Computação Natural

Bruno Pereira A75135 | João Coelho A74859 | Luís Fernandes A74748 | Maria Ana de Brito A73580

8 de Abril de 2018

Resumo

A previsão meteorológica tem por base, entre outros fatores, a utilização dos dados previamente recolhidos em dias ou anos anteriores (período homólogo). Estes são obtidos de estações meteorológicas, satélites ou radares de hora a hora ou com uma periodicidade diária. Neste trabalho pretende-se, considerando um dos *datasets* com dados das cidades de Cruzeiro do Sul (latitude: 7°37'S; longitude: 72°40'W; altitude: 170 m), Picos (latitude: 7°04'S; longitude: 41°28'W; altitude: 208 m) e Campos de Jordão (latitude: 22°45'S; longitude: 45°36'W; altitude: 1642 m), fazer a avaliação da previsão meteorológica (em termos de temperatura mínima e máxima) aplicando diversas configurações e algoritmos de aprendizagem.

Conteúdo

1 Introdução

As Redes Neuronais Artificiais e a sua capacidade de reconhecimento de padrões mostram-se como a ferramenta ideal para interpretar o problema. Assim, os objetivos deste trabalho passam por aplicar métodos de análise às séries temporais do *dataset*, com o objetivo de identificar as dependências entre os dados, particionar os dados em *subsets* de aprendizagem (treino) e de teste, desenvolver a arquitetura da RNA que melhor se adequa ao problema em causa, criar padrões de treino e testar a RNA com o conjunto de dados reservados para teste.

Para a construção e aplicação da RNA, escolheu-se o *package* de R "neuralnet". Esta biblioteca pode não ser a escolha mais apropriada para projetos mais complexos, mas na nossa opinião adequa-se ao problema e, por outro lado, tem na simplicidade da sua utilização uma vantagem. O treino da rede assenta na chamada da função neuralnet(), cujos parâmetros permitem controlar fatores como o número de camadas internas, a quantidade de nodos nas camadas, o número de iterações, a taxa de aprendizagem ou, como consta da descrição oficial acima, o erro e função de ativação. Outra vantagem associada a este *package* é a ferramenta de desenho do modelo criado, com representação dos pesos das várias ligações, invocada pela chamada à função plot(nn).

2 Preparação dos *datasets* originais

Rumo ao objetivo de realizar a previsão meteorológica com base em dados anteriormente recolhidos, adotou-se uma estratégia comum de extração de conhecimento, sendo o primeiro passo a análise dos *datasets* das três cidades cujos dados meteorológicos foram recolhidos: Campos de Jordão, Cruzeiro do Sul e Picos. Em todos foi verificada a existência de *missing values*, na sua maioria representados pela falta de valor no campo respetivo, mas também aparecendo como '#N/A'.

A existência de valores em falta implica, desde logo, o tratamento dos dados. Assim, optou-se pela introdução do símbolo '?' nas células dos *missing values*, fossem estes vazios ou '#N/A', pois ambas as representações não possuem significado.

Para além desta alteração, foram ainda convertidos todos os caracteres ';' em ',', com a finalidade de transformar o *dataset* num ficheiro .csv e facilitar a sua posterior análise e tratamento.

3 Tratamento dos dados

3.1 Remoção de linhas com valores em falta

Um dos passos fundamentais no pré-processamento e tratamento dos dados é lidar com os valores em falta nos *datasets*. Neste trabalho, a opção recaiu sobre a eliminação de todas as linhas com pelo menos um valor em falta, porque torna todo o tratamento mais fácil e simples. Porém, quando se toma este tipo de decisão, é necessário avaliar quão significativa é a perda de informação. Neste caso, verificou-se que a situação mais crítica acontece no *dataset* de Campos de Jordão, com uma percentagem de linhas sem valores nulos pouco superior a um quarto dos dados. Na tabela abaixo constam os resultados para os três *datasets*:

Campos de Jordão	Cruzeiro do Sul	Picos
28.34%	70.85%	89.30%

Tabela 1: Percentagem de linhas do *dataset* sem valores em falta.

Na procura de otimizar estes valores, nomeadamente para Campos de Jordão, pensou-se em remover apenas as linhas com valores em falta nos atributos relevantes para a previsão meteorológica na dada cidade. Para se perceber a relevância dos atributos, era necessário estabelecer uma correlação entre os atributos da previsão - temperaturas mínima e máxima - e os restantes atributos. Como convinha evitar valores nulos nestes cálculos, usaram-se os três *datasets* criados com a filtragem anterior, cujos resultados foram expressos na tabela.

