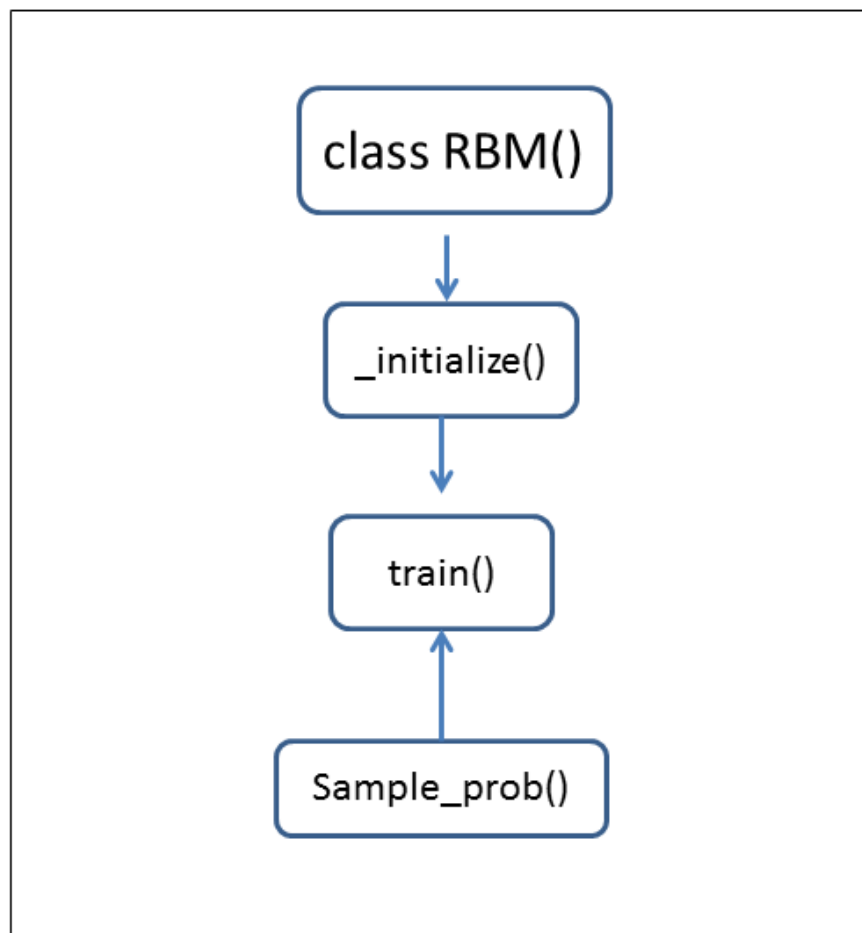


Figure 4: Overall Structure for TensorFlow RBM

In RBM part, we use function `_initialize()` to initialize all parameters and create Variables and placeholders. Then, using function `train()` to train the model. In this process, function `sample_prob()` is used to sample the Bernoulli distribution.

More specifically, `tf.nn.sigmoid()` is used to compute sigmoid probability, `tf.truncated_normal()` is used to initialize parameters and `assign_add()` is used to update parameters. In addition, `tf.nn.relu()` and `tf.random_uniform()` is used to create function `sample_prob()`.

(2) For the 10-way multi-class logistic regression part, we build a fully-connected neural network. The input can have 784 or 200 nodes, the output out has 10 nodes using a `softmax()` function. More

specifically, in the part, we use `tflearn.input_data()`, `tflearn.fully_connected()`, `tflearn.regression()` and `tflearn.DNN()` to train the model.

(3) For the stacked RBM, in the first RBM, there is 784 visible nodes and 500 hidden nodes, so the total number of weights is:

$$784 \times 500 + 784 + 500 = 393284 \quad (31)$$

In the second RBM, there are 500 visible nodes and 200 hidden nodes, the total number of weights is:

$$500 \times 200 + 500 + 200 = 100700 \quad (32)$$

So the total weights for the whole stacked RBM is:

$$393284 + 100700 = 493984 \quad (33)$$

**Note:** in the above calculation, the bias term  $b$  and  $c$  for each RBM are also thought to be part of the weights.

