

# 2ª Lista de Exercícios Computacionais - Resolução

Lucas F. P. Costa  
Universidade Tecnológica Federal do Paraná  
Ponta Grossa, Paraná  
luccos@alunos.utfpr.edu.br

## I. INTRODUÇÃO

Redes de Perceptron de Múltiplas Camadas (MLP, *Multi-layer Perceptron*) podem ser consideradas como a generalização do Perceptron comum usado repetidamente para formar uma estrutura capaz de realizar aproximações universais graças a sua composição de funções-base, muito embora não haja forma específica de dizer qual o melhor método para o cálculo de pesos, tampouco que diga a quantidade de neurônios ideal que leve ao melhor modelo possível [1].

O objetivo deste trabalho é a solução de duas questões propostas na disciplina de Tópicos Avançados Em Métodos Computacionais, também conhecida por Redes Neurais [2]. Na primeira questão, deve-se aplicar MLP para solução do comportamento de portas lógicas. Para a segunda, as redes MLP são aplicadas no problema de classificação de duas bases de dados diferentes, chamadas IRIS (classes de plantas do tipo iris) e WINE (resultados de análises químicas de vinhos) [3].

Este trabalho está organizado por ordem das questões, contendo para cada a sua devida descrição, desenvolvimento da solução e resultados obtidos. Ao final, é feita a conclusão.

## II. QUESTÃO 1

### A. Descrição

Considere as portas lógicas AND, OR e XOR. Monte a tabela verdade de cada uma e faça a classificação das mesmas utilizando como modelo a MLP. Para cada caso, determine um critério de parada e apresente o erro de treinamento em forma de um gráfico do MSE  $\times$  iteração. Discuta se o modelo conseguiu alcançar 100% de acerto e as razões para os resultados. Para cada caso, execute a rede pelo menos 10 vezes e varie o número de neurônios na camada escondida, mostrando a melhor configuração alcançada.

### B. Desenvolvimento

As portas lógicas são representadas na Tabela I, mostrando seu funcionamento para entradas  $X1$  e  $X2$  e suas respectivas saídas  $Y$ .

Tabela I  
FUNÇÕES LÓGICAS AND, OR E XOR

$X1$	$X2$	$Y_{AND}$	$Y_{OR}$	$Y_{XOR}$
0	0	0	0	0
1	0	0	1	1
0	1	0	1	1
1	1	1	1	0

Para obter mais dados de entrada e não depender somente da quantidade limitada de observações da Tabela I, é possível

usar valores aleatórios desde que respeitada a seguinte relação, sendo  $X$  equivalente tanto a  $X1$  como  $X2$ :

$$X = \begin{cases} 0, & \text{se } X < 0,5. \\ 1, & \text{se } X \geq 0,5. \end{cases} \quad (1)$$

Dessa forma, diversos pontos de entrada podem ser criados para posteriormente serem usados como dados de treino da rede.

A modelagem da rede MLP depende da definição de diversos parâmetros que impactam no seu desempenho, como a função de ativação ( $F_A$ ), o número de neurônios da camada escondida ( $N_N$ ), a taxa de aprendizado ( $\alpha$ ) considerando a otimização por gradiente descendente e a quantidade de épocas de treinamento ( $E_T$ ). Além disso, para evitar treinamento desnecessário, pode-se usar critérios de parada antecipada, fazendo com que o treinamento pare antes de atingir todas as  $E_T$  definidas por não identificar mais melhoria no cálculo do erro de cada iteração, por exemplo. Neste caso, também é possível definir outro parâmetro denominado de “paciência” ( $P$ ), que é a quantidade de iterações de treino que devem ser executadas mesmo sem que haja melhoria do critério de parada, na esperança de encontrar ainda algum resultado possível.

A confirmação dos modelos não deve ser considerada para apenas uma rodada da rede sobre os dados, uma vez que pode causar resultados otimistas sobre a habilidade de generalização da rede [4]. Para evitar isso, utiliza-se uma técnica chamada validação cruzada *K-Fold*, que consiste em dividir os dados em  $k$  grupos (ou *folds*) de tamanho similar. Depois, um destes grupos é usado para validação e os  $k - 1$  restantes para treino, sendo um processo realizado  $k$  vezes, alternando os grupos de dados [5]. Ao final, a média dos resultados obtidos é calculada e usada para avaliar as habilidades reais do modelo.

