Universidade do Estado do Rio de Janeiro

centro

unidade patrono

Artur Chiaperini Grover

Computational Intelligence Using Deep Neural Networks (BM)

Rio de Janeiro

2019

Artur Chiaperini Grover

Computational Intelligence Using Deep Neural Networks (BM)

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Curso, da Universidade do Estado do Rio de Janeiro.

Orientador: Cargo Titulação Nome Sobrenome
Coorientador: Cargo Titulação Nome Sobrenome

Rio de Janeiro
2019

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/D

D979 Chiaperini Grover, Artur
Computational Intelligence Using Deep Neural Networks (BM) / Artur Chiaperini Grover. – Rio de Janeiro, 2019-
29 f.

Orientador: Cargo Titulação Nome Sobrenome
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, unidade, Programa de Pós-Graduação em Curso, 2019.

1. primeira palavra chave.. 2. segunda palavra chave.. 3. terceira palavra chave.. I. Cargo Titulação Nome Sobrenome. II. Universidade do Estado do Rio de Janeiro. III. unidade. IV. Título

CDU 02:141:005.7

____________________________ ____________

Assinatura Data

Artur Chiaperini Grover

Computational Intelligence Using Deep Neural Networks (BM)

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Curso, da Universidade do Estado do Rio de Janeiro.

Aprovada em 22 de Novembro de 2019.

Banca Examinadora:

Cargo Titulação Nome Sobrenome (Orientador)
Unidade – Instituição

Cargo Titulação Nome Sobrenome (Coorientador)
Unidade – Instituição

primeiro membro titular da banca
instituição

segundo membro titular da banca
instituição

terceiro membro titular da banca
instituição

primeiro membro suplente da banca
instituição

segundo membro suplente da banca
instituição

terceiro membro suplente da banca
instituição instituição instituição instituição

Rio de Janeiro

2019

If no one died, it is just another story to be told.
[Daniel Mirolhaum]

RESUMO

CHIAPERINI GROVER, A. C. G. *Computational Intelligence Using Deep Neural Networks (BM)*. 2019. 29 f. Dissertação (Mestrado em Curso) – unidade, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2019.

Texto do resumo em português.

Palavras-chave: primeira palavra chave. segunda palavra chave. terceira palavra chave.

ABSTRACT

CHIAPERINI GROVER, A. C. G. *Inteligência Computacional Usando Redes Neuronais Profundas.* 2019. 29 f. Dissertação (Mestrado em Curso) – unidade, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2019.

Abstract in English.

Keywords: first keyword. second keyword. third keyword.

LIST OF FIGURES

LIST OF TABLES

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

sigla1	por extenso
sigla2	por extenso
sigla3	por extenso

LISTA DE SÍMBOLOS

$simbolo1$	significado e/ou valor
$simbolo2$	significado e/ou valor
$simbolo3$	significado e/ou valor

CONTENTS

INTRODUCTION

Nowadays, modelling intelligent complex systems uses two main paradigms, commonly referred to as Symbolism and Connectionism, as basic guidelines for achieving your goals of creating intelligent machines and understanding human cognition. These two approaches depart from different positions, each advocating advantages over the other in reproducing intelligent activity. The traditional symbolic approach argues that the algorithmic manipulation of symbolic systems is an appropriate context for modelling cognitive processes. On the other hand, connectionists restrict themselves to brain-inspired architectures and argue that this approach has the potential to overcome the rigidity of symbolic systems by more accurately modeling cognitive tasks that can only be solved, in the best case, approximately. Years of experimentation with both paradigms lead us to the conclusion that the solution lies between these two extremes, and that the approaches must be integrated and unified. In order to establish a proper link between them, much remains to be researched.

If in the 1980s the discussion of intelligence was placed at the distinct poles of the symbolists and connectionists, today the connectionists are divided by the reductionist arguments of the structuralists. For this structuralist current, the failure of the symbolists was due to the fact that their models despised brain architecture, and therefore connectionism must continue to explore more deeply the structural aspects of the thinking organ. In this project, the connectionist and structuralist aspects are approached, respectively, through the paradigm of artificial neural networks and realistic models of the brain, within the area called Computational Neuroscience. Through our models, we investigate ancient questions of Artificial Intelligence regarding the understanding of computability aspects of the human mind.

