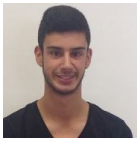
Exercício Prático Nº3 - Grupo 24 - Sistemas de Representação de Conhecimento e Raciocínio

Ângelo Dias Teixeira
a73312

Bruno Manuel Arieira
a70565

João Miguel Palmeira
a73864

Pedro Manuel Almeida
a74301



Maio 2018

Resumo

Neste último exercício prático, abandonamos a linguagem de programação em lógica PROLOG e o conceito de *Hard Computing* para passar a trabalhar com a linguagem R, linguagem esta que permite analisar dados e com eles treinar uma máquina, através da inteligência artificial, englobada já na segunda parte desta unidade curricular, que é referente ao *Soft Computing*. Ao longo deste relatório são abordados os vários aspetos a ter em conta para perceber o exercício, bem como para a sua resolução.

Conteúdo

1 Introdução

Este terceiro e último trabalho prático foi realizado no âmbito da unidade curricular Sistemas de Representação de Conhecimento e Raciocínio, do 2ºSemestre do 3ºAno do Mestrado Integrado em Engenharia Informática, sendo que o principal objetivo é consolidar os conteúdos aprendidos nas aulas destas últimas semanas.

De acordo com o enunciado, foi-nos proposto o desenvolvimento de um sistema de representação de conhecimento não simbólico, o que acarreta o desenvolvimento de mecanismos de raciocínio, utilizando **Redes Neuronais Artificiais (RNAs)** , usando a linguagem **R**. Para tal, é necessário a importação dos dados através de ficheiros CSV, para que de seguida seja possível treinar a RNA de acordo com as topologias que melhor possibilitem a redução do erro.

O primeiro passo foi importar os dados do ficheiro CSV, sendo posteriormente necessário proceder à sua normalização, de maneira a possibilitar a melhor aprendizagem possível à RNA.

2 Contextualização

Este trabalho prático tem como tema a análise de dados físico-químicos e de dados sensoriais (como a qualidade), implícitos nas variantes vermelhas e brancas do *"Vinho Verde"*, sendo que nós optámos por utilizar o conjunto de dados relativos a **variantes brancas**. Estes conjuntos de dados tem como objetivo a classificação da qualidade do vinho, baseando-se em variáveis caraterísticas deste. Neste trabalho focaremos as variáveis de saída (física e químicas) de forma a analisar a qualidade, através do conjunto de dados fornecidos.

3 Preliminares

Quebrando a dependência da representação de conhecimento através do uso de símbolos, foi criada a representação de conhecimento não-simbólico. Neste trabalho apenas se abordará um "ramo" deste amplo tema: Redes neuronais artificiais (**RNAs**). Para permitir uma melhor compreensão do tema de seguida explicar-se-ão conceitos essenciais à compreensão do abordado ao longo deste relatório. Redes neuronais artificiais são estruturas de resolução de problemas que quebram a dependência da utilização de símbolos. Baseia-se na conexão entre unidades de processamento e a sua nomenclatura é herdada da biologia. Desta forma uma rede neuronal é constituída por:

1. **Neurónio**: unidades de processamento;
2. **Dentrite**: associadas aos neurónios, recebem a informação que depois é processada;
3. **Axónio**: também associados aos neurónios, são responsáveis pela passagem de informação.

Um neurónio pode possuir várias dentrites, mas apenas um axónio. À passagem de informação dá-se o nome de transferência/sinápse, sendo que esta apenas ocorre caso o estado de excitação dos neurónios seja suficiente. Este estado é regulado pela informação que chega ao neurónio. A rede neuronal recebe então n parâmetros de um caso como input e, faz esta informação percorrer a sua rede até que é retornado um ou mais valor/valores de output. A aprendizagem da rede é definida pela regra de transferência que a rede neuronal implementa, e, consequentemente, será outro parâmetro que decidirá o funcionamento da mesma. O cálculo do valor de ativação dos neurónios é influenciado pela informação que chega aos mesmos, tanto pelos dados de input como pelo valor de ativação anterior (armazenado em memória). Apesar do uso das redes neuronais ser bastante vantajoso é de notar que todos os valores obtidos são apenas aproximações e que existe uma dependência na existência de "pré-conhecimento", ou seja, são necessários casos de treino com informação já real.

