

METODOLOGIA DE CÁLCULO DE COMPONENTES PRINCIPAIS DE DISCRIMINAÇÃO PARA CONJUNTOS DE DADOS MULTICLASSE

Gilvan Farias da Silva¹

Resumo

Neste trabalho é proposta uma metodologia para calcular os componentes principais de discriminação de um conjunto de dados. Os dados previamente discriminados foram utilizados para treinar redes neurais artificiais para classificar dados entre três classes. Os resultados obtidos foram comparados com redes neurais treinadas com dados apenas normalizados e com redes neurais treinadas com dados processados por análise de componentes principais. As redes neurais treinadas com dados discriminados com componentes principais de discriminação obtiveram maiores valores das médias dos produtos das eficiências e menor desvio padrão. Os resultados obtidos mostram que componentes principais de discriminação é uma alternativa viável a análise de componentes principais.

Palavras-chave: Componentes Principais de Discriminação (PCD), Análise de Componentes Principais (PCA), Redes Neurais Artificiais (RNA), Classificação.

¹ Mestrando em Engenharia Elétrica pela Universidade Federal da Bahia - UFBA.
E-mail: gilvan.farias.silva@gmail.com.

Introdução

Em aplicações com Redes Neurais Artificiais (RNA) voltadas para classificação de dados é interessante a utilização de técnicas de pré-processamento para facilitar o treinamento da rede neural e aumentar a eficiência de classificação (HAYKIN, 2009). Técnicas como Transformada de Fourier e normalização de dados são amplamente utilizadas em conjunto com Análise de Componentes Principais (PCA - *Principal Component Analysis*) para pré-processamento de dados. PCA é uma técnica que realiza uma transformação linear em um conjunto de dados transformando-os em um conjunto de dados ortogonais entre si, a ordem desses dados é proporcional à variância de cada componente. Visto que os primeiros componentes contêm a maior parte da variância, é possível selecionar apenas uma parte dos dados, reduzindo assim a dimensão dos dados que entram na rede neural (THARWAT, 2016).

Tendo em vista a importância do pré-processamento de dados para o treinamento de redes neurais, é plausível o estudo de novas técnicas para esta finalidade. PCA é uma técnica bem conhecida e que proporciona resultados muitas vezes satisfatórios. Em Simas Filho, Seixas e Calôba (2008) foi utilizada a técnica de Componentes Principais de Discriminação (PCD - *Principal Components for Discrimination*) para pré-processamento de dados provenientes do detector ATLAS, um dos detectores do Grande Colisor de Hádrons (LHC - *Large Hadron Collider*). Em seguida os dados foram utilizados para treinar uma rede neural, que proporcionou resultados melhores em comparação a rede neural treinada com dados pré-processados com PCA.

Tais resultados demonstram que a PCD é uma técnica com grande potencial de aplicação, principalmente em casos onde a é difícil obter bons resultados com as técnicas utilizadas atualmente. Entretanto, a PCD é uma técnica pouco conhecida e que foi utilizada até então apenas em pré-processamento de dados para problemas de classificação entre duas classes. Desenvolver uma metodologia para calcular PCD para problemas multi-classe proporcionará aos pesquisadores uma importante ferramenta para problemas de classificação.

Neste trabalho, uma metodologia de cálculo da PCD voltada para problemas de classificação multi-classe com redes neurais é proposta. Trata-se da generalização da PCD utilizada por Simas Filho, Seixas e Calôba (2008). Para analisar o desempenho da PCD foi realizado um estudo de caso utilizando o conjunto de dados Iris. Os resultados foram comparados com classificações realizadas por redes neurais treinadas

com dados pré-processados por PCA e com dados apenas normalizados. A rede neural treinada com dados pré-processados pela PCD proporcionou melhores resultados em comparação com a rede neural treinada com dados pré-processados pela PCA.

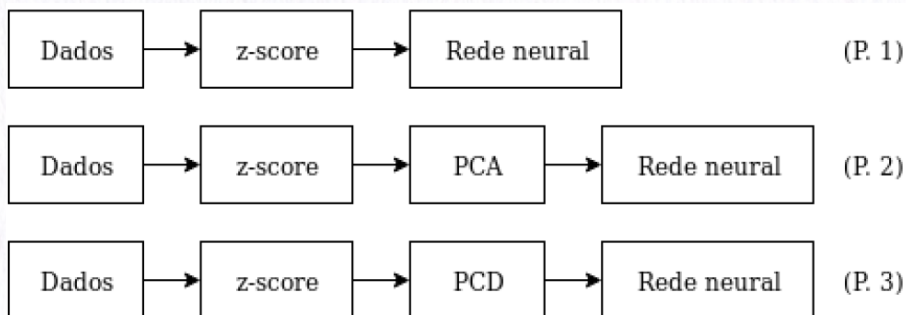
Os resultados deste trabalho contribuem para aumentar o referencial teórico a respeito de técnicas de pré-processamento de dados para redes neurais apresentando a PCD, que proporcionou bons resultados no caso analisado.

Metodologia

Para este trabalho foram utilizados os dados do conjunto Iris, que é formado por três classes de flores: *Iris setosa*, *Iris versicolour* e *Iris virginica*. Para cada classe há 50 instâncias com quatro dados numéricos: comprimento da sépala em cm, largura da sépala em cm, comprimento de pétala em cm e largura da pétala em cm (IRIS... 2019). Foram utilizadas três sequências de processamento, conforme mostrado no diagrama de blocos da Figura 1.

