ECE 544NA HW3

Jiaqi Mu jiaqimu2

Department of Electrical and Computer Engineering

November 2, 2016

1 Pencil-and-Paper

In this portion of the assignment, you will derive the update equations for a binary-binary RBM. Let (V, H) denote the visible and hidden random variable which takes values ($v \in \{0,1\}^m$, $h \in \{0,1\}^n$). Then the joint probabolity distribution is described by the distribution $p(v, h; \theta) = \frac{1}{Z} e^{-E(v,h;\theta)}$, where E is the following energy function,

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

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

- Find $p(v|h, \theta)$ and $\mathbb{E}(v|h, \theta)$. Show that $p(v_j|h)$ has the form $sigmoid\left(\sum_{i=1}^{n} w_{ij} h_i + b_j\right)$.

  Proof. Given the joint distribution, one knows the conditional distribution as,

$$\begin{aligned} p(v|h, \theta) &= \frac{p(v, h; \theta)}{p(h; \theta)} \\ &= \frac{e^{-E(v,h;\theta)}}{\sum_u e^{-E(u,h;\theta)}} \\ &= \frac{e^{v^T W h + v^T b + h^T c}}{\sum_u e^{u^T W h + u^T b + h^T c}} \\ &= \frac{1}{\sum_u e^{(u-v)^T W h + (u-v)^T b}} \\ &= \frac{1}{\sum_u e^{(u-v)^T (W h + b)}} \\ &= \frac{1}{\prod_{j=1}^{m} \left(1 + e^{-(2v_j - 1)(W_j^{\mathrm{T}} h + b_j)}\right)}, \end{aligned}$$

where W_j is the j-th row of W. Specially, we can see $p(v|h,\theta)$ can be factorized as,

$$p(v|h,\theta) = \prod_{j=1}^{m} \frac{1}{1+e^{-(2v_j-1)(W_j^{\mathrm{T}}h+b_j}}.$$

Thus, given h, v_1,...,v_m are conditionally independent, thus,

$$p(v_j|h,\theta) = \frac{1}{1+e^{-(2v_j-1)(W_j^{\mathrm{T}}h+b_j}} = \begin{cases} \frac{1}{1+e^{-(\sum_{i=1}^{n} w_{ij}h_i+b_j)}} & v_j = 1 \\ \frac{1}{1+e^{\sum_{i=1}^{n} w_{ij}h_i+b_j}} & v_j = 0 \end{cases}.$$

The conditional expectation is thus given as,

$$\mathbb{E}(v|h;\theta) = \left(E(v_1|h;\theta), ..., \mathbb{E}(v_n|h;\theta)\right),$$

where $\mathbb{E}(v_j|h;\theta)$ is given as,

$$\mathbb{E}(v_j|h;\theta) = p(v_j = 1|h;\theta) = sigmoid\left(\sum_{i=1}^{n} w_{ij}h_i + b_j\right).$$

□

- Find $p(h|v;\theta)$ and $E(h|v;\theta)$. Show that $p(h_i|v)$ has the form $sigmoid(\sum_{j=1}^{m} w_{ij}v_j + c_i)$.

  Proof. In $E(v,h;\theta)$, v and h are symmetric, thus following the steps in part 1 we know,

  $$p(h|v;\theta) = \prod_{i=1}^{n} \frac{1}{1+e^{-2(h_i-1)(v^{\mathrm{T}}W_i+c_i)}},$$

  where W_i is the i-th column of W. The conditional distribution and the conditional expectation are,

  $$\mathbb{E}(h_i;v;\theta) = sigmoid\left(\sum_{j=1}^{m} w_{ij}v_j + c_i\right)$$
  $$P(h_i|v;\theta) = \begin{cases} \frac{1}{1+e^{-(\sum_{j=1}^{m} w_{ij}v_j+c_i)}} & h_i = 1 \\ \frac{1}{1+e^{\sum_{j=1}^{m} w_{ij}v_j+c_i}} & h_i = 0 \end{cases}$$

  □

- Now you are given a dataset of $D = \{v_1, ..., v_N\}$. The log-likelihood of the dataset can be computed as,

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

  Compute $\frac{\partial \mathcal{L}(D|\theta)}{\partial \theta}$, and express it in terms of $p(h|v)$, $p(h,v)$ and $\frac{\partial E(v,h)}{\partial \theta}$.