## 2. Results

(1) In this part, using the raw image, image processed by single RBM, image processed by PCA and image processed by stacked RBM, we train a 1 layer neural network and calculate the training accuracy and testing accuracy. The result is shown in Table 1.

Table 1: Training and testing accuracy for different models

Methods	Training Accuracy	Testing Accuracy
Raw Image	92.99%	92.59%
RBM	94.99%	94.93%
PCA	92.32%	92.07%
Stacked RBM	96.35%	96.24%

From Table 1, we can see that, the result from PCA is similar to the result from raw image. But using RBM and stacked RBM, the training accuracy and testing accuracy both get improved.

(2) Training and testing classification confusion matrix. (See next page)

(a). For the raw image, the training and testing classification confusion matrix are shown in Figure 5 and Figure 6.

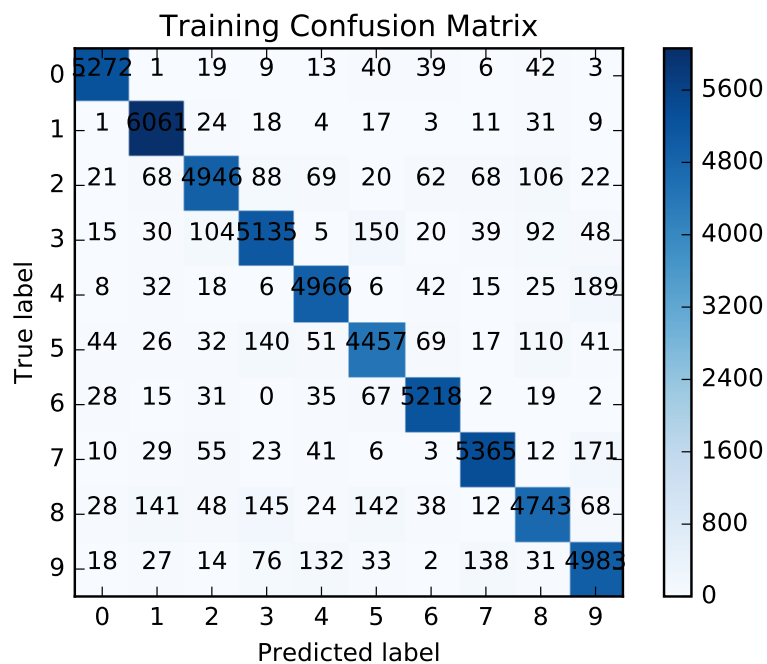


Figure 5: Training Set Confusion Matrix Using Raw Image

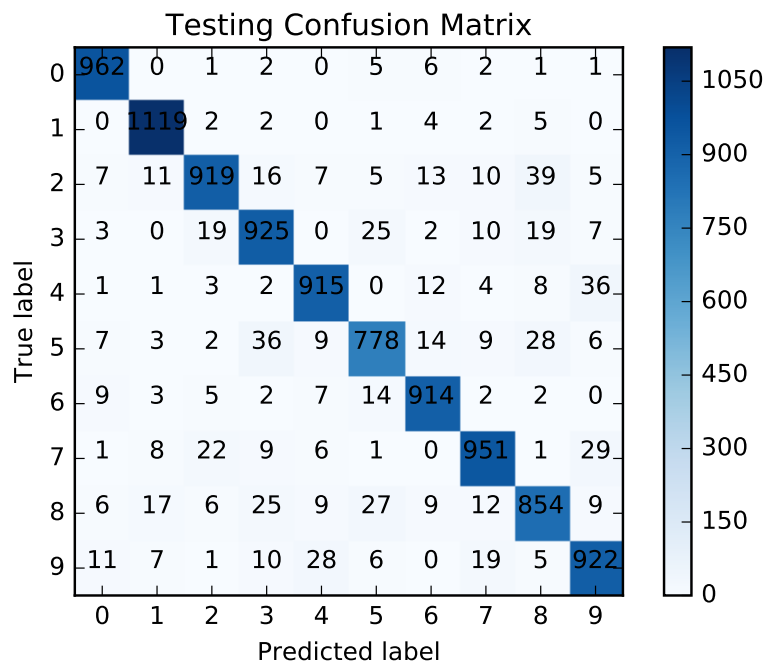


Figure 6: Test Set Confusion Matrix Using Raw Image

(b). For the image processed by RBM, the training and testing classification confusion matrix are shown in Figure 7 and Figure 8.

