

Comparação de estruturas de redes neurais aplicadas a classificação de tipos de vinho

Rodrigo Gomes Dutra¹ e Juliano Miranda²

¹ ²Faculdade de Engenharia Elétrica e Biomédica, Universidade Federal do Pará,
Av. Augusto Correa 01, Belém, Pará 66075-090, Brasil

Autor Correspondente¹: dutra.rgdgd@gmail.com

Autor Correspondente²: juca.m

Resumo

O artigo detalha a construção e o uso de 2 estruturas de rede, ambas aplicadas em um problema de classificação. A primeira estrutura baseia-se no modelo de rede Perceptron de múltiplas camadas (MLP), a outra estrutura é chamada de rede autoassociativa competitiva, também chamada de autoencoder competitiva. Essas estruturas foram aplicadas na classificação de tipos de vinho, visando-se assim comparar o desempenho e acurácia de ambas topologias no problema de classificação proposto. Ambas estruturas foram construídas utilizando software Python [3] juntamente com a biblioteca Keras, o código utilizado na produção dos resultados está disponibilizado no github.

Palavras-chave: Redes Neurais Artificiais. Redes Auto Associativas. Vinho. MATLAB.

1 INTRODUÇÃO

As Redes Neurais Artificiais (RNA) tratam-se de modelos matemáticos construídos a partir do conhecimento do neurônio biológico, tendo como objetivo mimetizar o comportamento da células nervosa [1]. Assim, simulando a característica de um sistema nervoso, de forma a ter capacidade de processar múltiplas entradas, reconhecer e classificar padrões. Dessa forma, RNA tem grande capacidade de aplicação em vários tipos de contexto.

A aplicação proposta nesse artigo trata-se de classificação de tipos de vinho. O banco de dados utilizado foi retirado de *UCI machine learning repository* [2]. Esse banco é composto de 179 amostras de 13 atributos de entrada distribuídos em 3 classes, as quais são o tipo de vinho dado a combinação dos valores dos 13 atributos de entrada.

Dessa maneira tendo-se em vista a o conjunto de dados de dados aplicado, foram aplicadas 2 topologias distintas de RNA no problema de classificação proposto, com o intuito de compara-las em termos de acurácia e desempenho. As topologias utilizadas foram rede do tipo MLP e autoassociativa competitiva, as quais serão discutidas e analisadas e caracterizadas mais adiante nesse artigo.

1.1 Metodologia

Para o fim proposto, foi usado o Software Python(3.65)[3] juntamente com a biblioteca Keras[4] para gerar as RNA.

1.2 Objetivos

O objetivo desse trabalho é primeiramente validar as topologias de redes apresentadas neste, aplicando-as em um problema de classificação. Feito isso, o trabalho também tem como objetivo comparar o desempenho das 2 estruturas de RNA para o problema proposto.

1.3 Organização do Trabalho

A organização do artigo foi estruturado da seguinte forma: Na seção 2 encontra-se a base teórica de RNA e a apresentação dos 2 modelos utilizados; A seção 3 detalha o banco de dados e seu uso dentro do artigo; A seção 4 detalha o desenvolvimento e aplicação da MLP; A seção 5 detalha o desenvolvimento e aplicação da rede autoencoder competitiva; A seção 6 apresenta os resultados; A seção 7 dispõe a conclusão.

2 ESTRUTURAS DE RNA

RNA podem apresentar vários tipos de estrutura, dado sua capacidade de ligar neurônios artificiais. Como estruturas de sistemas nervosos biológicos, cada estrutura distinta de RNA pode ser especializada em problemas específicos. Nesse sentido vale-se a discussão que não necessariamente uma rede com uma quantidade grande de neurônios e ligações será melhor que uma rede com poucas ligações e menor números de neurônios dado alguma aplicação específica. Um neurônio artificial é modelado tendo terminais de entrada x e um terminal de saída. O comportamento das sinapses é simulado através dos pesos

aplicados nas entradas do neurônio, a saída do neurônio é dada por :

$$S_j = \theta * \sum_{i=0}^m (w_{ij} * x_i + b_j) \quad (1)$$

Onde θ representa a função de ativação, b_j corresponde ao *bias*, p_{ij} são os pesos sinápticos[5].

