Proceedings of the IV Brazilian Conference on Neural Networks - IV Congresso Brasileiro de Redes Neurais
pp. 198-203, July 20-22, 1999 - ITA, São José dos Campos - SP - Brazil

Algoritmo Adaptativo com Reduzida Complexidade Computacional para Extração dos Principais Componentes de Análise

Adriana Rosas de Souza[1], Laerte Barbalho Jr.[1],
Adrião Duarte Dória Neto [1], Marco Antônio G. de Carvalho[2]

[1]Departamento de Engenharia Elétrica - Universidade Federal do Rio Grande do Norte
Campus Universitário, Lagoa Nova, CEP: 59072-970, Natal/RN – Brasil
[2]Departamento de Informática - UnP
adriana@telpa.com.br, adriao@leca.ufrn.br, laerte@leca.ufrn.br, magic@dca.fee.unicamp.br

Abstract

This work presents the development and application of an efficient algorithm of adaptative training for Principal Components Analysis (PCA). Such algorithm goal to provide a reduction of the computational complexity associated to a more appropriate rate of convergence when compared to the algorithms GHA (Generalized Hebbian Algorithm) and ALA (Adaptive Learning Algorithm). A comparative analysis of performace between this algorithm and the algorithms GHA and ALA is developed.

1. Introdução

Dentre os algoritmos [1]-[6] para extração das Principais Componentes de Análise usando Redes Neurais Artificiais, tem-se o trabalho original de Sanger [1] que combina a ortonormalização de Gram-Schmidt e o algoritmo unitário de Oja [2], resultando no algoritmo GHA. Este algoritmo foi inicialmente aplicado à compressão de dados/imagens , onde mostrou a capacidade de se obter elevadas taxas de compressão.

Entretanto, o problema de como selecionar automaticamente e adaptativamente a taxa de treinamento não foi considerada e o GHA poderá não convergir ou convergir muito lentamente se o valor da taxa de treinamento não for apropriadamente escolhida. Tal limitação de tempo de convergência torna impraticável a utilização deste algoritmo na sua forma básica em certas aplicações, como no caso de compressão de imagens, pois o volume dos dados a serem processados é bastante elevado.

Liang-Hwa Chen e Shyang Chang [3] desenvolveram um algoritmo (ALA) para selecionar adaptativamente a taxa de treinamento a fim de convergir para os PCA's mais rapidamente que o GHA tradicional. Esta nova implementação do GHA tornou a utilização de tal algoritmo mais viável em certas aplicações, porém acarretou em um crescimento na complexidade computacional do algoritmo e, consequentemente, em um tempo de processamento elevado.

Assim sendo, foi desenvolvido e será aqui apresentado o denominado algoritmo GHALO, que tem por meta a aceleração da convergência, otimização da forma como são encontrados os PCA's e redução da complexidade computacional.

2. Desenvolvimento teórico

Sendo o algoritmo GHALO uma derivação do algoritmo ALA, será apresentado um sumário deste algoritmo para, em seguida, serem enfocadas as contribuições propostas neste trabalho.

2.1. Algoritmo ALA

Para um vetor de entrada $\mathbf{x}$ de média zero e dimensão n com distribuição de probabilidade $p(\mathbf{x})$, o algoritmo ALA extrai os m principais componentes da seguinte forma:

i. Primeiramente, calculam-se os vetores pesos $\mathbf{w}_i(0) \in \Re^n$:

$$\left\|\mathbf{w}_i(0)\right\|^2 << 1/2 \qquad (1)$$

e adota-se um número pequeno positivo para os autovalores $\hat{\lambda}_i(0) = \delta$, para $i = 1, 2, ..., m$.

ii. Um novo padrão $\mathbf{x}(n)$ é apresentado à entrada da rede neural na iteração n, $n \geq 1$.