In this project we will continue with the study and implementation of Deep Neural Networks (RNPs), which have been used to solve artificial intelligence problems, in areas such as: automatic speech (or voice) recognition, image recognition and treatment, natural language processing, bioinformatics, among many others.

Our previous experience, both in the development of research in the field of artificial neural networks and general distributed processing and its technological applications, as well as in the pursuit of realistic models of brain biology, allows us to mature in the same direction of multidisciplinary research.

1 BOLTZMANN MACHINES

EXPLICAR OS ELEMENTOS DE PROBABILIDADE COM OS QUAIS ESTAMOS LIDANDO:QUEM SÃO AS VARIÁVEIS ALEATÓRIAS E COMO VAMOS DOMINÁ-LAS NO TEXTO, COMENTAR QUE ESTAMOS TRATANDO COM VARIÁVEIS DISCRETAS, COMO VAMOS IDENTIFICAR AS PROBABILIDADES,. . .

EXPLICAR O QUE OS ÍNDICES REPResentam, e as variiáveis.

EXPLICAR OS TERMOS ω e ϕ. ω É para os pesos, e ϕ para o bias de cada unidade.

EXPLICAR MINHA NOTAÇÃO sobre o sobre-escrito entre parênteses!!!

In this chapter will expose the Boltzmann Machine theory. A few considerations regarding the notation used in this exposition is required before stepping forward into the main content. A single random variables is denoted by x. A vector of random variables of size n is represented by $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n)$, where each x_i, $i \in \{1, 2, \ldots, n\}$, represents a single unit of the network. The value a random variable can assume is represented by x. Assuming a discrete scenario, the probability of a random variable x_i of assuming a certain value x_i is $P(\mathrm{x}_i = x_i)$.

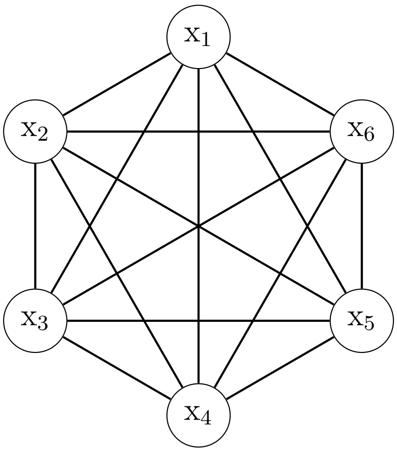
We begin this chapter by introducing the Hopfield Network. The reason is because the Boltzmann Machine is a generalization of the Hopfield Network. In Hopfield Networks units are deterministic while, in Boltzmann Machines, units are stochastic.

1.1 Hopfield Networks

Hopfield networks are a simple neural network architecture, often referred to as an associative memory network (HERTZ; KROGH; PALMER, 1991). As (GÉRON, 2017) mentioned, this kind of network is first taught a few patterns, and then when exposed to new patterns it will output the closest learned pattern. Figure 1 shows that a Hopfield network is a fully connected graph, this means that each unit is connected to every other single unit of the network. It is a different units arrangement compared to perceptron, for instance (HOPFIELD, 1982).

This kind of network is an energy-based model, because there is a global energy function associated to the network. This global energy function evolves to a low-energy state during training phase of the network, i.e., the network weights are modified so that the energy decreases. When connections ω between units, also know as weights or *synaptic strenght*, are symmetrical, i.e., $\omega_{ij} = \omega ji$, then (HOPFIELD, 1982) presented

Figure 1 - Hopfiel Network diagram.