Proof. First we compute $p(v;\theta)$. This can be computed via $p(v,h;\theta)$, i.e.,

$$p(v;\theta) = \sum_h p(v,h) = \frac{1}{Z} e^{-F(v)},$$

where $F(v)$ is defined to be $F(v) = -\log \sum_h e^{-E(v,h;\theta)}$, and Z is the partition function defined as $Z = \sum_v e^{-F(v)}$, and can be simplified as,

$$\begin{aligned} F(v) &= -\log \sum_h e^{(v^{\mathrm{T}}Wh + v^{\mathrm{T}}b + h^{\mathrm{T}}c)} \\ &= -\log e^{v^{\mathrm{T}}b} \sum_h e^{(W^{\mathrm{T}}v + c)^{\mathrm{T}}h} \\ &= -\log e^{v^{\mathrm{T}}b} \prod_{i=1}^{n} (1 + e^{v^{\mathrm{T}}W_i + c_i}) \\ &= -v^{\mathrm{T}}b - \sum_{i=1}^{n} \log(1 + e^{v^{\mathrm{T}}W_i + c_i}). \end{aligned}$$

Based on the chain rule, the answer is given as follow,

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

where,

$$\begin{aligned} \frac{\partial \log p(v|\theta)}{\partial \theta} &= \frac{\partial}{\partial \theta} \left(-F(v) - \log Z\right) \\ &= -\frac{\partial F(v)}{\partial \theta} - \frac{1}{Z} \sum_u \frac{\partial e^{-F(u)}}{\partial \theta} \\ &= -\frac{\partial F(v)}{\partial \theta} + \sum_u \frac{e^{-F(u)}}{Z} \frac{\partial F(u)}{\partial \theta} \\ &= -\frac{\partial F(v)}{\partial \theta} + \sum_u p(u) \frac{\partial F(u)}{\partial \theta}. \end{aligned}$$

As required, $\frac{\partial \log p(v|\theta)}{\partial \theta}$ can also be written as,

$$\frac{\partial \log p(D|\theta)}{\partial \theta} = -\sum_{t=1}^{N} \sum_h p(h|v_t;\theta) \frac{\partial E(v_t,h;\theta)}{\partial \theta} + N \sum_v \sum_h p(v,h;\theta) \frac{\partial E(v,h)}{\partial \theta}.$$

□

- Use the result in (2) compute $\frac{\partial \mathcal{L}(D|\theta)}{\partial W_{ij}}$, $\frac{\partial \mathcal{L}(D|\theta)}{\partial b_j}$ and $\frac{\partial \mathcal{L}(D|\theta)}{c_i}$.

Proof. Given $F(v)$, we know,

$$\begin{aligned}
\frac{\partial F(v;\theta)}{\partial W_{ij}} &= -\sum_{i=1}^{n} \frac{1}{1+e^{v^T W_i + c_i}} \frac{\partial e^{v^T W_i + c_i}}{\partial W_{ij}} = -\frac{1}{1+e^{-(v^T W_i + c_i)}} v_j \\
&= -v_j sigmoid(v^T W_i + c_i). \\
\frac{\partial F(v;\theta)}{\partial b_j} &= -v_j \\
\frac{\partial F(v;\theta)}{\partial c_i} &= --\sum_{i=1}^{n} \frac{1}{1+e^{v^T W_i + c_i}} \frac{\partial e^{v^T W_i + c_i}}{\partial c_i} = -\frac{1}{1+e^{-(v^T W_i + c_i)}} v_j \\
&= -sigmoid(v^{\mathrm{T}} W_i + c_i).
\end{aligned}$$

Thus assembling those together will give,

$$\begin{aligned}
\frac{\partial \mathcal{L}(D|\theta)}{\partial W_{ij}} &= \sum_{t=1}^{N} \left(v_j^{(t)} sigmoid((v^{(t)})^{\mathrm{T}} W_i + c_i) - \mathbb{E}_v \left(v_j sigmoid(v^{\mathrm{T}} W_i + c_j) \right) \right) \\
&= \sum_{t=1}^{N} \left(v_j^{(t)} sigmoid((v^{(t)})^{\mathrm{T}} W_i + c_i) \right) - N\mathbb{E}_{v,h}(v_j h_i) \\
\frac{\partial \mathcal{L}(D|\theta)}{\partial b_j} &= \sum_{t=1}^{N} \left(v_j^{(t)} - \mathbb{E}_v(v_j) \right) \\
&= \sum_{t=1}^{N} v_j^{(t)} - N\mathbb{E}_{v,h}(v_j)) \\
\frac{\partial \mathcal{L}(D|\theta)}{c_i} &= \sum_{t=1}^{N} \left(sigmoid((v^{(t)})^{\mathrm{T}} W_i + c_i) - \mathbb{E}_v(sigmoid(v^{\mathrm{T}} W_i + c_i)) \right) \\
&= \sum_{t=1}^{N} \left(sigmoid((v^{(t)})^{\mathrm{T}} W_i + c_i) \right) - N\mathbb{E}_{v,h}(h_i)
\end{aligned}$$

□

- The update equations in (3) contain computationally intractable terms, which ones are intractable? And how do we approximate those terms?

