UNIVERSIDADE DO MINHO

Conhecimento não simbólico: Redes Neuronais Artificiais

Mestrado integrado em Engenharia Informática

Sistemas de Representação de Conhecimento e Raciocínio

(2ºSemestre/2017-2018)

a77320 Carlos Pedrosa
a78938 David Sousa
a78869 Manuel Sousa

Universidade do Minho,
Departamento de Informática,
4710-057 Braga, Portugal

maio, 2018

Resumo

O presente trabalho tem como principal objetivo motivar os alunos para a utilização de sistemas não simbólicos na representação de conhecimento e no desenvolvimento de mecanismos de raciocínio, nomeadamente, Redes Neuronais Artificiais(RNAs) para a resolução de problemas. Deste modo, pretendemos descrever os esforços efetuados no sentido a atingir o objetivo proposto. De facto, depois de feita a introdução do projeto, é descrita detalhadamente a solução apresentada para dar resposta aos diversos exercícios. Por último, em jeito de conclusão apresentar-se-á uma reflexão crítica sobre os resultados que foram obtidos.

Índice

Índice de Figuras

1. Introdução

A linguagem de programação *R* é orientada para cálculos estatísticos e gráficos, sendo que fornece um ambiente de desenvolvimento integrado para o tratamento de dados.

Deste modo, neste trabalho iremos fazer uso desta linguagem para o estudo de redes neuronais artificias. Assim, estas são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neuronal de organismos inteligentes e que adquirem conhecimento através da experiência.

Inicialmente, foi atribuído a cada grupo, de acordo com a paridade do mesmo, um conjunto de *datasets* diferentes. Sendo assim, como o grupo possui numero par, ficará encarregue de estudar a temática relativa a vinhos.

Em suma, o principal objetivo é através de diferentes testes obter a melhor rede neuronal possível.

2. Preliminares

Para um melhor entendimento do projeto apresentado, é necessário que o leitor apresente conhecimentos na área das redes neuronais artificias, mais concretamente, no uso da linguagem R para tratar a informação. Nesse sentido, a aprendizagem efetuada pelo grupo no decorrer das aulas foi crucial para o desenvolvimento de todo o projeto. Para além disso, uma análise cuidada e prévia do enunciado proposto permitiu identificar qual seria o melhor método a seguir, de modo a dar resposta aos diversos elementos apresentados.

3. Descrição do trabalho e análise dos resultados

Como já referido, o nosso trabalho debruça-se sobre a qualidade do vinho proveniente do norte de Portugal. O objetivo é perceber a influência que os testes físicos e químicos efetuados têm na qualidade destes. Por forma a ter uma métrica de comparação, entre as diferentes redes neuronais, baseamo-nos no resultado da métrica *RMSE*, muito usada para medir diferença entre valores. Assim, pretendemos nos capítulos a seguir exemplificar todos os passos que foram dados para obter o melhor resultado possível.

3.1. Estudar os atributos mais significativos para a representação do conhecimento do problema em análise

Começamos por decidir quais os atributos que deveriam entrar na formula RNA que pretendíamos definir. Assim, seguimos diferentes metodologias, criando, desta forma, quatro fórmulas distintas:

- **Utilização de todos os atributos na fórmula**

  Uma das fórmulas com que efetuamos os diferentes testes, possui todos os atributos do *dataset* em causa. De facto, decidimos tentar perceber se a utilização total destes era realmente importante no estudo que efetuamos e se era com esta fórmula que iriamos conseguir obter o melhor resultado possível.

```
formulaRNA1 <- quality ~ type + fixed.acidity + volatile.acidity + citric.acid +
               residual.sugar + chlorides + free.sulfur.dioxide +
               total.sulfur.dioxide + density + pH + sulphates + alcohol
```

- **Utilização da ferramenta *Weka***

  Weka contém ferramentas para pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização. Neste sentido, este software foi usado com o intuito de obter quais as variáveis mais significativas. Depois de efetuarmos import dos dados normalizados (explicado todo o processo mais à frente), recorremos assim à secção “Select attributes”.