Na sequência de processamento P.1 os dados foram normalizados utilizando a normalização *z-score*, em seguida as redes neurais foram treinadas. Na sequência de processamento P. 2 após a normalização *z-score*, foram calculados os componentes principais com a PCA para três casos: 2, 3 e 4 componentes principais, em seguida as redes neurais foram treinadas. Na sequência de processamento P.3 os dados foram normalizados utilizando a normalização *z-score*, em seguida foram calculados os componentes principais com a PCD proposta neste trabalho para três casos: 2, 3 e 4 componentes principais. Estes componentes principais foram utilizados para treinar redes neurais.

Figura 1: Método utilizado para processamento de dados.

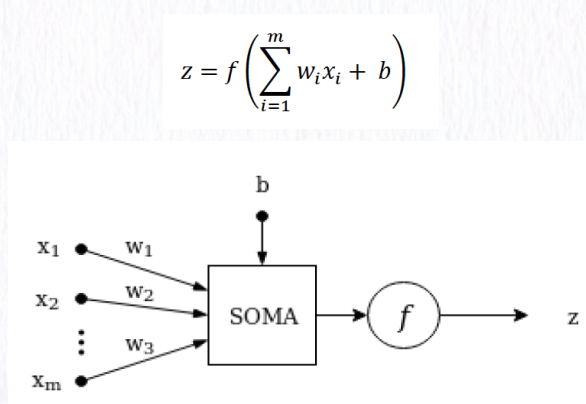


Fonte: Autor.

Redes Neurais Artificiais (RNA)

Um neurônio artificial é a representação matemática de um neurônio biológico. As entradas x_i , com i variando de 1 a m , são multiplicadas por pesos w_i , com i variando de 1 a m , denominados pesos sinápticos e então somadas ao bias b . Este somatório passa por uma função de ativação $f(\cdot)$ não linear, resultando na saída do neurônio (HAYKIN, 2009). O modelo matemático do neurônio artificial é representado pela equação (1) e pode ser visualizado no diagrama de blocos da Figura 2.

Figura 2: Representação em diagrama de blocos de um neurônio artificial.

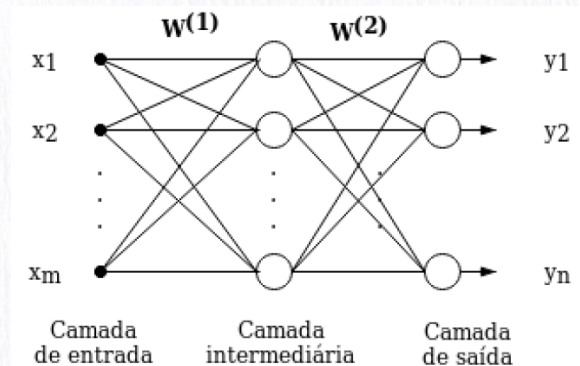


Fonte: Autor, baseado em (HAYKIN, 2009).

A interconexão de neurônios forma uma rede neural. Neste trabalho a topologia de rede neural utilizada é a *multi-layer perceptron* com uma camada intermediária (escondida).

A topologia *multi-layer perceptron* organiza os neurônios em camadas de forma que a saída de uma camada é entrada da camada seguinte (HAYKIN, 2009). A Figura 3 mostra uma rede neural *multi-layer perceptron* com uma camada de entrada composta por m entradas, uma camada intermediária e uma camada de saída composta por n saídas. Cada círculo representa um neurônio cujo bias não está indicado para melhor visualização. A matriz $\mathbf{W}(1)$ contém os pesos sinápticos entre a camada de entrada e a camada intermediária e a matriz $\mathbf{W}(2)$ contém os pesos sinápticos entre a camada intermediária e de saída,

Figura 3: Rede neural artificial multilayer perceptron com uma camada intermediária.



Fonte: Autor, baseado em (HAYKIN, 2009).

Uma importante aplicação das redes neurais é em problemas de classificação de dados (ZOLFAGHARI et al., 2017). Entretanto, para utilizar uma rede neural primeiramente é necessário treiná-la. Um método bastante utilizado é o *backpropagation*, que baseia-se na retro propagação do erro entre a saída desejada e obtida durante a fase de treinamento (HAYKIN, 2009).

Para obtenção de melhores resultados, os dados de entrada devem ser preparados de forma que determinados dados não sejam numericamente insignificante influenciando pouco na resposta da rede (HAYKIN, 2009). Neste trabalho os dados foram preparados utilizando a normalização *z-score*, que transforma os dados de forma a ter média zero e desvio padrão unitário (KELLEHER; NAMEE; D'ARCY, 2015). A normalização é realizada de acordo com a equação (2), em que a_i é o valor da instância i , \bar{a} é a média aritmética do conjunto a , $\sigma(a)$ é o desvio padrão do conjunto de dados a , \hat{a}_i é a instância i devidamente normalizada.

$$\hat{a}_i = \frac{a_i - \bar{a}}{\sigma(a)} \quad (2)$$

Foram criadas redes neurais com diferentes números de neurônios na camada intermediária. Sendo 2, 4, 6, 8, 10, 12, 14, 16, 18 e 20 neurônios. A camada de entrada tem dimensão igual a dimensão dos dados de entrada. A camada de saída possui 3 neurônios em que o primeiro neurônio deve apresentar o número 1 caso o dado em questão seja da classe *Iris setosa*, e 0 caso contrário. O segundo neurônio deve apresentar o número 1 se o dado em questão for da classe *Iris versicolour* e 0 caso

contrário. O terceiro neurônio deva apresentar o número 1 se o dado em questão for da classe *Iris virginica* e 0 caso contrário. A função de ativação utilizada foi a função logística. As redes neurais foram treinadas utilizando o método *backpropagation*.

