Capítulo 6

Redes Neurais Artificiais para Decomposição de um Espaço Vetorial em Sub-Espaços

Nos capítulos anteriores estudamos detalhadamente dois tipos de RNAs que apresentam a especial habilidade de aprender a partir de seu ambiente e, a partir do processo de aprendizado supervisionado, melhorar seu desempenho (as RNAs MLPs treinadas pelo algoritmo *backpropagation* e as RNAs RBFs).

Neste capítulo estudaremos um tipo de RNA que é submetido a um processo de aprendizado não-supervisionado (ou auto-organizado). O propósito de um algoritmo de aprendizado auto-organizado é descobrir padrões significativos ou características nos dados de entrada da RNA, e fazê-lo sem a presença de um tutor (ou seja, sem a presença de um conjunto de alvos de interesse, providos por um tutor externo).

Para atingir tal propósito, o algoritmo é provido de um conjunto de regras de natureza local, conjunto este que o habilita a aprender a computar um mapeameto entrada-saída, com específicas propriedades desejáveis. O termo "local" significa, neste contexto, que uma mudança aplicada ao peso sináptico de um neurônio é confinada à vizinhança imediata daquele neurônio.

O modelamento de estruturas de RNAs usadas para aprendizado auto-organizado tende muito mais a seguir as estruturas neuro-biológicas do que as estruturas utilizadas para o aprendizado supervisionado.

A heurística para aprendizado auto-organizado de Redes Neurais Artificiais que estudaremos neste capítulo é chamada "Algoritmo Hebbiano Generalizado" e é utilizada para proceder à Análise dos Componentes Principais (*Principal Components Analysis* -

PCA) ou Decomposição em Sub-Espaços (DSE) de um conjunto da dados de interesse. O Algoritmo Hebbiano Generalizado foi proposto por Sanger em 1989 [4] e combina a ortonormalização de Gram-Schmidt [10] ao modelo de um único neurônio linear introduzido por Oja em 1982 [4].

A Análise dos Componentes Principais de um conjunto de dados é uma técnica padrão comumente utilizada para redução da dimensionalidade de dados, em processamento de sinais.

Para introduzir o assunto, primeiramente iremos descrever os princípios que regem o aprendizado auto-organizado. Na seqüência, estudaremos a Transformada Karhunen-Loève, que constitui o conhecimento básico necessário ao estudo das RNAs utilizadas para Análise dos Componentes Principais ou Decomposição em Sub-Espaços, que são o objetivo deste capítulo.

6.1 Princípios do Aprendizado Auto-Organizado

Conforme estudamos no Capítulo 2, o aprendizado não-supervisionado (ou auto-organizado) consiste da modificação repetida dos parâmetros livres de uma RNA em resposta a padrões de ativação e de acordo com regras prescritas, até que seja desenvolvida uma desejada configuração final. A razão pela qual a configuração final obtida não é qualquer e, sim, uma configuração útil, reside na seguinte afirmativa:

Ordem global pode surgir a partir de interações locais.

Esta afirmativa é tanto válida para o cérebro, quanto no contexto de redes neurais artificiais. Em geral, muitas interações locais originalmente aleatórias entre neurônios vizinhos de uma rede podem "amalgamar-se" (ou, em outra interpretação alegórica, "cristalizar-se") em estados de ordem global e conduzir a um comportamento coerente na forma de padrões espaciais ou temporais, o que é a essência da auto-organização.

A organização ocorre em dois diferentes níveis, que interagem entre si na forma de um elo de realimentação. Estes dois níveis são:

- Atividade ⇨ Certos padrões de atividade são produzidos por uma dada rede em resposta a sinais de entrada.
- Conectividade ⇨ Os pesos sinápticos da rede são modificados em resposta a sinais neurais nos padrões de atividade, devido à plasticidade sináptica.

O elo de realimentação entre as mudanças nos pesos sinápticos e as mudanças nos padrões de atividade deve ser positivo para que se obtenha auto-organização da rede. De acordo com este critério pode ser extraído o primeiro princípio da auto-organização, devido a von der Malsburg (1990):

1. Modificações em pesos sinápticos tendem a se auto-amplificar.

O processo de auto-amplificação é restrito pelo requerimento de que modificações em pesos sinápticos devem ser baseadas em sinais disponíveis localmente, chamados sinais pré-sinápticos e pós-sinápticos.

Os requerimentos de auto-reforço e localidade especificam o mecanismo por meio do qual uma forte sinapse conduz à coincidência de sinais pré e pós-sinápticos. Por outro lado, a sinapse é aumentada em força, por tal coincidência.

Este mecanismo nada mais é do que o postulado de aprendizado de Hebb, estudado no Capítulo 2 (Processos de Aprendizado), que diz:

"Se dois neurônios em cada lado de uma sinapse são ativados simultaneamente, então a força daquela sinapse é seletivamente aumentada."

Para que o sistema possa ser estabilizado precisa haver alguma forma de competição por recursos "limitados".

Especificamente, um aumento na força de alguma sinapse na rede deve ser compensado pelo decréscimo em outras.

Assim, apenas as sinapses que obtêm sucesso podem crescer, enquanto as que obtêm menos sucesso tendem a enfraquecer e podem eventualmente desaparecer.

Esta observação nos leva a abstrair o segundo princípio da auto-organização (von der Malsburg, 1990):

2. Limitação de recursos conduz à competição entre as sinapses e, conseqüentemente, à seleção das sinapses que crescem de forma mais vigorosa (portanto, as mais adequadas) às custas de outras.

Este princípio é também possível devido à plasticidade sináptica.

Consideremos agora que uma única sinapse, por conta própria, não pode produzir de forma eficiente eventos favoráveis. Para que tal seja possível, é necesssário existir cooperação entre um conjunto de sinapses que convergem para um particular neurônio, levando sinais coincidentes, fortes o suficiente para ativar aquele neurônio. Desta observação podemos abstrair o terceiro princípio da auto-organização, também devido à von der Malsburg, (1990):

3. Modificações em pesos sinápticos tendem a cooperar.

A presença de uma sinapse vigorosa pode melhorar a adequação de outras sinapses, apesar da competição global na rede. Esta forma de cooperação pode surgir devido à plasticidade sináptica, ou devido ao estímulo simultâneo de neurônios pré-sinápticos, decorrente da existência de condições propícias no ambiente externo.

Todos os três princípios de auto-organização até agora descritos são relacionados somente à própria RNA. Entretanto, para que o aprendizado auto-organizado possa desempenhar uma função útil de processamento de informação, é necessário que haja redundância nos padrões de ativação supridos à rede pelo ambiente. O quarto princípio do aprendizado auto-organizado (devido a Barlow, 1989) pode ser, então, descrito como segue:

4. Ordem e estrutura nos padrões de ativação representam a informação redundante que é adquirida pela rede neural na forma de conhecimento, a qual é pré-requisito necessário ao aprendizado auto-organizado.

Uma parte deste conhecimento pode ser obtida através da observação de parâmetros estatísticos, tais com média, variância e matriz de correlação dos dados de entrada.

Os quatro princípios do aprendizado auto-organizado constituem a base dos algoritmos adaptativos para análise de componentes principais (ou decomposição em sub-espaços) que estudaremos neste capítulo. Servem também para o estudo dos mapas auto-organizados de Kohonen (*Self-Organizing Map* – SOM) que estudaremos no capítulo seguinte deste trabalho.

6.2 A Transformação Karhunen-Lòeve para PCA ou DSE de um Espaço Vetorial

A Transformação Karhunen-Loève (KLT) projeta um conjunto $\mathbf{X}$ de vetores de dados $\underline{x} \in \Re^M$ sobre uma base ortonormal em $\Re^M$ formada pelo conjunto dos M auto-vetores $\underline{e}_m \in \Re^M$, $m = 0, 1, \cdots, M-1$, da matriz de covariância de $\mathbf{X}$. A KLT é tal que a base será orientada de acordo com as direções de maior variância de $\mathbf{X}$ em $\Re^M$ [1][2] [3][4][5][6][7].

A m-ésima projeção de $\mathbf{X}$ sobre a direção do auto-vetor $\underline{e}_m$ é chamada de m-ésimo sub-espaço (ou m-ésimo componente principal). O m-ésimo auto-valor λ_m associado ao auto-vetor $\underline{e}_m$ corresponde à variância do m-ésimo sub-espaço de $\mathbf{X}$. Ainda, a variância de cada sub-espaço é um máximo local, no universo de todas as variâncias resultantes da projeção de $\mathbf{X}$ sobre todas as possíveis direções em $\Re^M$. Embora qualquer espaço de dimensão menor do que $\Re^M$ seja um sub-espaço de $\Re^M$, este estudo refere-se muitas vezes a um sub-espaço de $\mathbf{X}$ como o conjunto dos vetores de $\mathbf{X}$ que concentram-se de

forma alinhada ao longo de uma particular direção em $\Re^M$. Ainda que esta referência não seja precisa, julgou-se válido aplicá-la devido ao conceito intuitivo nela implícito.

Talvez a mais importante utilização da Transformação Karhunen-Loève, também conhecida por Análise dos Componentes Principais (PCA), seja a redução dimensional do conjunto $\mathbf{X}$. Esta característica torna a KLT bastante popular em processamento digital de sinais por permitir que as informações mais significativas contidas em um sinal possam ser representadas, mediante a introdução de algum erro considerado aceitável, em um espaço de dados de menores dimensões do que a dimensão original $\Re^M$. Como cada sub-espaço ou componente principal é orientado de acordo com as direções de maior variância de $\mathbf{X}$ em $\Re^M$, os sub-espaços de menor variância podem ser descartados – o que resultará em uma redução dimensional ótima no sentido do Erro Médio Quadrático (MSE - *Mean Square Error*) [3][4][5][6].

O desenvolvimento do método para aplicação da KLT à $\mathbf{X}$ apresentado neste estudo segue a proposta de Haykin em [7].

Seja $\mathbf{X}$ um processo estocástico vetorial representado pelo conjunto $\mathbf{X}$ de vetores $\underline{x} \in \Re^M$. Caso $\mathbf{X}$ não possua média zero – e, portanto, caso o vetor aleatório $\underline{x}$ não possua média zero – o vetor média deverá ser subtraído dos vetores de $\mathbf{X}$ antes de iniciar a transformação, i.e., $\underline{x} = \underline{x} - E\{\underline{x}\}$ onde $E\{\cdot\}$ é operador que resulta no valor esperado estatístico (média estatística) do argumento.

Seja $\underline{e} \in \Re^M$ um vetor unitário e adimensional qualquer, sobre o qual será projetado um vetor $\underline{x} \in \mathbf{X}$. Sendo $\underline{e}$ um vetor unitário, a norma Euclidiana de $\underline{e}$ é dada pela Equação (6.1).

$$\| \underline{e} \| = \sqrt{\underline{e}^T \underline{e}} = 1 \tag{6.1}$$

Como $\underline{e}$ é unitário e adimensional, o tamanho (norma Euclidiana) do vetor que resulta da projeção de $\underline{x}$ sobre $\underline{e}$, denominada de projeção *a*, possui a mesma unidade dimensional de $\underline{x}$, e é dado pela Equação (6.2).

$$a = \underline{x}^T \underline{e} = \underline{e}^T \underline{x} \tag{6.2}$$

Portanto, a define o tamanho da projeção do vetor $\underline{x}$ na direção do vetor $\underline{e}$. A projeção a também é uma variável aleatória, com média e variância associadas à estatística do vetor aleatório $\underline{x}$.

A média da projeção a é dada por

$$E\{a\} = E\left\{\underline{e}^T \underline{x}\right\} \tag{6.3}$$

$$E\{a\} = \underline{e}^T E\{\underline{x}\} = 0 \tag{6.4}$$

A variância da projeção a é dada por

$$\sigma_a^2 = E\left\{a^2\right\} \tag{6.5}$$

$$\sigma_a^2 = E\left\{\left(\underline{e}^T \underline{x}\right)\left(\underline{x}^T \underline{e}\right)\right\} \tag{6.6}$$

$$\sigma_a^2 = \underline{e}^T E\left\{\underline{x}\,\underline{x}^T\right\}\underline{e} \tag{6.7}$$

onde $E\left\{\underline{x}\,\underline{x}^T\right\}$ é a matriz de covariância $\mathbf{C_x}$, de dimensões $M \times M$, do conjunto $\mathbf{X}$ de vetores $\underline{x} \in \Re^M$.

Nota: Na prática $\mathbf{C_x}$ é aproximado por $\mathbf{C_x} = \frac{1}{L}\sum_{i=0}^{L-1} \underline{x}_i \cdot \underline{x}_i^T$ sendo L o número total de vetores conhecidos em $\mathbf{X}$.

Retornando à Equação (6.7), como $\mathbf{C_x} = E\left\{\underline{x}\,\underline{x}^T\right\}$, pode-se escrever que

$$\sigma_a^2 = \underline{e}^T \mathbf{C_x}\, \underline{e} \tag{6.8}$$

onde observa-se que a variância σ^2 da projeção a é uma função do vetor unitário $\underline{e}$. Expressando matematicamente,

$$\sigma_a^2 = f(\underline{e}) \tag{6.9}$$

A KLT objetiva determinar o conjunto de vetores $\underline{e}$ para cuja direção em $\Re^M$ a variância da projeção *a* é máxima. Portanto, a KLT varia o vetor $\underline{e}$ no universo de todas as possíveis direções em $\Re^M$ na busca daquela direção que resulta no máximo $f(\underline{e})$. Para que a norma de $\underline{e}$ não influa no resultado de $f(\underline{e})$ durante a busca, ela deve ser constante e unitária. Esta é a razão da condição preliminar definida em (6.1).

Se $\underline{e}$ alinha-se com uma direção em $\Re^M$ do conjunto $\mathbf{X}$ tal que, nesta direção $f(\underline{e})$ resulta em um máximo local dentro do universo de busca, então, devido à menor declividade de $f(\underline{e})$ nas vizinhanças de um máximo local, para qualquer pequena variação $\delta\underline{e}$ do vetor unitário $\underline{e}$ temos que

$$f(\underline{e}+\delta\underline{e}) \approx f(\underline{e}) \tag{6.10}$$

Nota: Embora (6.10) também seja válida nas vizinhanças de um mínimo local, esta ambigüidade é implicitamente resolvida quando utiliza-se a *eigen*-estrutura de $\mathbf{C}_\mathbf{x}$ [7].

Combinando as Equações (6.8) e (6.9) obtém-se que $f(\underline{e}) = \underline{e}^T \mathbf{C}_\mathbf{x}\, \underline{e}$. Assumindo que o valor de $\delta\underline{e}$ seja suficientemente pequeno de modo que $f(\underline{e})$ não se afaste das vizinhanças do máximo local , pode-se admitir a igualdade em (6.10). Portanto,

$$f(\underline{e}) = \underline{e}^T \mathbf{C}_\mathbf{x}\, \underline{e} = f(\underline{e}+\delta\underline{e}) = (\underline{e}+\delta\underline{e})^T \mathbf{C}_\mathbf{x} (\underline{e}+\delta\underline{e}) \tag{6.11}$$

$$f(\underline{e}+\delta\underline{e}) = \underline{e}^T \mathbf{C}_\mathbf{x}\, \underline{e} + \underline{e}^T \mathbf{C}_\mathbf{x} \delta\underline{e} + \delta\underline{e}^T \mathbf{C}_\mathbf{x}\, \underline{e} + \delta\underline{e}^T \mathbf{C}_\mathbf{x}\, \delta\underline{e} \tag{6.12}$$

Como a matriz de covariância $\mathbf{C}_\mathbf{x}$ do conjunto $\mathbf{X}$ é uma matriz simétrica, tem-se que

$$\underline{e}^T \mathbf{C}_\mathbf{x} \delta\underline{e} = \delta\underline{e}^T \mathbf{C}_\mathbf{x}\, \underline{e} \tag{6.13}$$

e a Equação (6.12) pode ser reescrita sob a forma

$$f(\underline{e}+\delta\underline{e}) = \underline{e}^T \mathbf{C}_\mathbf{x} \underline{e} + 2\delta\underline{e}^T \mathbf{C}_\mathbf{x} \underline{e} + \delta\underline{e}^T \mathbf{C}_\mathbf{x} \delta\underline{e} \tag{6.14}$$

Como $\delta \underline{e}$ é muito pequeno, o termo $\left(\delta \underline{e}^T \mathbf{C}_x \, \delta \underline{e}\right)$ no lado direito de (6.12) pode ser desconsiderado. De (6.14), (6.8) e (6.9) pode-se escrever que

$$f\left(\underline{e}+\delta \underline{e}\right)= f\left(\underline{e}\right)+ 2\delta \underline{e}^T \mathbf{C}_x \underline{e} \tag{6.15}$$

Ao levar a Equação (6.10) à Equação (6.15) conclui-se que

$$\delta \underline{e}^T \mathbf{C}_x \underline{e} = 0 \tag{6.16}$$

Mas, para que a norma de $\underline{e}$ não influa no resultado de $f\left(\underline{e}\right)$ durante a busca, ela deve ser mantida constante e unitária. Portanto, são admissíveis apenas as perturbações $\delta \underline{e}$ para as quais a norma do vetor perturbado $\underline{e}+\delta \underline{e}$ permaneça unitária, ou seja

$$\left\| \underline{e}+\delta \underline{e} \right\| = 1 \tag{6.17}$$

o que equivale a dizer que

$$\left(\underline{e}+\delta \underline{e}\right)^T \left(\underline{e}+\delta \underline{e}\right) = 1 \tag{6.18}$$

Expandindo o produto $\left(\underline{e}+\delta \underline{e}\right)^T \left(\underline{e}+\delta \underline{e}\right)$, temos

$$\left(\underline{e}+\delta \underline{e}\right)^T \left(\underline{e}+\delta \underline{e}\right)= \underline{e}^T \underline{e}+\underline{e}^T \delta \underline{e}+\delta \underline{e}^T \underline{e}+\delta \underline{e}^T \delta \underline{e} \tag{6.19}$$

onde

$$\underline{e}^T \delta \underline{e} = \delta \underline{e}^T \underline{e} \tag{6.20}$$

e onde o termo $\delta \underline{e}^T \delta \underline{e}$ pode ser desconsiderado. Portanto,

$$\left(\underline{e}+\delta \underline{e}\right)^T \left(\underline{e}+\delta \underline{e}\right)= \underline{e}^T \underline{e}+ 2\, \delta \underline{e}^T \underline{e} \tag{6.21}$$

como, de (6.18), $\left(\underline{e}+\delta \underline{e}\right)^T \left(\underline{e}+\delta \underline{e}\right) = 1$ então, de (6.21)

$$\underline{e}^T \underline{e}+ 2\, \delta \underline{e}^T \underline{e} = 1 \tag{6.22}$$

mas, de (6.1), $\underline{e}^T \underline{e} = 1$ em (6.22) , e portanto

$$\delta \underline{e}^T \underline{e} = 0 \tag{6.23}$$

Infere-se da Equação (6.23) que as variações $\delta \underline{e}$ devem ser ortogonais a $\underline{e}$ durante o processo de busca da direção de máximo $f(\underline{e})$. Este é o único tipo de variação permitida a $\underline{e}$ no espaço de busca $\Re^M$ tal que a Equação (6.17) seja obedecida.

