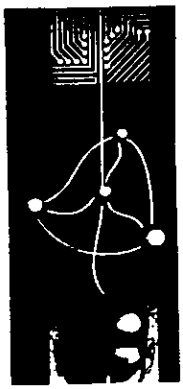

1° Congresso Brasileiro de Redes Neurais

Escola Federal de Engenharia de Itajubá
Itajubá. 24 a 27 de outubro de 1994

Um Método de Aprendizado para Redes Neuronais Analógicas de Hopfield com aplicação à Compactação de Imagens

Jane Tavares Alvarez
ILTC,Instituto de Lógica,Filosofia eTeoria da Ciência,R.Almirante Teffé,637,
CEP 24030 080,RJ,Brasil
Luis Alfredo Vidal de Carvalho
COPPE,Universidade Federal do Rio de Janeiro,Caixa Postal 68511,RJ,Brasil

Abstract. Este trabalho apresenta e avalia três novos algoritmos de aprendizado para Redes Neuronais Analógicas de Hopfield [1][2], utilizando padrões gerados aleatoriamente. Os testes mostraram que, dada uma rede neuronal analógica de Hopfield com *n* neurônios, o número máximo de padrões estáveis pode superar os 15%*n* [1]. Em seguida, como uma aplicação do algoritmo de aprendizado de melhor desempenho, treinaram-se grupos de imagens de 400 pixeis nas cores preto, cinza e branco, para em seguida serem compactadas/descompactadas, via rede de Hopfied. Os testes nesta fase também produziram bons resultados: primeiro, todas as imagens foram 100% aprendidas; segundo, as distorções produzidas na etapa de compactação/descompactação, mesmo utilizando uma alta taxa de compactação, não inviabilizaram a compreensão das imagens.

Introdução. Em geral, o aprendizado em redes neuronais consiste apenas na obtenção de um padrão de conexão W que permita associar padrões eficientemente. Apresentaremos neste trabalho, três algoritmos de aprendizado para redes neuronais analógicas de Hopfield [4], dois dos quais não se restringem tão-somente a obter uma matriz de pesos satisfatória. Assim, buscando maior liberdade, que em outros tipos de rede pode significar apenas um aumento do número de conexões. trabalharemos também variando o potencial limiar θ e o ganho γ (dos estados de ativação dos neurônios da rede) a fim de que o aprendizdo seja mais eficiente.

Estes três algoritmos trabalham, respectivamente, variando W [3], W e θ, e por fim, W, θ e γ. No entanto, apenas este último algoritmo será abordado de forma mais completa, pois além de ser obtido a partir dos anteriores, ele demonstrou, através dos testes realizados, uma *performance* superior.

Intuitivamente, o que se propõe é encontrar uma matriz de pesos W, um potencial limiar θ e um ganho γ tal que dados *m* padrões de atividade $v_1, v_2, \ldots, v_m$ se consiga ajustar a função de energia E, definida pela equação [1], a todos estes pontos, de forma que eles constituam mínimos locais.

Consideremos uma rede neuronal analógica de Hopfield com *n* neurônios $N_1, N_2, \ldots, N_n$, *m* padrões de atividade, capacitância C, resistência R e impulso total de entrada u_i, relativo ao neurônio N_i, dado pela equação:

$$C\frac{du_i(t)}{dt} = \sum_j w_{i,j}\, a_j(t) - \frac{u_i(t)}{R},$$

sendo que $v_i \in (0,1)$ representa o estado de ativação do neurônio N_i e $W = [w_{i,j}]$

onde $w_{i,j} = w_{j,i}$, os pesos sinápticos, significam que o padrão de conexão W é uma matriz simétrica.

Assim, para uma matriz W, limiar θ, ganho γ e padrão de atividade $v \in (0,1)^n$ a rede possui energia

$$E(W,\theta,\gamma,v) = \frac{-1}{2}v^T W v + \frac{1}{R}\sum_{i=1}^{n}\int_{y=0}^{y=v_i} s^{-1}(y)dy$$

$$\text{onde } s^{-1}(y) = \frac{-1}{\gamma}\ln\left(\frac{1}{y}-1\right)+\theta. \qquad [1]$$

Diz-se que um padrão v é *mínimo local* de $E(W,\theta,\gamma,v)$, se e somente se,

1. $\nabla_v E(W,\theta,\gamma,v) = 0$
2. se a matriz hessiana[1] $H(W,\theta,\gamma,v)$ de $E(W,\theta,\gamma,v)$ com respeito a v é do tipo *semi-positiva*, isto é, se $x^T H(W,\theta,\gamma,v)x \geq 0$, $\forall\, x \in \Re^n$.

