



Universidade do Minho

Mestrado Integrado em Engenharia Informática

Sistemas Autónomos

LSTM para a previsão de vendas

Luís Gomes (A78701)

Joel Rodrigues (A79068)

Elisa Valente (A79093)

26 de Maio de 2019

O presente documento destina-se à apresentação da abordagem utilizada na resolução do terceiro trabalho prático da unidade curricular de Sistemas Autónomos. O projeto proposto tem como objetivo a previsão de vendas de um produto dietético de controlo de peso, utilizando *deep learning*.

1 Análise do Conjunto de Dados

O conjunto de dados fornecido contém o registo de vendas e custos de publicidade de um produto dietético de controlo de peso durante 36 meses consecutivos. É constituído por 3 colunas e 36 entradas, sendo que as colunas correspondem, ordenadamente, ao mês, ao custo de publicidade e ao número de vendas. O custo varia entre 12 e 36.5 e o valor das vendas entre 1 e 65. Pretende-se, com base nestes dados, criar um modelo LSTM que seja capaz de prever as vendas de acordo com o investimento em publicidade e o mês em questão. Para isto serão utilizados 24 meses de treino e 12 para teste.

2 Pré-Processamento dos Dados

Os dados fornecidos apresentavam dados errados, como colunas sem informação e linhas com dados sem relevância para o problema em questão. Assim, estes dados foram retirados do *dataset* de forma a obter uma fonte de informação limpa para a construção do modelo.

Para além deste processo, e devido à curta extensão de atributos presentes, foram adicionadas novas informações que podem ser inferidas a partir dos dados existentes. Este tipo de operações tem como objetivo aumentar a quantidade de dados disponíveis para o treino e teste do modelo permitindo assim melhorar a sua precisão. A título de exemplo, podemos inferir a partir do atributo *Month* (Mês) os seguintes dados:

- Estações do ano

- Quantidade de dias (28, 29, 30, 31)
- Ano de vendas (1,2,3)

Este tipo de informações são codificadas de forma a representar informação de igual forma, como por exemplo, na representação do mês *arrays* binários de comprimento 12 em que apenas um dos *bits* se encontra a 1 representando o respetivo mês.

3 Construção do Modelo

O conjunto de dados fornecido apresenta uma série temporal, isto é, as entradas do *dataset* estão apresentadas cronologicamente, apresentando por isso uma correlação temporal entre dados consecutivos. Assim, uma troca na ordem das entradas do conjunto de dados significa uma destruição de informação. Os modelos mais indicados para este tipo de dados são as arquiteturas LSTM (Long Short-Term Memory), visto que apresentam capacidades de memória, permitindo manter um valor durante um curto ou longo prazo possibilitando à rede efetuar relações com dados passados.

Dado que o principal objetivo do problema é obtenção do melhor valor de desempenho da rede possível, foi necessário usar diferentes estratégias para alcançar valores cada vez melhores. Nesta secção são apresentadas as estratégias usadas durante a resolução do problema.

3.1 Estratégias Adotadas

Numa primeira instância, foi utilizada uma rede muito simples a partir da qual foram efetuadas alterações para tentar aproximar as previsões dos dados reais. Nesta fase, o modelo não conseguia aprender os padrões dos dados, e como tal, foram adotadas estratégias para colmatar o *underfitting* adjacente a este modelo. Neste sentido, foi aumentada a complexidade da rede, incrementando o número de camadas LSTM e Dense bem como o número de nodos de cada camada.

Com o aumento da complexidade das redes testadas, começaram a surgir modelos que apresentavam *overfitting*, ou seja, o RMSE dos dados de treino era melhor que nos de teste. Este tipo de característica é exibida por modelos com baixa capacidade de generalização. Para colmatar esta dificuldade optaram-se por diferentes estratégias:

- **Adição de mecanismos de regularização** - neste caso, adicionaram-se camadas **Dropout** ao modelo, fazendo com que vários nodos da rede sejam removidos ao longo do processo de treino, diminuindo a probabilidade dos nodos se adaptarem demasiado uns aos outros. Este processo introduz ruído nos dados permitindo generalizar os casos de treino.
- **Alteração de funções de ativação** - uma outra forma foi a variação das funções de ativação das diferentes camadas da rede. Neste caso foram testadas diferentes funções e escolhidas aquelas com melhores resultados.
- **Introdução de novos dados** - de forma a generalizar o modelo, prevenindo este problema, foram adicionadas informações no *dataset*, que apesar de serem teoricamente redundantes, para o modelo terão utilidade. Para esse efeito, como referido na secção 2, adicionaram-se várias colunas ao *dataset*: o número de dias do mês, o mês em questão, o ano (os dados são relativos a 3 anos) e ainda a informação das estações do ano.
- **Camada Bidirecional**

Uma estratégia que se decidiu adotar foi o uso de camadas **LSTM Bidirecionais** que, ao contrário de uma cada unidirecional, que só tem em conta a informação passada, este tipo de camadas usam mecanismos de duplas ligações à mesma saída, permitindo que a camada de saída obtenha informação dos estados passados e futuros simultaneamente.

