Graphical Model Project: **Classification Restricted Boltzmann Machine**

LE Son Tung - NGUYEN Thi Ha Giang

April, 2020

The objective of this project is building a Restricted Boltzmann Machine program for classification. We concentrated on the Classification Restricted Boltmann Machine (ClassRBM) with three types of training objective: Generative, Discriminative and Hybrid. We also try with semi-supervised learning. We then experiment on the character recognition problem using MNIST data and evaluate their classification performances.

1 Introduction

1.1 The Classification Restricted Boltzmann Machine (ClassRBM)

The ClassRBM models the joint distribution $p(\mathbf{x}, y)$, where $\mathbf{x} = (x_1, ...x_D)$ is the input and $y \in \{1, ...C\}$ is the target class. The model uses a binary hidden layer $\mathbf{h} = (h_1, ..., h_H)$. We define an energy function:

$$E(y, \mathbf{x}, \mathbf{h}) = -\mathbf{h}^T\mathbf{W}\mathbf{x} - \mathbf{b}^T\mathbf{x} - \mathbf{c}^T\mathbf{h} - \mathbf{d}^T\mathbf{e}_y - \mathbf{h}^T\mathbf{U}\mathbf{e}_y$$

where $\mathbf{e}_y = (1_{i=y})_{i=1}^C$ and $\mathbf{W}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{U}$ are parameters. Then we assign probabilities to every possible set of an input, a label and a hidden vector:

$$p(y, \mathbf{x}, \mathbf{h}) = \frac{\exp(-E(y, \mathbf{x}, \mathbf{h}))}{\sum_{y,\mathbf{x},\mathbf{h}} \exp(-E(y, \mathbf{x}, \mathbf{h}))} = \frac{\exp(-E(y, \mathbf{x}, \mathbf{h}))}{Z}.$$

We assume that elements of $\mathbf{x}$ are binary.

For classification, we must calculate the conditional probability $p(y|\mathbf{x})$. We obtain the following formula :

$$p(y|\mathbf{x}) = \frac{\exp(-F(y, \mathbf{x}))}{\sum_{y^* \in \{1,...,C\}} \exp(-F(y^*, \mathbf{x}))}$$

where $F(y, \mathbf{x}) = d_y + \sum_j \text{softplus}(c_j + U_{jy} \sum_i W_{ij}x_i)$ and $\text{softplus}(a) = \log(1 + \exp(a))$. (Detailed proof in [1].) The function F is called the free energy.

1.2 Our aproach

In order to train a ClassRBM to solve a classification problem, we consider three training objectives: Generative, Discriminative and Hybrid. Then for evaluating these training objectives, we experiment on the classification problem of character recognition. We implement ClassRBMs on MNIST dataset, which is a dataset of 60,000 images of handwritten single digits from 0 to 9. We use 50,000 observations for training and another 10,000 observations for validation.

For optimizing the objective functions, we have 3 different options: Gradient descent, Momentum and Adam optimizer. We fix the mini batch size equal to 100 and try performing ClassRBMs with 200 hidden units. We also try with 100 hidden units and 10,000 observations for training, which takes much less time and only slightly lower accuracy of 1-2% (around 9 minutes for 100 epochs, equivalent to 10,000 updates).

For observing training procedure, after each epoch we print the accuracy of training and validation. Especially, for generative training, we show a sample of reconstruction image of one digit, because objective function of this method focuses on the reconstruction error.

There are some problems associating with computational precision in `R` when we calculate the conditional probability $p(y|\mathbf{x})$. The function *softplus(a)* returns Inf when a is large (> 700), we fix this problem by using approximation $softplus(a) \approx a$ for large a. Another problem is at the last step, after obtaining $-F(y^*, \mathbf{x})$ for all class y^*, if there is a large value of $-F(y, \mathbf{x})$ then the final values of $p(y|\mathbf{x})$ will have some NA values, we fix this by setting $-F(y^*, \mathbf{x}) = -F(y^*, \mathbf{x}) - \max_{y^* \in \{1,\dots,C\}} -F(y^*, \mathbf{x})$, this does not affect the final values of $p(y|\mathbf{x})$.

