

EXTRAÇÃO NEURAL DE COMPONENTES PRINCIPAIS LINEARES EM APLICAÇÕES DE ELEVADA DIMENSIONALIDADE

J.B.O SOUZA FILHO, M.S. SOUZA, L.P. CALÔBA, J.M. SEIXAS

*Laboratório de Processamento de Sinais (LPS) - COPPE/EP/UFRJ
CP 68504, Rio de Janeiro 21945-970, RJ, Brazil*

E-mails: nash@lps.ufrj.br, marcs@lps.ufrj.br, caloba@lps.ufrj.br, seixas@lps.ufrj.br

Resumo— A análise das componentes principais é um método frequentemente utilizado para redução de dimensionalidade de grandes conjuntos de dados. Métodos clássicos ou neurais podem ser utilizados para extrair as componentes principais. Neste trabalho, um novo método neural para extração de componentes é proposto e comparado com métodos clássicos e neurais ordinários, sendo considerados nesta análise a qualidade da extração das componentes, o esforço computacional e a complexidade dos algoritmos. Para isto, um processo estocástico de uma aplicação de sonar com 60 componentes é utilizado. É mostrado que o método proposto é capaz de extrair de forma acurada e eficiente as principais componentes deste processo.

Abstract— Principal components analysis (PCA) is frequently used for dimensionality reduction of large data sets. Classical or neural methods can be used to extract the principal components. In this work, a new neural method for components extraction is proposed and compared with classical and ordinary neural methods, considering the quality of component extraction, the computational effort and the complexity of the algorithms. For this, a stochastic process from a sonar system application with 60 components is used. It is shown that the new neural method is able to extract accurately and efficiently the main components of this process.

Keywords— Adaptive principal components extraction, hierarchical neural network, principal components analysis, dimensionality reduction, backpropagation

1 Introdução

Um grande número de problemas, seja na área técnica, médica ou financeira, envolve ambientes com um grande número de variáveis. Em várias ocasiões, os conjuntos de dados relacionados possuem uma grande redundância entre suas variáveis, ou seja, grande parte da informação contida nos dados não é relevante ou não contribui de forma significativa para solução de um dado problema. Um exemplo é a construção de classificadores neurais. Para a maioria dos casos, a adoção de um sistema de pré-processamento se faz necessária, sem o qual a eficiência de discriminação da rede e/ou seu custo computacional podem não se apresentar de forma satisfatória.

Em geral, o objetivo dos sistemas de pré-processamento é enfatizar as características relevantes do processo, reduzindo o número de variáveis envolvidas. Com dados de menor dimensionalidade, a rede neural consegue discernir melhor quais são os atributos relevantes no espaço de dados e, assim, um melhor desempenho pode ser obtido. Para problemas de classificação em tempo real, por exemplo, a redução de dimensionalidade pode ser vital, sem a qual a operação de um dado sistema pode ser inviável.

Encontrar um método onde um conjunto de dados possa ser representado por um menor número de variáveis, com o mínimo de correlação entre si, e que seja ótimo do ponto de vista do erro-médio quadrático, “preservando ao máximo” as características relevantes do conjunto original é bastante interessante. A análise das componentes principais (PCA) é um método que atende os requisitos anteriores, sen-

do utilizado em um grande número de problemas que envolvem compactação baseada em energia [1].

A PCA pode ser realizada através de métodos clássicos ou neurais. Os métodos clássicos envolvem a decomposição por valor singular ou o cálculo da matriz de autocovariância e a extração dos autovetores e autovalores desta matriz. Quanto aos métodos neurais, redes de treinamento supervisionado ou não-supervisionado podem ser utilizadas.

A despeito de maior volume de processamento para extração de todas as componentes de um processo, os métodos neurais permitem a extração de um número qualquer de componentes e, por serem processos iterativos, permitem diferentes níveis de acuidade na estimativa. Redes construtivas são especialmente interessantes uma vez que, extraído certo número de componentes, é possível promover a extração de componentes de ordem superior, sem a necessidade de realizar um novo processo de treinamento desde o início. Por outro lado, na extração neural, quanto maior o número de passos de treinamento, melhor a estimativa produzida, logo é possível estabelecer um compromisso entre a qualidade necessária a estimativa e o custo computacional envolvido.