Após o treinamento as redes devem ser testadas. O resultado do teste pode ser esquematizado por meio de uma "matriz de confusão", em que as linhas correspondem as classes reais a qual os dados pertencem e as colunas correspondem as classes preditas pela rede neural (DANGETI, 2017). A Tabela 1 mostra a matriz de confusão utilizada neste trabalho. Os elementos C11, C22 e C33 correspondem situações em que as classes *Iris setosa*, *Iris versicolour* e *Iris virginica* foram corretamente classificadas.

Tabela 1: Matriz de confusão

		Classe predita		
		<i>Iris setosa</i>	<i>Iris versicolour</i>	<i>Iris virginica</i>
Classe Real	<i>Iris setosa</i>	C ₁₁	C ₁₂	C ₁₃
	<i>Iris versicolour</i>	C ₂₁	C ₂₂	C ₂₃
	<i>Iris virginica</i>	C ₃₁	C ₃₂	C ₃₃

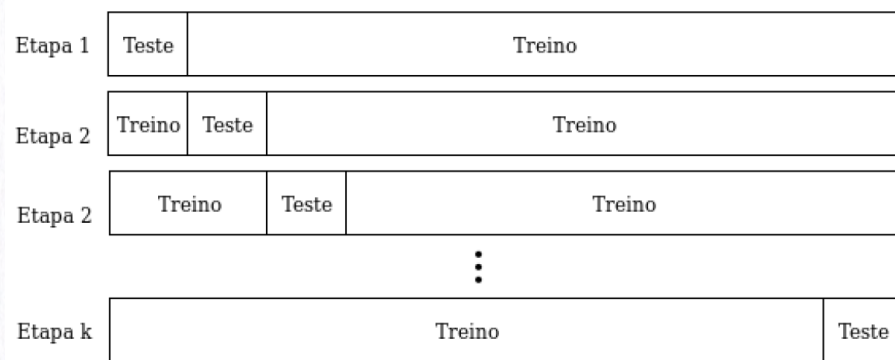
Fonte: Autor, baseado em (DANGETI, 2017)

Neste trabalho a eficiência de classificação foi avaliada a partir do produto das eficiências (PE) que é calculado a partir da média geométrica dos elementos da diagonal principal da matriz de confusão conforme a equação (3).

$$PE = \sqrt[3]{C_{11} \times C_{22} \times C_{33}} \quad (3)$$

Não é interessante testar a rede com os mesmos dados utilizados durante o treinamento, pois assim não teríamos informações quanto a capacidade de generalização da rede. Portanto, foi utilizado um procedimento de validação cruzada chamado *k-fold*, em que o conjunto de dados é dividido em k partes (*fold*). Para a fase de treinamento é utilizada k-1 *fold*, então o k-ésimo *fold* é utilizado para testar a capacidade de classificação da rede neural. O processo de repete k vezes de forma que todos os *fold* sejam utilizados como parte do conjunto de treino e como conjunto de teste (DANGETI, 2017; HAYKIN, 2009). A Figura 4 ilustra o método *k-fold*.

Figura 4: Validação cruzada k-fold.



Fonte: Autor baseado em (DANGETI, 2017).

Neste trabalho o conjunto de dados foi dividido em 10 partes. Com as primeiras 9 partes foi feito o treinamento da rede e a décima parte foi utilizada para testar a capacidade de classificação da rede neural. O processo se repete 10 vezes alterando a parte que é utilizada como teste, de forma que todas as partes foram utilizadas ao menos uma vez como conjunto de teste. O processo de treinamento foi repetido 10 vezes para cada *fold*, portanto cada rede foi treinada 100 vezes.

As camadas de entrada das redes neurais utilizadas neste trabalho foram dimensionadas de forma a coincidir com a dimensão dos dados de entrada. Para a sequência de processamento P. 1 as camadas de têm dimensão 4, coincidindo com o número de variáveis do conjunto Iris. Para as sequências de processamento P. 2 e P. 3 foram utilizadas camadas de entrada de dimensão 2, 3 e 4 de forma a coincidir com o número de componentes principais da PCA e PCD.

Os resultados obtidos nas três sequências de processamento foram comparados em relação aos valores de PE obtidos a partir do conjunto de teste. Como cada rede foi treinada 100 vezes foi calculada a média dos produtos das eficiências e seu desvio padrão, que é representado por $\sigma(PE)$. Foi feito também o gráfico boxplot de cada rede neural.

Análise de Componentes Principais (PCA)

A análise de componentes principais (PCA – *Principal Component Analysis*) é uma técnica que realiza uma transformação linear em um conjunto de dados transformando-o em outro conjunto cujos componentes são ortogonais entre si. Estes componentes são ordenados por ordem

decrecente da variância (ou energia) que os componentes contem. De forma que a maior parte da variância dos dados está contida nos primeiros componentes principais (MORAIS; LIMA, 2017; THARWAT, 2016).

A PCA é utilizada para diminuir a dimensão e redundâncias nos dados, sendo assim empregada como pré-processamento de dados que são utilizados como entrada de redes neurais. Primeiro é calculado o espaço PCA que é uma matriz que indica as direções de maior variância dos dados. Esta matriz é composta pelos autovetores da matriz de covariância dos dados que se deseja processar, os autovetores são ordenados em ordem decrescente de acordo com o valor do respectivo autovalor. Para obter os componentes principais z , a matriz B representando o espaço PCA é multiplicada pelo vetor de dados de entrada x (THARWAT, 2016). Conforme a equação (4).

$$z = Bx \quad (4)$$

Componentes Principais de Discriminação (PCD)

