

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп'ютерні науки,
освітня програма «Інформаційна аналітика та впливи»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

**“Інформаційний аналіз та прогнозування даних
автомобільного ринку”**

Студента 2-го курсу групи ІАВ-21

Науковий керівник:

Романенко Євгеній Миколайович

(прізвище, ім'я, по батькові)

Доктор технічних наук

(науковий ступінь, вчене звання)

Осауленко Ігор Анатолійович

(прізвище, ім'я, по батькові)

(підпис студента)

(дата)

(підпис)

Попередній захист:

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач
кафедри
технологій
управління

(підпис)

(прізвище, ініціали)

(дата)

Київ – 2023

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій**

Кафедра технологій управління
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 122 - Комп'ютерні науки
Освітня програма Інформаційна аналітика та впливи

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Морозов В.В.

« ____ » _____ 20__ року

**З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент Романенко Євгеній Миколайович
Група ІАВ-21

1. Тема кваліфікаційної роботи

Інформаційний аналіз та прогнозування даних автомобільного ринку

Затверджена наказом по від « ____ » _____ 20__ р. № ____.

2. Строк подання студентом готової роботи – “ ____ ” _____ 20__ р.

3. Цільова установка та вихідні дані до роботи: дослідження теоретичної бази сфери ринку автомобілів та застосування інформаційної аналітики у даній сфері; аналіз методів та методик аналізу та моделювання процесів, включно з програмними засобами реалізації; побудова прогностичної моделі та її оцінка; проведення інформаційної розвідки щодо потенційної імплементації отриманої моделі на підприємствах.

4. Зміст роботи: аналіз ринку автомобілів, інформаційна аналітика у сфері торгівлі автомобілів, методи інформаційного аналізу та прогнозування даних, математичні методи для моделювання, вибір даних для аналізу ринку автомобілів, обробка даних, побудова моделі методом нейронних мереж.

5. Перелік графічного матеріалу (слайдів): Виробництво автомобілів по категоріям; Динаміка виробництва автомобілів 2007-2018 роки; Динаміка продажів автомобілів 2007-2018 роки; Частка провідних компаній на ринку; Об'єм продажів автомобілів відомих марок; Приклад лінійної регресії; Результати роботи моделі; Графік прогнозів моделі; Графік середньоквадратичних помилок.

6. Календарний план виконання роботи:

| № з/п | Назва частин роботи | % | Виконання роботи | |
|-------|--|----|------------------|------------|
| | | | За планом | Фактично |
| 1 | Вибір теми дипломної роботи | 3 | 08.12.2022 | 08.12.2022 |
| 2 | Протокол кафедри ТУ про затвердження тем дипломних робіт та призначення наукових керівників | 2 | 08.12.2022 | 08.12.2022 |
| 3 | Формування переліку нормативних матеріалів, літератури з проблематики дипломної роботи | 10 | 08.01.2023 | 08.01.2023 |
| 4 | Складання розгорнутого плану кваліфікаційної роботи | 5 | 18.01.2023 | 18.01.2023 |
| 5 | Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін. | 5 | 20.01.2023 | 20.01.2023 |
| 6 | Підготовка розділу 1 “Теоретичні основи предметної області” | 10 | 13.02.2023 | 13.02.2023 |
| 7 | Підготовка розділу 2 “Прогнозування. Поняття і методи” | 15 | 06.03.2023 | 06.03.2023 |
| 8 | Підготовка розділу 3 “Інструменти прогнозування світового авторинку” | 20 | 03.04.2023 | 03.04.2023 |
| 9 | Підготовка розділу 4 “Розробка системи прогнозування світового авторинку” | 13 | 17.04.2023 | 17.04.2023 |
| 10 | Оформлення кваліфікаційної роботи. Підготовка висновків і пропозицій | 12 | 01.05.2023 | 01.05.2023 |
| 11 | Передача кваліфікаційної роботи науковому керівникові | 2 | 02.05.2023 | 02.05.2023 |
| 12 | Передача кваліфікаційної роботи рецензенту для рецензування | 1 | 10.05.2023 | 10.05.2023 |
| 13 | Попередній захист кваліфікаційної роботи | 2 | 17.05.2023 | 17.05.2023 |

Дата видачі завдання «___» _____ 20__ р.

Керівник роботи — Доктор технічних наук, Осауленко Ігор Анатолійович

(посада, прізвище, ім'я, по батькові)

_____ (підпис)

Завдання прийняв до виконання студент групи ІАВ-21

Романенко Свгеній Миколайович

(прізвище, ім'я, по батькові)

_____ (підпис)

ЗМІСТ

| | |
|---|----|
| АНОТАЦІЯ..... | 5 |
| ВСТУП | 7 |
| РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВНИ ПРЕДМЕТНОЇ ОБЛАСТІ..... | 9 |
| 1.1 Визначення автомобільного ринку | 9 |
| 1.2 Сучасний стан світового ринку автомобілів | 15 |
| РОЗДІЛ 2 ПРОГНОЗУВАННЯ. ПОНЯТТЯ І МЕТОДИ. | 28 |
| 2.1 Сутність прогнозування даних | 28 |
| 2.2 Методи прогнозування даних | 35 |
| 2.3 Якісні та кількісні методи | 39 |
| 2.4 Реляційні методи | 41 |
| РОЗДІЛ 3 ІНСТРУМЕНТИ ПРОГНОЗУВАННЯ СВІТОВОГО АВТОРИНКУ..... | 46 |
| 3.1 Нейронні мережі..... | 46 |
| 3.2 Інструментальні засоби розробки нейронних мереж | 57 |
| РОЗДІЛ 4 РОЗРОБКА СИСТЕМИ ПРОГНОЗУВАННЯ СВІТОВОГО АВТОРИНКУ..... | 65 |
| 4.1 Вимоги до технічного і програмного забезпечення | 65 |
| 4.2 Первинне тренування моделі..... | 66 |
| 4.3 Редукція моделі | 68 |
| 4.4 Прогнозування і відображення результатів | 70 |
| 4.5 Аналіз результатів роботи..... | 73 |
| ВИСНОВКИ..... | 76 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 77 |

АНОТАЦІЯ

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп'ютерні науки,

освітня програма «Інформаційна аналітика та впливи»

Дипломна робота магістранта Романенка Євгеній Миколайовича.