3.1.1 Cálculo da correlação entre atributos

Para o cálculo da correlação, aos conjuntos de dados filtrados aplicou-se a seguinte fórmula:

$$C_{x,y} = \frac{E[X \times Y] - E[X] \times E[Y]}{\sigma_x \times \sigma_y}$$

Aqui, as variáveis são:

- $E[X]$, média dos valores de X;
- $E[Y]$, média dos valores de Y;
- $E[X, Y]$, média da multiplicação de X por Y, ou seja, média após primeiro multiplicar os valores de X com os valores de Y;
- σ_X, desvio padrão de X;
- σ_Y, desvio padrão de Y.

Com estes cálculos, criou-se uma matriz de correlações, com os atributos do *output* da previsão, isto é, as temperaturas mínima e máxima, nas linhas e nas colunas todos os atributos do *dataset*.

3.2 Seleção de atributos

A interpretação da matriz de correlações, no sentido de selecionar os atributos com maior importância para o *output*, teve necessariamente de implicar o balanço entre os valores de correlação para as temperaturas mínima e máxima, uma vez que, para alguns atributos, o sinal do valor - positivo ou negativo - diferia para as diferentes temperaturas, sendo mais difícil perceber se o atributo era, ou não, relevante para a previsão meteorológica.

Assim, a estratégia usada consistiu em, na presença de um valor de correlação negativo para uma das temperaturas, comparar o módulo deste valor com o valor de correlação para a outra temperatura, considerando o atributo relevante para o *output* apenas se esse segundo valor de correlação fosse superior ao módulo. Naturalmente, o atributo era desde logo considerado relevante nas situações em que as correlações assumiam valores positivos para ambas as temperaturas. Para uma melhor perceção, abaixo está o pseudo-código da função de balanceamento:

$\forall a \in atributos$,
    Se $C_{tmin,a} < 0 \vee C_{tmax,a} < 0$:
        Se $C_{tmin,a} < 0$:
            Se $C_{tmax,a} > \|C_{tmin,a}\|$:
                $a \in atributosRelevantes$
        Senão:
        Se $C_{tmin,a} > \|C_{tmax,a}\|$:
            $a \in atributosRelevantes$
    Senão:
    $a \in atributosRelevantes$

Feito o balanceamento, os atributos selecionados nos vários *datasets* foram os seguintes:

	Prec	Insol	Evap	AvgT	AvgH	WindS	SolarN
Campos de Jordão	X		X	X			X
Cruzeiro do Sul		X	X	X			X
Picos		X	X	X		X	X

Tabela 2: Atributos relevantes para a previsão meteorológica, nos diferentes *datasets*.

Esta informação, por sua vez, permitiu filtrar as linhas com valores em falta dos *datasets* de uma forma mais adaptada ao problema, que procura tirar o máximo partido dos dados e evitar o desperdício de informação, que resultou numa nova tabela de percentagens das linhas filtradas:

Campos de Jordão	Cruzeiro do Sul	Picos
34.70%	70.85%	89.57%

Tabela 3: Percentagem de linhas do *dataset* sem valores em falta, após seleção de atributos.

3.3 Introdução de atributos em falta

Após uma primeira fase de tratamento dos dados, prosseguiu-se com a introdução de um novo atributo/campo: 'NoonAngle'. Este trata-se de uma variação do atributo 'SolarNoonAngle', existente nos *datasets* das três cidades. A sua adição foi aconselhada pela equipa docente mentora do projeto, sendo obtido através da seguinte fórmula:

$$\alpha_{noon} = 90 - \left| -23.44 \cos\left(\left(\frac{2\pi}{365.25}\right)(D+8.5)\right) - lat \right|$$

Figura 1: Fórmula para obtenção de 'NoonAngle' através do uso de 'SolarNoonAngle'.

Nesta é possível verificar que existem duas variáveis independentes: o atributo 'SolarNoonAngle', representado pela letra 'D', e a latitude, valor fornecido no enunciado do projeto, que corresponde à latitude da cidade à qual pertencem os dados.

3.4 Criação do *dataset* a inserir na Rede Neuronal Artificial

Estando, finalmente, na posse de três conjuntos de dados totalmente válidos, passou-se ao passo seguinte rumo à previsão meteorológica: a junção de atributos de 4 dias seguidos. Naturalmente, as previsões meteorológicas para um dado dia não passam de estimativas, que têm por base os registos climatéricos dos dias imediatamente anteriores. Assim, para que a RNA pudesse ser, primeiro, treinada e depois capaz de realizar previsões de temperaturas, era necessário que tivesse os dados de dias anteriores. A junção, na mesma linha, desses dados, constituiu uma forma de simplificar a organização e análise dos dados.

