

ДОДАТОК А

Слайди презентації

Атестаційна робота магістра

«Дослідження ефективності використання
нейромережових методів і моделей для підвищення
точності прогнозування вихідних показників об'єктів»

Виконав:
ст. групи ПЗм-17-1
Мінаков А.Г.

Керівник:
проф. Лесна Н.С.

1

Мета роботи. Актуальність теми

Метою роботи є визначення ефективності використання нейромережових методів і моделей з метою підвищення точності прогнозування вихідних показників об'єктів.

Актуальність теми атестаційної роботи обумовлена:

- прогнозування вихідних показників об'єктів дослідження, особливо фінансових часових рядів, є необхідним елементом будь-якої інвестиційної діяльності;
- моделі штучних нейронних мереж успішно використовуються у вирішенні складних задач, які не мають аналітичного рішення;
- безліч існуючих нейромережових моделей і розрізненні дані щодо їх ефективності не дозволяють здійснити однозначний вибір найбільш точних моделей прогнозування.

2

Об'єкт і предмет дослідження. Наукова новизна. Практичне значення

Об'єктом дослідження є прогнозування вихідних показників об'єктів.

Предметом дослідження є дослідження ефективності сучасних нейромережових методів і моделей для підвищення точності прогнозування вихідних показників об'єктів.

Наукова новизна полягає у визначенні критеріїв та розробці методики оцінювання ефективності нейромережових моделей прогнозування.

Практичне значення. Практична цінність роботи полягає порівнянні результатів аналізу показників ефективності нейромережових моделей MLP, GoogLeNet, Inception-V3 та ResNet.

3

Постановка задачі

- провести аналіз існуючих нейромережових методів і моделей;
- проаналізувати існуючі проблеми та визначити шляхи їх вирішення;
- визначити критерії та розробити методику оцінювання ефективності нейромережових моделей прогнозування;
- розробити нейромережову модель для прогнозування акцій компаній;
- розробити програму для визначення показників ефективності моделей прогнозування вихідних показників об'єктів;
- провести порівняльний аналіз показників ефективності нейромережових моделей прогнозування.

4

Навчання штучної нейронної мережі

Глибинне навчання – це техніка навчання нейромережі, яка використовує безліч шарів для вирішення складних проблем (наприклад, розпізнавання мови) за допомогою шаблонів.

Три загальні парадигми навчання:

- «з вчителем»;
- «без вчителя» (самонавчання);
- змішана.

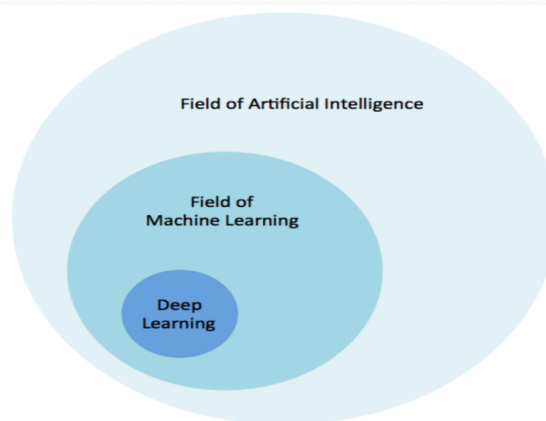
Основні правила навчання:

- правило Хебба;
- правило Хопфілда;
- навчання методом змагання.

