
**APLICAÇÃO DE REDES NEURAIS RECORRENTES E PROFUNDAS EM TAREFAS
DE CLASSIFICAÇÃO**

**APPLICATION OF RECURRENT AND DEEP NEURAL NETWORKS IN
CLASSIFICATION TASKS**

**APLICACIÓN DE REDES NEURONALES PROFUNDAS Y RECORRENTES EN TAREAS
DE CLASIFICACIÓN.**

Lidio Mauro Lima de Campos

Professor Adjunto IV da Universidade Federal do Pará - UFPA na Faculdade de Computação / Instituto de Ciências Exatas e Naturais (ICEN). Doutor em Engenharia Elétrica - PPGGE da Universidade Federal do Pará.

limadecampos@gmail.com

<https://orcid.org/0000-0003-4315-829X>

Danilo Souza Duarte

Graduando em Sistemas de Informação pela Universidade Federal do Pará,

danilo.duarte8@hotmail.com

<https://orcid.org/0000-0001-9094-3392>

Editor Científico: José Edson Lara
Organização Comitê Científico
Double Blind Review pelo SEER/OJS
Recebido em 29.06.2019
Aprovado em 16.05.2020



Este trabalho foi licenciado com uma Licença Creative Commons - Atribuição – Não Comercial 3.0 Brasil

Resumo

Objetivo: O objetivo desta pesquisa é apresentar uma ferramenta que dê suporte ao processo de decisão para seleção de cultivares de vinho e avaliação de carros, por meio da utilização de RNAs multilayer perceptron, profundas e recorrentes.

Metodologia/abordagem: Realizaram-se experimentos, com o uso de Redes Neurais (RNs), variando-se uma série de parâmetros de treinamento das mesmas, ao final dos mesmos comparam-se as acurácias das RNs e selecionam-se os modelos com maior acurácia.

Originalidade/relevância: A originalidade da pesquisa diz respeito a utilização de RNs profundas para solução dos problemas de classificação, visto que melhor modelo de classificação é obtido pelo ajuste gradual dos parâmetros. A Proposta é relevante, pois permite classificar adequadamente cultivares de vinho e avaliar carros. Verificando-se a eficácia dos modelos e a melhor convergência, por meio do Modelo de Validação Cruzada

Principais Resultados: Os resultados elencados indicam a eficiência da técnica, para ambos os problemas, haja vista que a capacidade de generalização das RNAs testadas para o dataset wine foi em média de 85,58% utilizando a arquitetura de 3 camadas, 86,58% para a rede profunda e 93,53% para a rede recorrente, e para o dataset car evaluation foi em média de 93,71% utilizando a rede recorrente.

Contribuições teóricas/metodológicas: A pesquisa pode melhor evidenciar a capacidade de classificação de modelos de RNs profundas para fins de estimação adequada de parâmetros, de modelos baseados em inteligência artificial.

Palavras-chave: Redes Neurais Artificiais; Tomada de Decisão; Classificação.

Abstract

Purpose: The objective of this research is to present a tool that supports the decision process for selecting wine cultivars and evaluating cars, through the use of multilayer perceptron RNAs, deep and recurring.

Methodology / approach: experiments were carried out with the use of Neural Networks (NBs), varying a series of training parameters, at the end of them, the accuracy of the NBs is compared and the models with the greatest selection are selected. accuracy.

Originality / relevance: The originality of the research concerns the use of deep NBs to solve the classification problems, since the best classification model is obtained by the gradual adjustment of the parameters. The Proposal is relevant, as it allows to properly classify wine cultivars and evaluate cars. Checking the effectiveness of the models and the best convergence, through the Cross Validation Model.

Main results: The results listed indicate the efficiency of the technique, for both problems, considering that the generalization capacity of the RNAs tested for the dataset wine was on average 85.58% using the 3-layer architecture, 86.58% for the deep network and 93.53% for the recurring network, and for the dataset car evaluation it was an average of 93.71% using the recurring network.

Theoretical / methodological contributions: The research can better show the ability to classify models of deep neural networks for purposes of adequate parameter estimation, of models based on artificial intelligence.

Keywords: Artificial Neural Networks; Decision Making; Classification.

Resumen

Objetivo: El objetivo de esta investigación es presentar una herramienta que respalde el proceso de decisión para seleccionar cultivares de vino y evaluar automóviles, mediante el uso de ARN de perceptrón multicapa, profundos y recurrentes.

Metodología / enfoque: Se realizaron experimentos con el uso de redes neuronales (NB), variando una serie de parámetros de entrenamiento, al final de ellos, se compara la precisión de los NB y se seleccionan los modelos con la mayor selección exactitud.

Originalidad / relevancia: La originalidad de la investigación se refiere al uso de NB profundos para resolver los problemas de clasificación, ya que el mejor modelo de clasificación se obtiene mediante el ajuste gradual de los parámetros. La propuesta es relevante, ya que permite clasificar adecuadamente los cultivares de vino y evaluar los automóviles. Comprobación de la efectividad de los modelos y la mejor convergencia, a través del Modelo de Validación Cruzada.

Resultados principales: Los resultados enumerados indican la eficiencia de la técnica, para ambos problemas, considerando que la capacidad de generalización de los ARN probados para el vino del conjunto de datos fue en promedio 85.58% usando la arquitectura de 3 capas, 86.58% para la red profunda y el 93.53% para la red recurrente, y para la evaluación del conjunto de datos fue en promedio 93.71% usando la red recurrente.

Contribuciones teóricas / metodológicas: La investigación puede mostrar mejor la capacidad de clasificar modelos de redes neuronales profundas para propósitos de estimación adecuada de parámetros, de modelos basados en inteligencia artificial.

Palabras clave: Redes Neuronales Artificiales; Toma de decisión; Clasificación.

1 Introdução

As redes neurais artificiais (RNAs) são ferramentas bastante eficientes para o reconhecimento de padrões. As mesmas tem sido utilizadas de forma crescente para em diversos ramos de negócios: predição de séries temporais, (Szegedy *et al.*, 2014) reconhecimento humano usando a medida biométrica da orelha (Sánchez & Melin, 2014; Sánchez, Melin, & Castillo, 2015) predição de radiação da energia solar (Chatziagorakis *et al.*, 2014), visão por computador (Poonia, Tiwari, & Mishra, 2018), o reconhecimento de voz (Sak, Senior, & Beaufays, 2014), tradução automática (Sutskever, Vinyals, & Le, 2014), tomada de decisão no mercado financeiro (Gambogi, 2013), biometria (Dasgaonkar & Chopade, 2018), (Sharma, Agarwal, & Pandey, 2018) sistema de reconhecimento de vídeo (Kaushik, Gupta, & Bhatia, 2018), visão por computador (Poonia *et al.*, 2018), biometria (Dasgaonkar & Chopade, 2018), sistema de reconhecimento de vídeo (Kaushik *et al.*, 2018), análise de crédito bancário (Steiner *et al.*, 2018), dentre outras (Skrizhevsky, Sutskever, & Hinton, 2012; Wang & Kosinski, 2017).