  Os resultados que obtivemos foram os seguintes:

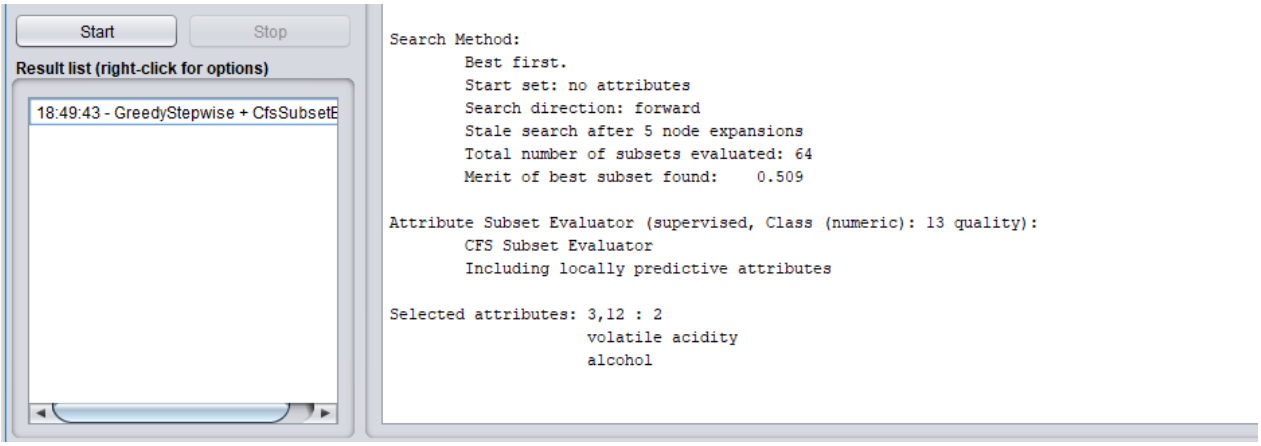

Figura 1 - Variáveis significativas segundo Weka

Pela análise da ferramenta, inferimos que os atributos mais significativos são "volatile acidity" e "alcohol". Construímos assim a seguinte fórmula:

```
formulaWeka <- quality ~ volatile.acidity + alcohol
```

- **Fórmula segundo as métricas do R**

  Para além do *Weka* decidimos usar o R para também estimar quais as variáveis mais significativas. Para o efeito, usamos a libraria *leaps* que fornece funções bastante úteis.

  Inicialmente usamos a função *regsubsets*, de forma a obter o melhor *subset.* Posteriormente, com a função *summary* desenhamos a tabela com a previsão das variáveis mais significativas.

	type	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)	"*"	" "	"*"	" "	"*"	" "	"*"	"*"	"*"	" "	"*"	"*"
9 (1)	"*"	"*"	"*"	" "	"*"	" "	"*"	" "	"*"	"*"	"*"	"*"
10 (1)	"*"	"*"	"*"	" "	"*"	" "	"*"	"*"	"*"	"*"	"*"	"*"
11 (1)	"*"	"*"	"*"	" "	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"
12 (1)	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"

Figura 2 - Tabela obtida com a função *summary*

No entanto, a tabela apresentada mostra as variáveis a usar em função do número de variáveis escolhidas para a construção da fórmula. Por exemplo, se optássemos por usar apenas uma variável usaríamos a "total.sulfur.dioxide".

Surge agora o desafio de escolher o número ótimo de variáveis a usar. Para tal, baseamo-nos em métricas fornecidas pelo *R* que permitem estimar segundo diferentes algoritmos quantas variáveis devem der usadas.

```
# Utilização de metricas para determinar as variáveis gerais mais significativas
best.subset.by.adjr2 <- which.max(best.subset.summary$adjr2)
best.subset.by.adjr2

best.subset.by.cp <- which.min(best.subset.summary$cp)
best.subset.by.cp

best.subset.by.bic <- which.min(best.subset.summary$bic)
best.subset.by.bic
```

Figura 3 - Métricas usadas para determinação do número ótimo de variáveis

Os outputs foram os seguintes:

```
> best.subset.by.adjr2
[1] 11
> best.subset.by.cp
[1] 11
> best.subset.by.bic
[1] 10
```

Figura 4 - Resultados das diferentes métricas

De acordo com a figura acima, duas das métricas indicam que devem ser usadas onze variáveis enquanto que a última indica que devem ser usadas dez. Importa então estudar o que acontece com os gráficos destas métricas.

