

Universidade do Minho - Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Sistemas Autónomos

Trabalho prático

Redes Neuronais Artificiais

LSTM

Carlos José Gomes Campos a74745

José Pedro Ferreira Oliveira a78806

Vitor José Ribeiro Castro a77870

27 de Maio de 2019

1 Tratamento de dados

Após a análise do *dataset* verificou-se que seria necessário tratar e eliminar algum do seu conteúdo que se verificou irrelevante para a construção do modelo proposto. Posto isto, foi feita uma separação do mês e do ano em cada registo, associando a cada um, o respetivo valor numérico categórico. No final obteve-se um total de quatro *features* correspondentes ao mês, publicidade, vendas e ano.

2 Carregamento dos dados

Ao nível do carregamento dos dados, foi necessário dividir os dados em duas partes específicas, tal como era pedido no enunciado, os dados dos dois primeiros anos para treino e os referentes ao terceiro ano para teste. Após alguns testes verificamos que o valor do tamanho da janela ideal para este problema seria de 2, o que corresponde ao número de *time steps* de cada sequência. Posteriormente, foram separadas as *features* e as *labels* com recurso a operações sobre matrizes.

3 Estrutura da rede neuronal

Após vários testes foi possível analisar a evolução dos valores RMSE, mediante várias estruturas. As redes com muitos nodos em cada camada, apresentaram valores de RMSE nos dados de teste elevados, comparativamente ao obtido nos dados de treino o que indicou a existência de *overfitting* do modelo. Por outro lado, se a quantidade de nodos for muito baixa, o valor de RMSE nos dados de treino é superior ao dos dados de teste o que indica *underfitting* do modelo. A estrutura que obteve melhores resultados foi três camadas com 256, 128 e 64 nodos LSTM, com um *dropout* de 0.2 entre elas à excepção da última. De seguida, uma camada *Dense*, ou seja, totalmente conectada, com 16 nodos e uma função de ativação *relu*. Na camada final, utiliza-se a função de ativação *linear*, uma vez que é um problema de regressão. Finalmente, define-se o *adam* como otimizador do algoritmo de *backpropagation* e a função de *loss* é MSE. É importante salientar que a adição de dropout contribuiu para que os valores de RMSE nos dados de treino e teste diminuíssem e se aproximassem, o que indica um melhoramento na qualidade do modelo desenvolvido. Na figura 1 está uma representação da estrutura da rede.

