

Estudo e Aplicação de Redes Neurais na Implementação de Memórias Associativas

Enio R. Marins

Departamento de Informática e Ciência da Computação
Instituto de Matemática e Estatística
Universidade do Estado do Rio de Janeiro, Brasil
enioromano@openlink.com.br

André S. Langkjer

Departamento de Informática e Ciência da Computação
Instituto de Matemática e Estatística
Universidade do Estado do Rio de Janeiro, Brasil
alangkjer@lucent.com

Roseli S. Wedemann

Departamento de Informática e Ciência da Computação
Instituto de Matemática e Estatística
Universidade do Estado do Rio de Janeiro, Brasil
roseli@ime.uerj.br

Inês de Castro Dutra

Programa de Engenharia de Sistemas e Computação
COPPE/UFRJ, Brasil
inês@cos.ufrj.br

Abstract

Este texto descreve o modelo de Hopfield e ilustra o seu funcionamento através de algumas aplicações simples, que facilitam o entendimento do modelo. O enfoque das aplicações é ilustrar o uso da rede de Hopfield como memória associativa, que é um problema clássico de redes neurais. Além disso, o texto trata do aspecto distribuído do processamento nas redes de Hopfield.

1. Introdução

1.1 Descrição da rede binária de Hopfield¹

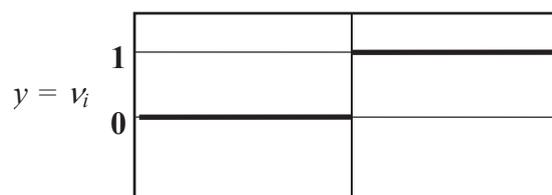
Existem dois modelos de redes neurais desenvolvidos por Hopfield: o analógico e o binário. Aqui trataremos apenas o modelo binário de Hopfield que, como veremos mais adiante, é utilizado como um modelo de memória associativa.

O modelo consiste de uma rede autômata representada por um grafo G do tipo parcialmente concorrente (PC) com n nós, ligados por $|E|$ conexões. A cada nó associa-se um estado v_i que depende dos estados dos outros nós conectados a ele. O estado de um nó n_i de uma rede neuronal no modelo binário de Hopfield está restrito ao conjunto $\{0,1\}$ e é dado pela expressão:

$$v_i := \text{step} \left(\sum_{j=1}^n w_{ij} v_j + e_i - \Theta_i \right) \quad (1)$$

$$\text{onde } \text{step}(y) = \begin{cases} 0, & \text{se } y \leq 0; \\ 1, & \text{se } y > 0 \end{cases} \quad \text{e } w_{ij} \text{ é o "peso"}$$

sináptico" da ligação que parte do neurônio (nó) n_j para o neurônio n_i , e_i é uma entrada externa para n_i e Θ_i é o potencial limite de n_i , como ilustrado na Figura 1.



$$\Theta_i \quad x = \sum_{j=1}^n w_{ij} v_j + e_i$$

Figura 1. O estado v_i de neurônio em uma rede binária de Hopfield é 1 se a entrada para o neurônio for estritamente maior que Θ_i , e é zero para entradas menores que Θ_i .

2. Simulação das redes de Hopfield

O comportamento coletivo dos neurônios interconectados entre si é em geral difícil de ser analisado. No entanto, uma função "energia" pode ser definida que evidencia propriedades muito interessantes do comportamento dinâmico da rede. A função a seguir define a energia em uma rede neuronal binária de Hopfield:

$$E = - \sum_{i=1}^n \sum_{j=1}^n w_{ij} v_i v_j - \sum_{i=1}^n e_i v_i + \sum_{i=1}^n \theta_i v_i. \quad (2)$$

A partir da função "energia", podemos analisar uma propriedade importante da rede neuronal binária de

¹ As sessões 1, 2, 3 e 4 foram baseadas em [BA93]

Hopfield que é dada pelo teorema a seguir, cuja prova se encontra em [BA93].

