



# Classificação de dados empregando rede neural perceptron multicamadas

## *Data classification using multilayer perceptron neural network*

Daniel Souza Batista\*, Hugo Valadares Siqueira\*, Yara de Souza Tadano†, Thiago Antonini Alves<sup>+</sup>

### RESUMO

Neste artigo foram abordados conceitos sobre a perceptron empregando a linguagem de programação Python. Uma base de dados específica do repositório *UCI machine learning* sobre as características dos vinhos foi então selecionada para realização de uma solução de classificação de dados. Os resultados obtidos foram de 90% na precisão da rede, o que é um valor adequado, visto que problemas de desproporcionalidade no dataset foram analisados.

**Palavras-chave:** Python, rede neural, características dos vinhos, classificação.

### ABSTRACT

In this paper, concepts about the multilayer perceptron network were discussed, which was implemented using the Python programming language. A specific database from the UCI machine learning repository about wine characteristics was then selected to carry out a data classification solution. The results obtained were 90% in network accuracy, which is an adequate value, since dataset disproportionality problems were analyzed.

**Keywords:** Python, neural network, wine characteristics, classification.

## 1 INTRODUÇÃO

A rede perceptron na sua versão clássica é constituída de apenas uma camada neural, tendo somente um neurônio na camada. Tal modelo é usado para dividir duas classes linearmente separáveis, apresentando somente uma saída. A rede perceptron pertence à arquitetura *feedforward* de camadas únicas, pois o fluxo de informações em suas estruturas reside sempre no sentido da camada de entrada em direção à camada neural de saída, inexistindo-se qualquer tipo de realimentação de valores produzidos pelo seu único neurônio (BROWLEE; JASON, 2019, p.76). As redes de múltiplas camadas caracterizam-se pelas elevadas possibilidades de aplicações em diversos tipos de problemas relacionados com as diferentes áreas do conhecimento, considerada uma das arquiteturas mais versáteis quanto a aplicabilidade. Nesta categoria destaca-se a rede perceptron de múltiplas camadas (*multilayer perceptron* - MLP).

Observando a Figura 1, vê-se que cada uma das entradas representa os sinais adivindos de determinada aplicação. Nesse caso, as saídas dos neurônios da primeira camada neural escondida serão as próprias entradas

\* Departamento Acadêmico de Engenharia Elétrica, [danielbatistawjk@gmail.com](mailto:danielbatistawjk@gmail.com), [hugosiqueira@utfpr.edu.br](mailto:hugosiqueira@utfpr.edu.br)

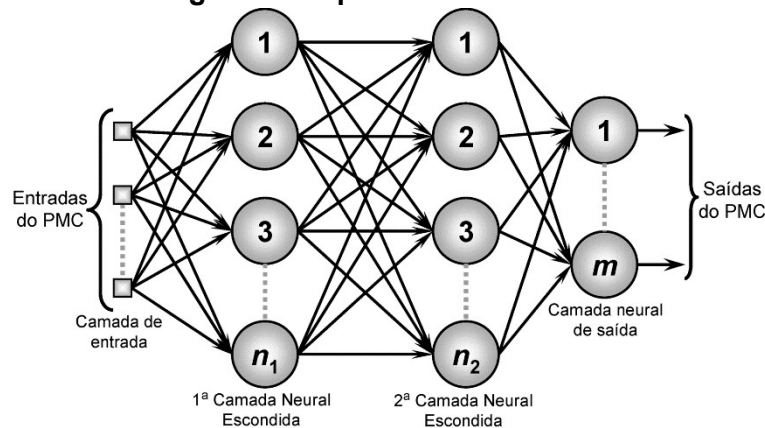
† Departamento Acadêmico de Matemática, [yaratadano@utfpr.edu.br](mailto:yaratadano@utfpr.edu.br)

<sup>+</sup> Departamento Acadêmico de Engenharia Mecânica, [antonini@utfpr.edu.br](mailto:antonini@utfpr.edu.br)

Universidade Tecnológica Federal do Paraná, Campus PG (Ponta Grossa);

daqueles neurônios pertencentes à segunda camada. Após o sinal ser passado por todas as camadas escondidas, as saídas são computadas pela camada de saída, gerando a resposta da rede na iteração corrente.

