

# CLASSIFICAÇÃO DE PADRÕES UTILIZANDO APRENDIZAGEM DE BOLTZMANN

Gustavo Bruno do Vale, Gilberto Arantes Carrijo  
Universidade Federal de Uberlândia. Faculdade de Engenharia Elétrica, Uberlândia-MG  
gbvale@reito.ufu.br, gilberto@ufu.br

**Resumo** - Este trabalho apresenta um algoritmo que realiza reconhecimento de padrões utilizando um método de aprendizagem estocástica conhecida com Máquina de Boltzmann. Este algoritmo possui a estrutura parecida com Redes Neurais Artificiais (RNA) utilizando um processo de reconhecimento parecido com o aprendizado de neurônios biológicos. Além disso, é aplicado um conceito da termodinâmica para se alcançar o equilíbrio de uma rede através do abaixamento da temperatura de forma a minimizar a energia do sistema, conhecido como Recozimento Simulado. Aplicou-se testes com uma rede de neurônios artificiais e através da simulação deste algoritmo, concluiu-se a eficácia da rede em reconhecer padrões.

**Palavras-Chave** - Boltzmann, reconhecimento de padrões, recozimento simulado, RNA.

## STANDARDS CLASSIFICATION USING LEARNING BOLTZMANN

**Abstract** - This paper presents an algorithm that performs pattern recognition using a stochastic learning method known as Boltzmann Machine. This algorithm has the similar structure to Artificial Neural Networks (ANN) using a process of recognition like the learning of biological neurons. Furthermore, a concept is applied to achieve the thermodynamic equilibrium of a network by lowering the temperature in order to minimize the system power, known as Simulated Annealing. Tests were applied with a network of artificial neurons and through the simulation of this algorithm, it was concluded the effectiveness of the network to recognize patterns.

**Keywords** - Boltzmann, pattern recognition, simulated annealing, ANN.

### NOMENCLATURA

RNA    Redes Neurais Artificiais.  
BM    Boltzmann Machine.

## I. INTRODUÇÃO

Cada vez mais necessita-se de sistemas que automatizem o reconhecimento de pessoas, de assinaturas e de outros tipos de padrões facilitando, agilizando e aumentando a segurança deste tipo de processo. Para atender esta demanda são utilizadas várias formas de padrões para posteriormente o sistema realizar o reconhecimento, tais como: reconhecimento de faces, identificação através da íris do olho, reconhecimento de escritas cursivas, reconhecimento de fala, entre outros.

Existem vários métodos de reconhecimentos de padrões que são aplicados nestes sistemas. Neste artigo será aplicado um método de aprendizagem estocástica conhecida com Máquina de Boltzmann (BM).

O método aplicado neste trabalho utilizará uma estrutura como das Redes Neurais Artificiais (RNA) porém com diferenças nos valores dos neurônios desta rede. As RNAs são aplicadas em alguns sistemas de reconhecimento de padrões, onde a estrutura desta rede utiliza um método parecido com o aprendizado de um neurônio biológico.

Porém, neste artigo o método conhecido com Máquina de Boltzmann utiliza esta estrutura de RNA com a diferença dos neurônios terem valores binários e aleatórios. Estes valores são modificados e os pesos da rede são ajustados até que se encontre valores que minimizem a função de energia da rede. Será aplicado algoritmos da Máquina Boltzmann que é um conceito da termodinâmica que realiza o abaixamento da temperatura de um sistema até que ocorra um equilíbrio de energia.

Para se testar a eficiência do algoritmo em realizar o reconhecimento de padrões foi aplicado diferentes algoritmos que buscam criar uma rede que otimize este processo de reconhecimento. Estes algoritmos foram testados através do software *Matlab* treinando uma rede, a diferença entre os algoritmos é a forma como os pesos são atualizados. Feito o treinamento da entrada da rede com sua respectiva saída, o algoritmo realizará a classificação da saída ideal de acordo com a entrada. Desta forma, pode-se mensurar a eficiência do sistema em realizar um reconhecimento de um padrão.