Outra vantagem do processo de extração neural é a questão da complexidade de implementação. A extração neural possui um algoritmo sensivelmente mais simples que o método clássico e, por consequência, exige menos recursos computacionais (complexidade de controle, cálculo de funções, utilização de memória), sendo especialmente indicado para implementações de sistemas de extração e clas-

sificação em microcontroladores, processadores digitais de sinais ou mesmo em *hardware*.

Neste trabalho é proposto um novo método de extração de componentes realizado através de uma rede construtiva *auto-supervisionada modificada* treinada pelo método de *backpropagation*. A qualidade das componentes extraídas e custo computacional envolvido no processo de extração são comparadas com os obtidos em um rede *backpropagation auto-supervisionada* ordinária [2] e com redes não-supervisionadas (Sanger [3] e APEX [4]). Inicialmente, descrevemos a análise PCA e alguns métodos neurais de extração e realizamos a proposição do novo método mencionado. Logo após, caracterizamos o processo utilizado e apresentamos as simulações realizadas para os vários métodos apresentados. As conclusões pertinentes e as referências bibliográficas encerram este trabalho.

2 Análise de Componentes Principais (PCA)

A análise das componentes principais é baseada na expansão das séries de *Karhunen-Löeve* [5]. Dado um espaço de dados de dimensão N , a PCA obtém um conjunto de M vetores ortogonais (direções principais), onde $M \leq N$, tal que a variância da projeção do conjunto de dados na direção de cada um destes vetores é máxima. Em processos de média nula, como a variância é igual à energia, a PCA fornece direções de máxima concentração de energia.

Para a grande parte das aplicações, verifica-se que a energia se concentra em um número reduzido de componentes. Assim, a PCA pode ser utilizada para uma efetiva redução da dimensão do espaço de dados, realizada através da projeção e reconstrução do conjunto dos dados nas direções de maior energia do processo, ou seja, nas direções que mais significativamente representam o processo no sentido do erro médio quadrático.

Recentes estudos com classificadores neurais, em áreas tão diversas como física experimental e processamento de sinais de sonar passivo, mostraram que a PCA pode reduzir em mais de 93% o número de componentes necessárias para realizar a tarefa de classificação [6],[7], mesmo operando sobre espaços de dimensão bastante elevada (entre 496 e 800 direções) viabilizando a necessária operação *on-line* destas aplicações.

3 Métodos Neurais para PCA

3.1 Treinamento não-supervisionado I (Sanger)

Segundo *Oja*, um neurônio com regra de aprendizado tipo *Hebbiana* constitui um filtro para primeira componente da distribuição de entrada. Por *Sanger*, este modelo pode ser expandido para uma rede de uma camada, com todos neurônios lineares, reali-

zando uma PCA para tamanhos arbitrários de entrada [3]. A atualização dos pesos desta rede é realizada pelo *Generalized Hebbian Algorithm* (GHA), ou seja:

$$\Delta w_{ji}(n) = \eta \cdot [y_j(n) \cdot x_i(n) - y_j(n) \cdot \sum_{k=1}^j w_{ki}(n) \cdot y_k(n)]$$

onde: $\Delta w_{ji}(n)$ é atualização sofrida pelo peso w_{ji} na n -ésima iteração e η é a taxa ou fator de aprendizado.

Para esta rede, de acordo com o teorema de convergência provado por Sanger [3], no limite:

$$\Delta w_j(n) \rightarrow 0, w_j \rightarrow q_j \text{ e } |w_j(n)| = 1 \quad n = 1, \dots, N,$$

ou seja, a rede converge para os autovetores normalizados (q_j) atribuídos aos k maiores autovalores da matriz de correlação do espaço de entrada, ordenados em ordem decrescente.

3.2 Treinamento não-supervisionado II (APEX)

Este método foi proposto por Kung e Diamantaras [4] e consiste em uma rede neural linear de n entradas conectadas as m saídas através de pesos *feed-forward* (w_{ij}). Adicionalmente, existem pesos laterais (c_j) que conectam as saídas das $m-1$ neurônios anteriores com o m -ésimo neurônio, promovendo a ortogonalização dos seus pesos sinápticos com as estimativas das componentes principais armazenadas nos pesos dos $m-1$ neurônios anteriores.