### C. Resultados

Para realização dos testes, foi gerada uma entrada com total de 400 valores. Os parâmetros escolhidos para a rede MLP são descritos na Tabela II. Foram efetuados testes para quantidades diferentes de  $N_N$ , sendo que os valores foram: 2, 20 e 200. A validação cruzada *K-Fold* foi aplicada com  $k = 10$ , ou seja, são feitas 10 rodadas para avaliar o modelo proposto, para cada porta lógica desejada, dividindo 90% dos dados para treino e 10% para validação.

Os resultados das melhores configurações de  $N_N$  para cada porta lógica estão registrados na Tabela III, para valores médios de Taxa de Acerto ( $T_A$ ), Erro Quadrático Médio (MSE),

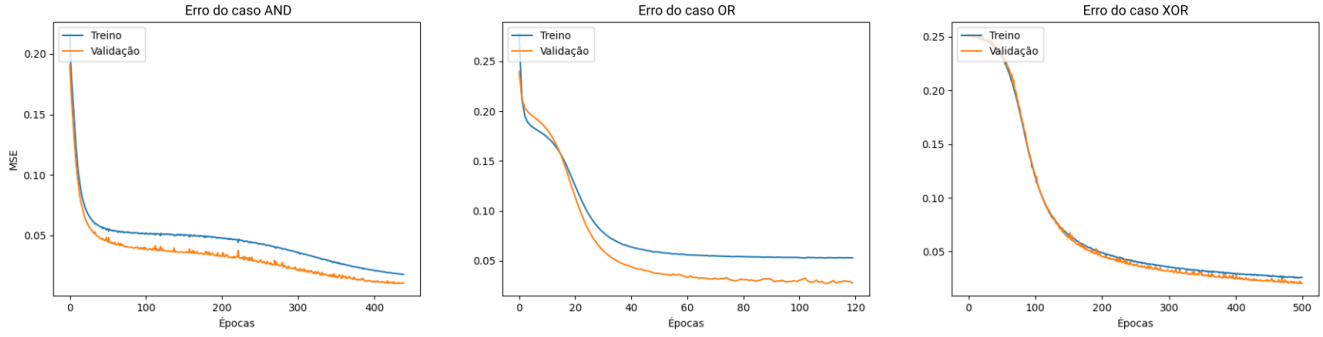


Figura 1. MSE de cada época durante o treinamento da rede MLP para cada porta lógica.

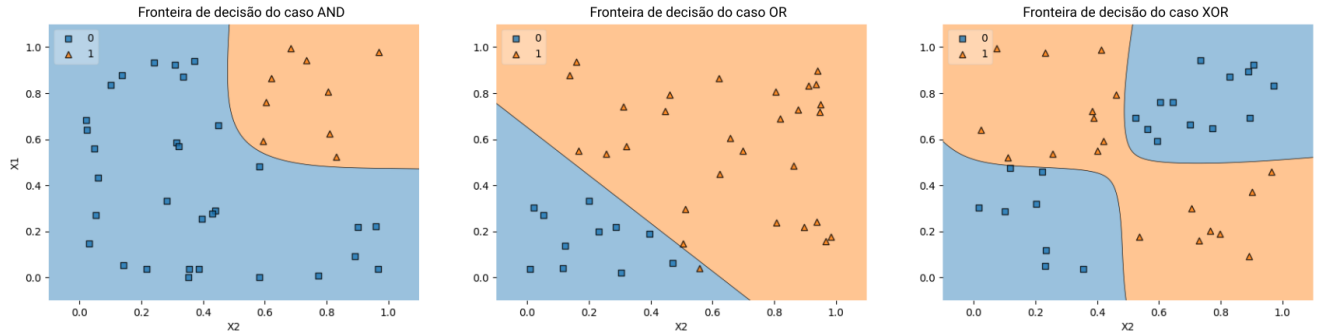


Figura 2. Fronteiras de decisão resultantes de cada caso.

Tabela II  
PARÂMETROS SELECIONADOS PARA TREINO DA MLP

Parâmetro	Valor
$F_A$	Tangente hiperbólica
$\alpha$	0,01
$E_T$	500
$P$	10

Mean Squared Error) e  $E_T$  levando em conta o critério de parada.

Tabela III  
RESULTADOS MÉDIOS PARA CADA CASO DE PORTA LÓGICA

	AND	OR	XOR
$N_N$	20	2	20
$T_A$	94,00%	92,75%	75,50%
MSE	0,0433	0,0493	0,1584
$E_T$	162,40	107,10	168,80

Pode-se notar que nenhum dos modelos foi capaz de alcançar 100% de acerto para a classificação de cada porta, sendo que o melhor resultado foi obtido para o caso AND tanto para  $T_A$  como MSE. Para o caso XOR, a melhor configuração obteve em média 75,50% de acerto, o pior dos três casos observados, o que mostra a dificuldade de lidar com este problema. É possível concluir também que não há uma quantidade certa

de  $N_N$  que devem ser usados, sendo que no caso OR o melhor resultado veio da utilização de apenas 2, diferente dos demais. Nos testes feitos, a alternativa de 200 neurônios foi a que retornou piores resultados para todos os casos, um exagero que deve ser evitado. Ainda, o uso do critério de parada teve atuação importante para que o treinamento não acontecesse por tempo desnecessário, acionando a parada sempre antes da metade do total estabelecido.