## II. MÁQUINA DE BOLTZMANN

No início de 1980 Geoffrey Hinton e Terry Sejnowski desenvolveu o conceito original de uma máquina de



XIII CEEL - ISSN 2178-8308  
12 a 16 de Outubro de 2015  
Universidade Federal de Uberlândia - UFU  
Uberlândia - Minas Gerais - Brasil

Boltzmann conhecida com BM (*Boltzmann Machine*). A máquina de Boltzmann é uma máquina de estados, onde há modificações ou repetições de certos eventos até que se chegue a um estado considerado estável. Quando esta máquina se estabiliza adotando um critério de parada, diz-se que ela aprendeu aquele padrão[3][4].

Em 1982, Hopfield propôs um tipo de Rede Neural Artificial (RNA) onde se as unidades da rede fossem simetricamente conectadas  $w_{ij} = w_{ji}$  e se os pesos fossem atualizados simultaneamente no tempo, seria possível a simulação da redução de energia num sistema físico, conhecida com entropia, ajustando-o para modelagem de determinado padrão. A Máquina de Boltzmann é uma generalização da rede de Hopfield com a diferença de seguir uma regra de decisão probabilística[5].

Para entender esta rede, primeiramente é necessário entender o funcionamento de uma Rede Neural Artificial conhecida como RNA. Uma RNA é uma rede que baseia o aprendizado em neurônios biológicos. Cada neurônio artificial recebe sinais de entrada que são multiplicados por pesos, esses produtos são somados e valor resultante é aplicado a uma função de ativação que tomará decisão do valor de saída. Quando vários neurônios são conectados conforme a figura 1, forma-se uma RNA com neurônios na camada de entrada, neurônios ocultos nas camadas intermediárias e neurônios na camada de saída. Através desta rede é possível a aprendizagem e classificação de padrões.

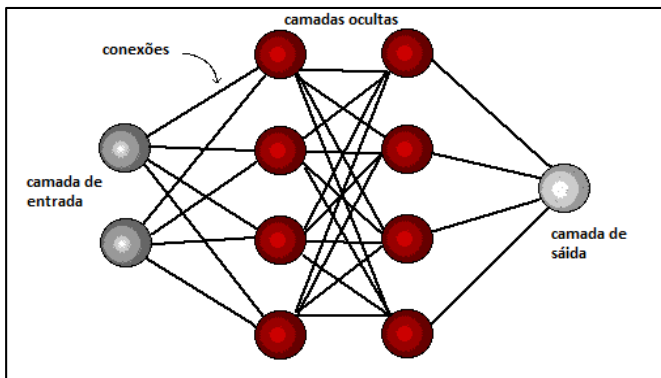


Fig. 1. Rede Neural Artificial.

Na arquitetura da Máquina de Boltzmann, existem similaridades com a RNA, pois também possui unidades chamadas neurônios, unidades de entrada, unidades ocultas e unidades de saída.

A máquina de Boltzmann é uma rede de unidades binárias estocásticas simetricamente acopladas[3]. O sistema tem uma camada de unidades binárias visíveis e uma camada de unidades binárias escondidas. Cada unidade tem uma ligação bidirecional com outra unidade no sistema. O objetivo deste equipamento é o de ser capaz de estimar as probabilidades que ambas as unidades ligadas teriam. Isto é feito, por permitir que a máquina resolva a distribuição próxima do equilíbrio. A ideia geral é a de ajustar os pesos de modo que as inicializações aleatórias resolvam os estados semelhantes aos que existem no meio ambiente. [2]

Se esta regra é aplicada repetidamente visando o ajuste da rede a algum padrão de entrada, ela chegará a um ponto em que se tornará estável, ou seja, não haverá mais trocas de

valores nas unidades, que será possível o aprendizado de um padrão.

A rede será dita tendo um modelo perfeito do ambiente se atingir exatamente a mesma distribuição de probabilidade sobre esses dois estados quando está funcionando livremente em equilíbrio térmico com nenhuma entrada do meio ambiente [5]. Embora o sistema nunca possa ser um modelo perfeito, regularidades nos dados podem ser capturados para dar uma estimativa perto da probabilidade real.

A arquitetura da máquina de Boltzmann é apresentada na Fig. 2 à seguir.