As RNAs podem aprender a partir de exemplos, reconhecer um padrão escondido em observações históricas e usá-los para prever valores futuros. Além disso, são capazes de lidar

com informação incompleta ou dados ruidosos e podem ser muito eficazes, especialmente em situações onde não é possível definir as regras ou etapas que levam à solução de um problema. Possuem a capacidade de aprender ao interagir com seu ambiente e, dessa forma, extrair conhecimento do mesmo, o que é realizado por meio de um processo de treinamento, que consiste de um processo iterativo, todas essas características justificam a sua utilização, como ferramenta de apoio ao processo de tomada de decisão complexo (Garcia & Campos, 2019).

Assim sendo, o ajuste, correto, dos parâmetros de uma RNA *Multi Layer Perceptron* (número de neurônios na camada intermediária, taxa de aprendizado, *momentum*) é essencial para um bom treinamento e para a validação da capacidade de generalização da mesma. Por isso, antes de se usar o modelo em produção, precisa-se avaliar o seu desempenho, ou seja, é necessário medir quanto o mesmo é eficiente em prever dados que ainda não conheceu, ou seja, testá-lo de acordo com sua capacidade de generalização.

Um dos métodos mais comuns de avaliação do modelo é chamado *hold out*. No qual separam-se os dados originais entre treino e teste. Os dados de treino são submetidos a RNA e produzem o modelo. Após isso, os dados de teste são submetidos ao mesmo para que seja feita a previsão. Normalmente, esse modelo usa 70% dos dados para treino e 30% para testes. Se o desempenho foi satisfatório o modelo pode ser colocado em produção para classificar novos dados. Uma vantagem do método de *hold out* é que os dados são totalmente independentes; além disso, só precisa ser executado uma vez, por isso tem custos computacionais mais baixos. Uma desvantagem é que a avaliação do desempenho está sujeita a uma maior variância, dado o menor tamanho dos dados.

Na técnica *k-fold cross validation* o conjunto de dados de treinamento, é dividido em k subconjuntos. Desses k subconjuntos, um é selecionado para ser utilizado para validação e os $k-1$ restantes são utilizados no treinamento. O processo de validação é repetido k vezes, de modo que cada um dos k subconjuntos sejam utilizados pelo menos uma vez como dados de validação para o modelo. Após isso avalia-se o desempenho médio do modelo nos k testes.

O objetivo desse trabalho é apresentar os resultados da utilização de RNAs em tarefas de classificação, especificamente em dois problemas de classificação. O primeiro é relacionado a origem de vinhos italianos, em que a partir da análise química dos vinhos é descoberta a sua origem. O segundo é sobre um modelo de decisão de aceitabilidade de carros, no qual se faz a análise do preço médio e das características técnicas do carro.

Na realização dos experimentos foram utilizadas diferentes arquiteturas de RNAs, dentre elas a *feedforward*, a rede neural profunda e a rede recorrente. Para a rede *feedforward* utilizaram-

se os algoritmos *backpropagation* com 3 e 4 camadas e para a rede recorrente utilizou-se o algoritmo *backpropagation through time* (BPTT) com 3 camadas.

Para que os objetivos da pesquisa sejam alcançados, na seção 2 faz-se a revisão da literatura, na seção 3 apresentam-se conceitos sobre RNAs, na seção 4 detalha-se a técnica de *k-fold cross validation*. Na seção 5, apresentam-se a execução das etapas da metodologia e os resultados de simulação obtidos e finalmente nas seções 6, 7 e 8 as conclusões, agradecimentos e as referências bibliográficas consultadas.

2 Revisão de literatura

Jia *et al.* (2014) realizam experimentos utilizando uma rede neural de função de base radial (FBR ou RBF). O objetivo do artigo é obter uma RNA RBF otimizada utilizando algoritmo genético sendo conhecido como GA-RBF. Para realizar os testes com o GA-RBF foi utilizado o *dataset wine*. O objetivo do trabalho foi fazer uma comparação entre a acurácia da RNA RBF comum e a RNA GA-RBF as quais obtiveram os resultados de 90,94% e 86,88%, respectivamente.

O trabalho de Rehman *et al.* (2018) descreve experimentos utilizando os classificadores *Naïve Bayesian* (NB) e uma Rede Neural Artificial *Perceptron* Multicamadas (RNAPM), ambas fazendo simulações no dataset *Car Evaluation*. Para a validação foram utilizadas várias técnicas, mas há um destaque para a *k-fold cross validation* na qual foi utilizado o valor de $k=10$. Os resultados obtidos utilizando a validação *k-fold* foram em média de 82,34% para o NB e 94,09% para a RNAPM.

Na pesquisa de Cortez, Cerdeira, Almeida, Matos e Reis (2009) os autores propõem uma abordagem de mineração de dados para prever preferências de gosto de vinho por humanos, baseada em testes analíticos facilmente disponíveis na etapa de certificação. Um grande conjunto de dados (quando comparado com outros estudos neste domínio) é considerado, com amostras de vinho verde branco e vermelho (de Portugal). Três técnicas de regressão foram aplicadas, sob um procedimento computacionalmente eficiente que realiza a seleção simultânea de variáveis e modelos. A máquina de vetores de suporte alcançou resultados promissores, superando os métodos de regressão múltipla e redes neurais. Tal modelo é útil para apoiar as avaliações de degustação de vinhos do enólogo e melhorar a produção de vinho.

O objetivo deste trabalho Rehman *et al.* (2018) de pesquisa é comparar o desempenho de classificação de dois algoritmos conhecidos : Naive Bayes (NB) e RNAs, no processo de

classificação usando o dataset Car Evaluation. O melhor desempenho do algoritmo (NB) foi de 83.68% e da RNA foi de 94.79%.

3 Redes Neurais Artificiais

Redes Neurais Artificiais são inspiradas biologicamente no funcionamento do cérebro, assim uma rede neural típica é constituída por um conjunto de neurônios interligados, influenciando uns aos outros formando um sistema maior capaz de reconhecer padrões por meio de treinamento. O algoritmo de aprendizado generaliza os dados e memoriza o conhecimento dentro dos parâmetros adaptáveis da rede, denominado de pesos.

Uma **Rede Direta**, como ilustrado na Figura 1, consiste em várias camadas compostas por nós (neurônios) onde cada neurônio de uma camada possui ligação com todos os neurônios da camada seguinte. Conforme visto na Figura 1, a rede direta, denominada *perceptron* multicamada, possui uma camada de entrada, que atua como um conjunto de sensores da rede captando os estímulos do ambiente e pode ter uma ou mais camadas intermediárias que é onde a maior parte do processamento é realizada através das conexões e seus pesos respectivos, podem ser considerados como extratoras de características e uma camada de saída onde o resultado final é concluído e apresentado. Ainda na Figura 1 não existem conexões entre a saída de um neurônio e algum outro neurônio localizado em uma camada anterior ao primeiro, ou seja, não possui ciclos, fato que caracteriza uma rede *feedforward*.

