UNIVERZITA J. SELYEHO

SELYE JÁNOS EGYETEM

Fakulta ekonómie a informatiky

Gazdaságtudományi és Informatikai Kar

Real-time stock market price data analysis using neural networks

Diplomamunka

Bc. Eugen Fekete

ISBN 000-00-000-0000-0

2025, Komárno

UNIVERZITA J. SELYEHO
SELYE JÁNOS EGYETEM

NÁZOV FAKULTY
Fakulta ekonómie a informatiky
Gazdaságtudományi és Informatikai Kar

NÁZOV PRÁCE
Analýza údajov o cenách na burze v reálnom čase pomocou neurónových sietí

Ide jön az

Študijný program:	Aplikovaná informatika
Tanulmányi program:	Alkalmazott Informatika
Študijný odbor:	Informatika
Tanulmányi szak:	Informatika
Školiteľ:	László Marák, PhD.
Témavezető:	László Marák, PhD.
Školiace pracovisko:	Katedra informatiky
Tanszék megnevezése:	Informatikai Tanszék

Označenie typu práce - Diplomamunka
Bc. Eugen Fekete
ISBN 000-00-000-0000-0
2025, Komárno

aláírt témakiírás

Contents

List of Figures

Feladatkiírás

A szerző egy feladatot oldott meg. Még egy kicsit hosszabb, még egy kicsit hosszabb,

Opis práce

Autor vyrešil úlohu, ešte trošku dlhšie

Abstrakt

ešte trošku dlhšie, ešte trošku dlhšie

Kľúčové slová: klúč1, klúč2, klúč3,

Absztrakt

A szerző egy feladatot oldott meg. Még egy kicsit hosszabb, még egy kicsit hosszabb,

Kulcsszavak: kulcs1, kulcs2, kulcs3

Abstract

Little longer, Little longer,

Keywords: key1, key2, key3

Introduction

Machine learning (ML) plays a pivotal role in many areas of modern sciences, whether in industry, healthcare, finance and other fields. It can be used to provide a better service for the users of a search engine, a social media site or a media service provider by learning from the behaviour of the average user, predict stock prices within a specific time interval based on company performance measures and economic data, identify the risk factors for certain health conditions derived from clinical and demographic variables, identify the characters in a handwritten address from a digitized image, and so on. [3]

The main objective of ML is to find rules or patterns in data to achieve certain goals. In the financial world, for example, this might involve extracting useful information from the available data to support or automate investment activities. These activities include observing the market and placing buy or sell orders based on the conclusions drawn. [5]

1. Elméleti rész

Lorem ipsum dolores
https://shayandavoodii.github.io/OnlinePortfolioSelection.jl/stable/FL/

1.1 Machine Learning

The more common way of making a computer do work is to execute a computer program created by a human programmer. This program contains the steps and rules that turn input data into the appropriate answers, called output data. Machine learning mixes up these steps: the machine examines the input and output data, and tries to figure out what the rules should be. A system working like this is said to be trained rather than programmed. It is during the training process that the system identifies these rules by learning the patterns and relationships in the available data. [1]
Learning can be described using the definition provided by the renowned computer scientist and machine learning researcher, Tom Michael Mitchell:

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." (Tom M. Mitchell, 1997, p. 2)[8]

As an example, a text recognition program, a so called Optical Character Recognition (OCR) software can be presented. The main goal of such a program is to correctly recognize and convert handwritten characters into digitized text. A collection of texts written in various styles is presented to the OCR software. This collection, which the system uses to learn, is called the training set, where each instance is labeled appropriately. The actual machine learning part of the software that learns and makes predictions is called the model. In this example, the task T is to recognize handwritten characters and correctly classify them, the

experience E is the training set provided for learning and the performance measure P could be the accuracy of the recognition.
The example mentioned described a ML system performing supervised learning and solving a classification problem. We talk about supervised learning when a training set with appropriately labeled data is available for the learning process. Two other well known types of learning are unsupervised learning, where no training set with labeled data is available, and reinforcement learning, where a software agent learns rules by interacting with its environment. A classification problem is a problem where each input can be sorted into discrete number of classes. In the previous example each letter in a text can be classified as one of the letters of the alphabet. In contrast, when predicting land prices, we do not expect discrete labels as outputs, so we can't speak of classification problems. This is known as a regression problem and we expect continuous numerical values as outputs, for which a regression algorithm is used.

1.1.1 Types of learning

Machine learning systems can be grouped based on the type and amount of supervision received during the training. The most common types of learning are:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

[2]

1.1.1.1 Supervised learning

For supervised learning a training set is available. The training set consists of numerous observations with predefined inputs and a corresponding output. The inputs are called predictors (also called features) and the output is referred to as the response. The observations represents individual data instances. [3]
As an example lets say we want to predict house prices and we have the following training set:

Housing type	**Size (m^2)**	**Number of rooms**	**Location**	**Price (€)**
Single-family home	250	4	Bratislava	412000
Single-family home	390	6	Komárno	186000
Apartment	55	2	Trnava	118900
Single-family home	180	4	Košice	375000
...				