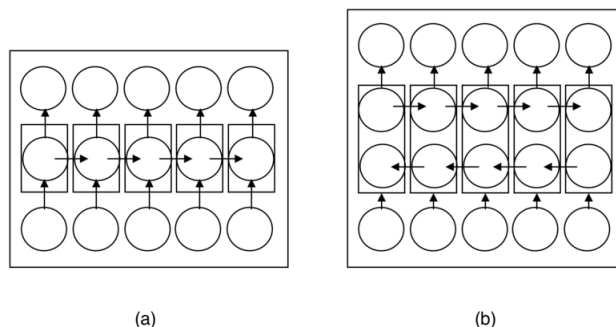


Figura 1: (a) Estrutura RNA unidirecional; (b) Estrutura RNA Bidirecional.

Depois de alguns testes, verificou-se que este mecanismo melhorou significativamente o desempenho do modelo.

• Restrições de Peso

As restrições de peso são uma abordagem utilizada para reduzir o *overfitting* de modelos de aprendizagem *Deep Learning* nos dados de treino e, assim, melhorar o desempenho do modelo em novos dados.

Ao contrário de outros tipos de camadas, as redes neurais recorrentes permitem que se defina uma restrição nos pesos de entrada, no *bias* e nos pesos de entrada recorrentes. Como tal, adotaram-se essas restrições para tentar melhorar a previsão do modelo. Abaixo segue o excerto de código de uma camada LSTM com restrições de peso.

```
LSTM(64, activation='relu', return_sequences=True, input_shape=(janela, 19),
↪ kernel_constraint=max_norm(3), recurrent_constraint=max_norm(3),
↪ bias_constraint=max_norm(3))
```

• *Shuffle* da Matriz de Dados

O *Shuffle* da ordem de *input* das janelas de treino é mais uma medida para uma melhor generalização do modelo. Assim, tendo cada janela a informação temporal

pretendida, os dados são passados ao modelo em diferente ordem, melhorando a capacidade de aprendizagem da rede.

• Algoritmos Genéticos

Para além de todas as alterações efetuadas ao modelo, como os resultados obtidos ainda não eram satisfatórios, decidiu-se implementar um algoritmo genético objetivando-se obter um modelo com melhor resultados inalcançáveis por força bruta. Para este algoritmo foi tido em conta o número de nodos por camada, a percentagem de *dropout* de cada camada, o número de *epochs* e o *batch_size*. Apesar de se executar o algoritmo para uma grande variação de parâmetros os resultados obtidos não foram tão diferentes dos alcançados anteriormente. Ainda assim, houve populações em que o melhor indivíduo apresentou resultados melhores (10 RMSE), no entanto, apresentaram-se muito instáveis quando submetidas a novos testes.

4 Resultados Obtidos

O modelo ótimo obtido foi conseguido com as camadas apresentadas na figura 2 e com os seguintes dados adicionais: estações do ano e os meses, tudo em variáveis categóricas. Para além disso, os valores das variáveis *advertising* e *sales* foram recalculadas de modo a diminuir a sua escala, estando agora 10 vezes mais pequenas. A figura 2 apresenta a rede criada.

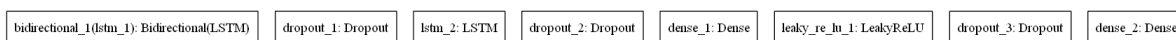


Figura 2: Modelo final.

Os resultados obtidos são apresentados na figura 3 com um *train score* de 1.4 MSE e um *test score* de 1.33 RMSE.

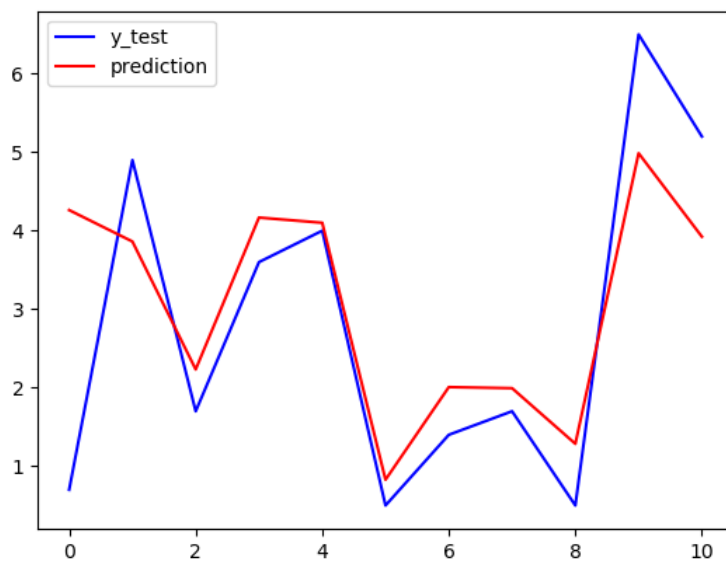


Figura 3: Resultados da previsão do modelo final.

5 Conclusão

Tendo em conta a quantidade dos dados fornecidos foram efetuadas um conjunto de técnicas que nos permitiram aumentá-los de forma a melhorar o desempenho do modelo criado. Para melhorar o modelo criado foram adotadas um conjunto de estratégias onde se pretendia melhorar o seu *underfitting* ou *overfitting*. Estas medidas mostraram-se satisfatórias na medida em que o modelos criados foram melhorando progressivamente com as alterações efetuadas.

Os resultados obtidos foram bastante satisfatórios, permitindo prever resultados com um erro médio relativamente baixo em relação à escala dos dados em questão.