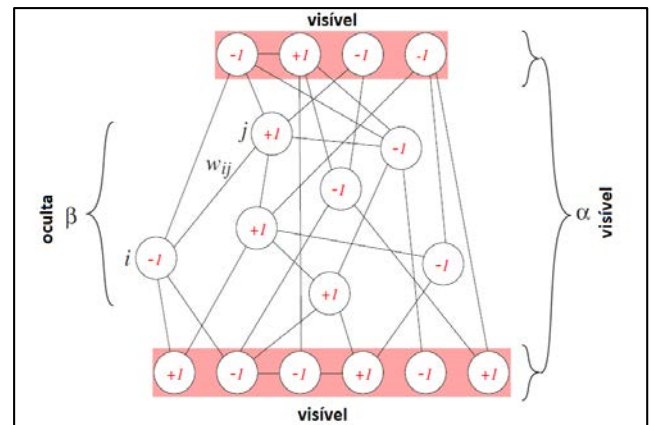


Fig. 2. Arquitetura da Máquina de Boltzmann.

Nesta rede, os valores das unidades ou neurônios binários são chamados de  $S_i$  e os pesos são denominados  $w_{ij}$ , sendo  $i$  e  $j$  os neurônios da rede. Para distinguir as unidades ou neurônios de entrada e saída da rede, utiliza-se os símbolos  $\alpha^i$  para unidades de entrada,  $\alpha^o$  para as unidades de saída e  $\beta$  para unidades ocultas [1]. Para se atingir o equilíbrio da rede é necessário realizar a otimização dos valores de  $S_i$  e ajustar os pesos  $w_{ij}$ , de modo a minimizar função custo ou energia  $E$  conforme expressa na equação 1 abaixo:

$$E = -\frac{1}{2} \sum_{i,j=1}^N w_{ij} S_i S_j \quad (1)$$

Onde:

- $w_{ij}$  - Pesos que conectam os nós  $i$  e  $j$ .
- $S_i$  - Valor do nó  $i$  selecionado.
- $E$  - Energia do sistema ou função custo.
- $N$  - Quantidade de neurônios na rede.

O algoritmo Máquina de Boltzmann será capaz de ajustar todos os valores dos neurônios binários em -1 ou +1 de forma que o valor  $E$  da equação 1 seja o menor possível. Para realizar esta otimização utiliza-se o algoritmo descrito no item III deste artigo.

### III. RECOZIMENTO SIMULADO

No aprendizado de Boltzmann, para otimizar ou reduzir a energia do sistema e fugir de soluções com mínimos locais, é utilizado a técnica ou algoritmo de recozimento simulado. No campo da termodinâmica, o recozimento é uma técnica em que o sistema inicia com alta temperatura e esta temperatura é

abaixada gradualmente até se atingir um equilíbrio e uma baixa energia do sistema.

O algoritmo da Máquina de Boltzmann, utiliza este mesmo princípio sendo repetido os estados várias vezes abaixando a temperatura até se alcançar um limiar escolhido. Cada vez que um estado é repetido ocorre a atualização dos pesos da rede e são modificados os valores dos neurônios de forma a otimizar a rede[1].

Será descrito a seguir alguns algoritmos que realizam a técnica do Recozimento Simulado, sendo uma técnica estocástica e a outra técnica determinística.

#### 1) Algoritmo recozimento simulado estocástico

Um dos algoritmos para realizar o recozimento simulado é conhecido com Recozimento Simulado Estocástico. Neste algoritmo inicializa-se com uma temperatura  $T(1)$  alta, com estados aleatórios  $S_i(1)$  e com um valor  $k_{max}$  que será a quantidade de vezes que a temperatura  $T$  será abaixada ou o algoritmo será realizado. Seleciona-se aleatoriamente um nó  $i$  da rede que poderá ser atualizado de acordo com a regra abaixo. Suponha-se que o estado  $S_i = +1$ , então calcula-se a energia  $E_a$  do sistema nesta configuração; depois recalcula-se a energia  $E_b$  para o novo estado  $S_i = -1$ . Se este estado candidato tiver energia menor, aceita-se esta mudança de estado. Caso a energia seja maior, aceita-se a mudança com a probabilidade igual a:

$$e^{-\Delta E_{ab}/T} \quad (2)$$

Onde:

$$\Delta E_{ab} = E_b - E_a$$

T - Temperatura.