Em termos práticos, é difícil satisfazer a condição 2 para qualquer $x \in \Re^n$, principalmente pelo fato de $\Re^n$ ser incontável. Portanto, nos deteremos apenas a primeira condição, apesar de não ser possível assegurar a minimalidade da função de energia, isto é, que um dado padrão de atividade v é de fato, mínimo da equação:

$$E = \frac{-1}{2}\sum_{i}^{n}\sum_{i}^{n} w_{i,j}v_i v_j + \sum_{i}^{n}\frac{1}{R_i}\int_{0}^{v_i(t)} s^{-1}(v_i)dv_i$$

$$\text{onde } s^{-1}(v_i) = \frac{-1}{\gamma}\ln\left(\frac{1}{v_i}-1\right)+\theta. \qquad [2]$$

O problema então consiste em, dados m padrões de atividade $v^1,\ldots,v^m \in (0,1)^n$, determinar uma matriz de pesos W simétrica, um novo potencial limiar θ e um novo ganho γ, de forma que $v^1,\ldots,v^m$ representem pontos de mínimo, ou pontos próximos aos pontos de mínimo, da superfície definida pela função E de energia (equação 2).

Algumas Considerações Importantes. Seja $S^{-1}(v)$ um vetor cuja i-ésima componente é dada por v_i, onde $v_i \in (0,1)$. Levando em conta apenas a primeira condição da definição dada anteriormente, pode-se ter que se v é um padrão estável de $E(W,\theta,\gamma,v)$ então v é um padrão estável sobre uma matriz de conexões W, um potencial limiar θ e um ganho γ. Portanto, se v é um padrão estável de $E(W,\theta,\gamma,v)$ então $\nabla_v E(W,\theta,\gamma,v) = 0$, ou seja, $Wv - \frac{S^{-1}(v)}{R} = 0$. Mas isso ocorre, se e somente se,

$$\left(Wv - \frac{S^{-1}(v)}{R}\right)^T\left(Wv - \frac{S^{-1}(v)}{R}\right) = 0.$$

Logo, dados m padrões de atividade, se eles são todos simultaneamente estáveis então

$$\varepsilon = \sum_{p=1}^{m}\left(Wv^p - \frac{S^{-1}(v^p)}{R}\right)^T\left(Wv^p - \frac{S^{-1}(v^p)}{R}\right)s$$

endo $\varepsilon = 0$. Para facilitar, seja $\delta^p = Wv^p - \frac{S^{-1}(v^p)}{R}$. Então, da equação anterior podemos ter que

$$\varepsilon = \sum_{p=1}^{m}(\delta^p)^T\delta^p \quad \text{e} \quad \varepsilon = 0$$

Nosso problema agora se reduz a obter um ε mínimo a fim de que $v^1, v^2,\ldots, v^m$ estejam enfim, bem próximos de pontos estáveis. Dessa forma, é preciso minimizar $\varepsilon = f(W,\theta,\gamma,v^1,v^2,\ldots,v^m)$ com respeito à W, θ e γ, ou seja, obter $\min_W \varepsilon$, $\min_\theta \varepsilon$ e $\min_\gamma \varepsilon$.

Minimizar ε com respeito à W significa calcular $\nabla_W \varepsilon$, isto é, $\nabla_W f$. Dessa forma,

$$\nabla_W f = \sum_{p=1,m}\delta^p(v^p)^T + v^p(\delta^p)^T$$

Analogamente, $\min_\theta \varepsilon$ significa obter o valor de $\nabla_\theta \varepsilon$. Assim,

$$\nabla_\theta f = \sum_{p=1}^{m}\frac{-2}{R}\delta^p$$

E por fim, para minimizar ε com respeito a γ, basta calcular

$$\nabla_\gamma f = \sum_{p=1}^{m}\frac{-2}{R\gamma^2}\delta^p \ln\left(\frac{1-v^p}{v^p}\right), \qquad \gamma \in \Re_+^n$$

[1] $\nabla_v E$ significa o gradiente de E com respeito a v. A matriz hessiana é a matriz das derivadas segundas.