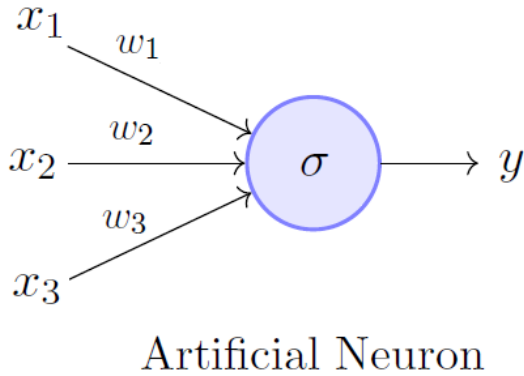


Figura 1. Estrutura de um neurônio artificial

De forma geral uma rede neural é constituída de unidades simples de processamento denominadas de neurônios e ligações massivas entre essas unidades.[5] Mais a frente nessa seção será detalhado a constituição de Redes MLP e anociassociativos competitivas

2.1 Perceptron de Múltiplas Camadas

A rede perceptron de múltiplas camadas como o próprio nome indica apresenta várias camadas em sua estrutura, tendo como composição mínima uma camada de entrada, uma camada de saída e uma camada intermediária. Segundo o modelo descrito por [5], essas camadas são ligadas por sinapses com pesos (p_{ij}) intrínsecos, assim compondo uma rede vasta de interconexões regidas por pesos e funções de ativação.

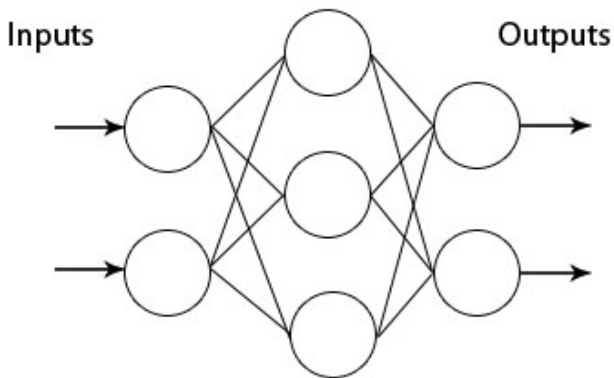


Figura 2. Estrutura de um neurônio artificial

O aprendizado da MLP consiste em 2 passos, o passo direto e o passo inverso. O passo direto ou *forward pass* consiste em passar as entradas pela rede, assim aplicando os pesos associados as sinapses, e obtendo assim a saída.

O segundo passo consiste na retropropagação do erro, é calculado o gradiente da função de perda na camada de saída, posteriormente esse gradiente é utilizado para aplicar recursivamente a regra da cadeia para atualizar os pesos de toda a rede. Esse algoritmo é chamado de *backpropagation*[1].

2.2 Redes Autoassociativas competitivas

Redes neurais do tipo autoassociativas ou *autoencoders* baseiam-se em MPL que apresentam o mesmo número de neurônios na camada de entrada e na camada de saída, a camada intermediária apresenta um número inferior de neurônios em relação as outras duas camadas.

A estrutura da camada de entrada juntamente com a camada intermediária funcionam como um encoder, também chamada de gargalo, comprime a informação inserida na entrada.

A estrutura de saída juntamente com a camada intermediária funciona como um decoder, descomprimindo a informação obtida da camada intermediária. Dessa maneira, a rede tenta recriar os dados que foram inseridos na entrada[6].