Nas Figuras 1 e 2 são ilustrados respectivamente gráfico de erros de treinamento e a fronteira de decisão formada por cada modelo no melhor caso, ou seja, da rodada que obteve os melhores resultados.

Nota-se pela análise das figuras o comportamento adequado e esperado de cada caso, sendo que a  $T_A$  destes modelos foi igual a 100% para a porta AND, 97,5% para OR e 100% para XOR, e o MSE de 0,0106 para AND, 0,0272 para OR e 0,0203 para XOR. Com exceção do caso OR, são modelos que conseguem classificar com total certeza todos os dados que foram a eles apresentados, todos demonstrando boa capacidade de generalização.

### III. QUESTÃO 2

#### A. Descrição

Encontre no site UCI Machine Learning as bases IRIS e WINE. Implemente a rede MLP com uma camada escondida, de modo que ela seja treinada pelo algoritmo gradiente descendente munido do backpropagation. Implemente também a validação cruzada tipo *holdout*. Mostre um gráfico do erro (MSE) de treinamento e de validação para

a melhor de 10 execuções completas, mostrando o ponto (em qual iteração) a validação apresentou menor erro. Faça um gráfico boxplot dos resultados de classificação em termos do MSE e apresente, para cada uma das 10 rodadas, a taxa percentual de acerto. Em cada caso, compare os resultados para números distintos de neurônios na camada intermediária. Use como função de ativação a tangente hiperbólica. Normalize sempre os dados no intervalo  $[-1; 1]$ , por meio da expressão abaixo:

$$z = -1 + 2 \left( \frac{x - x_{min}}{x_{max} - x_{min}} \right). \quad (2)$$

Ao final, para cálculo dos erros no espaço real, é preciso desfazer a normalização.

### B. Desenvolvimento

A base de dados IRIS é composta por três classes de plantas do tipo iris, contendo 50 instâncias de cada com informações sobre comprimento e largura das sépulas e pétalas (quatro tipos de informação, ao todo). Já a base WINE é o resultado da análise química de três tipos de vinho de uma mesma região da Itália, levantando ao todo 13 características e medidas sobre 178 instâncias ou amostras [3].

É de suma importância para correto funcionamento e treinamento da MLP que os dados, tanto de entrada como de saída para ambos conjuntos sejam normalizados, dada a discrepância que existe entre cada informação. Para tanto, a Equação 2 pode ser aplicada, como explanado na descrição da questão. Interessante notar que para ambos conjuntos de dados, existem três classes diferentes como resposta desejada, o que caracteriza este como um problema de multiclassificação.

Não obstante, a estrutura e parâmetros da MLP são os mesmos considerados anteriormente, devendo-se atentar tão somente para as camadas de entrada e saída da rede.

### C. Resultados

As configurações e os parâmetros usados para realização deste experimento são os mesmos definidos na Tabela II, com exceção da  $\alpha$  que foi alterada para 0,05 e o valor de  $P$ , para 50. Também, foram aplicados  $N_N$  de 8, 16 e 32. Importante salientar que, usando a validação cruzada  $K$ -Fold, é como se para cada  $k$  o conjunto de dados fosse separado usando *holdout* (divisão do conjunto de dados em subconjuntos de treino e validação) de 10%.

A seguir, são mostrados os resultados de classificação e sua sequente discussão para cada base de dados, separadamente, iniciando com IRIS e seguida por WINE.

1) *Base de dados IRIS*: Nesta base de dados, a melhor configuração da MLP foi a que usou  $N_N$  igual a 16, obtendo médias de  $T_A$  de 98%, MSE igual a 0,0092 e 212,6  $E_T$ . As Figuras 3 e 4 mostram como se comportou o MSE e a  $T_A$  para cada rodada de treino, respectivamente.