4 Normalização/Análise de Dados

Relativamente ao ficheiro que nos foi fornecido, são apresentados valores relativos às características do vinho utilizadas para classificação da qualidade, que são:

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates
- alcohol
- quality

Os dozes parâmetros referidos acima tem todos uma gama de valores disforme. Para tal decidimos por unanimidade normalizar todos os valores para uma gama de -1 a 1, uniformemente, de forma a não alterar os valores fornecidos, apenas mantendo todos dentro do mesmo intervalo. Surgiu tal necessidade porque dentro desta gama a rede tem melhor aprendizagem. Em baixo encontra-se um excerto dos dados normalizados:

	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
1	-0.09615384615	-0.156862745098	-0.14156626506	-0.09585889571	-0.196587537092	-0.1750871080	-0.063225058005	-0.1161075766	-0.122727272727	-0.116279069767	-0.185483870968	0.00000000000
2	-0.12980769231	-0.142156862745	-0.14759036145	-0.24233128834	-0.190652818991	-0.2290940767	-0.107308584687	-0.1835839599	0.013636363636	-0.093023255814	-0.129032258065	0.00000000000
3	-0.04326923077	-0.151960784314	-0.12951807229	-0.20168711656	-0.189169139466	-0.2012195122	-0.147911832947	-0.1729805282	-0.004545454545	-0.122093023256	-0.080645161290	0.00000000000
4	-0.08653846154	-0.176470588235	-0.15361445783	-0.18941717791	-0.177299703264	-0.1716027875	-0.044663573086	-0.1681607866	-0.036363636364	-0.145348837209	-0.096774193548	0.00000000000
5	-0.08653846154	-0.176470588235	-0.15361445783	-0.18941717791	-0.177299703264	-0.1716027875	-0.044663573086	-0.1681607866	-0.036363636364	-0.145348837209	-0.096774193548	0.00000000000
6	-0.04326923077	-0.151960784314	-0.12951807229	-0.20168711656	-0.189169139466	-0.2012195122	-0.147911832947	-0.1729805282	-0.004545454545	-0.122093023256	-0.080645161290	0.00000000000
7	-0.13461538462	-0.132352941176	-0.20180722892	-0.20092024540	-0.196587537092	-0.2012195122	-0.102668213457	-0.1749084249	-0.040909090909	-0.104651162791	-0.120967741935	0.00000000000
8	-0.09615384615	-0.156862745098	-0.14156626506	-0.09585889571	-0.196587537092	-0.1750871080	-0.063225058005	-0.1161075766	-0.122727272727	-0.116279069767	-0.185483870968	0.00000000000
9	-0.12980769231	-0.142156862745	-0.14759036145	-0.24233128834	-0.190652818991	-0.2290940767	-0.107308584687	-0.1835839599	0.013636363636	-0.093023255814	-0.129032258065	0.00000000000
10	-0.04326923077	-0.181372549020	-0.12048192771	-0.24309815951	-0.198071216617	-0.2047038328	-0.110788863109	-0.1855118566	-0.022727272727	-0.116279069767	-0.008064516129	0.00000000000
11	-0.04326923077	-0.156862745098	-0.12650602410	-0.24348159509	-0.214391691395	-0.2343205575	-0.187354988399	-0.2144303065	-0.127272727273	-0.052325581395	0.072580645161	-0.08333333333
12	-0.01923076923	-0.176470588235	-0.12951807229	-0.22239263804	-0.211424332344	-0.2238675958	-0.133990719258	-0.1768363216	-0.059090909091	-0.069767441860	-0.112903225806	-0.08333333333
13	-0.05288461538	-0.200980392157	-0.13855421687	-0.24539877301	-0.204005934718	-0.2256097561	-0.173433874710	-0.2028629265	-0.040909090909	-0.011627906977	-0.024193548387	-0.08333333333
14	-0.11538461538	-0.210784313725	-0.12951807229	-0.24309815951	-0.198071216617	-0.1698606272	-0.094547563805	-0.2105745132	0.122727272727	-0.075581395349	0.104838709677	0.08333333333
15	-0.03365384615	-0.083333333333	-0.06325301205	-0.10697852761	-0.204005934718	-0.1820557491	-0.060904872390	-0.1238191633	-0.131818181818	0.011627906977	-0.112903225806	-0.08333333333
16	-0.11538461538	-0.205882352941	-0.13554216867	-0.24309815951	-0.215875370920	-0.2047038328	-0.130510440835	-0.2086466165	-0.009090909091	-0.058139534884	0.024193548387	0.08333333333
17	-0.12980769231	-0.053921568627	-0.23795180723	-0.24616564417	-0.195103857567	-0.2012195122	-0.145591647332	-0.1951513399	-0.013636363636	-0.168604651163	-0.120967741935	0.00000000000
18	-0.13461538462	0.034313725490	-0.10542168675	-0.24539877301	-0.220326409496	-0.2029616725	-0.173433874710	-0.2298534799	0.027272727273	-0.151162790698	0.137096774194	0.16666666667
19	-0.07692307692	-0.122549019608	-0.12349397590	-0.24616564417	-0.214391691395	-0.2238675958	-0.062064965197	-0.2057547715	-0.068181818182	-0.069767441860	0.016129032258	0.00000000000
20	-0.12019230769	-0.137254901961	-0.20783132530	-0.19708588957	-0.198071216617	-0.1942508711	-0.106148491879	-0.1691247349	-0.022727272727	-0.087209302326	-0.129032258065	-0.08333333333