Os Algoritmos. À medida que se aumentavam os graus de liberdade da rede neuronal, um novo algoritmo surgia. O primeiro algoritmo criado foi obtido variando-se o padrão de conexão W. Utilizando-se o potencial limiar, deu-se origem ao aprendizado em W e θ. Finalmente, variando W, θ e γ gerou-se o algoritmo de aprendizado em W, θ e γ.

Algoritmo de Aprendizado em W, θ e γ

$W \leftarrow W_{inicial}$
$\theta \leftarrow \theta_{inicial}$
$\gamma \leftarrow \gamma_{inicial}$
enquanto f puder ser localmente minimizada **faça**
$W \leftarrow W - \eta_W \nabla_W f$, para algum $\eta_W > 0$
$\theta \leftarrow \theta - \eta_\theta \nabla_\theta f$, para algum $\eta_\theta > 0$
$\gamma \leftarrow \gamma - \eta_\gamma \nabla_\gamma f$, para algum $\eta_\gamma > 0$
fim_do_enquanto

Uma vez que buscamos os pontos de mínimo da função f, iremos considerar
$-\nabla f$. Assim, W, θ e γ variam proporcional e respectivamente, às taxas de aprendizado η_W, η_θ e η_γ, conforme $-\nabla f$.

Os critérios de parada. A condição de parada dos algoritmos consiste em ter o $\nabla_* f$ nulo para * = W, θ e γ, conforme o algoritmo considerado. Mas, $\nabla_* f = 0$ implica em $\|\nabla_* f\| = 0$. Como em termos computacionais não é fácil garantir que $\nabla_* f = 0$, utiliza-se um ERRO* considerado aceitável em relação aos mínimos procurados, e que possibilitou um tempo razoável de processamento para os algoritmos. Dessa forma então, o terceiro algoritmo pára quando $\frac{\|\nabla_W f\|}{m} < ERRO_W$, $\frac{\|\nabla_\theta f\|}{m} < ERRO_\theta$ e $\frac{\|\nabla_\gamma f\|}{m} < ERRO_\gamma$, isto é, quando $\|\nabla_* f\|$ atinge valores bem próximos de zero.

Resultados Experimentais. Segundo Hopfield [1], dada uma rede neuronal com n neurônios, o número máximo p de padrões estáveis que se pode obter é $p \approx 0.15n$. Mostraremos aqui, no entanto que, com os algoritmos propostos é possível, fazer com que este valor seja superado de maneira significativa.

Cada algoritmo de aprendizado foi testado utilizando-se 50 neurônios e m padrões de atividade gerados aleatoriamente. Para cada algoritmo foi feito um levantamento em função do número de padrões simultaneamente aprendidos. Assim, dados m padrões calcula-se uma fração ou porcentagem conforme o número m^* de padrões que tenham sido simultaneamente aprendidos.

Concluído o aprendizado, ocorre a fase de simulação em que a rede evolui dinamicamente até que o valor da energia computacional, definida pela equação [2] atinja um mínimo satisfatório, onde enfim estabiliza. No entanto, a fim de que a simulação tivesse um caráter ainda mais preciso, resolvemos considerar a norma quadrática média da diferença entre dois padrões de atividade em tempos consecutivos, ou seja,

$$\frac{\|V_{t+\Delta t} - V_t\|^2}{n}$$

Mesmo assim, poderia haver o risco de que a simulação fosse processada em um tempo muito aquém do necessário. Dessa forma, considerou-se que esta etapa deveria levar um tempo de simulação mínimo igual a 0.1, o que equivale a 100 iterações.

Para o primeiro algoritmo, foram considerados nulos a matriz W inicial e o potencial limiar θ, enquanto que R = C = γ = 1.0. Segundo os resultados dos testes que foram realizados para até m = 15 padrões, temos que, a partir de m = 3 começa a haver uma interferência gradual entre os padrões a serem ensinados. No entanto, o desempenho do algoritmo supera os 15% dos 50 neurônios utilizados, pois ao serem fornecidos à rede m = 10 padrões é possível ensinar todos os 10 em 50% dos casos, além de ser possível, no pior caso, ensinar m^* = 8 em 20% dos casos. Mesmo ao serem dados m = 15 padrões, é possível

ensinar, simultaneamente, $m^* = 13$ padrões em 50% dos casos e $m^* = 11$ padrões em 20% dos casos, apesar de não serem simultaneamente aprendidos todos os 15 padrões.