iii. As saídas $\mathbf{y}_i$'s são calculadas da seguinte forma:

$$\mathbf{y}_i(n) = \mathbf{w}_i^T(n)\mathbf{x}(n), \quad i = 1, 2, ..., m \qquad (2)$$

iv. Os autovalores λ_i's são estimados da seguinte forma:

$$\hat{\lambda}_i(n) = \hat{\lambda}_i(n-1) + \gamma(n)\left[\left(\mathbf{w}_i^T(n)\mathbf{x}_i(n)/\left\|\mathbf{w}_i(n)\right\|\right)^2 - \hat{\lambda}_i(n-1)\right] \qquad (3)$$

onde:

$$\mathbf{x}_i(n) \equiv \mathbf{x}(n) - \sum_{j=1}^{i-1} \mathbf{y}_j(n)\mathbf{w}_j(n).$$

O valor de $\gamma(n)$ deve ser menor que 1 e tende a 0

quando *n* se aproxima do infinito.

v. Modificar os pesos $\mathbf{w}_i$'s da seguinte forma:

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \\ + \eta_i(n)\mathbf{y}_i(n)\left[\mathbf{x}(n) - \sum_{j=1}^{i} \mathbf{y}_j(n)\mathbf{w}_j(n)\right] \quad (4)$$

onde

$$\eta_i(n) = \beta_i(n) / \hat{\lambda}_i(n). \quad (5)$$

O valor de $\beta_i(n)$ deve ser:

$$\beta_i(n) < 2(\sqrt{2}-1) \quad \text{e} \quad \lim_{n\to\infty} \beta_i(n) = 0$$

vi. Verificar o tamanho dos w_i's da seguinte forma:

(6)

$$\mathrm{w}_i(n+1) = \begin{cases} \sqrt{1/2}\left(\mathrm{w}_i(n+1)/\left\|\mathrm{w}_i(n+1)\right\|\right), \text{ se} \\ \left\|\mathrm{w}_i(n+1)\right\|^2 > \dfrac{1}{\beta_i(n+1)} + \dfrac{1}{2} \\ \mathrm{w}_i(n+1), \text{ caso contrário} \end{cases}$$

vii. Incrementar *n* e voltar ao passo *ii* para o próximo padrão de entrada até que todos os w_i's sejam mutuamente ortonormais.

2.2. Algoritmo GHALO

De acordo com o sumário do algoritmo ALA apresentado acima e com algumas simulações realizadas para o caso de compressão de imagens, comprovou-se que a Eq. (3) é responsável pela introdução de uma complexidade computacional bastante intensa ao treinamento da rede neural, acarretando em elevado tempo de convergência em certas aplicações práticas como, por exemplo, no processo de compressão de imagens.

Portanto, este problema foi contornado com o desenvolvimento do algoritmo GHALO. A principal contribuição deste novo algoritmo foi a sua forma simples e eficiente de estimação dos autovalores da matriz de autocorrelação **R**, definida por:

$$\mathbf{R} = \mathrm{E}[\,\mathbf{x}\,\mathbf{x}^T\,] \quad (7)$$

O desenvolvimento matemático abaixo apresenta o processo de estimação dos autovalores da matriz **R** e, com isto, determina-se de forma adaptativa a taxa de treinamento conforme a Eq. (5).

Considerando a equação dos autovalores definida por:

$$\mathbf{R}\,\mathbf{U} = \mathbf{U}\,\mathbf{D} \quad (8)$$

onde **R** é a matriz de correlação, **U** é a matriz dos autovetores organizados em vetores coluna, e **D** é a matriz diagonal que contém os autovalores. O método GHA utiliza autovetores ortonormais (vetores unitários) tais que $\mathbf{U}^T = \mathbf{U}^{-1}$. Então pode-se reescrever a Eq. (8) como:

$$\mathbf{U}^T \cdot \mathbf{R} \cdot \mathbf{U} = \mathbf{U}^T \cdot \mathbf{U} \cdot \mathbf{D} \\ \mathbf{U}^T \cdot \mathbf{R} \cdot \mathbf{U} = \mathbf{D} \quad (9)$$

