UNIVERSIDADE DO MINHO

Machine Learning de Qualidade de Vinhos

Mestrado Integrado de Engenharia Informática

Sistemas de Representação de Conhecimento e Raciocínio

(2ºSemestre/2017-2018)

Número	Nome do(s) Autor(es)
a78468	João Vieira
a78821	José Martins
a77049	Miguel Quaresma
a77689	Simão Barbosa

Braga, Portugal
May 11, 2018

1 Resumo

As redes neuronais artificiais são técnicas computacionais que apresentam um modelo matemático inspirado no cérebro (estrutura neuronal) de organismos inteligentes. Estas redes são capazes de realizar aprendizagem e de adquirirem conhecimento através da experiência com dados recebidos.
Neste projeto, o objetivo passa por criar uma rede neuronal capaz de estudar e aprender tendo em conta dados relativos a dois tipos de vinhos, branco e tinto, considerando como *input* informações relativas dos vinhos como o álcool ou o pH, e sendo o *output* a qualidade do mesmo.
Este relatório mostra e explica o processo de desenvolvimento de um programa utilizando a linguagem de programação *R* capaz de efetuar o objetivo pretendido, com um valor de erro bastante pequeno.

Contents

2 Introdução

As redes neuronais são cada vez mais usadas na área de informática e engenharia visto obter-se bons resultados para problemas por vezes complicados de codificar, tais como decifrar a escrita de uma pessoa. Para além disso permite encontrar padrões por vezes não percetiveis para os humanos. É por isso que neste trabalho usamos as redes neuronais de modo a que perante determinas caracteristicas de um vinho a mesma diga a qualidade do vinho. Sempre que se pretende construir uma rede neuronal é necessário *datasets* de modo a poder treinar a rede neuronal. Os *datasets* usados referentes a vinho branco e tinto podem ser encontrados em `https://archive.ics.uci.edu/ml/datasets/Wine+Quality`.

3 Descrição do Trabalho e Análise de Resultados

Como os dados a serem tratados neste projeto são dois *datasets* sobre vinhos (um de vinho branco e outro de vinho tinto) com as mesmas características, começamos por ler os mesmos e junta-los num só, realizando após a junção um sorteio da ordem de cada linha.

```
# leitura dos dados a processar
dadosRed <- read.csv(".../winequality-red.csv",header=TRUE,sep=";",dec=".")
dadosWhite <- read.csv(".../winequality-white.csv",header=TRUE,sep=";",dec=".")

# junção dos dados
dados <- rbind(dadosRed, dadosWhite)
```

Após isto, avaliamos os dados e procedemos à normalização dos mesmos, sendo que o *output* (quality) foi multiplicado por 0.1 de modo a tornar-se normalizado.

```
dadosR[,1:11] <- normalize(dadosR[,1:11], method="range", range=c(0,1))
dadosR[,12] <- dadosR[,12]*0.1
```

Realizamos de seguida um teste aos dados de modo a saber quais os valores mais influentes no valor *output*. Tendo em conta os resultados decidimos criar as seguintes fórmulas:

```
funcao <- quality ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+
                    free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol
funcaoOpt <- quality ~ alcohol+volatile.acidity+sulphates+residual.sugar+total.sulfur.dioxide
funcaoOpt2 <- quality ~ alcohol+volatile.acidity+sulphates
```

sendo que a primira (`funcao`) tem em conta todos os valores das caraterísticas dos vinhos presentes nos ficheiros de *input*, a segunda (`funcaoOpt`) apenas 5 destes valores, e por último apenas temos em conta 3 deles (`funcaoOpt2`).
Após isso decidimos dividir os dados em duas partes, uma parte para treinar a rede neuronal (os primeiros 4500) e o resto para testar a rede neuronal treinada. Por forma a descobrir a melhor rede neuronal, ou seja, a que obtém o menor erro com os dados de teste, é necessário realizar várias tentativas e verificações de resultados obtidos com as mesmas. O processo seguido por nós foi executado pela seguinte ordem:

- Treinar a rede neuronal:

```
rnaWine <- neuralnet(funcao, dadosTreino,
      lifesign=‘‘full’’, hidden = c(7,5), threshold = 0.01)
```

- Preparar os dados de teste:

```
dadosTeste1 <- subset(dadosTeste, select = c(‘‘fixed.acidity’’,
      ‘‘volatile.acidity’’, ‘‘citric.acid’’, ‘‘residual.sugar’’,
      ‘‘chlorides’’, ‘‘free.sulfur.dioxide’’, ‘‘total.sulfur.dioxide’’,
      ‘‘density’’, ‘‘pH’’, ‘‘sulphates’’, ‘‘alcohol’’))
```

- Testar a rede:

```
rnaWine1.resultados <- compute(rnaWine1,dadosTeste1)
```

- Comparar resultados:

```
resultados1 <- data.frame(atual = dadosTeste$quality,
                  previsao = rnaWine1.resultados$net.result)
```

- Arredondar os resultados:

```
resultados1$previsao <- round(resultados1$previsao, digits=1)
```

- Calcular o RMSE:

```
rmse(c(dadosTeste$quality),c(resultados1$previsao))
```

O arredondamento dos resultados é realizado com 1 casa decimal tendo em conta que o *output* dos *datasets* não toma o valor de 0 ou 1 (falso ou verdadeiro) mas sim um valor compreendido entre 0 e 10 (ou após serem normalizados de 0 a 1, de 0.1 em 0.1).
É também importante referir que como variamos a fórmula entre as várias tentativas na preparação de dados de teste os dados definidos mudam consoante a fórmula. É de notar também que durante os testes é variado o *hidden* e o *threshold* da rede neuronal, com o objetivo de obter os melhores resultados possíveis. O *hidden* é referente às camadas intermédias da rede neuronal, enquanto que a fórmula inserida define os neurónios de entrada e de saída.