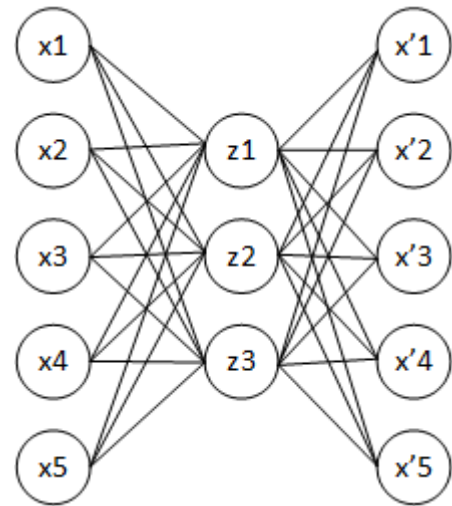


Figura 3. Estrutura de rede autoencoder

A estrutura competitiva baseia-se em criar e treinar um numero de redes autoencoders igual ao número de classes do problema de classificação, de forma a criar redes especializadas em recriar os dados de entrada de inseridos para cada classe. Dessa forma, as redes treinadas para uma dada classe irão apresentar um erro médio quadrático menor ao serem inseridos dados da classe a qual foram treinadas.

Após o treinamento essas redes são colocadas lado a lado, recebendo as mesmas entradas e tentando recria-las. Mede-se o erro quadrático médio de cada rede para cada índice de entrada, os valores de erro são avaliados e infere-se a classe de saída correspondente a rede que apresentou o menor erro na recriação dos dados de entrada[6]. Dessa maneira as redes competem entre si para assim gerar um valor de classe na saída.

3 BANCO DE DADOS

A base de banco de dados de vinho selecionada para a produção desse artigo foram resultados de uma análise química de vinhos cultivados na mesma região da Itália, mas derivados de três diferentes cultivares[2].

Os atributos de entrada da rede são Álcool, Acido málico, Cinzas, Alcalinidade das cinzas, Magnésio, Fenóis totais, Flavonoides, Fenóis não flavonoides, Proantocianidinas, Intensidade da cor, tonalidade, OD280 / OD315 de vinhos diluídos e Prolina.

A base de dados foi organizada de forma que cada grupo de 13 variável tem um índice associado e uma classe correspondente.

Tabela 1
Distribuição da base de dados

Faixa de Index	Classe correspondente
0-58	1
58-129	2
131-77	3

A divisão da base de dados em dados de treino, validação e teste foi realizada com a biblioteca Scikit[7], a qual randomiza a seleção de índice para a divisão do banco de dados em dada proporção previamente programada. A proporção utilizada para a divisão consta na Tabela 2.

Tabela 2
Divisão da base de dados

Distribuição dos dados	Classe 1		Classe 2		Classe 3		Total de Dados
	n	%	n	%	n	%	
Treino	35	60%	44	60%	27	60%	106
Validação	12	20%	14	20%	10	20%	36
Teste	12	20%	14	20%	10	20%	36

4 DESENVOLVIMENTO DA REDE MLP

A rede MLP foi desenvolvida em Python[3] com o auxílio da bibliotecas Keras[4]. A divisão de dados em treino validação e teste foi seguido como está proposto na Tabela 2, e apesar de que a escolha dos índices de conjunto de entradas é feito de forma randômica, foi fixado duma *seed* na biblioteca Scikit[7], de forma a sempre ser executada a mesma divisão de dados.

A estrutura da rede apresenta 3 camadas, na camada de entrada existe 13 neurônios para receber os dados, a camada de saída dispõe de apenas 1 neurônio. Para definir o número ótimo de neurônios na camada intermediária foi realizado um levantamento da acurácia com relação a esse número.