2 Different types of training objectives and evaluation

2.1 Generative training objective

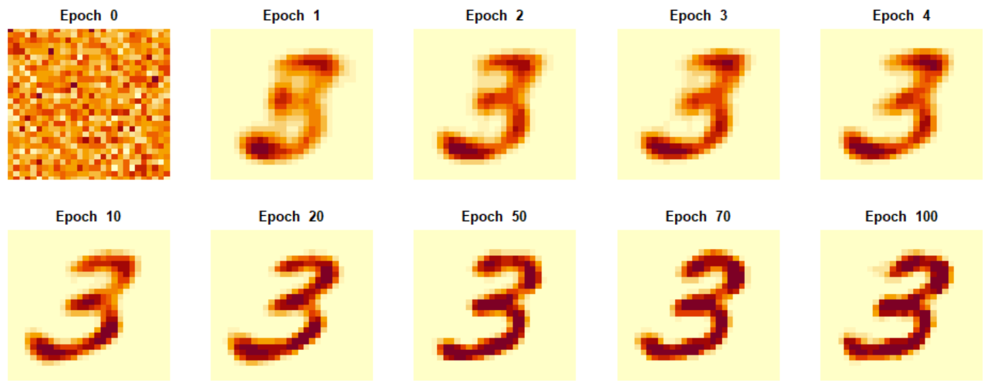
Generative model defines the value of the joint probability $p(y, \mathbf{x})$, the objective function for this model is given by:

$$\mathcal{L}_{gen}(\mathcal{D}_{train}) = -\sum_{t=1}^{|\mathcal{D}_{train}|} \log p(y_t, \mathbf{x}_t).$$

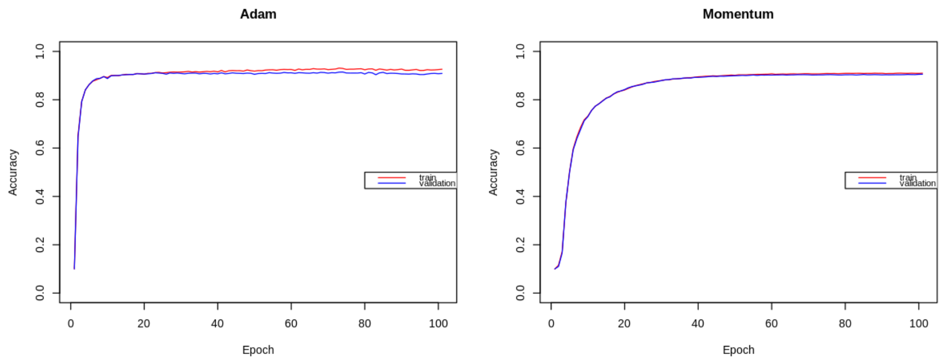
Its gradient is [1]:

$$\frac{\partial \log p(y_t, \mathbf{x}_t)}{\partial \theta} = -\mathbb{E}_{h|y_t,\mathbf{x}_t}\left[\frac{\partial}{\partial \theta} E(y_t, \mathbf{x}_t, \mathbf{h})\right] + \mathbb{E}_{y,\mathbf{x},\mathbf{h}}\left[\frac{\partial}{\partial \theta} E(y, \mathbf{x}, \mathbf{h})\right]$$

The second term is not tractable but can be estimated by the contrastive divergence. We perform this procedure with only one step of Gibbs sampling.



In the above example of reconstruction of a digit 3, we can see the progress of our training through 100 epochs with Adam optimizer: First few epochs for sharping the shape of digits, and after that making it clearer.



The figure above is the accuracy we obtain from training with 10,000 observations and 100 hidden units with learning rate = 0.01. We see that with Adam optimizer it converges faster. After 100 epochs, we got 92.6%

[1]See the Appendix 1 for the proof.

accuracy on training set and 90.9% on validation set.

The objective of this method is catching the generalization of data, that why we see that the accuracy of traning and validation is not much different.