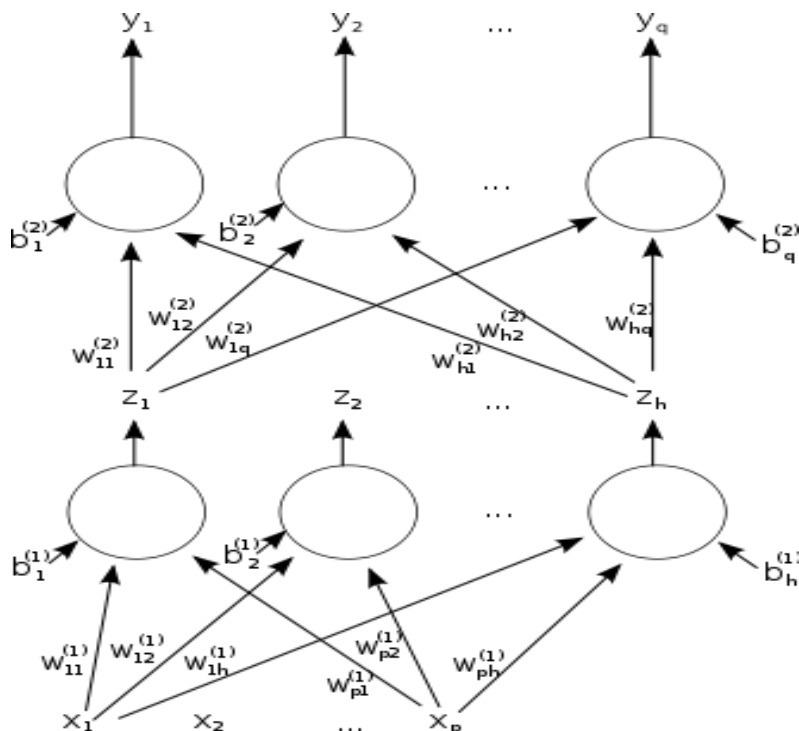


Figura 1. Rede Perceptron Multicamada

Fonte: *Wikipedia enciplopedia livre*. Recuperado em 01 junho, 2018, de https://pt.wikipedia.org/wiki/Rede_neural_artificial

A configuração pela qual os neurônios de uma RNA e estruturada depende do algoritmo de aprendizagem a ser utilizado para o treinamento. No presente trabalho e usado uma rede neural multicamadas compostas de camadas alinhadas e totalmente conectadas diretamente com uma camada de neurônios. Neste tipo de rede, as entradas são apresentadas na primeira camada, que é chamada camada de entrada. Esta distribui as informações de entrada para a camada oculta da rede, que por sua vez pode possuir uma ou mais camadas ocultas e uma camada de saída (Kovács, 2006). O valor de saída é obtido através da sequência de funções de ativação definido na camada oculta.

A aprendizagem é uma das características notáveis das RNAs. No aprendizado supervisionado, o treinamento possui um conhecimento sobre os dados, uma especie de instrutor que confere o quanto a rede está próxima de uma solução aceitável, este conhecimento, está representado sob forma de um conjunto de amostras de *entrada-saída*. O processo de treinamento modifica os pesos das RNAs com a finalidade de melhorar um determinado critério de desempenho, de tal forma que, para o conjunto de entrada informado, a rede seja capaz de calcular uma saída o mais próximo possível da saída desejada.

A aprendizagem de Redes Multilayer Perceptron (MLP) é um processo iterativo, conhecido como aprendizagem por experiência, no qual padrões de treinamento (exemplos) são apresentados à rede e com base nos erros obtidos, são realizados ajustes nos pesos sinápticos, com o intuito de diminuir o erro nas próximas iterações. Sua estrutura é formada pela camada de entrada, que atua como os sensores da rede captando os estímulos do ambiente e pode ter uma ou mais camadas intermediárias que é onde a maior parte do processamento é realizada através das conexões e seus pesos respectivos, podem ser considerados como extratoras de características e uma camada de saída onde o resultado final é concluído e apresentado.

O treinamento de um Perceptron de Múltiplas Camadas (MLP) consiste em ajustar os pesos e os *thresholds* (bias) de suas unidades para que a classificação desejada seja obtida. Quando um padrão é inicialmente apresentado à rede, ela produz uma saída e, após medir a diferença entre a resposta atual e a desejada, são aplicados ajustes apropriados nos pesos de modo a reduzir esta distância. Este procedimento é conhecido como Regra Delta. O algoritmo mais utilizado para o treinamento destas redes MLP é uma generalização da Regra Delta denominada de Backpropagation.

O **Backpropagation** é baseado na regra de aprendizagem por correção de erro. Durante o treinamento com o algoritmo *Backpropagation*, a rede opera em uma sequência de dois passos. No primeiro, um padrão é apresentado à camada de entrada da rede. O sinal resultante se propaga através da rede, camada por camada, até que a resposta seja produzida pela camada de saída. No segundo passo, a saída obtida é comparada à saída desejada para esse padrão particular. Se esta não estiver correta, o erro é calculado. Este erro é propagado a partir da camada de saída até a camada de entrada, e os pesos das conexões das unidades das camadas internas vão sendo modificados à medida que o erro é retropropagado. Ao utilizar a retro propagação o algoritmo aprende de forma supervisionada, em tempo discreto e utiliza um método de gradiente descendente para correção de erro, assim a codificação executa um mapeamento entrada-saída, através da minimização de uma função custo. Está por sua vez, realiza iterativamente ajustes nos pesos sinápticos de acordo com o erro quadrático acumulado para o conjunto de treinamento (Fabro, 2001). O processo de evolução da redução gradativa de erro pode levar a convergência. A medida que a rede aprende, o valor do erro converge para um valor estável, assim o processo prossegue, até o critério de um valor mínimo de erro global seja atingido ou o número de épocas (interações) seja alcançado. Os passos do algoritmo de retro propagação são descritos abaixo.

1) Inicializar os pesos da rede. Cada peso deve ser ajustado aleatoriamente para um número entre -0.1 e 0.1 .

$w_{1ij} = \text{aleatório}(-0.1;0.1)$, $w_{2ij} = \text{aleatório}(-0.1;0.1)$ para todo $i=0, \dots, A$; $j=1, \dots, B$; para todo $i=0, \dots, B$; $j=0, \dots, C$.

2) Inicializar as ativações das unidades limite. Seus valores nunca mudam.

$x_o = 1.0$ e $h_o = 1.0$

3) Escolher um par de padrão de entrada-saída. Supondo que o vetor de entrada seja x_i e o vetor de saída seja y_j . Atribuem-se níveis de ativação às unidades de entrada.