Nas Equações (6.16) e (6.23) residem as duas condições simultâneas a serem respeitadas para a determinação dos vetores $\underline{e}$ para os quais $f(\underline{e})$ terá valor máximo. Somando-se estas equações, pode-se escrever que

$$\delta \underline{e}^T \mathbf{C}_\mathbf{x} \underline{e} - \lambda \delta \underline{e}^T \underline{e} = \mathbf{C}_\mathbf{x} (\delta \underline{e})^T \underline{e} - \lambda (\delta \underline{e})^T \underline{e} = 0 \tag{6.24}$$

onde o fator de escala λ foi introduzido para compatibilizar a unidade dimensional do vetor unitário $\underline{e}$ (adimensional por definição) com a unidade dimensional da matriz $\mathbf{C}_\mathbf{x}$ representativa da covariância do conjunto $\mathbf{X}$. Por exemplo, se os vetores do conjunto $\mathbf{X}$ representam uma grandeza cuja unidade dimensional é $\left[\frac{metro}{segundo}\right]$, λ terá unidade dimensional $\left[\frac{metro^2}{segundo^2}\right]$.

Reescrevendo a equação (6.24),

$$(\delta \underline{e})^T (\mathbf{C}_\mathbf{x} \underline{e} - \lambda \underline{e}) = 0 \tag{6.25}$$

Para que a condição posta na Equação (6.25) seja satisfeita, é necessário e suficiente que

$$\mathbf{C}_\mathbf{x} \underline{e} = \lambda \underline{e} \tag{6.26}$$

que é a equação que define os vetores $\underline{e}$ para os quais $f(\underline{e})$ tem valor máximo.

A Equação (6.26) é reconhecida como **a equação dos auto-vetores da matriz $\mathbf{C}_\mathbf{x}$** e apresenta um conjunto de soluções não-triviais para aqueles valores de λ que são denominados **os auto-valores de $\mathbf{C}_\mathbf{x}$**. Seja a *k*-ésima solução de (6.26) tal que

$$(\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I})\underline{e}_k = \underline{0} \tag{6.27}$$

Por serem mapeados no vetor nulo $\underline{0}$ através de $(\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I})$, os auto-vetores $\underline{e}_k$ pertencem ao espaço nulo da transformação linear (6.27), isto é,

$$\underline{e}_k \in \Re^{\mathbf{N}} \tag{6.28}$$

onde $\Re^{\mathbf{N}}$ é denominado espaço nulo da matriz $(\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I})$, denotado por $\mathbf{N}\ \{\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I}\}$. A dimensão $\Re^{\mathbf{N}}$ do espaço nulo de uma transformação linear define o número $\mathbf{N}$ de vetores linearmente independentes que levam a transformação ao vetor nulo $\underline{0}$, os quais, portanto, definem uma base geradora de $\Re^{\mathbf{N}}$ [8][9].

O *rank* da matriz $(\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I})$, denotado $\mathbf{R}\ \{\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I}\}$, é o espaço $\Re^{\mathbf{R}}$ gerado pelo número $\mathbf{R}$ de vetores-colunas de $(\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I})$ que são linearmente independentes. Um teorema de Teoria de Matrizes prova que a dimensão M da matriz quadrada $(\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I})_{[M \times M]}$ é igual à soma das dimensões de seu *rank* e espaço nulo [8][9], isto é

$$\mathbf{N} + \mathbf{R} = M = \dim\{\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I}\} \tag{6.29}$$

Por definição, os auto-vetores $\underline{e}_k$ formam uma base ortonormal no espaço por eles gerado. Isto significa que em (6.27) o conjunto de auto-vetores $\underline{e}_k$ define a própria base para o espaço nulo $\Re^{\mathbf{N}}$. Portanto, de (6.29), para que a dimensão $\mathbf{N}$ do espaço nulo seja igual à dimensão da matriz $(\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I})$, é necessário que o *rank* da matriz seja nulo ($\mathbf{R} = 0$). Assim $\mathbf{N} = M = \dim\{\mathbf{C}_\mathbf{x} - \lambda_k \mathbf{I}\}$, de forma que existirão M vetores $\underline{e}_k$ linearmente independentes. Mas, uma matriz que possui *rank* nulo é uma matriz singular. Isto é, se

$$\mathbf{R}\ \{\mathbf{C}_\mathbf{x} - \lambda_m \mathbf{I}\} = 0 \Rightarrow (\mathbf{C}_\mathbf{x} - \lambda_m \mathbf{I}) \text{ é singular, } m = 0, 1, \cdots, M-1 \tag{6.30}$$

e, se a matriz $(\mathbf{C}_\mathbf{x} - \lambda_m \mathbf{I})$ é singular, pode-se escrever que

$$\det\{\mathbf{C}_\mathbf{x} - \lambda_m \mathbf{I}\} = 0 \tag{6.31}$$

onde $\det\{.\}$ representa o determinante da matriz argumento.

A Equação (6.31) é a equação utilizada para a determinação do *m*-ésimo auto-valor λ_m da matriz $\mathbf{C_x}$ representativa da covariância do conjunto $\mathbf{X}$, $m = 0, 1, \cdots, M-1$. Como $\mathbf{C_x}$ é uma matriz simétrica, possui auto-valores reais e não-negativos [8][9].

Uma vez determinados os *M* auto-valores $\lambda_0, \lambda_1, \cdots, \lambda_{M-1}$ da matriz $\mathbf{C_x}$ através de (6.31), obtém-se os auto-vetores associados $\underline{e}_0, \underline{e}_1, \cdots, \underline{e}_{M-1}$ através de (6.27).

Assim, de (6.27) pode-se escrever

$$\mathbf{C_x}\, \underline{e}_m = \lambda_m\, \underline{e}_m, \; m = 0,1,\cdots,M-1 \tag{6.32}$$

Seja $\mathbf{\Lambda}$ uma matriz diagonal , $M \times M$, composta pelos auto-valores da matriz $\mathbf{C_x}$, de acordo com a Equação (6.32),

$$\Lambda = \mathrm{diag}\left(\lambda_0, \lambda_1, \cdots, \lambda_j, \cdots, \lambda_{M-1}\right) \tag{6.33}$$

onde os *M* auto-valores estão ordenados de forma decrescente, de modo que λ_0 seja o maior auto-valor.

Seja $\mathbf{E}$ uma matriz $M \times M$ cujas colunas são formadas pelos *M* auto-vetores $\underline{e}_0, \underline{e}_1, \cdots, \underline{e}_{M-1}$ dados pela Equação (6.32) associados aos auto-valores $\lambda_0, \lambda_1, \cdots, \lambda_{M-1}$.

$$\mathbf{E} = [\underline{e}_0 \;\; \underline{e}_1 \;\; \cdots \;\; \underline{e}_j \;\; \cdots \;\; \underline{e}_{M-1}] \tag{6.34}$$

De acordo com (6.33) e (6.34), a Equação (6.32) pode ser reescrita como

$$\mathbf{C_x E} = \mathbf{E\Lambda} \tag{6.35}$$

Como os auto-vetores de $\mathbf{C_x}$ (vetores-coluna de $\mathbf{E}$) são ortonormais, a matriz $\mathbf{E}$ é dita ortonormal. Portanto, os vetores-coluna de $\mathbf{E}$ satisfazem a condição de ortonormalidade,

$$\underline{e}_i^T\, \underline{e}_j = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \tag{6.36}$$

A Equação (6.36) pode também ser escrita sob a forma

$$\mathbf{E}^{\mathbf{T}}\mathbf{E} = \mathbf{I} \tag{6.37}$$

de onde vem que

$$\mathbf{E}^{\mathbf{T}} = \mathbf{E}^{-1} \tag{6.38}$$

A Equação (6.35) pode ser reescrita, sob a forma de uma Transformação de Similaridade [8][9]. Pré-multiplicando ambos os lados da Equação (6.35) por $\mathbf{E}^{\mathbf{T}}$,

$$\mathbf{E}^{\mathbf{T}}\mathbf{C}_{\mathbf{x}}\mathbf{E} = \mathbf{E}^{\mathbf{T}}\mathbf{E}\boldsymbol{\Lambda} \tag{6.39}$$

De (6.39) e (6.38),

$$\mathbf{E}^{\mathbf{T}}\mathbf{C}_{\mathbf{x}}\mathbf{E} = \mathbf{E}^{-1}\mathbf{C}_{\mathbf{x}}\mathbf{E} = \mathbf{E}^{-1}\mathbf{E}\boldsymbol{\Lambda} \tag{6.40}$$

ou

$$\mathbf{E}^{\mathbf{T}}\mathbf{C}_{\mathbf{x}}\mathbf{E} = \boldsymbol{\Lambda} \tag{6.41}$$

que, na forma expandida, resulta em

$$\underline{e}_j^T \mathbf{C}_{\mathbf{x}} \underline{e}_k = \begin{cases} \lambda_j, & k = j \\ 0, & k \neq j \end{cases} \tag{6.42}$$

Das Equações (6.8) e (6.9) tem-se

$$\sigma_a^2 = f\left(\underline{e}\right) = \underline{e}^T \mathbf{C}_{\mathbf{x}} \underline{e} \tag{6.43}$$

Comparando as Equações (6.42) e (6.43), verifica-se que

$$f\left(\underline{e}_m\right) = \lambda_m, \; m = 0,1,\cdots,M-1 \tag{6.44}$$

Portanto, os M auto-vetores $\underline{e}_m$ da matriz $\mathbf{C}_{\mathbf{x}}$ representam as M direções principais ao longo das quais a variância $\sigma_m^2 = f\left(\underline{e}_m\right)$ é máxima e dada por λ_m.

6.2.1 Interpretação Geométrica da KLT

A decomposição em sub-espaços gerada pela KLT consiste em obter o conjunto de M auto-vetores e auto-valores da matriz de covariância $\mathbf{C}$ de um conjunto $\mathbf{U}$ de média zero, composto por L vetores de dados $\underline{u}_i \in \Re^M, i = 0,1,\cdots,L-1$. A restrição de que o conjunto $\mathbf{U}$ apresente média vetorial $\underline{0} \in \Re^M$ é transparente a nível de procedimento, já que o vetor média pode ser restaurado ao final. O conjunto de M auto-vetores $\underline{e}_m \in \Re^M$ obtido pela KLT, $m = 0,1,\cdots,M-1$, define uma base de vetores ortonormais em $\Re^M$ e, portanto, define um conjunto de M eixos ortogonais Cartesianos. A coordenada de origem deste sistema Cartesiano é a coordenada de origem dos vetores da base ortonormal definida pelo conjunto de M auto-vetores $\underline{e}_m$. Como a média do conjunto $\mathbf{U}$ é assumida zero, a coordenada de origem do sistema Cartesiano é $\underline{0} \in \Re^M$. Ao projetar o conjunto original de vetores $\underline{u}_i \in \Re^M$ sobre o m-ésimo eixo Cartesiano, a variância do conjunto projetado – ou variância do sub-espaço – é dada pelo m-ésimo auto-valor λ_m. Ainda, a variância do conjunto $\mathbf{U}$ é igual à soma das variâncias das projeções de $\mathbf{U}$ [3][4][5][6]. Isto é, a média do quadrado da norma Euclidiana dos vetores de $\mathbf{U}$ é igual à soma dos M auto-valores λ_m, onde λ_m equivale à média do quadrado da norma Euclidiana da projeção dos vetores $\underline{u}_i \in \Re^M$ sobre o m-ésimo eixo.

A título de interpretação da KLT, se quiséssemos obter a KLT através de um processo manual e experimental, tomaríamos um vetor arbitrário de módulo unitário $\underline{e} \in \Re^M$ com origem em $\underline{0} \in \Re^M$, o qual definiria uma direção arbitrária em que o conjunto $\mathbf{U}$ de vetores $\underline{u}_i \in \Re^M$ de dados seria projetado. Projetaríamos, então, a totalidade do conjunto $\mathbf{U}$ sobre a direção dada por $\underline{e}$ e mediríamos a variância λ da projeção. Após tentarmos todas as direções possíveis no espaço $\Re^M$, haverá uma direção dada por $\underline{e}$ na qual é obtida a maior variância λ_0. O vetor $\underline{e}$ que define tal direção é igual ao auto-vetor $\underline{e}_0$ associado ao maior auto-valor, obtidos pela KLT, com valor do maior auto-valor dado por

λ_0. O processo é repetido novamente para a obtenção do segundo maior auto-valor λ_1, com a restrição de que a busca da direção de maior variância em $\Re^M$ seja feita em direções ortogonais à do auto-vetor $\underline{e}_0$ associado ao maior auto-valor λ_0, recém determinados. A busca da direção de maior variância em $\Re^M$ para obtenção do terceiro maior auto-valor λ_2 é feita com a restrição de que as direções testadas sejam ortogonais às direções dos dois auto-vetores $\underline{e}_0$ e $\underline{e}_1$ associados aos maiores auto-valores λ_0 e λ_1 previamente encontrados. E assim prosseguiríamos neste processo recursivo até que os *M* auto-valores e auto-vetores fossem determinados.

Assim, como os sub-espaços obtidos pela KLT estão alinhados com as direções ortogonais de maior variância possível no espaço original $\Re^M$, a KLT é considerada uma transformação ótima no sentido do erro médio quadrático MSE [3][4][5][6][11][7] para efeito de reconstrução do espaço original $\Re^M$ a partir de suas *M* projeções ou componentes principais. Ou seja, o conjunto de *M* sub-espaços representa de maneira ótima, no sentido do MSE, o conjunto $\mathbf{U}$ de *L* vetores $\underline{u}_i \in \Re^M$, $i = 0,1,\cdots,L-1$. Conforme já discutido, os sub-espaços obtidos pela KLT encontram-se alinhados com os eixos Cartesianos que definem as direções de maior variância possíveis no espaço original $\Re^M$. Em conseqüência, isto resulta em uma maior concentração de pontos definidos pelos vetores $\underline{u}_i \in \Re^M$ do espaço original nas vizinhanças dos eixos Cartesianos que definem cada sub-espaço.

Como o *m*-ésimo eixo Cartesiano define a *m*-ésima região em $\Re^M$ de maior variância λ_m, aqueles vetores cuja média do quadrado de suas normas Euclidianas é próxima ao valor λ_m (λ_m é a média do quadrado das normas Euclidianas das projeções destes vetores sobre o *m*-ésimo eixo) caracterizarão um sub-conjunto de vetores do espaço $\Re^M$ aproximadamente alinhados com o *m*-ésimo eixo.

Parte destes vetores estará aproximadamente congruente (*i.e.*, alinhados no mesmo sentido) com o *m*-ésimo semi-eixo positivo, e parte dos vetores estará aproximadamente congruente com o *m*-ésimo semi-eixo negativo. Mas, independentemente do sentido

positivo ou negativo, a média do quadrado da norma Euclidiana destes vetores será, em maior ou menor grau, próxima ao valor λ_m, grau que depende do quanto os vetores alinham-se com o *m*-ésimo eixo Cartesiano. Adicionalmente, a média dos vetores que são congruentes com o semi-eixo negativo tende a ser igual à média dos vetores que são congruentes com o semi-eixo positivo, já que $\mathbf{U}$ apresenta média vetorial $\underline{0} \in \Re^M$. Assim, deve-se esperar um certo equilíbrio entre os vetores alinhados com cada um dos dois semi-eixos.

Portanto, o *m*-ésimo sub-espaço identifica uma nuvem de pontos em $\Re^M$ nas vizinhanças do *m*-ésimo eixo Cartesiano, com coordenada de cada ponto definida pelo respectivo vetor $\underline{u}_i \in \Re^M$ do conjunto $\mathbf{U}$ nas vizinhanças do *m*-ésimo eixo. Como a média do quadrado da norma Euclidiana dos vetores associados a estes pontos tende em maior ou menor grau para λ_m, a norma – ou distância – Euclidiana média destes pontos à origem aproxima-se do valor $\sqrt{\lambda_m}$.