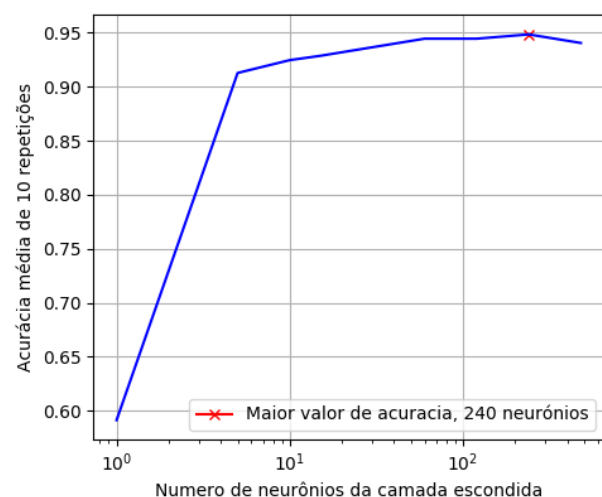


Figura 4. Acurácia vs Nº de neurônios na camada escondida

Depois de definido o número de neurônios da camada escondida, os dados utilizados no treino validação e teste foram normalizados, de forma que cada coluna dos 13 atributos foi dividida pelo maior número da coluna.

A função de ativação da camada escondida foi a *sigmoid*, e a função de ativação da camada de saída foi linear, a escolha das funções foi feita depois de vários testes de desempenho relacionando o erro com a função de ativação escolhida.

A saída da rede foi arredondada para o inteiro mais próximo, por conta que o este tratava-se de um número fracionário. Dessa forma, após arredondar o valor de saída se enquadrava em alguma das 3 classes presentes no banco de dados.

5 RESULTADOS

Pulvinar nullam felis vitae habitasse egestas aliquam senectus lobortis aptent, venenatis felis suspendisse aliquam torquent per sed leo donec, lobortis leo blandit leo aliquam nostra justo quis. leo praesent cras ultrices gravida velit ac suspendisse pharetra, hac scelerisque felis aliquet mattis velit placerat leo nibh, purus ad leo lacus eu sociosqu proin. ad ante felis hac lorem interdum tellus integer cursus ipsum pharetra torquent, per donec aenean sodales potenti fringilla arcu ultricies et. mattis tempus tempor sem primis ultricies curae, urna metus dictumst lectus tempor, lobortis aliquam netus egestas dictum.

Varius erat fames feugiat pellentesque eros laoreet lacinia integer condimentum sed vulputate, integer condimentum eleifend maecenas dapibus viverra nunc rhoncus eros. volutpat consectetur dictum purus etiam augue ipsum litora consectetur commodo mi, commodo est cursus sem ac cubilia conubia in at. ligula fermentum nisl erat sed posuere ac fusce sociosqu, porttitor vulputate tellus platea duis tristique tellus. vehicula varius pellentesque nam sapien aptent, et vivamus eget turpis ac, sit bibendum eros varius. venenatis ultrices duis sollicitudin aliquam pellentesque sagittis elit sociosqu ut, lorem consectetur adipiscing eu elementum ipsum nullam eros, egestas ligula dapibus congue est ornare vitae hendrerit.

5.1 Resultados

Pulvinar nullam felis vitae habitasse egestas aliquam senectus lobortis aptent, venenatis felis suspendisse aliquam torquent per sed leo donec, lobortis leo blandit leo aliquam nostra justo quis. leo praesent cras ultrices gravida velit ac suspendisse pharetra, hac scelerisque felis aliquet mattis velit placerat leo nibh, purus ad leo lacus eu sociosqu proin. ad ante felis hac lorem interdum tellus integer cursus ipsum pharetra torquent, per donec aenean sodales potenti fringilla arcu ultricies et. mattis tempus tempor sem primis ultricies curae, urna metus dictumst lectus tempor, lobortis aliquam netus egestas dictum.

Varius erat fames feugiat pellentesque eros laoreet lacinia integer condimentum sed vulputate, integer condimentum eleifend maecenas dapibus viverra nunc rhoncus eros. volutpat consectetur dictum purus etiam augue ipsum litora consectetur commodo mi, commodo est cursus sem ac cubilia conubia in at. ligula fermentum nisl erat sed posuere ac fusce sociosqu, porttitor vulputate tellus platea duis tristique tellus. vehicula varius pellentesque nam sapien aptent, et vivamus eget turpis ac, sit bibendum eros varius. venenatis ultrices duis sollicitudin aliquam pellentesque sagittis elit sociosqu ut, lorem consectetur

adipiscing eu elementum ipsum nullam eros, egestas ligula dapibus congue est ornare vitae hendrerit.