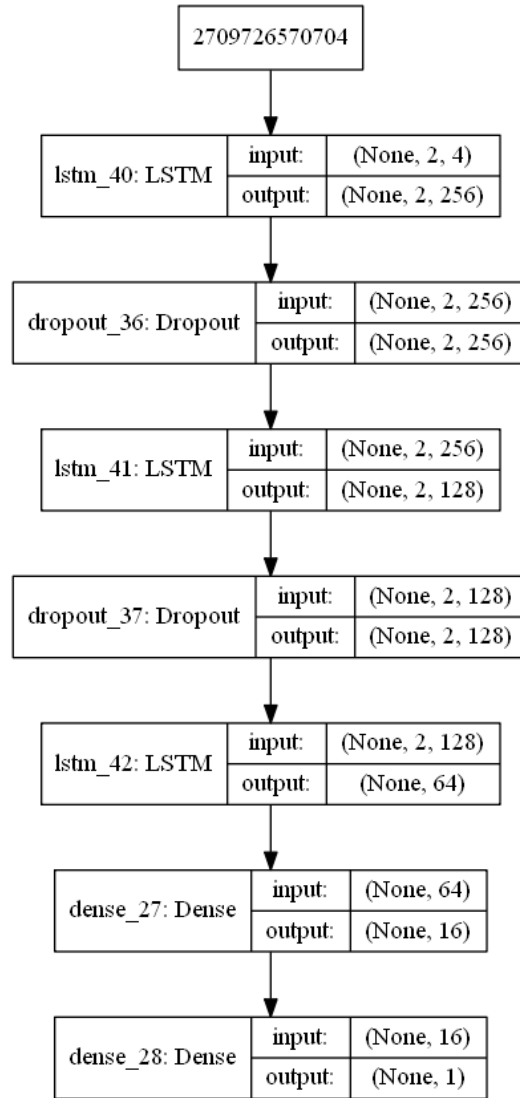


Figura 1: Estrutura do modelo LSTM

4 Construção do modelo

Antes de treinar e avaliar o modelo, foi feito um processamento nos dados que consiste em dividir por 100 todos os valores do *dataset*, deste modo, pretende-se diminuir a carga computacional no modelo, uma vez que se está a fazer cálculos com valores de grandeza inferior. Com isto, torna-se também possível analisar melhor, graficamente, a evolução de todas as métricas utilizadas nesta escala

mais pequena. O treino do modelo é efetuado com 350 *epochs*, sendo este valor definido após vários testes, que permitiram analisar a convergência do modelo mediante a evolução da função de *loss*. O valor de *batch_size* é o pré-definido e não foram utilizados dados para validação, uma vez que o *dataset* é pequeno, o grupo decidiu utilizar todos os dados no treino do modelo, melhorando os valores de RMSE.

Relativamente a dificuldades encontradas, uma delas foi interpretar se uma determinada alteração nos hiperparâmetros melhorava o modelo ou não, pois o fator aleatoriedade dos pesos iniciais de cada rede, interfere na convergência e no desempenho da rede, e, portanto, em alguns casos os valores iniciais eram bastante maus e a rede tinha dificuldade em aprender. Para além disso, face às otimizações testadas o valor de RMSE não exibia grandes variações o que tornou difícil detetar melhorias em determinadas situações.

5 Análise de resultados

Os valores de RMSE obtidos são relativamente baixos devido à redução da grandeza dos dados referidos anteriormente, no entanto, facilmente são obtidos os valores referentes aos dados originais revertendo o processo.

Para além de tentar obter o menor valor de RMSE possível, o grupo procurou também minimizar a diferença entre o valor de RMSE dos dados de treino e dos casos de teste.

O melhor modelo exibiu um RMSE de 0.14 nos dados de treino e 0.20 nos dados de teste.

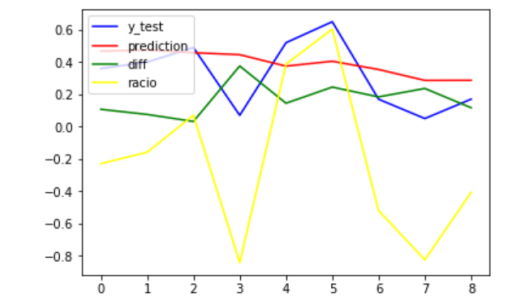


Figura 2: Gráfico das várias métricas tidas em conta

```

Train Score: 0.02 MSE (0.14 RMSE)
Test Score: 0.04 MSE (0.20 RMSE)
['loss', 'acc']
valor: 0.360000 ---> Previsão: 0.467225 Diff: 0.107225 Racio: -0.229493
valor: 0.400000 ---> Previsão: 0.475432 Diff: 0.075432 Racio: -0.158660
valor: 0.490000 ---> Previsão: 0.457856 Diff: 0.032144 Racio: 0.070206
valor: 0.070000 ---> Previsão: 0.445770 Diff: 0.375770 Racio: -0.842969
valor: 0.520000 ---> Previsão: 0.374884 Diff: 0.145116 Racio: 0.387095
valor: 0.650000 ---> Previsão: 0.405129 Diff: 0.244871 Racio: 0.604427
valor: 0.170000 ---> Previsão: 0.354133 Diff: 0.184133 Racio: -0.519954
valor: 0.050000 ---> Previsão: 0.286449 Diff: 0.236449 Racio: -0.825449
valor: 0.170000 ---> Previsão: 0.287064 Diff: 0.117064 Racio: -0.407797

```

Figura 3: Valores de RMSE do modelo