**Figura 1: Arquitetura de uma rede MLP**



**Fonte:** SILVA; SPATTI; FLAUZINO, 2010

Diferente da rede de uma camada, além da presença de camadas escondidas na topologia da MLP, observase que a camada neural de saída pode ser composta por diversos neurônios sendo que cada um desses partes do processo a ser mapeado. Sendo assim, se no processo consiste  $n$  saídas, a rede terá  $n$  neurônios em sua última camada neural.

O ajuste dos pesos e do limiar de cada neurônio é efetuado utilizando-se o processo de treinamento supervisionado. O algoritmo de aprendizado mais conhecido e usual no ajuste dos pesos de uma MLP é denominado *backpropagation* ou retro propagação do erro.

“O treinamento dos pesos sinápticos é executado de forma supervisionada, já que um sinal de referência é utilizado. Os dados no processo de aprendizagem são divididos em três partes: treinamento, validação e teste” (SILVA; SPATTI; FLAUZINO, 2010, p100). Explicam o algoritmo de aprendizagem *backpropagation* o qual mostra o cálculo da derivada da função custo para posterior utilização do método do gradiente ótimo. A etapa mais criteriosa e delicada é o treinamento, no qual se utilizam 60 a 90% Dos dados disponíveis.

A saída da rede é comparada com A esperada de cada amostra e, em seguida, é calculado o erro quadrático médio (EQM). Após, o algoritmo de retropropagação do erro será aplicado para calcular a derivada da função de custo com base no EQM, e o método do gradiente ótimo será usado para reajustar os pesos da rede.

“O processo de validação cruzada ocorre junto com a fase de treinamento, tendo como objetivo maximizar a generalização.” (SIQUEIRA, et al., 2012).

## 2 METODOLOGIA

Neste estudo, foi abordado um problema real de classificação de padrões. Uma base de dados específica do repositório *UCI machine learning* sobre as características dos vinhos foi então selecionada. Esses dados são os resultados de uma série de análises químicas de vinhos produzidos na mesma região, na Itália, mas derivados de três cultivares diferentes. As análises realizadas pelos autores da base de dados determinam a presença de 13 diferentes constituintes encontrados nos 3 tipos de vinhos. A base de dados possui 178.



Para satisfazer as premissas estabelecidas neste projeto, uma rede neural MLP foi construída em linguagem de programação Python, a fim de concretizar todos os conhecimentos adquiridos neste período.

Os conjuntos possuem a seguinte proporção: treinamento e validação, com 70 % das amostras e teste com 30%. O software utilizado para essa implementação foi o Spyder (Anaconda), uma ferramenta crescente na criação de redes neurais em Python.

Ressalta-se a utilização de bibliotecas de programação. Elas facilitam implementação e ajustes, sendo mais prático e intuitivo para os usuários que desejam ingressar na área. Neste trabalho, foram empregadas as bibliotecas Keras e sklearn.

Foram feitos 20 testes para cada amostra de vinho desconhecidas, com análise estatística dos seus resultados após 5000 épocas de treinamentos. As redes montadas com três camadas ocultas, ou seja, arquitetura [13 13 13 1] foram as que obtiveram os melhores resultados para a classificação de origem das amostras aleatórias de vinho.