5.2 Resultados

Pulvinar nullam felis vitae habitasse egestas aliquam senectus lobortis aptent, venenatis felis suspendisse aliquam torquent per sed leo donec, lobortis leo blandit leo aliquam nostra justo quis. leo praesent cras ultrices gravida velit ac suspendisse pharetra, hac scelerisque felis aliquet mattis velit placerat leo nibh, purus ad leo lacus eu sociosqu proin. ad ante felis hac lorem interdum tellus integer cursus ipsum pharetra torquent, per donec aenean sodales potenti fringilla arcu ultricies et. mattis tempus tempor sem primis ultricies curae, urna metus dictumst lectus tempor, lobortis aliquam netus egestas dictum.

Varius erat fames feugiat pellentesque eros laoreet lacinia integer condimentum sed vulputate, integer condimentum eleifend maecenas dapibus viverra nunc rhoncus eros. volutpat consectetur dictum purus etiam augue ipsum litora consectetur commodo mi, commodo est cursus sem ac cubilia conubia in at. ligula fermentum nisl erat sed posuere ac fusce sociosqu, porttitor vulputate tellus platea duis tristique tellus. vehicula varius pellentesque nam sapien aptent, et vivamus eget turpis ac, sit bibendum eros varius. venenatis ultrices dui sollicitudin aliquam pellentesque sagittis elit sociosqu ut, lorem consectetur adipiscing eu elementum ipsum nullam eros, egestas ligula dapibus congue est ornare vitae hendrerit.

6 CONSIDERAÇÕES FINAIS

6.1 Discussão

Interpretação dos resultados

Odio lobortis egestas senectus risus mattis consequat auctor, porta primis massa vel nibh consectetur inceptos, ut aliquam dapibus phasellus eu dolor. urna lectus pulvinar ultricies pellentesque vitae augue potenti, aliquet libero morbi aliquam tellus dictumst litora tincidunt, rhoncus phasellus nostra justo metus massa. lectus malesuada sagittis non nibh porttitor habitasse lacus maecenas mattis, volutpat cubilia risus imperdiet scelerisque praesent dictum lorem habitasse, aptent vel nam egestas euismod massa fringilla ultricies. a at platea dui mollis varius in aenean volutpat fusce mi, suscipit justo nullam bibendum placerat molestie inceptos vulputate. aliquam nisl conubia pharetra porttitor feugiat nam at mi augue, egestas ut fames tempus mollis sit aliquam cras.

Pulvinar nullam felis vitae habitasse egestas aliquam senectus lobortis aptent, venenatis felis suspendisse aliquam torquent per sed leo donec, lobortis leo blandit leo aliquam nostra justo quis. leo praesent cras ultrices gravida velit ac suspendisse pharetra, hac scelerisque felis aliquet mattis velit placerat leo nibh, purus ad leo lacus eu sociosqu proin. ad ante felis hac lorem interdum tellus integer cursus ipsum pharetra torquent, per donec aenean sodales potenti fringilla arcu ultricies et. mattis tempus tempor sem primis ultricies curae, urna metus dictumst lectus tempor, lobortis aliquam netus egestas dictum.

Varius erat fames feugiat pellentesque eros laoreet lacinia integer condimentum sed vulputate, integer condimentum eleifend maecenas dapibus viverra nunc rhoncus eros. volutpat consectetur dictum purus etiam augue ipsum litora consectetur commodo mi, commodo est cursus sem ac cubilia conubia in at. ligula fermentum nisl erat sed posuere ac fusce sociosqu, porttitor vulputate tellus

platea duis tristique tellus. vehicula varius pellentesque nam sapien aptent, et vivamus eget turpis ac, sit bibendum eros varius. venenatis ultrices dui sollicitudin aliquam pellentesque sagittis elit sociosqu ut, lorem consectetur adipiscing eu elementum ipsum nullam eros, egestas ligula dapibus congue est ornare vitae hendrerit.

