Regressão Linear - New York Stock Exchange

João Brito, M9984
Department of Computer Science
University of Beira Interior
Covilhã, Portugal
joao.pedro.brito@ubi.pt

***Abstract*—O presente documento tem como objetivo resolver um problema de Regressão Linear com recurso a 3 abordagens tecnicamente diferentes, mas com o mesmo propósito: encontrar, para o modelo escolhido, a configuração ideal que garante os melhores resultados em observações nunca vistas.**

I. Introdução

No âmbito deste trabalho, foi escolhido um *dataset* com informação sobre a prestação de algumas empresas no Bolsa de Valores de Nova Iorque (NYSE [1]). Os dados foram obtidos através da plataforma online Kaggle [2]. As secções seguintes abordarão o processamento ao qual os dados foram sujeitos, as experiências realizadas e algumas conclusões derivadas do trabalho realizado.

II. Processamento dos dados

A. Descrição geral

O ficheiro que serviu de base ao desenvolvimento de código relevante foi o "prices-split-adjusted.csv". Este ficheiro de valores separados por vírgulas, contém 851264 linhas e 7 colunas/*features*, sendo elas:

- *date*: *string* no formato aaaa-mm-dd com a data a que diz respeito a informação que se segue;
- *symbol*: sigla da empresa no mercado de ações considerado;
- *open*: valor com o qual uma dada empresa abriu a sessão;
- *close*: valor com o qual uma dada empresa fechou a sessão;
- *low*: ponto mais baixo na sessão considerada;
- *high*: ponto mais alto na sessão considerada;
- *volume*: quantidade de ações movimentadas na sessão.

Assim, o objetivo dos programas desenvolvidos passou por considerar as caraterísticas *date*, *open*, *low*, *high* e *volume* e prever o valor *close*.

B. Etapas consideradas

De modo geral, os dados foram processados ao longo das seguintes etapas (e na ordem apresentada):

- converter a coluna *date*, originalmente no formato *string* em inteiros usáveis;
- dividir o conjunto original em 3: treino, com cerca de 681000 instâncias (80%); validação com cerca de 85000 (10%) e teste também com 85000 instâncias (10%);
- normalizar os dados no intervalo [0,1] ao longo de cada coluna e de forma independente para os 3 conjuntos de acordo com a seguinte fórmula:

$$\frac{x - min(x)}{max(x) - min(x)}$$

III. Experiências realizadas e Discussão

A. Abordagens estudadas

A secção anterior descreveu um processo comum às 3 metodologias aplicadas. Daqui em diante, as ramificações começam a surgir de forma cada vez mais aparente:

- Método 1 - **Closed-Form**: É claramente a forma mais elegante e direta de obter a solução desejada. O seu nome advém do facto de, numa única etapa e com recurso a computações transversais, se obterem os parâmetros (θ's) ideais para a tarefa em questão. A fórmula apresenta-se a seguir:

  $$\theta^* = (X^T X)^{-1} X^T y,$$

  em que *X* representa as características do problema (numa matrix (*n_instâncias,n_caraterísticas*)), *y* os valores desejados e θ^* os parâmetros ideais.
- Método 2 - **Descida do Gradiente**: É o método base da literatura para problemas de otimização. Faz uso das noções de derivadas parciais, procura pelo mínimo global e a fórmula de atualização de parâmetros que se segue:

  $$\theta_i^+ = \theta_i - lr \frac{\partial J}{\partial \theta_i},$$

  com θ_i^+ a representar o novo valor de um determinado parâmetro θ_i, *lr* a taxa de aprendizagem (ou seja, a magnitude com que se atualiza um parâmetro de cada vez) e *J* a representar a função de custo escolhida (e que se pretende minimizar). A noção de derivada parcial é fundamental, pois mede o impacto do parâmetro θ_i no custo total.
- Método 3 - **Descida do Gradiente em Tensorflow**: É uma outra implementação do método 2, com aproveitamento da biblioteca de aprendizagem automática Tensorflow [3]. Tal biblioteca facilita o uso de GPU's (com o seu processamento paralelo e altamente otimizado para estas tarefas) e baseia-se no paradigma de grafos de execução para modelar as operações desejadas.

B. Configuração padrão

Com vista a obter resultados que se pudessem considerar de *baseline*, foram postas em prática as seguintes configurações:

Learning Rate	Número de Épocas	Intervalo de Normalização
0.0000001	200	[0,1]

TABLE I: Configurações padrão utilizadas

Importa referir que os parâmetros tiveram os seus valores inicializados dentro de intervalos que continham, de forma relativamente próxima, os parâmetros ideais obtidos com a Closed-Form (a inicialização foi claramente informada).

Tendo em conta estes meta-parâmetros, os resultados apresentam-se de seguida:

Método	Erro	Tempo de Execução
Closed-Form	0.0000001	5s
Descida do Gradiente	1.32	20m
Descida do Gradiente em Tensorflow	0.0000003	15s

TABLE II: Configurações padrão utilizadas

Pela análise da tabela pode-se concluir que a Closed-Form obtém os melhores parâmetros (com a vantagem adicional de, neste caso, também ser o método mais rápido). De notar que as versão em Tensorflow foi a que apresentou maior flutuação em termos de resultados.

C. Experiência 1 - Número de Épocas

A primeira experiência que se procurou realizar consistiu na alteração do número de épocas decorridas. Para efeitos de teste, o número de épocas foi incrementado para 500. O método 1 (Closed-Form) é alheio a estas mudanças de meta-parâmetros. Posto isto, seguem-se os resultados:

Método	Erro	Tempo de Execução
Closed-Form	0.0000001	5s
Descida do Gradiente	1.07	47m
Descida do Gradiente em Tensorflow	0.0000497	16s

TABLE III: Configurações utilizadas na experiência 1

A análise da tabela revela um facto curioso. A implementação normal de descida do gradiente continuou a sua tendência de descida do valor do erro, ao passo que a versão do Tensorflow piorou (é de realçar que o erro anterior já era bastante bom, logo a possibilidade de neste teste piorar era maior). Tais flutuações podem ser explicadas com a inicialização dos parâmetros, que sendo aleatória (dentro do que foi discutido anteriormente) acarreta estes riscos.

D. Experiência 2 - Learning Rate

A segunda experiência que se procurou realizar consistiu na alteração da taxa de aprendizagem. Este hiper-parâmetro viu o seu valor ser incrementado para 0.000001 (10 vezes superior). O resultados apresentam-se de seguida:

Método	Erro	Tempo de Execução
Closed-Form	0.0000001	5s
Descida do Gradiente	1.4	21m
Descida do Gradiente em Tensorflow	0.0000600	15s

TABLE IV: Configurações utilizadas na experiência 2

É possível perceber que a configuração de *Learning Rate* anterior era mais adequada à aprendizagem, uma vez que ambas as implementações de Descida de Gradiente viram os seus erros aumentar (em magnitudes distintas, mas a tendência manteve-se).

IV. Conclusão

Ao longo deste documento foi possível perceber como existem 3, entre muitas outras, formas de realizar a tarefa de aprendizagem de parâmetros que controlam a previsão final do modelo em uso. Foi possível, também, observar como a escolha inicial (e aparentemente arbitrária) de alguns meta-parâmetros pode impactar decisivamente a performance do modelo.

References

[1] NYSE. New York Stock Exchange. [Online] https://www.nyse.com/index

[2] Dataset. [Online] https://www.kaggle.com/dgawlik/nyse#prices.csv

[3] Tensorflow. [Online] https://www.tensorflow.org/