Estimação do Número de Neurônios Ocultos da Rede MLP Usando Kernel PCA

Jackson U. Ponte and Guilherme A. Barreto
Programa de Pós-graduação em Engenharia de Teleinformática (PPETI)
Universidade Federal do Ceará (UFC)
Centro de Tecnologia, Campus do Pici, Fortaleza-CE
Email: jackson.uchoa@gmail.com, gbarreto@ufc.br

***Resumo*—A rede perceptron multicamadas (MLP, *multilayer perceptron*) é uma importante arquitetura clássica de redes neurais artificiais, que encontra aplicação em diversos problemas complexos de classificação de padrões e aproximação de funções. Apesar do seu amplo uso, sabe-se que o desempenho dessa rede é fortemente dependente do número de neurônios ocultos escolhido, sendo a estimação deste hiperparâmetro responsável por boa parte do tempo gasto no projeto de arquiteturas baseadas na rede MLP. Isto posto, neste artigo introduzimos uma nova técnica para estimar de forma rápida o número de neurônios ocultos da rede MLP usando KPCA (*kernel principal componentes analysis*). Esta técnica é aplicada a três conjuntos de variáveis de estado, a saber, (i) saídas dos neurônios ocultos, (ii) erros retropropagados, e (iii) gradientes locais dos erros retropropagados, com o objetivo de reduzir o nível de redundância da informação carregada por estas variáveis. Uma avaliação comparativa abrangente do método proposto usando quatro conjuntos de dados reais e um conjunto artificial é levada a cabo neste artigo tendo como alvo problemas de classificação de de padrões. Os resultados iniciais ora reportados indicam claramente um desempenho superior da técnica proposta em comparação a uma versão anteriormente proposta que usa técnicas lineares.**

***Keywords*—Estimação de modelo; classificação de padrões; métodos de kernel; análise de componentes principais; rede MLP.**

I. Introdução

A rede MLP e o algoritmo de retropropagação do erro ainda desempenham relevante papel na área de aprendizado de máquinas mesmo após mais 30 anos de sua introdução [7], seja na forma de um classificador com umas poucas camadas ocultas, seja integrada em arquiteturas de aprendizado profundo [8]. Porém, apesar dessa importância, derivada em grande parte da propriedade de aproximação universal [1], uma limitação dessa rede que ainda persiste em dias atuais, é a necessidade de especificação *a priori* (i.e. antes do treinamento) do número de neurônios da camada oculta. Embora haja um número considerável de técnicas que podem ajudar neste processo (ver [4] e referências), a especificação desse hiperparâmetro não é trivial e exige muita experimentação com o conjunto de dados.

Um desses métodos, proposto por Santos *et al* [2], tem por base à aplicação de PCA a três conjuntos de variáveis de estado, a saber, (i) saídas dos neurônios ocultos, (ii) erros retropropagados, e (iii) gradientes locais dos erros retropropagados, a fim de reduzir o nível de redundância da informação carregada por estas variáveis. Aspectos positivos desta técnica residem em sua linearidade e na consequente facilidade de aplicação, bastando para isso montar as matrizes associadas às variáveis de estado supracitadas e, em seguida, estimar as matrizes de covariância correspondentes. O número de neurônios ocultos é então definido como o número de autovalores mais relevantes (i.e. àqueles associados às componentes principais) daquelas matrizes.

Apesar de a técnica baseada em PCA ter sido aplicada com sucesso a vários problemas de classificação de padrões, sua melhor característica (i.e. linearidade) é justamente sua maior limitação. Sabe-se que as variáveis de estado mencionadas no parágrafo anterior resultam do processamento *não-linear* da informação ao longo das camadas successivas da rede MLP. Assim, a aplicação de PCA padrão a estas variáveis consiste em uma aproximação, uma vez que correlações não-lineares entre as variáveis de estado de interesse podem não ser capturadas em sua plenitude. Em suma, pode haver redundância oriunda de relações não-lineares entre as variáveis de estado e estas não serem capturdas adequadamente pela técnica linear.

Isto posto, neste artigo introduzimos uma nova metodologia para estimação do número de neurônios ocultos de redes MLP com uma camada oculta que consiste na aplicação de um tipo de PCA não-linear, conhecido como *kernel* PCA (KPCA) [9]. A metodologia proposta é muito similar àquela descrita em [2], com a troca das matrizes de covariância das variáveis de estado pelas matrizes de kernel correspondentes. A hipótese de trabalho deste artigo é que o uso de KPCA leva a uma estimativa mais fiel do número adequado de neurônios ocultos da rede MLP em problemas de classificação de padrões, pois é capaz de diminuir a redundância não-linear resultante da representação dos dados de entrada pelas variáveis de estado.

A fim de testar tal hipótese, uma avaliação comparativa abrangente do método proposto usando quatro conjuntos de dados reais e um conjunto artificial é levada a cabo neste artigo tendo como alvo problemas de classificação de de padrões. Os resultados iniciais ora reportados indicam um desempenho superior da técnica proposta em comparação com o método baseado em PCA. Por desempenho superior entende-se àquele que produz a maior acurácia usando o menor número de neurônios ocultos possível.

O restante do artigo está organizado nas seguintes seções. Na Seção II são apresentados os fundamentos teóricos das técnicas PCA clássica (linear) e da técnica KPCA (não-linear). A metodologia proposta, que pode ser entendida como uma

extensão não-linear daquela proposta em [2], é apresentada na Seção III. Os resultados dos experimentos computacionais serão apresentados e discutidos na Seção IV. O artigo é concluído na Seção V.

II. Detalhamento técnico

Seja uma matriz de dados $\mathbf{X} \in \mathbb{R}^{p \times N}$ cujas N colunas correspondem aos vetores de atributos $\mathbf{x}_i \in \mathbb{R}^p$. Assim, a dimensão da matriz $\mathbf{X}$ é $(p \times N)$, podendo esta ser escrita como

$$\mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \dots | \mathbf{x}_N], \tag{1}$$

em que cada coluna $\mathbf{x}_i = [x_1, \dots, x_p]^T$. Este conjunto será considerado no desenvolvimento teórico para o PCA e para o KPCA.