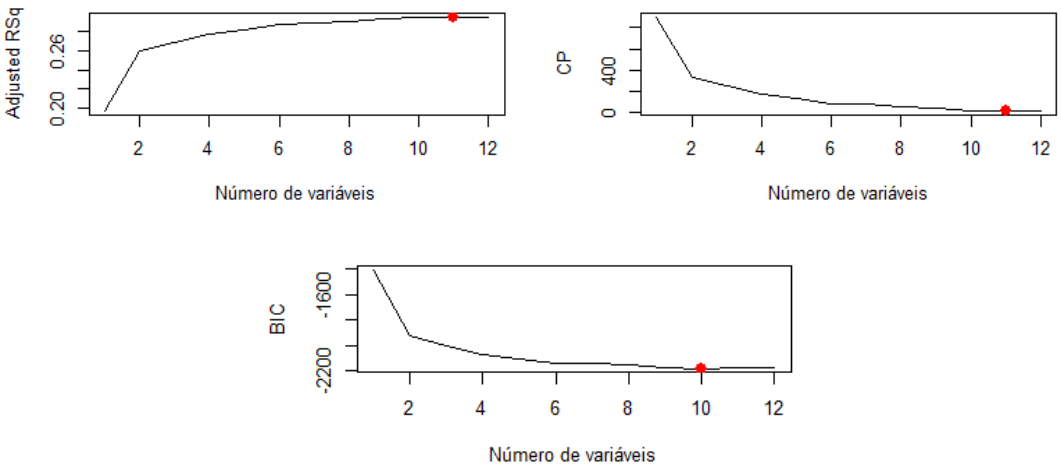

Figura 5 - Gráficos das diferentes métricas

Como podemos ver pelos gráficos, escolher para o critério BIC o número de onze variáveis, não vai produzir grande diferença visto que os valores são bastante próximos. Efetivamente, optamos pelas onze variáveis. Pela tabela anteriormente apresentada, quando o número de variáveis é onze, teremos:

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

- **Escolha das variáveis significativas segundo a informação apresentada**

De forma a perceber a importância de cada atributo, procuramos analisar alguns destes percebendo a relação entre eles.

Acidez fixa: É a soma dos ácidos fixos. Por princípio, quanto mais elevada for a acidez fixa, mais baixa é a volátil. As bactérias acéticas têm dificuldade em desenvolver-se em meios ácidas.

Acidez volátil: É a soma dos ácidos voláteis que se libertam por ebulição ou destilação do vinho, sendo que traduz o nível de ataque acético bacteriano ao vinho.

Acidez Cítrica: Presente nas uvas em baixa quantidade, nos vinhos o ácido cítrico tem pouca ou nenhuma expressão.

Açúcar residual: Em contacto com a levedura, o açucar da uva vai-se transformando em álcool. Em teoria, quanto mais açúcar houver na uva, mais álcool haverá no vinho. O açúcar que resta depois do processo de fermentação ter sido interrompido, é chamado de açúcar residual.

Cloretos: Serve para auxiliar no aumento do sabor, no volume em boca, na característica visual, deixando o vinho com um aspeto mais agradável.

Dióxido de enxofre: Existem dois tipos, SO_2 livre e o SO_2 combinado. O SO_2 total resulta da soma dos dois, sendo que depende de diversos fatores, tais como, teor em açúcar, nível de acetaldeído, pH e temperatura.

Neste sentido, conseguimos apurar que:

- ✓ As acidezes fixas e voláteis estão dependentes uma da outra. Não é necessária a inclusão das duas variáveis na fórmula.
- ✓ O açúcar residual e o álcool também se encontram relacionados. Apenas o álcool é relevante, pois conseguimos determinar, a partir deste, o açúcar residual.
- ✓ Não é necessária a inclusão do dióxido de enxofre livre visto que o dióxido de enxofre total já engloba este.