Table 1.1: Example training set with arbitrary values

In Table 1.1 each row corresponds to a single house for sale. These houses are the observations

and each column (excluding the last column) of the table, which represents the attributes of the houses, is an input, or a predictor. The last column is the output or the response.
The variables (be that predictor or response) can be described as either quantitative or qualitative. Quantitative variables are numerical values, like the size, number of rooms and price of a house in Table 1.1. Qualitative variables take values in one of K different categories, like the type of housing in Table 1.1. Problems with a quantitative response are referred to as regression problems, while problems with qualitative a response are referred to as classification problems. [4]
For each predictor measurement vector x_i (where $i = 1, ..., n$), there is a corresponding response measurement y_i. The goal of learning is to fit a model that links the response to the predictors, either to predict the response for future observations or to gain a deeper understanding of the relationship between the response and the predictors. [4]

Algorithms

CONTINUE: EXPLAIN THE FITTING METHOD USED IN THE FINISHED PROGRAM OR OTHER COMMON METHODS IN DEPTH

1.1.1.2 Unsupervised learning

If we only have a set of unlabeled measurement vectors x_i (where $i = 1, ..., n$) with no corresponding y_i response, we talk about unsupervised learning. Since we don't have an associated response, the goal of unsupervised learning is not to make predictions, but rather to analyze and discover patterns in the available data, which makes often makes working with unsupervised learning algorithms subjective. Furthermore, the lack of a corresponding response makes evaluating the obtained results difficult, as there is no way to check our work without knowing the true answer. This method of learning is often used as part of an exploratory data analysis. For example, a researcher might analyze gene expression levels in patients with certain health conditions to identify relationships or subgroups among the genes. A webstore could perform sales analyses to identify groups of customers with similar purchase histories for targeted advertisements and service customization. Similarly, a search engine might customize search results for users based on the click histories of other users with similar search patterns. [4]
For applications like these, where groups of objects or persons are identified, clustering algorithms are used. Another frequently used method in unsupervised learning is data visualization, which transforms complex, unlabeled data into a format suitable for plotting or further analysis. A related task is dimensionality reduction, where the aim is to simplify data with minimal information loss by performing feature extraction and merging correlated features. Anomaly detection algorithms can be used to detect fraudulent actions or to remove outliers from datasets during preprocessing. [2]

Algorithms
CONTINUE: EXPLAIN COMMON ALGORITHMS MORE IN DEPTH

1.1.1.3 Reinforcement learning

In reinforcement learning, a virtual object called an agent can make observation of its environment, commit actions resulting in rewards or penalties (usually defined as negative rewards) based on the outcome. [2]
In reinforcement learning, the goal is to figure out a strategy (called a policy) that tells the agent what actions to take in different situations, in order to get the most reward over time. The policy can either be deterministic, where the the agent always picks the same action for a given situation, or stochastic, where the action is chosen probabilistically. The agent needs to learn how actions and rewards are connected, which means it needs to explore the policy space through trial and error. Policy space refers to the set of all possible policies that an agent can adopt. [7]
More precisely, this policy could be a neural network, which takes observations as inputs and outputs the resulting action. Alternatively, the policy could be a genetic algorithm, where each generation consists of multiple policies and new generations of policy offsprings are created based on the genetic algorithm used. Reinforcement learning is commonly used in robotics and smart home appliances like thermostats. We can also find financial securities trading support systems using reinforcement learning. [2]
Stock Trading Bot using Deep Q-Learning [10]
A good example of a common approach in algorithmic trading can be illustrated by the "Stock Trading Bot using Deep Q-Learning" project created by Prabhsimran Singh. This project demonstrates an automatic stock trading bot using Deep Q-Learning, a type of reinforcement learning algorithm utilizing a deep neural network to approximate the optimal action-value function (also known as Q-function), which determines the best action to take in a given state. [9]. At any moment, the agent observes its current state (an n-day window of stock prices, which shows the stock prices of the last n days), selects and performs a buy/hold/sell action. The next state is evaluated and the agent receives a reward accordingly, based on changes in the portfolio value. The agent then updates its parameters by minimizing the computed loss. In reinforcement learning loss refers to the differnece between the target Q-value and the predicted Q-value, with the Q-value defining the optimal state-action (quality) values [2].
TEST RESULTS FROM THE CODE COMES HERE.
A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem [6]
Another example of the use of reinforcement learning in the financial world is presented in the paper of Zhengyao Jiang, Dixing Xu and Jinjun Liang, titled "A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem". The paper describes