Na etapa de propagação das entradas do m -ésimo neurônio, temos:

$$W_m = [w_1 \quad w_2 \quad \dots \quad w_{m-1}]^T$$

$$y_m(n) = W_m \cdot x(n) \text{ e } y_m(n) = w_m^T \cdot x(n) - c_m(n)^T \cdot y(n)$$

O ajuste dos pesos w_m e c_m é dado por:

$$\Delta w_m(n) = \beta \cdot y_m(n) \cdot \{x(n) - y_m(n) \cdot w_m(n)\}$$

$$\Delta c_m(n) = \beta \cdot y_m(n) \cdot \{y(n) - y_m(n) \cdot c_m(n)\}$$

Conforme deduções realizadas em [4], se fizermos $c_m = W_m \cdot w_m$, podemos reescrever y_m na forma:

$$y_m(n) = w_m^T \cdot \{x(n) - W_m^T \cdot W_m \cdot x(n)\} = w_m^T \cdot x'_m(n) = z_m,$$

onde: $x'_m(n)$ corresponde a $x(n)$ quando extraídas todas suas projeções nas direções das $m-1$ estimativas de componentes anteriores e z_m pode ser visto como a saída de um neurônio linear da camada escondida de uma rede de duas camadas, considerando que cada um destes neurônios está submetido a um conjunto de variáveis de entrada distintas dadas por $x'_m(n)$.

Reescrevendo a equação de ajuste dos pesos w_m , temos:

$$\Delta w_m(n) = \beta \cdot z_m(n) \cdot \{x(n) - z_m(n) \cdot w_m(n)\} = \beta \cdot z_m(n) \cdot e_m(k)$$

Observando a equação anterior, percebemos que ela é idêntica à equação de treinamento dos pesos que conectam os neurônios da camada escondida aos neurônios de saída de uma rede *backpropagation* auto-supervisionada. Assim, a rede APEX pode ser vista como uma rede *backpropagation* onde apenas a camada de reconstrução é treinada e seus pesos são replicados na camada de projeção. A diferença principal no método APEX reside no fato de que o produto que determina c_m não é calculado e sim estimado a cada iteração.

3.3. Treinamentos Supervisionados

Para extração supervisionada, redes de duas camadas ordinárias, com neurônios lineares, treinadas pelo algoritmo de *backpropagation* podem ser utilizadas [8]. Adotando um processo de extração sequencial, utilizamos uma rede construtiva, ou seja, uma estrutura onde neurônios são adicionados à camada escondida ao longo do processo de extração.

O processo de extração é inicializado com uma rede de N nós de entrada, um neurônio na camada intermediária e N nós de saída, onde N é o número de variáveis do processo. Os vetores alvo adotados correspondem às próprias entradas (treinamento auto-supervisionado) e os pesos são inicializados de forma aleatória. Esta rede é treinada até a estabilização do erro-médio quadrático. Ao final do processo, duas estimativas, em geral distintas, da primeira componente são fornecidas pelos pesos que conectam o neurônio da camada intermediária aos nós de entrada (1ª estimativa) e aos neurônios da camada de saída (2ª estimativa).

Para extração da segunda componente, mais um neurônio é adicionado à camada intermediária e as sinapses a ele relacionadas são inicializadas aleatoriamente. O treinamento é reinicializado, mantendo constantes todos pesos relacionados às componentes anteriormente extraídas, conforme ilustrado na Figura 1, onde as sinapses representadas em linha contínua correspondem às sinapses congeladas e aquelas representadas em linha tracejada correspondem às sinapses em treinamento. Este processo é repetido até a extração das M componentes desejadas.

É importante ressaltar o papel de cada camada na constituição da saída. Podemos observar que a camada escondida constitui uma camada de projeção do espaço de dados de entrada na direção de uma das estimativas da primeira componente e que a camada de saída constitui uma camada de reconstrução, tendo como direção a segunda estimativa da componente.

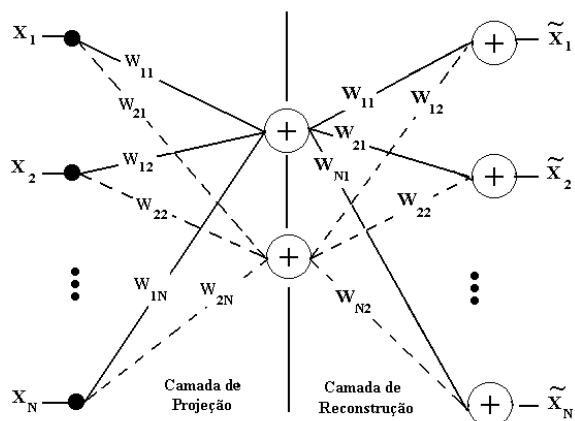


Figura 1. Rede Supervisionada na Extração da Segunda Componente.