  Proof. The computationally intractable terms come from conditional expectations, where the expectation is over all h and v. We estimate those by contrastive divergence. Let $\hat{h}$ be the hidden state in the current iteration:

  – W_{ij}: $\mathbb{E}_{h,v}(v_j h_i)$ is estimated by,

$$\begin{aligned}
\mathbb{E}_{h,v}(v_j h_i) &= \mathbb{E}_h(h_i \mathbb{E}_{v|h}(v|h)) = \mathbb{E}_h(h_i p(v_j = 1|h;\theta)) \approx \hat{h}_i p(v_j = 1|\hat{h};\theta) \\
&= \hat{h}_i sigmoid(W_j^{\mathrm{T}} \hat{h} + b_j).
\end{aligned}$$

– b_j: $\mathbb{E}_{v,h}(v_j)$ is estimated by,

$$\begin{aligned}\mathbb{E}_{h,v}(v_j) &= \mathbb{E}_h(\mathbb{E}_{v|h}(v|h)) = \mathbb{E}_h(p(v_j = 1|h;\theta)) \approx p(v_j = 1|\hat{h};\theta) \\ &= sigmoid(W_j^{\mathrm{T}}\hat{h} + b_j).\end{aligned}$$

– c_i: $\mathbb{E}_{v,h}(h_j)$ is estimated by $\hat{h}_j$.

Thus the gradients are,

$$\begin{cases} \frac{\partial \mathcal{L}(D|\theta)}{\partial W_{ij}} = \sum_{t=1}^{N} \left(v_j^{(t)} sigmoid((v^{(t)})^{\mathrm{T}} W_i + c_i) - \hat{h}_i^{(t)} sigmoid(W_j^{\mathrm{T}} \hat{h}^{(t)} + b_j) \right) \\ \frac{\partial \mathcal{L}(D|\theta)}{\partial b_j} = \sum_{t=1}^{N} \left(v_j^{(t)} - sigmoid(W_j^{\mathrm{T}} \hat{h}^{(t)} + b_j) \right) \\ \frac{\partial \mathcal{L}(D|\theta)}{\partial c_i} = \sum_{t=1}^{N} \left(sigmoid((v^{(t)})^{\mathrm{T}} W_i + c_i) - \hat{h}_j^{(t)} \right). \end{cases}$$

□

2 Code-From-Scratch

In this portion of the assignment, you will write a binary RBM and train on the MNIST dataset. The MNIST dataset contains $28 \times 28 \times 1$ image, reshape this into 784×1 vector, this is v. Choose h to be of dimension 200. Use the update equations you derived in the pencil-and-paper part to train a RBM.

2.1 Methods

- Describe the functions you wrote, and the overall structure of your code.

  Proof. In this part I wrote two files, `part2.py` to implement required plots and `rbm_part2.py` to implement a restricted Boltzmann machine class. In this class, three modules are implemented:

  – `__init__`: to initialize parameters, e.g., step size, iteration numbers, batch size, etc.

  – `fit`: to fit data and train paramters

  – `getWeights`: to return the learned filters.

□

2.2 Results

- Provide a figure, showing any 64 of the 200 learned filter.