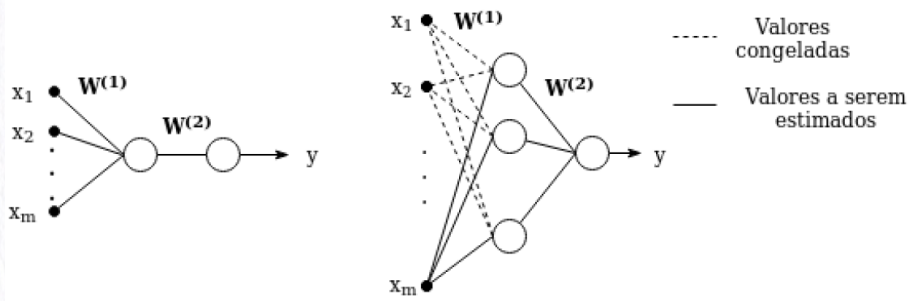
Segundo Simas Filho, Seixas e Calôba (2008) a técnica de Componentes Principais de Discriminação (PCD - *Principal Components for Discrimination*) realiza uma projeção linear que maximiza a separação entre as classes e compacta os dados simultaneamente. Para calcular a PCD em problemas binários devem-se seguir as seguintes etapas:

1. Inicializa-se uma rede neural do tipo *multilayer perceptron* com uma camada escondida, em que a camada de entrada tem a dimensão dos dados que se deseja processar, a camada intermediária tem um neurônio e a camada de saída tem um neurônio;
2. Treina-se esta rede neural de forma que o neurônio de saída apresente valores distintos para as duas classes;
3. Adiciona-se outro neurônio à camada intermediária;
4. Re-treina a rede de forma que os pesos entre a camada intermediária e de saída podem ser modificados, porém os pesos entre a camada de entrada e intermediária que já foram sintonizados ficam congelados e modificam-se apenas os pesos entre a camada de entrada e o novo neurônio;
5. Adicionam-se novos neurônios conforme a necessidade de redução dimensional e repete os passos 2, 3 e 4.

A Figura 5 ilustra o treinamento de uma PCD, em que $W(1)$ é a matriz com os pesos entre a camada de entrada e intermediária, $W(2)$ é a matriz

com os pesos entre a camada intermediária e de saída, x é o vetor com os dados a serem processados e y é a saída da rede neural. Após o treinamento da rede os componentes principais z são calculados como sendo os valores de entrada dos neurônios da camada escondida usando a equação (5), sendo z os componentes principais.

Figura 5: Treinamento da PCD para o caso de problemas de classificação binária.



Fonte: Autor, baseado em (SIMAS FILHO; SEIXAS; CALÔBA, 2008)

$$z = W^{(1)}x \quad (5)$$

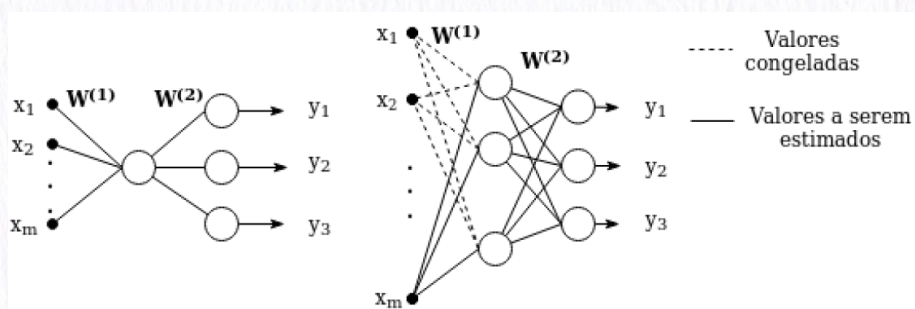
Método proposto de treinamento da PCD para problemas de classificação multi-classe

Neste trabalho é proposta uma generalização da técnica PCD, de forma a aplicá-la em problemas de classificação multi-classe. O método de treinamento da PCD multi-classe é proposto da seguinte forma:

1. Inicializa-se uma rede neural do tipo *multilayer perceptron* com uma camada escondida, em que a camada de entrada tem a dimensão dos dados que se deseja processar, a camada intermediária tem um neurônio e a camada de saída tem número de neurônios igual ao número de classes;
2. Treina-se esta rede neural com o método *Backpropagation*;
3. Adiciona-se outro neurônio à camada intermediária;
4. Re-treina a rede de forma que os pesos entre a camada intermediária e de saída podem ser modificados, porém os pesos entre a camada de entrada e intermediária que já foram sintonizados ficam congelados e modificam-se apenas os pesos entre a camada de entrada e o novo neurônio;
5. Adicionam-se novos neurônios conforme a necessidade de redução dimensional e repete os passos 2, 3 e 4.

A Figura 6 ilustra o treinamento de uma PCD para multi-classe, em que $W^{(1)}$ é a matriz com os pesos entre a camada de entrada e intermediária, $W^{(2)}$ é a matriz com os pesos entre a camada intermediária e de saída, x é o vetor com os dados a serem processados e y é a saída da rede neural. Após o treinamento da rede os componentes principais são calculados como sendo os valores de saída dos neurônios da camada escondida usando a equação (6), sendo z os componentes principais e b o bias.

Figura 6: Treinamento da PCD para o caso de problemas de classificação multi-classe.



