**Abordagem Adotada**

**Apresentação dos atributos**

Fazer a tradução do dataset (sem falar de que o duration não irá ser usado)

**Tratamento dos Dados**

De entre o lote de atributos apresentados à partida decidimos logo excluir o atributo \textit{duration}, já que na própria página onde obtivemos o \textit{dataset} utilizado no trabalho, tinha uma pequena indicação que aconselhava à não utilização deste atributo para algoritmos preditivos. Aceitamos esta indicação pelo que então não usamos o atributo \textit{duration}.

Após uma pequena verificação dos dados presentes no \textit{dataset} verificámos que os seguintes dados, “job, marital, education, default, housing, loan, contact, month, day\_of\_week ,poutcome, y”, são do tipo alfanumérico. O que se apresentava um problema, já que a função usada para treinar a rede, \textit{neuralnet}, só aceita inputs do tipo numéricos. Foi necessário então proceder à conversão destes atributos para o tipo requerido. Desta forma, aplicámos o seguinte procedimento, “as.numeric(atributo)”, a cada uma das colunas correspondentes aos atributos mencionados anteriormente. Desta forma, todos os valores presentes no nosso \textit{dataset} apresentavam-se com os tipos requerido para criar a rede neuronal.

De salientar que, que apesar de ser do tipo alfanumérico, todos os atributos tinham um conjunto de valores possíveis, pelo que o valor numérico atribuído também estaria num intervalo finito. Por exemplo os valores do campo “job” estavam limitados a 12 diferentes, pelo que o intervalo numérico estaria entre 1 e 12.

A nossa estratégia passava por limitar os dados de input fornecidos à rede, escolhendo quais dos atributos existentes os que poderiam ter mais influência sobre resultado final.

Antes de efetuar esta seleção, efetuámos uma análise dos dados. Nomeadamente qual o máximo, mínimo e a média de cada atributo, através da visualização “summary” dos dados e ainda analisámos o desvio padrão, para ter uma melhor noção sobre como é a variação dos resultados apresentados. Com esta análise pretendíamos eliminar os dados, que à partida poderiam ser viciosos, para que se fizesse uma correta seleção dos dados. Por exemplo, se um atributo, tivesse todos os valores iguais, por exemplo a 1, seria expectável, que, no final da análise para a seleção dos dados este aparecesse como o mais importante. O que estaria errado, já que este atributo não varia, viciando assim a qualidade dos resultados obtidos.

COLOCAR UMA FOTO DO DESVIAO PADRAO E DO SUMMARY?

Assim, à partida, eliminamos mais 4 atributos, “previous, default, loan, pdays”. Todos estes atributos foram eliminados pela mesma razão. Só pela análise do máximo/mínimo e da sua média que se aproximava bastante a um destes 2 parâmetros, dava a entender de que possuíam muitos registos com o mesmo valor. O que se veio a concluir, por exemplo, no caso do “pdays” dos 4119 valores cerca de X tinham o valor de 999, e valores semelhantes foram apresentados nos restantes atributos.

De ressalvar que até o próprio output “y” nos pareceu um dado vicioso, já que cerca de 90% das entradas têm o valor de “no” (transformado então para o valor numérico 1) e as restantes apresentam o valor de “yes”. Mas como se tratava de um dado de output, não o eliminámos nem efetuamos qualquer manipulação do mesmo, a não a transformação para um tipo de dados numérico, por razões mencionadas anteriormente.

Também efetuamos um agrupamento dos dados, para testar a alteração nos resultados. Por exemplo, no campo da idade, efetuámos dois agrupamentos, um baseado nos nossos pressupostos e posteriormente usando o “discretize” do R.

O primeiro agrupamento baseou-se na divisão das idades por faixas etárias. O valor mínimo da idade é de 18 anos e o máximo de 88. Decidimos então efetuar 3 divisões, uma dos 18 aos 30 anos, que para nós seria a faixa etária mais jovem, que apesar de se calhar serem o que mais poderiam precisar, poderiam estar mais reticentes devido a uma instabilidade financeira. Depois dos 30 aos 65, por considerarmos que é uma idade onde o individuo já tem o estatuto de adulto e que em princípio terá uma maior estabilidade financeira, e depois para cima de 65 anos, que já estão numa faixa etária que em princípio já não pretende efetuar créditos. Com esta divisão, esperávamos uma maior influência do grupo entre os 30 e 65 anos.

Para além desta abordagem, utilizamos o “discretize” do R, quer em 3 ou 4 conjuntos, deixando ao critério desta função, qual seria os intervalos mais adequados. Apesar da efetuação destes clusters, a definição dos atributos mais importantes, o treino da RNA e ainda os resultados obtidos não tiveram alterações significativas, pelo que decidimos não efetuar o cluster.