Exemplo 6.1:

Seja o conjunto $\mathbf{U}$ de $L = 6$ vetores $\underline{u}_i \in \Re^2$, $i = 0,1,\cdots,L-1$, de média zero definido por

$$\mathbf{U} = \left\{ \begin{bmatrix} 0.425 \\ -1.063 \end{bmatrix}, \begin{bmatrix} 0.595 \\ -0.943 \end{bmatrix}, \begin{bmatrix} 0.327 \\ -0.843 \end{bmatrix}, \begin{bmatrix} -0.497 \\ 0.820 \end{bmatrix}, \begin{bmatrix} -0.491 \\ 1.133 \end{bmatrix}, \begin{bmatrix} -0.359 \\ 0.897 \end{bmatrix} \right\}.$$

a) Plote os auto-vetores $\underline{e}_m$ da matriz $\mathbf{C}$ de covariância de $\mathbf{U}$ conjuntamente com os vetores de $\mathbf{U}$. Para facilitar a visualização, escalone os auto-vetores $\underline{e}_m$ pela raiz dos respectivos auto-valores λ_m, i.e., $\underline{\psi}_m = \underline{e}_m \cdot \sqrt{\lambda_m}$, $m = 0,1$.

b) Obtenha as projeções de $\mathbf{U}$ sobre os eixos Cartesianos definidos pelos auto-vetores de **C.**

Solução:

$$\mathbf{C} = \frac{1}{L}\sum_{i=0}^{L-1} \underline{x}_i \cdot \underline{x}_i^{\,T} = \frac{1}{6}\sum_{i=0}^{5} \underline{x}_i \cdot \underline{x}_i^{\,T} =$$

$$= \frac{1}{6}\left\{ \begin{bmatrix} 0.425 \\ -1.063 \end{bmatrix}\begin{bmatrix} 0.425 \\ -1.063 \end{bmatrix}^T + \begin{bmatrix} 0.595 \\ -0.943 \end{bmatrix}\begin{bmatrix} 0.595 \\ -0.943 \end{bmatrix}^T + \begin{bmatrix} 0.327 \\ -0.843 \end{bmatrix}\begin{bmatrix} 0.327 \\ -0.843 \end{bmatrix}^T + \right.$$

$$\left. + \begin{bmatrix} -0.497 \\ 0.820 \end{bmatrix}\begin{bmatrix} -0.497 \\ 0.820 \end{bmatrix}^T + \begin{bmatrix} -0.491 \\ 1.133 \end{bmatrix}\begin{bmatrix} -0.491 \\ 1.133 \end{bmatrix}^T + \begin{bmatrix} -0.359 \\ 0.897 \end{bmatrix}\begin{bmatrix} -0.359 \\ 0.897 \end{bmatrix}^T \right\} =$$

$$= \begin{bmatrix} 0.21 & -0.429 \\ -0.429 & 0.915 \end{bmatrix}$$

De (6.32), temos $\mathbf{C}\,\underline{e}_m = \lambda_m\,\underline{e}_m$ sendo $m = 0,1,\cdots,M-1$, com $M = 2$ ($M = 2$, porque $\underline{u}_i \in \Re^2$).

A solução de $\mathbf{C}\,\underline{e}_m = \lambda_m\,\underline{e}_m$ com $\mathbf{C} = \begin{bmatrix} 0.21 & -0.429 \\ -0.429 & 0.915 \end{bmatrix}$ consiste na determinação dos M auto-valores λ_m e dos M auto-vetores $\underline{e}_m$ que satisfazem esta equação.

Utilizando as funções eigenvals$(\cdot)$ e eigenvecs$(\cdot)$ do aplicativo MathCad da MathSoft Inc., obtemos :

Auto-valores de **C**:

$$\lambda_0 = 1.118 \text{ e } \lambda_1 = 7.035\times10^{-3}$$

Auto-vetores de **C** associados:

$$\underline{e}_0 = \begin{bmatrix} -0.427 \\ 0.904 \end{bmatrix} \text{ e } \underline{e}_1 = \begin{bmatrix} 0.904 \\ 0.427 \end{bmatrix}.$$

Nota: Qualquer outro aplicativo da área de matemática pode ser utilizado para determinar os auto-valores e auto-vetores de **C**, como, por exemplo, o MatLab da MathWorks Inc.

A Figura 6.1 mostra o conjunto $\mathbf{U}$, os $M = 2$ auto-vetores escalonados $\underline{\psi}_m = \underline{e}_m \cdot \sqrt{\lambda_m}$, $m = 0, 1, \cdots, M-1$ e os eixos Cartesianos por eles definidos.

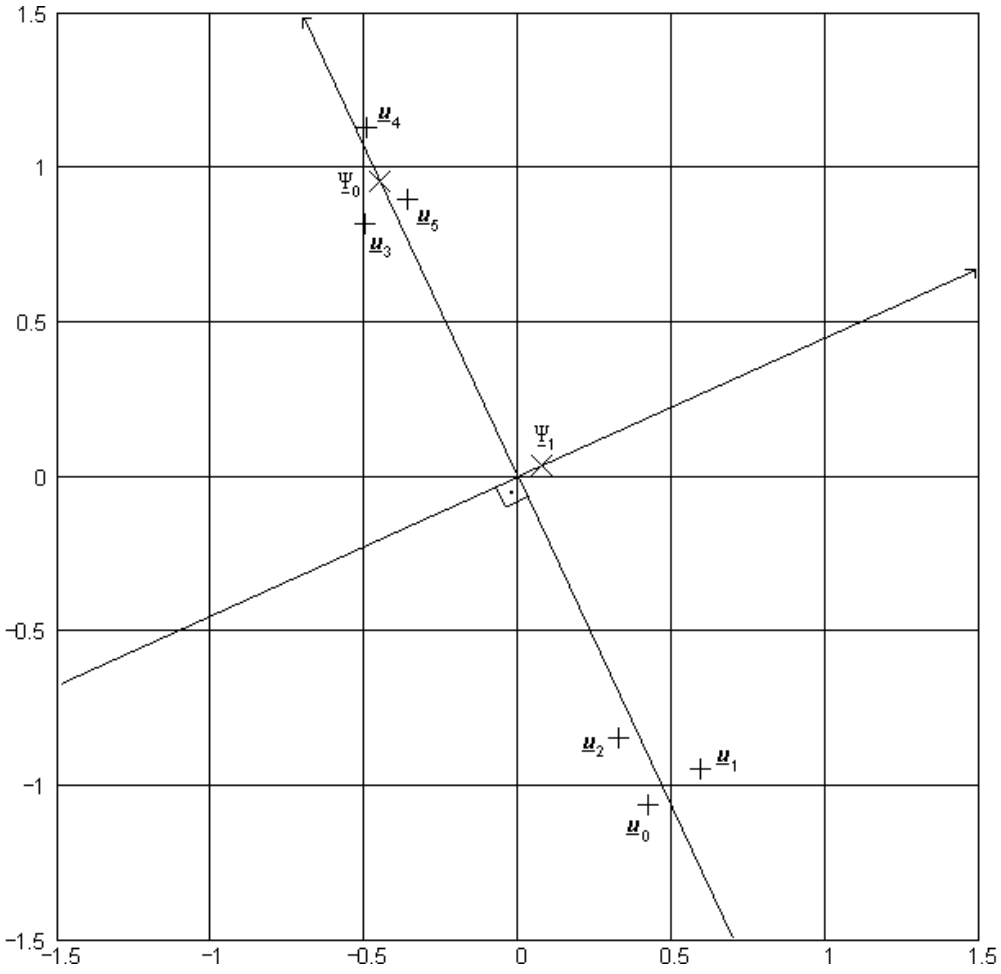

Figura 6.1: Conjunto $\mathbf{U}$, auto-vetores escalonados $\underline{\psi}_0$ e $\underline{\psi}_1$ e os eixos Cartesianos por eles definidos.

A Tabela 6.1 mostra a valor da projeção de cada vetor $\underline{u}_i \in \Re^2$ do conjunto $\mathbf{U}$ sobre os eixos Cartesianos definidos por $\underline{e}_0$ e $\underline{e}_1$.

i	0	1	2	3	4	5
$a_{0i} = \underline{u}_i^T \cdot \underline{e}_0$	-1.142	-1.107	-0.902	0.953	1.234	0.964
$a_{1i} = \underline{u}_i^T \cdot \underline{e}_1$	-0.07	0.135	-0.065	-0.099	0.04	0.059

Tabela 6.1: Projeções $\mathbf{a}_0$ e $\mathbf{a}_1$ de $\mathbf{U}$ sobre os eixos Cartesianos definidos por $\underline{e}_0$ e $\underline{e}_1$.

Observe que, como os auto-vetores $\underline{e}_0$ e $\underline{e}_1$ têm norma unitária e são adimensionais [8], o valor absoluto da projeção de cada $\underline{u}_i \in \Re^2$ sobre cada eixo define a norma Euclidiana da *i*-ésima projeção.

Note que a variância do conjunto $\mathbf{U}$ é dada pela soma dos auto-valores, i.e., $\frac{1}{L}\sum_{i=0}^{L-1}\|\underline{u}_i\|^2 = 1.125 = \lambda_0 + \lambda_1$, onde $\|\underline{u}\| = \sqrt{\sum_{m=0}^{M-1}(u_m)^2} = \sqrt{\underline{u}^T \cdot \underline{u}}$ é a norma Euclidiana do vetor $\underline{u} \in \Re^M$ [8].

Note ainda que as variâncias das projeções $\mathbf{a}_0$ e $\mathbf{a}_1$ de $\mathbf{U}$ equivalem aos respectivos auto-valores, i.e., $\frac{1}{L}\sum_{i=0}^{L-1}\left(\underline{u}_i^T \cdot \underline{e}_0\right)^2 = 1.118 = \lambda_0$ e $\frac{1}{L}\sum_{i=0}^{L-1}\left(\underline{u}_i^T \cdot \underline{e}_1\right)^2 = 7.035\times10^{-3} = \lambda_1$. Portanto, a variância do conjunto projetado, ou variância do sub-espaço, é dada pelo *m*-ésimo auto-valor λ_m.

Como observação adicional note que a distância Euclidiana média $\frac{1}{L}\sum_{i=0}^{L-1}\left|\underline{u}_i^T \cdot \underline{e}_m\right|$ dos vetores da *m*-ésima projeção à origem pode ser aproximada por $\sqrt{\frac{1}{L}\sum_{i=0}^{L-1}\left(\underline{u}_i^T \cdot \underline{e}_m\right)^2} = \sqrt{\lambda_m}$. No caso, $\frac{1}{L}\sum_{i=0}^{L-1}\left|\underline{u}_i^T \cdot \underline{e}_0\right| = 1.051$ para $\sqrt{\lambda_0} = 1.057$ e $\frac{1}{L}\sum_{i=0}^{L-1}\left|\underline{u}_i^T \cdot \underline{e}_1\right| = 0.078$ para $\sqrt{\lambda_1} = 0.083$.

6.2.2 O Processo de Compressão Possibilitado pela KLT

Na seção anterior vimos que os M eixos Cartesianos resultantes da KLT alinham-se com as direções de maior variância (=energia) de um conjunto $\mathbf{U}$ de vetores $\Re^M$ dimensionais, sendo $\underline{0} \in \Re^M$ o vetor média de $\mathbf{U}$. O Exemplo 6.1 ilustrou um caso para $M = 2$.

A Figura 6.2 mostra uma representação pictórica hipotética de uma nuvem de dados em $\Re^3$. Cada ponto da nuvem define a ponta de um vetor deste conjunto $\mathbf{U}$ em $\Re^3$. A figura mostra a maneira como a base ortonormal formada pelos auto-vetores alinha-se com as direções de maior variância (energia) de $\mathbf{U}$. Assim, no caso genérico de um conjunto $\mathbf{U}$ em $\Re^M$, podemos imaginar a KLT "girando" uma base de M vetores ortonormais em $\Re^M$ em todas as possíveis direções até que os vetores alinhem-se com as direções de maior variância possível em $\mathbf{U}$. Nesta situação os vetores da base ortonormal são os auto-vetores da matriz de covariância de $\mathbf{U}$.

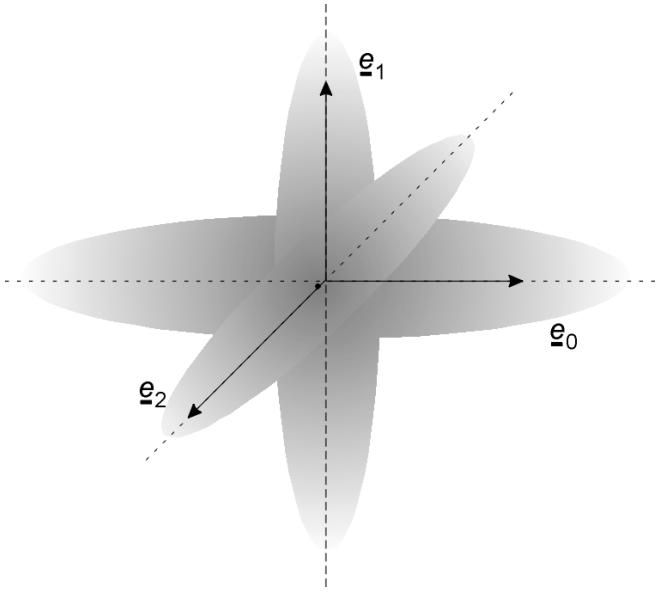

Figura 6.2: Representação pictórica hipotética de uma "nuvem" de dados em $\Re^3$ e a maneira como a base ortonormal formada pelos auto-vetores $\underline{e}_0$, $\underline{e}_1$ e $\underline{e}_2$ alinha-se com as direções de maior variância.

Vimos também que a energia contida em uma determinada direção (=variância da projeção de $\mathbf{U}$ na direção) é dada pelo auto-valor associado ao auto-vetor que define a direção. Portanto, podemos desprezar aquelas direções (=sub-espaços) definidas por auto-vetores cujo auto-valor associado é muito menor do que os demais auto-valores. Isto é possível porque a projeção de $\mathbf{U}$ sobre tais direções é insignificante se comparada com as projeções cujo auto-valor associado não é comparativamente pequeno.

Ao desprezar sub-espaços de menor energia estamos realizando **uma compressão com perdas** do conjunto $\mathbf{U}$. No entanto estas perdas são mínimas pois as componentes (=sub-espaços) descartadas tem pouca influência na formação de $\mathbf{U}$ porque os auto-valores a elas associados supostamente têm valor muito menor do que os demais. Neste sentido ocorre a **redução dimensional** de $\mathbf{U}$, porque o conjunto era originalmente representado em $\Re^M$ e, ao descartar sub-espaços não significativos, o conjunto passa a ser representado com boa aproximação em uma dimensão menor que *M*. Este é o motivo de a KLT ser considerada uma transformação ótima sob o ponto de vista do MSE.

Exemplo 6.2:

Seja o conjunto $\mathbf{U}$ do Exemplo 6.1.

a) Execute uma compressão com perdas de $\mathbf{U}$ utilizando a KLT.

b) Calcule a compressão obtida.

c) Calcule o MSE da compressão obtida.

d) Calcule a Relação Sinal – Ruído de Pico em dB, denotada PSNR (PSNR – *Peak Signal To Noise Ratio*), resultante do processo de compressão. Isto é, calcule o quadrado da componente de maior valor absoluto dentre todas as componentes dos vetores de $\mathbf{U}$ e normalize pelo MSE obtido em c).

Solução:

No Exemplo 6.1 foram determinadas as projeções de $\mathbf{U}$ sobre a base ortonormal formada pelos auto-vetores $\underline{e}_0$ e $\underline{e}_1$, conforme mostra a Tabela 6.1.

Observe na Tabela 6.1 que as projeções sobre a direção $\underline{e}_1$ são muito menores do que as projeções sobre a direção $\underline{e}_0$. Esta característica das projeções está de acordo com o fato de que $\lambda_1 = 7.035 \times 10^{-3}$ é muito menor do que $\lambda_0 = 1.118$. Sendo assim, podemos descartar as projeções ou componentes de $\mathbf{U}$ na direção $\underline{e}_1$ (a direção com menor auto-valor associado) e considerar as projeções de $\mathbf{U}$ na direção $\underline{e}_0$ (a direção com maior auto-valor associado) como uma boa aproximação de $\mathbf{U}$.

Portanto a compressão com perdas consiste em aproximar $\mathbf{U}$ através do auto-vetor $\underline{e}_0$ e do conjunto de projeções $a_{0i} = \underline{u}_i^{\,T} \cdot \underline{e}_0$ na direção por ele definida. A aproximação $\widetilde{\mathbf{U}}$ do conjunto de vetores originais $\underline{u}_i \in \mathbf{U}$, é obtida através de $\underline{\widetilde{u}}_i = a_{0i}\underline{e}_0$, $\underline{\widetilde{u}}_i \in \widetilde{\mathbf{U}}$, isto é,

i	0	1	2	3	4	5
$a_{0i} = \underline{u}_i^{\,T} \cdot \underline{e}_0$, $\underline{e}_0 = \begin{bmatrix} -0.427 \\ 0.904 \end{bmatrix}$	-1.142	-1.107	-0.902	0.953	1.234	0.964
$\underline{\widetilde{u}}_i = a_{0i}\underline{e}_0$	$\begin{bmatrix} 0.488 \\ -1.032 \end{bmatrix}$	$\begin{bmatrix} 0.473 \\ -1.001 \end{bmatrix}$	$\begin{bmatrix} 0.385 \\ -0.815 \end{bmatrix}$	$\begin{bmatrix} -0.407 \\ 0.862 \end{bmatrix}$	$\begin{bmatrix} -0.527 \\ 1.116 \end{bmatrix}$	$\begin{bmatrix} -0.412 \\ 0.871 \end{bmatrix}$

Tabela 6.2: Aproximação $\widetilde{\mathbf{U}}$ do conjunto original $\mathbf{U}$ obtida através de $\underline{\widetilde{u}}_i = a_{0i}\underline{e}_0$, onde $\underline{\widetilde{u}}_i \in \widetilde{\mathbf{U}}$.