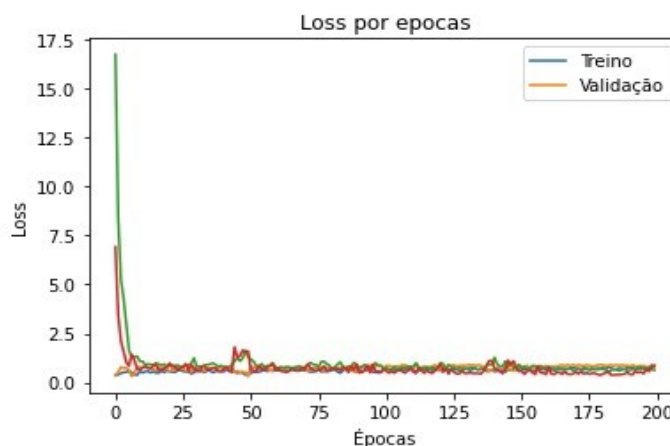
### 3 RESULTADOS

O treinamento de uma rede MLP se torna eficiente geralmente quando calculada pela medida do EQM durante as etapas de validação e treinamento. Nesse contexto um fato importante se refere ao parâmetro de taxa de aprendizado que define quão rápido vai ser o processo de ajuste dos pesos entre todas as camadas de neurônios da rede neural. Utilizando uma taxa de aprendizado igual a 0.01, a rede apresentou uma quantidade significativa de perda de generalização, por volta de 1500 épocas de treinamento.

Isso implica que a rede não foi capaz de inferir com uma certa precisão a resposta dada pela sua camada de saída, elevando o EQM em dado ponto do aprendizado. Isto ocasionou a ação chamada de sobre treinamento, indicando que a rede foi treinada em excesso ou perdeu a capacidade de generalização enquanto os pesos sinápticos se atualizavam. Mudando-se a taxa de aprendizado para 0,1, observamos que a rede consegue até o limite uma quantidade de épocas entorno de 5000, manter a sua capacidade de generalização, o que se reflete para as etapas de treinamento e validação um EQM cada vez menor.

Para analisar o comportamento da rede neural, foram plotados dois tipos de gráficos. A Loss (erro) por época e acurácia por época. Estes gráficos foram comparados na aprendizagem e validação.

**Figura 2: resultados do treinamento e validação da rede: EQM e Acurácia.**



Fonte: autoria própria.



É notório que a precisão variou, mas no final a precisão da validação e treinamento se mantiveram na casa dos 90% como melhor resultado, o que é um valor considerável, visto a desproporcionalidade do dataset. A acurácia na validação também variou de forma significativa e no final da validação manteve um valor consideravelmente bom para a rede.

#### 4 CONCLUSÃO

Nesse estudo fez-se a implementação computacional de uma rede neural do tipo perceptron multicamadas (MLP) para classificação de dados de vinhos.

Os resultados mostram que a rede teve um desempenho adequado, alimentada com um banco de dados referente para a determinação da origem de vinhos. As etapas de validação se mostraram eficientes para a taxa de aprendizado igual a 0,1. Constatou-se que analisar os dados e a confiabilidade dos mesmos influenciam nos resultados. Por isso a importância da paridade. O desbalanço de dados provoca um treinamento sem conformidade da rede e conseqüentemente leva a instabilidade.

Como trabalhos futuros pretende-se estudar e implementar novas arquiteturas de redes, bem como testar as mesmas em bases de dados diversas.

#### AGRADECIMENTOS

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela bolsa de Iniciação Tecnológica concedida ao acadêmico.

#### REFERÊNCIAS

- DA SILVA, I.N., SPATTI D., E FLAUZINO, R. **Redes neurais artificiais para engenharia e ciências aplicadas: curso prático**, Artliber Editora Ltda, São Paulo, SP, Brasil, 2010.
- HAYKIN, S., **Neural Networks: A Comprehensive Foundation**. Prentice Hall, Upper Saddle River, 1999..
- NIGRIN, A. **Neural Networks for Pattern Recognition**, Cambridge, MA: The MIT Press, 1993.
- S. MORO, P. CORTEZ AND P. RITA. **A Data-Driven Approach to Predict the Success of Bank Telemarketing**. Decision Support Systems, Elsevier, 62:22-31, June 2014.
- HUANG, G.-B., ZHOU, H., DING, X., ZHANG, R. **Extreme Learning Machines for Regression and Multiclass Classification**. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 42, no. 2, pp.513-529, 2012.
- SIQUEIRA, H. V., L. BOCCATO, R. ATTUX, AND C. LYRA FILHO. **Echo State Networks and Extreme Learning Machines: a Comparative Study on Seasonal Streamflow Series. Prediction**. Lecture Notes in Computer Science, p. 491-500, 2012.
- DUA, D. & GRAFF, C. **UCI Machine Learning Repository** [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science, 2019.