2.2 Discriminative training objective

Discriminative model defines directly the value of the conditional probability $p(y|\mathbf{x})$ with the objective function below:

$$\mathcal{L}_{disc}(\mathcal{D}_{train}) = -\sum_{t=1}^{|\mathcal{D}_{train}|} \log p(y_t|\mathbf{x}_t).$$

The most advantage of the discriminative training is that we can compute its gradient exactly [2]:

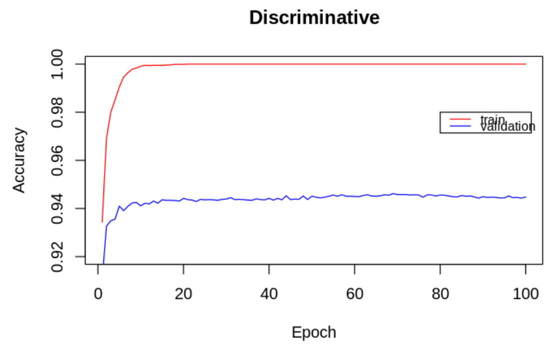
$$\frac{\partial \log p(y_t|\mathbf{x}_t)}{\partial \theta} = -\mathbb{E}_{\mathbf{h}|y_t,\mathbf{x}_t}\left[\frac{\partial}{\partial \theta}E(y_t,\mathbf{x}_t,\mathbf{h})\right] + \mathbb{E}_{y,h|\mathbf{x}_t}\left[\frac{\partial}{\partial \theta}E(y,\mathbf{x}_t,\mathbf{h})\right].$$

Notice the second term of this gradient:

$$\mathbb{E}_{y,\mathbf{h}|\mathbf{x}_t}\left[\frac{\partial}{\partial \theta}E(y,\mathbf{h}|\mathbf{x}_t)\right] = \mathbb{E}_{y|\mathbf{x}_t}\left[\mathbb{E}_{\mathbf{h}|y,\mathbf{x}_t}\left[\frac{\partial}{\partial \theta}E(y,\mathbf{x}_t,\mathbf{h})\right]\right].$$

We can compute this expectation by average the gradient $\frac{\partial}{\partial \theta}E(y,\mathbf{x}_t,\mathbf{h})$ for each class y and weighted by $p(y|\mathbf{x}_t)$.

Below is the result we got after 100 epochs when training on 10,000 observations, 100 hidden units. Using Adam optimizer, we got 100% accuracy on traning set and 94.5% on validation set. This is much better than generative model, however, the difference between training and validation accuracy is larger.



2.3 Hybrid training objective

The idea of this model is combining the advantages of both generative and discriminative models: The first one is for generalization data and the second focusing on classification task. The objective function is given by:

$$\mathcal{L}_{hybrid}(\mathcal{D}_{train}) = \mathcal{L}_{disc}(\mathcal{D}_{train}) + \alpha\mathcal{L}_{gen}(\mathcal{D}_{train}).$$

The gradient of this model can be computed based on the two precedent gradient.

The second term of the hybrid objective function can be considered as the regularization term for the discriminative objective, which makes our model more generalized. In our case, with only 100 hidden units, we do not see the sign of over-fitting problem, therefore, it is difficult to evaluate the effectiveness of the hybrid model.

With similar setting to the discriminative training, for hybrid training objective, with $\alpha = 0.01$, after 100 epochs, we have 92.3% accuracy on validation set, which is higher than generative but lower than discrimative one.

[2]See the Appendix 2 for the formulas.

3 Semi-supervised Learning

Because RBM is a generative model, we can use this method for semi-supervised learning. Our objective function is

$$\mathcal{L}_{semi}(\mathcal{D}_{train}, \mathcal{D}_{unlabel}) = \mathcal{L}_{sup}(\mathcal{D}_{train}) + \beta \mathcal{L}_{unsup}(\mathcal{D}_{unlabel})$$

where $\mathcal{L}_{sup}$ is one of the above objective functions and

$$\mathcal{L}_{unsup}(\mathcal{D}_{unlabel}) = - \sum_{t=1}^{|\mathcal{D}_{unlabel}|} \log p(\mathbf{x}_t)$$