  Proof. As in Figure 1. □

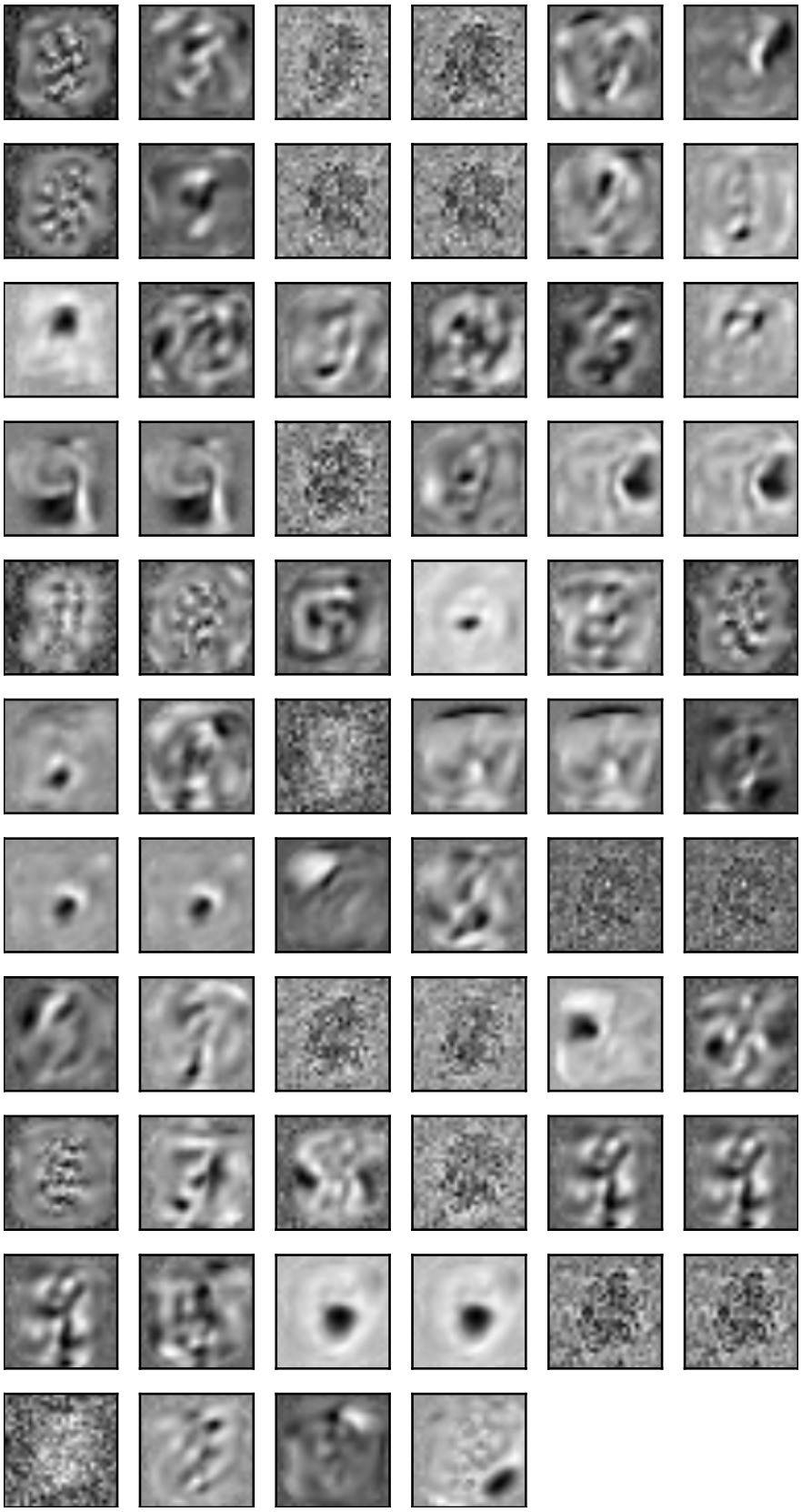


Figure 1: 64 filters obtained from RBM.

3 TensorFlow

In this portion of the assignment, you will use RBM, stack of RBM, and PCA to perform dimension reduction on the raw pixels of the MNIST dataset. From the dimension reduced features, you will perform 10-way classification using multi-class logistic regression.
Report training and testing accuracy for the following:

- Train a 10-way multi-class logistic regression on the raw-pixel data.
- Train a RBM with 200 hidden units. Perform 10-way multi-class logistic regression on this reduced dimension of 200.
- Use PCA to reduce the input dimension to 200. Perform 10-way multi-class logistic regression on this reduced dimension.
- Train a stacking of RBMs, with hidden units [500, 200] (i.e. train a RBM with 500 hidden units, then use the output of the first RBM to train a second RBM with 500 hidden units. Perform 10-way multi-class logistic regression on this reduced dimension.

For the classification part, you are allowed you use higher-level API such as TFLearn[1]. This will greatly reduced the amount of code you need to write.

3.1 Methods

- Describe the RBM and multi-class logistic regression architecture, discuss how you used TensorFlow functions to create such architecture. Make sure your description matches your code;

  Proof. We construct two classes `RMB` and `Logistic` in two files `rmb.py` and `logistic_regression.py` respectively. The `Logistic` class is the same as the one implemented in homework 1.

