Project Name:	Restricted Boltzmann Machine
Name:	Alec Mabhiza Chirawu
Chinese Name:	亚历克上
Student Number:	M202161029

Neural Networks and Deep Learning
Information And Engineering Department
USTB

Professor: 黄旗明(Roland)

22 April 2022

Summary

Restricted Boltzmann Machines (RBMs) are a class of undirected probabilistic graphical models containing a layer of observable variables and a single layer of latent variables. In RBMs, there are no connections within a layer - there are only correlations between layers. Markov random fields (RBM) are building blocks for deep belief networks. This tutorial introduces RBMs from the point of view of a Markov random field. Different learning algorithms for RBMs are discussed, as well as an introduction to sampling from RBMs using MCMC techniques.

TABLE OF CONTENTS

1 Introduction

Restricted Boltzmann machines (RBM) are a special case of general Boltzman machines, which can be thought of as bidirectionally connected networks of stochastic processing units. In the RBMs network graph, each neuron is connected to all the other neurons in the other layers, but there are no connections between them. This restriction gives the RBM its name and leads us to the topic of this project. The learning problem can be simplified by imposing restrictions on the network topology, which creates RBMs.

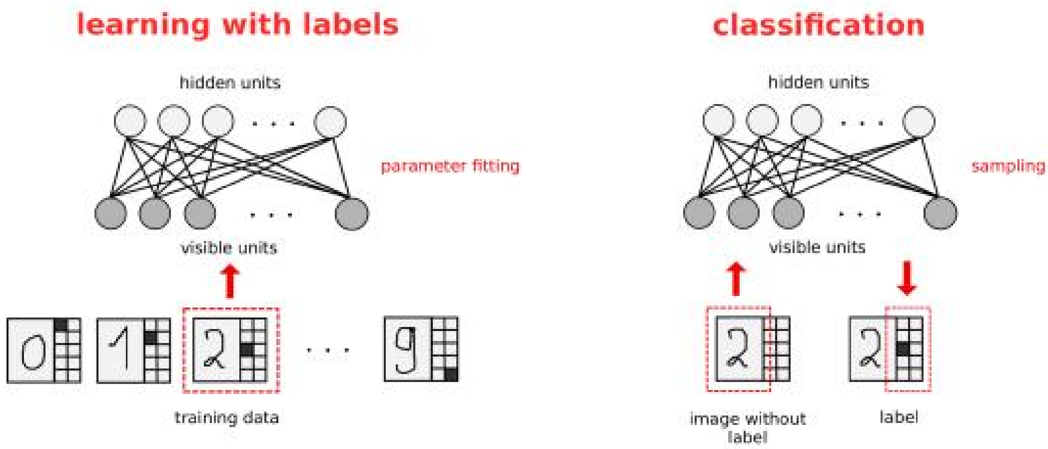
1.1 Methodology

An RBM provides a closed-form representation of the distribution underlying the training data. It is a generative model that allows sampling from the learned distribution. For example, we can fix some units corresponding to a partial observation and sample the remaining units to complete the observation. In this way, RBMs can also be used as classifiers.

2. Findings

2.1 Graphical models

When viewed as neural networks they are used as functions mapping the observations to the expectations of the latent variables in the top layer. These can be interpreted as the learned features, which can, for example, serve as inputs for a supervised learning system. It has been argued that this unsupervised pre training of the feed-forward neural network weights based on a generative model helps to overcome some of the problems faced by multi-layer neural networks.



2.2 ***Experiments***

The who system(hidden and visible nodes) is described by an energy function:

$$E(v,h) = -v^T Wh - v^T b - h^T c$$

As in statistical physics, high-energy configurations are less probable. The joint probability distribution is defined as:

$$p(v,h) = e^{-E(v,h)} / z$$

where Z is the partition function (intractable)

Our goal is to learn the joint probability distribution that maximizes the probability over the data, also known as likelihood.

$$p(v) = sum_h p^{(v,h)} = e^{-F(v)} / z$$

where F(v) is called Free Energy

Inference:
The Conditional distribution factorizes (no intra layer connections):

$$p(h_j = 1 \mid v) = p(h_j = 1, v) / (p(h_j = 0, v) + p(h_j = 1, v) = sigmoid(c_j + v^T W_{:j})$$

$$p(v_i = 1 \mid h) = sigmoid(b_i + W_{i:} h)$$

Learning

The parameters of our model are the weights W and the biases b, c.