Figura 1: Dados normalizados

Algoritmo que possibilita a normalização dos dados:

```
normalize <- function(df, cols)
{
        result <- df
            for (j in cols)
            { for (i in 1:nrow(result))
                {
                    result[i,j] <- (((result[i,j]-((max(result[1:4898,j])+
                    min(result[i:4898,j]))/2))) /(max(result[i:4898,j])
                    -min(result[i:4898,j]))/2)
    }
  }
  return(result)
}
```

5 Redes Neuronais

Após uma pesquisa sobre a quantidade de nodos e camadas a utilizar, chegou-se à conclusão que é possível utilizar duas redes. Podemos observar nas figuras seguintes que ambas as redes estão definidas em *Feed Forward* multi camada, mais concretamente em três camadas distintas, uma de entrada, uma de saída e outra intermédia com 4 nodos em cada uma das redes neuronais.

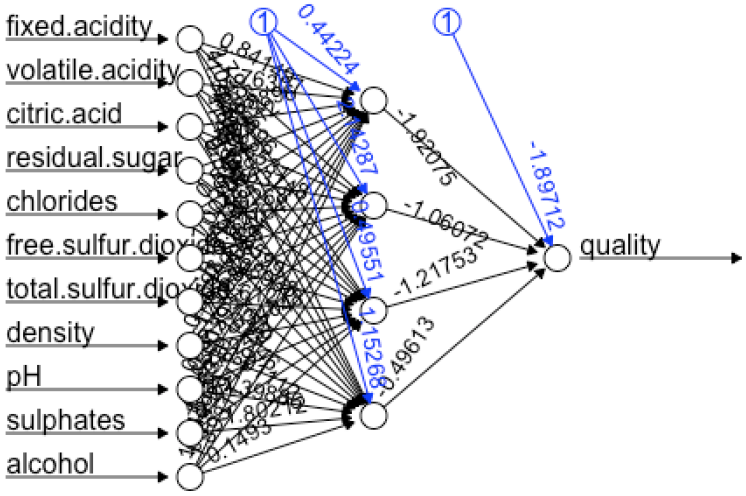


Figura 2: Rede Neuronal (4)