5

5

Основні поняття глибинного навчання

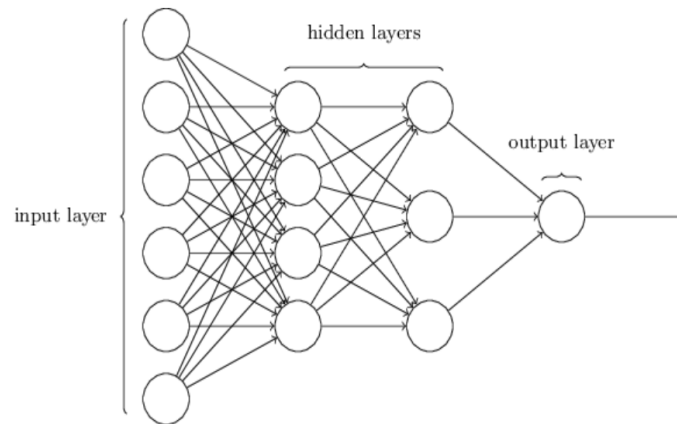


Візуалізація взаємодії глибинного навчання з машинним навчанням

6

6

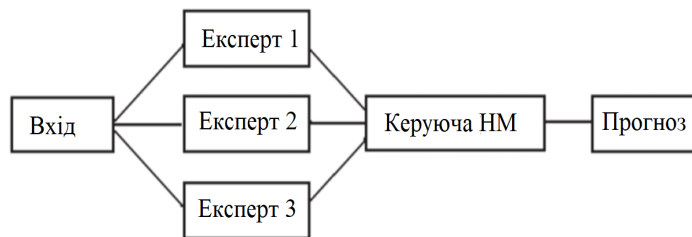
Архітектура глибокої мережі



7

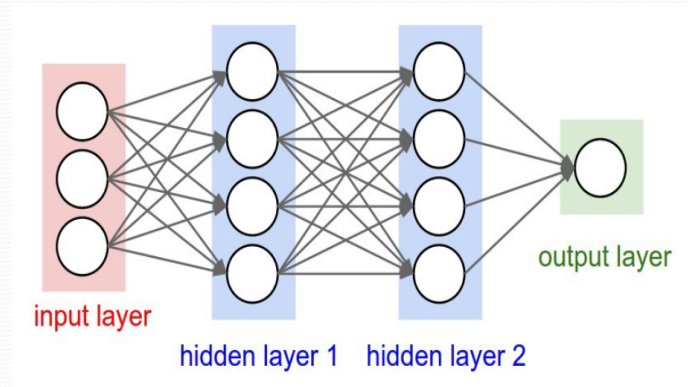
Нейромережеві моделі прогнозування фінансових часових рядів

Архітектура моделі прогнозування комітета нейроекспертів



8

Використання глибинних нейронних мереж для прогнозування часових рядів



Архітектура нейронної мережі для оцінки вартості квитка на літак

9

Методика оцінювання ефективності.

- Здійснити вибір критеріїв оцінювання ефективності нейромережевих методів прогнозування.
- Виконати моделювання точності прогнозування для моделей, обраних за визначеними критеріями.
- Здійснити вибір тієї моделі, яка демонструє найбільшу точність прогнозування.

10

Критерії оцінювання ефективності.

- Точність відноситься до близькості двох або більше вимірювань один до одного. Це повторюваність або відтворюваність вимірювання.
- Чутливість посиляється на частку релеванних екземплярів, котрі були отримані з загальної кількості екземплярів.
- AUC (Area under ROC curve, площа під ROC-кривою) – площа, обмежена ROC-кривою і віссю частки помилкових позитивних класифікацій. ROC-крива (Receiver operating characteristic) – графік, що дозволяє оцінити якість бінарної класифікації.

11

Порівняльний аналіз показників ефективності нейромережових моделей прогнозування

Модель прогнозування	Точність	Чутливість	AUC
MLP	89.44%	88.73%	0.93
GoogLeNet	86.11%	87.10%	0.92
Inception-V3	84.44%	86.44%	0.91
ResNet	82.78%	85.51%	0.89

12

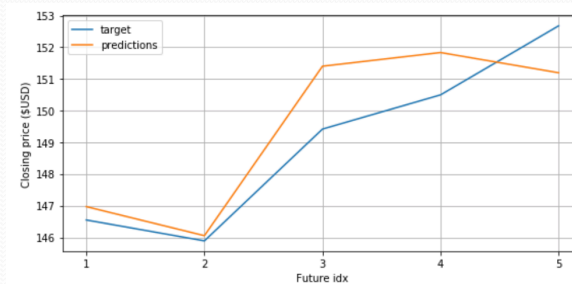
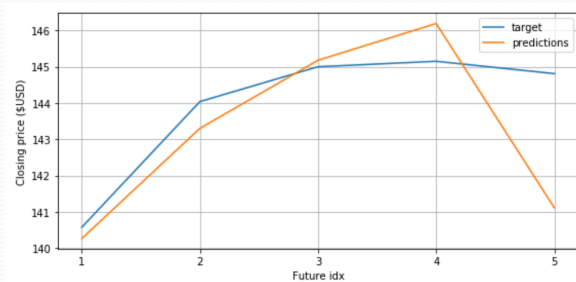
Прогнозування руху цін, акцій компанії на ринку