Estudos sobre previsão meteorológica mostram que apenas os três dias anteriores são significativos para prever a meteorologia de um certo dia, isto é, exemplificando, sabendo os dados meteorológicos, de uma certa região, nos dias de hoje, ontem e anteontem, é possível prever, com um grau de certeza acentuado, os valores para determinadas condições atmosféricas que irão ocorrer amanhã. Assim se justifica a junção dos dados em blocos de 4 dias consecutivos, em que o quarto representa o dia a prever.

Como, para a previsão, só eram relevantes as temperaturas mínima e máxima, decidiu-se retirar todos os restantes atributos relativos ao dia a prever. Além disso, como a junção em blocos de 4 dias consecutivos tinha já sido executada, os campos relativos à data foram totalmente removidos. De seguida, apresenta-se um excerto genérico do aspeto do cabeçalho dos *datasets* finais, gerados após o agrupamento por dias:

```
"AvgTemp_1","AvgTemp_2",...,"MinTemp","MinTemp_1",...,"MaxTemp","MaxTemp_1",....
```

Uma vez que foi, previamente, feita a seleção dos atributos mais significativos para o que se pretende prever, 'MaxTemp' e 'MinTemp', nesta fase os *datasets* apresentavam diferenças, contendo apenas os atributos mencionados como mais significativos para a cidade em questão.

3.5 Normalização dos dados

Após todo este processo, o *dataset* encontrava-se praticamente pronto a ser aplicado na RNA. Restava normalizar os dados, processo essencial para aumentar a integridade dos dados, bem como

melhorar o desempenho da rede, aumentando a sua capacidade de aprendizagem e consequentemente a sua capacidade de previsão. A ausência de normalização pode, dependendo do *dataset*, levar a resultados inúteis ou processos de treino demorados e difíceis.

Nesta situação, o processo de normalização foi efetuado com apenas três linhas de código R, responsáveis por converter os dados para a escala [0,1], usando um método que faz uso dos valores mínimos e máximos de todos os atributos do *dataset*:

```
max <- apply(df,2,max)
min <- apply(df,2,min)
df <- as.data.frame(scale(df, center = min, scale = max-min))
```

4 Criação e Treino da Rede Neuronal Artificial

4.1 Criação de testes para medidas de desempenho

A primeira decisão, associada a uma rede neuronal, passa por definir que percentagem do *dataset* será para treino da rede, ficando o restante para a testar. Aqui, definiu-se que 70% dos dados seriam para treino e os restante 30% para teste. Outra variável importante é a fórmula introduzida na rede, que define os atributos do *input* e *output*. Nesta situação, a fórmula pode diferir consoante os atributos relevantes do conjunto de dados, porém, a fórmula genérica é:

$$MinTemp + MaxTemp \sim atributo_1\_1 + atributo_1\_2 + atributo_1\_3 + atributo_2\_1 + ...$$

À esquerda do $\sim$ encontra-se o *output*. De entre os atributos, certas são as presenças da temperatura mínima ('MinTemp') e temperatura máxima ('MaxTemp'), sendo a lista completa pelos atributos que passaram na seleção.

Definidos os dados e a fórmula a aplicar na RNA, o que se seguiu foi uma série de experiências com variáveis da função usada para treino da rede (neuralnet, do *package R* com o mesmo nome), na procura de proporcionar o melhor treino à rede, de modo a otimizar os resultados da sua previsão. Testaram-se alterações do valor predefinido em variáveis como o *learning rate*, o *threshold*, o algoritmo de aprendizagem utilizado pela rede e o número de camadas e nodos internos, estando o procedimento e resultados dessas experiências exibidos na próxima secção.

4.2 Treino e comparação de resultados

Para possibilitar a comparação de resultados, calculou-se o RMSE - *Root Mean Square Error* associado à previsão da rede, em função do valor esperado. Já as experiências, essas começaram com uma rede neuronal em que não se alterou nenhum valor *default*, à exceção dos dados e da fórmula.

	Campos de Jordão	Cruzeiro do Sul	Picos
T. Mínima	12.24%	6.62%	10.96%
T. Máxima	14.03%	10.63%	10.75%

Tabela 4: RMSE's associados à primeira experiência.

Naturalmente, procurou-se a otimização destes resultados. O primeiro passo esteve na alteração do número de camadas e nodos internos. Inicialmente, procurou-se aumentar apenas o número de nodos, mantendo uma única camada interna. Sabendo que o número de nodos de *input* era elevado, estabeleceu-se que teria 10 nodos essa única camada interna. Após uma primeira tentativa de correr o teste, percebeu-se que o *stepmax* predefinido pela função não era suficiente para os dados da cidade de Picos, pelo que se aumentou para o dobro (200 mil), porém, ainda assim a rede não conseguia convergir, culminando no aumento para 300 mil.