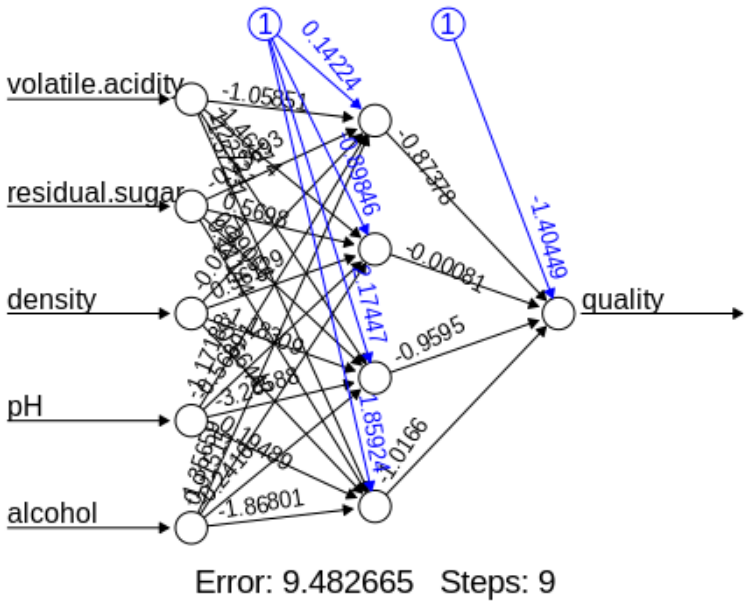
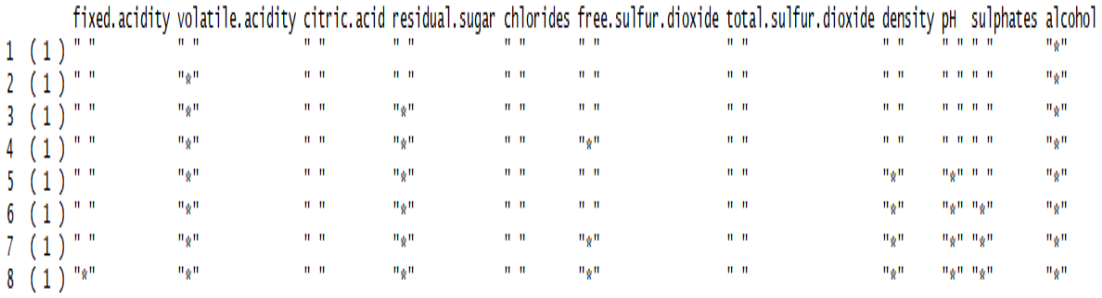


Figura 3: Rede Neuronal (4)

6 Níveis Pré-Definidos

6.1 Importância dos Atributos

Previamente, foi realizado um estudo da relevância de cada atributo para a representação de conhecimento do problema em análise. Para tal foi necessário utilizar todos os atributos disponíveis de forma a analisar-mos os mais relevantes.



	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol
1 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	"*"
2 (1)	" "	"*"	" "	" "	" "	" "	" "	" "	" "	" "	"*"
3 (1)	" "	"*"	" "	"*"	" "	" "	" "	" "	" "	" "	"*"
4 (1)	" "	"*"	" "	"*"	" "	"*"	" "	" "	" "	" "	"*"
5 (1)	" "	"*"	" "	"*"	" "	" "	" "	"*"	"*"	" "	"*"
6 (1)	" "	"*"	" "	"*"	" "	" "	" "	"*"	"*"	"*"	"*"
7 (1)	" "	"*"	" "	"*"	" "	"*"	" "	"*"	"*"	"*"	"*"
8 (1)	"*"	"*"	" "	"*"	" "	"*"	" "	"*"	"*"	"*"	"*"

Figura 4: Tabela gerada relativamente à importância dos atributos

6.2 Fórmula

Após o reconhecimento da importância dos atributos passou-se à definição das fórmulas que serão passadas como argumento no treino da rede neuronal. Para tal definimos duas fórmulas para avaliar o nível de qualidade (uma com a totalidade dos atributos e outra com os mais relevantes), sendo a determinação dos atributos mais relevantes feita com recurso ao comando regsubsets, o que permite gerar a tabela apresentada 6.1.

- Definição das camadas de entrada e de saída com todos os atributos:

```
formula01 <- quality ~ fixed.acidity + volatile.acidity + citric.acid +
residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
density + pH + sulphates + alcohol
```

- Definição das camadas de entrada e de saída com os 5 atributos mais relevantes:

```
formula01 <- quality ~ volatile.acidity + residual.sugar + density + pH + alcohol
```

- Determinação dos atributos mais relevantes:

```
regg1<-regsubsets(quality ~ fixed.acidity + volatile.acidity + citric.acid +
residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
density + pH + sulphates + alcohol, dados)
summary(regg1)
```

6.3 Definição da Rede Neuronal

Para o treino da rede neuronal foi utilizada a função neuralnet em R, cujos parâmetros definidos foram:

- formula01: define as camadas de entrada e de saída;
- treino: matriz com os dados para treinar a rede;
- hidden: vetor de inteiros que define a quantidade de nodos por camada intermédia;
- lifesign: especifica o quanto vai ser impresso na execução;
- linear.output: especifica a utilização dos nodos exteriores;
- threshold: valor de erro no qual a função pára;
- stepmax: valor que não pode ser ultrapassado quando a rede está a ser treinada.

```
neuralnet(formula01,treino,hidden,lifesign,linear.output,threshold,stepmax)
```