Fonte: Autor.

$$z = f(W^{(1)}x + b) \quad (6)$$

Resultados e discussão

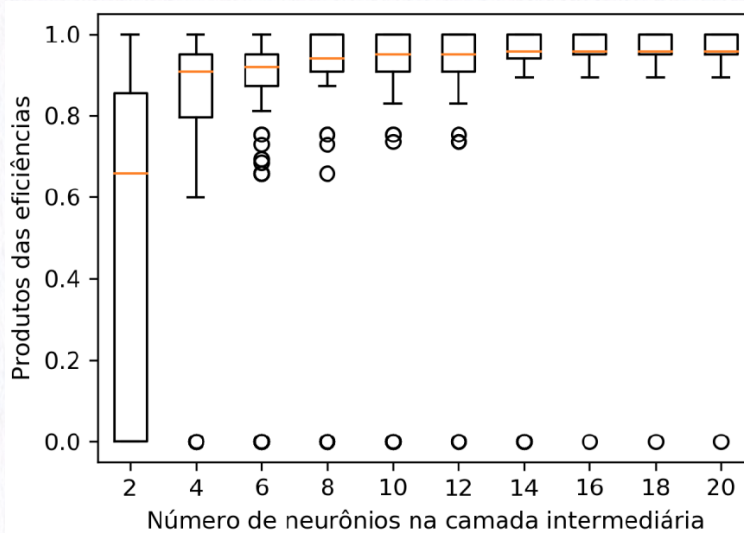
Sequência de processamento (P. 1): Rede treinada apenas com dados normalizados

inicialmente as redes neurais foram treinadas com dados apenas normalizados. A média dos produtos das eficiências aumentou com o aumento do número de neurônios da camada intermediária, estabilizando para redes com 16 ou mais neurônios. O tempo de treinamento aumentou com o aumento do número de neurônios da camada intermediária, tal aumento é devido ao aumento do número de pesos sinápticos que precisam ser ajustados durante o treinamento. A Tabela 2 mostra os valores médios e desvio padrão do produto das eficiências e do tempo de treinamento de cada topologia de rede treinada para o caso (P. 1). O boxplot dos produtos das eficiências pode ser visto na Figura 7.

Tabela 2: Dados de produto das eficiências e tempo de processamento para cada topologia de rede neural no método de processamento (P. 1).

Neurônios	Média PE	σ (PE)	Media Tempo	σ (Tempo)
2	0,532830	0,374180	0,812086	0,027659
4	0,807020	0,285116	0,852336	0,031577
6	0,820543	0,285432	0,887244	0,022562
8	0,848306	0,288827	0,917884	0,028650
10	0,863230	0,276191	0,947937	0,030876
12	0,907597	0,194877	0,982378	0,044294
14	0,914231	0,211932	1,015731	0,055244
16	0,952815	0,100799	1,030152	0,036642
18	0,953060	0,101045	1,060118	0,026907
20	0,954622	0,101077	1,087401	0,020104

Figura 7: Boxplot do produto das eficiências para a sequência P. 1.



Sequência de processamento (P. 2): Rede treinada com dados processados por PCA

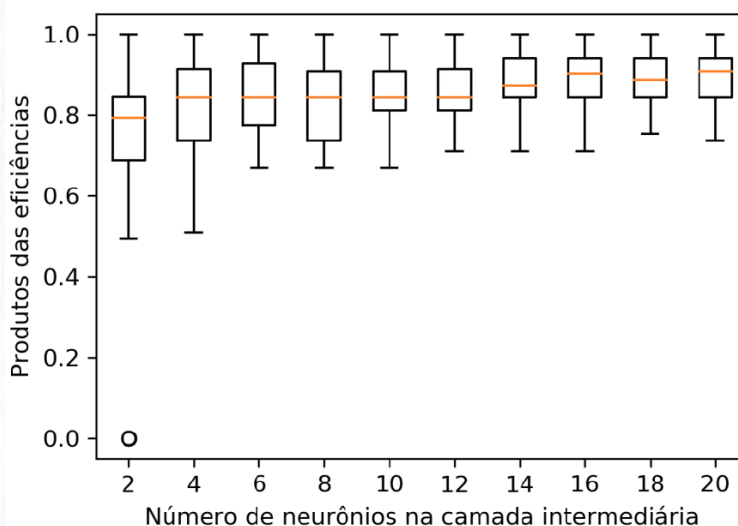
Utilizando dois componentes PCA houve melhora na média do produto das eficiências para redes com até 6 neurônios. Nas demais redes neurais treinadas no método (P. 2) a média do produto das eficiências foi menor que as médias dos produtos das eficiências das redes treinadas pelo método (P. 1). Entretanto, em todos os casos o desvio padrão do PE

foi menor para o caso (P. 2) com 2 componentes que no caso (P. 1). A Tabela 3 mostra os valores médios e desvio padrão do produto das eficiências e do tempo de treinamento de cada topologia de rede treinada para o caso (P. 2) utilizando 2 componentes PCA. O boxplot dos produtos das eficiências pode ser visto na Figura 8.

Tabela 3: Dados de produto das eficiências e tempo de processamento para cada topologia de rede neural na sequência de processamento (P. 2) utilizando 2 componentes PCA.

Neurônios	Média PE	σ (PE)	Média Tempo	σ (Tempo)
2	0,716497	0,252308	0,833647	0,030225
4	0,838046	0,098265	0,867447	0,030279
6	0,845144	0,090542	0,902887	0,016370
8	0,844278	0,092501	0,935517	0,030266
10	0,850551	0,083464	0,968624	0,027999
12	0,859202	0,081522	0,986048	0,019846
14	0,877241	0,073030	1,012202	0,016850
16	0,881702	0,071332	1,042146	0,024227
18	0,882502	0,069089	1,068357	0,022533
20	0,884464	0,069212	1,104063	0,030422

Figura 8: Boxplot do produto das eficiências das redes treinadas na sequência de processamento (P. 2) com 2 componentes PCA.



