Universidade do Minho
Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Previsão de Vendas Utilizando Deep Learning

Trabalho Prático

Sistemas Autónomos

Nome: Ana Esmeralda Fernandes — Número: A74321
Nome: Thanira Rates — Número: PG35406
Nome: Rui Miranda — Número: A75488

Implementação e Escolhas Tomadas

O objetivo deste trabalho prático é o desenvolvimento de um modelo de Deep Learning para previsão de vendas. Podemos entender este problema como sendo de regressão, onde queremos prever as vendas do mês seguinte com dados de meses anteriores. O dataset fornecido contém o registo de vendas e custos de publicidade de um produto dietético durante o período de tempo de 36 meses (3 anos). Foi também definido que os dois primeiros anos seriam utilizados para treino, sendo o último usado para testes.

Sendo o dataset uma série temporal, onde a ordem das instâncias importa, foi implementado então um modelo baseado em Long-Short Term Machines (LSTM), com recurso ao Keras e Tensorflow.

LSTMs são sensíveis à escala dos dados de entrada, especificamente quando as funções de ativação são usadas. Sendo assim, os dados foram normalizados com a classe de pré-processamento *MinMaxScaler* da biblioteca *scikit-learn.* A variável da data foi transformada para informação relativa ao mês referente. A normalização dos dados é efetuada entre 0 e 1:

	Month=1	Month=2	Month=3	Month=4	Month=5	Month=6	Month=7	Month=8	Month=9	Month=10	Month=11	Month=12	Advertising	Sales
1	1	0	0	0	0	0	0	0	0	0	0	0	12	15
2	0	1	0	0	0	0	0	0	0	0	0	0	20.5	16
3	0	0	1	0	0	0	0	0	0	0	0	0	21	18
4	0	0	0	1	0	0	0	0	0	0	0	0	15.5	27
5	0	0	0	0	1	0	0	0	0	0	0	0	15.3	21
6	0	0	0	0	0	1	0	0	0	0	0	0	23.5	49
7	0	0	0	0	0	0	1	0	0	0	0	0	24.5	21
8	0	0	0	0	0	0	0	1	0	0	0	0	21.3	22
9	0	0	0	0	0	0	0	0	1	0	0	0	23.5	28
10	0	0	0	0	0	0	0	0	0	1	0	0	28	36
11	0	0	0	0	0	0	0	0	0	0	1	0	24	40
12	0	0	0	0	0	0	0	0	0	0	0	1	15.5	3
13	1	0	0	0	0	0	0	0	0	0	0	0	17.3	21
14	0	1	0	0	0	0	0	0	0	0	0	0	25.3	29
15	0	0	1	0	0	0	0	0	0	0	0	0	25	62
16	0	0	0	1	0	0	0	0	0	0	0	0	36.5	65
17	0	0	0	0	1	0	0	0	0	0	0	0	36.5	46
18	0	0	0	0	0	1	0	0	0	0	0	0	29.6	44
19	0	0	0	0	0	0	1	0	0	0	0	0	30.5	33
20	0	0	0	0	0	0	0	1	0	0	0	0	28	62
21	0	0	0	0	0	0	0	0	1	0	0	0	26	22
22	0	0	0	0	0	0	0	0	0	1	0	0	21.5	12
23	0	0	0	0	0	0	0	0	0	0	1	0	19.7	24
24	0	0	0	0	0	0	0	0	0	0	0	1	19	3
25	1	0	0	0	0	0	0	0	0	0	0	0	16	5
26	0	1	0	0	0	0	0	0	0	0	0	0	20.7	14
27	0	0	1	0	0	0	0	0	0	0	0	0	26.5	36
28	0	0	0	1	0	0	0	0	0	0	0	0	30.6	40
29	0	0	0	0	1	0	0	0	0	0	0	0	32.3	49

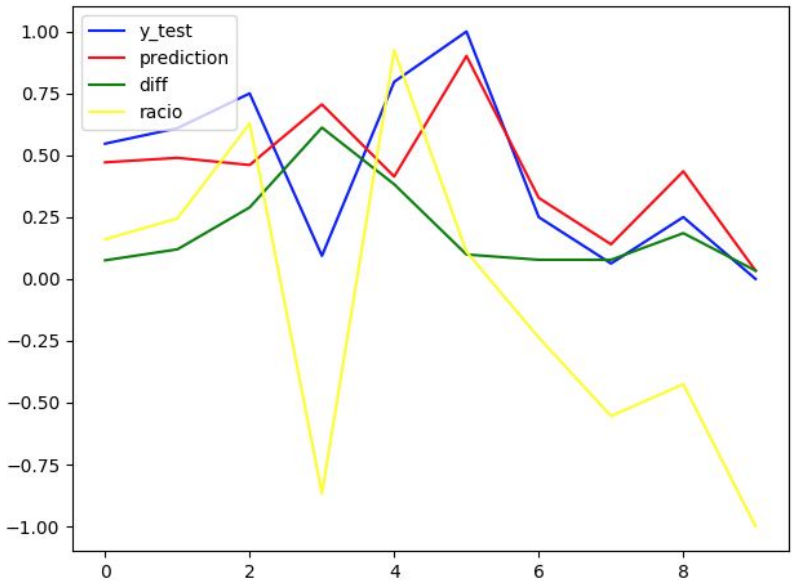
A janela deste modelo é de 2, ou seja, por exemplo, dados os custos de publicidade e vendas dos meses de janeiro e fevereiro, o modelo implementado vai prever as vendas do mês de março. Comparando com várias janelas, esta foi a que apresentava os melhores resultados.