Legend: Each white circle represents a single unit of the Hopfield Network.
Source: Author.

that the energy function is give by

$$H = -\frac{1}{2}\sum_i\sum_j \omega_{ij}x_ix_j - \sum_i \phi_i x_i, \tag{1}$$

Binary units with recurrent connections between them. Is the connections are symmetric, there is a global energy function: each binary configuration of the whole network has an energy. Binary threshold decision rule causes the network to settle to a minimum of this energy function. Energy function is the sum of many contributions. Each contribution depends on one connection weight and the ninary state of two neurons is. (Energy is bad, so that is why we have a minus sign. The less energy, the better.)

$$H = -\frac{1}{2}\sum_i\sum_j \omega_{ij}x_ix_j - \sum_i \phi_i x_i, \tag{2}$$

where the first term is the symmetric connection between neurons ω_{ij} and the activity of the two connected neurons x_i and x_j. The second term only involves the state of individual units.

The quadraticc energy function makes it possible for each unit to compute locally how it's state affects the global energy, in another words, how each unit affects the global energy when its state is changed. Define the energy gap ΔH_i for a unit x_i, this measure is the global energy function difference when the unit x_i has its state changed.

$$\Delta H_i = H(x_i = 0) - H(x_i = 1) = \sum_j \omega_{ij}x_j + \phi_i \tag{3}$$

the energy gap equation can also be read as the difference between the energy when x_i is off and the energy when x_i is on. The energy gap can also be computed by differenciating the energy function H, equation (2). Refer to appendix (Número) for differenciation. The Hopfield network will go down hill in this global energy. To find the energy minimum, start from random state, update this unit one at a time in random order. Update each unit to whichever of its two states gives the lowest global energy. Hopfield According to [HOPFIELD], memories could be seen as energy minima. One way of interpreting this is that an item could be accessed by just knowing some parts of it. An interesting analogy expose by [HINTON] is that Hopfield networks is like reconstructing a dinosaur from a few bones. FALTA COLOCAR A REGRA DE ATUALIZACAO DOS PESOS! PARA MOSTRAR COMO AS MEMORIAS SAO ARMAZENADAS.

1.2 Boltzmann Machines

Boltzmann Machines (BM) are a type of stochastic neural networks (SNN) where the connections between units, which are described by ω, are symmetrical, i.e., $\omega_{ij} = \omega_{ji}$ (HERTZ; KROGH; PALMER, 1991). This kind of stochastic neural networks are capable of learning internal representation and to model an input distribution. Boltzmann Machines were named after the Boltzmann distribution. Due to its stochatics behaviour, the probability of the state of the system to be found in a certain configuration is given by previous mentioned distribution [HERTZ]. According to [MONTUFAR, 2018], BM can be seen as an extension of Hopfield networks to include hidden units.

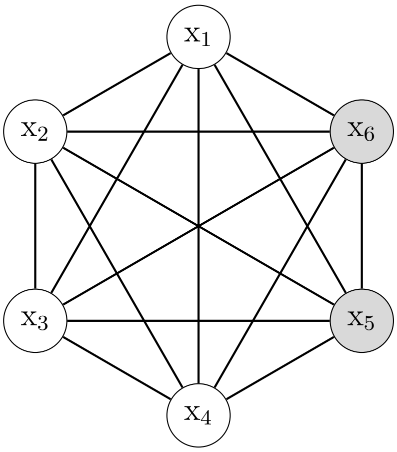
Boltzamann Machines have visible and hidden units. The visible units are linked to the external world and they correspond to the components of an observation. On the other hand, the hidden units do not have any connection outside of the network and model the dependencies between the components of the observations [FISCHER, 2012]. In BM, there is no connection restriction, this means that every unit, visible or hidden, can be connected to every other unit as in a complete graph, this pattern is not mandatory as some of the connections may not exists depending on the network layout.

Training Boltzmann Machines means finding the right connection between the units.

Boltzmann Machines (BM) are stochastics neural networks with symmetric connections, i.e., $\omega_{ij} = \omega_{ji}$. Boltzmann Machines use the Boltzmann distribution to determine the probability of the state of the system of the network. BM ressambles the Hopfield networks with the inclusion of hidden units. Finding the right connections between the hidden units without knowing it from the training patterns what the hidden units represent is part of the solving the Boltzmann Machine problem.

Units x_i in BM are split into two kinds: visible and hidden units. The visible

Figure 2 - Boltzmann machine diagram.