Тема роботи – «Інформаційний аналіз та прогнозування даних автомобільного ринку».

Мета дипломної роботи магістра – визначення та аналіз інформаційних методів та моделей для ефективного аналізу та прогнозування даних автомобільного ринку з метою покращення стратегічного планування та прийняття управлінських рішень.

Об'єкт дослідження - автомобільний ринок із своєю структурою, динамікою та факторами впливу, що включають продажі автомобілів, ціни, виробництво, витрати на маркетинг та інші параметри, що характеризують ринок.

Предмет дослідження - інформаційний аналіз та прогнозування даних автомобільного ринку, включаючи методи, моделі та інструменти для збору, обробки, аналізу та прогнозування даних, що впливають на автомобільний ринок.

Наукова новизна полягає у визначенні важливих аспектів функціонування ринку автомобілів, застосуванні алгоритмів машинного навчання для аналізу великої кількості даних про ринок автомобілів, розробці ефективної моделі машинного навчання, що допомагає виявляти складні зв'язки та тренди, що впливають на ціни.

Для досліджень теоретичних аспектів автомобільного ринку були використані методи дедукції та індукції, аналізу та синтезу. Для аналізу та прогнозування ціни було застосовано системний та комплексний підхід, статистичні методи збору та обробки інформації. Для прогнозування було використано метод нейронної мережі.

Дипломна робота складається зі вступу, основної частини, що включає 4 розділи, висновку, списку використаних джерел. Всього налічує 89 сторінку, перелік з 114 джерел.

Ключові слова: машинне навчання, методи, інформаційна аналітика даних, ринок автомобілів, моделювання, аналіз ринку автомобілів, прогнозування цін.

ВСТУП

Актуальність роботи. Ринок автомобілів грає важливу роль у економіці. Зацікавлені у розумінні сучасного стану та прогнозуванні тенденцій є як і звичайні покупці, так і компанії, які займаються ремонтом, продажем, пітримкою автомобілів.

Оскільки ринок автомобілів має велике значення для економіки країни, структурний аналіз та активний розвиток цього ринку є важливими напрямками досліджень. Дані методи стали первинним джерелом вдосконалених можливостей для консолідації, аналізу, прогнозування, звітності та візуалізації цін на автомобілі.

Мета дослідження: Визначення та аналіз інформаційних методів та моделей для ефективного аналізу та прогнозування даних автомобільного ринку з метою покращення стратегічного планування та прийняття управлінських рішень.

Завдання дослідження:

1. Здійснити огляд літератури та визначити основні теоретичні підходи та методи аналізу даних автомобільного ринку.
2. Зібрати та систематизувати наявні дані про автомобільний ринок, включаючи статистику продажів, ціни, виробництво, витрати на маркетинг та інші фактори, що впливають на ринок.
3. Провести статистичний аналіз даних для виявлення взаємозв'язків та тенденцій на автомобільному ринку.
4. Розробити моделі прогнозування на основі зібраних даних та використовуючи підходи машинного навчання та статистичного аналізу.

5. Здійснити валідацію та перевірку точності розроблених моделей прогнозування за допомогою історичних даних та порівняння прогнозів з реальними результатами.

Об'єктом дослідження є автомобільний ринок із своєю структурою, динамікою та факторами впливу, що включають продажі автомобілів, ціни, виробництво, витрати на маркетинг та інші параметри, що характеризують ринок.

Предмет дослідження є інформаційний аналіз та прогнозування даних автомобільного ринку, включаючи методи, моделі та інструменти для збору, обробки, аналізу та прогнозування даних, що впливають на автомобільний ринок.

Методи дослідження. Теоретичною основою дослідження стали загальнонаукові методи пізнання: аналітичний, індуктивний, дедуктивний, системний. Для розробки моделей були використані наступні інструменти:

- операційна система Windows XP і вище;
- інтерпретатор Python 3.7 або вище;
- встановлені бібліотеки, такі як:
 - pandas;
 - numpy;
 - sklearn;
 - matplotlib.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВНИ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення автомобільного ринку

Автомобільний ринок - це сукупність економічних відносин, завдяки яким відбувається взаємодія суб'єктів ринку з метою обміну готових автомобілів на кошти або їх еквіваленти.

Під автомобільним ринком у народі також мається на увазі якийсь територіально обмежений простір, призначений для торгівлі, в основному фізичними особами та індивідуальними підприємцями, готовими автомобілями та запасними частинами до них. У цьому сенсі часто вживають слово «Авторинок».

У цьому випадку автомобільний ринок сприймається як економічна категорія у світовому масштабі.

Об'єкт та суб'єкт

Об'єктом (товаром) автомобільного ринку є готовий автомобіль. Під готовим автомобілем мається на увазі механічний транспортний засіб, що має більше двох коліс, здатний до пересування без внесення додаткових змін до його конструкції. Товаром автомобільного ринку не можна вважати, наприклад: мотоцикли, човни тощо, окремі агрегати автомобіля та запчастини. Автомобіль як товар повинен задовольняти потреби чи потреби покупців автомобільного ринку.