A. Análise de componentes principais (PCA)

A matriz de covariância de $\mathbf{X}$ é definida pela seguinte expressão:

$$\mathbf{C} = E[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T] = \mathbf{R} - \boldsymbol{\mu}\boldsymbol{\mu}^T, \tag{2}$$

em que $E[\cdot]$ é o operador valor esperado, $\boldsymbol{\mu} = E[\mathbf{x}]$ é o vetor de médias e $\mathbf{R} = E[\mathbf{x}\mathbf{x}^T]$ é a matriz de correlação. Estimativas de máxima verossimilhança da matriz $\mathbf{C}$ podem ser obtida por meio das seguintes expressões:

$$\hat{\mathbf{C}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T, \tag{3}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T - \bar{\mathbf{x}}\bar{\mathbf{x}}^T = \hat{\mathbf{R}} - \bar{\mathbf{x}}\bar{\mathbf{x}}^T, \tag{4}$$

em que $\bar{\mathbf{x}}$ é o vetor médio (centróide) amostral, enquanto $\hat{\mathbf{R}}$ é a matriz de correlação amostral.

A análise de componentes principais (PCA) consiste na projeção do conjunto de dados $\mathbf{X}$ em uma nova base vetorial, de forma que as projeções estejam nas direções de maior variância no conjunto de dados original. Em outras palavras, as coordenadas do novo sistema representam as direções de maior variabilidade dos dados. Considere a seguinte matriz de transformação $\mathbf{S} \in \mathbb{R}^{p \times q}$, em que $q \leq p$, tal que:

$$\mathbf{Y} = \mathbf{S}_q^T\mathbf{X} \tag{5}$$

As q colunas da matriz de transformação $\mathbf{S}_q$ são formadas pelas componentes principais, ou seja, pelos q autovetores normalizados da matriz de covariância $\mathbf{C}$ e ordenadas de acordo com a ordem de magnitude dos autovalores associados, os quais devem ser organizados em ordem decrescente. A matriz de transformação pode ser escrita da seguinte forma:

$$\mathbf{S}_q = [\mathbf{v}_1 \mid \dots \mid \mathbf{v}_q], \tag{6}$$

em que $\{\mathbf{v}_i\}_{i=1}^q$ são os q primeiros autovetores da matriz de covariância dispostos ao longo das colunas da matriz $\mathbf{S}_q$.

Os autovalores e autovetores da matriz de covariância $\mathbf{C}$ são calculados obtidos a partir da aplicação sucessiva da seguinte equação:

$$\lambda_i\mathbf{v}_i = \mathbf{C}\mathbf{v}_i. \tag{7}$$

A quantidade q de componentes principais a ser escolhida para um dado problema está associada com a ordem de grandeza dos autovalores e a quantidade de informação que queremos preservar na transformação. Se usarmos todos os p autovetores, toda a informação presente no conjunto de dados original será preservada. Se usarmos menos componentes (i.e. q), a informação preservada por estas q componentes é dada pela seguinte expressão:

$$VE(q) = \frac{\sum_{i=1}^{q}\lambda_i}{\sum_{l=1}^{p}\lambda_l}, \tag{8}$$

também chamada de *variância explicada* pelas q primeiras componentes principais. Assim, o número de componentes pode ser definido em função de um valor mínimo, aqui denotado de γ, que admitimos para a quantidade de informação (i.e. variância) a ser preservada. Matematicamente, esta idéia pode ser traduzida por meio da seguinte expressão:

$$q = \arg\min_{q=1,\dots,p}\{VE(q) \geq \gamma\}. \tag{9}$$

Em suma, a expressão (9) sintetiza um processo de busca pelo menor número de componentes principais que satisfaz o critério de tolerância representado por γ.

B. Análise de Componentes Principais Kernalizada (KPCA)

O desenvolvimento teórico da técnica KPCA foi proposto por [9]. Considere o seguinte mapeamento não-linear de cada vetor coluna $\mathbf{x}_i$ do conjunto de dados $\mathbf{X}$:

$$\begin{aligned}\phi : \mathbf{R}^p &\longrightarrow \mathbf{F} \\ \mathbf{x}_i &\mapsto \phi(\mathbf{x}_i),\end{aligned} \tag{10}$$

em que $\mathbf{F}$ é referenciado como espaço de características, geralmente um espaço de alta dimensão, tal como um espaço de Hilbert reproduzido por kernel (*reproducing kernel Hilbert space*, RKHS).

A técnica KPCA pode ser entendida como uma generalização do PCA, sendo aplicada a casos em que se está interessado nas componentes principais no espaço de características, que é não-linearmente relacionado com as variáveis de entrada originais. Assumindo que os vetores $\{\phi(\mathbf{x}_i)\}_{i=1}^N$ estão centralizados, i.e., $\sum_{i=1}^N \phi(\mathbf{x}_i) = \mathbf{0}$, podemos escrever a matriz de covariância $\bar{\mathbf{C}}$ no espaço de características da seguinte forma:

$$\bar{\mathbf{C}} = \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T. \tag{11}$$

O cálculo das componentes principais no espaço de características é feito a partir da decomposição espectral da matriz de covariância $\bar{\mathbf{C}}$. Dado um vetor $\bar{\mathbf{v}} \in \mathbf{F}$, podemos escrever a decomposição espectral:

$$\tilde{\lambda}\bar{\mathbf{v}} = \bar{\mathbf{C}}\bar{\mathbf{v}} \tag{12}$$

Lembrando que todas as soluções de $\bar{\mathbf{v}}$ são combinações lineares de $\{\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_N)\}$. Portanto, existem coeficientes α_i $(i = 1, 2, \dots, N)$ tal que

$$\bar{\mathbf{v}} = \sum_{i=1}^{N}\alpha_i\phi(\mathbf{x}_i). \tag{13}$$

Considere agora a expressão abaixo como sendo equivalente àquela da Eq. (12):