Obtemos, deste modo, a seguinte fórmula:

```
formulaInfo <- quality ~ type + fixed.acidity + citric.acid +
               chlorides + total.sulfur.dioxide +
               density + pH + sulphates +alcohol
```

3.2. Transformar os dados de modo a que se apresentem adequadamente preparados para a "alimentação" de RNAs

Os *datasets* que nos foram fornecidos, abrangiam dados relativos a dois tipos de vinho distintos, o vinho branco e o vinho tinto. Cada um destes *dataset* possui doze variáveis. Como visto no ponto anterior, as variáveis, "fixed acidity", "volatile acidity", "citric acidity", "residual sugar", "chlorides", "free sulfur dioxide", "total sulfur dioxide", "density", "pH", "sulphates" e "alcohol", foram usadas como input, enquanto que a variável "quality" foi usada com output. No entanto, os dados não se encontravam normalizados. Deste modo, o grupo optou por transformar os dados da forma que considerou ser adequada. Nesse sentido, recorremos ao Excel para normalizar cada um dos dataset fornecidos. Para tal, regemo-nos pela fórmula: x_i

$$z = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

Para além disso, tomamos a decisão de juntar os dois *dataset*, ou seja, criamos um documento csv(comma separated values) em que acrescentamos mais um atributo que fazia referência ao tipo do vinho a que a linha de dados pertencia.

Em suma, a normalização dos dados permitirá obter valores mais coerentes e com um menor número de erros enquanto que a junção dos dois *dataset* agilizará todo o processo.

type	fixed acid	volatile ac	citric acid	residual si	chlorides	free sulfu	total sulfu	density	pH	sulphates	alcohol	quality
0	0.297521	0.413333	0	0.019939	0.111296	0.034722	0.064516	0.206092	0.612403	0.191011	0.202899	5
0	0.330579	0.533333	0	0.030675	0.147841	0.083333	0.140553	0.186813	0.372093	0.258427	0.26087	5
0	0.330579	0.453333	0.024096	0.026074	0.137874	0.048611	0.110599	0.190669	0.418605	0.241573	0.26087	5
0	0.61157	0.133333	0.337349	0.019939	0.109635	0.055556	0.124424	0.209948	0.341085	0.202247	0.26087	6
0	0.297521	0.413333	0	0.019939	0.111296	0.034722	0.064516	0.206092	0.612403	0.191011	0.202899	5
0	0.297521	0.386667	0	0.018405	0.109635	0.041667	0.078341	0.206092	0.612403	0.191011	0.202899	5
0	0.338843	0.346667	0.036145	0.015337	0.099668	0.048611	0.12212	0.179102	0.449612	0.134831	0.202899	5
0	0.289256	0.38	0	0.009202	0.093023	0.048611	0.034562	0.144399	0.51938	0.140449	0.289855	7
0	0.330579	0.333333	0.012048	0.021472	0.106312	0.027778	0.02765	0.186813	0.496124	0.196629	0.217391	7

Figura 6 - Porção do ficheiro Excel produzido

3.3. Selecionar as regras de aprendizagem para treinar a(s) rede(s)

Usamos duas formas diferentes para treinar e testar a rede. Assim, num primeiro momento, optamos por usar dados de treino, o mais uniformes possíveis. Efetivamente, optamos por usar uma amostra de 800 de tamanho para o vinho tinto e uma amostra de 800 de tamanho para o vinho branco, juntando estas duas para fazer o "trainset". Metade do vinho tinto foi usado para treino enquanto que o restante foi usado para teste (daí terem só sido usadas 800 unidades). O "testset", neste caso, seriam as restantes linhas do ficheiro.

```
treinoRed <- trainsetNorm[1:800,]
treinoWhite <- trainsetNorm[5698:6497,]

testeSet <- trainsetNorm[801:5697,]
treinoSet <- rbind(treinoRed,treinoWhite)
```

Para além disso, consideramos ainda usar uma amostra de treino totalmente aleatória[1]. Assim, o ideal poderá não ser uma amostra equilibrada, mas sim uma amostra diversificada. Esta poderá ser mais propicia à rede neuronal, sendo que os restantes dados que não são usados para treino serão usados para teste. O tamanho manter-se-á pelo que será usada uma amostra de treino de 1600 unidades.