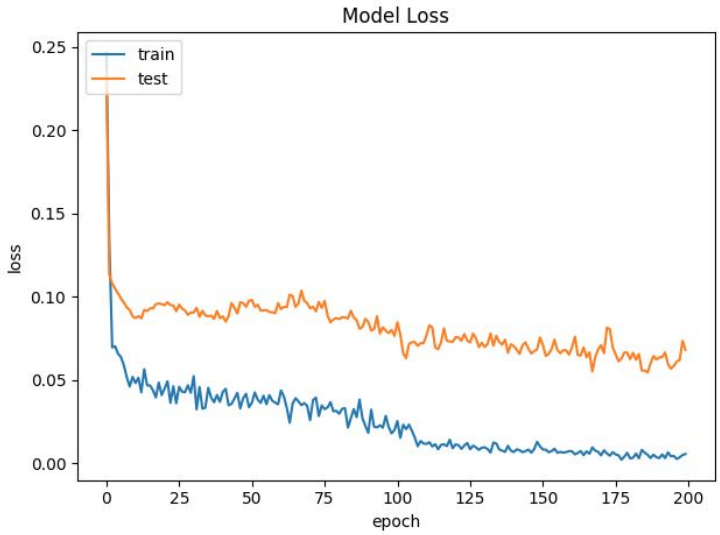
```python
def build_model3(janela):
    model = Sequential()
    model.add(LSTM(30, input_shape=(janela, 14), return_sequences=True))
    model.add(Dropout(0.1))
    model.add(LSTM(20, input_shape=(janela, 14), return_sequences=True))
    model.add(Dropout(0.1))
    model.add(LSTM(10, input_shape=(janela, 14), return_sequences=False))
    model.add(Dropout(0.1))
    model.add(Dense(16, activation="relu", kernel_initializer="normal"))
    model.add(Dense(1, activation="linear", kernel_initializer="normal"))
    model.compile(loss='mse', optimizer='nadam', metrics=['mse', 'accuracy'])
    return model
```

O modelo utilizado apresenta 5 camadas, 3 LSTMs e duas Dense. Entre estas existem também camadas de Dropout para a redução de *overfitting.* Nas camadas Dense foram utilizadas a função de ativação *relu* na camada intermédia e na camada de saída a função de ativação *linear* pois durante as fases de teste foram as que trouxeram os resultados com o RMSE mais baixo como pretendido. Pelo mesmo motivo o *optimizer* escolhido foi o Nesterov Adam optimizer, que é funciona de forma parecida ao *optimizer* Adam mas utiliza o momentum de Nesterov.

Desempenho do Modelo

De seguida apresentam-se os gráficos criados pelo Keras, nomeadamente o gráfico relativo à previsão com os casos de testes, e o gráfico de *loss:*





Através da análise dos gráficos podemos verificar que durante o treino o modelo comporta-se bem em termos de *loss* mas que durante o teste este apesar de continuar baixo aumenta comparativamente aos resultados de treino.

```
 - 0s - loss: 0.0052 - mean_squared_error: 0.0052 - acc: 0.0417 - val_loss: 0.06
dict_keys(['val_loss', 'val_mean_squared_error', 'val_acc', 'loss', 'mean_square
['loss', 'mean_squared_error', 'acc']
valor: 0.546875 ---> Previsão: 0.387720 Diff: 0.159155 Racio: 0.410491
valor: 0.609375 ---> Previsão: 0.555327 Diff: 0.054048 Racio: 0.097327
valor: 0.750000 ---> Previsão: 0.476537 Diff: 0.273463 Racio: 0.573856
valor: 0.093750 ---> Previsão: 0.640535 Diff: 0.546785 Racio: -0.853638
valor: 0.796875 ---> Previsão: 0.403279 Diff: 0.393596 Racio: 0.975990
valor: 1.000000 ---> Previsão: 0.842423 Diff: 0.157577 Racio: 0.187052
valor: 0.250000 ---> Previsão: 0.284503 Diff: 0.034503 Racio: -0.121274
valor: 0.062500 ---> Previsão: 0.140387 Diff: 0.077887 Racio: -0.554803
valor: 0.250000 ---> Previsão: 0.481411 Diff: 0.231411 Racio: -0.480694
valor: 0.000000 ---> Previsão: 0.022768 Diff: 0.022768 Racio: -1.000000
Train Score: 0.00 MSE (0.06 RMSE)
Test Score: 0.06 MSE (0.25 RMSE)
PyDev console: starting.
```

Como se pode verificar na imagem acima, o valor de RMSE obtido em treino foi 0.06 (0.00 MSE) e em teste foi 0.25 (0.06 MSE). Na imagem pode-se verificar algumas previsões de vendas com os respectivos valores.

Infelizmente, não fomos capazes de reduzir o *overfitting* existente quando não se utiliza previamente a normalização dos valores, tendo experimentado com várias configurações de modelos. A normalização ajudou a reduzir o RMSE, apresentando um comportamento ligeiramente melhor que o desnormalizado. Visto que Deep Learning é essencialmente utilizado quando existe uma grande quantidade de dados que se possa utilizar, e dado o reduzido número de casos, apenas dois anos de informação, apesar de terem sido feitos vários testes com várias configurações de redes não se conseguiu melhorar os resultados mais que os indicados.