$$\tilde{\lambda}\left[\phi(\mathbf{x}_k)\cdot\bar{\mathbf{v}}\right]=\left[\phi(\mathbf{x}_k)\cdot\bar{\mathbf{C}}\bar{\mathbf{v}}\right], \tag{14}$$

para todo $k=1,\ldots,N$. Assim, substituindo $\bar{\mathbf{v}}$ e $\bar{\mathbf{C}}$ das Eqs. (11) e (13) na Eq. (14), obtemos

$$\tilde{\lambda}\sum_{i=1}^{N}\alpha_i(\phi(\mathbf{x}_k)\phi(\mathbf{x}_i))=$$

$$\frac{1}{N}\sum_{i=1}^{N}\alpha_i((\phi(\mathbf{x}_k))\sum_{j=1}^{N}\phi(\mathbf{x}_j))(\phi(\mathbf{x}_j))\phi(\mathbf{x}_i)), \tag{15}$$

para todo $k=1,\ldots,N$. Definindo uma matriz $\mathbf{K}$ $(N\times N)$, que chamaremos de matriz de kernel, como

$$\mathbf{K}_{ij}:=\phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j)\equiv\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) \tag{16}$$

temos então

$$N\tilde{\lambda}\mathbf{K}\boldsymbol{\alpha}=\mathbf{K}^2\boldsymbol{\alpha}, \tag{17}$$

em que $\boldsymbol{\alpha}=[\alpha_1\ \cdots\ \alpha_N]^T$ denota um vetor coluna. De forma equivalente, temos

$$N\tilde{\lambda}\boldsymbol{\alpha}=\mathbf{K}\boldsymbol{\alpha}. \tag{18}$$

Considere que $(\lambda_1>\ldots>\lambda_N)$ represente os autovalores da matriz de kernel $\mathbf{K}$; ou seja

$$\lambda_j=N\tilde{\lambda}_j, \tag{19}$$

com $j=1,\ldots,N$. Em que $\tilde{\lambda}_j$ é o j-ésimo autovalor da matriz de correlação $\bar{\mathbf{C}}\in\mathbf{F}$. Então a Eq. (18) toma a forma padrão dada por

$$\lambda\boldsymbol{\alpha}=\mathbf{K}\boldsymbol{\alpha}, \tag{20}$$

em que $\boldsymbol{\alpha}$ desempenha papel de autovetor associado ao autovalor λ da matriz de kernel $\mathbf{K}$.

De acordo com [9], a exigência de que os autovetores da matriz de covariância $\bar{\mathbf{C}}$ devem ser normalizados é colocada como

$$(\bar{\mathbf{v}}^k\cdot\bar{\mathbf{v}}^k)=1,\forall k=p,\ldots,N \tag{21}$$

em que p corresponde ao primeiro autovalor não nulo de $\bar{\mathbf{C}}$, os quais previamente são organizados em ordem crescente. A condição dada pela Eq. (21) se traduz na seguinte normalização para os autovetores $\boldsymbol{\alpha}^k$:

$$(\boldsymbol{\alpha}^k\cdot\boldsymbol{\alpha}^k)=\frac{1}{\lambda_k} \tag{22}$$

com $k=1,\ldots,N$. Para mais detalhes consulte [9].

De acordo com [11], podemos extrair as primeiras q $(1\leq q\leq N)$ componentes principais não-lineares (i.e., autovetores da matriz $\mathbf{K}$) sem a operação custosa de explicitamente projetarmos as amostras no espaço $\mathbf{F}$. De acordo com [9], os autovalores de $\mathbf{K}$ darão exatamente a solução de $N\tilde{\lambda}$ da Eq. (18).

De modo similar à versão linear, a quantidade de componentes principais não-lineares q está associada com a ordem de grandeza dos autovalores e a uma tolerância que chamamos de γ , de acordo com a seguinte equação:

$$q=\arg\min_{q=1,\ldots,N}\{VE(q)\geq\gamma\} \tag{23}$$

Assim, a implementação do método KPCA funciona de modo semelhante à sua versão linear, trocando-se a matriz de covariância estimada pela Eq. (3) ou pela Eq. ((4)) por uma matriz de kernel. Um ponto importante a ser ressaltado é que no caso da técnica PCA linear, o número total de autovetores (i.e. componentes principais) é igual à dimensão do vetor de atributos (p), enquanto no caso da versão não-linear, o número de autovetores é igual ao tamanho do conjunto de dados de treinamento (N).

C. Matriz de Kernel

Podemos extrair a matriz de kernel diretamente do conjunto de dados de treinamento $\mathbf{X}$ dado pela Eq. (1). Uma matriz Gram é definida como uma matriz $(N\times N)$ em que entradas são produtos internos entre os vetores colunas que compõem a matriz $\mathbf{X}$, na forma $\mathbf{G}_{ij}=(\mathbf{x}_i\cdot\mathbf{x}_j)$. Se usarmos uma **função de kernel** $k(\cdot,\cdot)$ para avaliar os produtos internos no espaço de características mapeado por ϕ teremos a matriz Gram associada, chamada de matriz de kernel $\mathbf{K}$:

$$\mathbf{G}_{ij}=(\phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j))=k(\mathbf{x}_i,\mathbf{x}_j) \tag{24}$$

portanto

$$\mathbf{K}_{ij}=k(\mathbf{x}_i,\mathbf{x}_j), \tag{25}$$

para $i,j=1,\ldots,N$.

1) Função de kernel: Uma função de kernel tem por objetivo calcular o produto interno no espaço de características. Apresentamos a seguir algumas funções de kernel que testamos na metodologia apresentada neste artigo.

- Função de kernel gaussiana:

$$k(\mathbf{x}_i,\mathbf{x}_j)=\exp\left(-\frac{\parallel\mathbf{x}_i-\mathbf{x}_j\parallel^2}{2\sigma^2}\right) \tag{26}$$

  para $i,j=1,\ldots,N$, em que o raio σ é um hiperparâmetro a ser definido.
- Função kernel sigmoidal:

$$k(\mathbf{x}_i,\mathbf{x}_j)=\tanh(\mathbf{x}_i\cdot\mathbf{x}_j+b) \tag{27}$$

  para $i,j=1,\ldots,N$, em que o limiar b é um hiperparâmetro da função.
- Função kernel log:

$$k(\mathbf{x}_i,\mathbf{x}_j)=-\log\left(1+\frac{\parallel\mathbf{x}_i-\mathbf{x}_j\parallel}{\sigma^2}\right) \tag{28}$$

  para $i,j=1,\ldots,N$, em que σ é um hiperparâmetro a ser definido.