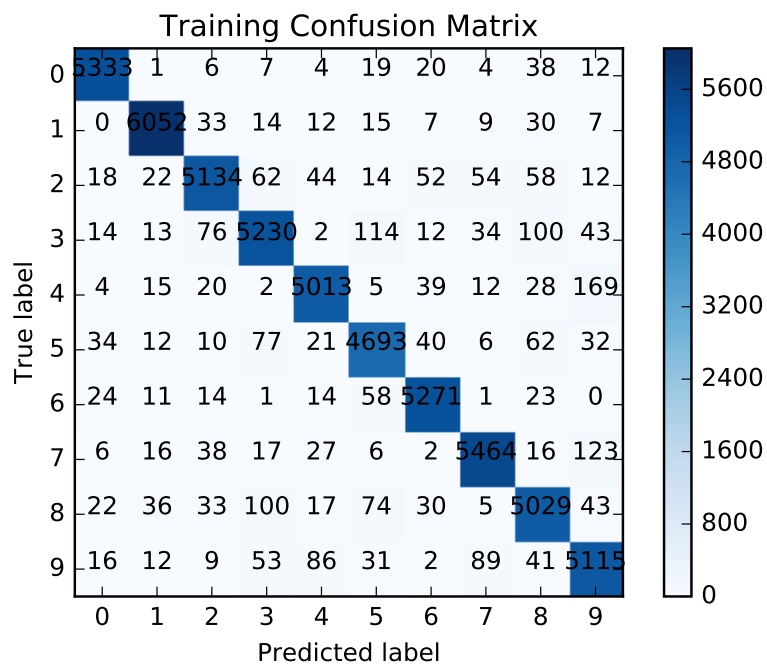


Figure 7: Training Set Confusion Matrix Using RBM

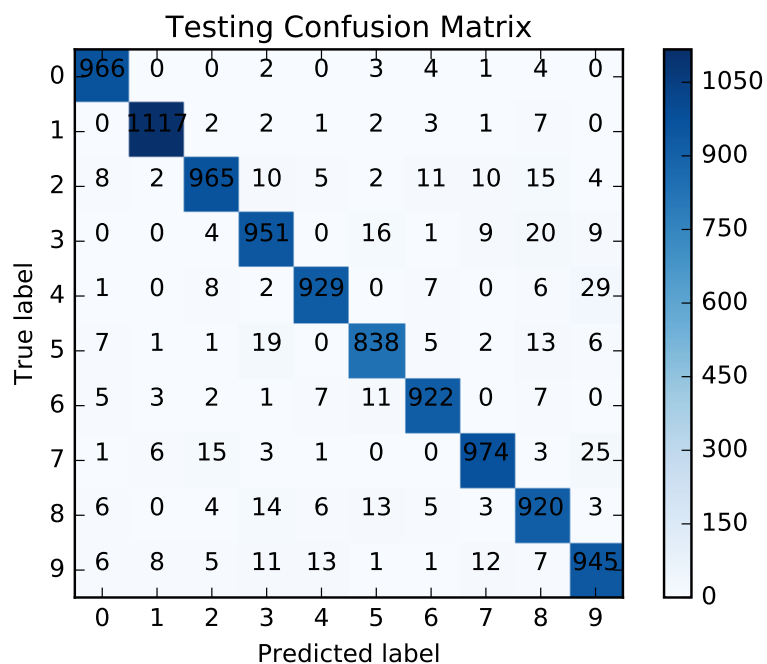


Figure 8: Test Set Confusion Matrix Using RBM

(c). For the image processed by PCA, the training and testing classification confusion matrix are shown in Figure 9 and Figure 10. (The PCA part is implemented using scikit-learn PCA)