Суб'єкти автомобільного ринка[1]:

- Представники попиту: домашні господарства (фізичні особи), а також комерційні та державні організації (юридичні особи).
- Представники пропозиції: виробники автомобілів (промислові підприємства) та продавці автомобілів (автомобільні дилери). У силу

галузевої специфіки, виробники автомобілів у окремих випадках реалізують товар безпосередньо кінцевому споживачеві.

- Держава, в особі Уряду.

Історія та розвиток

Історія автомобільного ринку починається з винаходом першого автомобіля та зародженням автомобілебудування в Німеччині в XIX столітті. Задовго до цього були винайдені екіпажі, що саморухаються, на парових і двигунах внутрішнього згорання, проте реалізація подібних автомобілів була незначною. Тільки наприкінці XIX століття Карл Бенц запропонував покупцеві готовий до експлуатації транспортний засіб, а Готліб Даймлер запустив у виробництво функціональний автомобільний двигун.

Першим у світі автомобілем, запущеним у виробництво в 1886 став триколісний автомобіль К. Бенца, сконструйований ним в 1885 році. Він мав такі технічні характеристики[2]:

- об'єм двигуна - 1,7 л, який розташовувався горизонтально (сконструйований К. Бенц в 1885 році автомобіль мав двигун об'ємом 985 см³, чого було недостатньо для покупців);
- потужність двигуна - 0,75 л. с.;
- електричне запалювання;
- двоступінчасту коробку передач.

Автомобілі Daimler, розроблені спільно Г. Даймлером і Вільгельмом Майбахом, були запущені у виробництво в 1895 році. Завдяки деяким технічним рішенням (наприклад: легко розташована легка рама), автомобілі «Daimler» були більш надійними, безпечними і легкими в керуванні. Приблизно з 1890 року почало розвиватись автомобільне виробництво у Франції. Найбільш відомою французькою фірмою того періоду була Panhard et Levassor. Першим самостійно розробленим та поставленим на виробництво

в Англії став автомобіль марки Lanchester у 1900 році. Таким чином, світовий автомобільний ринок зародився і почав розвиватись у Європі.

На розвиток автомобільного ринку великий вплив мали технічні зміни в конструкції та агрегатах автомобіля, а також технології виробництва. Вплив вказаних факторів зберігається й у час. На перших етапах розвитку автомобільного ринку покупцям нерідко доводилося купувати двигун і шасі в одного виробника, а кузов в іншого, оскільки багато фірм спеціалізувалися на чомусь одному. При цьому, в більшості випадків, неможливо було купити готовий товар: все робилося «на замовлення». Це було з переважанням у галузі кустарного виробництва. Перші автомобілі були розраховані лише на двох пасажирів. Зі зростанням попиту та збільшенням конкуренції, виробники стали прислухатися до вимог споживачів: з'явилися автомобілі на чотири пасажери.

За часів формування автомобільного ринку не існувало автомобільних брендів у сучасному їх розумінні. Кожна майстерня могла виготовляти авто під своїм ім'ям. Для того щоб виділити свою продукцію серед численних конкурентів, автовиробники прагнули брати участь у спортивних перегонах, що є проявом маркетингової стратегії. В даний час цим прийомом користуються лише виробники спортивних автомобілів.

Перші автомобілі були дуже дорогими, що обмежувало коло споживачів. Надалі автомобіль із предмета розкоші перетворився на предмет споживання. Особливо це відбувалося в США, де був найємніший автомобільний ринок у світі. Перший американський доступний автомобіль був виготовлений компанією "Oldsmobile" в 1901 році і називався "American Curved Dash".

Новий етап розвитку автомобільного ринку розпочався із запровадження в автомобільній промисловості масового способу виробництва з використанням конвеєра. Автомобілем, що ознаменував початок нової епохи, став Ford Model T в 1908 році, сконструйований Генрі

Фордом. Це був недорогий (більш ніж на третину дешевше за конкурентів) і надійний автомобіль, здатний проїхати навіть по сільських дорогах (дорожня система США на той час була слаборозвинена). Після декількох років виробництва Ford Model T, в 1913 році на підприємстві був введений метод стандартизації та взаємозамінності частин автомобіля, а також конвеєрна технологія складання. До 1923 року кожен другий автомобіль США (50 % ринку) носив марку «Ford»[3].

Третій етап розвитку автомобільного ринку пов'язані з використанням системи «ощадливого» виробництва (lean production). Вперше цей спосіб виробництва автомобілів був застосований компанією Toyota Motor Corporation наприкінці 1940-х років. Бережливе виробництво дозволяє досягти більш вигідного співвідношення ціна/якість в умовах дефіциту ресурсів і падіння попиту [4]. Також це дозволило скоротити терміни постачання товару та прискорити впровадження технічних новацій. Застосування "ощадливого" виробництва дозволило автовиробникам з Японії, особливо "Toyota", досить швидко завоювати велику частку ринку, потіснивши колишніх лідерів, таких як GM. Критерії конкурентоспроможності з цього моменту зазнали зміни. Споживачі побачили, що якісний автомобіль не обов'язково має бути дорогим.

З розвитком комп'ютерних технологій зростає вплив Інтернету розвиток автомобільного ринку. Багато людей користуються Інтернетом для купівлі-продажу вживаних автомобілів. Торгівля новими автомобілями через Інтернет поки що незначна навіть у розвинених країнах.

З початком світової економічної кризи 2008 року світовий автомобільний ринок пішов на спад. Обсяги продажів зменшилися на всіх ринках. Найменші втрати зазнав ринок Китаю. У 2010-х роках світовий ринок поступово зростає, наближаючись до докризових результатів. Важливу роль цьому виконали програми утилізації старих автомобілів, запроваджені більшості розвинених країн у період спаду.