III. Metodologia Proposta

Considere uma rede MLP (já treinada) com p unidades de entrada, um elevado número de q neurônios na camada oculta e M neurônios na camada de saída. A saída do neurônio oculto i na iteração t é dada por

$$y_i^{(h)}(t) = \varphi\left[u_i^{(h)}(t)\right] = \varphi\left[\sum_{j=0}^{p} w_{ij}x_j(t)\right], \qquad (29)$$

em que w_{ij} é o peso sináptico entre a unidade de entrada j e o neurônio i da camada oculta, p é a dimensão dos vetores de entrada (excluindo o limir) e $\varphi(.)$ é uma função de ativação sigmoidal. Seja $\tilde{\boldsymbol{W}} \in \mathbb{R}^{q\times(p+1)}$ a matriz de pesos formada por todos os pesos sinápticos entre os elementos de entrada e os neurônios ocultos, representado por

$$\tilde{\boldsymbol{W}} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_q^T \end{bmatrix}, \qquad (30)$$

em que cada linha $\mathbf{w}_i^T = [w_{i0}\ w_{i1}\ \cdots\ w_{iq}]$ corresponde aos pesos sinápticos de $p+1$ unidades de entrada, incluindo o limiar, para o i-ésimo neurônio oculto.

Similarmente, seja $\mathbf{X} \in \mathbb{R}^{(p+1)\times N}$ a matriz cujas colunas são compostas por N vetores de entradas usadas no treinamento da rede:

$$\mathbf{X} = [\mathbf{x}_1|\mathbf{x}_2|\ldots|\mathbf{x}_N], \qquad (31)$$

em que cada vetor-coluna $\mathbf{x}_i = [x_0\ x_1\ \cdots\ x_p]^T$ corresponde ao vetor de atributos na iteração t, com $x_0 = -1$. Portanto, a Eq. (29) pode ser agora representada na forma matricial como

$$\mathbf{Y}^{(h)} = \varphi\left[\tilde{\mathbf{W}}\mathbf{X}\right], \qquad (32)$$

em que $\varphi[.]$ é a função de ativação aplicada às componentes do resultado da multiplicação $\tilde{\mathbf{W}}\mathbf{X}$. A matriz $\mathbf{Y}^h \in \mathbb{R}^{q\times N}$ armazena as saídas dos q neurônios ocultos, calculados para todos os N exemplos do conjunto de treinamento da rede. A matriz $\mathbf{Y}^{(h)}$ pode ser representada na forma expandida como

$$\mathbf{Y}^{(h)} = \begin{bmatrix} y_1^h(1) & y_1^h(2) & \ldots & y_1^h(N) \\ y_2^h(1) & y_2^h(2) & \ldots & y_2^h(N) \\ \vdots & \vdots & \vdots & \vdots \\ y_q^h(1) & y_q^h(2) & \ldots & y_q^h(N) \end{bmatrix} \qquad (33)$$

A fase de retropropagação dos erros começa a partir da camada de saída, pela projeção dos erros $e_k^{(h)}(t) = d_k(t) - y_k^{(o)}(t)$ para a camada oculta, em que $d_k(t)$ e $y_k^{(o)}(t)$ são, respectivamente, as saídas-alvo e as respostas de saída do k-ésimo neurônio de saída. Seja o erro retropropagado para o i-ésimo neurônio oculto definido como

$$e_k^{(h)}(t) = \sum_{k=1}^{M} m_{ki}\delta_k^{(o)}(t), \quad i = 0,\ldots,q, \qquad (34)$$

em que m_{ki} é o peso sináptico que conecta o i-ésimo neurônio oculto ao k-ésimo neurônio de saída. O termo $\delta_k^{(o)}(t) = \varphi_k'(t)e_k^{(o)}(t)$ é o gradiente local do k-ésimo neurônio de saída. O termo $\varphi_k'(t)$ é a derivada da função de ativação do k-ésimo neurônio de saída.

Seja $\tilde{\mathbf{M}} \in \mathbb{R}^{M\times(q+1)}$ a matriz de pesos entre a camada oculta e a camada de saída dos neurônios:

$$\tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \vdots \\ \mathbf{m}_M^T \end{bmatrix}, \qquad (35)$$

em que cada linha $\mathbf{m}_k^T = [m_{k0}\ m_{k1}\ \cdots\ m_{kq}]$ corresponde ao vetor de pesos associado ao k-ésimo neurônio de saída, incluindo o seu limiar.

Considere agora $\boldsymbol{\Delta}^{(o)} \in \mathbb{R}^{M\times M}$ como a matriz cujas N colunas são formadas pelos gradientes locais dos M neurônios de saída, construída a partir dos N exemplos de treinamento disponíveis:

$$\boldsymbol{\Delta}^{(o)} = [\boldsymbol{\delta}^{(o)}(1)|\boldsymbol{\delta}^{(o)}(2)|\ldots|\boldsymbol{\delta}^{(o)}(N)], \qquad (36)$$

em que cada vetor-coluna $\boldsymbol{\delta}^{(o)}(t) = [\delta_1^{(o)}(t)\ \ \delta_2^{(o)}(t)\ \cdots\ \delta_M^{(o)}(t)]^T$. Portanto, a Eq. (34) pode agora ser representada em forma matricial como

$$\mathbf{E}^{(h)} = \tilde{\mathbf{M}}^T\boldsymbol{\Delta}^{(o)} \qquad (37)$$

em que a matriz $\mathbf{E}^{(h)} \in \mathbb{R}^{(Q+1)\times N}$ armazena em suas colunas o erro retropropagado associado com os neurônios da camada oculta, para todo o conjunto de treinamento. A matriz $\mathbf{E}^{(h)}$ em sua forma expandida é expressa por

$$\mathbf{E}^{(h)} = \begin{bmatrix} e_0^{(h)}(1) & e_0^{(h)}(2) & \ldots & e_0^{(h)}(N) \\ e_1^{(h)}(1) & e_1^{(h)}(2) & \ldots & e_1^{(h)}(N) \\ \vdots & \vdots & \vdots & \vdots \\ e_Q^{(h)}(1) & e_Q^{(h)}(2) & \ldots & e_Q^{(h)}(N) \end{bmatrix} \qquad (38)$$