Este algoritmo realiza a seleção e teste várias vezes dos nós  $i$  da rede sempre abaixando a temperatura. O recozimento simulado termina quando a temperatura está muito baixa (definido através do valor de  $k$ ). A figura 3 a seguir descreve os passos do algoritmo explicado.

```

1 Inicializar T(k),  $k_{max}$ ,  $S_i(1)$ ,  $w_{ij}$  para  $i, j = 1, \dots, N$ 
2    $k \leftarrow 0$ 
3   do  $k \leftarrow k + 1$ 
4     do selecionar nó i aleatoriamente; estado  $s_i$ 
5        $E_a \leftarrow -\frac{1}{2} \sum_{i,j=1}^N w_{ij} S_i S_j$ 
6        $E_b \leftarrow E_a$ 
7       if  $E_b < E_a$ 
8         then  $S_i \leftarrow -S_i$ 
9         else if  $e^{-(E_b - E_a)/T(k)} > \text{rand}(0,1)$ 
10          then  $S_i \leftarrow -S_i$ 
11          until
12            until  $k = k_{max}$  ou encontrar critério de parada
13          return E,  $S_i$ , for  $i=1, \dots, N$ 
14        end

```

Fig. 3. Algoritmo de recozimento simulado estocástico.

Caso o resfriamento seja suficiente, o sistema terá grande probabilidade de estar em baixo estado de energia, atingindo o mínimo global de energia.

#### 2) Recozimento simulado determinístico (Algoritmo 2)

Um outro algoritmo utilizado para realizar o recozimento é chamado de determinístico porque em princípio é possível deterministicamente resolver as equações para se encontrar o valor de  $S_i$  quando a temperaturas é reduzida.

Neste método de recozimento determinista, seleciona-se aleatoriamente um nó  $i$  da rede, depois calcula-se as forças exercidas dos nós conectados a este nó  $i$  e este somatório é chamado  $l_i$  conforme a equação 3 a seguir:

$$l_i = \sum_j w_{ij} S_j \quad (3)$$

Onde:

$w_{ij}$  - Pesos que conectam os nós  $i$  e  $j$ .

$S_i$  - Valor do nó  $i$  selecionado.

O valor de  $S_i$  a ser atualizado dependerá deste somatório  $l_i$  e da temperatura em andamento. Sendo aplicada a equação (4), calculando a tangente hiperbólica de  $l_i/T$ .

$$s_i = f(l_i, T) = \tanh[l_i/T] \quad (4)$$

Onde:

$w_{ij}$  - Pesos que conectam os nós  $i$  e  $j$ .

$S_i$  - Valor do nó  $i$  selecionado.

$l_i$  - Somatório explicado na equação 3.

Os novos valores de  $S_i$  selecionados serão substituídos na rede seguindo os valores determinados na equação 4. Estes passos do algoritmo são descritos na figura 4.

```

1 Inicializar T(k),  $k_{max}$ ,  $w_{ij}$ ,  $S_i(1)$ ,  $i, j = 1, \dots, N$ 
2    $k \leftarrow 0$ 
3   do  $k \leftarrow k + 1$ 
4     selecionar nó i aleatoriamente; estado  $s_i$ 
5      $l_i = \sum_j w_{ij} S_j$ 
6      $s_i = f(l_i, T) = \tanh[l_i/T]$ 
7     until  $k = k_{max}$  ou encontrar critério de convergência
8     return E,  $S_i$ ,  $i = 1, \dots, N$ 
9   end

```

Fig. 4. Algoritmo de recozimento simulado determinístico.

Ao final deste processo, também espera-se encontrar valores de  $S_i$  que reduzam a energia total do sistema.

## IV. APRENDIZAGEM DE BOLTZMANN

No algoritmo de Boltzmann, o recozimento é utilizado para diminuir a energia do sistema, podendo ser qualquer um dos dois métodos explicados. Porém para realizar reconhecimento de padrões é necessário manter fixo alguns valores dos neurônios  $S_i$  e realizar o recozimento dos demais, este algoritmo será descrito nos próximos itens deste artigo.