Teorema 1 – “Suponha que $w_{ij} = w_{ji}$ para todo $n_i, n_j \in N$, e seja $K \subseteq N$ tal que, se $n_i, n_j \in K$, então $w_{ij} = 0$. Se E_1 e E_2 são, respectivamente, os valores de E imediatamente antes e depois que os neurônios $n_i \in K$ são atualizados concorrentemente como em (1), então $E_2 - E_1 \leq 0$ ”.

Ou seja, se dois neurônios interconectados por um peso sináptico diferente de zero não atualizam seus estados concorrentemente, a energia da rede diminui conforme os estados dos neurônios são atualizados.

2.1. Algoritmo Sequencial

O comportamento dinâmico das redes neuronais binárias de Hopfield não depende de um parâmetro de tempo pertencente ao conjunto dos números reais, devido à função degrau (Eq. 1 e Fig. 1), que descreve a atualização dos estados dos neurônios. Seu comportamento depende então essencialmente da concorrência da atualização dos estados dos neurônios conforme enunciado no Teorema 1 da seção anterior, que define uma condição suficiente para que a atualização dos neurônios leve a uma sucessão não crescente de valores da função de energia.

Outro aspecto que determina o comportamento das redes binárias de Hopfield é a frequência relativa com a qual os neurônios têm seus estados atualizados. Só é possível garantir que mínimos locais da função energia (2) serão atingidos se a cada neurônio é dada a oportunidade de atualizar seu estado com alguma frequência (ver [BA93] pág. 122).

Estes dois aspectos do comportamento de redes binárias de Hopfield, a concorrência relativa e a frequência com a qual os estados dos neurônios são atualizados, caracterizam estas redes como redes autômatas do tipo PC. Esta rede é obtida escolhendo como arestas de G os pares $n_i, n_j \in N$ tais que $w_{ij} \neq 0$, de forma que em G o conjunto K mencionado no Teorema 1 forma um conjunto independente.

A função de atualização f neste caso é tal que $x_i(s)$ é dado pelo valor inicial de v_i para $s = 0$, e para $s > 0$ pelo valor de v_i obtido após o pulso mais recente no qual n_i estava no conjunto independente. Esta rede autômata pode ser simulada por um algoritmo sequencial que em cada iteração percorre todos os neurônios em uma ordem pré-fixada e atualiza seus estados de acordo com (1). Este algoritmo simples (Figura 2) é executado até que os estados atualizados de todos os neurônios sejam iguais aos seus estados anteriores, após uma iteração de atualização completa (i.e. os estados não mudam mais).

```
for  $i := 1$  to  $n$  do
```

```
   $v_i := v_i^0$ ;
```

```
repeat
```

```
for  $i := 1$  to  $n$  do
```

```
begin
```

```
   $v_i^- := v_i$ ;
```

```
   $v_i := \text{step} \left( \sum_{j=1}^n w_{ij} v_j + e_i - \Theta_i \right)$ 
```

```
end
```

```
until  $v_i = v_i^-$  for all  $n_i \in N$ .
```

Figura 2. Algoritmo Sequencial

O algoritmo sequencial (Fig. 2) que simula a rede neuronal apresentada na seção anterior é relativamente simples. Nele são utilizados três vetores que representarão os estados: inicial, anterior e atual, dos neurônios. Estes vetores são, respectivamente, v^0 , v^- e v . No primeiro passo a rede será inicializada, isto é, os estados dos neurônios terão seus valores iniciais iguais à v^0 , posteriormente, cada neurônio terá seu estado atualizado de acordo com a função de atualização da rede, até que a rede se torne estável, isto é, até que cada neurônio tenha seu estado atual constante ($v_i^- = v_i$).