A primeira coluna de $E^{(h)}$ corresponde aos erros retropropagados associados com o limiar $m_{k0} = \theta_k^{(o)}, \forall k = 1,\ldots,M$. A primeira linha de $E^{(h)}$ não é relevante e, portanto, pode ser removida. Os gradientes locais dos neurônios ocultos são definidos como

$$\delta_i^{(h)} = \varphi_i'(t)\sum_{k=1}^{M} m_{ki}\delta_k^{(o)}(t) = \varphi_i'(t)e_i^{(h)}(t), \ \ i = 0,\ldots,q \qquad (39)$$

em que $\varphi_i'(t)$ é a derivada da função de ativação do i-ésimo neurônio oculto. Seja $\boldsymbol{\Phi}^{(h)} \in \mathbb{R}^{(q\times N)}$ a matriz formada por todas essas derivadas para N exemplos de treinamento, ou seja

$$\boldsymbol{\Phi}^{(h)} = [\boldsymbol{\varphi}'(1)|\boldsymbol{\varphi}'(2)|\ldots|\boldsymbol{\varphi}'(N)] \qquad (40)$$

em que $\boldsymbol{\varphi}'(t) = [\varphi_1'(t)\varphi_2'(t)\ldots\varphi_i'(t)\ldots\varphi_q'(t)]^T$. Portanto, a Eq. (39) pode ser escrita na forma matricial como

$$\boldsymbol{\Delta}^{(h)} = \boldsymbol{\Phi}^{(h)} \odot \mathbf{E}^{(h)} \qquad (41)$$

em que $\odot$ denota o operador de multiplicação de Hadamard, ou seja, $[\mathbf{A}\odot\mathbf{B}]_{i,j} = [\mathbf{A}]_{i,j}[\mathbf{B}]_{i,j}$. Portanto, a matriz $\boldsymbol{\Delta}^{(h)} \in \mathbb{R}^{(q\times N)}$ é composta pelos gradientes locais dos neurônios

ocultos para todo o conjunto de treinamento. Em sua forma expandida, $\boldsymbol{\Delta}^{(h)}$ pode ser escrita como:

$$\boldsymbol{\Delta}^{(h)} = \begin{bmatrix} \delta_1^{(h)}(1) & \delta_1^{(h)}(2) & \dots & \delta_1^{(h)}(N) \\ \delta_2^{(h)}(1) & \delta_2^{(h)}(2) & \dots & \delta_2^{(h)}(N) \\ \vdots & \vdots & \vdots & \vdots \\ \delta_q^{(h)}(1) & \delta_q^{(h)}(2) & \dots & \delta_q^{(h)}(N) \end{bmatrix} \quad (42)$$

A. Estimação do número de neurônios ocultos

Os métodos abordados neste artigo assumem que a rede MLP(p, q, M) é totalmente conectada. Isto posto, a rede MLP é inicialmente treinada com um número superestimado de neurônios da camada oculta q, a fim de se montar as matrizes que serão usadas para estimação da quantidade adequada de neurônios desta camada.

Logo após a fase de treinamento da rede MLP, os dados de entrada são reapresentados com o objetivo de se construir as matrizes $\mathbf{Y}^{(h)}, \mathbf{E}^{(h)}$ e $\boldsymbol{\Delta}^{(h)}$. Essas matrizes serão usadas como dados de entrada para os métodos aqui abordados.

1) Métodos lineares baseado em PCA: São métodos de estimação lineares, discutidos e propostos em [2], usados na estimação do número de neurônios ocultos da rede MLP em problemas de classificação de padrões. Estes consistem na determinação dos autovalores das matrizes de covariância dos dados de entrada, ordenando-os em ordem decrescente para então se determinar o número adequado de neurônios ocultos de acordo com a expressão em (9). A regra é bem simples: o número q assim obtido corresponderá ao número de neurônios ocultos a ser utilizado. Este processo é repetido por K vezes, sendo o número de neurônios ocultos dado pelo valor de q com maior frequência de ocorrência ao longo das K repetições. Tendo o valor de q sido definido, a rede MLP deve ser treinada e testada novamente, a fim de se validar a nova topologia.

A idéia subjacente à associação entre o número de neurônios ocultos e as primeiras q componentes principais das matrizes de covariância de $\mathbf{Y}^{(h)}, \mathbf{E}^{(h)}$ e $\boldsymbol{\Delta}^{(h)}$ está ligada ao conceito de *máxima transferência de informação*. A matriz de dados original $\mathbf{X}$ contém em si uma quantidade de informação que deve ser preservada o tanto quanto possível ao ser processada pelas camadas sucessivas da rede MLP. Quando se usa um número elevado de neurônios ocultos, teremos redundância na representação do problema de interesse, o que pode levar a um fraco desempenho da rede.

Como a técnica PCA pode ser entendida como uma técnica de compressão de informação, podemos fazer a seguinte indagação: qual seria a quantidade adequada de neurônios ocultos de modo a se preservar a informação contida nas matrizes $\mathbf{Y}^{(h)}, \mathbf{E}^{(h)}$ e $\boldsymbol{\Delta}^{(h)}$ sem comprometer o desempenho do classificador? Em suma, queremos maximizar a transferência de informação ao longo das camadas da rede MLP, reduzindo redundância, mas sem comprometer o desempenho da rede.