Legend: Gray circles represent the hidden units of the Boltzmann Machine, while the white circles are the visible units.
Source: Author.

units have connection to the outside world and are the units that receive the data input. On the other hand, the hidden units do not have any connection to the outside of the network and they are resposible to find the data relation from the input. In a BM, the connections between units can be complete or not. Regardless of how the connections are, every connection in a BM is symmetric.

BM are made of stochastics units x_i. Stochastics units are random variables that can assume a binary value with a certain probability. We will consider that a random variable x_i can assume a value $x_i \in \{0, 1\}$, iė;

$$x_i = \begin{cases} 1 \text{ with probability } g(h_i) \\ 0 \text{ with probability } 1 - g(h_i) \end{cases}, \tag{4}$$

where the probability is given by

$$g(h_i) = \frac{1}{1 + e^{-2\beta h_i}}, \tag{5}$$

and

$$h_i = \sum_j \omega_{ij} x_j. \tag{6}$$

Due to the symmetrical connections, there is an energy function give by

$$H(\mathbf{x}) = -\frac{1}{2} \sum_i \sum_j \omega_{ij} x_i x_j - \sum_i \phi_i x_i, \tag{7}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, and n is equal to the number of units in the network, and the above energy function has minimum when there is a stable state characterised by

$$x_i = sgn(h_i). \tag{8}$$

The probability P of finding the system in a given state $\mathbf{x}$ after the equilibrium is reached can be computed as follows:

$$P(\mathbf{x}) = \frac{1}{Z} e^{-\beta H(\mathbf{x})}, \tag{9}$$

where

$$Z = \sum_{\mathbf{x}'} e^{-\beta H(\mathbf{x}')} \tag{10}$$

is the partition function.

The learning process of a Boltzmann Machine consists in ajusting the connections ω_{ij} in such a way that the state of the visible units have a particular desired probability distribution.

Let us identify the state of the visible units by an index v and the state of the hidden units by an index h. Considering a system which has N visible units and K hidden units, the whole system have 2^{N+K} possibilities of states in which it can be found.

The joint probability P_{vh} is the probability of finding the visible and hidden units in the states v and h, respectively. This probabiblity measument is given by the Boltzmann distribution:

$$P_{vh} = \frac{e^{-\beta H_{vh}}}{Z}, \tag{11}$$

where

$$Z = \sum_{u} \sum_{k} e^{-\beta H_{uk}}, \tag{12}$$

and

$$H_{vh} = -\sum_{i} \sum_{j} \omega_{ij} x_i^{(vh)} x_j^{(vh)} - \sum_{i} \phi_i x_i^{(vh)}. \tag{13}$$

As metioned above, the problem a Boltzmann Machine is trying to solve is determining the connections ω_{ij} between units such that the visible units have a certain probability distribution. In order to do that, we need to find the marginal probability of the state v in which the visible units are found regardless of the state h of the hidden units. The marginal probability P_v is given by

$$P_v = \sum_h P_{vh} = \sum_h \frac{e^{-\beta H_{vh}}}{Z}. \tag{14}$$

Although we know that P_v is a function of the connections ω_{ij}, and that this is the probability of finding the visible units in the state v. We want the states to have a certain probability R_v, i.e., a desired probability. This means that ideally we would like to match the empirical distribution of the data, even though we do not have access to the correct distribution, only to what the observed data has given us as an input to training the model.

One way to evaluate the difference between two probability distribution, for example, P_v and R_v, is using the Kullback-Leibler divergence, which can also be referred to relative entropy, E, which will be our cost function. (EXPLICAR FUNÇÃO DE CUSTO e D_{KL}!!!).

$$E = \sum_v R_v \ln\left(\frac{R_v}{P_v}\right). \tag{15}$$

The relative entropy E has the property of always being equal or greater than zero. It reaches zero only if $P_v = R_v$, which means that we are able to retrieve the exactly probability distribution of the input data at the visible units.

$$\begin{aligned}
E &= \sum_v R_v \ln\left(\frac{R_v}{P_v}\right) \\
&\geq \sum_v R_v \left(1 - \frac{P_v}{R_v}\right) \\
&= \sum_v (R_v - P_v) \\
&= \sum_v R_v - \sum_v P_v = 1 - 1 \\
&\Rightarrow E \geq 0.
\end{aligned} \tag{16}$$