Графіки змінення цін акцій компанії Shopify



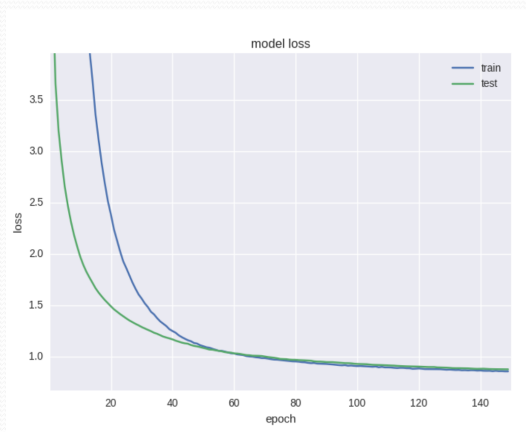
13

Графіки фактичних та прогнозованих цін акцій компанії



14

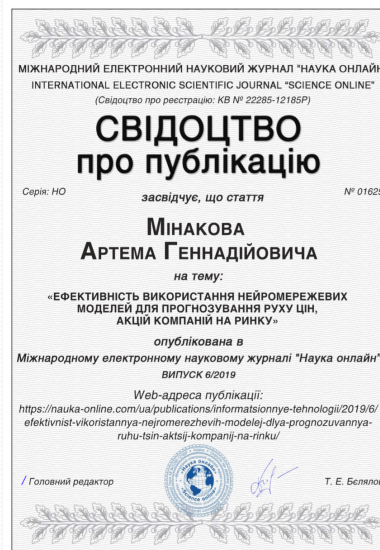
Відображення ефективності моделей прогнозування вихідних показників об'єктів у програмі



15

Апробація результатів

Стаття «Ефективність використання нейромережових моделей для прогнозування руху цін, акцій компаній на ринку» у міжнародному науковому електронному журналі «Наука онлайн»



16

ВИСНОВКИ

- Розглянуті нейромережеві методи і моделі прогнозування, виявлені їх недоліки та переваги.
- Розроблена методика оцінювання ефективності нейромережевих моделей, сформульовані критерії ефективності та їх вплив на вибір типу моделі.
- Розроблена програма для визначення показників ефективності моделей прогнозування вихідних показників об'єктів.
- Виконано порівняльний аналіз ефективності нейромережевих моделей прогнозування MLP, GoogLeNet, Inception-V3 та ResNet відповідно до обраних критеріїв оцінювання, що дозволяє обрати найбільш точну модель прогнозування.
- Розроблена нейромережева модель прогнозування руху цін акцій компаній на ринку

17

ДЯКУЮ ЗА УВАГУ!

18

ДОДАТОК Б

Лістинг коду програми

```
# multi-step encoder-decoder lstm
from numpy import array
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
from keras.layers import RepeatVector
from keras.layers import TimeDistributed
# define dataset
X = array([[10, 20, 30], [20, 30, 40], [30, 40, 50], [40, 50, 60]])
y = array([[40,50],[50,60],[60,70],[70,80]])
# reshape from [samples, timesteps] into [samples, timesteps, features]
X = X.reshape((X.shape[0], X.shape[1], 1))
y = y.reshape((y.shape[0], y.shape[1], 1))
# define model
model = Sequential()
model.add(LSTM(100, activation='relu', input_shape=(3, 1)))
model.add(RepeatVector(2))
model.add(LSTM(100, activation='relu', return_sequences=True))
model.add(TimeDistributed(Dense(1)))
model.compile(optimizer='adam', loss='mse')
# fit model
model.fit(X, y, epochs=100, verbose=0)
# demonstrate prediction
x_input = array([50, 60, 70])
x_input = x_input.reshape((1, 3, 1))
yhat = model.predict(x_input, verbose=0)
print(yhat)
```