Para realizar uma transformação nos dados de modo a acelerar e melhorar o processo de aprendizagem, decidimos efetuar uma normalização dos dados de input, colocando-os aproximadamente com uma média de zero e num intervalo [-1,1].

A função utilizada para normalizar os dados foi a seguinte:

valorNormalizado =\frac{valor - \frac{valorMax+valorMin}{2}}{\frac{valorMax-valorMin}{2}}

Após todo o tratamento dos dados, efetuamos a seleção dos inputs. Desta forma, extraíamos a informação mais relevante, e de certa forma iriamos reduzir o tempo de processamento da RNA (já que teríamos menos inputs) e ainda, de uma forma geral, redes maiores têm uma capacidade de aprendizagem mais difícil.

Para esta seleção, utilizámos a função “regsubstes” fornecida pelo R, sendo que os atributos mais relevantes foram os seguintes: “age, education, month, emp.var.rate, cons.price.idx, euribor3m, poutcome, nr.employed”.

COLOCAR AQUI A FOTO?

Foi dentro deste conjunto de dados, que efetuámos mais testes, para a obtenção dos resultados, sendo que o resultado final foi até obtido com os atributos presentes deste conjunto.

Análise de Resultados:

Os testes efetuados para a tentativa de obtenção do melhor resultado possível estão apresentados em anexo.

Para os testes efetuados efetuamos uma seleção dos dados de input, posteriormente efetuamos vários testes de treino da RNA, com diferentes algoritmos ou então diferentes camadas, e por fim analisámos os resultados obtidos através de métricas que o R, com a biblioteca “hydroGOF”, nos disponibilizou. São estes o rmse, nrmse e o ssq.

No anexo A1 foram feitos essencialmente 3 conjunto de testes. Um onde colocávamos como input todas as 19 variáveis que possuíamos, outro onde se fez a seleção das 8 que mais relevância tinham e um último conjunto onde se fazia a seleção a seleção das 8 variáveis (que foram: "age", "campaign", "education", "month", "poutcome", "cons.price.idx", "cons.conf.idx", "nr.employed") mas das 15 que inicialmente tínhamos disponível (para o resto dos testes apenas utilizamos o conjunto de dados que não possuía colunas que poderiam viciar os dados). De salientar que nestes testes realizados não foi feita qualquer manipulação dos dados. O que se verificou, foi que não houve mudança nos valores apresentados pelas diferentes redes testadas.

No anexo A2 apresentamos mais testes, mas desta vez onde se fez um agrupamento de alguns dados. Para tal usamos o “discretize” do R. O nível de agrupamento utilizado foi decidido consoante a observação dos dados por parte do grupo. De notar que, nos atributos: “loan, housing, campaign, poutcome, emp.var.rate” não foi feito este agrupamento. A seleção dos 8 atributos mais relevantes foi igual à do anexo1 e mais uma vez os resultados obtidos não variaram.

No anexoA3 procedemos a uma normalização dos dados entre -1 e 1. Houve uma mudança na seleção dos dados, passando os seguintes atributos: "age", "education", "month", "poutcome", "emp.var.rate", "cons.price.idx", "euribor3m", "nr.employed".

Com a normalização dos dados já obtivemos resultados mais interessantes e que variavam, quando nós efetuávamos uma mudança da RNA que estávamos a treinar. Com estes testes realizados, podemos comprovar que quanto mais aumentarmos nos nodos escondidos, maior será a complexidade da rede e então aumentará o tempo de aprendizagem, e na maioria dos casos os resultados obtidos foram até inferiores dos resultados apresentados por uma rede menos complexa, provando que nem sempre uma rede mais complexa apresenta então melhores resultados.

Por fim no anexoA4 decidimos juntar todas as técnicas, ou seja, efetuámos uma normalização e um agrupamento dos dados e ainda a seleção dos mesmos ("age", "education", "month", "poutcome", "emp.var.rate", "cons.price.idx", "campaign", "nr.employed"). Os resultados, em comparação com os dois primeiros anexos mostraram-se mais satisfatórios, porém em comparação com os resultados de só efetuarmos normalização foram um bocado piores. De salientar, que neste grupo de testes conseguimos obter por uma vez 89 “yes” no output estimado pela rede, porém com um erro de 46%. Nós testávamos este número apenas por curiosidade, já que não refletia ao certo a \textit{accuracy} da rede, já que de entre os valores estimados dava bastantes “2” que deveriam de ser “1”.

SELEÇÃO DA MELHOR REDE