

# EE782 : Advanced Machine Learning assignment 2 report

Viraj Nadkarni (16D070013)

November 2019

## 1 Restricted Boltzmann Machine

The restricted boltzmann machine I used in this assignment was defined using the class *RBM* that I made in the file "*rbm1.py*". This class contains the following important functions that can be useful in the main program (File *main.py*) :

- *trainUsingCD* : This function trains the RBM using contrastive divergence and using the given learning rate ( $\eta$ ), regularization factor ( $\lambda$ ) and momentum parameter ( $\alpha$ ).
- *showFilters* : This function displays all of the feature filters in greyscale that arise after training.
- *outputHgivenV* : Gives a sample output of the hidden layer given an image in the visible layer.

Apart from the functions mentioned above, I also defined other separate functions to do the Gibbs Sampling, training updates and so on.

### 1.1 Results after training RBM

I took the following parameter values to obtain the results :

- Learning rate  $\eta = 0.01$
- L2 regularization factor  $\lambda = 0.01$
- Momentum parameter  $\alpha = 0.9$
- Gibbs sampling iteration  $k = 20$
- Batch size = 20
- Hidden layer size = 100
- Number of epochs = 5

## 2 Using softmax regression to classify images

### 2.1 Without RBM

I modified the code of assignment 1 and used it to define a logistic regression classifier in the file "*mainF.py*". The following is the output I obtained for this classifier on the MNIST data :

```
Epoch 0
validation accuracy 77.4
Epoch 1
validation accuracy 80.08
Epoch 2
validation accuracy 81.22
Epoch 3
```

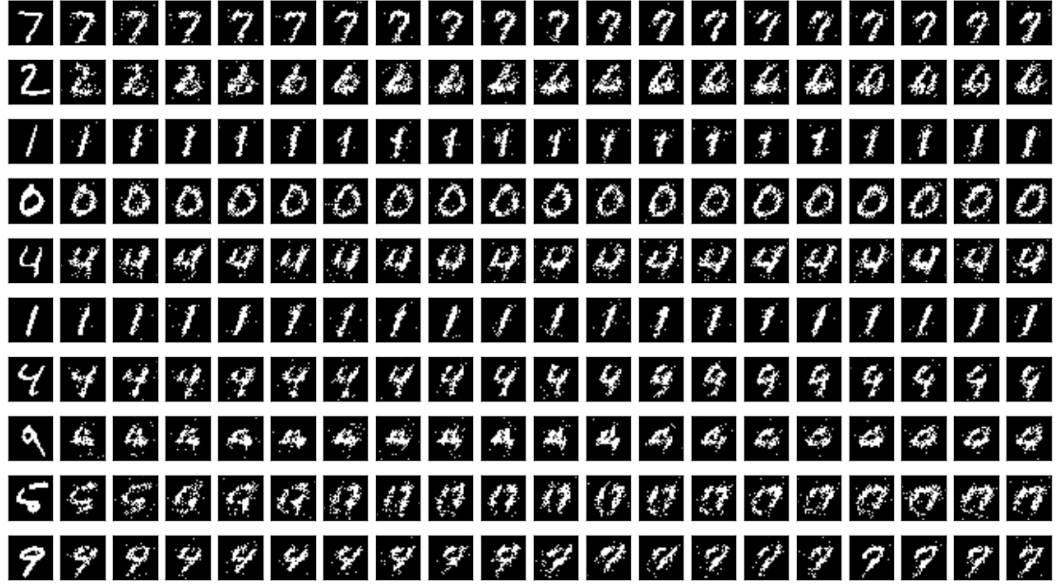


Figure 1: Sampled versions of test images for  $k = 20$  iterations of Gibbs sampling (leftmost image is the original image)

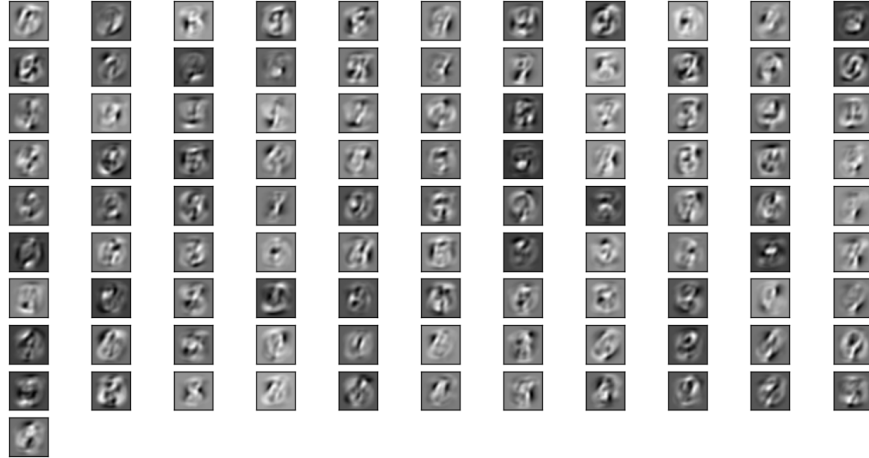


Figure 2: All of the 100 feature detecting filters from the visible to the hidden layer

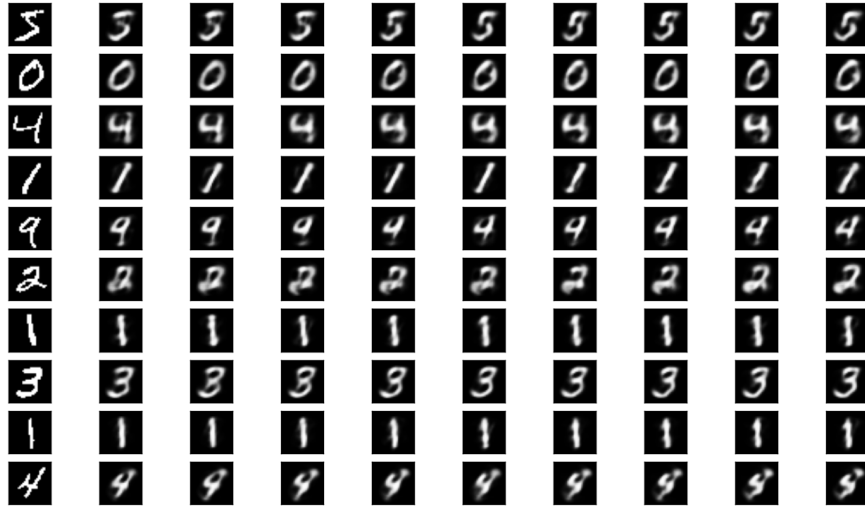


Figure 3: The probability of the Gibbs sampled image as a greyscale image with increasing  $k$  (leftmost image is the original image)

```
validation accuracy 81.36
Epoch 4
validation accuracy 81.39
Epoch 5
validation accuracy 82.45
Epoch 6
validation accuracy 83.08
final accuracy 83.9
```

## 2.2 With RBM

I then passed all of the training and test images of MNIST to extract 100 features per image. Then I trained the regression classifier on this data. The result that I obtained was the following:

```
Epoch 0
validation accuracy 90.71
Epoch 1
validation accuracy 90.75
Epoch 2
validation accuracy 90.75
Epoch 3
validation accuracy 90.75
Epoch 4
validation accuracy 90.75
Epoch 5
validation accuracy 90.75
Epoch 6
validation accuracy 90.75
final accuracy 90.24
```