Seja **W**, matriz dos ganhos sinápticos. De acordo com o algoritmo GHA [7] , os ganhos sinápticos tendem a convergir para os autovetores da matriz de autocorrelação, logo, **W** converge para a matriz $\mathbf{U}^T$ e , assim, a Eq. (9) pode ser escrita como: como:

$$\mathbf{W}\,\mathbf{R}\,\mathbf{W}^T = \mathbf{D} \quad (10)$$

A matriz de autocorrelação de forma estocástica e instantânea pode ser estimada pela seguinte expressão:

$$\hat{\mathbf{R}}(\mathrm{n}) = \mathbf{x}(\mathrm{n})\,\mathbf{x}^T(\mathrm{n}) \quad (11)$$

observando-se que foi desprezado o operador estatístico. Aplicando-se esta estimativa na Eq. (10), tem-se:

$$\hat{\mathbf{W}}(\mathrm{n})\,\hat{\mathbf{R}}(\mathrm{n})\,\hat{\mathbf{W}}^T(\mathrm{n}) = \hat{\mathbf{D}}(\mathrm{n}) \quad (12)$$

Substituindo-se a Eq. (11) na Eq. (12) e observando-se que

$$\hat{\mathbf{y}}(\mathrm{n}) = \hat{\mathbf{W}}(\mathrm{n})\,\mathbf{x}(\mathrm{n}) \quad (13)$$

tem-se:

$$\hat{\mathbf{D}}(\mathrm{n}) = \hat{\mathbf{y}}(\mathrm{n})\,\hat{\mathbf{y}}^T(\mathrm{n}). \quad (14)$$

Portanto, através da Eq. (14), é possível estimar os autovalores da matriz **R**, evitando assim a complexidade computacional imposta pela Eq. (3) do algoritmo ALA. Conforme será observado na análise dos resultados, isto implica em uma brusca redução do tempo de processamento do algoritmo, além de contribuir para um comportamento estável na curva de convergência.

3. Resultados

Nesta seção será inicialmente observada o desempenho do algoritmo GHALO no referente a convergência comparando o seu desempenho com os algoritmos GHA e ALA. O referido algoritmo será utilizado na compressão de imagens onde os aspectos de capacidade de compactação, generalização e recuperação da imagem serão observados. Foram utilizadas imagens de dimensão 128 x 128 pixels como mostra a Fig. 1.

(a)

(b)

(c)

Figura 1 - Imagens originais utilizadas

Para que os dados referentes a uma imagem possam ser utilizados como padrões de entrada para a rede neural realizam-se os seguintes procedimentos [10]:

- Segmentação dos dados em blocos 8 x 8;
- Construção de vetores colunas a partir desses blocos, resultando em vetores de 64 elementos, que serão as entradas das Redes Neurais;
- Normalização destes elementos em uma faixa entre 0 e 1;
- Eliminação do nível médio.

A partir deste ponto os dados encontram-se prontos para serem utilizados como entrada do sistema, dando início ao treinamento neural.

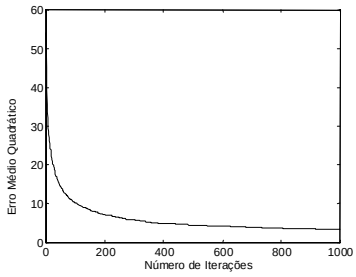
3.1. Análise de convergência

Inicialmente foi realizado um treinamento com a tradicional imagem da Lena da Fig. 1.b para efeito de comparação de convergência entre os algoritmos GHA e ALA e do mecanismo de aceleração de convergência desenvolvido.

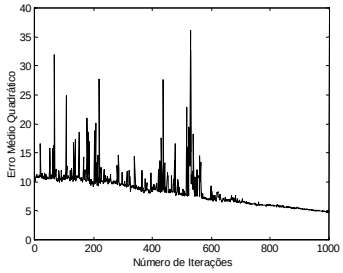
Para o caso do GHA, foi necessário primeiramente realizar uma análise para determinar o valor adequado para a taxa de treinamento η. Para o caso do ALA, esta dificuldade foi contornada tendo em vista que a taxa de treinamento é adaptativa mas, em contrapartida, houve outros parâmetros a serem estimados, o que resultou em uma tarefa bastante engenhosa.

Os gráficos apresentados nas Fig. 2 e 3 ilustram uma comparação de desempenho de treinamento dos algoritmos GHA, ALA e GHALO, onde tais algoritmos foram utilizados para o treinamento e compactação da imagem da Lena da Fig. 1.b com taxas de compressão da ordem de 64:32 e 64:8, respectivamente, e número de iterações igual a 1000.

