isec
Politécnico de Coimbra

Inteligência Computacional

Docente
Inês Dominguês
Carlos Pereira

Alunos
Paulo Henrique Figueira Pestana de Gouveia - a2020121705
Nuno Alexandre Almeida Santos - a2019110035

November 13, 2022

Índice

1 Introdução

Este trabalho foi realizado no âmbito da Unidade Curricular de Inteligência Computacional,tem por objetivo treinar uma rede neuronal capaz de estimar o valor da Bitcoin num determinado minuto.

2 Descrição do caso de estudo e objetivos do problema

O Dataset escolhido foi Bitcoin Price USD, neste conjunto de dados os dados são gerados no intervalo de 1 minuto por uma API (Binance API) entre 1 de janeiro de 2021 a 12 de Maio de 2021. Inclui várias colunas que mostram a mudança real no preço da Bitcoin também mostra o preço Open, High, Low, Close da Bitcoin em minutos específicos.

- Features
  1. Preço de abertura num minuto específico (Open Price of particular minute);
  2. Preço alto num minuto específico (High Price of particular minute);
  3. Preço baixo num minuto específico (Low Price of particular minute);
  4. Fechar Preço num minuto específico (Close Price of particular minute);
  5. Volume total num minuto específico (Total volume of particular minute);
  6. Volume de ativos de cotação (Quote asset volume);
  7. Número de negócios para determinado minuto (Number of trades for particular minute);
  8. Volume de ativos base de compra do tomador (Taker buy base asset volume);
  9. Volume de ativos de cotação de compra do tomador (Taker buy quote asset volume).
- Exemplos: 188318

Como o Horário de abertura (Open Time) e o Horário de fecho (Close Time) são sempre iguais, retirámos essas colunas das features.

3 Descrição da implementação dos algoritmos

Para treinar o nosso modelo optámos por usar o PyTorch que é uma biblioteca Python de aprendizagem máquina de código aberto. A principal diferença entre PyTorch e TensorFlow é a escolha entre simplicidade e desempenho: o PyTorch é mais fácil de aprender (especialmente para programadores Python), enquanto o TensorFlow tem uma curva de aprendizagem, mas tem um desempenho melhor e é mais utilizado na comunidade de desenvolvedores.

Os algoritmos de treino usádos foram:

- MSE (Erro quadrado médio)

$$MSE : \sum_{i=1}^{D} (x_i - y_i)^2 \tag{1}$$

- MAE (Erro absoluto médio)

$$MAE : \sum_{i=1}^{D} |x_i - y_i| \tag{2}$$

- RMSE (Raíz quadrada do erro médio)

$$RMSE : \sqrt{\frac{1}{n} \Sigma_{i=1}^{n} \Big(\frac{d_i - f_i}{\sigma_i} \Big)^2} \tag{3}$$

- RSQUARED (Coeficiente de determinação)

(4)

Funções de ativação usadas:

- Linear

$$Linear : xA^T + b \tag{5}$$

- AdaptiveLogSoftmaxWithLoss

- RReLU

$$RReLU(x) = \begin{cases} x, & \text{if } x \geq 0 \\ ax, & \text{caso contrário} \end{cases} \tag{6}$$

- LogSigmoid

$$LogSigmoid(x) = \log(\frac{1}{1 + \exp(-x)}) \tag{7}$$

Treinámos 4 redes para testar valores:

1. Rede MSE
   - Nome - network MSE 2Lay 256 Linear 8 94e 05.tar
   - Função de Treino - MSE
   - Função de Ativação - Linear
   - Nº de Neurónios - 256
   - Nº de Camadas - 2
   - Learning Rate - 3e-4
   - Erro de teste - 8,94E-05

2. Rede RMSE:
   - Nome - network RMSE 2Lay 256 Linear 0 00228.tar
   - Função de Treino - RMSE
   - Função de Ativação - Linear

- Nº de Neuronios - 256
- Nº de Camadas - 2
- Learning Rate - 3e-4
- Erro de teste - 0,0028

3. Rede MAE:
    - Nome - network MAE 2Lay 256 Linear 0 00173 .tar
    - Função de Treino - MAE
    - Função de Ativação - Linear
    - Nº de Neuronios - 256
    - Nº de Camadas - 2
    - Learning Rate - 3e-4
    - Erro de teste - 0,00173

4 Análise de Resultados

5 Conclusões

6 Referências

[?]