#### 1) Algoritmo de Aprendizagem de Boltzmann

No algoritmo de aprendizagem de Boltzmann, é necessário inicializar com os valores de taxa de aprendizagem  $\eta$ , temperatura inicial  $T(k)$  e pesos  $w_{ij}$ . Inicialmente define-se aleatoriamente os estados  $S_i$ . O próximo passo será manter fixo os neurônios de entrada e saída a serem treinados, e realizar o recozimento dos demais neurônios que são os neurônios da

camada oculta. No final deste passo, é necessário abaixar a temperatura  $T$  e calcular o produto  $S_i S_j$ .

Novamente, define-se de forma aleatória os estados  $S_i$ . Inicia-se então outro processo de recozimento, desta vez mantendo fixo apenas a camada de entrada da rede e realizando o recozimento da camada oculta e camada de saída. Abaixa-se novamente a temperatura  $T$  do sistema e calcula-se o produto  $S_i S_j$ .

Através dos produtos  $S_i S_j$  calculados nos dois processos, da taxa de aprendizagem  $\eta$  e da temperatura  $T$ , atualiza-se os pesos da rede seguindo a seguinte equação 5.

$$w_{ij} = w_{ij} + \eta/T \left[ [S_i S_j]_{\alpha^i \alpha^o \text{ fixas}} - [S_i S_j]_{\alpha^i \text{ fixa}} \right] \quad (5)$$

Onde:

- $w_{ij}$  - Pesos que conectam os nós  $i$  e  $j$ .
- $S_i$  - Valor do nó  $i$  selecionado.
- $\eta$  - Taxa de aprendizagem.
- $T$  - Temperatura.
- $\alpha^i$  - Unidades de entrada da rede.
- $\alpha^o$  - Unidades de saída da rede.

Este processo de atualização será realizado  $k$  vezes, sendo  $k$  a quantidade de vezes que a temperatura é reduzida. A figura 5 descreve este algoritmo explicado. Ao final do processo, a rede terá os pesos  $w_{ij}$  atualizados e os valores  $S_i$  dos neurônios terá valores que tendem a abaixar a energia do sistema e manter a rede estável.

```

1 Inicializar D, η, T(k), wij, i, j = 1, ..., N
2 do Selecionar aleatoriamente padrão de treinamento x
3 Estados aleatórios si
4 Recozimento da rede com entradas e saídas fixas
5 No final, abaixar T, calcular [SiSj]αiαo fixos
6 Estados aleatórios si
7 Recozimento com entradas fixas e saídas livres
8 No final, abaixar T, calcular [SiSj]αi fixos
9 wij ← wij + η/T [ [SiSj]αiαo fixos - [SiSj]αi fixos ]
10 until k = kmax ou encontrar critério de convergência
11 return wij
12 end

```

Fig. 5. Algoritmo determinístico de Boltzmann.

Neste processo, a rede formada terá valores com grande probabilidade de realizar reconhecimento de padrões. Assim, com os valores de entrada e saída treinados na rede, será possível realizar a classificação de padrões de acordo com a entrada.

## V. METODOLOGIA

Neste trabalho os algoritmos descritos nos tópicos anteriores, foram simulados através do software *Matlab*. Para se testar a capacidade da rede em reconhecer padrões, criou-se uma rede com 7 neurônios, sendo 2 neurônios na camada de entrada, 3 neurônios na camada oculta e 2 neurônios na camada de saída. Para testar o processo de aprendizagem, as saídas  $S_6$  e  $S_7$ , foram igualadas às entradas  $S_1$  e  $S_2$  respectivamente, conforme ilustrado na figura 7.

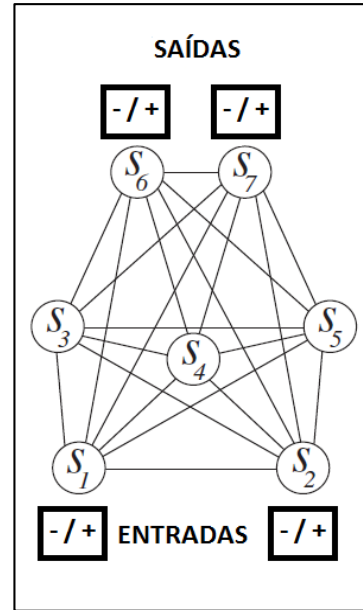


Fig. 6. Rede testada com 7 unidades sendo 2 unidades de entrada, 2 unidades de saída e 3 unidades ocultas.