Portanto, da Tabela 6.2, o conjunto $\widetilde{\mathbf{U}}$ resultante é

$$\widetilde{\mathbf{U}} = \left\{ \begin{bmatrix} 0.488 \\ -1.032 \end{bmatrix}, \begin{bmatrix} 0.473 \\ -1.001 \end{bmatrix}, \begin{bmatrix} 0.385 \\ -0.815 \end{bmatrix}, \begin{bmatrix} -0.407 \\ 0.862 \end{bmatrix}, \begin{bmatrix} -0.527 \\ 1.116 \end{bmatrix}, \begin{bmatrix} -0.412 \\ 0.871 \end{bmatrix} \right\}$$

Observe que o conjunto original $\mathbf{U} = \left\{ \begin{bmatrix} 0.425 \\ -1.063 \end{bmatrix}, \begin{bmatrix} 0.595 \\ -0.943 \end{bmatrix}, \begin{bmatrix} 0.327 \\ -0.843 \end{bmatrix}, \begin{bmatrix} -0.497 \\ 0.820 \end{bmatrix}, \begin{bmatrix} -0.491 \\ 1.133 \end{bmatrix}, \begin{bmatrix} -0.359 \\ 0.897 \end{bmatrix} \right\}$ de $L = 6$ vetores $\underline{u}_i \in \Re^2$, $i = 0,1,\cdots,L-1$ necessita de 12 números em ponto-flutuante para ser representado.

Por outro lado, o conjunto aproximado $\widetilde{\mathbf{U}}$ foi gerado a partir do auto-vetor $\underline{e}_0 = \begin{bmatrix} -0.427 \\ 0.904 \end{bmatrix}$ e de um conjunto de 6 projeções $\mathbf{a}_0 = \{-1.142, -1.107, -0.902, 0.953, 1.234, 0.964\}$.

Portanto, o conjunto $\widetilde{\mathbf{U}}$ necessita de apenas 8 números em ponto-flutuante para ser representado.

Define-se como fator de compressão ρ o quociente

$$\rho = \frac{\text{Total de unidades de armazenamento necessário para representar } \widetilde{\mathbf{U}}}{\text{Total de unidades de armazenamento necessário para representar } \mathbf{U}} \tag{6.45}$$

Portanto o fator de compressão obtido é $\rho = 8/12 = 0.67$.

O MSE da aproximação $\widetilde{\mathbf{U}}$ pode ser obtido através de

$$\text{MSE} = \frac{1}{L}\sum_{i=0}^{L-1} (\underline{\widetilde{u}}_i - \underline{u}_i)^T (\underline{\widetilde{u}}_i - \underline{u}_i) \tag{6.46}$$

onde $\underline{u}_i \in \mathbf{U}$ e $\underline{\widetilde{u}}_i \in \widetilde{\mathbf{U}}$, $i = 0,1,\cdots,L-1$.

De (6.46) obtemos $\text{MSE} = 7.025 \times 10^{-3}$.

A PSNR é definida como $$\text{PSNR} = 10\log\left(\frac{(\max comp\{\mathbf{U}\})^2}{\text{MSE}} \right) \tag{6.47}$$

sendo $\max comp\{\mathbf{U}\}$ a componente de maior valor absoluto dentre todas as componentes dos vetores de $\mathbf{U}$.

De (6.47) obtemos $$\text{PSNR} = 10\log\left(\frac{(1.133)^2}{7.025 \times 10^{-3}} \right) = 22.6 \text{ dB} .$$

Exemplo 6.3:

Obtenha o conjunto de vetores $\underline{\tilde{u}}_i \in \tilde{\mathbf{U}}$ a partir de **todos** os sub-espaços do conjunto de vetores $\underline{u}_i \in \mathbf{U}$ do Exemplo 6.2 (i.e., não execute nenhuma compressão, apenas reconstrua o conjunto original a partir de ambos sub-espaços $\mathbf{a}_0$ e $\mathbf{a}_1$). Determine a PSNR do conjunto reconstruído $\tilde{\mathbf{U}}$.

Solução:

A partir da Tabela 6.1 obtemos a Tabela 6.3 abaixo.

i	0	1	2	3	4	5
$a_{0i} = \underline{u}_i^T \cdot \underline{e}_0,\ \underline{e}_0 = \begin{bmatrix} -0.427 \\ 0.904 \end{bmatrix}$	-1.142	-1.107	-0.902	0.953	1.234	0.964
$a_{1i} = \underline{u}_i^T \cdot \underline{e}_1,\quad \underline{e}_1 = \begin{bmatrix} 0.904 \\ 0.427 \end{bmatrix}$	-0.07	0.135	-0.065	-0.099	0.04	0.059
$\underline{\tilde{u}}_i = a_{0i}\underline{e}_0 + a_{1i}\underline{e}_1$	$\begin{bmatrix} 0.424 \\ -1.062 \end{bmatrix}$	$\begin{bmatrix} 0.595 \\ -0.943 \end{bmatrix}$	$\begin{bmatrix} 0.326 \\ -0.843 \end{bmatrix}$	$\begin{bmatrix} -0.496 \\ 0.819 \end{bmatrix}$	$\begin{bmatrix} -0.491 \\ 1.133 \end{bmatrix}$	$\begin{bmatrix} -0.358 \\ 0.897 \end{bmatrix}$

Tabela 6.3: Reconstrução $\tilde{\mathbf{U}}$ do conjunto original $\mathbf{U}$ obtida através de $\underline{\tilde{u}}_i = a_{0i}\underline{e}_0 + a_{1i}\underline{e}_1$, onde $\underline{\tilde{u}}_i \in \tilde{\mathbf{U}}$. Como nenhum sub-espaço é descartado, nenhuma compressão é obtida e o conjunto $\tilde{\mathbf{U}}$ é considerado uma reconstrução do conjunto original $\mathbf{U}$ a partir dos sub-espaços $\mathbf{a}_0$ e $\mathbf{a}_1$.

Da Tabela 6.3 obtemos, $\tilde{\mathbf{U}} = \left\{ \begin{bmatrix} 0.424 \\ -1.062 \end{bmatrix}, \begin{bmatrix} 0.595 \\ -0.943 \end{bmatrix}, \begin{bmatrix} 0.326 \\ -0.843 \end{bmatrix}, \begin{bmatrix} -0.496 \\ 0.819 \end{bmatrix}, \begin{bmatrix} -0.491 \\ 1.133 \end{bmatrix}, \begin{bmatrix} -0.358 \\ 0.897 \end{bmatrix} \right\}$.

De (6.46) obtemos $\text{MSE} = 5.3 \times 10^{-7}$.

E, de (6.47) obtemos $\text{PSNR} = 10 \log\left(\frac{(1.133)^2}{5.3 \times 10^{-7}} \right) = 63.8\,\text{dB}$.

Nota: Observe que o MSE não é zero (e portanto a PSNR não é infinita) unicamente devido à precisão (3 casas após a vírgula) nas operações de ponto-flutuante efetuadas. Se estivéssemos trabalhando com uma precisão suficiente o MSE seria zero porque neste exemplo, ao contrário do Exemplo 6.2, nenhuma informação é descartada. Aqui o conjunto original $\mathbf{U}$ é apenas reconstruído a partir de todos os seus sub-espaços sem ocorrer qualquer compressão.

6.2.3 Compressão de Imagens através da KLT

Nesta seção estudaremos como efetuar a compressão de imagens através da KLT [3][4][5][6]. Por simplicidade trabalharemos com imagens em tons de cinza (*grayscale*). Uma imagem em *grayscale* tem seus valores de pixel variando entre 0 e 255. O valor 0 representa preto e o valor 255 representa branco. Todos os demais valores intermediários representam tons de cinza. A tonalidade cinza de um pixel de uma imagem *grayscale* clareia à medida que o valor do pixel aumenta. A Figura 6.3 mostra a imagem *grayscale* "Lenna" de tamanho 128×128 pixels.

Figura 6.3: Imagem *grayscale* "Lenna" de tamanho 128×128 pixels.

Para comprimir a imagem da Figura 6.3 através da KLT, adotaremos o seguinte procedimento:

1) Subdivide-se a imagem em $L = 256$ blocos de 8×8 pixels e registra-se a posição de cada bloco na imagem (p/ certas imagens, melhor resultado é obtido c/ blocos 16×16) .

2) O *i*-ésimo bloco $\mathbf{B}_i$ de 8×8 pixels é convertido em um vetor $\Re^M$ dimensional $\underline{u}_i \in \mathbf{U}$, sendo $M = 64$ (devido ao bloco possuir 8×8 pixels), $i = 0,1,\cdots,L-1$. A conversão bloco-vetor $\mathbf{B}_i \rightarrow \underline{u}_i$ é efetuada lendo-se as 8 linhas de $\mathbf{B}_i$ da esquerda para a direita sendo a linha ao alto a primeira a ser lida e transferindo o valor de cada pixel lido nesta ordem para as respectivas posições em seqüência no vetor $\underline{u}_i$.

3) Calcula-se o vetor média do conjunto de vetores $\mathbf{U}$ obtido em 2) e subtrai-se este vetor média de todos os $L = 256$ vetores $\underline{u}_i \in \mathbf{U}$.

4) Determina-se os sub-espaços de **U** conforme já discutido na Seção 6.3.

5) Descarta-se aqueles sub-espaços associados aos **menores** auto-valores .

6) Obtém-se o conjunto de vetores $\underline{\tilde{u}}_i \in \tilde{\mathbf{U}}$ sendo $\underline{\tilde{u}}_i = a_{0i}\underline{e}_0 + a_{1i}\underline{e}_1 + \cdots + a_{N-1i}\underline{e}_{N-1}$, onde $\{\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{N-1}\}$ são os sub-espaços definidos pelos auto-vetores $\{\underline{e}_0, \underline{e}_1, \cdots, \underline{e}_{N-1}\}$ cujos auto-valores associados são os *N* **maiores** auto-valores (os *N* sub-espaços de maior energia). *N* é obtido experimentalmente e define o compromisso entre ρ e PSNR.

7) Soma-se ao conjunto de *L* vetores $\underline{\tilde{u}}_i \in \tilde{\mathbf{U}}$ o vetor média originalmente subtraído de **U**.

8) Obtém-se o conjunto de blocos $\tilde{\mathbf{B}}_i$ através da conversão bloco-vetor inversa $\underline{\tilde{u}}_i \rightarrow \tilde{\mathbf{B}}_i$, $i = 0,1,\cdots,L-1$, e remonta-se a imagem comprimida a partir do registro da posição de cada bloco.

A Figura 6.4 mostra a amplitude relativa em ordem decrescente dos 32 primeiros maiores auto-valores do conjunto **U** formado a partir da Figura 6.3 de acordo com o procedimento acima descrito.

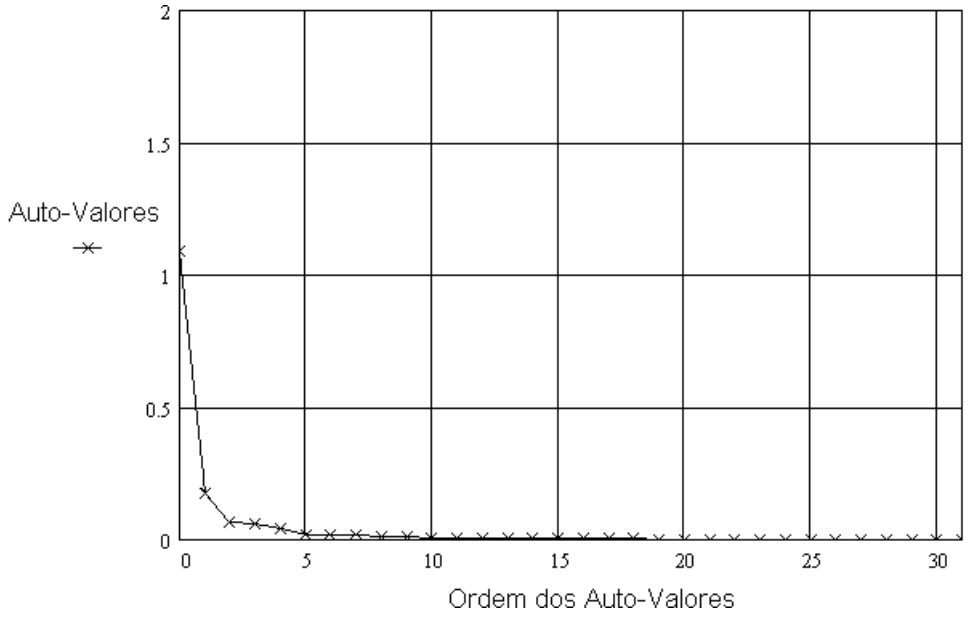

Figura 6.4: Amplitude relativa em ordem decrescente dos 32 primeiros maiores auto-valores do conjunto **U** formado a partir da Figura 6.3. Observe que o número total de auto-valores é 64, já que cada vetor de **U** é de dimensão $\Re^{64}$. No entanto, visto que aproximadamente a partir do vigésimo maior auto-valor a amplitude relativa torna-se pequena, escolheu-se representar apenas as 32 primeiras maiores amplitudes.

A Figura 6.5 mostra a imagem comprimida resultante formada a partir dos 32 primeiros sub-espaços de maior variância e a Figura 6.6 mostra a imagem comprimida resultante formada a partir dos 16 primeiros sub-espaços de maior variância.

Figura 6.5:

Imagem da Figura 6.3 comprimida pela KLT utilizando os $N = 32$ primeiros sub-espaços de maior energia.

O coeficiente de compressão resultante é $\rho = 0.629$ (Equação (6.45)).

A fidelidade da imagem comprimida com relação à original é dada por $\text{PSNR} = 37.3\,\text{dB}$ (Equação (6.47)).

Figura 6.6:

Imagem da Figura 6.3 comprimida pela KLT utilizando os $N = 16$ primeiros sub-espaços de maior energia.

O coeficiente de compressão resultante é $\rho = 0.316$ (Equação (6.45)).

A fidelidade da imagem comprimida com relação à original é dada por $\text{PSNR} = 31.4\,\text{dB}$ (Equação (6.47)).

6.3 O Algoritmo Hebbiano Generalizado para PCA ou DSE de um Espaço Vetorial

Conforme estudamos na Seção 6.2 deste capítulo, para proceder à Análise dos Componentes Principais ou Decomposição de um Espaço Vetorial em Sub-Espaços é necessário computar a matriz de covariância do conjunto de dados de entrada para, então, aplicar um procedimento numérico que extraia os auto-valores e os correspondentes auto-vetores desta matriz. Os auto-vetores correspondentes aos auto-valores mais significativos são usados para extrair os componentes principais dos dados.

No entanto, para grandes conjuntos de dados, as dimensões da matriz covariância crescem significativamente, tornando este método inadequado para a extração dos componentes principais. Além disto, todos os auto-valores e auto-vetores precisam ser computados, mesmo que somente poucos auto-vetores (aqueles correspondentes aos maiores auto-valores) venham a ser utilizados no processo.

A solução eficiente para este problema deve encontrar os principais auto-vetores sem a necessidade de determinar a matriz covariância. Tal solução é alcançada ao abordar o problema utilizando Redes Neurais Artificiais [15].

A técnica que utiliza RNAs para efetuar a Análise dos Componentes Principais sobre um espaço vetorial de interesse a ser abordada neste estudo é denomidada "Algoritmo Hebbiano Generalizado" e foi proposta por Sanger em 1989 [7]. Este algoritmo combina a ortonormalização de Gram-Schmidt [14] ao modelo de um único neurônio linear introduzido por Oja em 1982 [7].

O Algoritmo Hebbiano Generalizado (denominado GHA na literatura) é um dos algoritmos mais usados em aplicações práticas, porque:

- extrai os componentes principais individualmente, um após o outro, em ordem decrescente de variância e

- porque possibilita que, após treinada a rede, os resultados obtidos possam ser aplicados a um outro conjunto de dados de estatística "semelhante" (ou seja, que resulte em uma matriz de covariância "semelhante") [7].

O desenvolvimento apresentado neste capítulo segue a exposição de Haykin em [7], Hassoun em [11] e Hertz, Krogh e Palmer em [15].

6.3.1 O Aprendizado Hebbiano

Na Seção 6.1 estudamos os princípios que regem o aprendizado auto-organizado, inspirados no postulado de aprendizado de Hebb.

O Aprendizado Hebbiano é uma classe de aprendizado não-supervisionado ou auto-organizado, em que os parâmetros livres da rede são adaptados pela exposição da RNA ao ambiente em que está contida, até que uma pré-determinada condição seja atingida, através:

- da repetição dos padrões de entrada por um número suficiente de vezes e
- da aplicação de um conjunto de regras de aprendizado muito específicas, adequadas à solução do problema.