3. Uso da rede como uma memória associativa

Uma memória associativa, também chamada de memória endereçável pelo conteúdo, é um dispositivo que permite armazenar itens que depois podem ser recuperados baseando-se apenas em informação incompleta sobre o próprio item a ser recuperado. Memórias associativas possuem numerosas aplicações e nestas aplicações os itens a serem armazenados são tipicamente particionados em campos, cada um com uma semântica especial. Um ou mais destes campos são as chaves para o item, e são apresentados para a memória quando o item precisa ser recuperado. A recuperação do item disponibiliza os outros campos para uso.

A rede neuronal de Hopfield pode ser usada como um modelo de memória associativa. A idéia básica é simples: como a evolução da rede leva a estados estáveis, cada item a ser armazenado pode ser associado à um destes estados. Quando inicializada em um estado “próximo” ao estado que representa o item desejado, a rede evolui para este estado estável e o item pode ser recuperado totalmente. Configurar a rede binária de Hopfield para funcionar como uma memória associativa envolve pelo menos três pontos importantes: estabelecer uma medida de “distância” entre os estados; uma técnica para a determinação dos parâmetros da rede (essencialmente os pesos das sinapses); e a determinação do número de itens que poderão ser armazenados na rede (em função do número de neurônios).

4. A Regra de aprendizado de Hebb na inicialização dos pesos das conexões e propriedades

Em redes neuronais, regras de aprendizado consistem nos meios para se configurar os pesos das conexões entre os neurônios. A regra utilizada para o armazenamento de um padrão de entrada no modelo binário de Hopfield é dada por:

$$w_{ij} = \begin{cases} \frac{1}{n} \mu_i \mu_j, & \text{se } i \neq j \\ 0, & \text{se } i = j \end{cases} \quad (3)$$

onde μ é um padrão de entrada para a rede.

Para o armazenamento de P padrões de entrada, pode-se utilizar a regra de *Hebb* dada a seguir:

$$w_{ij} = \begin{cases} \frac{1}{n} \sum_{p=1}^P \mu_i^p \mu_j^p, & \text{se } i \neq j; \\ 0, & \text{se } i = j \end{cases} \quad (4)$$

para todo n_i, n_j pertencente N .

A expressão (4) acima descrita é conhecida como a regra de Hebb para o aprendizado. A configuração dos pesos sinápticos pode ser vista como o processo através do qual a rede “aprende” quais padrões serão armazenados. Para um padrão μ^q para algum q tal que $1 \leq q \leq P$, a entrada para n_i é, pela expressão anterior:

$$\begin{aligned} \sum_{j=1}^n w_{ij} \mu_j^q &= \frac{1}{n} \sum_{\substack{j=1 \\ j \neq i}}^n \sum_{p=1}^P \mu_i^p \mu_j^p \mu_j^q \\ &= \frac{n-1}{n} \mu_i^q + \sum_{\substack{j=1 \\ j \neq i}}^n \sum_{\substack{p=1 \\ p \neq q}}^P \frac{\mu_i^p \mu_j^p \mu_j^q}{n}, \end{aligned} \quad (5)$$

para todo $n_i \in N$.

Agora considere o somatório duplo na expressão anterior, freqüentemente chamado de termo de

crosstalk. Se este somatório é menor que $(n-1)/n$ para todo $n_i \in N$, então a entrada para cada n_i tem o sinal de μ_i^q , de forma que μ^q é estável, já que a condição de estabilidade de μ^q é dada por (ver Eq. 1, onde os estados assumem os valores 1 e -1 e a função *step* foi substituída pela função *sign*):

$$\mu_i^q = \text{sign} \left(\sum_{j=1}^n w_{ij} \mu_j^q \right), \quad (6)$$

para todo $n_i \in N$ com os w_{ij} calculados conforme (4).

Se o termo de *crosstalk* é maior ou igual a $(n-1)/n$ para algum padrão μ^q em algum neurônio n_i , então μ^q é instável. Certamente, espera-se um número maior de padrões instáveis a medida que P aumenta, isto é, a medida que tentamos armazenar mais padrões na rede. Uma questão fundamental é a determinação da capacidade da rede, isto é, o número máximo de padrões que a rede pode armazenar.