É possível perceber que a variação de MSE foi alta para as rodadas 2 e 4, embora tenham atingido 100% de acerto ao final. Além disso, os piores resultados foram para as rodadas 1, 3 e 5, únicos casos onde a  $T_A$  ficou abaixo de 100%. Entretanto, nota-se um padrão de comportamento principalmente nas rodadas finais, o que denota serem as mais indicadas para trabalhar com o conjunto de dados, sendo que

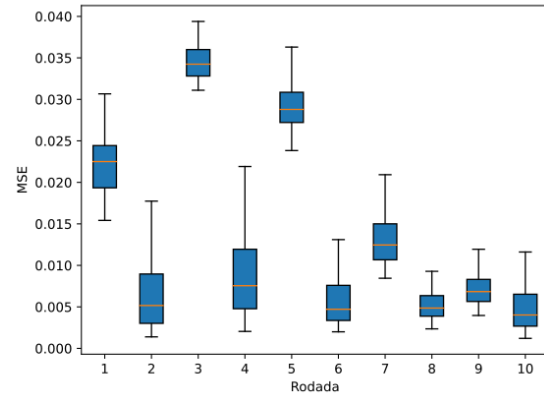


Figura 3. Boxplot de MSE de cada rodada usando os dados da base IRIS.

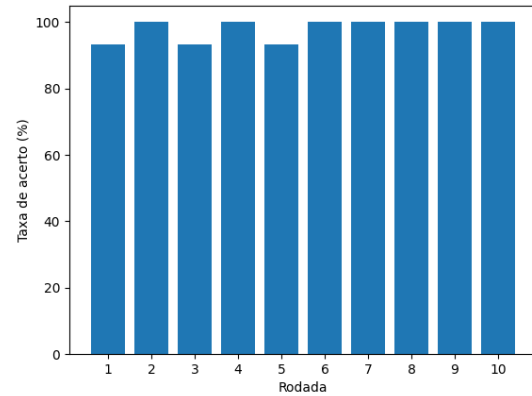


Figura 4.  $T_A$  de cada rodada sobre os dados da base IRIS.

a melhor rodada foi a 10ª, cujo gráfico de erro está ilustrado na Figura 5.

Para este modelo, a  $T_A$  encontrada é de 100% e MSE de 0,0012, sendo encontrado na época 204 de treino, como indicado na figura, tendo parada antecipada acionada depois de 50 iterações sem melhoria, ou seja, na época 254, mostrando-

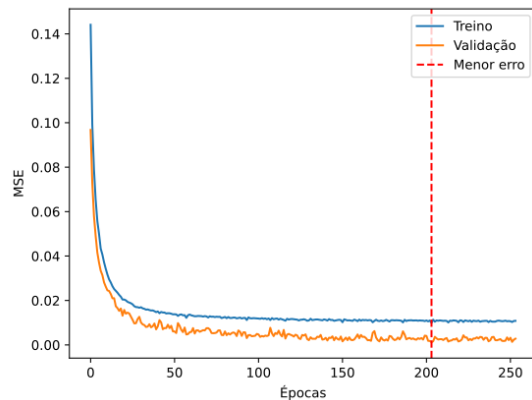


Figura 5. Progresso do MSE durante treino do melhor modelo para base IRIS.

se o melhor classificador para este conjunto de dados.

2) *Base de dados WINE*: Diferentemente do caso anterior, usando  $N_N$  de 8 obteve-se a melhor configuração, com  $T_A$  médio de 98,89%, MSE médio de 0,0084 e  $E_T$  médio de 370,4. Provavelmente a necessidade de mais  $E_T$  se deve ao fato da menor quantidade de neurônios na camada intermediária. Mesmo assim, os resultados médios foram levemente melhores que para a base de dados anterior, sendo que um possível fator pode ter sido a maior quantidade de entradas diferentes (13 contra quatro). Nas Figuras 6 e 7 tem-se o MSE e a  $T_A$  de cada rodada.

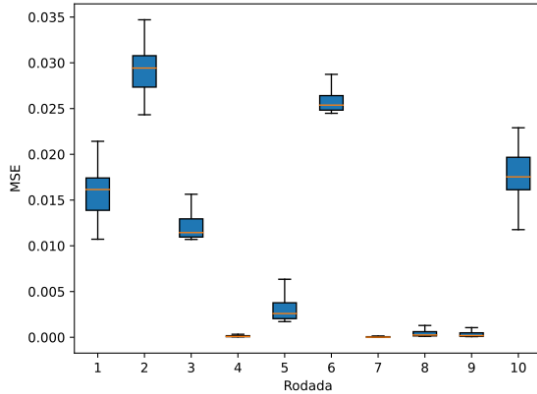


Figura 6. Boxplot de MSE por rodada com dados da base WINE.