Uma RNA treinada sob aprendizado não-supervisionado descobre padrões significativos ou característicos dos dados de entrada sem o auxílio de um tutor externo. Ou seja, não há realimentação do ambiente para auxiliar a rede a determinar se a saída está ou não correta. A RNA deve descobrir por si padrões, características, regularidades, correlações ou categorias nos dados de entrada. As unidades e conexões devem, portanto, apresentar algum grau de auto-organização.

Em alguns casos, os vetores de entrada podem ser mapeados em um conjunto dimensionalmente menor de padrões, tal que este conjunto de padrões preserve as relações existentes no conjunto original.

O conhecimento que vai sendo "cristalizado" na RNA é obtido da observação repetida de parâmetros estatísticos (média, variância e matriz correlação) dos dados de entrada. Neste sentido, redundância de padrões provê conhecimento.

Na rede de uma única unidade, mostrada na Figura (6.7), os sinais pré-sinápticos e pós-sinápticos, respectivamente $\underline{x}$ e y, podem ser equacionados por

$$\underline{w}_0(n+1) = \underline{w}_0(n) + \eta\, y_0(n)\underline{x}(n) \tag{6.48}$$

$$y_0(n) = \underline{x}(n)^T\, \underline{w}_0(n) = \underline{w}_0(n)^T\, \underline{x}(n) \tag{6.49}$$

onde η é uma constante positiva que determina a razão de aprendizado, $\underline{x}(n)$ é o vetor de entrada da rede e $y_0(n)$ é a saída da rede.

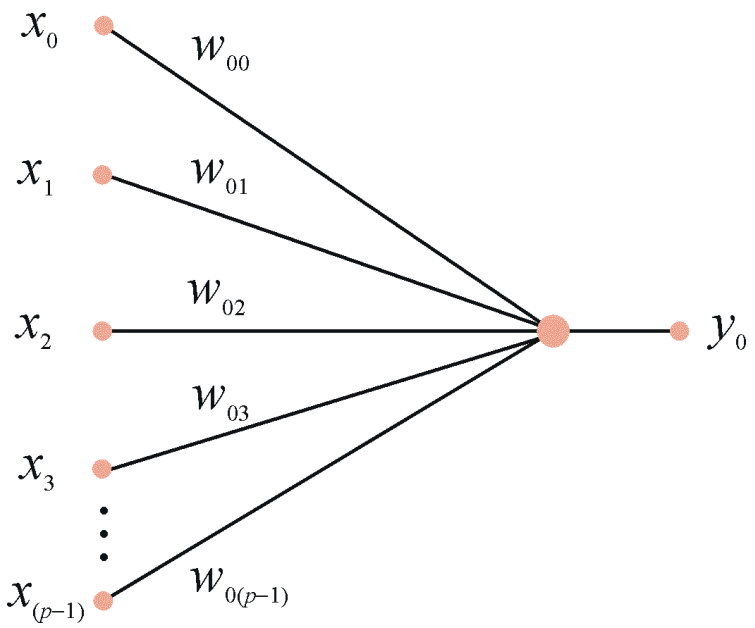

Figura 6.7: RNA de uma única unidade, a ser treinada pelo Aprendizado Hebbiano.

Observe que a Equação (6.48) expressa a mudança no peso sináptico, que ocorre na proporção da correlação entre os sinais pré e pós-sinápticos, enquanto que a Equação (6.49) expressa a saída da RNA.

O conjunto de vetores $\underline{x}$ de entrada possui distribuição de probabilidade arbitrária. A cada tempo *n* um vetor $\underline{x}$ é apresentado à rede, escolhido aleatoriamente do conjunto.

Conforme vimos na Seção 6.1, segundo os princípios do Aprendizado Hebbiano:

(I) quanto mais provável for uma particular entrada $\underline{x}$, maior será a correlação da saída *y* com esta entrada. Assim, quanto mais provável for $\underline{x}$, maior será a saída *y*;

(II) quanto maior for a saída *y*, mais a variação do peso sináptico que a encorajou aumentará.

As afirmações (I) e (II) nada mais são do que o 1º princípio da auto-organização de von der Malsburg: "Modificações em pesos sinápticos tendem a se auto-amplificar". Ou seja:

- a correlação entre as mudanças nos pesos sinápticos e as mudanças nos padrões de atividade deve ser positiva para que se obtenha auto-organização da rede;
- o processo de auto-amplificação é restrito pelo requerimento de que modificações em pesos sinápticos devem ser baseadas em sinais sinais pré-sinápticos e pós-sinápticos (disponíveis localmente);
- uma forte sinapse conduz à coincidência de sinais pré e pós-sinápticos e, por outro lado, a sinapse é aumentada em força, por tal coincidência.

Ou, ainda, segundo o postulado de Hebb: "Se dois neurônios em cada lado de uma sinapse são ativados simultaneamente, então a força daquela sinapse é seletivamente aumentada."

O "*loop*" configurado por (I) e (II), no entanto, faria com que os pesos sinápticos permanecessem crescendo sem limite, impedindo a continuação do processo de aprendizado e levando o modelo à instabilidade.

Na realidade, o que ocorre é que, pelo aumento demasiado do peso sináptico, a rede é levada à saturação precoce e é incapaz de aprender os padrões apresentados.

Sabemos, através do 2º princípio da auto-organização de von der Malsburg ("Limitação de recursos conduz à competição entre as sinapses e, conseqüentemente, à seleção das sinapses que crescem de forma mais vigorosa (portanto, as mais adequadas) às custas de outras"), que precisa ser estabelecida alguma forma de competição por recursos "limitados", para que o sistema possa ser estabilizado.

A solução consiste em compensar o aumento na força de alguma sinapse na rede, pelo decréscimo em outras, para que apenas as sinapses que obtêm sucesso possam crescer, enquanto outras enfraqueçam e, eventualmente, desapareçam.

O problema de estabilidade do modelo pode ser investigado conforme segue:

Suponhamos que, após um tempo de aprendizado suficientemente grande, os pesos sinápticos atinjam um ponto de equilíbrio.

Tomando-se o valor esperado da mudança dos pesos sinápticos (de acordo com a Equação (6.48)) para esta suposta condição de equilíbrio,

$$E\{\Delta \underline{w}_0(n)\} = E\{\eta\, y_0(n)\underline{x}(n)\} \tag{6.50}$$

Substituindo a Equação (6.49) na Equação (6.50),

$$E\{\Delta \underline{w}_0(n)\} = E\{\eta\, \underline{x}(n)^T\, \underline{w}_0(n)\, \underline{x}(n)\} \tag{6.51}$$

Considerando que o parâmetro razão de aprendizado η é uma entidade determinística e que os vetores de dados $\underline{x}$ e os vetores pesos sinápticos $\underline{w}$ são estatisticamente independentes,

$$E\{\Delta \underline{w}_0(n)\} = \eta\, E\{\underline{x}(n)\underline{x}^T(n)\}E\{\underline{w}_0(n)\} \tag{6.52}$$

E que, no equilíbrio,

$$E\{\Delta \underline{w}_0(n)\} = 0\,, \tag{6.53}$$

$$E\{\underline{x}(n)\underline{x}(n)^T\} = C_x \quad \text{e} \tag{6.54}$$

$$E\{\underline{w}_0(n)\} = \underline{e}_0\,, \tag{6.55}$$

pois, para uma condição de equilíbrio, o número de iterações *n* seria suficientemente grande para que o vetor de pesos sinápticos ($\underline{w}_0(n)$) tendesse a um valor constante (valor de equilíbrio). Ou seja, quando $n \to \infty$, $\underline{w}_0(n) \to \underline{e}_0$, onde $\underline{e}_0$ é um valor constante.

Nestas condições, a Equação (6.52) se torna

$$C_x \, \underline{e}_0 = 0 \tag{6.56}$$

Assim, no suposto ponto de equilíbrio, a solução da Equação (6.56) – conforme a teoria de auto-valores e auto-vetores – é um auto-vetor $\underline{e}$ de C_x cujo correspondente auto-valor é zero.

Algumas considerações podem ser traçadas a partir da Equação (6.56):

- a matriz de correlação C_x é uma matriz Hermitiana, que é sempre não-nula, pois denota a correlação do sinal com ele próprio (mesmo um sinal de ruído tem auto-correlação não nula);
- os auto-valores de uma matriz Hermitiana são reais e poderão ser positivos.

Qualquer flutuação de um peso sináptico que tenha uma componente ao longo da direção de um auto-vetor associado a um auto-valor positivo crescerá exponencialmente.

Assim, a direção relativa ao maior auto-valor da matriz covariância poderá se tornar dominante, de forma que o peso sináptico $\underline{w}$ irá se aproximar gradualmente de um auto-vetor $\underline{e}$ correspondente a um auto-valor máximo, cuja norma crescerá demasiadamente.

O auto-vetor $\underline{e}$, portanto, nunca se estabilizará e haverá somente soluções instáveis para o algoritmo de Aprendizado Hebbiano.

O problema da instabilidade do algoritmo, acima exposto, pode ser equacionado da seguinte forma:

Ao tomar a derivada da variação do peso sináptico $\Delta \underline{w}_0$ em relação ao próprio peso sináptico $\underline{w}_0$, poderemos investigar de que forma a variação do peso sináptico se comporta ao longo do tempo *n*. Assim,

$$\frac{d}{d(\underline{w}_0(n))}(\Delta \underline{w}_0(n)) = \frac{d}{d(\underline{w}_0(n))}(\eta\, y_0(n)\underline{x}(n)) \tag{6.57}$$

Considerando as formas expandidas na Equação (6.57),

$$\frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) = \frac{d}{d(w_{0i}(n))}(\eta\, x_i(n)\sum_i x_i(n) w_{0i}(n)) \tag{6.58}$$

A derivada do somatório da expressão (6.58) em relação a $w_{0i}(n)$ só existirá para o termo $x_i(n)w_{0i}(n)$, quando será igual a $x_i(n)$. A Equação (6.58) se tornará, portanto,

$$\frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) = \eta\, {x_i}^2(n) \tag{6.59}$$

De acordo com a Equação (6.59), a razão de variação da variação do peso sináptico é positiva para todo *n* e todo p . Isto implica que, ao longo de todo o processo de atualização dos pesos sinápticos, a variação será crescente, o que torna o auto-vetor $\underline{e}$ uma solução instável para o algoritmo.

Solução proposta por Oja para o problema da instabilidade do algoritmo:

Para evitar a divergência do algoritmo de aprendizado Hebbiano é preciso restringir o crescimento do vetor $\underline{w}$. Oja propôs em 1982 [7] como uma alternativa para a

estabilização do Aprendizado Hebbiano, uma modificação no algoritmo que permite ao vetor peso sináptico $\underline{w}$ atingir norma unitária, além de corresponder ao maior auto-valor de C_x.

A regra de Oja corresponde a adicionar um decaimento ao peso sináptico, proporcional a ${y_0}^2(n)$.

A Equação (6.60) expressa (novamente sob a forma vetorial) a regra de Aprendizado Hebbiano, após deflacionada de um fator $\eta\ {y_0}^2(n)\underline{w}_0(n)$.

$$\underline{w}_0(n+1) = \underline{w}_0(n) + \eta\ \ y_0(n)\underline{x}(n) - \eta\ {y_0}^2(n)\underline{w}_0(n) \tag{6.60}$$

A regra deflacionada para o Aprendizado Hebbiano apresentada na Equação (6.60) pode ser reescrita como,

$$\Delta\underline{w}_0(n) = \eta\ \ y_0(n)\{\underline{x}(n) - y_0(n)\underline{w}_0(n)\} \tag{6.61}$$

Para provar que $\|\underline{w}_0\| = 1$ e que o peso sináptico $\underline{w}_0$ está "depositado" na direção do maior auto-vetor de C_x, considere-se a Equação (6.61). Tomando o valor esperado da mudança dos pesos sinápticos,

$$E\{\Delta\underline{w}_0(n)\} = E\{\eta\ y_0(n)[\underline{x}(n) - y_0(n)\underline{w}_0(n)]\} \tag{6.62}$$

Reescrevendo a Equação (6.62),

$$E\{\Delta\underline{w}_0(n)\} = E\{\eta[\ \underline{x}(n)y_0(n) - y_0(n)y_0(n)\underline{w}_0(n)]\} \tag{6.63}$$

Considerando que η é uma entidade determinística e levando a Equação (6.49), $y_0(n) = \underline{x}(n)^T\ \underline{w}_0(n) = \underline{w}_0(n)^T\underline{x}(n)$, à Equação (6.63),

$$E\{\Delta\underline{w}_0(n)\} = \eta\ E\{\ \underline{x}(n)\underline{x}(n)^T\underline{w}_0(n) - \underline{w}_0(n)^T\underline{x}(n)\underline{x}(n)^T\underline{w}_0(n)\underline{w}_0(n)\} \tag{6.64}$$

Considerando ainda que os vetores $\underline{x}$ e $\underline{w}$ são estatisticamente independentes e que, no equilíbrio,

$$E\{\Delta \underline{w}_0(n)\} = 0, \tag{6.65}$$

$$E\{\underline{w}_0(n)\} = \underline{e}_0 \quad \text{e} \tag{6.66}$$

$$E\{\underline{x}(n)\underline{x}(n)^T\} = C_x \tag{6.67}$$

A Equação (6.64) se torna,

$$0 = C_x \underline{e}_0 - (\underline{e}_0{}^T C_x \underline{e}_0)\underline{e}_0 \tag{6.68}$$

Assim, um vetor no equilíbrio deve obedecer a

$$C_x \underline{e}_0 = (\underline{e}_0{}^T C_x \underline{e}_0)\underline{e}_0 \tag{6.69}$$

Conforme visto na Seção 6.2 (em que estudamos a KLT para PCA ou DSE de um Espaço Vetorial), de acordo com a Equação (6.42),

♦ $\underline{e}_j^T \mathbf{C}_\mathbf{x} \underline{e}_k = \lambda_j$ para $k = j$ e ♦ $\underline{e}_j^T \mathbf{C}_\mathbf{x} \underline{e}_k = 0$ para $k \neq j$

Assim,

$$\underline{e}_0{}^T C_x \underline{e}_0 = \lambda_0 \tag{6.70}$$

e a Equação (6.69) pode ser reescrita como,

$$C_x \underline{e}_0 = \lambda_0 \underline{e}_0 \tag{6.71}$$

A análise da Equação (6.71) mostra que, no equilíbrio, $\underline{w}_0(n)$ deve ser o auto-vetor $\underline{e}_0$ de C_x.

Pré-multiplicando ambos os lados da igualdade expressa na Equação (6.71) por $\underline{e}_0{}^T$ pode-se escrever que,

$$\underline{e}_0{}^T C_x \underline{e}_0 = \underline{e}_0{}^T \lambda_0 \underline{e}_0 \tag{6.72}$$

À esquerda do sinal de igualdade vê-se claramente a Equação (6.70). Além disto, como, à direita da igualdade, λ_0 é um escalar e

$$\underline{e}_0^T \underline{e}_0 = \|\underline{e}_0\| \tag{6.73}$$

onde $\|\underline{e}_0\|$ é a norma Euclidiana do vetor $\underline{e}_0$, pode-se afirmar que

$$\lambda_0 = \lambda_0 \underline{e}_0^T \underline{e}_0 = \lambda_0 \|\underline{e}_0\| \tag{6.74}$$

de onde conclui-se que o valor convergido $\underline{e}_0$ do vetor peso sináptico $\underline{w}_0$ tem norma unitária, conforme pretendíamos mostrar.

Há p possíveis auto-vetores da matriz C_x que satisfazem à condição da Equação (6.71), no entanto, apenas o auto-vetor $\underline{e}_0$, correspondente ao maior auto-valor $\lambda_0 = \lambda_{\max}$, representa uma solução estável.

Demonstremos, agora, que a solução representada pelo auto-vetor associado ao maior auto-valor é a solução estável:

Considere-se que, após um grande número de iterações *n*, o peso sináptico $\underline{w}_0(n)$ esteja na vizinhança de um vetor $\underline{e}_j$ tal que,

$$\underline{w}_0(n) = \underline{e}_j(n) + \underline{\varepsilon}, \qquad n \to \infty \tag{6.75}$$

onde $\underline{\varepsilon}$ é um vetor pequeno que representa a distância do peso sináptico ao auto-vetor.

Partindo da Equação (6.64) onde tomou-se o valor esperado da variação dos pesos sinápticos e considerando as condições expressas pelas Equações (6.65), (6.66), (6.67) e (6.68), encontra-se

$$E\{\Delta \underline{w}_0(n)\} = \eta \{C_x(\underline{e}_j + \underline{\varepsilon}) - (\underline{e}_j + \underline{\varepsilon})^T C_x(\underline{e}_j + \underline{\varepsilon})(\underline{e}_j + \underline{\varepsilon})\}, \; n \to \infty \tag{6.76}$$

Efetuando as operações à direita da igualdade na Equação (6.76), considerando a condição de simetria da matriz Hermitiana C_x e desprezando os termos de segunda ordem

e de ordens maiores em $\underline{\varepsilon}$, é possível aproximar o valor esperado tomado em (6.76) para primeira ordem em $\underline{\varepsilon}$ por

$$E\{\Delta \underline{w}_0(n)\} \cong \eta \{C_x \underline{e}_j + C_x \underline{\varepsilon} - \underline{e}_j{}^T C_x \underline{e}_j \underline{e}_j - 2\underline{\varepsilon}^T C_x \underline{e}_j \underline{e}_j - \underline{e}_j{}^T C_x \underline{e}_j \underline{\varepsilon}\}, n \to \infty \tag{6.77}$$

Uma vez que $\underline{e}_j$ é um auto-vetor da matriz de correlação C_x, pode-se escrever que

$$C_x \underline{e}_j = \lambda_j \underline{e}_j, \qquad j = 1, 2, ..., p\text{-}1 \tag{6.78}$$

e

$$\underline{e}_i{}^T \underline{e}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \tag{6.79}$$

a Equação (6.77) pode ser reescrita na forma simplificada

$$E\{\Delta \underline{w}_0(n)\} \cong \eta \{C_x \underline{\varepsilon} - 2\lambda_j \underline{\varepsilon}^T \underline{e}_j \underline{e}_j - \lambda_j \underline{\varepsilon}\} \tag{6.80}$$

ou,

$$E\{\Delta \underline{w}_0(n)\} \cong \eta \{C_x \underline{\varepsilon} - 2\lambda_j \underline{e}_j \underline{e}_j{}^T \underline{\varepsilon} - \lambda_j \underline{\varepsilon}\} \tag{6.81}$$

onde considerou-se que $\underline{\varepsilon}^T \underline{e}_j$ é um escalar e que este produto interno também pode ser expresso como $\underline{e}_j{}^T \underline{\varepsilon}$.