We have the form of the gradient for unsupervised part:

$$\frac{\partial \log p(\mathbf{x}_t)}{\partial \theta} = -\mathbb{E}_{y,\mathbf{h}|\mathbf{x}_t}\Big[\frac{\partial}{\partial \theta} E(y, \mathbf{x}_t, \mathbf{h})\Big] + \mathbb{E}_{y,\mathbf{h},\mathbf{x}}\Big[\frac{\partial}{\partial \theta} E(y, \mathbf{x}, \mathbf{h})\Big].$$

The first term is the same as the second term of the discriminative gradient. For estimating the second term of unsupervised part, we use Contrastive Divergence with $\mathbf{x}_0 = \mathbf{x}_t$ and y_0 is a sample from $p(y|\mathbf{x}_t)$.

We try this method with 1,000 labeled data and 49,000 unlabeled data and 100 hidden units, $\beta = 0.1$ and use mini batch size = 1,000. After 20 epochs, we got 94.8% accuracy on train set and 86.8% accuracy on validation set.

4 Conclusion

RBM is a relatively simple shallow model, with only two layers, however, it is quite effective for classification problems, for example, the character recognition problem. In our experiments with MNIST, with only 10,000 training observations and 100 hidden units, we obtain the results of over 90% accuracy on validation set. With 50,000 training observation and 200 hidden units, the accuracy increases about 2%. We didn't see sign of overfitting, so we can not evaluate the performance of hybrid training objective. As for semi-supervised learning with 10,000 training observations, we were unable to find an optimal setting. For better results, we think it is necessary to increase the number of training observations and number of hidden units.

References

[1] Hugo Larochelle. "Learning Algorithms for the Classification Restricted Boltzmann Machine". In: *Journal of Machine Learning Research 13* (2012).

[2] Geoffrey Hinton. *A practical guide to training restricted Boltzmann machines.*

Appendix 1

Generative objective function: Form of the gradient

We have:

$$\begin{aligned}
\log p(y_t, \mathbf{x}_t) &= \log\Big(\sum_{\mathbf{h}} p(y_t, \mathbf{x}_t, \mathbf{h})\Big) \\
&= \log\Big(\sum_{\mathbf{h}} \frac{\exp(-E(y_t, \mathbf{x}_t, \mathbf{h}))}{Z}\Big) \\
&= \log\Big(\sum_{\mathbf{h}} \exp(-E(y_t, \mathbf{x}_t, \mathbf{h}))\Big) - \log(Z)
\end{aligned}$$

Consider the first term:

$$\begin{aligned}
\frac{\partial}{\partial\theta}\log\Big(\sum_{\mathbf{h}} \exp(-E(y_t, \mathbf{x}_t, \mathbf{h})))\Big) &= \frac{1}{(\sum_{\mathbf{h}} \exp(-E(y_t, \mathbf{x}_t, \mathbf{h})))}\frac{\partial}{\partial\theta}\sum_{\mathbf{h}} \exp(-E(y_t, \mathbf{x}_t, \mathbf{h}))) \\
&= \frac{1}{(\sum_{\mathbf{h}} \exp(-E(y_t, \mathbf{x}_t, \mathbf{h})))}\sum_{\mathbf{h}} \exp(-E(y_t, \mathbf{x}_t, \mathbf{h}))\frac{\partial}{\partial\theta}(-E(y_t, \mathbf{x}_t, \mathbf{h})) \\
&= \sum_{\mathbf{h}} p(\mathbf{h}|y_t, \mathbf{x}_t)\frac{\partial}{\partial\theta}(-E(y_t, \mathbf{x}_t, \mathbf{h})) \\
&= \mathbb{E}_{\mathbf{h}|y_t,\mathbf{x}_t}\Big[\frac{\partial}{\partial\theta} - E(y_t, \mathbf{x}_t, \mathbf{h})\Big]
\end{aligned}$$

Similarly,

$$\begin{aligned}
\frac{\partial}{\partial\theta}\log(Z) &= \frac{\partial}{\partial\theta}\sum_{y,\mathbf{x},\mathbf{h}} \exp(-E(y, \mathbf{x}, \mathbf{h})) \\
&= \mathbb{E}_{y,\mathbf{x},\mathbf{h}}\Big[\frac{\partial}{\partial\theta}E(y, \mathbf{x}, \mathbf{h})\Big]
\end{aligned}$$