From the gradient descent equation

$$\Delta\omega_{ij} = -\eta \frac{\partial E}{\partial \omega_{ij}}, \tag{17}$$

where

$$\begin{aligned}
E &= \sum_v R_v \ln\left(\frac{R_v}{P_v}\right) \\
&= \sum_v \left[\ln(R_v) - \ln(P_v)\right].
\end{aligned} \tag{18}$$

In the following steps, we present the gradient descent derivation

$$\begin{aligned}\Delta\omega_{ij} &= -\eta\frac{\partial E}{\partial\omega_{ij}}\\ &= -\eta\frac{\partial}{\partial\omega_{ij}}\left[\sum_v R_v\left(\ln\left(R_v\right)-\ln\left(P_v\right)\right)\right]\\ &= \eta\frac{\partial}{\partial\omega_{ij}}\left[\sum_v R_v\ln\left(P_v\right)\right]\\ &= \eta\sum_v R_v\frac{\partial}{\partial\omega_{ij}}\left[\ln\left(P_v\right)\right]\\ \Rightarrow\Delta\omega_{ij} &= \eta\sum_v\frac{R_v}{P_v}\frac{\partial P_v}{\partial\omega_{ij}}.\end{aligned} \tag{19}$$

To continue with the computation of $\Delta\omega_{ij}$, we have to find the derivative of $\partial P_v/\partial\omega_{ij}$, from the marginal probability, equation 14,

$$P_v = \frac{\sum_h e^{-\beta H_{vh}}}{\sum_u\sum_k e^{-\beta H_{uk}}}, \tag{20}$$

thus the derivative of P_v follows

$$\begin{aligned}\frac{\partial P_v}{\partial\omega_{ij}} &= \frac{\partial}{\partial\omega_{ij}}\left[\frac{\sum_h e^{-\beta H_{vh}}}{\sum_u\sum_k e^{-\beta H_{uk}}}\right]\\ &= \frac{1}{\sum_u\sum_k e^{-\beta H_{uk}}}\sum_h(-\beta)e^{-\beta H_{vh}}\frac{\partial H_{vh}}{\partial\omega_{ij}}\\ &-\sum_h e^{-\beta H_{vh}}\frac{1}{(\sum_u\sum_k e^{-\beta H_{uk}})^2}\sum_u\sum_k e^{-\beta H_{uk}}(-\beta)\frac{\partial H_{uk}}{\partial\omega_{ij}}.\end{aligned} \tag{21}$$

Following the equation 21, we need to compute the term $\partial H_{vh}/\partial\omega_{ij}$,

$$\begin{aligned}\frac{\partial H_{vh}}{\partial\omega_{ij}} &= \frac{\partial}{\partial\omega_{ij}}\left[-\frac{1}{2}\sum_m\sum_n\omega_{mn}x_m^{(vh)}x_n^{(vh)}-\sum_m\phi_{mm}x_m^{(vh)}\right]\\ &= \frac{\partial}{\partial\omega_{ij}}\left[-\frac{1}{2}\sum_{m\neq i,j}\sum_{n\neq i,j}\omega_{mn}x_m^{(vh)}x_n^{(vh)}-\frac{1}{2}\omega_{ij}x_i^{(vh)}x_j^{(vh)}-\frac{1}{2}\omega_{ji}x_j^{(vh)}x_i^{(vh)}-\sum_m\phi_m x_m^{(vh)}\right],\end{aligned} \tag{22}$$

as the connections between units are symmetric, i.e., $\omega_{ij} = \omega_{ji}$, then we can simplify equation 22,