A mudança média no vetor peso sináptico, projetada sobre a coordenada representada por outro auto-vetor $\underline{e}_i$ da matriz C_x é obtida pré-multiplicando $E\{\Delta \underline{w}_0(n)\}$ por $\underline{e}_i{}^T$. Desta forma,

$$\underline{e}_i{}^T E\{\Delta \underline{w}_0(n)\} \cong \eta \{\underline{e}_i{}^T C_x \underline{\varepsilon} - 2\lambda_j \underline{e}_i{}^T \underline{e}_j \underline{e}_j{}^T \underline{\varepsilon} - \lambda_j \underline{e}_i{}^T \underline{\varepsilon}\} \tag{6.82}$$

ou, considerando as Equações (6.78) e (6.79) e a propriedade de simetria de C_x,

$$\underline{e}_i^T E\{\Delta \underline{w}_0(n)\} \cong \begin{cases} -2\eta\,\lambda_i\,\underline{e}_i^T\,\underline{\varepsilon}, & i = j \\ \eta\,(\lambda_i - \lambda_j)\,\underline{e}_i^T\,\underline{\varepsilon}, & i \neq j \end{cases} \tag{6.83}$$

A análise da Equação (6.83) permite concluir que:

- para $i = j$, A projeção do valor esperado da variação do peso sináptico na direção do próprio auto-vetor associado é sempre negativa.

  Portanto, como a variação do peso sináptico é sempre contrária à distância $\underline{\varepsilon}$, a solução é estável em todas as possíveis direções.

- para $i \neq j$, A projeção do valor esperado da variação do peso sináptico associado ao auto-vetor *j* na direção de um outro auto-vetor p qualquer somente é negativa se $\lambda_j > \lambda_i$.

  Portanto, a variação do peso sináptico será contrária à distância $\underline{\varepsilon}$, indicando uma solução estável, se e somente se o auto-valor associado ao auto-vetor *j* for maior do que todos os demais auto-valores correspondentes a outras direções.

O problema da instabilidade do algoritmo após a incorporação da Regra de Oja (ou seja, após a regra de Aprendizado Hebbiano ter sido deflacionada de um fator $\eta\, y_0^{\,2}(n)\underline{w}_0(n)$) pode ser assim equacionado:

Adotando procedimento semelhante ao utilizado no caso da regra não deflacionada:

Ao tomar a derivada da variação do peso sináptico $\Delta\underline{w}_0$ em relação ao próprio peso sináptico $\underline{w}_0$, poderemos investigar de que forma a variação do peso sináptico se comporta ao longo do tempo *n*.

Assim,

$$\frac{d}{d(\underline{w}_0(n))}(\Delta\underline{w}_0(n)) = \frac{d}{d(\underline{w}_0(n))}(\eta\, y_0(n)(\underline{x}(n) - y_0(n)\underline{w}_0(n))) \tag{6.84}$$

Considerando as formas expandidas, na Equação (6.84),

$$\frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) =$$

$$= \frac{d}{d(w_{0i}(n))}\{\eta\,[\sum_i x_i(n)w_{0i}(n)][\,x_i(n) - w_{0i}(n)\sum_i x_i(n)w_{0i}(n)]\} \tag{6.85}$$

Tomando a derivada,

$$\frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) =$$

$$= [\eta \sum_i x_i(n)w_{0i}(n)][-w_{0i}(n)x_i(n) - \sum_i x_i(n)w_{0i}(n)] +$$
$$+ [x_i(n) - w_{0i}(n)\sum_i x_i(n)w_{0i}(n)][\eta\, x_i(n)] \tag{6.86}$$

Distribuindo os produtos mostrados em (6.86),

$$\frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) =$$

$$= [-\eta\, w_{0i}(n)x_i(n)\sum_i x_i(n)w_{0i}(n) - \eta \sum_i x_i(n)w_{0i}(n) \sum_i x_i(n)w_{0i}(n)] +$$
$$+ [\eta\, x_i(n)x_i(n) - \eta\, x_i(n)w_{0i}(n)\sum_i x_i(n)w_{0i}(n)] \qquad (6.87)$$

Assim, a derivada da variação do peso sináptico Δw_{0i} em relação ao próprio peso sináptico w_{0i} pode ser expressa por

$$\frac{d}{d(w_{0i}(n))}(\Delta w_{0i}(n)) =$$

$$= \eta\ \ x_i{}^2(n) - 2\eta\, x_i(n)w_{0i}(n)\sum_i x_i(n)w_{0i}(n) - \eta\,(\sum_i x_i(n)w_{0i}(n))^2 \qquad (6.88)$$

- **A observação da Equação (6.88) não mais permite afirmar que a derivada será sempre positiva, como em (6.59).**
- **Portanto, a variação da variação do peso sináptico não é mais sempre positiva.**
- **A convergência do algoritmo ocorre se os valores de $\underline{x}(n)$ e $\underline{w}_0(n)$ são tais que a Equação (6.88) resulta em um valor negativo.**

De acordo com o desenvolvimento seguido, podemos agora afirmar que:

- um único neurônio linear submetido à regra Hebbiana de aprendizado auto-organizado tende a extrair a primeira componente principal do espaço vetorial de dados de entrada [16],
- componente esta que corresponde ao maior auto-valor da matriz de covariância dos dados de entrada.

6.3.2 O Algoritmo Hebbiano Generalizado

A generalização do algoritmo apresentada nesta seção segue a proposta de Sanger (1989) em [7] e é obtida a partir do Algoritmo Hebbiano estabilizado pela Regra de Oja.

O processo de generalização consiste em aplicar as regras propostas por Sanger e Oja a uma RNA progressiva, composta não mais por um único neurônio linear (conforme Figura 6.7) e, sim, por uma camada de neurônios lineares, conforme mostra a Figura 6.8.

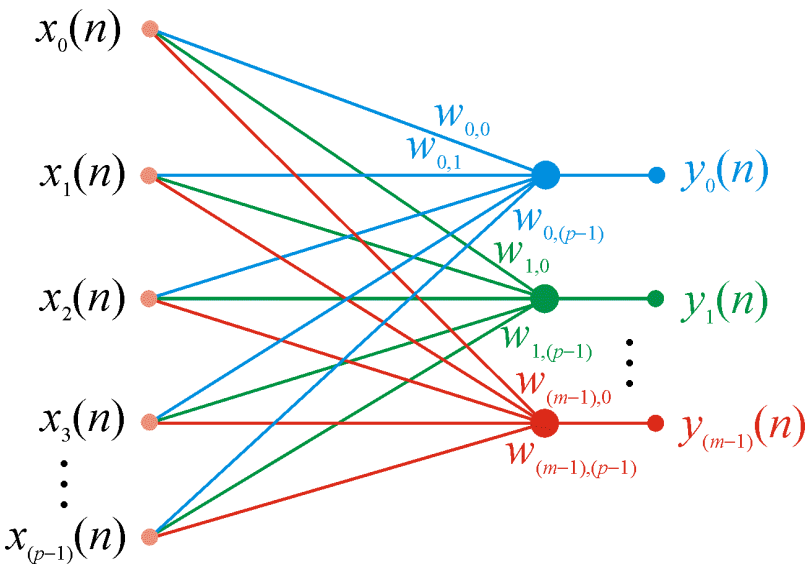

Figura 6.8: RNA progressiva com uma única camada de neurônios lineares, a ser treinada pelo Aprendizado Hebbiano Generalizado.

- A RNA treinada por esta regra executará a análise dos componentes principais sobre o espaço vetorial que compõe os dados de entrada.

Dito de outra forma:

- Cada um dos neurônios da camada de neurônios lineares da RNA extrairá cada um dos componentes principais (do espaço vetorial de dados original formado pelos vetores $\underline{x}(n)$), associado a cada um dos auto-valores e auto-vetores correspondentes da matriz C_x.

Observe que:

- A RNA da Figura 6.8 tem p nós na camada de entrada e m neurônios na camada de saída, com $m < p$.

- Os pesos sinápticos w_{ji} conectam os nós de entrada i aos neurônios da camada de saída j, com $i = 0, 1, ..., p-1$ e $j = 0, 1, ..., m-1$ (considerando, novamente, a notação expandida para o vetor de pesos sinápticos).

- A saída $y_j(n)$ produzida pelo neurônio *j*, no tempo *n*, em resposta ao conjunto de entradas $x_i(n)$ é dada por

$$y_j(n) = \sum_{i=0}^{p-1} w_{ji}(n) x_i(n), \qquad j = 0, 1, ..., m\text{-}1 \tag{6.89}$$

- A adaptação do peso sináptico devida ao Algoritmo Hebbiano Generalizado é expressa pela Equação (6.90),

$$\Delta w_{ji}(n) = \eta \{ y_j(n) x_i(n) - y_j(n) \sum_{k=0}^{j} w_{ki}(n) y_k(n) \}, \quad \begin{matrix} i = 0, 1, ..., p\text{-}1 \\ j = 0, 1, ..., m\text{-}1 \end{matrix} \qquad (6.90)$$

onde $\Delta w_{ji}(n)$ é a mudança aplicada ao peso sináptico $w_{ji}(n)$ no tempo n e η é o parâmetro razão de aprendizado.

> Ao observar a Equação (6.90), considerando $j = 0$, percebe-se que o algoritmo para um simples neurônio (apresentado na Seção anterior) é um caso particular do Algoritmo Hebbiano Generalizado ora apresentado.

A equação que expressa a variação no peso sináptico imposta pelo Algoritmo Hebbiano Generalizado pode também ser escrita como:

$$\Delta w_{ji}(n) = \eta \; y_j(n) x'_i(n) - \eta \, y_j{}^2(n) w_{ji}(n), \quad \begin{matrix} i = 0, 1, ..., p\text{-}1 \\ j = 0, 1, ..., m\text{-}1 \end{matrix} \qquad (6.91)$$

onde o vetor $\underline{x}'(n)$ representa a forma deflacionada do vetor de entrada $\underline{x}(n)$, conforme Equação (6.92).

$$x'_i(n) = x_i(n) - \sum_{k=0}^{j-1} w_{ki}(n) y_k(n) \qquad (6.92)$$

Cabe, neste ponto, examinar a operação do algoritmo passo a passo, o que é feito na Tabela 6.4:

(I) **Para o primeiro neurônio $(j = 0)$:**

- A Equação (6.91) reduz-se ao caso de um único neurônio (examinado anteriormente).
- O primeiro neurônio a convergir é aquele associado ao maior auto-valor.
- A rede extrai o primeiro componente principal dos vetores de dados de entrada $\underline{x}(n)$.

(II) **Para o segundo neurônio $(j = 1)$:**

- A Equação (6.92) torna-se $x_i'(n) = x_i(n) - w_{0i}(n)y_0(n)$. (6.93)
- Desde que o primeiro neurônio já tenha convergido para o primeiro componente principal, o segundo neurônio vê vetores de entrada $\underline{x}'(n)$ dos quais o primeiro auto-vetor da matriz C_x já foi extraído (deflacionado), conforme pode ser visto na Equação (6.90).
- O segundo neurônio extrairá, portanto, o primeiro componente principal de $\underline{x}'(n)$, que é, na verdade, o segundo componente principal do espaço vetorial de entrada original dos vetores $\underline{x}(n)$ (segundo auto-valor e correspondente auto-vetor da matriz C_x).

(III) **Para os demais neurônios da rede:**

- Cada conjunto de pesos sinápticos convergido representa um auto-vetor da matriz de correlação do conjunto de dados de entrada (espaço vetorial de entrada).
- Os auto-vetores obtidos desta forma encontram-se ordenados em ordem decrescente de valor dos auto-valores associados.

Tabela 6.4: Operação do Algoritmo Hebbiano Generalizado.

6.3.3 O Treinamento da RNA para Proceder à DSE (ou PCA) de um Conjunto de Vetores

Nesta seção utilizaremos o Algoritmo Hebbiano Generalizado para treinar uma RNA progressiva, composta por uma única camada de neurônios lineares, com o objetivo de proceder à DSE (ou PCA) de um conjunto de vetores de dados originado da vetorização dos elementos de uma imagem (*picture elements* ou *pixels*).

Estudaremos também o processo de compressão permitido pela Decomposição em Sub-Espaços obtida através do Algoritmo Hebbiano Generalizado.

O Algoritmo Hebbiano Generalizado doravante será referido como GHAPCA *(Generalized Hebbian Algorithm to perform Principal Component Analysis*).

Sabemos que o GHAPCA aplica-se apropriadamente à compressão de dados pois concentra em um número reduzido de elementos a maior parte da energia (variância) dos dados, permitindo que a escolha dos componentes a serem descartados seja ótima no sentido do erro médio quadrático (MSE).

- As matrizes que representam as imagens são particionadas em blocos (submatrizes), conforme visto na Seção 6.2.3.
- Estas submatrizes são transformadas em vetores linha, que formarão o conjunto de treino da RNA.
- A RNA é treinada sob a regra desenvolvida na Seção anterior (6.3.2) para o Aprendizado Hebbiano Generalizado, tendo por objetivo extrair os componentes principais da imagem.

- A compressão obtida pelo método está relacionada ao número de componentes principais da imagem que serão descartadas.
- Quanto mais componentes descartadas, maior será a compressão e menor a fidelidade da imagem reconstruída, com respeito à imagem original. Este compromisso é regido pelo tipo de aplicação a que se destina a compressão.

- A RNA terá tantos neurônios quantas componentes principais se desejar preservar.
- Após o treinamento, a rede terá armazenado em seus pesos sinápticos os auto-vetores correspondentes aos auto-valores da matriz covariância do conjunto de dados de entrada que se especificou preservar.

- A convergência do algoritmo baseia-se em um critério de parada derivado das características esperadas dos vetores estimados obtidos após um número suficientemente grande de iterações.

- Duas otimizações são propostas para o algoritmo, objetivando redução de tempo computacional:
  - o critério para atualização da razão de aprendizado;
  - a aceleração de convergência do algoritmo por janelamento de treino.

- A partir dos auto-vetores obtidos e das saídas dos neurônios após a convergência do algoritmo, para cada vetor pertencente ao conjunto de treino é obtida a reconstrução da imagem, estimada a partir das componentes principais identificadas.

- A fidelidade da imagem reconstruída pelo algoritmo é determinada pelo parâmetro Relação Sinal-Ruído de Pico (*PSNR*), definido na Seção 6.2.1, Equação (6.47).

6.3.3.1 A Representação dos Dados

Quando aplicado à compressão de imagens digitais, o conjunto de dados a ser apresentado à RNA é composto de um conjunto de vetores originado da vetorização dos elementos da imagem.

Seja uma imagem digital, bidimensional, representada por uma matriz $f(x,y)$ de $N \times N$ posições, sendo N uma potência inteira de dois.

Da mesma forma que na Seção 6.2.3, utilizaremos imagens em tons de cinza (*grayscale*), conforme lá definidas. Portanto, quantizadas em 256 níveis de cinza.

Antes de qualquer processamento, os valores de pixel da imagem são normalizados para que o maior valor de pixel seja unitário.

Os dados são apresentados à camada de entrada da RNA sob a forma de vetores unidimensionais de p elementos. Cada um destes vetores é obtido do particionamento da imagem em submatrizes não sobrepostas (ou blocos), de dimensões $l \times l$, onde $l = 2^n$ elementos (n é uma potência inteira de dois e é determinado de acordo com as características estatísticas da imagem). O número de elementos dos vetores unidimensionais é, portanto, $p = l^2$. As imagens utilizadas neste trabalho possuem 128×128 elementos.

Em imagens, *pixels* vizinhos são, em geral, correlacionados. À medida que as distâncias entre pixels aumentam, a correlação decai substancialmente. Portanto, o tamanho escolhido para as submatrizes deve ser tal que permita captar elementos suficientemente correlacionados. Sendo originários de pixels correlacionados, os vetores gerados das submatrizes, quando apresentados à RNA, possibilitam uma boa estimativa dos componentes principais do conjunto de dados representativo da imagem.

Uma submatriz de 8×8 elementos foi determinada experimentalmente como sendo a que provê melhores resultados para imagens de 128×128 pixels. Submatrizes maiores que 8×8 implicam em uma maior diversidade de variações por vetor de treino (menor correlação), tornando necessário aumentar o número de componentes principais

utilizadas para que se obtenha a mesma *PSNR*. Submatrizes menores que 8×8 pixels têm seus elementos consideravelmente correlacionados – o que melhora a *PSNR* – no entanto, a taxa de compressão cai devido ao maior número de vetores utilizados para representar a imagem.