3.1 Tentativas

3.1.1 threshold=0.1

Com funcao

Rede Neuronal	Erro
$11 \rightarrow 6 \rightarrow 1$	0.07857646844
$11 \rightarrow 3 \rightarrow 3 \rightarrow 1$	0.07777579064
$11 \rightarrow 10 \rightarrow 8 \rightarrow 1$	0.07716171911
$11 \rightarrow 7 \rightarrow 4 \rightarrow 1$	0.07611629613

Com funcaoOpt

Rede Neuronal	Erro
$5 \rightarrow 5 \rightarrow 3 \rightarrow 1$	0.08027868335

Com funcaoOpt2

Rede Neuronal	Erro
$3 \rightarrow 6 \rightarrow 2$	0.07958958319

3.1.2 threshold=0.01

Com funcao

Rede Neuronal	Erro
$11 \rightarrow 6 \rightarrow 3 \rightarrow 1$	0.07598460693
$11 \rightarrow 7 \rightarrow 4 \rightarrow 1$	0.07738852289
$11 \rightarrow 12 \rightarrow 4 \rightarrow 1$	0.07732378969
$11 \rightarrow 5 \rightarrow 3 \rightarrow 1$	0.0760833952
$11 \rightarrow 10 \rightarrow 1$	0.07528945978
$11 \rightarrow 8 \rightarrow 7 \rightarrow 1$	0.09155639716
$11 \rightarrow 7 \rightarrow 5 \rightarrow 1$	0.07595164896

Com funcaoOpt

Rede Neuronal	Erro
$5 \rightarrow 6 \rightarrow 2 \rightarrow 1$	0.07628058791
$5 \rightarrow 8 \rightarrow 4 \rightarrow 1$	0.07761466391
$5 \rightarrow 4 \rightarrow 1$	0.07860832592
$5 \rightarrow 6 \rightarrow 3 \rightarrow 1$	0.07719416045
$5 \rightarrow 6 \rightarrow 5 \rightarrow 1$	0.07780797595
$5 \rightarrow 4 \rightarrow 3 \rightarrow 1$	0.07784014795
$5 \rightarrow 4 \rightarrow 2 \rightarrow 1$	0.07738852289
$5 \rightarrow 3 \rightarrow 2 \rightarrow 1$	0.07835310341
$5 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 1$	0.07706431316

Com funcaoOpt2

Rede Neuronal	Erro
$3 \rightarrow 4 \rightarrow 1$	0.07946365023
$3 \rightarrow 2 \rightarrow 1$	0.07999749621
$3 \rightarrow 2 \rightarrow 2 \rightarrow 1$	0.07930595281
$3 \rightarrow 3 \rightarrow 2 \rightarrow 1$	0.07984085322
$3 \rightarrow 4 \rightarrow 3 \rightarrow 1$	0.07971531721
$3 \rightarrow 4 \rightarrow 4 \rightarrow 1$	0.0796839023
$3 \rightarrow 5 \rightarrow 1$	0.0796839023
$3 \rightarrow 5 \rightarrow 3 \rightarrow 1$	0.07996619216
$3 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 1$	0.07930595281
$3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 1$	0.08015383307

O menor valor de erro conseguido foi obtido como esperado usando todas as caracteristicas dos vinhos, contudo, o erro não aumenta assim tão significantemente se se usar apenas as 5 principais caracteristicas. É de referir também que mesmo usando apenas 3 caraterísticas se consegue um valor para o erro obtido que não é assim tão diferente dos referidos anteriormente.
Desta forma, podemos concluir com isto que não usando todas as caraterísticas dos vinhos, ou seja, treinando uma rede neuronal de uma forma não tão complexa, consegue-se bom resultados e um nível de erro bastante pequeno. Sendo assim, a melhor rede neuronal seria $11 \rightarrow 7 \rightarrow 5 \rightarrow 1$ com um erro de aproximadamente 7.595% de a resposta dada pela rede ser errada. Contudo uma boa alternativa seria a rede $5 \rightarrow 6 \rightarrow 2 \rightarrow 1$ com um erro de aproximadamente 7.628% de a resposta dada pela rede ser incorreta.

4 Conclusões e Sugestões

Após várias tentativas de redes neuronais construídas, chegamos à conclusão de que não é de uma importância relevante usar todas as caracteristicas dos vinhos recebidas como *input*, porque é possível verificar que o erro diminui pouco em relação a uma rede com as mais importantes caraterísticas (respetivamente, utilizando apenas 5 e 3 dados), e também porque o tempo de aprendizagem da rede neuronal utilizando todos os dados é muito superior, acabando por não compensar a diferença entre tempos tendo em conta a pequena diferença nos erros verificada. Uma possível melhoria para o programa passaria por continuar a testar novas redes neuronais, até criar um programa que em ciclo teste por si só as diferentes redes neuronais.