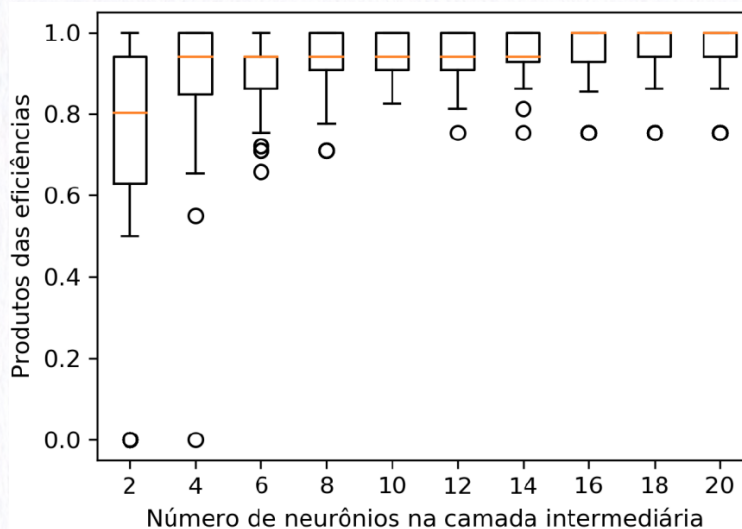
Utilizando três componentes PCA houve melhora na média do PE para todas topologias de redes, a exceção da rede com 16 que obteve

média 0.00025 menor que no método (P. 1). Em todos os casos o desvio padrão do PE foi menor que no caso (P. 1). A Tabela 4 mostra os valores médios e desvio padrão do produto das eficiências e do tempo de treinamento de cada topologia de rede treinada para o caso (P. 2) utilizando 3 componentes PCA. O boxplot dos produtos das eficiências pode ser visto na Figura 9.

Tabela 4: Dados de produto das eficiências e tempo de processamento para cada topologia de rede neural na sequência de processamento (P. 2) utilizando 3 componentes PCA.

Neurônios	Média PE	σ (PE)	Média Tempo	σ (Tempo)
2	0,701096	0,325283	0,835919	0,029036
4	0,890164	0,138805	0,876251	0,030297
6	0,910599	0,079612	0,912452	0,031931
8	0,927965	0,063760	0,942431	0,038123
10	0,940169	0,048282	0,966304	0,019833
12	0,945396	0,050557	0,992203	0,019258
14	0,953860	0,051161	1,028077	0,041252
16	0,952565	0,065121	1,041885	0,019758
18	0,954040	0,065996	1,076554	0,024685
20	0,955693	0,068135	1,109134	0,036139

Figura 9: Boxplot do produto das eficiências das redes treinadas na sequência de processamento (P. 2) com 3 componentes PCA.

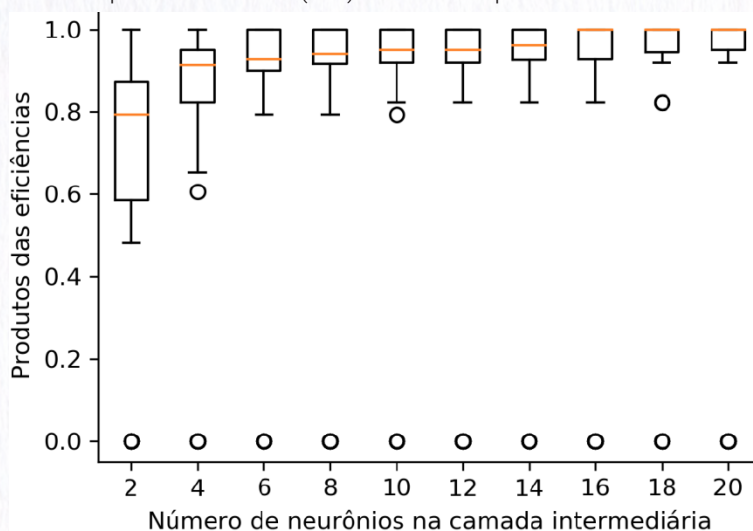


Utilizando quatro componentes PCA houve melhora na média do PE para até 8 neurônios na camada intermediária, para as demais topologias as médias do PE foram inferiores aos encontrados com o método de processamento (P. 1). Houve aumento do desvio padrão do PE para redes com 6 ou mais neurônios na camada intermediária em relação ao caso (P. 1). A Tabela 5 mostra os valores médios e desvio padrão do produto das eficiências e do tempo de treinamento de cada topologia de rede treinada para o caso (P. 2) utilizando 4 componentes PCA. O boxplot dos produtos das eficiências pode ser visto na Figura 10.

Tabela 5: Dados de produto das eficiências e tempo de processamento para cada topologia de rede neural no método de processamento (P. 2) utilizando 4 componentes PCA.

Neurônios	Média PE	σ (PE)	Média Tempo	σ (Tempo)
2	0,651169	0,336786	0,846225	0,031524
4	0,832981	0,243896	0,872548	0,021988
6	0,844843	0,286975	0,911130	0,022200
8	0,848557	0,288867	0,943706	0,020307
10	0,856476	0,290734	0,969454	0,021628
12	0,859150	0,290856	1,006413	0,046252
14	0,863989	0,291770	1,032551	0,032802
16	0,875263	0,293815	1,062866	0,044010
18	0,876643	0,294601	1,087841	0,037347
20	0,882015	0,295129	1,123896	0,038621

Figura 10: Boxplot do produto das eficiências das redes treinadas na sequência de processamento (P. 2) com 3 componentes PCA.



