
Inteligência Computacional

Docente

Inês Dominguês
Carlos Pereira

Alunos

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

January 7, 2023

Índice

1	Descrição do Problema	1
2	Descrição das Metodologias	1
2.1	Metodologias Comum em ambas as fases	1
2.2	Metodologias Manual	1
2.3	Metodologias GridSearch	2
3	Referências	2

1 Descrição do Problema

Enfrentamos um problema de regressão em que nosso objetivo é treinar uma rede densa e uma rede de convolução de uma dimensão para estimar o valor do Bitcoin a cada minuto.

Sabemos que não será possível estimar o valor exato do Bitcoin, pois existem fatores externos que não podemos controlar ou prever (como Elon Musk e outros influenciadores), mas nosso objetivo é ficar o mais próximo possível.

Dado que nossa rede será treinada com exemplos de 1/01/2021 a 12/05/2021, após o treinamento da rede, compararemos nossos valores obtidos com os valores reais nos dias após o último dia em que a rede foi treinada.

Inicialmente, a escolha dos hiperparâmetros é feita manualmente, realizamos vários testes com diferentes variações, analisando o impacto de cada um na performance.

Em seguida, usaremos a "GridSearch" para tomar essa decisão de hiperparâmetros por nós e usaremos o PSO para explorar iterativamente o espaço de pesquisa e encontrar a solução ótima.

Compararemos ambas as metodologias para ver qual foram as vantagens e desvantagens e qual teve a melhor performance.

2 Descrição das Metodologias

2.1 Metodologias Comum em ambas as fases

O ficheiro CSV "main.csv" é lido e armazenado em um conjunto de dados, que é dividido em um conjunto de treinamento que contém 80% dos dados e um conjunto de teste com o resto.

O conjunto de dados é pré-processado usando o MinMaxScaler para garantir que todas as colunas tenham valores entre 0 e 1.

2.2 Metodologias Manual

Uma rede neural é criada usando o TensorFlow. A rede é composta por uma camada de entrada, seguida por camadas ocultas de neurônios e uma camada de saída.

A quantidade de camadas, neurônios, as funções de ativação usadas variavam conforme eram variados os testes.

Em seguida, o modelo é compilado usando um otimizador com um "learning rate" que também era variado, e uma função de perda "binary_crossentropy", pois estamos lidando com um problema de classificação binária (prever se o valor do Bitcoin aumentará ou diminuirá).

Após compilar o modelo, ele é treinado usando o método `fit()` do TensorFlow. O conjunto de treinamento e as etiquetas de treinamento são passados como argumentos para o método. Algumas configurações adicionais também são especificadas, como o número de épocas (iterações através do conjunto de treinamento) e o tamanho do batch (quantas amostras são usadas em cada atualização dos pesos da rede).

Durante o treinamento, o modelo ajusta seus pesos e biases para minimizar a função de perda. Isso é feito usando o algoritmo de otimização Adam, que atualiza os pesos da rede a cada lote usando uma combinação de gradiente descendente estocástico e momentum.

Após o treinamento, o modelo é avaliado usando o conjunto de teste e as etiquetas de teste correspondentes. O método `evaluate()` do TensorFlow é usado para calcular a precisão do modelo. A precisão é medida como a percentagem de previsões corretas feitas pelo modelo.

Finalmente, as previsões são feitas usando o método `predict()` do TensorFlow. O conjunto de teste é passado como argumento para o método e as previsões são armazenadas em uma variável.

Por fim, o modelo é salvo em um arquivo para que possa ser usado posteriormente. Isso é feito usando o método `save()` do TensorFlow.

2.3 Metodologias GridSearch

3 Referências

- [1] Firefly vs PSO.
- [2] Firefly parametros.