[1] Apesar de ser gerada uma amostra aleatória, a função set.seed() permite obter sempre a mesma aleatoriedade para que nos diferentes testes a amostra não se altere. O mesmo propósito se verifica em relação ao "weights" presente na *neuralnet*

```
set.seed(3)
amostra <- sample(nrow(trainsetNorm), 1600)
treinoAl  <- trainsetNorm[amostra,]
testeAl  <- trainsetNorm[-amostra,]
```

3.3. Identificar a(s) topologia(s) de rede mais adequada(s)

Com o intuito de verificar a melhor topologia, foram efetuados diversos testes, usando as várias fórmulas RNA acima definidas. Assim, fizemos uso da libraria *neuralnet* fornecida pelo *R* que ao fornecer a função *neuralnet()* permitiu fazer um estudo diversificado graças à alteração dos vários parâmetros. Assim, o objetivo passou por analisar a métrica *RMSE*, tentando obter o menor valor possível. Foram vários os parâmetros explorados, pelo que foram alterados campos relativos à fórmula RNA, ao algoritmo, número de nodos e camadas da rede neuronal. Para além disso, também foram efetuados testes para diferentes amostras. Os diferentes testes foram organizados em tabelas para que fosse mais fácil a sua análise. Foram realizados 12 testes, mais dois testes extra. Estes testes extra foram realizados de acordo com o melhor valor obtido nos 12 testes anteriores, assim, eram alterados mais alguns parâmetros com o intuito de melhorar ainda mais o valor já obtido.

- **FormulaRNA1, com os trainset uniformes:**

Nº Teste	Algoritmo	Nº nodos (1ªCamada)	Nº nodos (2ªCamada)	Nº nodos (3ªcamada)	Iterações	RMSE
1	backprop	2	2	-	-	-
2	backprop	4	8	-	28	0.909244164
3	backprop	9	7	5	35	0.909245652
4	backprop	2	3	4	25	0.909360472
5	rprop+	2	2	-	5355	0.874053895
6	rprop+	4	8	-	35051	0.987284953
7	rprop+	9	7	5	-	-
8	rprop+	2	3	4	3915	0.928349076
9	rprop-	2	2	-	3931	0.841073015
10	rprop-	4	8	-	-	-
11	rprop-	9	7	5	60716	0.95405755
12	rprop-	2	3	4	-	-
EXTRA	rprop-	3	2	-	4324	0.822136535
EXTRA	rprop-	1	4	-	2627	0.85927772
						0.822136535

Em geral, para esta fórmula, o algoritmo "rprop-", é o mais benéfico, sendo que o número de cadas como duas e consequentemente, os nodos de 3 e 2 respetivamente, permitem um

RMSE mais baixo. É de realçar que em alguns casos, o algoritmo utilizado não permitiu a convergência dos valores para o *STEPMAX* definido.

- **FormulaRNA1, com os trainset aleatórios:**

Nº Teste	Algoritmo	Nº nodos (1ªCamada)	Nº nodos (2ªCamada)	Nº nodos (3ªcamada)	Iterações	RMSE
1	backprop	2	2	-	-	-
2	backprop	4	8	-	29	0.8723884
3	backprop	9	7	5	35	0.872391916
4	backprop	2	3	4	25	0.872466282
5	rprop+	2	2	-	13759	0.825068145
6	rprop+	4	8	-	76419	1.280737093
7	rprop+	9	7	5	-	-
8	rprop+	2	3	4	-	-
9	rprop-	2	2	-	6335	0.848169747
10	rprop-	4	8	-	-	-
11	rprop-	9	7	5	-	-
12	rprop-	2	3	4	12392	0.812565025
EXTRA	rprop-	2	4	3	8565	0.808817886
EXTRA	rprop-	2	4	5	8242	0.80977409
						0.808817886

Neste caso, novamente se destaca o "rprop-", com três nodos, 2 na primeira camada 4 na segunda e 3 na terceira. Obtivemos um RMSE mínimo de 0.808817886.