2) Métodos não-lineares baseado em KPCA: A proposta deste artigo consiste em usar KPCA em vez de PCA linear para estimar o número adequado de neurônios ocultos da rede MLP. A justificativa para tal abordagem reside na percepção de que as matrizes $\mathbf{Y}^{(h)}, \mathbf{E}^{(h)}$ e $\boldsymbol{\Delta}^{(h)}$ são em essência resultantes de operações não-lineares via funções de ativação sigmoidais.

Deste modo, o uso de PCA linear é uma aproximação, de modo que correlações não-lineares entre as variáveis de estado de interesse, ou seja, (i) saídas dos neurônios ocultos, (ii) erros retropropagados, e (iii) gradientes locais dos erros retropropagados, podem não ser capturadas em sua plenitude. Em suma, pode haver redundância oriunda de relações não-lineares entre as variáveis de estado. Nossa hipótese é que a técnica KPCA é capaz de capturar tais relações, melhorando ainda mais a estimativa do número de neurônios ocultos.

A metodologia proposta é muito similar àquelas propostas em em [2], com a troca das matrizes de covariância de $\mathbf{Y}^{(h)}, \mathbf{E}^{(h)}$ e $\boldsymbol{\Delta}^{(h)}$, pelas matrizes de kernel correspondentes. Determinam-se então os autovalores dessas matrizes de kernel, ordenando-os em ordem decrescente para, então se escolher os q maiores autovalores segundo o critério estabelecido na expressão (9). Este processo é repetido K vezes, sendo o número final de neurônios ocultos aquele que ocorre com maior frequência ao longo das K repetições de treinamento. Uma vez definido o valor final de q, a rede MLP deve ser treinada e testada novamente, a fim de se validar a nova topologia.

A função de kernel escolhida para os experimentos reportados neste artigo foi a função kernel log, definida na Eq. (28), por apresentar maiores taxas de acerto médio em uma comparação preliminar usando a metodologia ora proposta. Outro fator importante nesta esoclha é o fato de esta função kernel ter se mostrado pouco sensível à variações no hiperparâmetro σ. Em resumo, para cada conjunto de dados aplicam-se 3 métodos lineares e 3 não-lineares; ou seja, PCA aplicada ás matrizes $\mathbf{Y}^{(h)}, \mathbf{E}^{(h)}$ e $\boldsymbol{\Delta}^{(h)}$ e KPCA aplicada às matrizes $\mathbf{Y}^{(h)}, \mathbf{E}^{(h)}$ e $\boldsymbol{\Delta}^{(h)}$.

IV. Resultados e Discussões

A rede MLP aqui descrita foi treinada usando o algoritmo *backpropagation* com a função de ativação tangente hiperbólica para todos os neurônios ocultos e de saída. Os procedimentos de treinamento e teste foram repetidos 100 vezes para cada conjunto de dados. Definiu-se o número de épocas de treinamento como 1500, pois esta quantidade garantiu a convergência da minimização do erro médio quadrático para os conjuntos testados.

Em cada rodada de treinamento/teste, os dados foram aleatoriamente selecionados na proporção de 80% para treinamento e 20% para testes. Os dados de entrada foram normalizados para média zero e variância unitária. Os pesos e limiares foram inicializados aleatoriamente no intervalo $[-0.1, 0.1]$. O passo de aprendizagem inicial foi definido como η_0 = 0,75, decaindo linearmente até zero. A tolerância mínima para a variância explicada - vide Eqs. (9) e (23) - foi definido como $\gamma = 95\%$. Para a técnica KPCA aplicou-se a função kernel log com o hiperparâmetro $\sigma = 10$.

Os resultados numéricos para as simulações estão apresentados na Tabela I, sendo N_c a quantidade total de pesos da

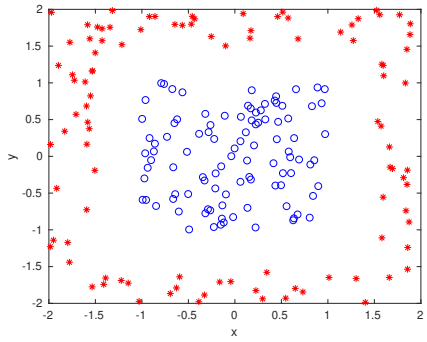

Figura 1. Distribuição espacial para o conjunto dados artificiais.

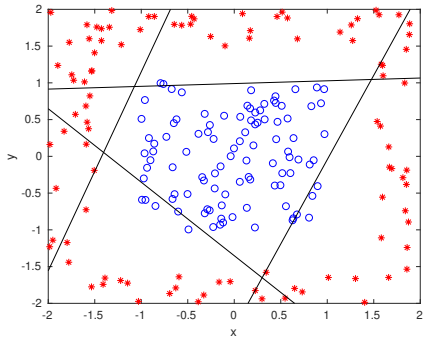

Figura 2. Visão em duas dimensões dos planos que compõem a superfície de decisão gerada pela rede MLP com 4 neurônios na camada oculta.

arquitetura, $T_m(\%)$ a taxa de acerto médio e ε_m a média das somas dos erros quadráticos médios dos dados de teste.

A seguir, são brevemente descritas as características de cada conjunto de dados que utilizamos nas simulações. Os conjuntos colunas vertebral, câncer de mama, ionosfera e abalone são gratuitamente disponibilizados em [1]*UCI Machine Learning Repository* da Universidade da Califórnia.

A. Dados Artificiais

Como o nome sugere, este conjunto foi gerado artificialmente por [3]. Esse conjunto é composto por dados bidimensionais divididos em duas classes não-linearmente separáveis. Foram gerados um total de 200 pontos, os quais foram igualmente distribuídos em duas classes. Para este conjunto, a arquitetura inicial da rede MLP foi definida como MLP(2,10,2). A Figura 1 ilustra como os dados artificiais estão distribuídos espacialmente.

A aplicação de PCA à matriz $\mathbf{E}^{(h)}$ resultou na sugestão de um único neurônio na camada oculta, porém, sabe-se que pela natureza não-linear dos dados, este número não resolve o problema. Já a aplicação da técnica KPCA à matriz $\mathbf{Y}^{(h)}$ resultou em uma indicação de que o problema poderia ser resolvido com 4 neurônios na camada oculta. Esse resultado é bastante coerente, pois as duas classes podem ser facilmente separadas pela combinação de 4 retas associadas aos 4 neurônios ocultos, conforme figura 2.

