

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. Classificação. Vinho. Python. Keras.

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 :

$$Sj = \theta * \sum_{i=0}^m (wij * xi + bj) \quad (1)$$

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

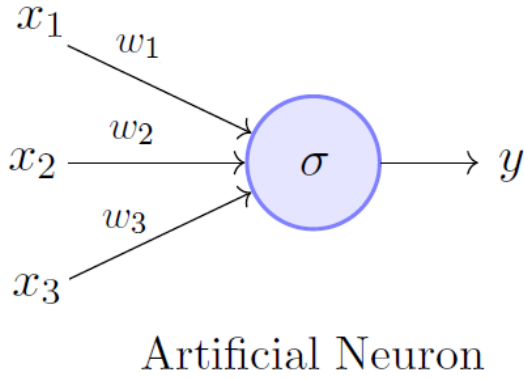


Figura 1. Estrutura de um neurônio artificial

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.

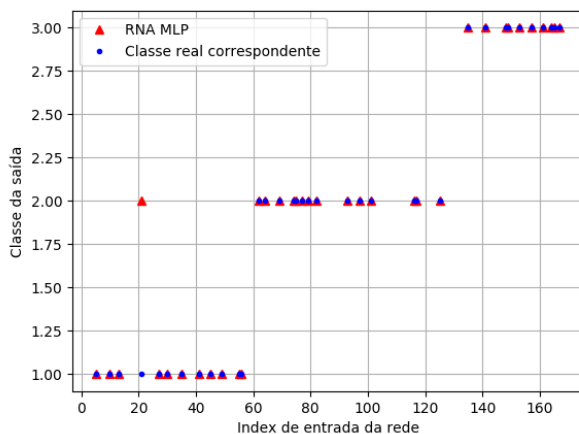


Figura 2. Estrutura de um neurônio artificial

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].

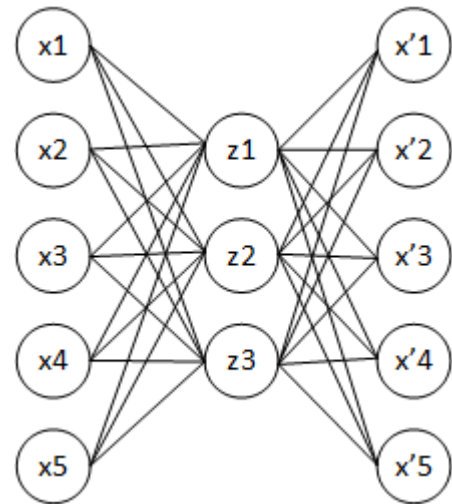


Figura 3. Estrutura de rede autoencoder

A estrutura competitiva baseia-se em criar e treinar um número 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 recriá-las. Posteriormente, 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, Ácido 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 variáveis 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 n %		Classe 2 n %		Classe 3 n %		Total de Dados
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.

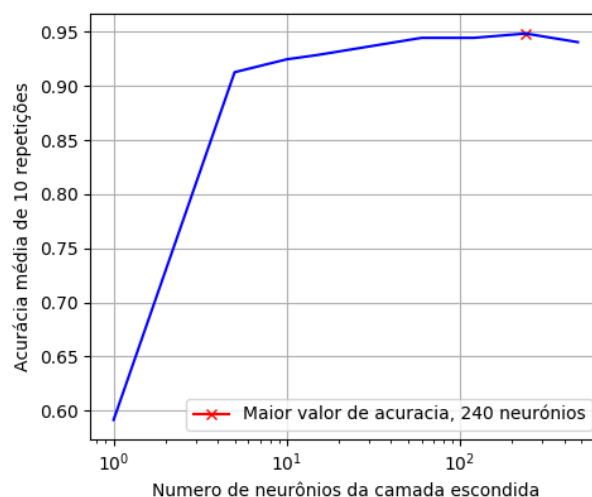


Figura 4. Acurácia vs Nº de neurônios na camada escondida

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 escolhida foi *sigmoid*, 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 realizar a aproximação, o valor de saída enquadra-se em alguma das 3 classes presentes no banco de dados.

5 DESENVOLVIMENTO DA REDE AUTOENCODER COMPETITIVA

Assim como a Rede MLP, a rede autoencoder competitiva foi desenvolvida utilizando os mesmo software e bibliotecas. A divisão dos dados de entrada em treino validação e teste ocorreu de forma idêntica a rede MLP construída anteriormente.