Dito deste modo, o conceito de capacidade da rede é obviamente impreciso, já que ele depende intimamente da precisão desejada no processo de recuperação dos padrões. Várias análises têm sido feitas na literatura para se determinar a capacidade de uma rede binária de Hopfield. Para análises feitas utilizando padrões aleatoriamente gerados, com cada neurônio tendo igual probabilidade de assumir o estado -1 ou 1, resumimos as seguintes conclusões. Se erros em uma pequena fração de neurônios são tolerados, a capacidade da rede é proporcional a n , mas não maior do que $0,138n$ (neste caso, uma fração de 0,36% de neurônios com estados errados quando a rede estabiliza será tolerada). Se, por outro lado, é necessário que todos os neurônios sejam perfeitamente recuperados, então a capacidade da rede é proporcional a $n / \log(n)$ [HE91].

Quando uma rede binária de Hopfield é empregada como uma memória associativa, o grafo G que dá a estrutura da rede autômata do tipo PC correspondente depende intimamente dos padrões a serem armazenados. Suponha, por exemplo, que calculamos os pesos sinápticos a partir da equação 4. Se os padrões são gerados aleatoriamente, como discutimos anteriormente, então a probabilidade de que um peso sináptico seja zero, se P é par (nenhum peso sináptico pode ser zero se P é ímpar), é de [BA93]:

$$\begin{cases} \left(\frac{P}{P/2} \right) \left(\frac{1}{2} \right)^P, & \text{se } P \text{ é par} \\ 0, & \text{se } P \text{ é ímpar.} \end{cases} \quad (7)$$

Para $P = 138$, que é o maior número de padrões que podem ser armazenados em uma rede com 1000 neurônios, de acordo com um dos critérios vistos anteriormente, temos uma probabilidade aproximada de

0,07 (7%) de um peso sináptico ser igual a zero. Conforme aumentamos o número de padrões armazenados esta probabilidade tende a diminuir ainda mais, tornando G um grafo muito denso, fazendo assim com que a simulação paralela e distribuída da rede autômata do tipo PC tenha muito pouca concorrência. Os pesos sinápticos gerados por aplicações reais, por outro lado, podem se comportar de modo diferente, e podem então permitir uma concorrência razoável durante a simulação paralela e distribuída.

5. Implementações

Para ilustrar o funcionamento da rede de Hopfield foram desenvolvidos três programas. O primeiro programa foi implementado para a verificação da capacidade da rede de “memorizar” padrões de estados de neurônios. O segundo e o terceiro programas são adaptações do primeiro de modo a tornar os exemplos mais visuais. No segundo programa, que é uma implementação simples de uma memória associativa, cada grupo de seis neurônios representam um caracter e, deste modo, a rede pode armazenar e recuperar palavras ou frases. O terceiro programa trabalha com imagens formadas por caracteres que são mapeados para estados de neurônios, fazendo com que a rede armazene e recupere (reconstrua) estas imagens.

5.1. Descrição das Aplicações

O primeiro programa é utilizado para verificação de dados estatísticos da rede. Ele permite criar uma rede neuronal com um número variável de neurônios. É possível apresentar padrões para que a rede os armazene; estes padrões podem ser gerados tanto manualmente (informados pelo usuário) ou pelo próprio programa (geração aleatória automática). Uma vez configurados os pesos das conexões (fase de aprendizado), é possível apresentar a esta rede padrões a serem recuperados por ela. Estes padrões contém alguns estados incorretos em relação a um dos padrões armazenados na rede, de forma que enquanto os estados dos neurônios são atualizados pela rede, os estados incorretos vão sendo corrigidos até que um padrão muito próximo ou igual a um dos armazenados é retornado, ou seja, a rede converge para um estado armazenado. Os dados estatísticos do programa mostram quantos testes foram realizados, o número de acertos da rede e a taxa de acertos (porcentagem de padrões testados que foram corretamente recuperados). Relatamos estes resultados na sessão 6.