6.4 Sets

Para a realização dos testes, após o treinamento, é preciso definir sets em que só esteja disponível as colunas dos atributos que são dados em input. Para tal, definimos este *subset* relativo a qualidade do vinho e contém os cinco atributos mais relevantes determinados através da função *regsubsets*.

```
teste.01 <- subset(teste, select = c("volatile.acidity","residual.sugar",
"density","pH","alcohol"))
```

6.5 Qualidade

De seguida encontra-se o resultado do treino da rede neuronal recorrendo aos parâmetros passados na função neuralnet.

```
> rna <- neuralnet( formula01, treino, hidden = c(4), lifesign = "full",
linear.output=FALSE, threshold = 0.1, stepmax= 1e6)

hidden: 4    thresh: 0.1    rep: 1/1    steps:       9 error: 9.48266
time: 0.07 secs
```

Uma vez treinada a rede, procedeu-se então à fase de teste da mesma através da função *compute* e, posteriormente, determinou-se o RMSE.

```
> rmse(c(teste$quality),c(resultados$previsao))
[1] 0.09027300944
```

7 Conclusão

Fazendo uma avaliação global da realização deste projeto, deparámo-nos com algumas dificuldades que são naturais da pouca experiência e da recente aprendizagem da linguagem R, mas que tentámos ultrapassar da melhor forma possível.
O objetivo deste trabalho prático é no fundo aplicar de uma forma global os conhecimentos adquiridos nas aulas práticas durante as últimas semanas. Desta forma, seria esperada a implementação com sucesso das tarefas pedidas, tais como, a utilização de sistemas não simbólicos na representação de conhecimento através de Redes Neuronais Artificiais e posterior otimização e diminuição do erro.
Em primeiro lugar, a deteção dos obstáculos do projeto que poderiam impedir o bom funcionamento da nossa rede, obstáculos alguns de baixo grau de dificuldade, como o tratamento dos dados iniciais, foram ultrapassados com uma certa facilidade. Embora, existiram "contratempos" maiores, como encontrar uma topologia de rede, casos de treino e de teste que nos permitissem o chegar ao menor erro, sendo este problema resolvido com um pouco de imaginação na criação de casos que, após analisados de forma concisa, nos pudessem dar um rumo para o caminho a percorrer para a desconstrução e consequente resolução do exercício que temos em mãos.
Em segundo lugar, todo o trabalho segue uma linha de raciocínio bastante semelhante à seguida nas aulas práticas sendo que apenas existe a necessidade de aplicar essas capacidades aprendidas e utilizá-las de forma útil na resolução deste projeto.
Em suma, fundamentamos as nossas escolhas, tal como é notório na estrutura do trabalho, e apesar de algumas arestas por limar estamos satisfeitos com o plano de ação que elaboramos e o método de trabalho imposto. Contudo, as conclusões não satisfazem os padrões pelos quais este grupo se tem regido, não conseguindo responder de forma sucinta e simplesmente correta ao problema em análise. Concluímos assim o terceiro e ultimo exercício prático da cadeira de Sistemas de Representação de Conhecimento e Raciocínio.

8 Referências

- Textos pedagógicos disponibilizados na página da Unidade Curricular;
- Bibliotecaneuralnet - http://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf;
- Perelli, Layne P. Fatigue Stressors in Simulated Long-Duration Flight. Effects on Performance, Information Processing, Subjective Fatigue, and Physiological Cost. No. SAM-TR-80-49. SCHOOL OF AEROSPACE MEDICINE BROOKS AFB TX, 1980;
- Pimenta A., Carneiro D., Novais P., Neves J., Detection of Distraction and Fatigue in Groups through the Analysis of Interaction Patterns with Com- puters, Intelligent Distributed Computing VIII, Springer-Verlag - Studies in Computational Intelligence, David Camacho, Lars Braubach, Salvatore Venticinque and Costin Badica (Eds) Vol. 570, pp 29-39, ISBN: 978-3- 319-10421-8, 2014;
- Pimenta A., Carneiro D., Novais P., Neves J., Monitoring Mental Fatigue through the Analysis of Keyboard and Mouse Interaction Patterns, Hybrid Artificial Intelligent Systems - 8th International Conference HAIS 2013, Jeng-Shyang Pan, Marios M. Polycarpou, Michael Wózniak, André C. P. L. F. de Carvalho, Héctor Quintián, Emilio Corchado (eds), Lecture Notes in Computer Science, Vol 8073, ISBN 978-3-642-40845-8, pp 222-231, 2013;