- **FormulaWeka, com os trainset uniformes:**

Nº Teste	Algoritmo	Nº nodos (1ªCamada)	Nº nodos (2ªCamada)	Nº nodos (3ªcamada)	Iterações	RMSE
1	backprop	2	2	-	-	-
2	backprop	4	8	-	-	-
3	backprop	9	7	5	29	0.909261746
4	backprop	2	3	4	-	-
5	rprop+	2	2	-	319	0.782474902
6	rprop+	4	8	-	23717	0.865407622
7	rprop+	9	7	5	-	-
8	rprop+	2	3	4	247	0.782626613
9	rprop-	2	2	-	1929	0.788134971
10	rprop-	4	8	-	841	0.78208873
11	rprop-	9	7	5	-	-
12	rprop-	2	3	4	449	0.782299098
EXTRA	rprop-	7	8	-	185	0.78203832
EXTRA	rprop-	7	7	-	56332	0.870467682
						0.78203832

Neste caso, novamente se destaca o "rprop-", com dois nodos, 7 na primeira camada e 8 na segunda. Obtivemos um RMSE mínimo de 0.78293832.

- **FormulaWeka, com os trainset aleatório:**

Nº Teste	Algoritmo	Nº nodos (1ªCamada)	Nº nodos (2ªCamada)	Nº nodos (3ªcamada)	Iterações	RMSE
1	backprop	2	2	-	-	-
2	backprop	4	8	-	-	-
3	backprop	9	7	5	29	0.872387842
4	backprop	2	3	4	-	-
5	rprop+	2	2	-	5577	0.740570262
6	rprop+	4	8	-	-	-
7	rprop+	9	7	5	513	0.740515437
8	rprop+	2	3	4	4398	0.750739875
9	rprop-	2	2	-	256	0.750935377
10	rprop-	4	8	-	14280	0.741044162
11	rprop-	9	7	5	-	-
12	rprop-	2	3	4	698	0.750992641
EXTRA	rprop+	2	1	-	771	0.740522894
EXTRA	rprop+	1	2	-	3220	0.740522894
						0.740515437

Nos trainset aleatório obtivemos melhores resultados, com um menor número de nodos.

- **FormulaMetr, com os trainset uniforme:**

Nº Teste	Algoritmo	Nº nodos (1ªCamada)	Nº nodos (2ªCamada)	Nº nodos (3ªcamada)	Iterações	RMSE
1	backprop	2	2	-	-	-
2	backprop	4	8	-	28	0.909244993
3	backprop	9	7	5	34	0.909260511
4	backprop	2	3	4	66	0.909371184
5	rprop+	2	2	-	3184	0.804602593
6	rprop+	4	8	-	38077	0.832144217
7	rprop+	9	7	5	-	-
8	rprop+	2	3	4	-	-
9	rprop-	2	2	-	2824	0.804814516
10	rprop-	4	8	-	44534	0.838002706
11	rprop-	9	7	5	-	-
12	rprop-	2	3	4	-	-
EXTRA	rprop+	2	3	-	3701	0.790350434
EXTRA	rprop+	7	7	-	7458	0.816649756
						0.790350434

O melhor valor obtido foi, com o algoritmo "rprop+" sendo que apenas usamos duas camadas com dois e três nodos respetivamente.

- **FormulaMetr, com os trainset aleatório:**

Nº Teste	Algoritmo	Nº nodos (1ªCamada)	Nº nodos (2ªCamada)	Nº nodos (3ªcamada)	Iterações	RMSE
1	backprop	2	2	-	-	-
2	backprop	4	8	-	28	0.872392066
3	backprop	9	7	5	34	0.872388137
4	backprop	2	3	4	25	0.87247807
5	rprop+	2	2	-	13336	0.721131964
6	rprop+	4	8	-	33195	0.761526408
7	rprop+	9	7	5	-	-
8	rprop+	2	3	4	12963	0.718432188
9	rprop-	2	2	-	5862	0.719313283
10	rprop-	4	8	-	-	-
11	rprop-	9	7	5	-	-
12	rprop-	2	3	4	59747	0.731628963
EXTRA	rprop+	3	2	4	17122	0.722974037
EXTRA	rprop+	2	1	2	12574	0.718390594
						0.718390594

Neste caso o melhor *RMSE* corresponde ao "rprop+" com três camadas, 2, 1 e 2 nodos respetivamente.