ДОДАТОК В

Наукові публікації

В.1 Стаття у міжнародному науковому журналі Наука Онлайн

Інформаційні технології

УДК 004.7

Лесна Наталя Советівна

кандидат технічних наук,

професор кафедри програмної інженерії

Харківський національний університет радіоелектроніки

Мінаков Артем Геннадійович

студент

Харківського національного університету радіоелектроніки

**ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ НЕЙРОМЕРЕЖЕВИХ
МОДЕЛЕЙ ДЛЯ ПРОГНОЗУВАННЯ РУХУ ЦІН, АКЦІЙ
КОМПАНІЙ НА РИНКУ**

Анотація. У роботі були розглянуті моделі прогнозування фінансових часових рядів. Проаналізовано методи стандартного інтелектуального аналізу даних та його взаємодія з методами обчислювального інтелекту для вирішення задач прогнозування.

Ключові слова: прогнозування фінансових часових рядів, Keras, нейронні мережі, регресія, класифікація.

Проблема і актуальність дослідження. Задача прогнозування фінансових часових рядів була і залишається актуальною, оскільки передбачення є необхідним елементом будь-якої інвестиційної діяльності, адже сама ідея інвестування - вкладення грошей з метою отримання доходу в майбутньому - ґрунтується на ідеї прогнозування майбутнього. Останнім часом, коли стали доступні потужні засоби збору та обробки інформації, задача прогнозування фінансових часових рядів також стала однією з найпопулярніших задач для практичного застосування різних

методів інтелектуального аналізу даних (Data Mining) [1, с. 552]. Широке застосування Data Mining методів в даній області обумовлено наявністю в більшості часових рядів складних закономірностей, які не виявляються лінійними методами.

На сьогоднішній день існує безліч моделей прогнозування часових рядів: регресивні і авторегресійні моделі, нейромережеві моделі, моделі експоненціального згладжування, моделі на базі ланцюгів Маркова, класифікаційні моделі та ін. Найбільш популярними і широко використовуваними є класи авторегресійних і нейромережевих моделей.

Класичні штучні нейронні мережі багаторазово показали свою придатність в області фінансового аналізу в силу своєї високостійкої природи, здібностей до універсальної апроксимації.

Виходячи з цього слід зазначити, що використання моделей та методів нейронних мереж для прогнозування часових рядів, а саме фінансових рядів, які мають випадкову природу та непередбачуваність є актуальною метою даної роботи.

Огляд поточного стану об'єкту дослідження. Однією з основних переваг нейронних мереж є можливість імітації процесів з великою кількістю впливаючих параметрів, що є дуже складним або неможливим для традиційних методів в подібних умовах.

Найважливіша особливість мережі, яка свідчить про її широкі можливості і величезний потенціал, полягає в паралельній обробці інформації усіма ланками, що дозволяє значно прискорити процес обробки інформації. Інша, не менш важлива властивість - здатність до навчання і узагальнення накопичених знань. Нейронна мережа має риси штучного інтелекту [2, с. 120]. В даний час нейронні мережі використовуються для вирішення цілої низки задач, однією з яких є задача прогнозування.

Метою прогнозування є зменшення ризику при прийнятті рішень. Прогноз зазвичай виходить помилковим, але помилка залежить від

використовуваної прогнозуючої системи. Надаючи прогнозу більше ресурсів, можна збільшити точність прогнозу і зменшити збитки, пов'язані з невизначеністю при прийнятті рішень. Типовими додатками техніки прогнозу є передбачення цін на фондовій біржі, прогноз погоди, прогноз споживання електроенергії, прогноз відмов технічних систем та ін.

Найчастіше в роботах розглядається застосування нейронної мережі для вирішення задачі прогнозування часових рядів. Користувач вибирає довільний часовий ряд, що містить N відліків, і розбиває його на навчальну, тестову і контрольну вибірки, які потім подаються на вхід мережі. Результатом прогнозування є значення часового ряду в необхідний момент часу [3, с. 118].

Мета дослідження. Метою роботи є дослідження ефективності використання нейромережевих моделей для підвищення точності прогнозування вихідних показників об'єктів.

Однією з основних задач дослідження є проектування та реалізація системи для підвищення точності прогнозування вихідних показників на основі нейронних мереж.