O segundo programa, de forma bastante aproximada, visa a funcionar como um “corretor ortográfico”. A idéia é a mesma: consiste em apresentar frases a serem armazenadas pela rede e frases de teste a serem ‘corrigidas’ pela rede. Assim, é possível armazenar a palavra ‘projeto’ e quando a palavra ‘projet’ é apresentada como teste, a rede converge para um estado final estável e ‘projeto’ é dado como resposta.

O terceiro programa, mais uma vez, apresenta a mesma proposta, mas utilizando um aspecto mais visual: ele armazena imagens simples. Assim, conforme mostrado na Figura 3, se a imagem 1 é armazenada na fase de treinamento da rede. Quando a imagem 2 é apresentada posteriormente, a rede retorna a imagem 3. Desta forma, a rede pode ser utilizada como corretora de imagens.

***	***	***
**	*	**
***	***	***
Imagem 1 (entrada)	Imagem 2 (teste)	Imagem 3 (saída)

Figura 3. Teste de funcionamento do terceiro programa. É dado como entrada a Imagem 1,como teste a Imagem 2, tendo como resposta a Imagem 3.

6. Resultados

Em primeiro lugar, foi testada a capacidade de armazenamento da rede usando a implementação descrita em 5.1. Os testes foram feitos para redes com 10, 20, ... 100 neurônios. Para cada uma destas redes foram armazenados um número p de padrões tal que $p = \lfloor 0,138n \rfloor$ conforme descrito na sessão 4 [HE91]. Utilizando o primeiro programa descrito na sessão 5.1, verificamos que quando os próprios padrões armazenados são fornecidos à rede, sua capacidade obedece à regra de proporcionalidade mencionada neste texto conforme mostra o gráfico da Figura 4, cuja reta que representa os resultados das experiências tem um coeficiente aproximadamente igual a 0,138. Ou seja, a rede sempre converge para um padrão correto quando apresentada um dos padrões armazenados.

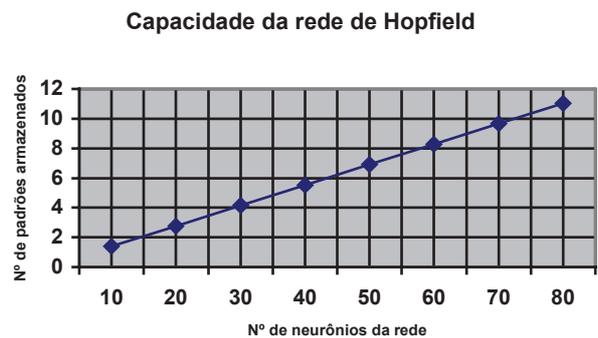


Figura 4. Gráfico da capacidade da rede neuronal de Hopfield.

Para padrões de teste diferentes dos padrões armazenados, verificou-se que nem sempre a rede converge para um padrão armazenado. Na verdade,

vimos que a rede erra para um número considerável de padrões de teste, principalmente quando temos uma rede com um grande número de neurônios. Em relação a este fato, devemos ter em mente duas considerações importantes.

Em primeiro lugar devemos lembrar que uma rede neuronal artificial é um sistema baseado no aprendizado assim como as redes neuronais biológicas, portanto, não se pode esperar que estas redes forneçam sempre resultados corretos. A precisão dos resultados esperados estará diretamente ligada à regra de aprendizado e ao modelo adotado. Desse modo, modelos de redes diferentes com diferentes regras de aprendizado fornecerão resultados próximos, porém não iguais.