В останні роки на розвиток автомобільного ринку зростає вплив технологічних новинок у розробці силових агрегатів автомобілів. Через подорожчання автомобільного палива та забруднення навколишнього середовища, споживачі прагнуть купувати більш економічні автомобілі. Свою роль у цьому грають і уряди, встановлюючи дедалі жорсткіші екологічні норми на викид шкідливих речовин автомобілями. Зростає, особливо у розвинених країнах Європи, кількість автомобілів із дизельними двигунами. Декілька компаній вже продають на ринку автомобілі з гібридними паливно-електричними двигунами. Серед них: Toyota, Lexus, VW, Honda. З'являються автомобілі виключно на електричній тязі, що заряджаються від електричної розетки. Розробляються автомобілі на водневих паливних елементах. Великим мінусом автомобілів на альтернативних джерелах енергії є їхня висока ціна.

Способи реалізації продукції

На ранніх етапах розвитку ринку, щоб купити автомобіль, покупцю необхідно було звертатися безпосередньо до фірми-виробника. На сучасному практичному практично всі автовиробники реалізують свою продукцію через посередників — автомобільних дилерів. Розрізняють офіційних та неофіційних («сірих») автомобільних дилерів. Дилери надають послуги не лише з продажу автомобілів, а й з технічного обслуговування та ремонту. Деякі автомобільні дилери можуть придбати підтримані автомобілі різних марок[5].

Класифікація ринку

За станом автомобіля:

- Ринок нових автомобілів.
- Ринок вживаних автомобілів.

За територіальною ознакою:

- ринок США;

- ринок Японії;
 - ринок Китаю тощо.
- За типом автомобілів:
- Ринок легкових автомобілів.
 - Ринок легких комерційних автомобілів (часто поєднується з ринком легкових автомобілів).
 - Ринок автобусів.
 - Ринок вантажівок та спецтехніки.

1.2 Сучасний стан світового ринку автомобілів

Автомобільна індустрія та автомобільний ринок тісно взаємопов'язані. Проте, створенням автомобілів та його реалізацією над ринком який завжди займаються одні й самі компанії. Посередниками між автовиробниками та кінцевими покупцями виступають автомобільні дилери, які пропонують покупцям автомобілі різних марок. Окрім продажу автомобілів, дилери пропонують послуги з ремонту автомобілів, проведення технічного обслуговування тощо. Причому, на сьогоднішній день автомобільні дилери стали настільки численною структурою, що існує в рамках автомобільного ринку, що з'являються навіть неофіційні автодилери. Вони переважно займаються збутом «автомобілів із пробігом». Розглядати автомобільних дилерів на світовому ринку та конкуренцію між ними недоцільно, оскільки ці учасники автомобільного ринку конкурують переважно у межах окремих країн.

Розвиток автомобільного ринку призвело до появи класифікацій автомобілів за різними параметрами: за класами, за моделями, за типом кузова, за кількістю кінських сил тощо. Крім того, враховуючи розвиток автомобільного ринку по світу, варто взяти до уваги і класифікацію автомобільних ринків. Наприклад, за територіальною ознакою прийнято виділяти автомобільний ринок США, Німеччини, Японії, Південної Кореї тощо. Залежно від новизни автомобілів, що продаються, виділяють ринки нових автомобілів і підтриманих.

Автомобільний ринок прийнято ділити на 4 сегменти: легкові автомобілі, LCV, вантажні автомобілі та автобуси.

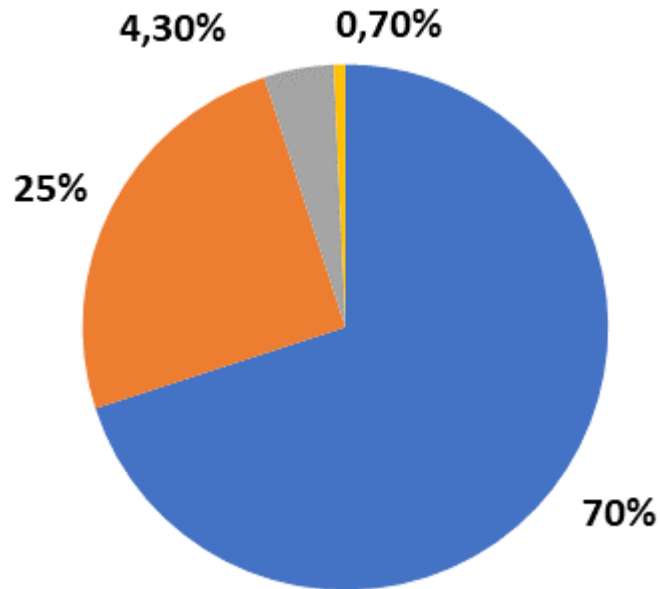


Рисунок 1.1 — Світове виробництво автомобілів по категоріям

Причому співвідношення світового виробництва автомобілів різних сегментів нерівномірне, що з потребами у тому чи іншому вигляді машин. Вже довгий час на світовому ринку зі значною перевагою переважають легкові автомобілі, займаючи понад 70% ринку. Далі йдуть автомобілі категорії LCV з часткою ринку 25%. На вантажні автомобілі та автобуси припадає 4,3% та 0,7% відповідно [7].

Оскільки найбільше на світовому ринку представлено легкових автомобілів, особливу увагу необхідно приділити класифікації, яка існує в цій категорії авто.