$$\begin{aligned}
\frac{\partial H_{vh}}{\partial \omega_{ij}} &= \frac{\partial}{\partial \omega_{ij}}\left[-\frac{1}{2}\sum_{m\neq i,j}\sum_{n\neq i,j}\omega_{mn}x_m^{(vh)}x_n^{(vh)} - \omega_{ij}x_i^{(vh)}x_j^{(vh)} - \sum_m \phi_m x_m^{(vh)}\right]\\
&= \frac{\partial}{\partial \omega_{ij}}\left[-\frac{1}{2}\sum_{m\neq i,j}\sum_{n\neq i,j}\omega_{mn}x_m^{(vh)}x_n^{(vh)}\right] + \frac{\partial}{\partial \omega_{ij}}\left[-\omega_{ij}x_i^{(vh)}x_j^{(vh)}\right] + \frac{\partial}{\partial \omega_{ij}}\left[-\sum_m \phi_m x_m^{(vh)}\right]\\
\Rightarrow &\frac{\partial H_{vh}}{\partial \omega_{ij}} = -x_i^{(vh)}x_j^{(vh)}.
\end{aligned} \tag{23}$$

Analagous to $\partial H_{vh}/\partial \omega_{ij}$, we have H_{uk} derivative, which is

$$\frac{\partial H_{uk}}{\partial \omega_{ij}} = -x_i^{(uk)}x_j^{(uk)}. \tag{24}$$

Going back to equation 21, we can replace the derivatives of H, and solve the derivative of the marginal probability P_v,

$$\begin{aligned}
\frac{\partial P_v}{\partial \omega_{ij}} &= \frac{1}{Z}\sum_h e^{-\beta H_{vh}}(-\beta)(-x_i^{(vh)}x_j^{(vh)}) - \frac{1}{Z^2}\sum_h e^{-\beta H_{vh}}\sum_u\sum_k e^{-\beta H_{uk}}(-\beta)(-x_i^{(uk)}x_j^{(uk)})\\
&= \beta\left[\sum_h \frac{e^{-\beta H_{vh}}}{Z}x_i^{(vh)}x_j^{(vh)} - \sum_h \frac{e^{-\beta H_{vh}}}{Z}\sum_u\sum_k \frac{e^{-\beta H_{uk}}}{Z}x_i^{(uk)}x_j^{(uk)}\right]\\
&= \beta\left[\sum_h P_{vh}x_i^{(vh)}x_j^{(vh)} - P_v\sum_u\sum_k P_{uk}x_i^{(uk)}x_j^{(uk)}\right]\\
\Rightarrow &\frac{\partial P_v}{\partial \omega_{ij}} = \beta\left[\sum_h P_{vh}x_i^{(vh)}x_j^{(vh)} - P_v\langle x_i x_j\rangle\right].
\end{aligned} \tag{25}$$

Given the derivative of P_v in relation to ω_{ij}, we can compute the learning term $\Delta\omega_{ij}$, from equation 19,

$$\begin{aligned}
\Delta\omega_{ij} &= \eta\sum_v \frac{R_v}{P_v}\beta\left[\sum_h P_{vh}x_i^{(vh)}x_j^{(vh)} - P_v\langle x_i x_j\rangle\right]\\
&= \eta\beta\left[\sum_v\sum_h \frac{R_v}{P_v}P_{vh}x_i^{(vh)}x_j^{(vh)} - \sum_v \frac{R_v}{P_v}P_v\langle x_i x_j\rangle\right]\\
&= \eta\beta\left[\sum_v\sum_h R_v\frac{P_{vh}}{P_v}x_i^{(vh)}x_j^{(vh)} - \sum_v R_v\langle x_i x_j\rangle\right]\\
&= \eta\beta\left[\sum_v\sum_h R_v P_{h|v}x_i^{(vh)}x_j^{(vh)} - \langle x_i x_j\rangle\right].
\end{aligned} \tag{26}$$

In the above derivation, we have used the following relations,

$$P_{h|v} = \frac{P_{vh}}{P_v}, \tag{27}$$

which is the conditional probability equation. In our scenario this equation means that the probability distribution of the hidden units in state h given the state v of the visible units is the joint probability distribution of both states, v and h, divided by the marginal probability distribution of the visible units in state v.

The second term in equation 26, is the average of units i and j over all combinations of states v and h of the system. In other words, we would have to compute all possible combination of states of visible and hidden units, v and h, and then average over the specific units i and j.