4) Propagar a ativação das unidades da camada de entrada para as unidades da camada oculta, usando a função de ativação.

$$h_j = \frac{1}{1 + e^{-\sum_{i=0}^A w_{1ij} \cdot x_i}}, \text{ para todo } j=1, \dots, B \quad (1)$$

5) Propagam-se as ativações das unidades da camada oculta para as unidades da camada de saída, usando a função de ativação.

$$o_j = \frac{1}{1 + e^{-\sum_{i=0}^B w_{2ij} \cdot h_i}}, \text{ para todo } j=1, \dots, C \quad (2)$$

6) Computar os erros das unidades da camada de saída, denotados por $\delta 2_j$. Os erros baseiam-se na saída real da rede (o_j) e na saída (y_j).

$$\delta 2_j = o_j(1 - o_j)(y_j - o_j) \text{ para todo } j=1, \dots, C \quad (3)$$

7) Computar os erros das unidades da camada oculta, denotados por $\delta 1_j$.

$$\delta 1_j = h_j(1 - h_j) \sum_{i=0}^C \delta 2_i \cdot w_{2ij} \text{ para todo } j=1, \dots, B \quad (4)$$

- 8) Ajuste dos pesos entre a camada oculta e a camada de saída. O coeficiente de aprendizagem é denotado por η , sua função é a mesma de na aprendizagem por perceptrons.

$$\Delta w_{2_{ij}}(t+1) = \Delta w_{2_{ij}}(t) + \eta \delta_{2_j} \cdot h_i \quad \text{para todo } i=0, \dots, B; j=1, \dots, C \quad (5)$$

Ajuste os pesos entre a camada de entrada e a camada oculta

$$\Delta w_{1_{ij}}(t+1) = \Delta w_{1_{ij}}(t) + \eta \delta_{1_j} \cdot x_i \quad \text{para todo } i=0, \dots, A; j=1, \dots, B \quad (6)$$

Vá para a etapa 4 e repita. Quando todos os pares entrada-saída tiverem sido apresentados à rede, uma época terá sido completada. Repita as etapas de 4 a 10 para tantas épocas quantas forem desejadas.

As equações (7) e (8) mostram as fórmulas para atualização dos pesos, da camada de saída e da camada intermediária, respectivamente.

$$\Delta w_{2_{km}}(t+1) = \Delta w_{2_{km}}(t) + \eta [(y_{nm} - o_m)(1 - o_m) \cdot o_m] \cdot h_k \quad (7)$$

$$\Delta w_{1_{jk}}(t+1) = \Delta w_{1_{jk}}(t) + \eta [(1 - h_k) h_k \cdot \sum_{m=0}^C (y_{nm} - o_m)(1 - o_m) \cdot o_m \cdot w_{2_{km}}] x_{nj} \quad (8)$$

O interesse em redes de *feedforward* profundas (*deep feedforward networks*) foi revivido por volta de 2006 (Bengio, Lamblin, Popovici, & Larochelle, 2006) por um grupo de pesquisadores reunidos pelo Instituto Canadense de Pesquisa Avançada (CIFAR).

O algoritmo de retropropagação, descrito anteriormente, não é capaz de implementar mapeamentos dinâmicos. A questão, é como estender a estrutura das redes MLP – *Multi Layer Perceptron* – para que as mesmas possam mapear um comportamento variante no tempo, sendo assim capaz de tratar sinais temporais. O que significa oferecer características dinâmicas ao mapeamento realizado pela rede, tornando-a sensível a sinais que variem com o tempo. Para uma RNA ser considerada dinâmica, é preciso que possua memória. A forma abordada nessa pesquisa, para incorporar memória a RNA, foi a utilização de redes neurais recorrentes, tais como a BPTT. A Figura 2 mostra uma arquitetura de RNA recorrente.

4 Técnica de validação K-Fold Cross Validation

Uma das etapas mais importantes em um projeto de classificadores é a etapa de validação dos resultados. Muitas das técnicas mais poderosas de aprendizado apresentam uma grande quantidade de parâmetros e quanto menos restrições colocarmos no nosso modelo maior a probabilidade de encontrarmos um super ajustamento, ou como é mais conhecido, o *overfitting*.

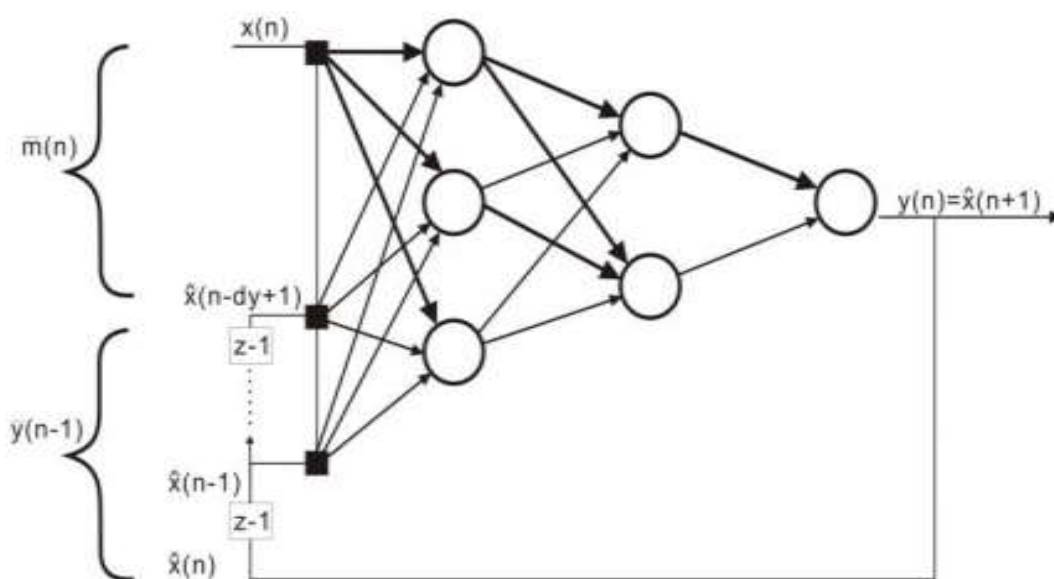


Figura 2. Rede Recorrente

Fonte: elaborado pelos autores (2019).

Overfitting ocorre quando o método que estimamos consegue bons resultados apenas nos dados que eles foram treinados. Já na presença de novas observações, percebemos uma grande piora na qualidade da predição. Ou seja, o *overfitting* ocorre quando o método de aprendizado não consegue generalizar os resultados para dados que não foram utilizados no processo de treino. A técnica *k-fold Cross Validation* estima o erro do método de aprendizado em observações não utilizadas no treino, ou seja, estima como o modelo construído irá se comportar em novos dados, claro que isto é válido apenas se mantivermos a mesma probabilidade conjunta das variáveis explicativas e da variável resposta utilizada durante o treino. O *k-fold Cross Validation* consiste em dividir a base em *k* pedaços. Para cada pedaço, estimamos o método sem a presença desta parte e verificamos o erro médio no pedaço não utilizado durante o treino. Abaixo, na Figura 3, apresenta-se a descrição desse algoritmo.