Os valores da taxa de aprendizado η_W são completamente empíricos, variando à medida que se aumenta a quantidade m de padrões a serem treinados e se deseja manter um bom nível de treinamento.

No segundo algoritmo, tanto o padrão de conexão W inicial quanto o potencial limiar inicial foram gerados aleatoriamente. No entanto, γ, R e C assumiram o mesmo valor constante igual a 1.0, como no algoritmo anterior.

Os testes foram realizados para até m = 25 padrões e apresentaram um desempenho superior ao do algoritmo 1, o que se atribui essencialmente, a um maior grau de liberdade no aprendizado. Por exemplo, dados m = 9 padrões, foram simultaneamente aprendidos todos os 9 em 60% dos casos, enquanto que pelo primeiro algoritmo isso ocorreu apenas em 40% dos casos. Já ao se considerar m = 15 padrões é possível aprender todos os 15 em 30% dos casos e $m^* = 9$ padrões em 10% dos casos. Considerando m = 20 ou m = 25, foi possível ensinar, respectivamente, $m^* = 20$ e $m^* = 25$ em 20% dos casos testados.

Para o último dos algoritmos a ser testado, utilizou-se $W_{inicial}$ e $\theta_{inicial}$ aleatórios, γ_i inicial igual a 0.5, para $1 \leq i \leq 50$, sendo preservados os valores de R = C = 1.0. A variação de γ permitiu um aprendizado ainda melhor em relação ao obtido pelos demais algoritmos. Por exemplo, dados m = 15 padrões foi possível ensinar todos os 15 padrões em 50% dos casos contra 0% e 30% obtidos respectivamente, pelos algoritmos 1 e 2. Ainda, ao serem utilizados m = 20 padrões foi possível ensiná-los todos em 30% dos casos, sendo também possível (no pior caso) ensinar $m^* = 15$ padrões em 20% dos testes realizados.

Pela análise dos testes realizados, concluimos empiricamente que, o algoritmo de aprendizado em W, θ e γ apresenta um desempenho superior em relação aos demais. Dessa forma, iniciaram-se os testes de aplicação, que consistiram em utilizar grupos de uma até nove imagens de 400 pixeis, ensiná-las, depois compactá-las e finalmente, restaurá-las. Entretanto, é importante notar que, cada pixel de uma imagem qualquer I representa na verdade, o estado de ativação de um neurônio que a constitui.

No que diz respeito ao aprendizado destas imagens, que serão melhor detalhadas adiante, temos que não houve o mínimo possível de degradação, isto é, o aprendizado foi 100%, inclusive nos casos em que se utilizou um número maior de imagens para serem simultaneamente aprendidas.

O Método de Compactação Utilizado. O método de compactação aqui utilizado é bastante intuitivo e está intimamente ligado ao uso da rede neuronal, apesar de existirem outros métodos [5] que não necessitam da rede para realizar tal tarefa.

Suponhamos uma imagem I constituída por n pixeis ou neurônios. Considerando uma taxa de armazenamento de neurônios t_a, sabemos que dos n pixeis ou neurônios que formam a imagem, apenas $t_a * n$ neurônios serão armazenados. Este método propõe, então, que sejam recrutadas as unidades que adquiriram maior conhecimento a respeito de I.

A fase de descompactação de I consiste em fornecer à rede a imagem I_c compactada, que constitui parte da informação total. Como a propriedade fundamental de uma rede neuronal de Hopfield é a memória de acesso pelo conteúdo (CAM), basta fornecermos à rede a imagem compactada I_c, para que ela evolua até atingir um padrão estável, ou seja, uma imagem I_f descompactada.

Cabe agora compararmos a imagem final I_f com a imagem I_o previamente concebida no início de todo o processo. Tal comparação é feita analisando-se unidade a unidade, isto é, o pixel ou neurônio N_i^o de I_o e o pixel ou neurônio N_i^f de I_f, e assim

sucessivamente. Quando as unidades N_i^o e N_i^f, para $1 \leq i \leq n$, são distintas, computamos o valor do número de neurônios errados (NNE), que é uma forma adequada e prática se considerarmos o aspecto visual da imagem.

Os Testes de Compactação. As imagens originais I_o foram criadas com 400 pixeis na cores preto, cinza e branco. Visto que cada pixel constitui na verdade um neurônio N_i com um estado de ativação $a_i(t)$ pertencente ao intervalo (0,1), foi preciso relacionar cada cor da imagem I_o com um valor de ativação neste intervalo. Assim, o primeiro passo foi dividir o intervalo (0,1) em três sub-intervalos, atribuindo a cada cor um valor no meio do subintervalo.