Прогнозування часових рядів в Keras. Keras - відкрита нейромережева бібліотека, написана на мові Python [4, с. 243]. Вона являє собою надбудову над фреймворками DeepLearning, TensorFlow і Theano.

Підготовка даних. Для прикладу були взяті ціни акцій компанії Apple з 2005 по сьогоднішній день, які наведено на рисунку 1. Їх можна завантажити на Yahoo Finance в форматі .csv.



Рис. 1. Графік змінення цін акцій компанії Apple з 2005 року по сьогоднішній день

Графік має вигляд випадкового процесу. Для того, щоб вирішити задачу прогнозування на день або декілька днів вперед, необхідно задачу прогнозування описати ближче до задач машинного навчання. Метою є прогнозування значення ціни на наступний день (або через пару днів), прогнозування змінення ціни на наступний день в порівнянні з останнім днем, або логарифм цієї різниці, все це є задачею регресії. При вирішенні задач регресії виникає проблема нормалізації даних.

У разі класифікації, або у разі задачі регресії, на вхід ми візьмемо вікно часового ряду (30 днів) і передбачимо рух ціни на наступний день (класифікація) або значення зміни ціни (регресія).

Після вирішення задачі нормалізації значення параметру змінення цін знаходяться в інтервалі $[-0.5; 0.5]$. Для поділу на навчальну і тренувальну вибірки візьмемо перші 85% вікон у часі для навчання і останні 15% - для перевірки роботи нейронної мережі.

Для навчання нейронної мережі ми отримаємо наступні пари X, Y : ціни в момент закриття ринку за 30 днів і $[1, 0]$ або $[0, 1]$ в залежності від того, зросло або впало значення ціни для бінарної класифікації; процентна зміна цін за 30 днів і зміна на наступний день для регресії.

Архітектура нейронної мережі. В якості базової моделі використовується багат шаровий перцептрон. Реалізуємо сітку - вхідний шар з 30 нейронами (довжина вікна), перший прихований шар з 64 нейронами, після нього шар нормалізації, потім активаційна функція, далі вихідний шар з 1 нейроном (у разі задачі регресії), з 2 нейронами (у разі задачі класифікації). Для задачі регресії в кінці параметр активації повинен бути лінійним. Далі визначаються функції помилки і алгоритм оптимізації. Довжина кроку градієнтного спуску є 0.001. Для нормалізації параметру втрат в задачах класифікації потрібно визначити крос-ентропію, а для задачі регресії - середню квадратичну помилку.

Важливим моментом є навчання. Вчити алгоритми на таких даних треба довше 50-100 епох. Якщо провести аналіз тренувальних даних, буде видно, що 55% вікон були для одного патерну (підвищення, наприклад), а решта 45% - для іншого (зниження). У нашому випадку 53% вікон класу "зниження", а 47% - "підвищення", при цьому нашою метою є отримання точності вище 53%.

Результати вирішення задачі класифікації показали, що похибка і точність для тестової вибірки весь час залишається на приблизно одному рівні. При навчанні на тренувальній вибірці, похибка падає, а точність зростає, що говорить про перенавчання. Для порівняння було використано глибшу модель з двома шарами, але результати майже не відрізнялись.

Якщо з'являється ефект перенавчання, потрібно додати регуляризацию. Під час перенавчання будується модель, яка "запам'ятовує" тренувальні дані і не дозволяє узагальнити знання на нові дані. В процесі регуляризації накладаються певні обмеження на ваги нейронної мережі, щоб не було великого розкиду в значеннях і, незважаючи на велику кількість параметрів (тобто ваг мережі), частина з них перетворюється на нуль для спрощення. З точки зору функції похибки, така нейронна мережа навчається краще, але точність все ще страждає.

Варто додати ще більше регуляризації за допомогою популярної в останні роки техніки Dropout (метод вирішення проблеми перенавчання в нейронних мережах) - це випадкове "ігнорування" деяких ваг в процесі навчання, щоб уникнути ко-адаптації нейронів (щоб вони не вивчали однакові ознаки) [5, с. 533]. Результати використання техніки Дропаут показали, що графіки помилки і точності вже мають кращі результати. Якщо зупинити навчання мережі трохи раніше, можна отримати 58% точності передбачення руху ціни.

