ĐỒ ÁN TTNHT

# Phát biểu bài toán giải quyết

# phát biểu lstm là gì

# phát biểu thực nghiệm

# bộ dữ liệu dụng la gi

# traning test

# chia như nào rồi

# xử lí dữ liệu ra sao?

# mô hình cụ thể của bài toán

# vẻ mô hình neural của bài toán đang xét

#định lượng độ chính xác của model bằng con số vd như trên tập 3000 dữ liệu thì đánh giá như nào

**mô hình dự đoán bằng cách:**

lấy 30 ngày đầu tiên(1, 30) tính ra ngày 31'

lấy 30 ngày tiếp theo (2, 31) tính ra ngày 32'

lấy 2 tập 31-> ... ss với 31' ->...

chia dữ liệu

x\_train(số giá trị, ngày, số feature)

y\_train(số giá trị, số feature)

x\_test và y\_test tương tự

1 lớp lưu thông tin lớp trước đó để dự doán tương lai

input\_shape(số ngày, số feature)

dense gộp các node thành 1 (fit vào thành 1)

- chia dữ liệu train/test theo tỉ lệ là 8:2

- build mạng bằng việc stack nhiều LSTM,

- Dropout là kĩ thuật giảm độ phức tạp của model bằng việc remove p% node sau mỗi lần backpro, ,

- define optimizer với loss function. Loss Function là mean square error nên khả năng bài toán của bạn là Regression.

- Bởi vì ở bước preprocessing bạn Normalize data train trong trường hợp này là data của bạn sẽ nằm trong khoản từ 0 đến 1 (Nghĩa là data nó bị scale theo 1 tỉ lệ nào đó). Đến khi train xong có model rồi. Bạn muốn test thì phải scale data test về tỉ lệ tương tự như train (mục đích để là đưa data mà model train với data mà model test về cùng distribution, Ko thể bạn train con chó mà bạn test con mèo dc.).

- Còn việc tại sao phải scale data về như vậy, lý do là: Khi bạn sử dụng Gradient Descent, hoặc adam thì quá trình backpropagation để tìm weight nó phụ thuộc vào data. Nếu một số điểm data nó có giá trị quá lớn hoặc quá nhỏ. Thì nó dễ gây ra 1 trong 2 hiện tượng đó là: Exploding Gradient Descent hoặc Vanishing Gradient. Nên bạn cần Normalize để Gradient được ổn định hơn.

tạo 1 model lstm:

dùng n lớp lstm lưu lượng x bit liền kề nhau

qua 1 lớp dense

Chương trình gồm:

1. chọn cổ phiểu cần giao dịch

2. chọn lệnh mua hoặc bán

3. xuất dữ liệu của cổ phiếu đấy cho tới ngày hôm nay

- xuất theo bảng hoặc biểu đồ

4. tiến hành dự doán

\* mua: dự doán giá giá cao nhất phiên(High)

\* bán: dự doán giá thấp nhất phiên(Low)

- lấy dữ liệu của 30 ngày trước đó để dự doán

5. kiểm tra độ chính xác của model đc train

- vẻ đồ thị để nhận biết độ chính xác một cách trực quan

=> Cuối cùng dự doán giá của cổ phiếu trong những phiên giao dịch tiếp theo