As dimensões da matriz representativa da imagem são múltiplas inteiras das dimensões das submatrizes consideradas (de dimensões $l \times l$). Do particionamento resultam $n_b = (N/l)^2$ submatrizes.

O número de submatrizes resultantes do particionamento da matriz é o número de vetores que serão utilizados para treino da RNA.

A apresentação de todo o conjunto de vetores à RNA, uma vez, corresponde a uma época.

Neste trabalho, $N = 128$ e $l = 8$. Portanto, o conjunto de treino apresentado à RNA é composto de

$$\left(\frac{N}{l}\right)^2 = \left(\frac{128}{8}\right)^2 = 256 \quad \text{vetores de} \quad l^2 = 64 \quad \text{elementos.} \tag{6.94}$$

As submatrizes são transformadas em vetores linha. Cada vetor linha é obtido de acordo com a lei de formação mostrada na Tabela 6.5[1] :

(1)	Os l primeiros elementos deste vetor linha são os pertencentes à primeira linha da submatriz de $l \times l$ elementos, lida da esquerda para a direita.
(2)	Os próximos l elementos deste vetor linha são os pertencentes à segunda linha da submatriz, lida da mesma forma.
(3)	Assim prossegue-se, até que sejam esgotados todos os l^2 elementos da submatriz.

Tabela 6.5: Lei de formação para cada vetor linha.

[1] Observação: Sugere-se o experimento em que os vetores sejam montados a partir de alguma diferente lei de formação, mais adequada às características de uma particular imagem que se deseja comprimir. Lembrando sempre que o principal requerimento a ser atendido ao vetorizar a submatriz é que o grau de correlação existente entre pixels vizinhos seja maximizado ou, pelo menos, aumentado. Por exemplo: os pixels de cada submatriz podem ser lidos de acordo com uma "trajetória" espiral.

Uma exigência para a análise dos componentes principais é que os vetores envolvidos possuam média zero (Seção 6.2). Portanto, obtém-se o vetor média, resultante da média dos $n_b = (N/l)^2$ vetores e subtrai-se este vetor de cada vetor do conjunto, assim constituindo o conjunto de treino da RNA.

Cada vetor apresentado à RNA constitui uma amostra do conjunto de treino, em nosso caso, composto de 256 vetores de 64 elementos.

A cada época, a ordem de apresentação dos vetores do conjunto de treino é mudada aleatoriamente, de forma semelhante ao ato de embaralhar um conjunto de cartas. São justificativas e características do ato de embaralhamento:

- Os vetores de dados são apresentados à RNA em ordem aleatória, para impedir que a apresentação ordenada destes vetores gere um ciclo de aumento e decréscimo de valores dos pesos sinápticos.
- Este ciclo (gerado pela ordem de apresentação dos vetores de dados) poderá interferir no aprendizado da RNA, dificultando a convergência dos pesos sinápticos para os auto-vetores da matriz covariância dos dados.
- O embaralhamento é implementado por um algoritmo que escolhe aleatoriamente dois vetores do conjunto, permutando-os entre si.
- Esta operação é executada ao final de cada época, tantas vezes quantos forem o número de vetores de dados.
- A função densidade de probabilidade [16] do gerador de números aleatórios utilizado é uniforme.

6.3.3.2 A Estrutura da Rede Neural Artificial

O modelo de RNA que é treinado pelo algoritmo GHAPCA é aquele mostrado na Figura 6.8, em que a camada de entrada da RNA possui p nós. Os p nós da camada de entrada da RNA recebem os respectivos elementos dos vetores de dados pertencentes ao conjunto de treino obtido do particionamento da matriz representativa da imagem. Os

vetores são compostos por $p = l^2$ elementos, definindo o número de nós computacionais da camada de entrada da RNA.

Cada vetor representativo dos dados de entrada é da forma

$$\underline{x} = [x_0 \; x_1 \; ... \; x_{p-1}]^T \tag{6.95}$$

A camada de saída é formada por m neurônios, representados por

$$\underline{y} = [y_0 \; y_1 \; ... \; y_{m-1}]^T \tag{6.96}$$

A cada neurônio da camada de saída está associado, após a convergência do algoritmo, o conjunto de pesos sinápticos representativo do auto-vetor associado àquele componente principal.

O número de neurônios da camada de saída é definido pelo número de componentes principais que se deseja extrair do conjunto de dados representativo da imagem original (associados aos auto-valores e auto-vetores da matriz covariância), em nosso caso, m .

Quanto menor o número de componentes principais considerados, maior a taxa de compressão e menor a *PSNR*, para uma mesma imagem.

Os pesos sinápticos, que conectam os nós fonte i aos neurônios j da camada de saída são representados por

$$w_{j\,i} \; ; \;\; i = 0, 1, ..., p-1 \, , \;\; j = 0, 1, ..., m-1 \tag{6.97}$$

Após a convergência do algoritmo, a RNA terá "cristalizado" em seus pesos sinápticos (cujos elementos são os $w_{j\,i}$ que compõem o vetor de pesos $\underline{w}_j$) os auto-vetores (cujos elementos são os $e_{j\,i}$ que compõem o auto-vetor $\underline{e}_j$) associados aos auto-valores (λ_j) da matriz covariância dos dados de entrada. Ou seja:

- O auto-vetor $\underline{e}_0$, associado ao maior auto-valor, terá seus elementos "cristalizados" nos elementos w_{0i} , $i = 0, 1, ..., p-1$, do vetor de pesos sinápticos $\underline{w}_0$, associado ao primeiro neurônio da camada de saída.

- O neurônio y_1 (segundo neurônio da camada de saída) conterá, nos elementos w_{1i} , $i = 0, 1, ..., p-1$, de seu vetor de pesos sinápticos associado $\underline{w}_1$, os elementos e_{1i} , $i = 0, 1, ..., p-1$, do segundo auto-vetor $\underline{e}_1$, associado ao segundo auto-valor e assim, sucessivamente até o último neurônio, y_{m-1}.

A título de exemplo, a Figura 6.9 apresenta a representação gráfica dos pesos sinápticos (auto-vetores) "cristalizados" nos neurônios da camada de saída da RNA, após a convergência do algoritmo, para o modelo particular de RNA utilizado neste estudo. A rede da figura possui $p = l^2 = 64$ nós computacionais na camada de entrada e são considerados k auto-vetores ($k < m$) para a reconstrução dos dados.

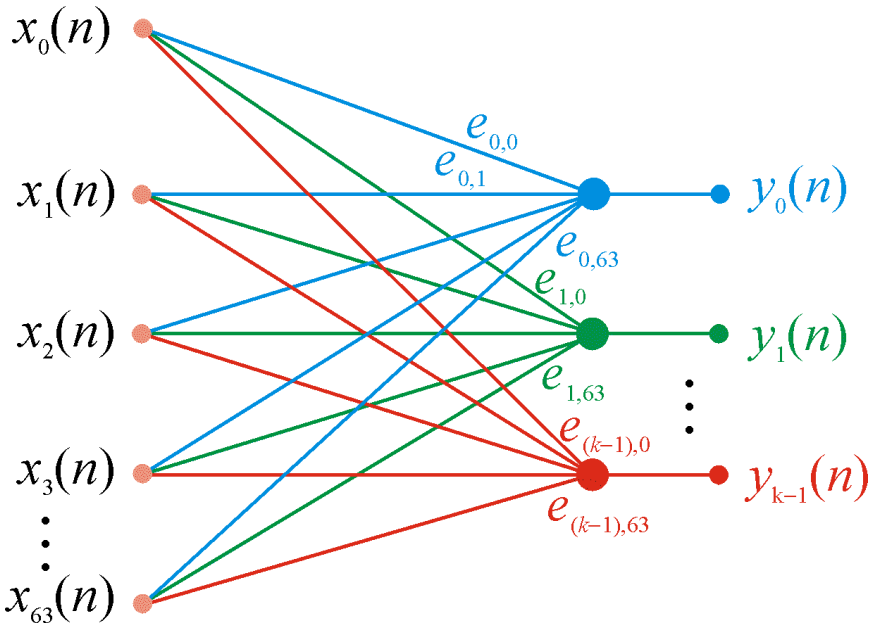

Figura 6.9: Representação gráfica da RNA com p =64 nós computacionais na camada de entrada e k neurônios na camada de saída, apresentando os auto-vetores da matriz covariância dos dados de entrada, após a convergência do algoritmo *GHAPCA*.

A saída $y_0(n)$ do neurônio da camada de saída da RNA para a iteração *n,* produzida em resposta ao conjunto de entradas $x_i(n)$, componentes do vetor $\underline{x}(n)$, é expressa por

$$y_0(n) = \underline{w}_0(n)^T \underline{x}(n) = \underline{x}(n)^T \underline{w}_0(n) = \sum_{i=0}^{p-1} w_{0i}(n) x_i(n) \tag{6.98}$$

A Equação (6.99) expressa a regra de Aprendizado Hebbiano, após deflacionada de um fator $\eta\, y_0^{\,2}(n)\underline{w}_0(n)$,

$$\Delta w_0(n) = \eta\ y_0(n)\underline{x}(n) - \eta\, y_0^{\,2}(n)\underline{w}_0(n) \tag{6.99}$$

Partindo das Equações (6.98) e (6.99) e implementando a condição adicional necessária à ortonormalização, a regra generalizada fica definida pelas Equações (6.100) e (6.101), aqui apresentadas na forma expandida.

$$y_j(n) = \underline{w}_j(n)^T \underline{x}(n) = \sum_{i=0}^{p-1} w_{ji}(n) x_i(n), \qquad j = 0, 1, ..., m\text{-}1 \tag{6.100}$$

A atualização dos pesos sinápticos é dada por

$$w_{ji}(n+1) = w_{ji}(n) + \eta\ y_j(n)\left\{ x_i(n) - \sum_{k=0}^{j} w_{ki}(n)\, y_k(n) \right\}, \begin{array}{l} i = 0, 1, \cdots, p\text{-}1 \\ j = 0, 1, \cdots, m\text{-}1 \end{array} \tag{6.101}$$

Sob esta regra de aprendizado, os k vetores peso sinápticos convergem para os k auto-vetores, correspondentes aos k maiores auto-valores dos dados de entrada, após um número de iterações *n* suficientemente grande.

6.3.3.3 O Critério de Parada

O critério de parada adotado para a convergência do algoritmo é derivado das características esperadas dos vetores estimados obtidos.

A convergência do algoritmo ocorre quando os pesos sinápticos armazenados na RNA estabilizam em valores que correspondem aos auto-vetores associados aos auto-valores da matriz Covariância dos dados de entrada. Então, no equilíbrio , quando $n \to \infty$, é correto que

$$E\{\Delta w_{ji}(n)\} = 0 \tag{6.102}$$

Baseado nesta propriedade, pode-se considerar que os pesos sinápticos do neurônio j tenham convergido para o auto-vetor associado, quando as razões entre todos os seus p pesos sinápticos, de uma iteração para a próxima, sejam unitárias, isto é, $(w_{ji}(n+1)/w_{ji}(n)) \approx 1$, $i = 0, 1, \ldots, p-1$.

Para evitar efeitos de variação transiente em $w_{ji}(n)$ a cada iteração (devida ao uso eventual de uma alta razão de aprendizado) e no intuito de simplificar o critério de convergência, utilizou-se neste trabalho a média em três iterações da razão $\|w_j(n+1)\| / \|w_j(n)\|$, onde w_j representa o vetor de p pesos sinápticos do neurônio *j*. Especificamente, se

$$\frac{1}{3}\left\{\frac{\|w_j(n+1)\|}{\|w_j(n)\|} + \frac{\|w_j(n+2)\|}{\|w_j(n+1)\|} + \frac{\|w_j(n+3)\|}{\|w_j(n+2)\|}\right\} - 1 < 1\times 10^{-4} \tag{6.103}$$

considera-se que o vetor peso sináptico do neurônio *j* tenha convergido para o auto-vetor associado. O limiar de 1×10^{-4} foi determinado experimentalmente.

6.3.3.4 O Critério para Atualização da Razão de Aprendizado

É conveniente, para a convergência do algoritmo, que a razão de aprendizado seja inicializada com um valor baixo (da ordem de 1×10^{-9}, por exemplo).

Após a convergência do primeiro auto-vetor, no entanto, a razão de aprendizado deve ser aumentada, pois, como já ocorreu a deflação da primeira componente principal do vetor de treino, o processo de aprendizado pode ser acelerado sem detrimento da convergência.

O critério adotado baseia-se na proposta de Chen e Chang [13]. A razão de aprendizado é feita, a cada época, igual ao inverso do auto-valor instantâneo dividido por uma constante arbitrária α que varia com o tipo de imagem.

$$\eta = \frac{1/\lambda}{\alpha} \tag{6.104}$$

Experimentalmente, determinou-se que a constante α deve estar contida no intervalo [500, 2000].

Um valor baixo para α acelera a convergência do algoritmo. No entanto, valores inferiores a 500 podem conduzir a *overflow* das variáveis de ponto flutuante de 8 bytes utilizadas neste trabalho.

Um valor muito alto para α torna a razão de aprendizado demasiadamente pequena, retardando a convergência.

6.3.3.5 A Aceleração de Convergência do Algoritmo por Janelamento de Treino

À medida que um neurônio converge para um auto-vetor, todos os subseqüentes ajustam-se correspondentemente, obedecendo ao princípio da deflação, conforme Equação (6.101).

No entanto, é característica do aprendizado Hebbiano que cada um dos m auto-vetores inicie o próprio processo de convergência após a convergência do auto-vetor anterior, não sendo alterados os pesos sinápticos dos neurônios já convergidos.

Observou-se experimentalmente que somente estão realmente próximos do ponto de convergência alguns poucos neurônios imediatamente subseqüentes ao que já convergiu.

O processo de acomodação continuada de todos os neurônios subseqüentes a um neurônio que já convergiu implica em um custo computacional desnecessário, já que somente estarão realmente no caminho da convergência aqueles poucos neurônios imediatamente subseqüentes ao que já convergiu.

Para evitar que todos os m auto-vetores sejam ajustados durante a deflação de um particular auto-vetor, é proposto para o *GHAPCA* o algoritmo denominado janelamento de convergência. O janelamento de convergência é um novo método que considera a convergência de k_{jt} $(k_{jt} < m)$ auto-vetores de cada vez. Havendo convergido o primeiro auto-vetor da janela de convergência, a janela se desloca uma posição para a frente.

Este processo de janelamento é aplicado ao algoritmo sem detrimento algum para a precisão do processo de convergência.

Determinou-se experimentalmente a conveniência de uso de uma janela de convergência que considera a convergência de três auto-vetores.

O ganho de tempo de processamento devido à aceleração de convergência do algoritmo por janelamento de treino, nos experimentos realizados neste trabalho (em que é considerada uma janela de convergência de três auto-vetores), é da ordem de

$$t_{cjt} \approx \frac{t_c}{10} \tag{6.105}$$

onde t_{cjt} denota o tempo de processamento com aceleração do algoritmo por janelamento de treino e t_c o tempo de processamento que considera o ajuste de todos os auto-vetores durante a convergência de um auto-vetor.

A relação entre os parâmetros t_{cjt} e t_c foi experimentalmente investigada neste estudo. Esta relação é evidentemente dependente do número total de auto-vetores que estão sendo considerados (m), o que equivale a dizer que é dependente do número total de componentes principais que estão sendo consideradas. Quanto maior for m, maior será o ganho de tempo computacional devido à aplicação do janelamento de convergência.

6.3.3.6 O Treinamento da RNA

A cada época de treino são apresentados à RNA as n_b submatrizes (ou blocos) da imagem, sob a forma de vetores de treino. Portanto, uma época é constituída de n_b iterações. A cada iteração *n*, $n = 0, 1, ..., n_b - 1$, é apresentado o *n-ésimo* vetor do conjunto de treino à RNA , produzindo a saída $y(n)$.

Após a convergência, a RNA apresenta um conjunto de pesos sinápticos constituído por m vetores de p elementos, equivalente aos auto-vetores associados aos auto-valores da matriz covariância dos dados de entrada, denotados por $\underline{e}_j$ ($\underline{e}_j$ (equilíbrio)$= \underline{e}_j$).

A ordem de convergência dos pesos sinápticos é tal que, nos pesos sinápticos associados ao primeiro neurônio da camada de saída da RNA $(\underline{w}_0(n))$ estão as

componentes do auto-vetor $(\underline{e}_0)$ associado ao maior auto-valor (λ_0) da matriz de covariância dos dados de entrada.

Nos pesos sinápticos associados ao segundo neurônio da camada de saída da RNA $(\underline{w}_1(n))$ estão as componentes do auto-vetor $(\underline{e}_1)$ associado ao segundo maior auto-valor (λ_1) da matriz covariância dos dados de entrada e, assim, em ordem decrescente até o último peso sináptico $(\lambda_0 \Leftrightarrow \underline{e}_0,\ \lambda_1 \Leftrightarrow \underline{e}_1,\ \lambda_2 \Leftrightarrow \underline{e}_2,\ \cdots,\ \lambda_j \Leftrightarrow \underline{e}_j)$.

6.3.3.7 A Reconstrução dos Dados

A Figura 6.10 apresenta uma RNA já convergida para os auto-vetores $\underline{e}$, com p nós na camada de entrada e m neurônios na camada de saída. Está sendo apresentada à rede a *n-ésima* submatriz que compõe a imagem.