No processo de reconhecimento de padrões, espera-se que com a rede treinada, as saídas  $S_6$  e  $S_7$  possam ser encontradas apenas fornecendo os valores  $S_1$  e  $S_2$  de entrada. Desta forma, a rede será capaz de reconhecer um padrão treinado e selecioná-lo na saída. Esta capacidade da rede dependerá do ajuste dos pesos que será realizado através dos processos de aprendizagem e dos algoritmos descritos no item anterior.

O processo utilizado nesta rede pode ser aplicado em processos mais complexos. Por exemplo, pode-se treinar uma rede onde a entrada são dígitos de 0 a 9, e posteriormente esta rede será capaz de reconhecer um dígito qualquer e apresentá-lo na camada de saída.

## VI. RESULTADOS

Conforme mencionado, os algoritmos descritos nos tópicos acima, foram simulados através do software *Matlab*. Primeiramente foram testados os algoritmos de recozimento para abaixar a energia total do sistema. Criou-se uma função que gera os valores de  $S_i$ , sendo que foi gerado todas as configurações possíveis de valores -1 e +1 para uma rede de 7 neurônios. Através do algoritmo força bruta que testa a rede com todas as configurações possíveis, encontrou-se os valores de  $S_i$  que mantem a rede com a menor energia total. Porém este método foi utilizado apenas para testar o algoritmo de recozimento simulado, pois seria inviável em rede com grande quantidade de neurônios.

No algoritmo determinístico de recozimento explicado no item III – 2, foi criada uma rede de 7 neurônios, foi escolhida uma temperatura inicial  $T(1)=1$  e esta temperatura foi reduzida  $k$  vezes com  $k = 20$  seguindo a seguinte expressão a cada passo:  $T(k)=T(1)*0,9^k$ . No algoritmo testado encontrou-se o valor  $S_i$  dos neurônios que minimizam a energia do sistema testando o algoritmo força bruta e o valor  $S_i$  dos neurônios utilizando o algoritmo recozimento simulado. Observou-se que o algoritmo de recozimento simulado teve um resultado satisfatório pois teve uma saída parecida com a do algoritmo de força bruta.

Nota-se que a energia do sistema foi reduzida a cada interação que foi reduzida a temperatura do sistema e consequentemente atualizado o valor dos neurônios  $S_i$ . Isto pode ser confirmado através do gráfico da figura 7 em cada uma das 20 interações.

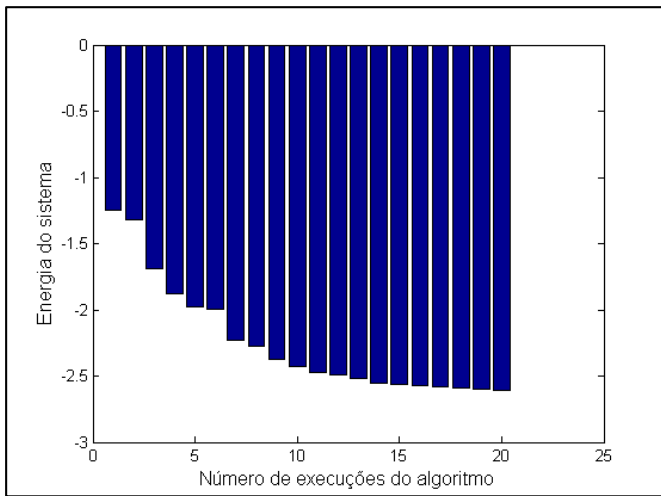


Fig. 7. Gráfico da energia do sistema em relação a cada execução do algoritmo de recozimento simulado.