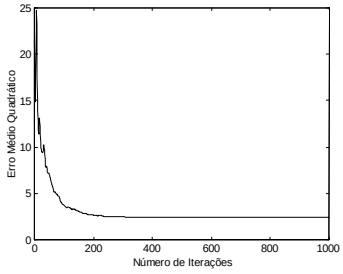
Para o GHA, foi utilizada uma taxa de treinamento bastante pequena, cujo valor adequado observado foi de 0.001. Quando utilizado um valor maior, o algoritmo tornava-se mais rápido porém divergia em alguns casos. Quando utilizado um valor menor, a convergência do algoritmo tornava-se mais lenta. Os valores iniciais adequados encontrados para os parâmetros λ, β e γ do algoritmo ALA foram, respectivamente, 0.01, 0.09 e 0.005.



(a)



(b)

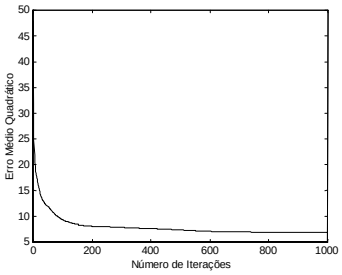


(c)

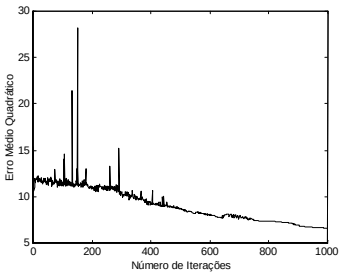
Figura 2. Gráficos de desempenho de treinamento para taxa de compressão 64:32. (a) Gráfico do GHA. (b)Gráfico do ALA. (c) Gráfico do GHALO.

Conforme mostram as Figs. 2.a e 3.a, a curva de desempenho do GHA foi bastante estável, porém o treinamento foi lento devido ao baixo valor utilizado para a taxa de treinamento. Para o ALA, como pode-se observar através das Figs. 2.b e 3.b, houve uma certa instabilidade no treinamento, mas com poucas iterações o algoritmo convergiu para o mesmo valor final do erro médio quadrático do treinamento realizado com o algoritmo GHA. Utilizando o GHA com o mecanismo de aceleração de convergência desenvolvido, a convergência do algoritmo tornou-se bem mais rápida, conforme mostram as Figs. 2.c e 3.c. Isto pode ser explicado devido às características de estabilidade do método aqui apresentado.

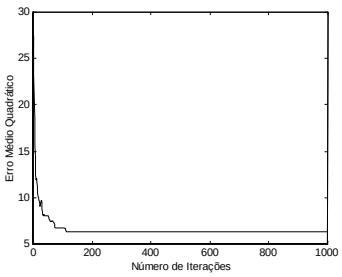
Verifica-se portanto que o algoritmo do GHALO desenvolvido neste trabalho possibilitou uma aceleração bastante significativa na convergência, tornando-se, desta forma, mais adequada a sua utilização na compressão de imagens.



(a)



(b)



(c)

Figura 3. Gráficos de desempenho de treinamento para taxa de compressão 64:8.
(a) Gráfico do GHA. (b)Gráfico do ALA.
(c) Gráfico do GHALO.

3.2. Compactação de imagens e generalização

O objetivo nesta seção é aplicar o algoritmo GHALO na compressão de imagens [1], [4], [8], [9], verificando sua capacidade de generalização e analisando a recuperação das imagens originais. O treinamento foi realizado para a imagem da Fig. 1.a.

A generalização foi efetuada para as imagens das Figs. 1.b. e 1.c. utilizando os pesos sinápticos determinados na fase de treinamento da rede neural. É importante salientar que não é necessário realizar outro treinamento quando deseja-se compactar uma imagem diferente daquela utilizada no treinamento, caracterizando dessa forma a capacidade de generalização da rede neural.