Feito a divisão de dados de treino de todas as 3 classes, essas foram criadas 3 MLP conectadas de forma competitiva[6]. Essas redes contavam com 13 neurônios na camada inicial e na camada de saída, na camada intermediária foi executado a análise do número de neurônios versus a acurácia total da rede competitiva para descobrir o valor ideal de neurônios na camada intermediária.

Tabela 3
Acurácia versus Nº de neurônios

Nº de neurônios na camada escondida	Acurácia média % de 10 repetições
1	94.44444444
4	100
8	91.11111111
12	69.16666667

A função de ativação escolhida visando a maior acurácia da rede foi linear tanto na camada escondida quanto na camada de saída.

6 RESULTADOS

Após Realizar o treinamento das 2 estruturas de redes propostas, as redes foram colocadas a prova a fim de avaliar seus resultados com os mesmos dados de treino validação e teste, conforme exposto na seção 3 sobre o banco de dados.

6.1 Resultados MLP

Depois de selecionado o número ideal para teste dessa estrutura de RNA, foi executado a resposta da rede tendo-se como base dados de teste das 3 classes do banco de dados.

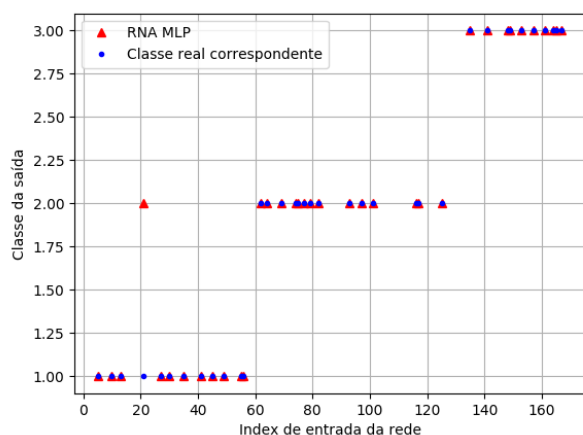


Figura 5. Resposta da rede MLP

Realizado 10 repetições de treino validação e teste da rede com os mesmos dados, a percentagem de acerto da rede atingiu a marca de 95.27777778%.

6.2 Resultados RNA autoassociativa competitiva

Feito a análise do melhor número de neurônios na camada escondida para a rede autoassociativa competitiva, cada RNA presente nessa estrutura apresentou um comportamento de erro quadrático médio diferente ao receber dados de teste das 3 classes de vinho presentes no banco de dados.