Algumas observações sobre este método podem ser realizadas:

- i) Os pesos da camada de projeção e reconstrução não são iguais, ou seja, temos um problema de estimação de $2k$ direções, das quais apenas k direções serão utilizadas.
- ii) Os pesos não possuem módulo unitário, ou seja, cabe ao algoritmo de treinamento ajustar os pesos de forma que o produto de seus módulos tenha valor unitário.
- iii) A convergência da camada de reconstrução para uma boa estimativa da componente em questão é sensivelmente mais rápida que a da camada de projeção.

As observações anteriores nos levaram à pesquisa de um novo método de treinamento, onde pesos da camada de projeção e reconstrução fossem inicializados iguais e sua atualização fosse semelhante ao longo de todo treinamento, de forma que o problema de estimação envolvesse apenas a estimação de k direções. Considere a extração da p -ésima componente. A saída do j -ésimo neurônio é dada por:

$$\tilde{X}_j = W_{jp} \left(\sum_{i=1}^N X_i \cdot W_{pi} \right) + K_{p-1} \quad (I)$$

onde: X_j é a saída do j -ésimo neurônio e K_{p-1} é a contribuição das $(p-1)$ componentes já extraídas.

Fazendo $W_{jp} = W_{pj}$ para $\forall p$ e $\forall j$, a sensibilidade de cada saída em relação a um dado peso é dada por:

$$\frac{\partial \tilde{X}_j}{\partial W_{pk}} = \delta(j-k) \cdot \sum_{i=1}^N X_i \cdot W_{pi} + W_{pj} \cdot X_k \quad (II)$$

onde: $\delta(j-k) = 1$, se $j=k$ e $\delta(j-k) = 0$, se $j \neq k$

O gradiente do erro-médio quadrático para um padrão de entrada pode ser escrito como:

$$\frac{\partial emq^2}{\partial W_{pk}} = -2 \cdot \sum_{j=1}^k (X_j - \tilde{X}_j) \cdot \frac{\partial \tilde{X}_j}{\partial W_{pk}} \quad (\text{III})$$

Considerando:

$$\Delta W_{pk} = -\alpha \frac{\partial emq^2}{\partial W_{pk}}, \quad \varepsilon_k = X_k - \tilde{X}_k, \quad \text{as eq. II e III, a}$$

regra de atualização dos pesos [9] é dada por:

$$\Delta W_{pk} = 2 \cdot \alpha \cdot X_k \cdot \sum_{j=1}^N \varepsilon_j \cdot W_{pj} + 2 \cdot \alpha \cdot \gamma, \quad \text{onde: } \gamma = \varepsilon_k \cdot \sum_{i=1}^N X_i \cdot W_{pi} \cdot \gamma$$

É possível observar que a equação de treinamento anterior é idêntica à equação de treinamento por *backpropagation* de uma rede de duas camadas de neurônios lineares no que se refere aos pesos da camada de projeção, excetuando-se o fator $2 \cdot \alpha \cdot \gamma$, relacionado ao efeito sobre a função objetivo erro-médio quadrático da manutenção dos pesos de ambas camadas iguais.

4 Caracterização Clássica do Processo

O método clássico forneceu uma base comparativa para a comparação da qualidade das componentes extraídas pelos métodos neurais. Alguns autovalores do processo utilizado são apresentados na Tabela 1.

Tabela 1. Alguns autovalores obtidos na Análise Clássica.

C	Autovalor	C	Autovalor
1	0,5413	6	0,0827
2	0,3082	7	0,0710
3	0,1492	8	0,0440
4	0,1148	9	0,0371
5	0,0909	10	0,0349

Para avaliar a quantidade de energia representada por um número crescente de componentes principais, procedemos a soma normalizada dos autovalores, tendo como fator de normalização a energia total do processo, ou seja, o somatório de todos autovalores obtidos. A quantidade percentual de energia dos dados por grupo de componentes principais está representada na Figura 2 e na Tabela 2. Através delas, podemos perceber o caráter de concentração de energia da PCA.

Tabela 2. Percentual de Energia por grupo de Componentes.