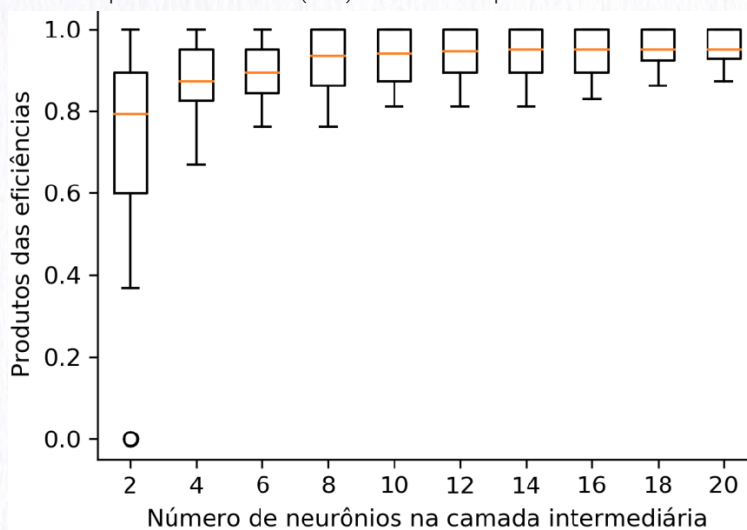
Sequência de processamento (P. 3): Rede treinada com dados processados por PCD

Utilizando dois componentes PCD houve melhora na média do PE em relação ao caso (P. 1) para todas topologias de rede, a exceção da rede com 16 neurônios na camada intermediária. Houve diminuição do desvio padrão do PE para todas as topologias de rede em relação ao caso (P. 1). A Tabela 6 mostra os valores médios e desvio padrão do produto das eficiências e do tempo de treinamento de cada topologia de rede treinada para o caso (P. 3) utilizando 2 componentes PCD. O boxplot dos produtos das eficiências pode ser visto na Figura 11.

Tabela 6: Dados de produto das eficiências e tempo de processamento para cada topologia de rede neural no método de processamento (P. 3) utilizando 2 componentes PCD.

Neurônios	Média PE	σ (PE)	Média Tempo	σ (Tempo)
2	0,682849	0,310117	0,851206	0,072939
4	0,885284	0,077142	0,888106	0,017502
6	0,904292	0,070197	0,925416	0,021457
8	0,919072	0,067577	0,951214	0,017759
10	0,929714	0,062721	0,993557	0,032265
12	0,941477	0,056637	1,011484	0,021965
14	0,943642	0,052422	1,049001	0,036375
16	0,948543	0,049031	1,079795	0,045929
18	0,954479	0,044613	1,098516	0,026131
20	0,957526	0,040825	1,130271	0,034108

Figura 11: Boxplot do produto das eficiências das redes treinadas na sequência de processamento (P. 3) com 2 componentes PCD.

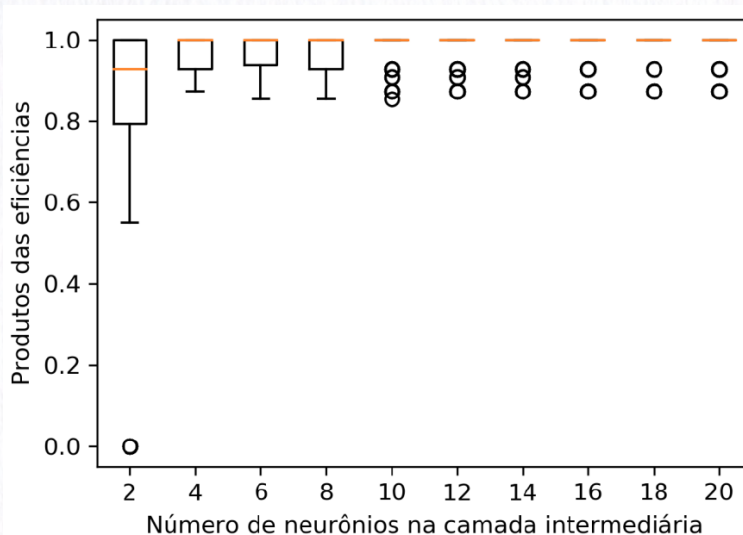


Utilizando três componentes PCD houve melhora na média do PE em relação ao caso (P. 1) para todas topologias de rede. Houve diminuição do desvio padrão do PE para todas as topologias de rede em relação ao caso (P. 1) e ao caso (P. 2) com três componentes PCA. A Tabela 7 mostra os valores médios e desvio padrão do produto das eficiências e do tempo de treinamento de cada topologia de rede treinada para o caso (P. 3) utilizando 3 componentes PCD. O boxplot dos produtos das eficiências pode ser visto na Figura 12.

Tabela 7: Dados de produto das eficiências e tempo de processamento para cada topologia de rede neural no método de processamento (P. 3) utilizando 3 componentes PCD.

Neurônios	Média PE	σ (PE)	Média Tempo	σ (Tempo)
2	0,803563	0,314604	0,847714	0,021105
4	0,966162	0,046082	0,885134	0,023209
6	0,973903	0,045224	0,919489	0,021833
8	0,973632	0,046602	0,950544	0,027644
10	0,975997	0,044688	0,981002	0,035411
12	0,979078	0,042310	1,008320	0,023267
14	0,978163	0,042833	1,033728	0,019234
16	0,981623	0,040885	1,062358	0,036447
18	0,984491	0,039537	1,080392	0,022168
20	0,982340	0,040571	1,112266	0,054706

Figura 12: Boxplot do produto das eficiências das redes treinadas na sequência de processamento (P. 3) com 3 componentes PCD.