1. Arranjar os exemplos de treinamento em ordem randômica
2. Dividir os exemplos de treinamento em “k” conjuntos não sobrepostos D_1, D_2, \dots, D_k . (K pedaços de aproximadamente D/k exemplos cada).
3. Para $i=1 \dots k$;
Treine o classificador usando todos os exemplos que não pertencem ao conjunto i ($D \setminus D_i$).
Teste classificador em todos os exemplos no conjunto i (D_i).
Computar n_i , o número de exemplos no conjunto i que foram classificados erradamente.
4. Retorne a seguinte estimativa de erro para o classificador:
$$E = \frac{\sum_{i=1}^k n_i}{D} * 100\%$$

OBS: Para t execuções do algoritmo: $E = \frac{\sum_{i=1}^t E_i}{t}$

Figura 3. Algoritmo *k-fold cross-validation*

Fonte: elaborado pelos autores (2019).

5 Experimentos Realizados

Nessa seção apresentam-se a descrição dos experimentos e os resultados de simulação para os dois problemas estudados “Wine” e “Car Evaluation”.

5.1 Wine

O primeiro “dataset” escolhido para os experimentos foi o *Wine* obtido no seguinte repositório: <https://archive.ics.uci.edu/ml/datasets/Wine>. A arquitetura de RNA utilizada foi a rede *feedforward* que implementa os algoritmos *backpropagation* com 3 e 4 camadas.

O primeiro problema consistiu em fazer a análise química de vinhos com o objetivo de se descobrir sua origem. Os dados foram coletados de 3 cultivares diferentes de uma mesma região na Itália, totalizando 178 registros, 13 atributos e 3 classes. Cada atributo representa um elemento químico existente no vinho. O vinho tipo 1 possui 59 atributos, o tipo 2 possui 71 e o tipo 3 possui 48. Para a realização das simulações foram utilizadas uma rede neural *feedforward* com três camadas, uma rede neural profunda com 4 camadas utilizando o algoritmo *backpropagation* e uma rede recorrente utilizando o algoritmo BPTT. Para compreender melhor os atributos que compõem esse dataset a Tabela 1 apresenta os mesmos e mostra seus significados.

Tabela 1

Descrição dos atributos do dataset Wine

Atributos		
1	Álcool	
2	Ácido málico	
3	Substância inorgânica remanescente	
4	Alcalinidade da substância inorgânica remanescente	
5	Magnésio	
6	Fenóis totais	
7	Flavonoides	
8	Fenóis não flavonoides	
9	Proantocianidinas	
10	Intensidade da cor	
11	Matiz	
12	OD280 / OD315 de vinhos diluídos	
13	Prolina	
Classes		
	Tipo 1	Tipo 2
		Tipo 3

Nota: Fonte: elaborado pelos autores (2019).

A matriz de dispersão, mostrada na Figura 4, mostra as relações entre todos os pares de variáveis possíveis das entradas, onde os pontos vermelhos representam o vinho do tipo 1, azuis o tipo 2 e os verdes o tipo 3.

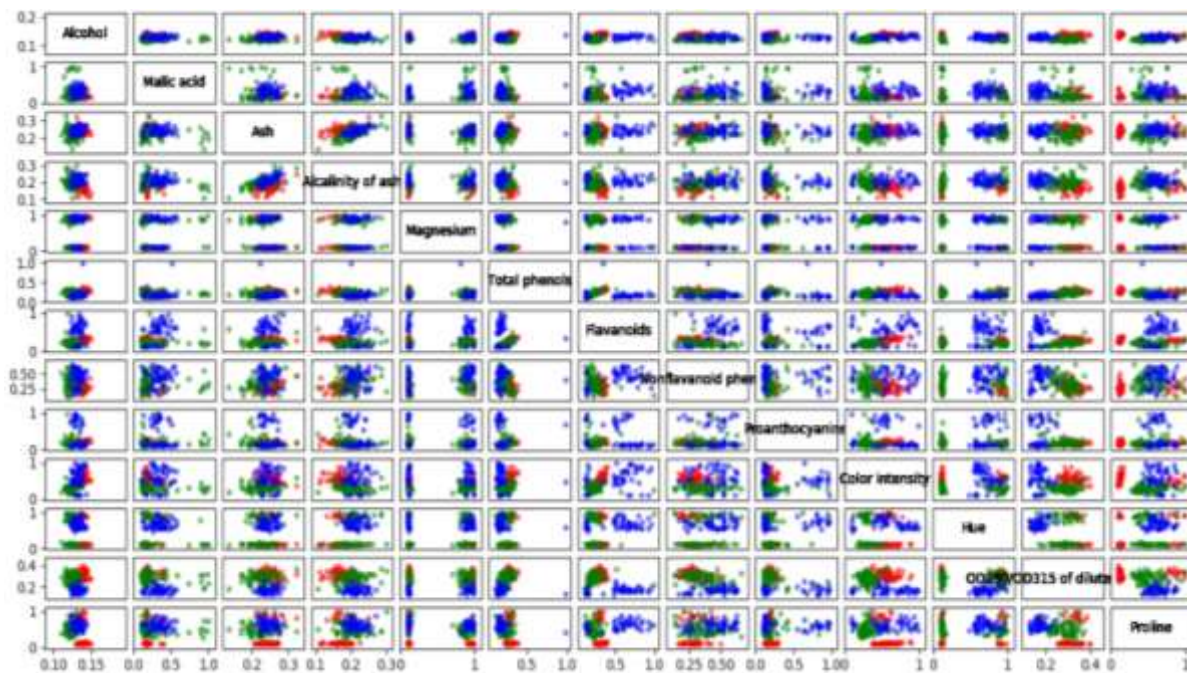


Figura 4. Matriz de dispersão das variáveis de entrada do Dataset Wine

Fonte: elaborado pelos autores (2019).

A Tabela 2 resume os resultados das 36 simulações realizadas para o *dataset Wine*, utilizando a *técnica k-fold cross validation*. O campo (Acertos %), mostra a acurácia do modelo obtida na fase de validação das redes. Os resultados mostrados acima das linhas verdes, foram obtidos utilizando arquitetura de RNA *feedforward* com 3 camadas e os que estão acima das linhas azuis são resultados obtidos utilizando redes profundas com 4 camadas. A melhor acurácia obtida para a rede de 3 camadas foi de 92,13% a mesma utilizou para treino/teste os subconjuntos de dados D2/D1, 9 neurônios na camada intermediária e 200000 épocas para o treino. Utilizando a rede profunda com 4 camadas, com a primeira camada intermediária com 15 neurônios, a segunda com 9 e 450000 em épocas para treino, com os subconjuntos para treino/teste D2/D1, a melhor RNA obteve 89,89% de acertos.

As Tabelas 3 e 4, apresentadas abaixo, mostram a matriz de confusão dos melhores resultados para cada algoritmo, essa matriz serve para mostrar como foram classificados os dados nas simulações. Destacados em verde estão os dados que estão corretos, os demais são os dados que foram classificados de forma errada e mostra também como os mesmos foram classificados.