The first term can be simplified by

$$\sum_v R_v \sum_h P_{h|v} x_i^{(vh)} x_j^{(vh)} = \sum_v R_v \langle x_i x_j \rangle^{(v)} = \langle \langle x_i x_j \rangle^{(v)} \rangle. \tag{28}$$

Then equation 26 becomes

$$\Delta\omega_{ij} = \eta\beta \left[\langle \langle x_i x_j \rangle^{(v)} \rangle_{clamped} - \langle x_i x_j \rangle_{free} \right], \tag{29}$$

it is important to notice that the subscripts *clamped* means that we have to fix a certain v state on the visible units otherwise the second term in the equation does not have a reference. On the other hand, the subscript *free* identify the ...

2 TÍTULO DO CAPÍTULO 1

Texto do capítulo. Texto, texto, Figura 3. Texto Figura 4(a).

Figure 3 - Título da figura.

Legend: Texto da legenda.

Source: Citação da fonte ou ‘O autor.’.

Table 1 - Título da tabela.

X	Y
1,20	15,7
1,23	15,6
1,19	15,3
1,26	15,1
1,22	15,5
1,16	15,3
1,37	15,7

Legend: Texto da legenda.

Source: Citação da fonte ou ‘O autor.’.

Figure 4 - Título da figura.

UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO
UERJ
UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO
UERJ
UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO
UERJ

(a)

(b)

(c)

Legend: Texto da legenda. (a) Texto da imagem. (b) Texto da imagem. (c) Texto da imagem.

Source: Citação da fonte ou ‘O autor’.

3 TÍTULO DO CAPÍTULO 2

Texto do capítulo. Texto, texto Algoritmo 1. Texto.

Algoritmo 1 - Título do algoritmo.

Documentação

Título

Nome do algoritmo

Propósito

Propósito do algoritmo.

Método

Método utilizado no algoritmo.

Entradas

a, m: multiplicador e módulo
n0: semente
i: contador auxiliar

Saídas

n: número aleatório

Observações, Restrições, Requisitos

Observações, restrições e requisitos.

```
algoritmo Identificação
    declarar $a, m, i$ numéricos
    declarar $n0, n$ numéricos
    $m \leftarrow 13$
    $n0 \leftarrow 1$
    para $a$ de 2 até $m - 1$ , fazer      {para cada possível valor de 'a'}
    |   escrever "a = ", $a$, ": n = {"
    |   $n \leftarrow n0$      {reinicia a geração com a semente n0}
    |   para $i$ de 0 até $m - 1$ , fazer
    |   |   $n \leftarrow resto(a * n, m)$      {gerador de números aleatórios}
    |   |   se $(n == n0)$, então      {se fim da sequencia ...}
    |   |   |   escrever $n$,"}"
    |   |   |   parar
    |   |   senão
    |   |   |   escrever $n$
    |   |   fim se
    |   fim para
    fim para
```

— *continua* —

Algoritmo 1 - Título do algoritmo. (continuação)

```
    — continuação —
    $a \leftarrow 1$
    **enquanto** ($a < 10$), **fazer**     {*comentário*}
     |   **escrever** $a$
     |   $a \leftarrow a + 1$
    **fim enquanto**
    $a \leftarrow 1$
    **repetir**     {*comentário*}
     |   **escrever** $a$
     |   $a \leftarrow a + 1$
    **até que** ($a \geq 10$)

    $a \leftarrow 1$
    **fazer**     {*comentário*}
     |   **escrever** $a$
     |   $a \leftarrow a + 1$
    **enquanto** ($a < 10$)
  FIM ALGORITMO
FIM DOCUMENTAÇÃO
```

CONCLUSÃO

Texto da conclusão.

BIBLIOGRAPHY

GÉRON, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. 1. ed. USA: O'Reilly Media Inc., 2017.

HERTZ, J.; KROGH, A.; PALMER, R. D. *Introduction to the theory of neural computation*. USA: Westview Press, 1991. (Santa Fe Institute Series).

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, v. 79, p. 2554–2558, Apr 1982.

APPENDIX A – Kullback-Leibler Divergence or Relative Entropy

A.1 **Kullback-Leibler Divergence**

The Kullback-Leibler divergence is the measure of the difference between two probability distributions.