No algoritmo determinístico de recozimento simulado a energia do sistema foi calculada  $k$  vezes, totalizando 20 execuções. A energia mínima encontrada foi -2,60 e a configuração de neurônios encontrada foi:

$$s = -1 \ -1 \ -0.561 \ -0.989 \ 1 \ 1 \ -0.999$$

Já no algoritmo força bruta a energia do sistema foi calculada  $2^N$  vezes, sendo  $N$  o número de neurônios, totalizando 128 execuções. A energia mínima encontrada foi -2,64 e a configuração de neurônios encontrada foi:

$$s = -1 \ -1 \ -1 \ -1 \ 1 \ 1 \ -1$$

Nota-se nestes dois casos, que a energia mínima encontrada no sistema foi muito próxima, porém com o algoritmo de recozimento simulado, a solução de otimização é encontrada em apenas 20 execuções enquanto que de outro modo testando todas as opções da rede deveríamos executar 1 vez, para encontrar a rede que otimizaria a energia.

Além disso, também foi testado o algoritmo estocástico de recozimento simulado explicado no item III – 1 que teve uma saída parecida com algoritmo determinístico.

Por fim, foi criado um algoritmo que implementa o método de aprendizagem de Boltzmann conforme o item IV deste artigo. Para isto, criou-se uma rede com 7 neurônios, sendo 2 neurônios na camada de entrada, 3 neurônios na camada oculta e 2 neurônios na camada de saída. Para testar o processo de aprendizagem, as saídas  $S_6$  e  $S_7$ , foram igualadas às entradas  $S_1$  e  $S_2$  respectivamente, conforme ilustrado na figura 6.

Neste algoritmo desenvolvido manteve-se inicialmente os neurônios de entrada e saída fixos e através do algoritmo de recozimento simulado e calculou-se os novos valores dos neurônios  $S_i$ . Em seguida foi calculado o produto  $S_i * S_j$  denominado  $S1(i,j)$ , sendo  $i$  e  $j$  todos os 7 neurônios da rede.

Na próxima etapa, manteve-se fixo apenas os valores da entrada e novamente calculou-se os novos valores dos neurônios  $S_i$ . Novamente foi calculado o produto  $S_i * S_j$  denominado  $S2(i,j)$ . Ao final deste processo encontrou-se os novos valores dos pesos da rede através da equação (5) já apresentada.

Com a rede encontrada com novos valores dos pesos  $w_{ij}$  e valores dos neurônios ocultos  $S_i$ , foi possível testar a capacidade da rede em reconhecer um padrão de entrada. Esperava-se que as saídas  $S_6$  e  $S_7$ , fossem respectivamente iguais às entradas  $S_1$  e  $S_2$ . Com 4 combinações diferentes de entradas binárias, a rede conseguiu ter um acerto na camada de saída variando entre 3 e 4 acertos, ou seja, de 75% a 100%.

## VII. CONCLUSÕES

Este trabalho desenvolvido mostra a importância desta técnica de Boltzmann, pois se mostrou eficiente em reconhecer padrões. O trabalho apresentou teste com poucas variáveis, porém pode ser aplicado em problemas bem mais complexos.

Além disso, a técnica utilizada de recozimento simulado auxilia na diminuição da carga computacional para situações que exigem muitas variáveis de entrada para o reconhecimento. Problemas tais como: reconhecimento de faces, preenchimento de padrões defeituosos, reconhecimento de íris, identificação de escrita e demais padrões. Estas situações problemas podem ser aplicadas em trabalho futuros.

## REFERÊNCIAS

- [1] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, John Wiley & Sons Inc., 2ª Edição, Nova York, 2001.
- [2] S. N. Sivanandam, S. Sumathi, *Introduction to Neural Networks using Matlab 6.0*, Tata McGraw-Hill Education, New Delhi, 2006.
- [3] R. Salakhutdinov, G.E. Hinton, “Deep boltzmann machines”, in *Proceedings of the international conference on artificial intelligence and statistics*, vol. 5, no. 2, pp. 448-455, Cambridge, MA: MIT Press.
- [4] S.E. Fahlman, G.E. Hinton, “Connectionist Architectures for Artificial Intelligence”. *Computers for Artificial Intelligence Processing*. WHA, B. W. & RAMAMOORTHY, C. V. John Wiley & Sons Inc., Nova York, 1990.
- [5] G.E. Hilton, T.J. Sejnowski, “Learning and Relearning in Boltzmann Machines”, MIT Press, Cambridge, Mass., 1, 282–317, 1985.