	Campos de Jordão	Cruzeiro do Sul	Picos
T. Mínima	11.83%	6.14%	10.30%
T. Máxima	13.35%	9.73%	10.87%

Tabela 5: RMSE's associados à segunda experiência.

Registaram-se melhorias nos resultados. Numa segunda tentativa, aumentou-se o número de *hidden layers* para 2, ambas com 10 nodos: c(10,10).

	Campos de Jordão	Cruzeiro do Sul	Picos
T. Mínima	13.79%	6.44%	10.33%
T. Máxima	15.88%	9.85%	10.65%

Tabela 6: RMSE's associados à terceira experiência.

Desta feita, os resultados mostraram-se, regra geral, menos positivos, pelo que se retrocedeu para uma camada apenas e testou-se a alteração de outro parâmetro: o *threshold*. Com este parâmetro a 0.07 (em vez de 0.01) e uma camada com 10 nodos internos, estes foram os resultados:

	Campos de Jordão	Cruzeiro do Sul	Picos
T. Mínima	12.48%	6.10%	9.50%
T. Máxima	12.52%	9.38%	9.53%

Tabela 7: RMSE's associados à quarta experiência.

Com esta experiência sim, registaram-se melhorias nos resultados, o que levou a que se experimentasse, com o mesmo *threshold*, o aumento do número de camadas internas para duas, ambas com 10 nodos. Esta situação mostrou-se mais delicada de analisar, uma vez que alguns valores melhoraram e outros pioraram, mas qualquer das alterações não foi muito substancial. Ainda assim, como para o *dataset* de Campos de Jordão houve alguma melhoria mais significativa, consideraram-se estes resultados positivos.

	Campos de Jordão	Cruzeiro do Sul	Picos
T. Mínima	11.35%	6.30%	9.75%
T. Máxima	12.32%	9.22%	9.55%

Tabela 8: RMSE's associados à quinta experiência.

Após esta experiência, foram testadas outras mudanças: aumento do *threshold*, aumento do número de camadas internas, alteração do algoritmo de aprendizagem para "backprop"e "rprop-"e, até, alterações no *learning rate*, todavia, nenhuma das alterações melhorou os resultados.

5 Conclusão

Após uma extensa análise e tratamento dos *datasets*, chegou-se a três conjuntos de dados passíveis de serem alvo de treino e teste por parte da rede neuronal. Levados a cabo alguns testes, o treino da rede gerou previsões mais assertivas perante os seguintes valores da função neuralnet:

- algoritmo de aprendizagem: rprop+
- *threshold*: 0.07

- *learning rate*: NULL
- *hidden layers*: c(10,10)
- *linear.output*: TRUE

Assim, dado que os resultados, considerando o domínio do problema, ou seja, a previsão meteorológica, se mostraram bastante assertivos, abaixo, na tabela, estão os RMSE's associados à previsão da rede para as temperaturas mínima e máxima, para as diferentes cidades. Na imagem seguinte, encontram-se os gráficos alusivos aos desvios das previsões da rede, para os vários *datasets* e temperaturas.

	Campos de Jordão	Cruzeiro do Sul	Picos
T. Mínima	11.35%	6.30%	9.75%
T. Máxima	12.32%	9.22%	9.55%

Tabela 9: RMSE's associados à previsão da rede.

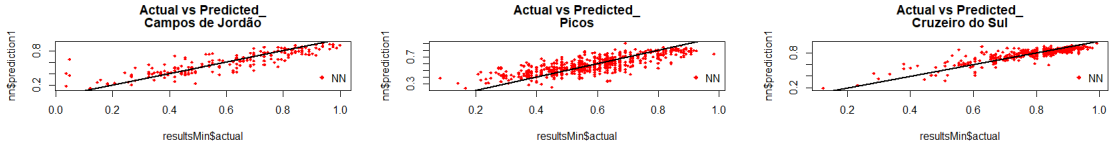

Figura 2: Desvios das previsões dos vários *datasets* para a temperatura mínima.

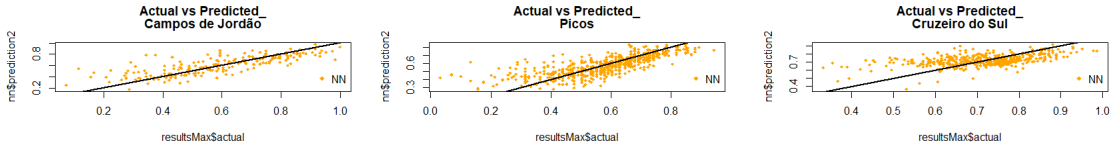

Figura 3: Desvios das previsões dos vários *datasets* para a temperatura máxima.