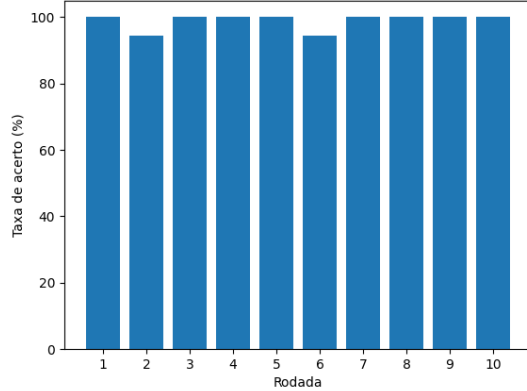


Figura 7.  $T_A$  para as rodadas sobre a base WINE.

É perceptível pela análise de ambas figuras o pior desempenho das rodadas 2 e 6, sendo as únicas que não conseguiram atingir 100% de acerto na classificação, alcançando 94,44%. As rodadas 1, 3, 5 e 10, embora tenham atingido  $T_A$  máxima, tiveram MSE mediano, sendo as rodadas 4, 7, 8 e 9 as melhores indicadas e com variação de MSE muito baixa. Essa pequena variação pode ser melhor observada pelo gráfico de erro do melhor modelo, que corresponde à rodada 7 e é mostrado na Figura 8.

Percebe-se o rápido decaimento do erro ainda nas épocas iniciais tanto para os dados de treino como validação, chegando a quase se juntarem ao final. Neste caso, a cada época

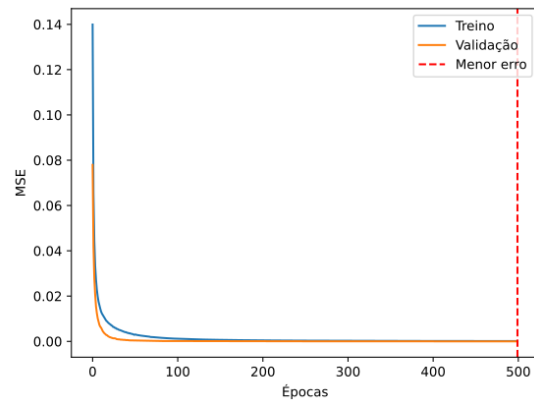


Figura 8. MSE por iteração de treino na base WINE.

havia redução do MSE, ainda que mínima, levando como critério de parada o limite máximo de épocas configurada igual a 500. Para esse modelo, a  $T_A$  é de 100%, com MSE de  $5,7030 \cdot 10^{-6}$ , se apresentando como modelo genérico ideal para classificar os dados deste caso.

#### IV. CONCLUSÃO

Duas questões referentes ao uso de MLP para problemas de classificação foram explicadas e respondidas. Na primeira, foi analisado o comportamento de três portas lógicas comuns e pode-se concluir que a MLP tem potencial para solucionar todas, inclusive o caso XOR, o que não é possível usando estruturas mais simples como Perceptron ou Adaline [6]. Já na segunda questão, modelos com alta capacidade de generalização foram encontrados para ambas bases de dados aplicadas, com valores expressivos de acerto e baixos valores de erro, demonstração importante da habilidade que esta arquitetura possui.

#### REFERÊNCIAS

- [1] SIQUEIRA, H. V. **Máquinas Desorganizadas para Previsão de Séries de Vazões**. 2013. 244 f. Tese (Doutorado) - Curso de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas. Campinas, 2013.
- [2] SIQUEIRA, H. V. **Lista 2**. Disponível em: <http://paginapessoal.utfpr.edu.br/hugosiqueira/disciplinas-mestrado/redes-neurais-artificiais/Lista%20%20-%20Redes%20Neurais.pdf>. Acesso em: 12 mar. 2021.
- [3] DUA, D. GRAFF, C. **UCI Machine Learning Repository**. Disponível em: <http://archive.ics.uci.edu/ml>. Irvine, CA: University of California, School of Information and Computer Science. Acesso em: 12 abr. 2021.
- [4] RUSSELL, S; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3. ed. New Jersey: Prentice Hall, 2009.
- [5] JAMES, G. et al. **An Introduction to Statistical Learning: with applications in R**. 1. ed. New York: Springer, 2013.
- [6] COSTA, L. F. P. **1ª Lista de Exercícios Computacionais - Resolução**. Disponível em: [https://mega.nz/file/W6hykL6R#Q9O2wuLtPB\\_CJWW-6RBVf7yXk9T5M6KqmjXMLGhq-NM](https://mega.nz/file/W6hykL6R#Q9O2wuLtPB_CJWW-6RBVf7yXk9T5M6KqmjXMLGhq-NM). Acesso em: 19 abr. 2021.