A outra consideração advém de outras experiências realizadas com a rede. À medida que a rede armazena novos padrões, estes tendem a “atrapalhar” outros padrões que já estão armazenados, i.e., a rede tende a “esquecer” padrões já armazenados a medida que ela “aprende” novos padrões. Entretanto, se forcarmos que a rede “aprenda” novamente um padrão que ela “esqueceu”, perceberemos que as chances de ela retornar este padrão aumenta. Trata-se, portanto, de mais uma característica semelhante ao cérebro humano. Por exemplo, quando temos que estudar para algum exame, temos que ler a matéria duas, três ou mais vezes para que possamos memorizá-la.

7. Conclusões e Trabalhos Futuros

Neste trabalho, tentamos mostrar o funcionamento de uma memória associativa baseada no modelo binário de Hopfield. Vimos que o papel da rede neuronal de Hopfield é fazer evoluir no tempo os estados dos neurônios até que um valor mínimo da função de energia associada aos estados da rede seja atingido. Através desta propriedade, a rede pode ser adaptada para a solução de alguns problemas computacionais, além da possibilidade de se modelar de forma exageradamente aproximada o funcionamento da memória humana.

Quanto ao campo da modelagem cerebral, percebemos que a rede neuronal binária de Hopfield tem algumas características semelhantes à memória humana. A memória humana pode ser também considerada uma memória endereçada pelo conteúdo, já que, por exemplo, muitas vezes podemos lembrar de acontecimentos inteiros tendo apenas uma pequena parte dos fatos. Além disso, verificamos outras características importantes da rede de Hopfield que é a tolerância a falhas e o processo de memorização.

Vimos que a Regra de Hebb, apesar de sua simplicidade, se mostrou ineficiente para padrões de testes não aleatórios. Vale lembrar que existem outros algoritmos de aprendizado até mesmo mais precisos que o descrito neste trabalho, por exemplo os algoritmos genéticos.

Como desdobramento deste trabalho podemos citar a implementação do algoritmo de escalonamento de

neurônios a serem atualizados para a simulação paralela e distribuída da Rede Neuronal Binária de Hopfield e a pesquisa e implementação de novos algoritmos de aprendizado.

Referências Bibliográficas

- [BA93] Barbosa, V. C., *Massively Parallel Models of Computation*, Ellis Horwood Limited, West Sussex, Inglaterra, 1993.
- [HE91] Hertz, J., A. Krogh e R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Company, 1991.
- [HO82] Hopfield, J. J., “Neural Networks and Physical Systems with Emergent Collective Computational Abilities”, In *Proceedings of the National Academy of Sciences*, USA 79, 1982 pp.20554-2558.
- [JA96] Jain, A. K. e J. Mao, “Artificial Neural Networks: A tutorial”, *Computer*, Vol. 29, Nº 3, Mar 1996, pp. 31-44.
- [MC43] McCulloch, W. S. e W. Pitts, “A Logical Calculus of Ideas Immanent in Nervous Activity”, *Bull. Mathematical Biophysics*. Vol. 5 – 1943.
- [PA96] Pal, S. K. e P. K. Srimani, “Neurocomputing. Motivation, models, and hybridization”, *Computer*, Vol. 29, Nº 3, Mar 1993, pp. 24-28.
- [RU86] Rumelhart, D. E. e J. L. McClelland, *Paralel Distributed Processing: Exploration in the Microstructure of Cognition*, MIT, Press, Cambridge, Mass., 1986.
- [SE96] Serbedzija, N. B., “Simulating Artificial Neural Networks on Parallel Architectures”, *Computer*, Vol. 29, Nº 3, Mar 1993, pp. 56-63.
- [SH93] Shang, Y. e B. W. Wah, “Global Optimization for Neural Network Training”, *Computer*, Vol. 29, Nº 3, Mar 1993, pp. 45-54.
- [RO00] Romano, E. e A. Langkjer, *Estudo de Redes Neuronais na Implementação de Memórias Associativas*, Monografia de Final de Curso de Graduação apresentado ao Departamento de Informática e Ciência da Computação do Instituto de Matemática e Estatística da Universidade do Estado do Rio de Janeiro, 2000.