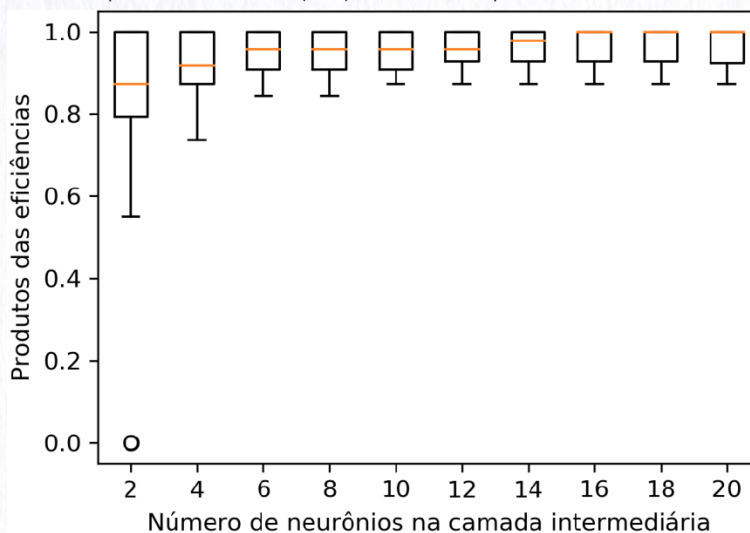


Utilizando quatro componentes PCD houve melhora na média do PE em relação aos casos (P. 1) e (P. 2) para todas topologias de rede. Houve diminuição do desvio padrão do PE para todas as topologias de rede em relação ao caso (P. 1) e ao caso (P. 2) com quatro componentes PCA. A Tabela 8 mostra os valores médios e desvio padrão do produto das eficiências e do tempo de treinamento de cada topologia de rede treinada para o caso (P. 3) utilizando 4 componentes PCD. O boxplot dos produtos das eficiências pode ser visto na Figura 13.

Tabela 8: Dados de produto das eficiências e tempo de processamento para cada topologia de rede neural no método de processamento (P. 3) utilizando 4 componentes PCD.

Neurônios	Média PE	σ (PE)	Média tempo	σ (Tempo)
2	0,819051	0,247465	0,838244	0,015929
4	0,934485	0,061408	0,876645	0,029490
6	0,947899	0,054670	0,916271	0,030825
8	0,953548	0,050490	0,941252	0,019023
10	0,956333	0,046638	0,968773	0,016325
12	0,959539	0,043692	1,005124	0,028239
14	0,959931	0,044023	1,028469	0,023109
16	0,962938	0,044981	1,050222	0,014890
18	0,964271	0,043444	1,088594	0,025753
20	0,963380	0,043023	1,124047	0,061946

Figura 13: Boxplot do produto das eficiências das redes treinadas na sequência de processamento (P. 3) com 4 componentes PCD.



Considerações finais

Neste trabalho foi proposta uma metodologia para obter os componentes principais de discriminação para problemas de classificação multi-classe, tomando como base o conjunto de dados Iris. A PCD proporcionou maior média do produto das eficiências em comparação com a PCA, com destaque para a rede neural treinada com 3 componentes PCD. A rede neural com maior média dos produtos das eficiências foi treinada com 3 componentes PCD e 18 neurônios na camada intermediária proporcionando a média de 0.9845. Os desvios padrões dos PE foram menores para as redes neurais treinadas com PCD se comparados com as redes treinadas com a PCA.

O tempo de treinamento da rede neural em geral aumentou proporcionalmente ao número de neurônios na camada intermediária independente da sequência de processamento. Isto ocorre devido ao aumento de pesos sinápticos que dever ser alterados a cada iteração durante o treinamento. Não houve variação significativa na média dos tempos de treinamento das redes neurais, nem no desvio padrão.

A PCD é uma alternativa viável a PCA, facilitando o processo de classificação de dados por redes neurais. Por ser uma técnica nova é preciso que novas pesquisas sejam realizadas futuramente para proporcionar mais informações sobre a aplicabilidade da PCD.

Referências

DANGETI, Pratap. **Statistics for Machine Learning: Build supervised, unsupervised, and reinforcement learning models using both Python and R.** Birmingham: Packt Publishing, 2017. ISBN-13: 978-1-78829-575-8.

HAYKIN, Simon. **Neural networks and learning machines.** 3. ed. New York: Pearson Prentice Hall, 2009. ISBN-13: 978-0-13-147139-9.

IRIS Data Set. Disponível em: <<https://archive.ics.uci.edu/ml/datasets/iris>>. Acesso em: 20 mar. 2019.

KELLEHER, John D.; NAMEE, Brian Mac; D'ARCY, Aoife. **Fundamentals of machine learning for predictive data analytics: Algorithms, worked examples, and case studies.** Cambridge, Ma: Mit Press, 2015. ISBN-13: 978-0-262-02944-5.

MORAIS, Camilo; LIMA, Kássio. Principal Component Analysis with Linear and Quadratic Discriminant Analysis for Identification of Cancer Samples Based on Mass Spectrometry. **Journal Of The Brazilian Chemical Society**, [s.l.], v. 29, n. 3, p.472-481, 2017. Sociedade Brasileira de Química (SBQ). <http://dx.doi.org/10.21577/0103-5053.20170159>.

SIMAS FILHO, Eduardo Furtado de; SEIXAS, José M.; CALÔBA, Luiz P.. Optimized Calorimeter Signal Compaction for an Independent Component based ATLAS Electron/Jet Second-Level Trigger. In: **ADVANCED COMPUTING AND ANALYSIS TECHNIQUES IN PHYSICS RESEARCH**, 12., 2008, Erice. **Proceedings...** . Trieste: PoS, 2009. v. 70, p. 1 – 10. <https://doi.org/10.22323/1.070.0102>.

THARWAT, Alaa. Principal component analysis - a tutorial. **International Journal Of Applied Pattern Recognition (ijapr)**, Genève, v. 3, n. 3, p.197-240, jan. 2016. <http://dx.doi.org/10.1504/IJAPR.2016.079733>.

ZOLFAGHARI, Sahar et al. Broken Rotor Bar Fault Detection and Classification Using Wavelet Packet Signature Analysis Based on Fourier Transform and Multi-Layer Perceptron Neural Network. **Applied Sciences**, [s.l.], v. 8, n. 1, p.1-21, 25 dez. 2017. MDPI AG. <http://dx.doi.org/10.3390/app8010025>.