  The `RMB` class contains following modules:

  - `__init__`: to initialize hyperparamters
  - `nextBatch`: to generate next batch index
  - `fit`: to fit and train the input data. We realize this function via following modules:
    * `build_model`: to build the overall graph model for binary binary restricted boltzmann machine
    * `initialize_tf`: to initialize all tensorflow variables and tensors.
    * `train_model`: to update variables for each batch, where we use `assign_add` to assign a specific value to update parameters manually.
  - `transform`: to transform data into hidden layer

  □

[1] `http://tflearn.org/`

- Report the total number of weights in the stacked RBM.

  Proof. The number of parameters in the first layer is

  $$500 * 784 + 500 + 784 = 393284.$$

  The number of parameters in the second layer is,

  $$500 * 200 + 500 + 200 = 100700.$$

  The total number of weights in the stacked RBM is,

  $$393284 + 100700 = 493984.$$

  □

3.2 Results

- Provide a table, showing the training and testing accuracy for each of the methods mentioned above.

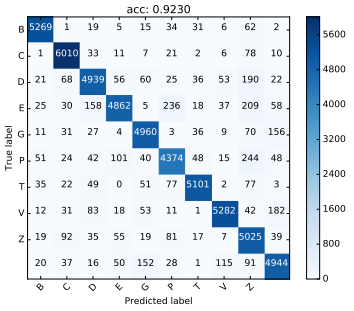
  Proof. The accuracy is provided in Table 1. □

	raw	pca	RBM-200	RBM-500-200
train	92.30	92.44	84.80	90.27
test	92.35	92.12	85.47	89.47

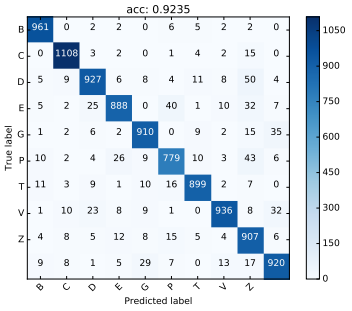
Table 1: Accuracy for four different models.

- Plot training and testing classification confusion matrix for each of the methods mentioned above.

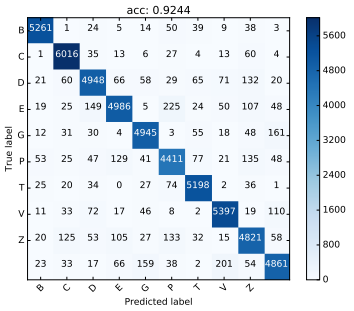
  Proof. The figures for training and testing are plotted in Figure 2. □



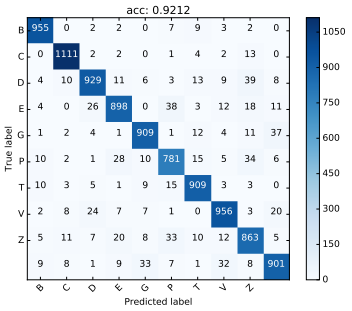
(a) train-raw



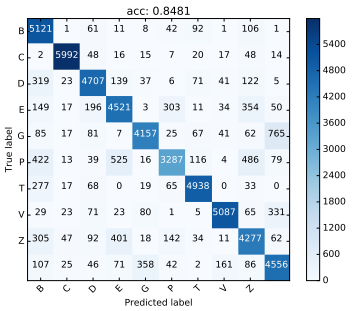
(b) test-raw



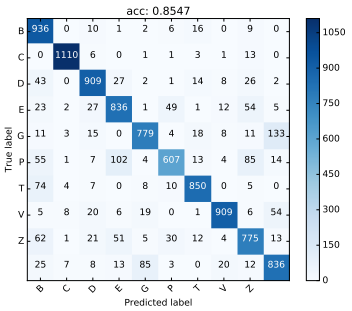
(c) train-pca



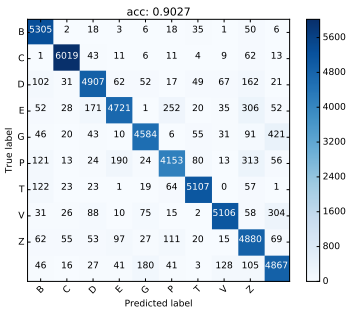
(d) test-pca



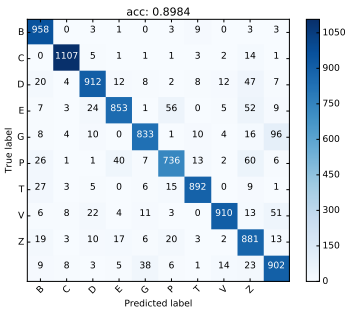
(e) train-rbm-200



(f) test-rbm-200



(g) train-rbm-500-200



(h) test-rbm-500-200

Figure 2: Confusion matrices.