Документація до проекту ”Створення HFT-системи”

студента ЕК-2 Сусленка Володимира

Зміст

[1. Вступ 2](#_Toc470682657)

[2. Опис системи 3](#_Toc470682658)

[3. Архітектура системи 4](#_Toc470682659)

[3.1 Компоненти системи 4](#_Toc470682660)

[3.2 Торгівельний алгоритм системи 5](#_Toc470682661)

[3.3 Вхідні дані моделі (Feature Selection) 6](#_Toc470682662)

[3.4 Вихідний вектор моделі (Model Output) 8](#_Toc470682663)

[3.5 Алгорітмічна композиція 9](#_Toc470682664)

[3.6 Загальна логіка системи 11](#_Toc470682665)

[4. Клієнт в MS Office 12](#_Toc470682666)

[4.1 Інсталляція 12](#_Toc470682667)

[5. Недоліки, переваги та особливості системи 13](#_Toc470682668)

1. **Вступ**

Під час роботи над проектом та вивчення особливостей HFT, мною було досліжено, що будь-яка система високочастотної торгівлі має задовольняти наступні критерії:

* Мати достатню швидкість для опрацювання ордеру на біржі за мілі-/мікро- частки секунди.
* Мати мінімальну затримку між відсилкою даних з серверу трейдера до біржі.
* Мати прямий доступ до даних біржі у режимі реального часу.
* Мати прямий доступ для історичних даних біржі за певним індексом, валютою, тощо.
* Мати прямий доступ до створення ордеру на біржі.

Тобто, створена система не може бути системою HFT через наступні ознаки:

* Система створена за допомогою високорівневих мов та бібліотек, та не може конкурувати з системами створеними на низькорівневих мовах та технологіях (ASM, C, програмування на FPGA) через невелику швидкість.
* Відсутність історичних та поточних intraday-даних з біржи за певним індексом/валютою/тощо, з точністю за мілі-/мікросекундами у відкритому доступі в мережі Internet для навчання та створення алгоритму, аналізу змін цін, та знаходження кореляцій, залежностей у даних.
* Відсутність прямого доступу до біржі.
* Відсутність необхідного обладнання для роботи HFT-алгоритму.

Тобто, у завданні існує неточність.

Створена система є системою алгоритмічного трейдингу з базою даних, архітектурою сервер-клієнт, сервер – програмний код створений спеціально для проекту, клієнт - MS Excel.

1. Опис системи

Створена система є системою алгоритмічного трейдингу з базою даних, архітектурою сервер-клієнт, сервер – програмний код створений спеціально для проекту, клієнт - MS Excel.

Сервер оновлює дані, нормалізує та додає у БД, створює прогноз за допомогою нейронної мережі,

За допомогоє клієнта у MS Excel трейдер може побачити:

* Зміни ціни певного індексу/актива/валюти/тощо.
* Прогноз алгоритму щодо зміни ціни.
* Власний баланс та портфель.

Нейронна мережа навчена на курсі індексу S&P 500, дані якого знаходяться у БД.

**3. Архітектура системи**

**3.1 Компоненти системи**

Система створена за допомогою:

* MongoDB – нереляційна база даних, яка виконує функцію збереження:
  + фінансових даних.
  + даних, які використовуються для навчання алгоритму.
  + мета-даних.
  + даних, що відображують статус трейдера: його баланс та наявну кількість акцій.
* Python – мова програмування, яка має велику кількість бібліотек, доповнень, та фреймворків необхідних для фінансових обчислень, веб-програмування та передачі даних.
* Flask та його доповнення – фреймворк для Python з додатковими розширеннями для високорівневого, зрозумілого та ефективного веб-програмування.
* Keras – бібліотека для створення нейронних мереж (які були вибрані у якості алгоритмічної композиції для завдання аналізу зміни цін), має зручний графовий API для зручного створення моделі та можливість зберегти та відтворити натреновану модель.
* Numpy – бібліотека для зручної та ефективної маніпуляції даними, рахування необхідних математичних показників.
* Excel – клієнт, використаний як GUI для слідкування за роботою алгоритму.
* Додаткові бібліотеки та доповнення необхідні для Python.

Перелічені вище програмні доповнення є актуальними на момент здачі проекту.

## 3.2 Торгівельний алгоритм системи

У систему закладено простий алгоритм ведення торгів:

* У разі прогнозу зменшення ціни на актив – продати актив за поточною ціною.
* У разі прогнозу збільшення ціни на актив – купити актив за поточною ціною.

Вибрана стратегія торгів не передбачує збільшення цін на актив на послідовних часових проміжках, тобто система може отримувати прибуток лише від перепадів цін, тобто змін напрямків ціни.

## 3.3 Вхідні дані моделі (Feature Selection)

У якості вхідних даних до моделі були використані:

* **“Log\_return”** - Логаріфмічна різниця цін закриття (close price) за послідовні часові інтервали.
* **“Pseudo\_log\_return”** - Логаріфмічна різниця середніх цін (average price) за послідовні часові інтервали.
* **“Std”** - Середнеквадратичний розподіл цін (low, high, close, open) за часовий інтервал.
* Показник зростання(“**price\_growth**”) – бінарний показник, який показує чи зросла ціна закриття на актив у порівнянні з минулим часового проміжком. 1 – ціна закриття перевищує попередню ціну закриття. 0 – навпаки.