Tabela 2

Resultados de classificação e parâmetros de simulação obtidos para o dataset Wine para o algoritmo com 3 e utilizando redes profundas com 4 camadas

Treino k-fold	Teste k-fold	Erros	Acertos	Neurônios Da Camada Intermediária 1	Neurônios Da Camada Intermediária 2	Taxa Aprendizado	Taxa Aprendizado 1	Épocas	Acertos %	Media Acertos %
D1	D2	12	77	7	0	0,9	-	150000	86.52	82,025
D2	D1	20	69	7	0	0,9	-	150000	77.53	
D1	D2	11	78	7	0	0,9	-	200000	87.64	85,395
D2	D1	15	74	7	0	0,9	-	200000	83.15	
D1	D2	9	80	7	0	0,9	-	300000	89.89	87,64
D2	D1	13	76	7	0	0,9	-	300000	85.39	
D1	D2	14	75	8	0	0,9	-	150000	84.27	82,025
D2	D1	18	71	8	0	0,9	-	150000	79.78	
D1	D2	14	75	8	0	0,9	-	200000	84.27	85,395
D2	D1	12	77	8	0	0,9	-	200000	86.52	
D1	D2	7	82	8	0	0,9	-	300000	92.13	88,76
D2	D1	13	76	8	0	0,9	-	300000	85.39	
D1	D2	14	75	9	0	0,9	-	150000	84.27	83,145
D2	D1	16	73	9	0	0,9	-	150000	82.02	
D1	D2	15	74	9	0	0,9	-	200000	83.15	87,64
D2	D1	7	82	9	0	0,9	-	200000	92.13	
D1	D2	12	77	9	0	0,9	-	300000	86.52	88,205
D2	D1	9	80	9	0	0,9	-	300000	89.89	
D1	D2	15	74	9	8	0.3	0.5	300000	83.15	85,395
D2	D1	11	78	9	8	0.3	0.7	300000	87.64	
D1	D2	13	76	9	8	0.6	0.4	350000	85.39	87,075

D2	D1	10	79	9	8	0.4	0.7	350000	88.76	85.39
D1	D2	13	76	9	8	0.5	0.3	400000	85.39	
D2	D1	13	76	9	8	0.3	0.5	400000	85.39	

(continua)

Tabela 2

Resultados de classificação e parâmetros de simulação obtidos para o dataset Wine para o algoritmo com 3 e utilizando redes profundas com 4 camadas (conclusão)

Treino k-fold	Teste k-fold	Erros	Acertos	Neurônios Da Camada Intermediária 1	Neurônios Da Camada Intermediária 2	Taxa Aprendizado	Taxa Aprendizado 1	Épocas	Acertos %	Media Acertos %
D1	D2	11	78	12	10	0.3	0.7	400000	87.64	87.64
D2	D1	11	78	12	10	0.3	0.7	400000	87.64	
D1	D2	9	80	12	10	0.3	0.7	450000	89.89	87,08
D2	D1	14	75	12	10	0.3	0.7	450000	84.27	
D1	D2	13	76	12	10	0.3	0.7	500000	85.39	84,27
D2	D1	15	74	12	10	0.3	0.5	500000	83.15	
D1	D2	12	77	15	9	0.3	0.5	350000	86.52	87,08
D2	D1	11	78	15	9	0.4	0.7	350000	87.64	
D1	D2	12	77	15	9	0.3	0.5	400000	86.52	87,08
D2	D1	11	78	15	9	0.4	0.7	400000	87.64	
D1	D2	12	77	15	9	0.3	0.5	450000	86.52	88,205
D2	D1	9	80	15	9	0.4	0.7	450000	89.89	

Nota: Fonte: elaborado pelos autores (2019).

Além das simulações com a rede *feedforward*, também se realizaram outras simulações utilizando a RNA recorrente que implementa o algoritmo BPTT. Como as simulações com esse algoritmo apresentaram melhores resultados que o *backpropagation* direto, então o mesmo foi utilizado para realizar as simulações com o *dataset Car Evaluation*.

Visto que todos os resultados melhoraram utilizando a rede recorrente, foi decidido a utilização da mesma arquitetura de rede para as simulações utilizando o próximo dataset. Em uma breve comparação, com base na Tabela 5, pode-se observar que o melhor pior e melhor resultados utilizando a rede recorrente são, respectivamente, 91,01% e 95,51% de acertos. Diante disso, observa-se que houve uma melhora significativa em relação aos testes realizados com as redes *feedforward* de três camadas, no qual o melhor resultado foi de 92,13%, e no *backpropagation* de rede profunda com 4 camadas, o melhor resultado obtido foi 89,89%.

Tabela 3

Matriz de confusão do melhor resultado para o algoritmo com 3 camadas

Tipo 1	Tipo 2	Tipo 3
27	3	0
2	33	0
0	2	22
Total:		92,13%

Nota: Fonte: elaborado pelos autores (2019).

Tabela 4

Matriz de confusão do melhor resultado para o algoritmo com 4 camadas

Tipo 1	Tipo 2	Tipo 3
24	5	0
0	35	1
0	3	21
Total:		89,89%

Nota: Fonte: elaborado pelos autores (2019).

Tabela 5

Resultados de classificação e parâmetros de simulação obtidos para o dataset Wine utilizando o algoritmo backpropagation through time

Treino k-fold	Teste k-fold	Erros	Acertos	Neurônios Da Camada Intermediária	Taxa Aprendizado	Épocas	Acertos %	Acurácia	Media Acertos %
D1	D2	8	81	9	0,9	200000	91.01%	0,910112	93,26
D2	D1	4	85	9	0,9	200000	95.51%	0,955056	
D1	D2	6	83	9	0,9	300000	93.26%	0,932584	93,8
D2	D1	5	84	9	0,9	300000	94.38%	0,94382	

Nota: Fonte: elaborado pelos autores (2019).

5.2. Car Evaluation

O segundo o dataset escolhido para a realização das simulações foi o *Car Evaluation*. Este dataset foi obtido do repositório: <https://archive.ics.uci.edu/ml/datasets/car+evaluation>. A arquitetura de RNA utilizada nas simulações foi uma rede recorrente que implementa o algoritmo BPTT.

Os dados desse *dataset*, são derivados de um modelo de decisão hierárquico de aprendizado sobre aquisições e decisões com múltiplas variáveis. São avaliados 6 atributos e o resultado é a aceitabilidade de um carro. Cada instância pertence a uma das 4 classes possíveis: inaceitável, aceitável, bom e muito bom.