Серед легкових автомобілів по типу кузова виділяють:

- кросовери (CUV, SUV; автомобілі підвищеної прохідності),
- позашляховики (автомобілі, спеціалізовані для їзди позашляхом),
- пікапи (властиві мінімальні вантажні можливості та комфорт),
- мінівени (сімейні автомобілі підвищеної місткості),

- седани (відрізняються від мінівени окремим від пасажирського салону багажним відділенням),
- хетчбеки (відрізняються особливими можливостями багажного відділення),
- універсали,
- купе («укорочені» автомобілі, в основному лише 2 сидіннями),
- кабриолети та родстери (автомобілі з посиленням кузова та можливістю відкидного м'якого/жорсткого верху)
- спортивні автомобілі (має максимальні динамічні якості та якості керуваності).

Залежно від розміру автомобіля прийнято поділ легкових автомобілів на класи.

- А-клас («особливо малий клас», малогабаритні автомобілі: довжина – не більше 3,8 м, ширина – не більше 1,6 м);
- В-клас («малий клас»: довжина-3,4 ~ 4,4 м; ширина -1,6 ~ 1,75 м);
- С-клас («малий середній клас»: довжина – 4,2 ~ 4,6 м, ширина – 1,6 ~ 1,75 м);
- D-клас («середній клас»: довжина – 4,6 ~ 4,8 м, шириною – 1,7 ~ 1,8 м);
- Е-клас («вищий середній клас»: довжина – 4,8 ~ 5,0 м, а ширина понад 1,8 м);
- F-клас («представницький клас»: довжина понад 5,0 метрів, а ширина – понад 1,8 м).

Крім того, легкові автомобілі прийнято розділяти за ціновим критерієм на автомобілі бюджетні, середнього цінового діапазону та преміальні. Але такий поділ досить «розмите» і не дозволяють точно оцінити приналежність

автомобіля до тієї чи іншої групи, тому що необхідно враховувати не тільки конкретну ціну на автомобіль, але і комплекс технічних характеристик, пропонує виробників за певну плату.

LCV ("Light Commercial Vehicle" - це легкий комерційний автомобіль) - автомобіль, що використовується для перевезення вантажів малої тоннажності або пасажирів до 17 осіб. Ця категорія автомобілів включає пікапи, малотоннажні фургони, вантажівки (для вантажів не більше 3,5 т) і мікроавтобуси (місткість до 17 осіб). У деяких країнах виділяють LCV1 (автомобілі повної маси до 2,7 т) та LCV2 (повної маси від 2,8 до 3,5 т). Автомобілі LCV характеризуються потужним та економічним двигуном, низькими експлуатаційними витратами, пристосовані для використання у містах.

Вантажні автомобілі використовуються переважно у виробничих цілях на підприємствах, оскільки цей тип автомобілів спеціалізується на перевезенні вантажів. Вантажні автомобілі також мають складну структурну класифікацію. Залежно від можливостей перевезення прийнято виділяти вантажні автомобілі транспортного призначення (для перевезення всіх видів вантажів) та спеціалізовані (для перевезення лише певних видів вантажів). До категорії спеціалізованих автомобілів входять самоскиди, фургони, плитовози, скотовози, птаховози, причепа, лісовози, цистерни та багато інших. Крім того, важливим критерієм класифікації вантажних автомобілів є тип кузова. Тому виділяють вантажні автомобілі з відкритим кузовом та закритим. Причому для «вибагливішого» вантажу використовуються авто із закритим кузовом, і навпаки. До вантажного автотранспорту з відкритим типом кузова відносять бортовий автомобіль, довгомір, контейнервоз, автовоз, лісовоз та ін., а до автотранспорту із закритим кузовом відносять ізометричний фургон, фургон рефрижераторський, автоцистерна, фургон «Метелик» та ін.

Іншим не менш важливим критерієм класифікації є вантажопідйомність і тут виділяють автомобілі:

- Особливої малої вантажопідйомності – до 0,5 т
- Малий – від 0,5 до 2 т;
- Середній – від 2 до 5 т (легкі вантажівки);
- Великий – від 5 до 15 т (вантажівки середнього класу);
- Особливо великої вантажопідйомності – від 15 т і більше (важкі вантажівки).

Перевезення вантажу може здійснюватись як на близькі, так і на далекі відстані. Тому при виборі автотранспорту для транспортування товару необхідно брати до уваги середню відстань маршруту та стан доріг. На цій основі виділяють автомобілі: обмежену прохідність (для доріг з твердим покриттям), підвищену прохідність (для неупорядкованих дорого і частково для бездоріжжя) та високу прохідність (для подолання глибоких ровів, ям та інших складних перешкод).

Невелику частку на ринку завоював такий вид автомобіля, як автобус. Відмінністю автобуса від інших видів транспорту є наявність 9 і більше місць сидіння, призначених для перевезення пасажирів. Залежно від призначення автобуси можна поділити на транспортні (пасажирські) та спеціальні. Основна відмінність між ними – мета поїздки, перевезення громадян. До спеціальних автобусів відносять шкільні автобуси, медичні (реанімаційні, лабораторні та інших.), рекламні, для аварійно-рятувальних служб та інших. Пасажирські автобуси орієнтовані попри населення. Важливим критерієм для класифікації автобусів є місткість автобуса, причому враховується загальна кількість місць: як сидячих, так і стоячих. Місткість автобуса може бути: особливо мала (9-15 місць), мала (16-25 місць), середня (26-35 місць), велика (36-45 місць), особливо велика (більше 45 місць). Залежно від кількості поверхів в автобусі виділяють:

одноповерхові, півтораповерхові та двоярусні. Деякі автобуси відрізняються між собою корпусом, кількістю дверей, висотою та іншими зовнішніми та внутрішніми характеристиками. Ці відмінності пов'язані із призначенням автобусів. За призначенням автобуси бувають: міські, приміські та міжміські (туристичні). Іноді в цю класифікацію входять автобуси для обслуговування сільського населення, оскільки передбачається наявність певних технічних характеристик для руху по дорозі. Але найчастіше є різновидом міських автобусів.

