

**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**

[Summary ii](#_Toc670)

[TABLE OF CONTENTS iii](#_Toc27499)

[1 Introduction 1](#_Toc1445)

*[1.1 Methodology](#_Toc22660)* [1](#_Toc22660)

[2. Findings 2](#_Toc23792)

[3. Results 5](#_Toc29567)

[4. Conclusion 6](#_Toc13197)

[5. Reference List 7](#_Toc23205)

**1 Introduction**

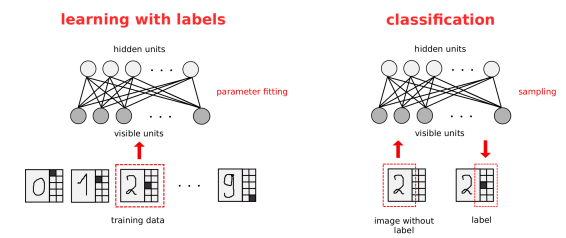
Restricted Boltzmann machines (RBM) are a special case of general Boltzman machines, which can be thought of as bidirectionally connected networks of stochastic processing units. In the RBMs network graph, each neuron is connected to all the other neurons in the other layers, but there are no connections 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. ***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.

1. **Findings**
   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.



* 1. ***Experiments***

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



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



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.



where F(v) is called Free Energy

**Inference:**

The Conditional distribution factorizes (no intra layer connections):





**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: 

- Negative divergence: 

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

**Pseudocode:**

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

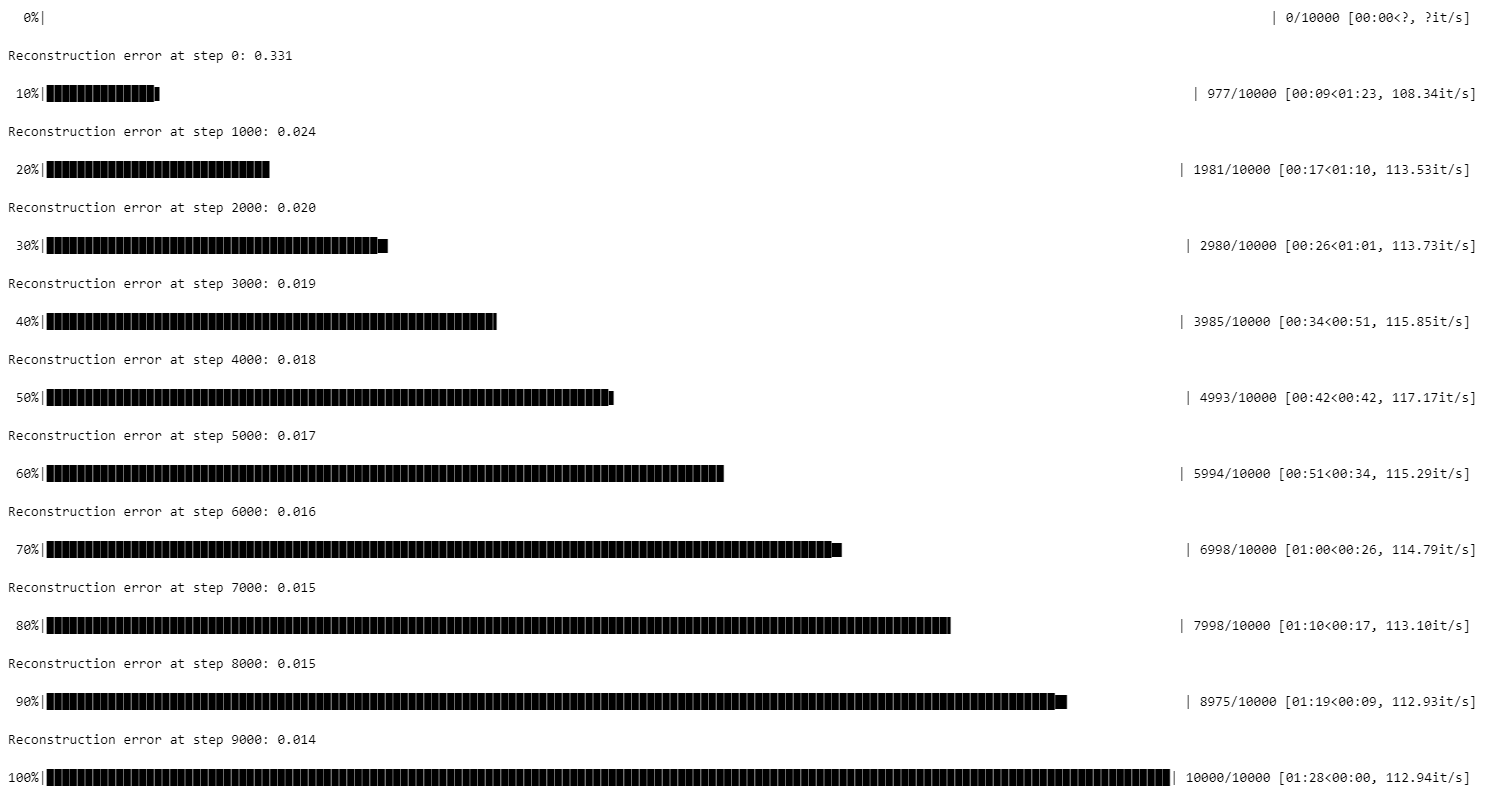






Go back to 1. until stopping criteria

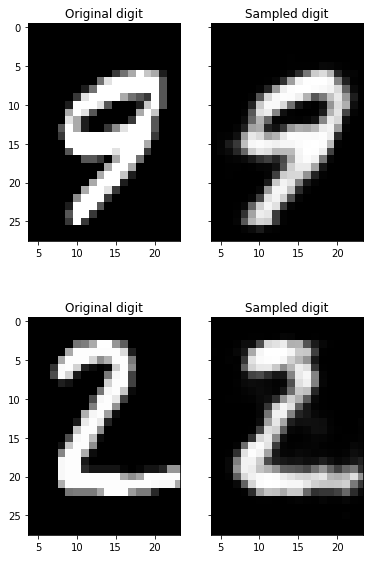
1. **Results**



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:



1. **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.

1. **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.