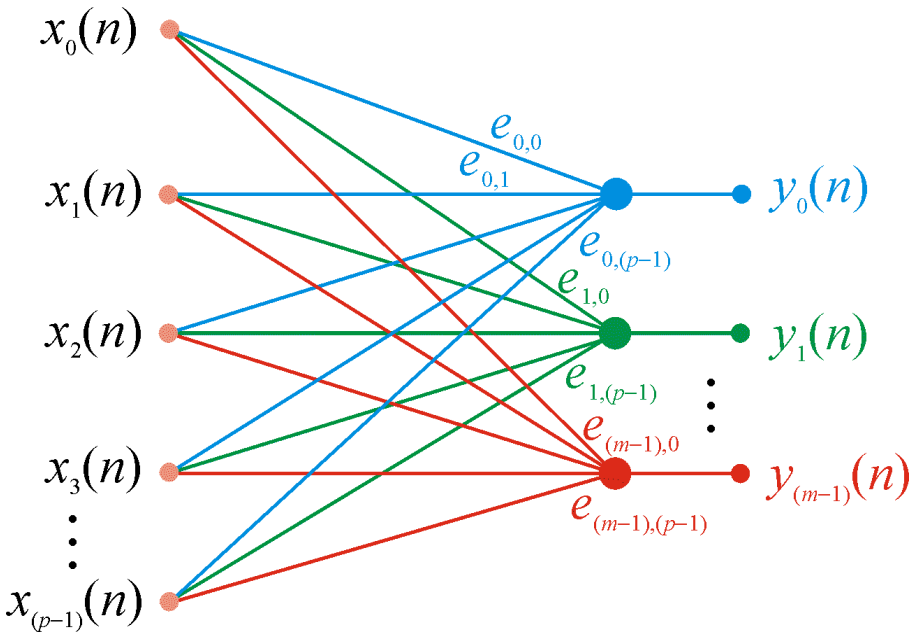

Figura 6.10: RNA com p nós na camada de entrada e m neurônios na camada de saída, apresentando os auto-vetores $\underline{e}$ da matriz covariância dos dados de entrada, após a convergência do algoritmo *GHAPCA*.

A reconstrução dos dados é obtida pela Equação (6.106).

$$\hat{x}_i(n) = \sum_{j=0}^{m-1} y_j(n)\, e_{j\,i} \tag{6.106}$$

onde cada $\hat{x}_i(n)$ representa a reconstrução do vetor *n* associado ao bloco ou submatriz de ordem *n* da imagem estimada.

Por exemplo, para reconstrução da submatriz de ordem zero, as componentes do vetor $\hat{x}_i(0)$ com $i = 0, 1, ..., p-1$ são reconstruídas por

$$\hat{x}_0(0) = y_0(0)\, e_{00} + y_1(0)\, e_{10} + \cdots + y_{m-1}(0)\, e_{(m-1)0}$$

$$\hat{x}_1(0) = y_0(0)\, e_{01} + y_1(0)\, e_{11} + \cdots + y_{m-1}(0)\, e_{(m-1)1}$$

...

$$\hat{x}_{p-1}(0) = y_0(0)\, e_{0(p-1)} + y_1(0)\, \mathrm{e}_{1(p-1)} + \cdots + y_{m-1}(0)\, e_{(m-1)(p-1)} \tag{6.107}$$

Ao *n-ésimo* vetor reconstruído, $\underline{\hat{x}}(n)$, com $n = 0, 1, ..., n_b - 1$, é acrescentado o vetor média previamente extraído. O vetor resultante é transformado no bloco (ou submatriz) de ordem *n,* componente da imagem estimada. O vetor linha $\underline{\hat{x}}(n)$ é transformado na *n-ésima* submatriz, de acordo com o seguinte algoritmo:

- A primeira linha da *n-ésima* submatriz (vista aqui como uma matriz de $l \times l$ elementos) é formada pelos l primeiros elementos do *n-ésimo* vetor.
- A primeira linha da *n-ésima* submatriz (vista aqui como uma matriz de $l \times l$ elementos) é formada pelos l primeiros elementos do *n-ésimo* vetor.
- A segunda linha da submatriz é formada pelos Segundos l elementos do *n-ésimo* vetor.
- Procede-se desta forma até que todos os $p = l^2$ elementos do *n-ésimo* vetor sejam transferidos para a *n-ésima* submatriz.

O conjunto de n_b submatrizes irá compor a imagem, sendo $N \times N$ o tamanho da imagem original.

6.3.3.8 A Compressão dos Dados

A compressão decorrente do uso do método é função do número de componentes principais da imagem que vierem a ser descartadas. Quanto maior o número de componentes principais descartadas, maior a compressão e menor a fidelidade da imagem reconstruída com relação à imagem original.

A compressão é definida a partir da equação de reconstrução dos dados (Equação 6.106). Para reconstrução completa dos dados, todos os m auto-vetores (associados aos m componentes principais) são considerados.

A compressão será obtida se, ao invés de m auto-vetores, forem considerados k auto-vetores, com $k < m$.

Para exemplificar, considere-se a Figura 6.11.

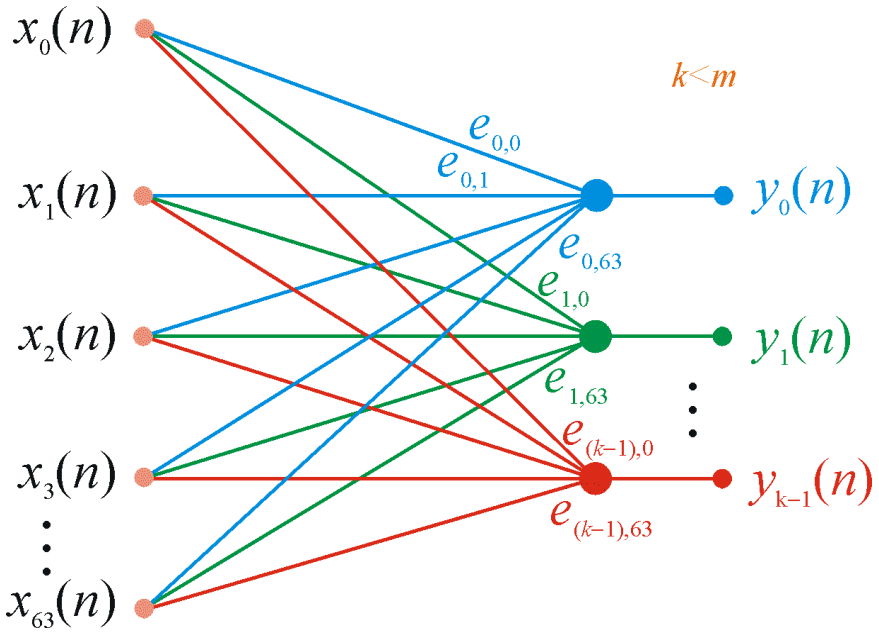

Figura 6.11: Modelo de RNA utilizada para compressão. A rede apresenta p=64 nós de entrada (para submatrizes com $p = l^2 = 64$ elementos) e k neurônios na camada de saída (para extrair k componentes principais).

Na rede mostrada na figura, as componentes do vetor $\underline{\hat{x}}(0)$, $\hat{x}_i(0)$, com $i = 0, 1, ..., p-1$ são reconstruídas por

$$\hat{x}_0(0) = y_0(0)\, e_{00} + y_1(0)\, e_{10} + \cdots + y_{k-1}(0)\, e_{(k-1)0}$$
$$\hat{x}_1(0) = y_0(0)\, e_{01} + y_1(0)\, e_{11} + ... + u_{k-1}(0)\, e_{(k-1)1}$$
$$...$$
$$\hat{x}_{p-1}(0) = y_0(0)\, e_{0(p-1)} + y_1(0)\, e_{1(p-1)} + \cdots + y_{k-1}(0)\, e_{(k-1)(p-1)} \qquad (6.108)$$

Como $k < m$, as componentes do vetor $\underline{\hat{x}}(0)$, $\hat{x}_i(0)$, com $i = 0, 1, ..., p-1$, sofrem efetiva redução dimensional.

Os auto-vetores $\underline{e}_{ji}$, com $j = k, k+1, \cdots, m\text{-}1$ não considerados na reconstrução de $\underline{\hat{x}}(0)$ e responsáveis pela redução dimensional, não representam perda considerável de informação por estarem associados aos auto-valores de menor energia da matriz covariância do conjunto de dados.

Se a arquitetura da rede for tal que possua k neurônios na camada de saída, cada um com $p = l^2$ sinapses, após a convergência do algoritmo a rede apresentará k vetores de p elementos (referentes aos auto-vetores convergidos) e n_b vetores de k elementos (referentes aos $n_b = (N/l)^2$ conjuntos de valores de saída dos k neurônios da camada de saída da rede). A partir destes elementos e do vetor média previamente extraído (composto de p elementos) será reconstruída a imagem comprimida.

A razão entre o número de unidades de armazenamento necessárias para representar a imagem original e o número de unidades de armazenamento necessárias para representar a imagem comprimida é chamada de razão de compressão e é denotada por ρ. A razão de compressão pode, então, ser determinada por

$$\rho = \frac{k\,(p + n_b) + p}{N \times N}, \text{ com } n_b = (N/l)^2 \qquad (6.109)$$

O método permite atingir uma compressão efetiva maior do que a compressão expressa pela razão de compressão ρ, através da adoção de um critério para alocação dos bits necessários para o armazenamento da imagem comprimida.

Neste trabalho a compressão é avaliada através da razão de compressão ρ. No entanto, são sugeridos dois critérios para otimizar a compressão baseados na alocação de diferentes números de bits por unidade de armazenamento da imagem comprimida.

No primeiro critério a otimização é alcançada através da atribuição de um maior número de bits para o armazenamento dos auto-vetores mais significativos da imagem (associados aos maiores auto-valores) e de um menor número de bits para os auto-vetores menos significativos (associados aos menores auto-valores) [7].

A segunda sugestão é a utilização do Código Huffman que é baseado na entropia da imagem comprimida [17] [18].

Avalia-se a fidelidade dos resultados obtidos pelo *GHAPCA*, através da determinação da Relação Sinal-Ruído de Pico (*PSNR*), definida na Seção 6.2.1, Equação (6.47), onde as diferenças entre a imagem comprimida e a imagem original são encaradas como se fossem ruído.

A Figura 6.12 mostra novamente a imagem em *grayscale* "Lenna" de tamanho 128×128 pixels, utilizada na Seção 6.2.3, em que a compressão foi devida à Transformada Karhunen-Loève.

A Figura 6.13 mostra a imagem comprimida resultante, formada a partir dos 32 primeiros auto-vetores, associados aos 32 auto-valores mais significativos; isto é, a RNA utilizada para treinamento pelo GHAPCA para extração dos componentes principais da imagem era composta de 32 neurônios na camada de saída.

A Figura 6.14 mostra a imagem comprimida resultante, formada a partir dos 16 primeiros auto-vetores, associados aos 16 auto-valores mais significativos; caso em que a RNA apresentava 16 neurônios na camada de saída.

Figura 6.12: Imagem *grayscale* "Lenna" de tamanho 128×128 pixels.

Figura 6.13: Imagem da Figura 6.12 comprimida através do GHAPCA, considerando $k = 32\,(k < m)$ neurônios na camada de saída da RNA. O coeficiente de compressão resultante é $\rho = 0.632812$. A fidelidade da imagem comprimida com relação à original é dada por $\mathrm{PSNR} = 37.3\,\mathrm{dB}$.

Figura 6.14: Imagem da Figura 6.12 comprimida através do GHAPCA, considerando $k = 16\,(k < m)$ neurônios na camada de saída da RNA. O coeficiente de compressão resultante é $\rho = 0.316406$. A fidelidade da imagem comprimida com relação à original é dada por $\mathrm{PSNR} = 31.4\,\mathrm{dB}$.

6.3.3.9 Sumário do *GHAPCA* Aplicado à Compressão de Imagens Digitais

I – <u>Inicialização</u>:

1. A imagem é normalizada para que o maior valor de pixel seja 1 e o menor valor de pixel seja zero.
2. A imagem é particionada em submatrizes não sobrepostas, de dimensões $l \times l$, onde $l = 2^n$ elementos (n é uma potência inteira de dois e é determinado de acordo com as características estatísticas da imagem). Os dados são, portanto, apresentados à RNA sob a forma de vetores unidimensionais de p elementos, $p = l^2$. As imagens utilizadas neste trabalho possuem 128 $\times$ 128 elementos.
3. Determina-se o vetor média do conjunto de vetores unidimensionais.

4. Subtrai-se o vetor média do conjunto de vetores, formando o conjunto de treino a ser apresentado à RNA.
5. Inicializa-se os pesos sinápticos de maneira aleatória (distribuição de probabilidade uniforme) com valores pertencentes ao intervalo $\left[-\frac{r}{p}, \frac{r}{p}\right]$, sendo p o número de pesos sinápticos por neurônio e r um parâmetro a ser experimentalmente estipulado, de acordo com o conjunto de treino da RNA.
6. Arbitra-se uma razão de aprendizado inicial (usualmente no intervalo $[1\times10^{-9},\ 5\times10^{-9}]$).

II - <u>Treinamento</u>:

1. Apresenta-se o conjunto de treino à RNA.
2. A cada vetor apresentado, atualiza-se os pesos sinápticos de acordo com a regra de aprendizado do *GHAPCA*.
3. Procede-se desta maneira até que todos os vetores de treino sejam apresentados à RNA, o que constitui uma época de treino.
4. Ao final de cada época embaralha-se o conjunto de vetores de treino de maneira aleatória (distribuição de probabilidade uniforme).
5. Determina-se o auto-valor do neurônio que está convergindo, associado ao maior auto-valor na janela de treino.
6. Determina-se a nova razão de aprendizado proporcionalmente ao inverso do auto-valor (Equação 6.104).
7 Atualiza-se a razão de aprendizado deste neurônio e dos próximos incluídos na janela de treino com a nova razão de aprendizado.
8. Os passos (5) a (7) são repetidos até a convergência do neurônio associado ao maior auto-valor na janela de treino, após o que, a janela de treino é deslocada um neurônio para a frente.
9. Repete-se os passos (5) a (8) até que todos os neurônios convirjam.
10. Reconstrói-se a imagem a partir dos componentes estimados pela RNA (auto-vetores e saídas dos neurônios da RNA).
11. Soma-se a todos os vetores reconstruídos o vetor média previamente subtraído.
12. A imagem é desnormalizada para que o maior valor de pixel seja 255 e o menor valor de pixel seja zero.

6.4 Referências Bibliográficas do Capítulo 6:

[1] E. Oja and J. Karhunen. "On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix". *Journal of Mathematical Analysis and Applications*, 106:69-84, 1985.

[2] E. Oja. "Principal Components, Minor Components and Linear Neural Networks". *Neural Networks*, 5:927-935,1992.

[3] M. C. F. De Castro, F. C. C. De Castro, J. N. Amaral and P. R. G. Franco, "A Complex Valued Hebbian Learning Algorithm", *1998 IEEE World Congress on Computational Intelligence, International Joint Conference on Neural Networks,* pp. 1235-1238, Anchorage, Alaska, May 1998.

[4] M. C. F. De Castro, F. C. C. De Castro, J. N. Amaral and P. R. G. Franco, "A New Training Algorithm to Reduce the Computational Complexity of Principal Component Analysis by Hebbian Learning", *III Congresso Brasileiro de Redes Neurais,* pp. 7-11, Florianópolis, SC, Brazil, July 1997.

[5] Maria Cristina Felippetto De Castro, Fernando César C. De Castro, José Nelson Amaral e Paulo Roberto G. Franco, "Uma formulação complexa para o algoritmo Hebbiano generalizado aplicada à compressão de imagens." III Simpósio Brasileiro de Redes Neurais, pp. 55-62 , Recife, PE, Brazil, November 1996.

[6] M. C. F. de Castro. "Algoritmo Hebbiano Generalizado para Extração dos Componentes Principais de um Conjunto de Dados no Domínio Complexo". Tese de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, RS, Brasil, 1996.

[7] S. Haykin, *Neural Networks*, 2nd ed., Prentice Hall, Upper Saddle River, New Jersey, 1999.

[8] C. T. Chen, *Linear System Theory and Design*, Harcourt Brace College Publishers, 1984.

[9] F. R. Gantmacher, *The Theory of Matrices*, vol.1, Chelsea Publishing Company, New York, NY, 1977.

[10] R. Bronson, *Matrix Methods: An Introdution*, Academic Press Inc., San Diego, CA, 1991.

[11] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, 1995.

[12] S. Bannour e M. R. Azimi-Sadjadi, "Principal Component Extraction Using Recursive Least Squares Learning", *IEEE Transactions on Neural Networks*, vol.6, n° 2, 1995.

[13] L. H. Chen e S. Chang, "An Adaptive Learning Algorithm for Principal Component Analysis", *IEEE Transactions on Neural Networks*, vol.6, n° 5, 1995.

[14] F. Ayres Jr., *Theory and Problems of Matrices*, Schaum's Outline Series in Mathematics, McGraw-Hill Book Company, 1962.

[15] J. Hertz, A. Krogh e R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991.

[16] K. S. Shanmugan e A. M. Breipohl, *Random Signals: Detection, Estimation and Data Analysis*, John Wiley & Sons, 1988.

[17] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[18] R. Gonzales e P. Wintz, *Digital Image Processing*, Addison-Wesley, 1987.