



## TRABALHO

### RECONHECIMENTO DE PADRÕES VIA RNA

Realizar o treinamento de uma rede neural tipo MLP para aprender a classificar o valor de um automóvel. Para isso, deve ser usada uma base pública, que se encontra disponível no endereço <https://archive.ics.uci.edu/ml/datasets/Car+Evaluation>. Essa base contém 1.728 instâncias, cada uma com seis atributos mais a classe do valor do carro. Os valores desses atributos e a classe estão em texto, assim é necessário passar os valores textuais para valores numéricos, a fim de serem usados pela RNA.

Os dados de entrada podem ser pré-processados, assim como os dados de saída podem ser pós-processados. Uma explicação da base de dados está no mesmo repositório, copiada integralmente como anexo a este texto.

Assim, este trabalho consiste em:

- projetar os dados de entrada para a rede, realizando um pré-processamento ou não, além de dividir a base em dados de treino, validação e teste;
- definir como será a saída da rede, de modo a apresentar corretamente a classe;
- construir a RNA, definindo sua estrutura e pesos iniciais;
- por a rede em treinamento, ajustando a configuração da rede, conforme explanação posterior;
- treinar a rede sempre em uma quantidade fixa de épocas com um programa de treino de confecção própria;
- fazer a validação do treinamento simultaneamente;
- testar a generalização do aprendizado, com dados apropriados, no melhor treinamento obtido, salvando os valores dos pesos desse treinamento.

O trabalho deverá ser realizado confeccionando um programa em linguagem MatLab para o treinamento da rede. Esse programa recebe os dados de entrada e promove o ajuste dos pesos via algoritmo BP.

Para efeito de comparação, todo treino deve ter um número fixo de épocas, valor este a ser definido pelo projetista. A cada 10 treinamentos, os valores do erro quadrático médio de saída da RNA em cada época devem ser usados para calcular a média desses valores em cada época, ou seja, se cada treinamento consiste de 1.000 épocas, serão calculadas 1.000 médias dos 10 valores desse erro para cada época.

Desta forma, serão realizados treinamentos para cada caso a seguir:

1. 10 treinamentos reiniciando os pesos aleatoriamente a cada treinamento, sem alterar nada mais na configuração da rede, neste caso não será necessário tirar a média dos erros em cada época;
2. 10 treinamentos reiniciando os pesos aleatoriamente a cada treinamento e variando a função de ativação (sigmoide e tanh) dos neurônios escondidos a cada 10 treinamentos,



**UNIVERSIDADE FEDERAL DO MARANHÃO**  
**Centro de Ciências Exatas e Tecnologia**  
**Programa de Pós-Graduação em Ciência da Computação**  
**Núcleo de Estudos Tecnológicos em Mecatrônica - MECANET**



- perfazendo um total de 20 treinamentos, sem alterar nada mais na configuração da rede;
3. 10 treinamentos reinicializando os pesos aleatoriamente a cada treinamento e variando a quantidade de neurônios escondidos a cada 10 treinamentos, em um total de 10 variações nessa quantidade, perfazendo um total de 100 treinamentos, sem alterar nada mais na configuração da rede;
  4. 10 treinamentos reinicializando os pesos aleatoriamente a cada treinamento e variando a taxa de aprendizado a cada 10 treinamentos, em um total de cinco variações nessa taxa perfazendo um total de 50 treinamentos, sem alterar nada mais na configuração da rede.

Os programas devem ser todos plenamente comentados.

Por fim, um texto explicativo deve ser confeccionado e salvo em formato pdf. Esse texto deve informar como os treinamentos foram realizados, explicando por texto e gráficos: os dados usados, se houve pré-processamento e como foi este; a estrutura da rede; o treinamento em si; a saída do resultado da rede e qualquer outra informação importante a respeito do treinamento, além de como os programas funcionam, de modo a ser executado por outra pessoa facilmente.

Para entrega, os arquivos com os códigos fonte, o texto explicativo e os pesos do melhor treinamento (arquivo com extensão mat) devem ser compactados no formato zip e enviados pelo SIGAA.

Prof. Areolino de Almeida Neto



**UNIVERSIDADE FEDERAL DO MARANHÃO**  
**Centro de Ciências Exatas e Tecnologia**  
**Programa de Pós-Graduação em Ciência da Computação**  
**Núcleo de Estudos Tecnológicos em Mecatrônica - MECANET**



1. Title: Car Evaluation Database

2. Sources:

- (a) Creator: Marko Bohanec
- (b) Donors: Marko Bohanec (marko.bohanec@ijs.si)  
Blaz Zupan (blaz.zupan@ijs.si)
- (c) Date: June, 1997

3. Past Usage:

The hierarchical decision model, from which this dataset is derived, was first presented in

M. Bohanec and V. Rajkovic: Knowledge acquisition and explanation for multi-attribute decision making. In 8th Intl Workshop on Expert Systems and their Applications, Avignon, France. pages 59-78, 1988.

Within machine-learning, this dataset was used for the evaluation of HINT (Hierarchy INduction Tool), which was proved to be able to completely reconstruct the original hierarchical model. This, together with a comparison with C4.5, is presented in

B. Zupan, M. Bohanec, I. Bratko, J. Demsar: Machine learning by function decomposition. ICML-97, Nashville, TN. 1997 (to appear)

4. Relevant Information Paragraph:

Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX (M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.). The model evaluates cars according to the following concept structure:

CAR	car acceptability
. PRICE	overall price
. . buying	buying price
. . maint	price of the maintenance
. TECH	technical characteristics
. . COMFORT	comfort
. . . doors	number of doors
. . . persons	capacity in terms of persons to carry
. . . lug_boot	the size of luggage boot
. . safety	estimated safety of the car

Input attributes are printed in lowercase. Besides the target concept (CAR), the model includes three intermediate concepts: PRICE, TECH, COMFORT. Every concept is in the original model related to its lower level descendants by a set of examples (for these examples sets see <http://www-ai.ijs.si/BlazZupan/car.html>).

The Car Evaluation Database contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, maint, doors, persons, lug\_boot, safety.

Because of known underlying concept structure, this database may be



**UNIVERSIDADE FEDERAL DO MARANHÃO**  
**Centro de Ciências Exatas e Tecnologia**  
**Programa de Pós-Graduação em Ciência da Computação**  
**Núcleo de Estudos Tecnológicos em Mecatrônica - MECANET**



particularly useful for testing constructive induction and structure discovery methods.

5. Number of Instances: 1728  
(instances completely cover the attribute space)

6. Number of Attributes: 6

7. Attribute Values:

buying	v-high, high, med, low
maint	v-high, high, med, low
doors	2, 3, 4, 5-more
persons	2, 4, more
lug_boot	small, med, big
safety	low, med, high

8. Missing Attribute Values: none

9. Class Distribution (number of instances per class)

class	N	N[%]
unacc	1210	(70.023 %)
acc	384	(22.222 %)
good	69	( 3.993 %)
v-good	65	( 3.762 %)