- **FormulaInfo, com os trainset uniforme:**

Nº Teste	Algoritmo	Nº nodos (1ªCamada)	Nº nodos (2ªCamada)	Nº nodos (3ªcamada)	Iterações	RMSE
1	backprop	2	2	-	-	-
2	backprop	4	8	-	28	0.909246531
3	backprop	9	7	5	33	0.909246536
4	backprop	2	3	4	1283	0.909369626
5	rprop+	2	2	-	20384	0.842394235
6	rprop+	4	8	-	65393	0.854707763
7	rprop+	9	7	5	-	-
8	rprop+	2	3	4	-	-
9	rprop-	2	2	-	36275	0.819861726
10	rprop-	4	8	-	-	-
11	rprop-	9	7	5	-	-
12	rprop-	2	3	4	3367	0.81113338
EXTRA	rprop+	2	2	3	3119	0.809905797
EXTRA	rprop+	2	2	2	11858	0.816656224
						0.809905797

Novamente, o número de camadas corresponde a 3 com o número de nodos de 2, 2 e 3 respetivamente.

- **FormulaInfo, com os trainset aleatório:**

Nº Teste	Algoritmo	Nº nodos (1ªCamada)	Nº nodos (2ªCamada)	Nº nodos (3ªcamada)	Iterações	RMSE
1	backprop	2	2	-	-	-
2	backprop	4	8	-	28	0.872391685
3	backprop	9	7	5	33	0.872392123
4	backprop	2	3	4	1067	0.872519026
5	rprop+	2	2	-	5801	0.75692932
6	rprop+	4	8	-	-	-
7	rprop+	9	7	5	-	-
8	rprop+	2	3	4	2321	0.75809859
9	rprop-	2	2	-	14060	0.758903585
10	rprop-	4	8	-	85934	0.790868939
11	rprop-	9	7	5	-	-
12	rprop-	2	3	4	2087	0.757691275
EXTRA	rprop+	1	2	-	17475	0.765754778
EXTRA	rprop+	2	1	-	30540	0.758760938
						0.75692932

O melhor *RMSE* obtido neste caso possui o valor de 0.75692932.

Considerações finais:

- ✓ Regra geral, a fórmula que permite melhores resultados é a formulaMetr;
- ✓ Os algoritmos que se destacaram foram “rprop+” e “rprop-”;
- ✓ Testes com dados aleatórios permitem obter melhores resultados;
- ✓ Por vezes, não é possível obter resultados para o *STEPMAX* definido, o que poderá dever-se a threshold demasiado alto.

Para além disto, ao longo do trabalho foi percetível a influência que um correto tratamento dos dados, tem nos resultados obtidos. Para além disso, é ainda visível que nem sempre, um maior número de camadas ou até mesmo nodos, permite obter melhores resultados.

Assim, a rede neuronal ótima será a seguinte:

```
rna1 <- neuralnet(formulaRNA1, treinoA1, hidden = c(2,1,2), lifesign = "full", linear.output = TRUE,
                  threshold = 0.1, algorithm = "rprop+", startweights = weights )
```