B. Conjunto Coluna Vertebral

Este conjunto de dados foi gerado pelo Cirugião de coluna, Dr. Henrique da Mota, durante residência médica no Grupo de Pesquisa Aplicada em Ortopedia (GARO) do Centre médico-chirurgical de réadaptation des Massues (CMRC des Massues), Lyon, França. A tarefa consiste em classificar vetores de atributos biomecânicos extraídos de tomografias da coluna lombar de pacientes em três categorias: normal (100 pacientes), hérnia de disco (60 pacientes) ou espondilolistese (150 pacientes). Para esse conjunto, definimos a arquitetura inicial como MLP(6,24,3). Mais detalhes sobre esse conjunto pode ser encontrado em [6].

[1]http://archive.ics.uci.edu/ml/index.php

A aplicação de KPCA na estimação do número de neurônios ocultos resultou em arquiteturas com taxas de acerto médio bem mais altas em comparação com as resultantes da aplicação de PCA. Destaca-se aqui o resultado obtido pela aplicação do método proposto à matriz $\mathbf{E}^{(h)}$, o qual produziu o classificador MLP(6,2,3) com taxa de acerto médio com desempenho de $85,29\%$. Tal resultado é superior ao desempenho de todos os classificadores gerados pela aplicação do método linear baseado em PCA.

C. Câncer de Mama

O conjunto câncer de mama *(Breast Cancer Database)* é proveniente do hospital da Universidade de Wisconsin. Este conjunto contém vetores de atributos biomédicos de pacientes, em que cada instância pertence a duas possíveis classes: benigno (458 pacientes) ou maligno (241 pacientes). Para este conjunto, definimos a arquitetura inicial como MLP(31,50,2).

Em duas das três aplicações possíveis de KPCA a esse conjunto houve o indicativo de que o problema poderia ser solucionado apenas $q = 1$ neurônio na camada oculta. A interpretação sugerida para este resultado é a de que há evidências de que o problema é linearmente separável. Este fato pode ser verificado pelas altas taxas de acerto obtidas, mesmo com apenas um neurônio oculto. Outro ponto digno de nota é que as taxas de acerto médio foram equivalentes para as abordagens com PCA e com KPCA. Um classificador perceptron simples aplicado a este conjunto de dados obteve taxas de acerto da mesma ordem de grandeza, confirmando a sugestão de que o problema pode ser tratado adequadamente por um classificador linear.

D. Ionosfera

O conjunto Ionosfera contém dados que foram coletados por um sistema instalado em Goose Bay, Labrador, Canadá. Este sistema consiste em um arranjo em fase de 16 antenas de alta frequência com uma potência total transmitida da ordem de 6,4 kilowatts. A arquitetura inicial para este problema foi definida como MLP(32,28,2).

O conjunto trata de um problema de classificação binária linearmente separável, para mais detalhes recomenda-se a consulta da referẽncia [10]. Para este conjunto, a melhor

arquitetura resultante da aplicação do método linear baseado em PCA possui $q = 13$ neurônios ocultos. Por outro lado, a aplicação da técnica proposta baseada em KPCA resultou em arquiteturas com mesmo desempenho, mas com um número bem inferior de neurônios ocultos ($q = 2$). Est resultado mostra a capacidade da técnica proposta em capturar correlações não-lineares que o método baseado em PCA não foi capaz de capturar, o que levou a técnica KPCA a sugerir arquiteturas mais compactas e eficientes.

E. Conjunto Abalone

Os experimentos finais envolveram um conjunto de dados de extrema complexidade, mesmo para classificadores não-lineares, porque o número de classes ($m = 23$) é bem elevado. O objetivo da tarefa de classificação é a estimação da faixa etária de abalones (gênero de moluscos gastrópodes) a partir de suas medidas físicas. Mais informações sobre o conjunto abalone pode ser encontrada em [5]. A arquitetura inicial para este problema foi definida como MLP(8,30,23).

A taxa de acerto média dos classificadores indicados pela aplicação da técnica proposta baseada em KPCA teve um desempenho ligeiramente superior que àqueles resultantes da aplicação da técnica baseada em PCA. Houve uma concordância de 100% entre o valor sugerido de $q = 3$ neurônios ocultos resultantes da aplicação da técnica KPCA às três matrizes de estado $\mathbf{Y}^{(h)}, \mathbf{E}^{(h)}$ e $\boldsymbol{\Delta}^{(h)}$.

Ainda sobre este último comentário, vale destacar que para todos os conjuntos de dados utilizados na avaliação da técnica proposta baseada em KPCA, esta apresentou maior estabilidade nos resultados do que a técnica linear baseada em PCA. Por maior estabilidade entende-se o fato de haver uma maior concordância entre os valores sugeridos de q pela técnica KPCA, quando aplicada às três matrizes de estado $\mathbf{Y}^{(h)}, \mathbf{E}^{(h)}$ e $\boldsymbol{\Delta}^{(h)}$, do que os valores sugeridos pela técnica PCA a estas mesmas matrizes. Do ponto de vista do usuário, esta estabilidade confere maior confiança à sugestão dada pelo método KPCA para a estimação do número de neurônios ocultos da rede MLP.

V. Conclusões

Dado que os métodos não-lineares obteve taxas de acerto médio de até 11,01% maior do que àquela obtida pelo método linear equivalente, podemos afirmar que o estimador do número de neurônios da camada oculta da rede MLP baseado em KPCA produz, em média, classificadores com desempenho melhores.

A maior parte dos classificadores estimados pelos métodos não-lineares apresentaram menores desvios padrões quando comparados com os classificadores estimados pelos métodos lineares, assim, podemos concluir que os métodos não-lineares são capazes de reduzir, em média, o desvio padrão.