Ще один цікавий і інтуїтивно зрозумілий момент прогнозування фінансових часових рядів полягає в тому, що коливання в наступний день має випадкову природу, але коли ми дивимося на графіки, ми все-таки можемо помічати тренд на наступні 5-10 днів.

Треба перевірити, чи може з таким завданням впоратися наша нейронна мережа. Спрогнозуємо рух ціни через 5 днів з останньою вдалою архітектурою мережі і навчимо модель на більшій кількості епох. Якщо зупинити навчання досить рано (з часом все одно настає оверфітінг), то можна отримати 60% точності, що є дуже непоганим результатом.

Для задачі регресії візьмемо нашу останню успішну архітектуру для класифікації (вона вже показала, що вміє вивчати необхідні ознаки), приберемо Dropout і навчимо на більшій кількості ітерацій. Також в даному випадку ми можемо дивитися вже не тільки на значення похибки, а й візуально оцінити якість прогнозування. Результати навчання нейронної мережі для задачі регресії наведемо на рисунку 2.

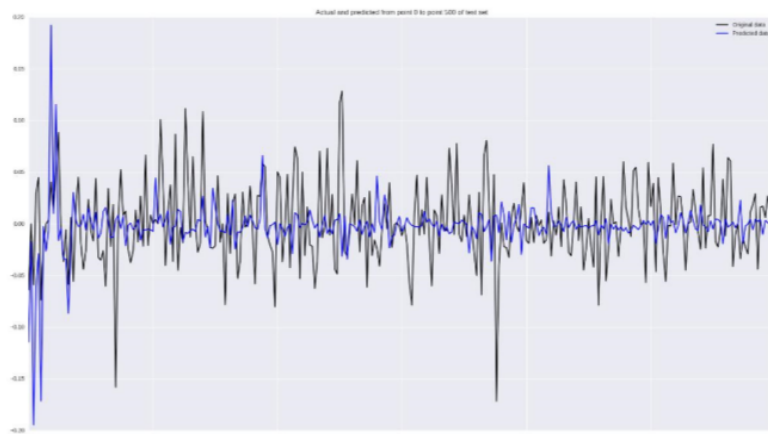


Рис. 2. Результати навчання нейронної мережі

Висновки. В рамках цієї роботи використовувалася нейронна мережа на одновимірних даних без особливої попередньої обробки. Є ряд методів, які дозволяють вивести точність на рівень 60-70% за рахунок навчання на високочастотних даних (кожну годину, кожні п'ять хвилин) або використання більш складних архітектур нейронних мереж, які призначені для роботи з послідовностями.

В цій роботі були застосовані найпростіші архітектури нейронних мереж для прогнозування руху цін на ринку. В якості базової моделі використовувався багаточаровий перцептрон, в якості фреймворка для імплементації - нейромережеву бібліотеку Keras. На основі порівняння результатів графіків похибок і точності навчання нейронної мережі при вирішенні задач класифікації та регресії, було правильно обрано алгоритм нормування даних, визначено архітектуру мережі, оцінено якість роботи алгоритму. В нашому випадку з точністю 60%, вдалося передбачати тренд через 5 днів, використовуючи вікно цін в попередні 30 днів, що є досить хорошим результатом.

Література

1. Friedman, J. The Elements of Statistical Learning. Data Mining, Inference, and Prediction / J. Friedman, T. Hastie, R. Tibshirani // Berlin: Springer. – 2003. – 552 p.
2. Dalto M. Deep neural networks for time series prediction with applications in ultra-short-term wind forecastin / M. Dalto, J. Matusko, M. Vasak // Spain: IEEE Internatiol Industrial Technology, Seville, March 17–19 th. – 2015. – 120 p.
3. Palit A. K. Computational Intelligence in Time Series Forecasting / Palit A. K.// Springer. – 2005. – 118 p.
4. Мейсман А. Основы Data Science и Big Data. Python и наука о данных / А. Мейсман, Д. Силен, М. Али // Питер – 2018. – 243 p.
5. Hastie T. The Elements of Statistical Learning, 2nd edition / T. Hastie, R. Tibshirani, J. Friedman // Springer – 2009. – 533 p.

ДОДАТОК Г
Електронні матеріали