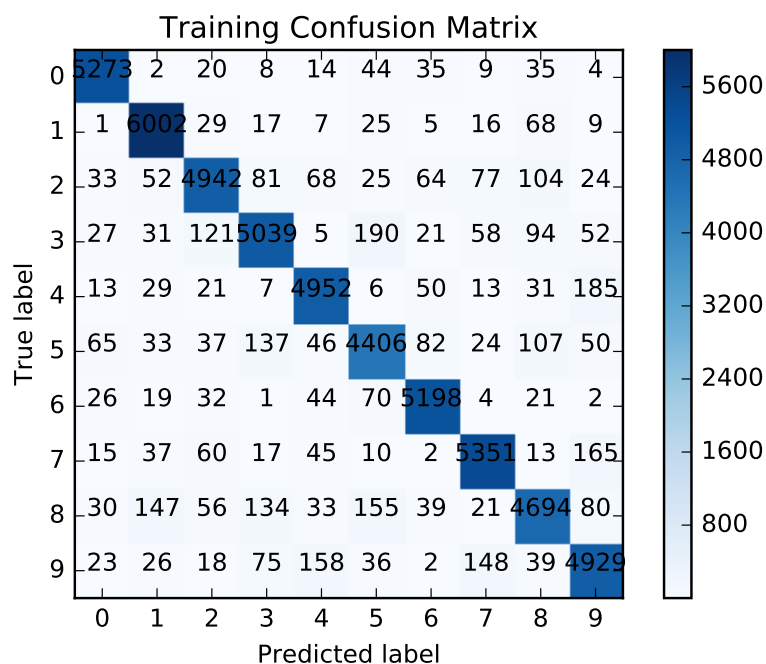


Figure 9: Training Set Confusion Matrix Using PCA

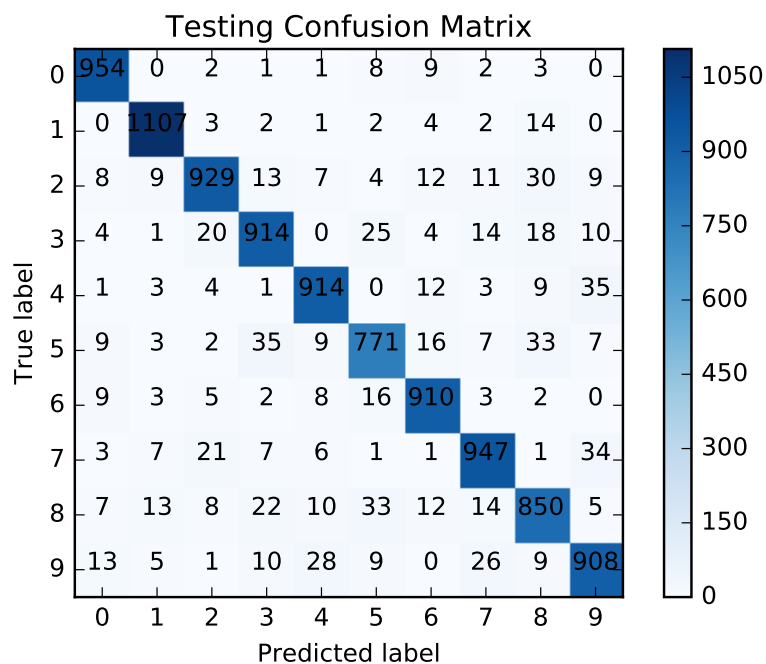


Figure 10: Test Set Confusion Matrix Using PCA

(d). For the image processed by stacked RBM, the training and testing classification confusion matrix are shown in Figure 11 and Figure 12.