Модель використовує вищеперелічені фактори за останні 5 хвилин торгів. Тобто “**rolling window**” моделі дорівнює 5. У вхідному векторі існують 20 значень.

Можливими додатковими значеннями до вхідних даних моделі можуть бути:

* Година часового проміжку торгів. Число від 0 до 6.
* Хвилина часового проміжку торгів. Числов від 0 до 59.
* Перелічені часові значення повинні бути нормалізовані для існуванні у вхідному векторі моделі.

У якості нормалізації даних використовувався “Z-score”:

Також “**Z-score**” значень **“log\_return”** та **“pseudo\_log\_return”** був **помножений на 1000.**

Використання MinMax, Sigmoid нормалізації даних для вхідного вектора опціонально закладене в систему та може бути активоване.

## 3.4 Вихідний вектор моделі (Model Output)

У якості вихідного вектору при навчанні моделі використовувався “показник зростання” – “**price\_growth”.**

Загалом, датасет складається з часової послідовності з 450.000 значень у базі даних, 50.000 (найбільш близькі до поточного дня дані) з яких використовуються для тренування моделі.

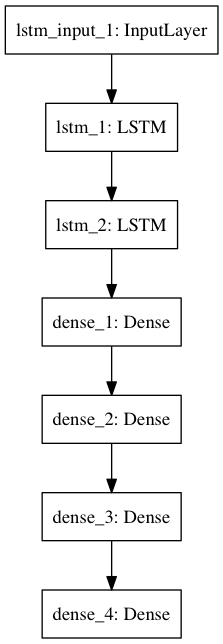
Датасет для тренування був розділений на тренувальний датасет та датасет для тестування у відношенні 85/15 відповідно.

У систему закладена можливість зробити тренування моделі на будь-якому часовому проміжку, на будь-яких даних, які були додані в БД.

## 3.5 Алгорітмічна композиція

У якості алгоритмічної композиції були використані нейронні мережі, оскільки, дані не можливо виразити за допомогою лінійної регресії.

Нейронна мережа була створена завдяки Keras та має наступну топологію:



Функція втрати – середньоквадратична похибка (mse, Mean Squared Error).

Функції активації нейронів перших 4 прошарків – гіперболічний тангенс.

Функція активації нейронів останніх 2 прошарків – лінійна.

Перші 2 прошарки є модифікацією LSTM рекурентного прошарку, та використані для роботи з даними часових послідовностей.

Наступні 4 прошарки – звичайна мережа, тобто “full-connected neural network”.

Dropout (випадання) не використовується у нейронній мережі.

Кількість нейронів у мережі за прошарками: 20:20:16:12:8:1.

Функція навчання мережі було підібрано еврестично – найкращі результати були отримані за допомогою алгоритму навчання RMSProp (82% тестів) та Nadam(Nesterov + Adam) (85% тестів).

3.6 Загальна логіка системи

Система працює за наступним циклом:

1. Отримати дані за поточну хвилину
2. Додати дані у БД
3. Нормалізувати дані
4. Створити вхідний вектор
5. Отримати прогноз від мережі
6. У разі росту ціни та дозволу трейдера - купити акції, інакше – продати

Цей цикл повторюється з інтервалом в одну хвилину.

4. Клієнт в MS Office

4.1 Інсталляція

У модулі Module 1 файлу проекту змінити адрес <http://localhost:3000> на адрес сервера в інтернеті(або на адрес локального сервера).

Клієнт може побачити графік зміни курсу, прогноз мережі на наступну секунду, власний баланс та кількість активів/акцій/валюти/тощо, які він має на даний момент, дані оновлюються один раз у хвилину.

5. Недоліки, переваги та особливості системи

**Недоліки:**

* Евристично підібраний вхідний вектор нейронної мережі.
* Можливий кращий тюнінг алгоритму системи.
* Слабка оптимізація системи.
* Необхідність кластеризації системи при її розширенні.
* Не враховані комісії біржі при купівлі/продажу активів.

**Способи покращення:**

* Покращити стабільність системи.
* Нормалізовувати дані від -1 до 1.
* Перехід до проблеми бінарної класифікації (зростання – зменшення цін) .
* Покращення стратегії торгів.
* Тюнінг моделі.
* Пошук інших значень для вхідного вектору або перехід до «навчання без вчителя».

**Особливості:**

* Система працює у режимі реального часу, усі дані обновлюються кожну хвилину.
* Усі фінансові та нормалізовані фінансові дані зберігаються у власній базі даних.
* Торги проходять без втручання трейдера.
* Точність прогнозування на тестовій виборці з 10000 послідовних хвилин: **85%.**
* Гнучкість клієнт-серверної архітектури.
* Постійне навчання мережі на нових даних(опціонально).
* Відсутність аналогічних проектів **з відкритим кодом.**