Hence,

$$\frac{\partial \log p(y_t, \mathbf{x}_t)}{\partial\theta} = -\mathbb{E}_{\mathbf{h}|y_t,\mathbf{x}_t}\Big[\frac{\partial}{\partial\theta}E(y_t, \mathbf{x}_t, \mathbf{h})\Big] + \mathbb{E}_{y,\mathbf{x},\mathbf{h}}\Big[\frac{\partial}{\partial\theta}E(y, \mathbf{x}, \mathbf{h})\Big]$$

Appendix 2

Discriminative objective function: Form of the gradient for a $(y_t, \mathbf{x}_t)$ for each parameter

For $\mathbf{W}$:

$$\mathbb{E}_{\mathbf{h}|y^*,\mathbf{x}_t}\Big[\nabla_{\mathbf{W}}E(y^*, \mathbf{x}_t, \mathbf{h})\Big] = \mathbb{E}_{\mathbf{h}|y^*,\mathbf{x}_t}\Big[\mathbf{h}\otimes\mathbf{x_t}\Big] = -\vec{p}(\mathbf{h}|y^*, \mathbf{x}_t)\otimes\mathbf{x}_t$$

$$\begin{aligned}
\implies \mathbb{E}_{y,\mathbf{h}|\mathbf{x}_t}\Big[\nabla_{\mathbf{W}}E(y^*, \mathbf{x}_t, \mathbf{h})\Big] &= \mathbb{E}_{y|\mathbf{x}_t}\Bigg[\mathbb{E}_{\mathbf{h}|y,\mathbf{x}_t}\Big[\nabla_{\mathbf{W}}E(y, \mathbf{x}_t, \mathbf{h})\Big]\Bigg] \\
&= \sum_{y^*\in\{1,\dots C\}} \mathbb{E}_{\mathbf{h}|y^*,\mathbf{x}_t}\Big[\nabla_{\mathbf{W}}E(y^*, \mathbf{x}_t, \mathbf{h})\Big]p(y^*|\mathbf{x}_t) \\
&= -\sum_{y^*\in\{1,\dots C\}} [\vec{p}(\mathbf{h}|y^*, \mathbf{x}_t)\otimes\mathbf{x}_t]\ p(y^*|\mathbf{x}_t)
\end{aligned}$$

$$\implies \nabla_{\mathbf{W}}\log p(y_t|\mathbf{x}_t) = \vec{p}(\mathbf{h}|y_t, \mathbf{x}_t)\otimes\mathbf{x}_t - \sum_{y^*\in\{1,\dots C\}} [\vec{p}(\mathbf{h}|y^*, \mathbf{x}_t)\otimes\mathbf{x}_t]\ p(y^*|\mathbf{x}_t)$$

Similarly for other parameters:

$$
\begin{aligned}
\nabla_{\mathbf{U}} \log p(y_t|\mathbf{x}_t) &= \vec{p}(\mathbf{h}|y_t, \mathbf{x}_t) \otimes \mathbf{e}_{y_t} - \sum_{y^* \in \{1, \ldots C\}} \left[\vec{p}(\mathbf{h}|y^*, \mathbf{x}_t) \otimes \mathbf{e}_{y^*}\right] \, p(y^*|\mathbf{x}_t) \\
\nabla_{\mathbf{c}} \log p(y_t|\mathbf{x}_t) &= \vec{p}(\mathbf{h}|y_t, \mathbf{x}_t) - \sum_{y^* \in \{1, \ldots C\}} \vec{p}(\mathbf{h}|y^*, \mathbf{x}_t) p(y^*|\mathbf{x}_t) \\
\nabla_{\mathbf{d}} \log p(y_t|\mathbf{x}_t) &= \mathbf{e}_{y_t} - \sum_{y^* \in \{1, \ldots C\}} \mathbf{e}_{y^*} \, p(y^*|\mathbf{x}_t) \\
\nabla_{\mathbf{b}} \log p(y_t|\mathbf{x}_t) &= 0
\end{aligned}
$$