the development of a system capable of automating portfolio management. Portfolio management is defined as the action of continuous reallocation of a capital into a number of financial assets. The core of the system is the Ensemble of Identical Independent Evaluators (EIIE) topology. In an EIIE, there are multiple evaluators (models or algorithms), following the same structure or design, but each working independently examining an asset's historical data to predict future growth. The outputs of their individual evaluations are aggregated to make a final decision. The resulting scores from this evaluation are used to determine the portfolio weights for the next trading period. These weights represent the actions the agent will take. If an asset's weight increases, it'll be bought or if it goes down, it'll be sold. The portfolio weights from the last period are fed back into the EIIE for future predictions.
Three different types of evaluators were tested in this work, a Convolutional Neural Network (CNN), a Recurrent Neural Network (RNN) and a Long Short Term Memory (LSTM) architecture. The framework was evaluated on a cryptocurrency exchange market in three different time period (to ensure that the tests cover different market conditions, like fast- or slow-moving market), each lasting 30 minutes. The tests included benchmarks, such as Best Stock (selects and invests in a single asset that achieved the highest final adjusted portfolio value), Uniform Buy and Hold (divides the initial capital equally among assets and holds them without rebalancing or trading), Uniform Constant Rebalanced Portfolio (divides the initial capital equally among assets and constantly rebalances the portfolio). Additionally, well-known strategies, such as ”Follow-the-Winner”, ”Follow-the-Loser”, ”Pattern-Matching”, and ”Meta-Learning” were also tested.
At the beginning of each period, the trading agent reallocates the fund among the assets. The trading fee for certain actions on the site is calculated as the maximum possible percentage, which is 0.25%. theframework was tested on historical data only with ideal conditions: zero slippage (meaning trade is executed immediately when an order is placed) and zero market impact (so the value of the trade is too insignificant to influence the market). For the portfolio preallocation the 11 most-volumed assets were selected. For training, historic price data from the past two days were used. The optimal policy was obtained using gradient ascent algorithm.
Algorithms
CONTINUE: EXPLAIN COMMON ALGORITHMS MORE IN DEPTH

1.1.2 Neural networks

1.2 Mégegy alfejezet

felsorolás

- LLE - *low-level* emuláció - alacsony szintű emulátorok

- HLE - *high-level* emuláció - magas szintű emulátorok

1.2.1 Al-alfejezet

Még egy kicsit hosszabb, még egy kicsit hosszabb[1], még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hossz

felsorolás

- minden függvény-blokk atomikus, vagyis mindig lefut az elejétől a végéig, és nem szakad meg soha
- a függvény-blokkban nincsenek elágazások
- minden függvény-blokknak van egy maximum nagysága

Monospace betűtípus `fetch`, `decode` és `execute`

Nevesített paragrafus. A *dőlt* idézet [**Ubershaders:ARidiculous**].

Beilleszteni egy képet így kell: A képre mindig kell hivatkozni a szövegből!!!

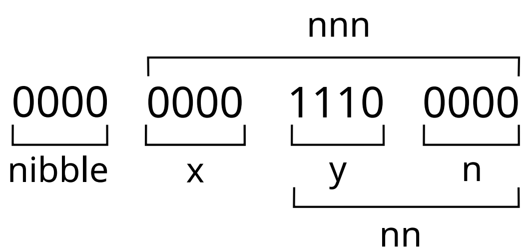

Figure 1.1: Utasítás felépítése

1.2.2 Így szedjük helyesen a C# nyelvet

[1] lábjegyzet

2. Gyakorlati rész

Még egy kicsit hosszabb, még egy kicsit hosszabb[1], még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hossz

[1] lábjegyzet

Befejezés

Még egy kicsit hosszabb, még egy kicsit hosszabb[2], még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hosszabb, még egy kicsit hossz

[2] lábjegyzet

Resumé

Autor vyrešil úlohu, ešte trošku dlhšie

References

[1] Francois Chollet. *Deep Learning with Python, Second Edition*. Manning, 2021. ISBN: 9781617296864 (cited on page 12).

[2] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, Third Edition*. O'Reilly Media, 2022. ISBN: 9781098125974 (cited on pages 13–15).

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer, 2009. ISBN: 9780387848570 (cited on pages 11, 13).

[4] Gareth James et al. *An Introduction to Statistical Learning: with Applications in Python*. Springer, 2023. ISBN: 9783031387463 (cited on page 14).

[5] Stefan Jansen. *Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python, Second Edition*. Packt Publishing, 2020. ISBN: 9781839217715 (cited on page 11).

[6] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. "A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem". In: (June 2017), page 31. DOI: 10.48550/arXiv.1706.10059 (cited on page 15).

[7] Jens Kober, J. Bagnell, and Jan Peters. "Reinforcement Learning in Robotics: A Survey". In: *The International Journal of Robotics Research* 32 (Sept. 2013), pages 1238–1274. DOI: 10.1177/0278364913495721 (cited on page 15).

[8] Tom Michael Mitchell. *Machine Learning*. McGraw-Hill Education, 1997. ISBN: 9780070428072 (cited on page 12).

[9] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518 (Feb. 2015), pages 529–33. DOI: 10.1038/nature14236 (cited on page 15).

[10] Prabhsimran Singh. *Stock Trading Bot using Deep Q-Learning*. https://github.com/pskrunner14/trading-bot. 2018. (Visited on 01/05/2025) (cited on page 15).

ISBN 000-00-000-0000-0