(a)

(b)

Figura 4. Imagens com compactação 64:32

(a)

(b)

Figura 5. Imagens com compactação 64:16

(a)

(b)

Figura 6. Imagens com compactação 64:8

As taxas de compressão foram 64:32, 64:16 e 64:8, e as imagens reconstruídas após a compactação e descompactação são mostradas nas Figs. 4, 5 e 6, respectivamente.

A presença de distorções na imagem reconstruída já era esperada, tendo em vista que foi realizado um processo que acarreta em perda de elementos constituintes da imagem original.

Um ponto importante a salientar é que a taxa de compressão a ser desenvolvida é função do nível de distorção aceitável para aplicação de interesse. O mecanismo de aceleração de convergência desenvolvido neste trabalho mostrou-se não somente mais rápido que GHA convencional assim como o ALA, como apresentou uma excelente capacidade de generalização.

A análise do efeito da compactação sobre as imagens é feita através dos seguintes parâmetros: relação sinal ruído (SNR), Erro Médio Quadrático (MSE), Média e Variância. As tabelas 1, 2 e 3 mostram os resultados referentes às compressões 64:32, 64:16 e 64:8 respectivamente.

Tabela 1. Compactação 64:32

Imagem	SNR	MSE	Média	Variância
Lena	53.40	2.58	0.3843	0.0394
Barbie	58.20	2.12	0.4360	0.0293

Tabela 2. Compactação 64:16

Imagem	SNR	MSE	Média	Variância
Lena	42.94	4.72	0.3843	0.0385
Barbie	47.50	3.96	0.4360	0.0287

Tabela 3. Compactação 64:8

Imagem	SNR	MSE	Média	Variância
Lena	35.97	7.05	0.3843	0.0369
Barbie	40.65	5.84	0.4360	0.0276

4. Conclusão

Neste trabalho foi apresentado o desenvolvimento e aplicação do algoritmo hebbiano generalizado com taxa de treinamento adaptativa e otimizada no aspecto de complexidade computacional, denominado de GHALO. Este algoritmo foi comparado em desempenho com os algoritmo GHA e ALA, mostrando um melhor desempenho nos aspectos de convergência, capacidade de compressão e generalização especialmente quando aplicado ao complexo problema de compressão de imagens.

5. Referências

[1] T. D. Sanger, "Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Networks," Neural Networks, **12**: 459-473, 1989.

[2] E. Oja, "Neural Networks, A Simplifield Neuron Model as a Principal Components Analizer,," J. Math. Biology, **15**: 267-273, 1982.

[3] L. H. Chen e S. Chang , "An Adaptive Learning Algorithm for Principal Component Analysis," IEEE Trans. on Neural Networks, **6**: 5,1995.

[4] M. C. Castro, F. C. Castro, J. N. Amaral e P. R. G. Franco, "A New Algorithm to Reduce the Computacional Complexity of Principal Components Analysis by Hebbian Learning," III Congresso Brasileiro de Redes Neurais, 7 –11, Florianópolis, Brasil, 1997.

[5] P. Baldi e K. Hornik, "Neural Networks and Principal Components Analysis: Learning from Examples without Local Minima," Neural Networks, **2**, 53-58,1989.

[6] P. Fodiák, "Adaptive Network for Optimal Linear Feature Extraction," Porc. Int. Joint Conf. Neural Networks, **1**: 401-405, San Diego, 1989.

[7] S. Haykin, *Neural Networks.* Macmillan College New York, NY, 1994.

[8] A. R. Souza, A. D. Dória Neto., L. Barbalho Jr. e M. A. G. Carvalho, "Self-Organized Neural Systems Applied to Data/Images Compression/Reconstruction," IEEE Int. Conf. on Intelligent Signal Processing and Communication Systems, 392-395, Nov. 1998, Melbourne, Austrália.

[9] R. Pizzio, J. N. Amaral e P. R. G. Franco, “Compressão de Images de Cintolografia Miocárdia através do Algoritmo Hebbiano Generalizado,” III Congresso Brasileiro de redes Neurais, 190-194, Florianopólis, Julho, 1997.

[10] A. R. Sousa, “Sistema Neural Auto-Supervisonado Aplicado à Compressão/Reconstrução de Dados/Imagens,” Tese de Mestrado, UFRN, Dezembro, 1998.