Розвиток автомобільного ринку можна розглядати в 3 стадіях, що кардинально відрізняються один від одного принципом, що лежить в основі виробництва. Спочатку автомобіль розглядався як предмет розкоші, доступний лише малій частині населення. Потім, настала стадія масового виробництва та споживання автомобілів. Остання стадія пов'язана з обмеженістю ресурсів та використанням «ощадливого виробництва».

Історія розвитку автомобільного ринку починається із виробництва першого автомобіля. Основоположниками автомобільного ринку та творцями перших автомобілів, прийнято вважати, Карла Бенца та Готліба Даймлера, а країною зародження автомобілебудування – Німеччину. Хоча механізми на основі двигуна внутрішнього згорання були вигадані набагато раніше, вони не були введені в експлуатацію. У 1885 році Карл Бенц створив триколісний «віз із бензиновим двигуном», а трохи пізніше Готліб Даймлер створив функціональний автомобільний двигун. Вже за рік було накладено виробництво цих розробок.

У 1889 році Г. Даймлер та його друг Вільгельм Майбах створили свій перший функціональний автомобіль «Daimler», який у 1895 році був ведений в експлуатацію. Автомобілі «Daimler» відрізнялися особливою надійністю та безпекою. У 1892 році компанія Benz створила перший чотириколісний автомобіль на основі своїх попередників. Машина відрізнялася довговічністю

та особливою простою у використанні та в ремонті. У 1900 році з'явився прототип сучасного автомобіля - модель "Mercedes", яка поєднала в собі колишні стандарти моделей "Daimler" та технічні нововведення. З цього часу всі моделі «Daimler» називатимуться «Mercedes».

На той час виробництво автомобілів почало розвиватися у Франції. Лідруючі позиції займала фірма Panhardet Levassor, яка випускала автомобілі на основі двигуна Daimler. Іншою не менш відомою фірмою на автомобільному ринку Франції кінця XIX століття стала "De Dion-Bouton", створена графом Де Діоном та інженером Жоржем Бутоном. Ця фірма випускала недорогі автомобілі, легкі у використанні та ремонті, за рахунок чого мала велику популярність. Створений у рамках фірми "De Dion-Bouton" двигун ліг в основу створення перших автомобілів Renault.

Лише 1897 року почався розвиток автомобілебудування в Англії. Одним з найвідоміших і найкреативніших виробників автомобілів по праву можна вважати компанію «Lanchester». У 1900 році був розроблений перший автомобіль марки Lanchester, що відрізняється особливою надійністю і м'якістю їзди за рахунок унікальних технічних розробок. Але автомобілі не мали великого попиту через великий вплив системи «Panhard». Англія відома виробництвом спортивних автомобілів. Перші спортивні автомобілі з'явилися у компанії Napier. Одним із найпопулярніших автомобілів Англії, створений у 1906 році, вважається "Rolls-Royce 40/50", відомий більше як "Silver Ghost". Автомобіль не застарів і продовжував випускатися до 1925 року.

Згодом, з часом автомобіль з предмету розкоші перетворювався на предмет повсякденного споживання. Прискорено цей процес переходу відбувся у США, де автомобільний ринок вважається найемнішим у світі. У 1901 році в США з'явився перший доступний автомобіль, випущений компанією "Oldsmobile", під назвою "American Curved Dash". Справжнім проривом у галузі автомобільного виробництва прийнято вважати створення конвеєра Генрі Фордом, який дозволив забезпечити масове виробництво

автомобільної промисловості. Початком нового етапу розвитку автомобілебудування є створення Генрі Фордом в 1908 року автомобіля «Ford Model T». Дана модель мала особливу надійність, можливість справлятися з нерівностями на дорогах, причому була на третину дешевше своїх конкурентів. Саме таке співвідношення ціни та можливостей автомобіля зробило цю модель найпопулярнішою в Америці на той час. У 1923 році компанія «Ford» зайняла близько 50% ринку (Ford – кожен другий автомобіль у США).

Значні зміни в автомобільному ринку припадають на кінець 1940-х років. Компанія Toyota Motor Corporation вперше використовувала спосіб ощадливого виробництва (lean production), який враховує проблему обмеженості ресурсів і падіння попиту при формуванні співвідношення ціна/якість. Використання даних технологій дозволяли автовиробникам з Японії збільшити свої частки на світовому ринку і скласти сильну конкуренцію компаніям, що їдуть у цій галузі. На сьогоднішній день виробництво автомобілів у всіх країнах орієнтоване на існуючі екологічні проблеми та на способи зменшення шкоди від виробництва та від використання автомобілів на навколишнє середовище.

Розглянемо стан автомобільного ринку за останнє десятиліття щодо динаміки виробництва та продажу автомобілів у світі, а також позначимо причини зниження обсягів виробництва та продажу.

Виробництво на автомобільному ринку має переважно позитивну динаміку. Найбільші показники припадають на 2016 рік. Різкий спад помітний у 2008-2009 роках, що пов'язано зі світовою економічною кризою. Причому автомобільні ринки всіх країн зазнали сильних збитків, окрім ринку Китаю, який зміг збільшити сферу свого впливу. Автомобільна галузь відіграє важливу роль в економіці як окремої країни, так і світу в цілому.

Відповідно, суттєві зміни, що відбуваються на світовій арені, відбиваються на виробництві та продажах автомобілів у світі.