Existem 1728 atributos no total no *dataset*, sendo que 1210 pertencentes a classe inaceitável, 384 pertencem a classe aceitável, 69 são bons e 65 são muito bons, portanto os atributos não são balanceados, o que dificulta a classificação, por isso optou-se em utilizar a rede recorrente para a classificação, haja vista o seu bom desempenho na simulação do *dataset Wine*. Foi utilizada a técnica *k-fold cross validation* (k=4) para divisão randômica dos dados, ou seja, os 1728 registros do *dataset*, foram divididos 4 subgrupos (D1, D2, D3, D4), cada um com 432 registros. Para melhor compreensão das etapas desenvolvidas é importante descrever

os dados que compõem o *dataset Car Evaluation*. A Tabela 6, apresenta os atributos que compõe os registros.

Tabela 6

Descrição dos atributos do dataset Car Evaluation

Atributos			
1	Preço de compra (<i>buying</i>)		
2	Preço de manutenção (<i>maint</i>)		
3	Conforto (<i>doors</i>)		
4	Número de portas (<i>persons</i>)		
5	Capacidade de pessoas (<i>lug_boot</i>)		
6	Tamanho do assento (<i>safety</i>)		
Classes			
Inaceitável (<i>Unacc</i>)	Aceitável (<i>Acc</i>)	Bom (<i>Good</i>)	Muito bom (<i>V-good</i>)

Nota: Fonte: elaborado pelos autores (2019).

A matriz de dispersão abaixo apresenta as relações entre todos os pares de variáveis possíveis das entradas, onde os pontos vermelhos representam “Inaceitável”, amarelos representam “Aceitável”, verdes representam “Bom” e azuis representam “Muito bom”.

Para que os dados ficassem adequados para o processamento foi realizada a normalização dos dados. Diferente do dataset anterior, esse possuía valores do tipo texto, porém o algoritmo de treinamento aceita apenas dados que estão entre o intervalo [0,1] e para que isso ocorresse houve uma adaptação desses valores. Os valores foram adaptados de acordo com a Tabela 7.

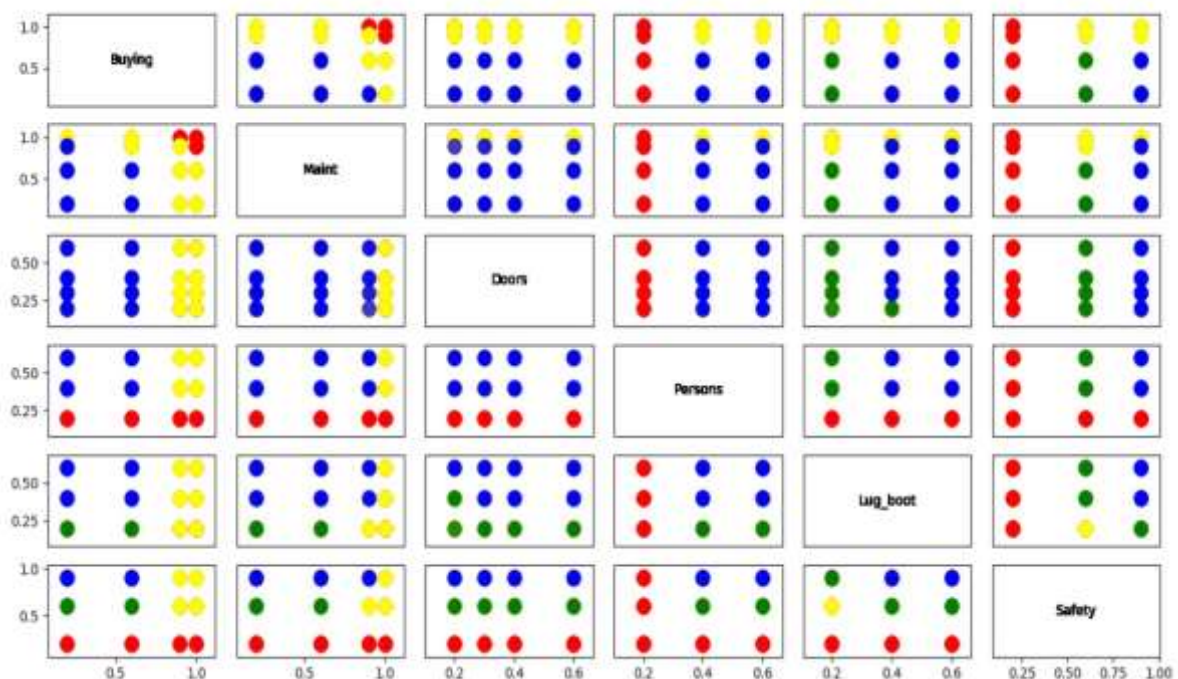


Figura 5. Matriz de dispersão das variáveis de entrada do Dataset Car Evaluation

Fonte: elaborado pelos autores (2019).

Tabela 7

Descrição da adaptação dos dados

• Atributos	• Valores
• <i>Buying</i>	• <i>v-high</i> (1.0), <i>high</i> (0.9), <i>med</i> (0.6), <i>low</i> (0.2)
• <i>Maint</i>	• <i>v-high</i> (1.0), <i>high</i> (0.9), <i>med</i> (0.6), <i>low</i> (0.2)
• <i>Doors</i>	• <i>v-high</i> (1.0), <i>high</i> (0.9), <i>med</i> (0.6), <i>low</i> (0.2)
• <i>Persons</i>	• 2 (0.2), 4 (0.4), more (0.6)
• <i>Lug_boot</i>	• <i>small</i> (0.2), <i>med</i> (0.4), <i>big</i> (0.6)
• <i>Safety</i>	• <i>low</i> (0.2), <i>med</i> (0.6), <i>high</i> (0.9)
• Classes	
• <i>Unacc</i> (0.0)	• <i>Acc</i> (0.3)
	• <i>Good</i> (0.6)
	• <i>V-good</i> (0.9)

Nota: Fonte: elaborado pelos autores (2019).

Em cada iteração do método de validação cruzada, ilustrada em cada linha da Tabela 8, utilizaram-se para treino de cada rede neural os parâmetros que estão destacados na coluna 5 (número de neurônios na camada intermediária), na coluna 6 (taxa de aprendizado), na coluna 7 (número de épocas de treinamento). Nas colunas 3, 4 e 8 apresentam-se os resultados dos testes das redes neurais recorrentes.

Os testes foram feitos estimando parâmetros de forma aleatória, à medida que a rede respondia a cada alteração dos parâmetros, eram avaliados os resultados até ser obtida a melhor combinação de parâmetros para os quais a rede neural apresentava a maior taxa de acertos, sendo esses registrados na Tabela 8. Foram 2 simulações que tiveram os melhores resultados com a rede BPTT, em que ambas obtiveram 95,83% de acertos. As redes utilizam para treino/teste os subconjuntos de dados D2/D3 e D2/D4, 10 neurônios na camada intermediária e 400000 épocas para treino. A Tabela 9, apresentada abaixo, mostra a matriz de confusão da melhor simulação com o algoritmo BPTT.