C	Energia	C	Energia
1	31,94 %	15	93,68 %
3	58,94 %	20	97,07 %
5	71,07 %	30	99,28 %
7	80,14 %	40	99,86 %
10	86,98 %	50	99,99 %

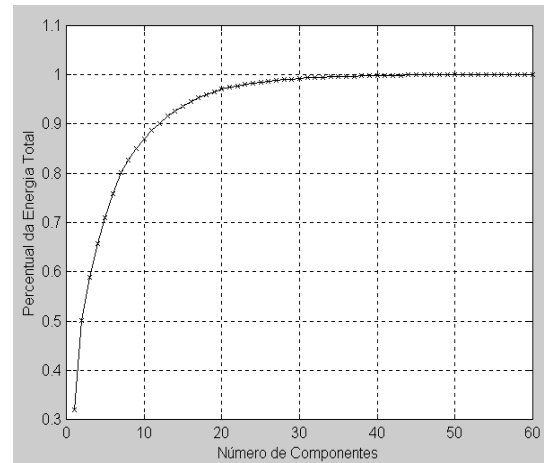


Figura 2. Percentual da Energia por Grupo de Componentes.

5 Resultados

Todas análises desenvolvidas foram implementadas através de programas desenvolvidos em linguagem C ANSI. Optamos por esta linguagem pela maior velocidade das implementações, permitindo um maior número de simulações na busca de parâmetros ótimos de treinamento, assim como por visarmos uma futura implementação em processadores digitais de sinais (DSPs).

5.1 Método não-supervisionado I (Sanger)

O conjunto de dados foi dividido em conjuntos de treino, teste e validação, com 104, 52 e 52 eventos, respectivamente. Adotamos a regra-delta como processo de treinamento, ou seja, uma atualização de pesos é realizada para cada evento do conjunto de treinamento. A taxa de aprendizado (η) e de decréscimo de aprendizado (γ) foram 0,2 e 0,999985, ou seja, a cada passo de treinamento, o valor de η é multiplicado por γ , caracterizando um decaimento exponencial. Verificamos que a rede foi capaz de extrair 27 componentes, através de 500.000 passos de treinamento. Os parâmetros apresentados foram levantados através de algumas simulações e representam o melhor compromisso entre custo computacional e qualidade de extração.

Para avaliar a qualidade do processo de extração, ou seja, as direções das componentes extraídas e a ortogonalidade entre as diferentes componentes, o gráfico da matriz de ângulos apresentado na Figura 3 foi realizado. Cada elemento ij desta matriz corresponde ao módulo do ângulo formado entre a i -ésima componente fornecida pelo método clássico e o j -ésima componente extraída pelo método neural. Podemos observar que as componentes corretamente extraídas pelo método neural apresentam ângulos pequenos em relação as componentes clássicas e a ortogonalidade entre componentes distintas é mantida. Na Tabela 3, os ângulos obtidos para as 28 primeiras componentes são apresentados.

Tabela 3. Módulo dos Ângulos por Componente (graus).

C	Valor	C	Valor	C	Valor	C	Valor
1	1,08	8	4,20	15	10,08	22	14,81
2	0,78	9	5,81	16	17,09	23	11,59
3	3,32	10	8,29	17	11,76	24	21,07
4	3,63	11	4,56	18	7,10	25	16,16
5	5,87	12	27,89	19	4,79	26	8,62
6	3,59	13	28,26	20	9,71	27	8,73
7	4,03	14	4,17	21	10,65	28	37,99

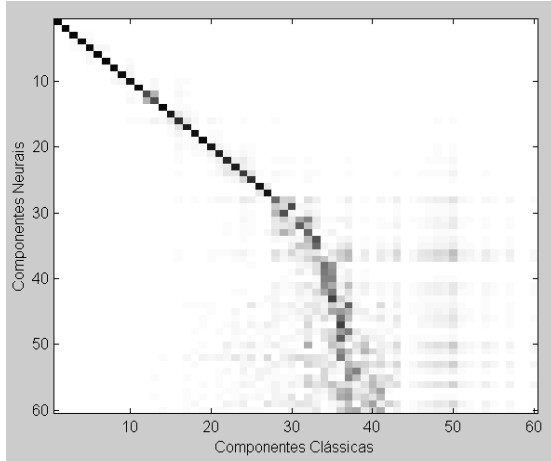


Figura 3. Ângulos entre as Componentes Clássicas e as Componentes extraídas pela rede Neural (método Sanger).