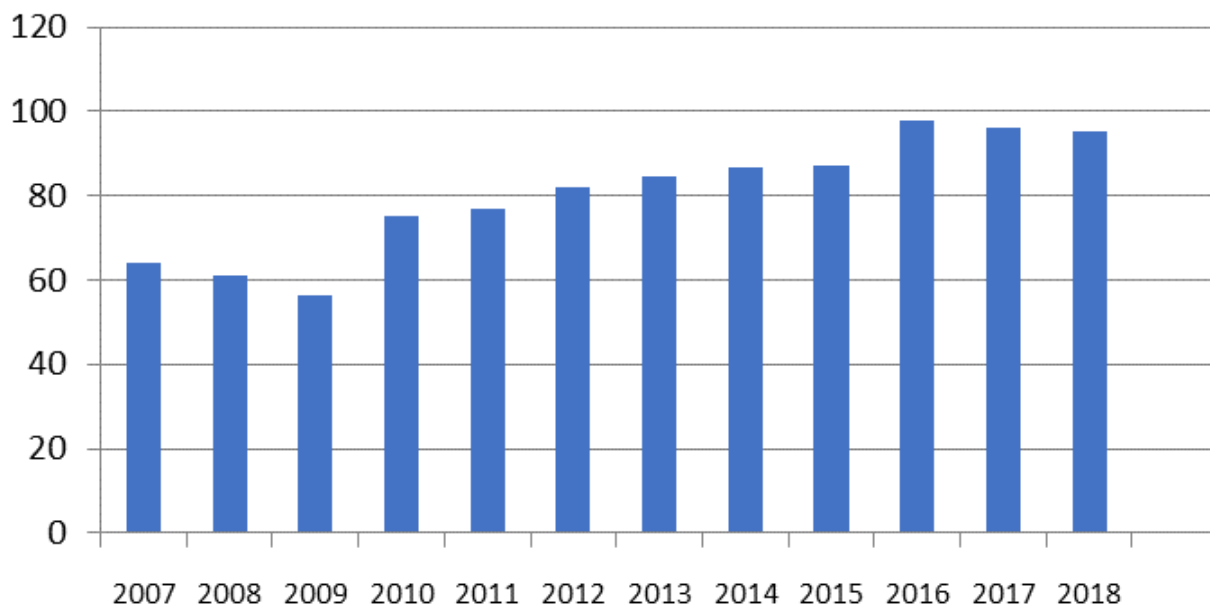


Рисунок 1.2 — Світове виробництво автомобілів, млн. шт

Якщо розглядати лідерів за обсягом виробництва автомобілів на 2018 рік, то до цієї групи входять Китай, США, Японія, Німеччина та Індія. Такі позиції країни посідають останні 5 років. Виняток становить Японія, яка змогла у 2016 році обігнати США за обсягом виробництва, та Індія, яка вперше потрапила до п'ятірки лідерів лише у 2016 році, а до цього моменту її місце посідала Південна Корея. Причому автомобільний ринок Індії вважається одним із найбільш швидкозростаючих у світі. Китай є абсолютним лідером, його обсяг виробництва на 2018 рік становив 27,8 млн шт. США, яке посідає друге місце, сильно відстає за показниками свого конкурента. Їх обсяг виробництва становив 11,3 млн шт. У 2018 році Німеччині вдалося трохи обійти Індію з різницею в обсязі виробництва на 0,2 млн шт. За даними на 2017 рік, Росія в загальному рейтингу посідає 16 місце із загальним обсягом виробництва автомобілів, рівним 1,5 млн шт. [8].

Згідно зі статистичними дослідженнями за 2018 рік за кількістю проданих автомобілів найбільшими автовиробниками світу є Volkswagen

Group (10,8 млн.), Toyota M.C. (10,4 млн.), Renault-Nissan Alliance (10,3 млн.), General Motors (8,6 млн.), Hyundai-Kia (7,4 млн.), Ford M.C. (5,6 млн.), Honda M.C. (5,2 млн), Fiat-Chrysler Alliance (4,8 млн), Peugeot-Citroen Alliance (4,1 млн), Suzuki (3,3 млн) [8].

Розглянемо динаміку продажів автомобілів у світі. В останнє десятиліття спостерігається збільшення обсягів продажу автомобілів. Спад у продажах автомобілів у 2008-2009 роках пов'язаний зі світовою економічною кризою. Ослаблення позицій середнього класу по всьому світу позначилося на продажі автомобілів.