Tabela 8

Resultados de classificação e parâmetros de simulação obtidos para o dataset Car Evaluation

Treino k-fold	Teste k-fold	Erros	Acertos	Neurônios Da Camada Intermediária	Taxa Aprendizado	Épocas	Acertos %	Media Acertos %
D1	D2	28	404	10	0,9	400000	93.52	93,3666
	D3	35	397	10	0,9	400000	91.90	
	D4	23	409	10	0,9	400000	94.68	
D2	D1	27	405	10	0,9	400000	93.75	95,1366
	D3	18	414	10	0,9	400000	95.83	
	D4	18	414	10	0,9	400000	95.83	
D3	D1	30	402	10	0,9	400000	93.06	93,5966
	D2	28	404	10	0,9	400000	93.52	
	D4	25	407	10	0,9	400000	94.21	
D4	D1	28	404	10	0,9	400000	93.52	92,7466
	D2	35	397	10	0,9	400000	91.90	
	D3	31	401	10	0,9	400000	92.82	

Nota: Fonte: elaborado pelos autores (2019).

Tabela 9

Matriz de confusão da melhor simulação do dataset car evaluation

Unacc (0.0)	Acc (0.3)	Good (0.6)	V-Good (0.9)
303	0	0	0
1	93	2	0
0	0	15	2
0	0	13	3
Total:			95,83%

Nota: Fonte: elaborado pelos autores (2019).

6 Conclusões

Para os testes com o *dataset Wine*, a rede neural com arquitetura *feedforward* ou direta com aprendizado *backpropagation* não foi uma ferramenta tão eficiente no reconhecimento de padrões de comportamento. Ao utilizar o algoritmo *backpropagation* implementado com 3 e 4 camadas podemos observar que a média geral são, respectivamente 85,58% e 86,58% de acertos. Utilizando a rede recorrente com a implementação do algoritmo BPTT observa-se uma melhora significativa nos acertos da rede obtendo-se uma média de 93,53%.

Para o dataset *Car Evaluation*, no qual foi utilizada a arquitetura recorrente, também com o algoritmo BPTT, a média dos resultados foi de 93,71%, ou seja, houveram menos de 7% de erros durante a fase de validação. Observa-se que o pior e melhor resultado são,

respectivamente, 91,90% e 95,1366%. O melhor resultado utilizou a base D2 para treino e as restantes (D1, D3, D4) para testes.

Agradecimentos

Agradecemos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela concessão da bolsa de iniciação científica.

Referências

- Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2006). Greedy layer-wise training of deep networks. In B. Schölkopf, J. C. Platt & T. Hoffman (Eds.). *Advances in Neural Information Processing Systems 19* (pp. 153-160). San Diego: Neural Information Processing Systems Foundations.
- Chatziagorakis, P., Georgoulas, N., Papadopoulou, S., Elmasides, C., Giaouris, D., Seferlis, P., Sirakoulis, Ch. G., Karafyllidis, I., Papadopoulos, A. I., Stergiopoulos, F., Ziogou, C., Ipsakis, D., Voutetakis, S., & Andreadis, I. (2014). Application of neural networks solar radiation prediction for hybrid renewable energy systems. *Engineering Applications of Neural Networks*, 459, 133-144.
- Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4).
- Dasgaonkar, K., & Chopade, S. (2018). Analysis of multi-layered perceptron, radial basis function and convolutional neural networks in recognizing handwritten digits. *International Journal of Advance Research, Ideas and Innovations in Technology*, 4(3).
- Fabro, J. A. (2001, agosto). *Redes neurais artificiais*. [Curso de Especialização em Inteligência Computacional]. Recuperado em 01 junho, 2018, de <http://www.dainf.ct.utfpr.edu.br/~fabro/IF67D/redesneuraisartificiais.pdf>
- Gambogi, J. A. (2013). *Aplicação de redes neurais na tomada de decisão no mercado de ações*. Dissertação de mestrado, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil.
- Garcia, J. C. P., & Campos, L. M. L. (2019). Redes neurais apoiando a tomada de decisão na análise de crédito bancário e detecção do câncer de mama. *Revista Gestão e Tecnologia*, 19, 90-112.
- Jia, W., Zhao, D., Shen, T., Su, C., Hu, C., & Zhao Y. (2014). A new optimized GA-RBF Neural Network Algorithm. *Computational Intelligence and Neuroscience*, 2014.
- Kaushik, A., Gupta, S., & Bhatia, M. (2018). A movie recommendation. System using Neural Networks. *International Journal of Advance Research, Ideas and Innovations in Technology*, 4(2).
- Kovács, Z. L. (2006). *Redes neurais artificiais: fundamentos e aplicações*. (4a ed.). São Paulo: Livraria da Física.
- Poonia, V., Tiwari, H. L., & Mishra, S. (2018). Hydrological analysis by Artificial Neural Network: a review. *International Journal of Advance Research, Ideas and Innovations in Technology*, 4(3).
- Rede Neural Artificial, (n.d.). In *Wikipedia enciplopedia livre*. Recuperado em 01 junho, 2018, de https://pt.wikipedia.org/wiki/Rede_neural_artificial

- Rehman, Z. U., Fayyaz, H., Shah, A. A., Aslam, N., Hanif, M., & Abbas, S. (2018). Performance evaluation of MLPNN and NB: a comparative study on Car Evaluation Dataset. *International Journal of Computer Science and Network Security*, 18(9), 144-147.
- Sak, H., Senior, A., & Beaufays, F. (2014). *Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition*. Retrieved June 01, 2018, from arxiv.org/abs/1402.1128
- Sánchez, D., & Melin, P. (2014). Optimization of modular granular neural networks using hierarchical genetic algorithms for human recognition using the ear biometric measure. *Engineering Applications of Artificial Intelligence*, 27, 41-56.
- Sánchez, D., Melin, P., & Castillo, O. (2015). Optimization of modular granular neural networks using a hierarchical genetic algorithm based on the database complexity applied to human recognition. *Information Sciences*, 309, 73-101.
- Sharma, N., Agarwal, P., & Pandey, U. (2018). Offline handwriting recognition using neural networks. *International Journal of Advance Research, Ideas and Innovations in Technology*, 4(2).
- Skrizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25* (pp. 1097-1105). San Diego: Neural Information Processing Systems Foundations.
- Steiner, M. T. A., Soma, N. Y., Shimizu, T., Nievola, J. C., Lopes, F. M., & Smiderle, A. (2004, novembro). *Redes neurais e arvores de decisão na análise do crédito bancário*. Trabalho apresentado em Simpósio Brasileiro de Pesquisa Operacional, São João del-Rei, MG, Brasil, 36. Recuperado em 01 junho, 2018, de <http://www.din.uem.br/sbpo/sbpo2004/pdf/arq0035.pdf>
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to sequence learning with neural networks*. Retrieved June 01, 2018, from <http://arxiv.org/abs/1409.3215>
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). *Intriguing properties of neural networks*. Retrieved June 01, 2018, from arxiv.org/abs/1312.6199
- Wang, Y., & Kosinski, M. (2018, February). Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. *Journal of Personality and Social Psychology*, 114(2), 246-257.