Diferentemente dos métodos supervisionados construtivos, neste método, todas componentes são estimadas de forma simultânea. Simulações envolvendo o método de batelada também foram realizadas, porém os resultados obtidos não compensaram seu maior custo computacional. Vale lembrar que no método batelada, para uma atualização de pesos é utilizado o valor médio das atualizações calculadas para cada evento do conjunto de treinamento

5.2 Método não-supervisionado II (APEX)

Dada a analogia verificada entre o método APEX e o *backpropagation auto-supervisionado*, foi verificado que para uma qualidade de estimação equivalente à obtida nos métodos supervisionados, um número maior de passos de treinamento foi necessário, o que está relacionado a estimação de c_m ao longo do treinamento. Os resultados obtidos nas simulações não se apresentaram significativos e não serão apresentados.

5.3 Métodos supervisionados

Designamos o método que realiza duas estimações (projeção e reconstrução independentes) como método A e o outro, como método B (projeção e reconstrução iguais). O conjunto de dados foi dividido em 104 eventos para o conjunto de treino e 105 eventos para o conjunto de teste. O método de batelada foi adotado. A cada época de treinamento, o EMQ conjunto de teste foi avaliado, onde consideramos que

uma época de treinamento corresponde a dez atualizações do pesos da rede.

Para o método A, o melhor compromisso entre a velocidade de treinamento e a qualidade de extração é obtido para valores da taxa de aprendizado (η) entre 0.5 e 1. Adotamos $\eta=0.85$. Como critério de mudança da componente em extração, adotamos a estabilização do EMQ para um valor de $5 \cdot 10^{-7}$, ou seja, caso a diferença entre o EMQ atual e o EMQ da iteração anterior atinja um valor inferior a esta constante, a componente é considerada extraída. Para extração das 60 componentes, 85748 passos de treinamento foram necessários. A Tabela 5 apresenta o número de passos envolvidos na extração de algumas componentes. Através dela, o custo computacional da extração de cada componente pode ser avaliado para o método proposto.

Tabela 4. Passos para Extração das Componentes.

C	Passos	C	Passos
1	505	6	693
2	395	7	198
3	239	8	336
4	1507	9	83
5	2915	10	208

Quanto a qualidade das componentes extraídas, verificamos que as estimativas fornecidas pela camada de reconstrução são sensivelmente melhores que as da camada de projeção. O processo de extração não é crítico quanto as direções da camada de projeção, uma vez que seu efeito é a diminuição do módulo das projeções, a qual é compensada pelo ajuste do módulo dos vetores da camada de reconstrução. Comparando as direções fornecidas pelo método clássico e pelo método A, verificamos que o método foi capaz de extrair 27 componentes do processo. Na Figura 4, o gráfico da matriz de ângulos é apresentado. Na Tabela 5, apresentamos alguns ângulos entre as componentes fornecidas por ambos métodos.

Tabela 5. Módulo dos Ângulos por Componente (graus).

C	Valor	C	Valor	C	Valor	C	Valor
1	0,54	8	0,59	15	9,72	22	11,4
2	0,39	9	3,23	16	21,7	23	14,6
3	1,68	10	7,14	17	17,1	24	28,4
4	0,55	11	6,11	18	7,1	25	27,5
5	5,87	12	44,2	19	5,78	26	8,98
6	5,68	13	43,2	20	10,9	27	3,28
7	0,14	14	2,81	21	10,6	28	39,6

Considerando o método B, adotamos como taxa de aprendizado o valor 0.75. Para extração das 60 componentes, um total de 62764 passos de treinamento foram necessários. O critério de mudança da componente em extração foi o mesmo adotado no método A. A Tabela 6 apresenta o número de passos envolvidos na extração de algumas componentes.

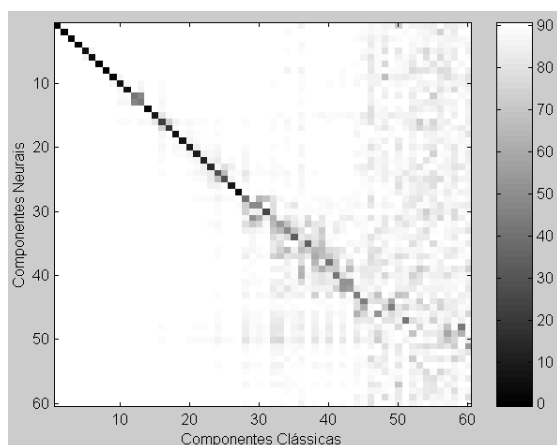


Figura 4. Ângulos entre as Componentes Clássicas e as Componentes extraídas pela rede Neural (método A).