6.2 Conclusão

Dapibus gravida tristique sodales purus condimentum porttitor, aliquam vulputate condimentum donec sapien justo praesent, sociosqu pellentesque dictum eros auctor. odio amet sem pretium eros facilis curabitur velit tempus sapien, sodales praesent rutrum interdum tincidunt habitant euismod augue, tristique vehicula tempus molestie at quisque erat potenti. lacinia pulvinar class dictumst suspendisse eget etiam, molestie lectus class aenean purus eros primis, quam purus lectus viverra est. ante eget pretium lacus torquent cras ullamcorper neque, elit platea diam nulla potenti class auctor lectus, tempor dapibus a justo aptent rhoncus. praesent aliquet purus felis nostra pellentesque odio quisque praesent porttitor, curae maecenas placerat nostra maecenas erat ac tristique, iaculis porttitor habitant aptent suscipit posuere accumsan curabitur.

Himenaeos rutrum augue nec nunc vulputate senectus vel aptent blandit, curae pulvinar gravida enim condimentum pretium ante posuere vehicula, pellentesque ut dolor amet ante cras cubilia neque. laoreet aliquet rutrum eros mattis torquent curae habitasse, pulvinar turpis nulla convallis molestie netus tincidunt, habitant et ut integer inceptos massa. sapien etiam sed posuere viverra ullamcorper rutrum euismod, platea netus imperdiet ultrices feugiat lectus sit, cursus rutrum tincidunt mollis risus ligula. dui quisque sapien tellus curabitur proin lacus proin lorem, magna aliquam adipiscing dictum leo consequat nisl orci, etiam vitae mi eros augue mauris imperdiet.

6.3 Trabalhos Atuais e Sugestões Para Trabalhos Futuros

Dapibus gravida tristique sodales purus condimentum porttitor, aliquam vulputate condimentum donec sapien justo praesent, sociosqu pellentesque dictum eros auctor. odio amet sem pretium eros facilis curabitur velit tempus sapien, sodales praesent rutrum interdum tincidunt habitant euismod augue, tristique vehicula tempus molestie at quisque erat potenti. lacinia pulvinar class dictumst suspendisse eget etiam, molestie lectus class aenean purus eros primis, quam purus lectus viverra est. ante eget pretium lacus torquent cras ullamcorper neque, elit platea diam nulla potenti class auctor lectus, tempor dapibus a justo aptent rhoncus. praesent aliquet purus felis nostra pellentesque odio quisque praesent porttitor, curae maecenas placerat nostra maecenas erat ac tristique, iaculis porttitor habitant aptent suscipit posuere accumsan curabitur.

ACKNOWLEDGMENT

7 PROOF OF THE FIRST ZONKLAR EQUATION

Appendix one text goes here.

8

Appendix two text goes here.

REFERÊNCIAS

- [1] Kovács, Z. Redes neurais artificiais. (Editora Livraria da Física, 2002)
- [2] Blake, C. & Merz, C. UCI repository of machine learning databases, 1998.
- [3] Rossum, G. Python reference manual. (CWI (Centre for Mathematics and Computer Science, 1995)
- [4] Chollet, F. Keras: Deep learning library for theano and tensorflow. *Url: <https://keras.io/k.7>*, T1 (201)
- [5] Haykin, S. Neural networks: a comprehensive foundation. (Prentice Hall PT, 1994)
- [6] A, V., Castro, A. & Lima, S. Diagnosing faults in power transformers with autoassociative neural networks and mean shift. *Ieee Transactions On Power Delivery*. **27**, 1350–1357 (2012)
- [7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. & Dubourg, V. Scikit-learn: Machine learning in Python. *Journal Of Machine Learning Research*. **12**, 2825–2830 (201)