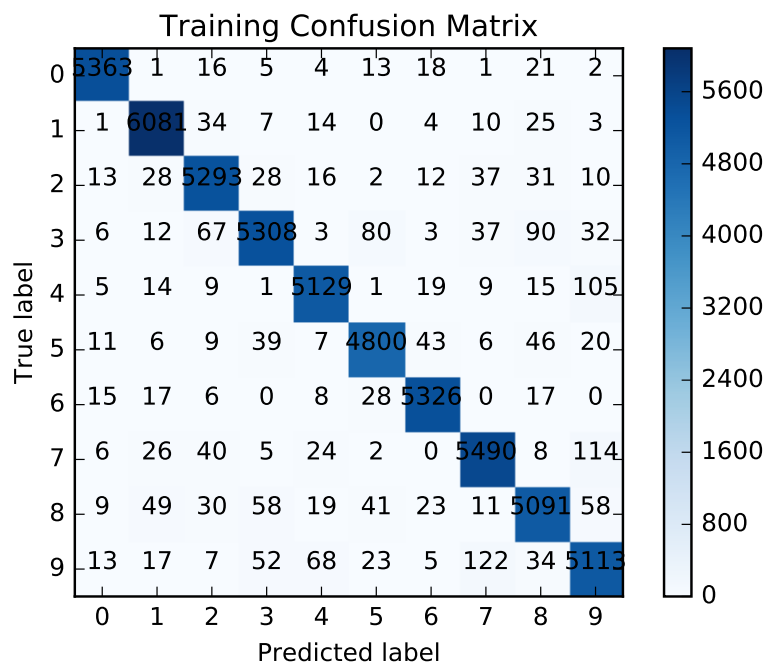


Figure 11: Training Set Confusion Matrix Using Stacked RBM

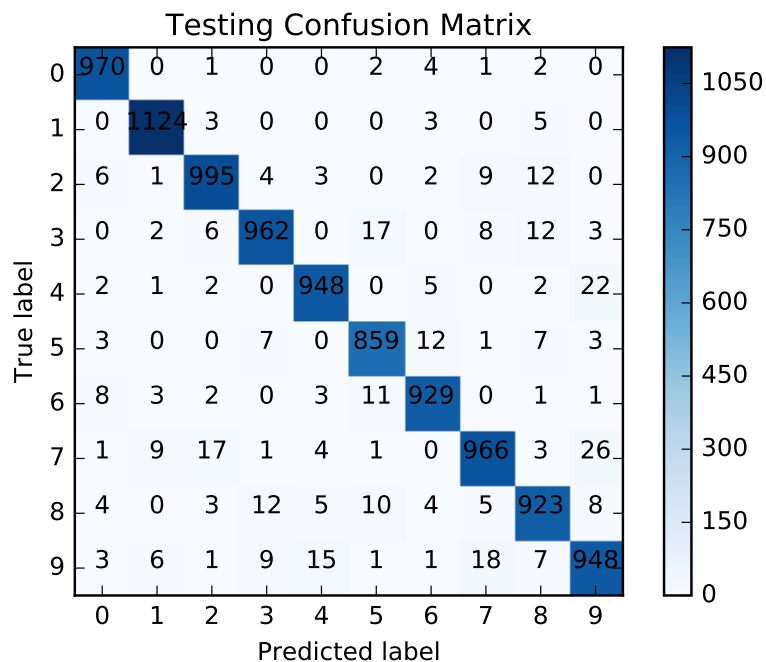


Figure 12: Test Set Confusion Matrix Using Stacked RBM