Maximizing the log-Likelihood

Derive log-likelihood and gradient formulas.
It is impractical to compute the exact log-likelihood gradient (expectation of the joint distribution).

Contrastive divergence

Idea:

1. Replace the expectation by a point estimate at v'
2. Obtain the point v' by Gibbs Sampling
3. Start sampling chain at v(t)

1-step divergence:

- Positive divergence: $h(v)v^T$
- Negative divergence: $h(v`)v`^T$

where v' is reconstructed from a sample from h(v)

Pseudocode:

1. For each training example v(t):
   i. Generate a negative sample v' using k steps of Gibbs Sampling, starting at v(t)
   ii. Update parameters:

$$w_{new} = w_{old} + \backslash epsilon * (hv(t))v(t)^T - h(v`)v`^T$$

$$b_{new} = b_{old} + \backslash epsilon * (h(v(t)) - h(v`))$$

$$c_{new} = c_{old} + \backslash epsilon * (v(t) - v`)$$

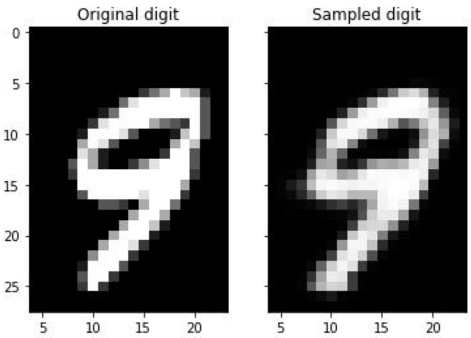
Go back to 1. until stopping criteria

3. Results

```
  0%|          | 0/10000 [00:00<?, ?it/s]
Reconstruction error at step 0: 0.331
 10%|          | 977/10000 [00:09<01:23, 108.34it/s]
Reconstruction error at step 1000: 0.024
 20%|          | 1981/10000 [00:17<01:10, 113.53it/s]
Reconstruction error at step 2000: 0.020
 30%|          | 2980/10000 [00:26<01:01, 113.73it/s]
Reconstruction error at step 3000: 0.019
 40%|          | 3985/10000 [00:34<00:51, 115.85it/s]
Reconstruction error at step 4000: 0.018
 50%|          | 4993/10000 [00:42<00:42, 117.17it/s]
Reconstruction error at step 5000: 0.017
 60%|          | 5994/10000 [00:51<00:34, 115.29it/s]
Reconstruction error at step 6000: 0.016
 70%|          | 6998/10000 [01:00<00:26, 114.79it/s]
Reconstruction error at step 7000: 0.015
 80%|          | 7998/10000 [01:10<00:17, 113.10it/s]
Reconstruction error at step 8000: 0.015
 90%|          | 8975/10000 [01:19<00:09, 112.93it/s]
Reconstruction error at step 9000: 0.014
100%|          | 10000/10000 [01:28<00:00, 112.94it/s]
```

The weights of RBMs with 16 and 100 hidden units trained on MNIST for 100 epochs. Each square corresponds to the 784 weights of one hidden neuron to the 28 × 28 = 784 visibleneurons. The squares are ordered according to the probabilities of the corresponding hidden units tobe 1 given the training set in decreasing order.

The goal of this project was to introduce RBMs from the probabilistic graphical model perspective. It is meant to supplement existing programs and focus on material that are helpful in this work. The outputs are represented as follows:



4. Conclusion

At different stages of the training, we clamped the first column of the input image to a certain pattern and sampled from the RBM. As the model distribution got close to the training distribution, the RBM successfully reconstructed the BAS pattern in a few sampling steps, as can be seen in this project. After some training, the samples started to reveal the structure of the BAS distribution.

5. Reference List

Fischer, Asja, and Christian Igel. "An introduction to restricted Boltzmann machines." *Iberoamerican congress on pattern recognition*. Springer, Berlin, Heidelberg, 2012.

Neukart, Florian. *Reverse Engineering the Mind: Consciously Acting Machines and Accelerated Evolution*. Vol. 94. Springer, 2016.

Zhang, N., Ding, S., Zhang, J., & Xue, Y. (2018). An overview on restricted Boltzmann machines. *Neurocomputing*, *275*, 1186-1199.