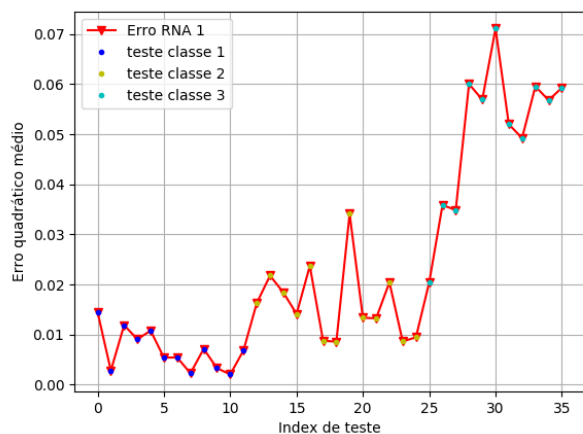


Figura 6. Erro da Rede 1 exposta os dados de treino

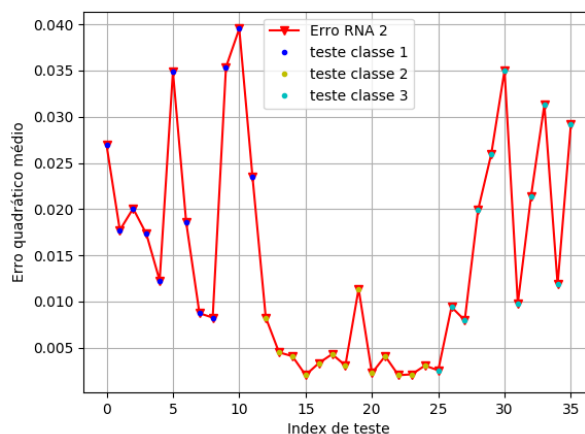


Figura 7. Erro da Rede 2 exposta os dados de treino

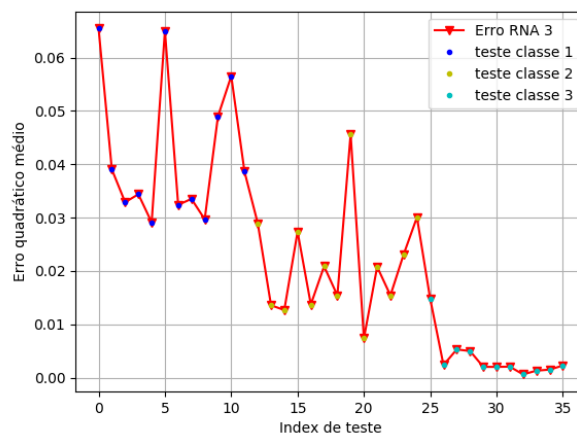


Figura 8. Erro da Rede 3 exposta os dados de treino

Nota-se que a Rede 1, Rede 2 e Rede 3 foram treinadas com bases de dados respectivamente da classe 1, 2 e 3.

A saída dessa estrutura de rede se beneficiou do comportamento do erro quadrático médio demonstrado acima, apresentando uma marca de 100% de acerto médio em 10 repetições de treino validação e teste, utilizando os mesmos dados de entrada.

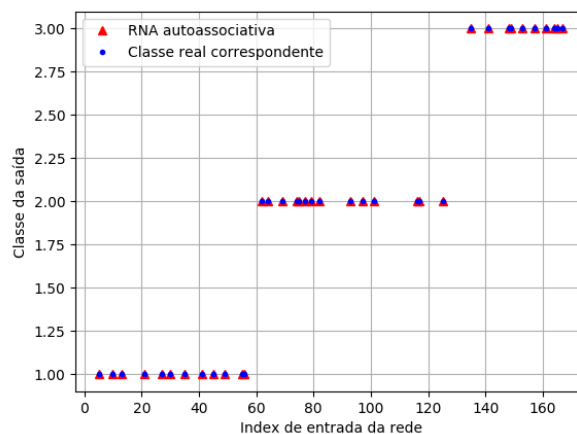


Figura 9. Saída da Rede Autoassociativa Competitiva

6.3 Comparativo do desempenho

Após realizados os testes de desempenho das 2 estruturas, o comparativo de desempenho de ambas foi disposto na Tabela 4.

Tabela 4
Comparativo de Desempenho

Estrutura da RNA	Acurácia média %	Tempo de execução(s)	Nº de Épocas
	de 10 repetições		utilizadas no treinamento
MLP	95.277778	3.908075	400
Autoassociativa	100	8.684611	3000 p/
competitiva			cada rede

Com base nos dados dispostos na Tabela 4, é possível afirmar que a Rede autoassociativa apresentou a marca impecável de 100% de acerto, devido ao comportamento das redes associadas, de forma que cada rede treinada com dados de apenas 1 classe desempenhava um papel de filtro com relação aos dados das outras classes. De forma que uma rede autoencoder que foi treinada apenas com dados de 1 classe apresentava erro quadrático médio menor quando comparado-se com as outras redes, assim, quando realizado a competição entre as 3 redes treinadas, a classe de saída era escolhida com uma maior acurácia quando comparado-se com o processo de escolha da de apenas uma MLP.

7 CONCLUSÃO

7.1 Discussão

Ambas estruturas RNA abordadas nesse artigo apresentaram um desempenho bastante satisfatório, com destaque da Rede autoassociativa competitiva que apresentou 100% de acerto, uma melhora de de quase 5% comparado com a estrutura MLP padrão. Para tanto, a rede *autoencoders* competitiva também apresentou um custo computacional mais elevado, com a utilização de 3000 épocas no treinamento e validação de cada 1 das 3 RNA *autoencoders*.

Dessa forma, para treinar redes dedicadas para cada classe, e nesse processo melhorar o desempenho de classificação a rede autocompetitiva foi mais custosa computacionalmente que a MLP padrão.

7.2 Conclusão

Com base no problema de classificação proposto, a rede autoassociativa competitiva apresentou um melhor desempenho na classificação dos tipos de vinho que a rede MLP padrão, com a ressalva do tempo de execução e maior número de épocas de treino validação e teste, o que indica que para outras aplicações na qual há uma maior necessidade em relação a rapidez de treino, ou em aplicações na qual não se dispõe de muito poder computacional, como aplicações em sistemas embarcados, a MLP padrão pode-se apresentar como uma solução melhor.

Tendo-se em vista também a quantidades de index de conjunto de dados do banco de dados analisados, é possível afirmar que o uso de um banco de dados maior poderia acarretar ressaltando o resultado obtido na análise desse artigo.

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)