Os métodos baseados em KPCA, quando aplicados em conjuntos não-linearmente separáveis, geram arquiteturas com desempenho superiores aos métodos lineares. Já quando aplicamos em conjuntos que são linearmente separáveis os classificadores apresentam desempenho equivalentes aos dos métodos lineares.

Os ganhos obtidos pelos métodos não-lineares são extremamente necessários quando trabalhamos com conjuntos mais sensíveis a erros, e. g., em problemas de classificação envolvendo diagnóstico médico.

Atualmente, a metodologia proposta neste artigo está sendo testada em problemas de regressão e em arquiteturas da rede MLP com várias camadas ocultas. Os primeiros resultados neste sentido tem se mostrado promissores.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Os autores também agradecem ao CNPq (Concessão no. 309451/2015-9) por apoiarem esta pesquisa.

Referências

[1] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[2] G. A. B. J. D. A. Santos and C. M. S. Medeiros. Estimating the number of hidden neurons of the MLP using singular value decomposition and principal components analysis: a novel approach. In *Proceedings of 11th Brazilian Symposium on Neural Networks*, pages 19–24, 2010.

[3] C. M. Medeiros and G. A. Barreto. Pruning the multilayer perceptron through the correlation of backpropagated errors. In *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*, pages 64–69. IEEE, 2007.

[4] C. M. S. Medeiros and G. A. Barreto. A novel weight pruning method for mlp classifiers based on the MAXCORE principle. *Neural Computing and Applications*, 22(1):71–84, 2013.

[5] W. J. Nash, T. L. Sellers, S. R. Talbot, A. J. Cawthorn, and W. B. Ford. The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. *Sea Fisheries Division, Technical Report*, 48, 1994.

[6] A. R. Rocha Neto and G. A. Barreto. On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis. *IEEE Latin America Transactions*, 7(4):487–496, 2009.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[8] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61(6088):85–117, 2015.

[9] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[10] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Tech. Dig*, vol. 10:262–266, 1989. in.

[11] M.-H. Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Fgr*, volume 2, page 215, 2002.

Tabela I
RESULTADOS DOS ESTIMADORES E SEUS RESPECTIVOS DESEMPENHOS APLICADOS EM PROBLEMAS DE CLASSIFICAÇÃO

Conjunto de Dados	Téc.	Matriz	Q,q	N_c	$T_m(\%)$	Desvio Padrão	ε_m
Arquitetura inicial	-	-	10	52	99.0500	2.9895	0.0004
	⌈	$\mathbf{Y}^{(h)}$	3	17	91.9000	3.9428	0.3302
	PCA	$\mathbf{E}^{(h)}$	1	7	69.9250	8.2607	0.8586
Dados Artificiais	⌊	$\boldsymbol{\Delta}^{(h)}$	3	17	91.2250	5.5448	0.5792
	⌈	$\mathbf{Y}^{(h)}$	4	22	96.3750	4.6788	0.0003
	KPCA	$\mathbf{E}^{(h)}$	2	12	81.0250	6.0103	0.6067
	⌊	$\boldsymbol{\Delta}^{(h)}$	2	12	80.4250	6.7799	0.6647
Arquitetura inicial	-	-	24	243	81.8710	4.5105	0.6303
	⌈	$\mathbf{Y}^{(h)}$	10	103	82.3710	4.0961	0.5850
	PCA	$\mathbf{E}^{(h)}$	1	13	77.5000	4.9825	0.6749
Coluna Vertebral	⌊	$\boldsymbol{\Delta}^{(h)}$	1	13	77.1935	5.6621	0.5943
	⌈	$\mathbf{Y}^{(h)}$	12	123	81.6774	4.2551	0.7016
	KPCA	$\mathbf{E}^{(h)}$	2	23	85.2903	3.5683	0.3941
	⌊	$\boldsymbol{\Delta}^{(h)}$	2	23	84.5000	4.6015	0.6376
Arquitetura inicial	-	-	50	1702	97.2105	1.4719	0.1741
	⌈	$\mathbf{Y}^{(h)}$	7	240	96.7456	1.4662	0.1205
	PCA	$\mathbf{E}^{(h)}$	1	36	92.8772	15.1772	0.0337
Câncer de Mama	⌊	$\boldsymbol{\Delta}^{(h)}$	2	70	96.8860	1.4345	0.1568
	⌈	$\mathbf{Y}^{(h)}$	9	308	96.5965	1.7689	0.2308
	KPCA	$\mathbf{E}^{(h)}$	1	36	95.6491	10.0232	0.0832
	⌊	$\boldsymbol{\Delta}^{(h)}$	1	36	97.0000	1.4709	0.0697
Arquitetura inicial	-	-	28	1010	89.8732	3.8905	0.1281
	⌈	$\mathbf{Y}^{(h)}$	13	470	89.7606	3.9577	0.1436
	PCA	$\mathbf{E}^{(h)}$	1	38	86.7606	4.1753	0.0551
Ionosfera	⌊	$\boldsymbol{\Delta}^{(h)}$	1	38	85.8451	3.6879	0.1189
	⌈	$\mathbf{Y}^{(h)}$	14	506	90.1831	3.4758	0.1099
	KPCA	$\mathbf{E}^{(h)}$	2	74	89.4507	3.5894	0.0984
	⌊	$\boldsymbol{\Delta}^{(h)}$	2	74	89.5070	3.4464	0.0984
Arquitetura inicial	-	-	30	363	56.0167	1.6131	0.2555
	⌈	$\mathbf{Y}^{(h)}$	2	27	55.0084	1.5223	0.2648
	PCA	$\mathbf{E}^{(h)}$	1	15	54.8852	1.3911	0.2631
Conjunto Abalone	⌊	$\boldsymbol{\Delta}^{(h)}$	2	27	55.0789	1.5329	0.2517
	⌈	$\mathbf{Y}^{(h)}$	3	39	56.0407	1.6751	0.2655
	KPCA	$\mathbf{E}^{(h)}$	3	39	56.0371	1.4817	0.2693
	⌊	$\boldsymbol{\Delta}^{(\boldsymbol{h})}$	3	39	55.7751	1.4678	0.2623