Tabela 6. Passos para Extração das Componentes.

C	Passos	C	Passos
1	16	6	113
2	19	7	69
3	61	8	203
4	74	9	734
5	158	10	60

Um total de 23 componentes foram extraídas pelo método, a um custo computacional sensivelmente menor. O gráfico da matriz de ângulos é apresentado na Figura 5. Na Tabela 7, apresentamos os ângulos obtidos na comparação deste método com o método clássico. Podemos perceber que do ponto de vista de qualidade de estimação, este método é equivalente ao anterior, em especial, em relação às componentes mais energéticas.

Tabela 7. Módulo dos Ângulos por Componente (graus).

C	Valor	C	Valor	C	Valor	C	Valor
1	0,54	8	0,59	15	7,05	22	11,75
2	0,39	9	3,12	16	18,0	23	12,32
3	1,68	10	10,3	17	13,7	24	75,1
4	0,55	11	9,56	18	14,4	25	60,6
5	5,87	12	47,7	19	14,7	26	77,1
6	5,68	13	47,4	20	13,7	27	52,2
7	0,14	14	2,71	21	12,4	28	57,3

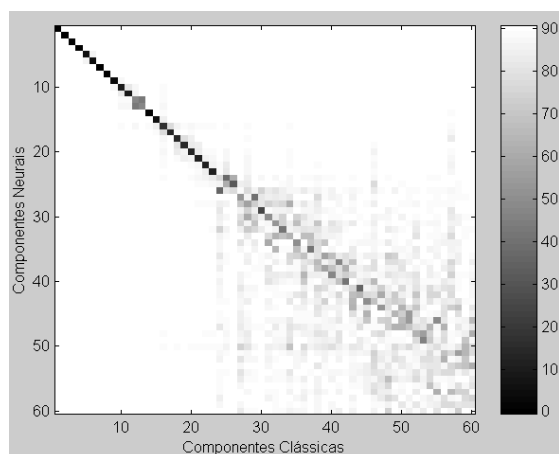


Figura 5. Ângulos entre as Componentes Clássicas e as Componentes extraídas pela rede Neural (método B).

Conclusões

Métodos neurais para extração PCA foram apresentados, discutidos e comparados quanto à qualidade das componentes extraídas e quanto ao custo computacional envolvido. A melhor relação custo-benefício entre estes parâmetros foi obtida pelo novo método proposto, o *backpropagation auto-supervisionado modificado*. Para este método, uma velocidade de convergência razoavelmente maior foi obtida e a qualidade das componentes se apresentou equivalente ou superior a dos demais, principalmente na extração das componentes mais significativas do processo.

Referências Bibliográficas

- [1] Haykin, S. *Neural Networks: a Comprehensive Foundation*, Second Edition, Prentice Hall, 1999.
- [2] Hertz, J. K and Palmer, R. G. *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991
- [3] Sanger, T. D. Optimal unsupervised learning in a single-layer feedforward neural network. *Neural Networks*, 1989.
- [4] Kung, S.Y. and Diamantaras, K.I. and Taur, J.S. Adaptive Principal Components Extraction and Applications, *IEEE Transactions on Signal Processing*, vol 42, No. 5, May 1994, pp 1202-1217
- [5] Papoulis, A.. *Probability, Random Variables, and Stochastic Processes*, Third Edition, Mc Grall-Hill, 1991.
- [6] Vassali, M. R. and Seixas, J. M. Principal Component Analysis for Neural Electron/Jet Discrimination in Highly Segmented Calorimeters, *VII International Workshop on Advanced Computing and Analysis Techniques, ACAT 2000*.
- [7] Soares-Filho and Seixas, J. M. and Calôba, L.P. Principal Components for Classification of Passive Sonar Signals. *2001 IEEE International Symposium on Circuits and Systems*, Sydney, Australia, 2001
- [8] Baldi, P. and Hornik, K. Neural Networks and PCA learning from Examples without Local Minima, *Neural Networks*, vol2, 1985, pp 53-58
- [9] Widrow, Bernard and Lehr, M. A. 30 years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation, *IEEE proceedings* v78, n9, 1990.