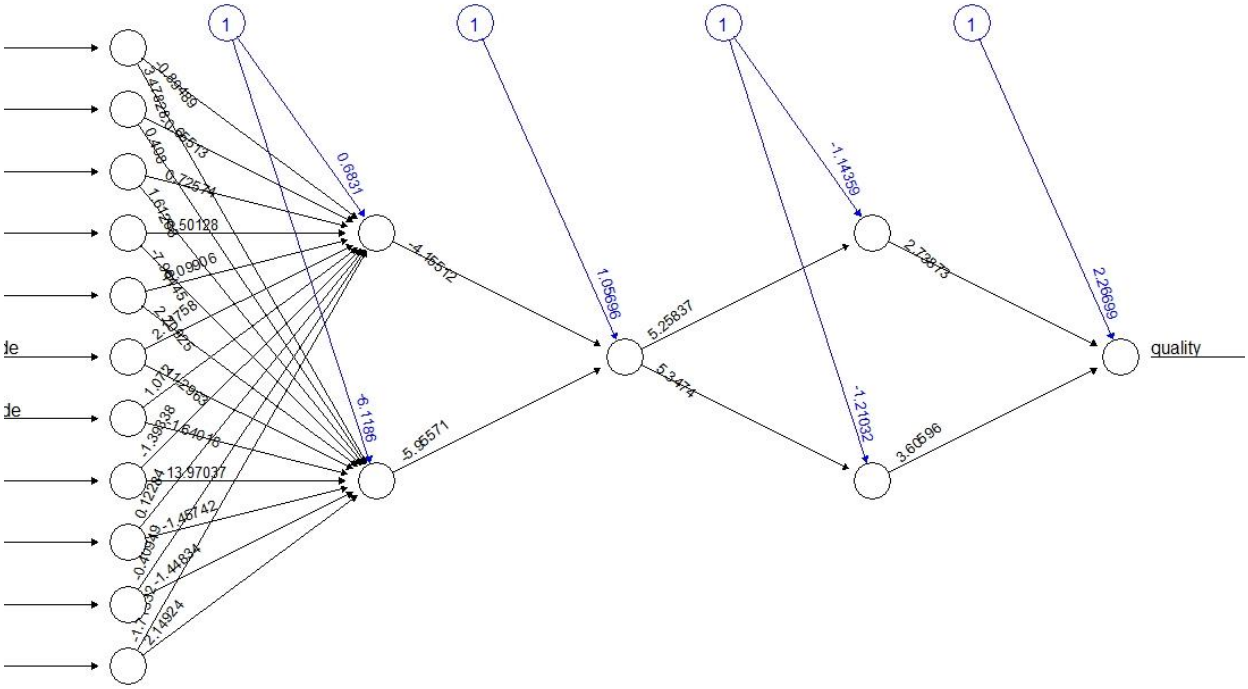

Figura 7 - Rede neuronal ótima obtida

Conclusões e sugestões

De forma geral consideramos que o objetivo do trabalho foi concluído. Nesse sentido, o projeto apresentado foi crucial para a consolidação de conceitos relativos a redes neuronais artificias e às variantes que esta apresenta.

O principal desafio com que nos deparamos foi na basta possibilidades de escolha que existem no R para controlo das redes neuronais. No entanto, consideramos que seguimos o melhor método abrangendo, deste modo, um conjunto bastante amplo de possibilidades.

Em suma, este trabalho formatou a nossa maneira de pensar aquando da resolução de um problema, tornando-nos mais objetivos e racionais. Foi assim possível um pensamento mais crítico relativamente a redes neuronais artificiais.

4. Referências

- Ivan Bratko, "PROLOG: Programming for Artificial Intelligence", 3rd Edition, Addison-Wesley Longman Publishing Co., Inc., 2000;

- Hélder Coelho, "A Inteligência Artificial em 25 lições", Fundação Calouste Gulbenkian, 1995;

- “Sugestões para a Redacção de Relatórios Técnicos, Cesar Analide, Paulo Novais, José Neves.

Anexos

I. Anexo 1 –Código R auxiliar

```r
# Equilibrar os arrays com o mesmo número de vinhos
weights <- seq(from = -1, to = 200, by = 0.01)


# Plot para a percepção das métricas
par(mfrow=c(2,2))
plot(best.subset.summary$adjr2, xlab="Número de variáveis", ylab="Adjusted RSq", type="l")
points(best.subset.by.adjr2, best.subset.summary$adjr2[best.subset.by.adjr2], col="red", cex =2, pch =20)
plot(best.subset.summary$cp, xlab="Número de variáveis", ylab="CP", type="l")
points(best.subset.by.cp, best.subset.summary$cp[best.subset.by.cp], col="red", cex =2, pch =20)
plot(best.subset.summary$bic, xlab="Número de variáveis", ylab="BIC", type="l")
points(best.subset.by.bic, best.subset.summary$bic[best.subset.by.bic], col="red", cex =2, pch =20)
```