Na representação das imagens aprendida (gerada ao final do treinamento e da simulação) e descompactada, o processo foi inverso, uma vez que os estados de ativação deviam ser mapeados em uma das cores usadas. Neste caso então, a cor foi fornecida conforme o subintervalo do intervalo (0,1) em que se encontrava os estados de ativação, já que cada sub-intervalo podia ser relacionado com uma das cores usadas.

A compactação foi realizada para grupos de até nove imagens previamente aprendidas pela rede neuronal, utilizando-se para as taxas de compactação valores que variaram entre 0.9 e 0.1, conforme a necessidade em armazenar, respectivamente, um número menor ou maior de neurônios.

Foi possivel observar que, à medida que aumentamos o número de imagens a serem compactadas, a taxa média de compactação em que o número de neurônios (pixeis) errados atinge zero, diminui progressivamente. Isso ocorre porque, durante o processo de aprendizado a rede neuronal obteve conhecimento acerca de todas as imagens simultaneamente, não guardando necessariamente, as características específicas de cada uma delas. Além disso, ao se dar o processo de compactação são guardados os neurônios mais fortes que compõem cada imagem, podendo em muitos casos, os estados de ativação relativos aos neurônios mais marcantes de uma imagem estarem bem próximos das ativações dos neurônios mais significativos de outras imagens. Portanto, quanto maior o número de imagens no processo de compactação, maior a quantidade necessária de pixeis a serem armazenados, a fim de que haja o mínimo possivel de superposição de imagens e consequentemente, um bom nível na qualidade das imagens restauradas.

Considerando a relação entre a taxa de compactação e a média do número de neurônios errados, dado um número k de imagens que foram simultaneamente apendidas, onde $2 \leq k \leq 9$, temos que, à medida que diminuimos a taxa de compactação, a média do número de neurônios ou pixeis errados (MNNE) também diminui, até atingir zero. No caso de $k = 1$, a MNNE foi nula logo ao considerarmos 90% de compactação. Ao se aumentar o número k de imagens a serem compactadas e computando a média do número de neurônios errados, temos que esta média tende a baixar mais lentamente à medida que incrementamos o número de imagens. Portanto, são necessárias taxas cada vez menores, conforme se queira utilizar um número maior de imagens, e assim obter uma MNNE bem baixa.

Conclusão. Os testes realizados com os três algoritmos de aprendizado propostos para redes neuronais analógicas de Hopfield, mostraram de forma empírica, que mesmo o mais simples desses algoritmos pode superar os resultados existentes até então. Dessa forma, o terceiro algoritmo, que obteve melhor desempenho entre os algoritmos propostos pôde, dados uma rede com 50 neurônios e 15 padrões de atividade gerados aleatoriamente, ensiná-los todos em 50% dos casos, sendo que no pior caso, foi possivel ensinar 11 padrões dos 15 fornecidos.

A etapa de aplicação, embora tenha utilizado um número baixo de cores,

também produziu bons resultados, principalmente se considerarmos o aspecto visual das imagens.

Apesar de termos tratado com imagens pequenas de apenas 400 pixeis, por limitação de tempo e de máquina, podemos encarar estas imagens, em termos de trabalhos futuros, como sub-imagens formando imagens maiores que as utilizadas neste trabalho.

Bibliografia.

[1] Hopfield, J.J.,"Neural Networks and Physucal Systems with Emergent Collective Computational Abilities", Proc. National Academy of Sciences USA 79, 2554-2558., 1982.

[2] Hopfield, J.J., "Neurons with Graded Response have Collective Computational Properties like those of Two-state Neurons", Proc. National Academy of Sciences USA, 81, 3088-3092, 1984.

[3] Barbosa, V. C., Carvalho, L.A.V., "Learning in Analog Hopfield Networks", International Joint Conference on Neural Networks - IJCNN, 1991.

[4] Alvarez, Jane Tavares, "Um Método de Aprendizado para Redes Neuronais Analógicas de Hopfield com Aplicação à Compactação de Imagens", Tese de Mestrado, Rio de Janeiro, COPPE/UFRJ, 1992.

[5] Soares, Luis Fernando Gomes, et al., "Fundfamentos de Sistemas Multimídia", Porto Alegre: Instituto de Informática da UFRGS, VIII Escola de Computação, 1992.