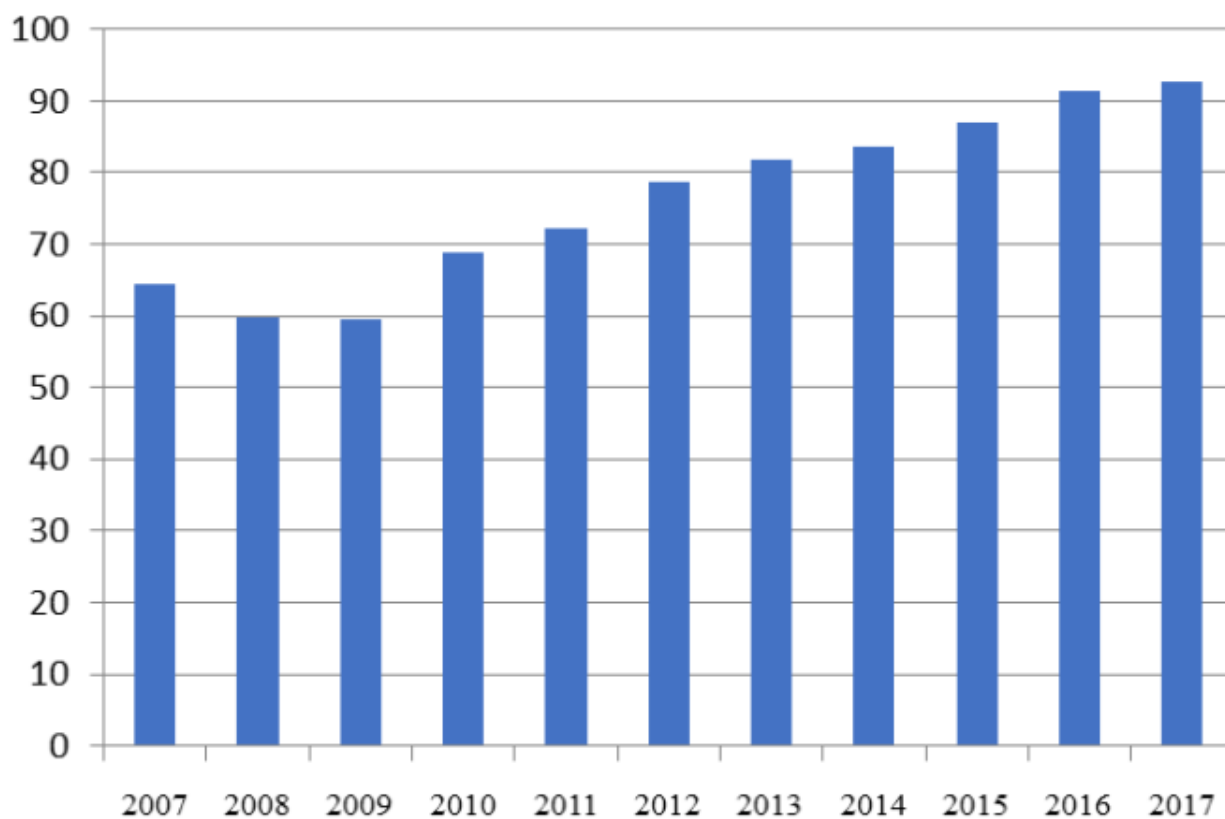


Рисунок 1.3 — Продажі автомобілів по світу, млн. шт.

Провідними світовими виробниками є Volkswagen Group, Toyota Group, Renault-Nissan Alliance, Hyundai-Kia, General Motors, Ford Group, Honda Motor, F.C.A. (Fiat Chrysler Automobiles), PS. (Peugeot Society Anonyme).

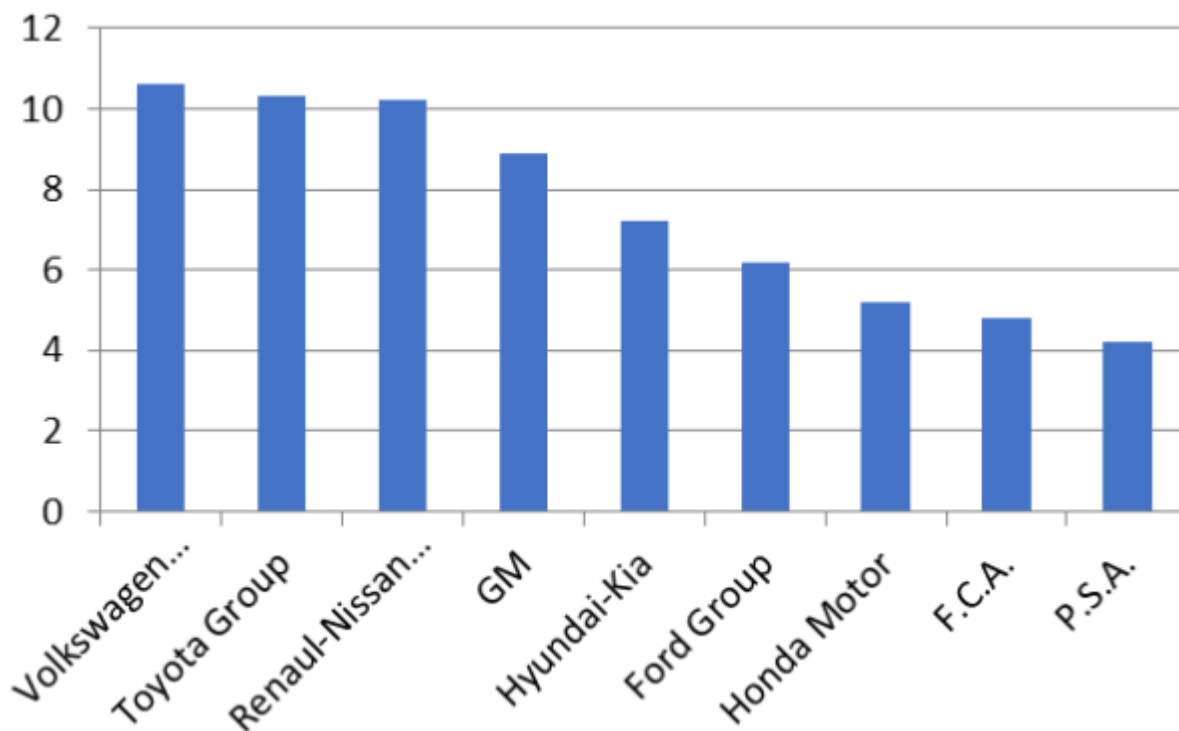


Рисунок 1.4 — Продаж автомобілів провідними компаніями, млн. шт

Великі автомобільні концерни включають кілька марок, що спеціалізуються на продукції різного класу і різної цінової категорії. Марки, зорієнтовані певного споживача конкурують як у межах одного концерну, і у умовах всього ринку. Згідно зі статистикою за 2017 рік, лідируючі позиції займають такі марки: Toyota (7,8 млн шт.), Volkswagen (6,6 млн шт.), Ford (5,9 млн шт.), Honda (4,9 млн шт.), Nissan (4,8 млн шт.), Hyundai (3,9 млн шт.), Шевроле (3,8 млн шт.), Suzuki (2,8 млн шт.), Mercedes-Benz (3,5 млн шт.), Kia (2,5 млн шт.).

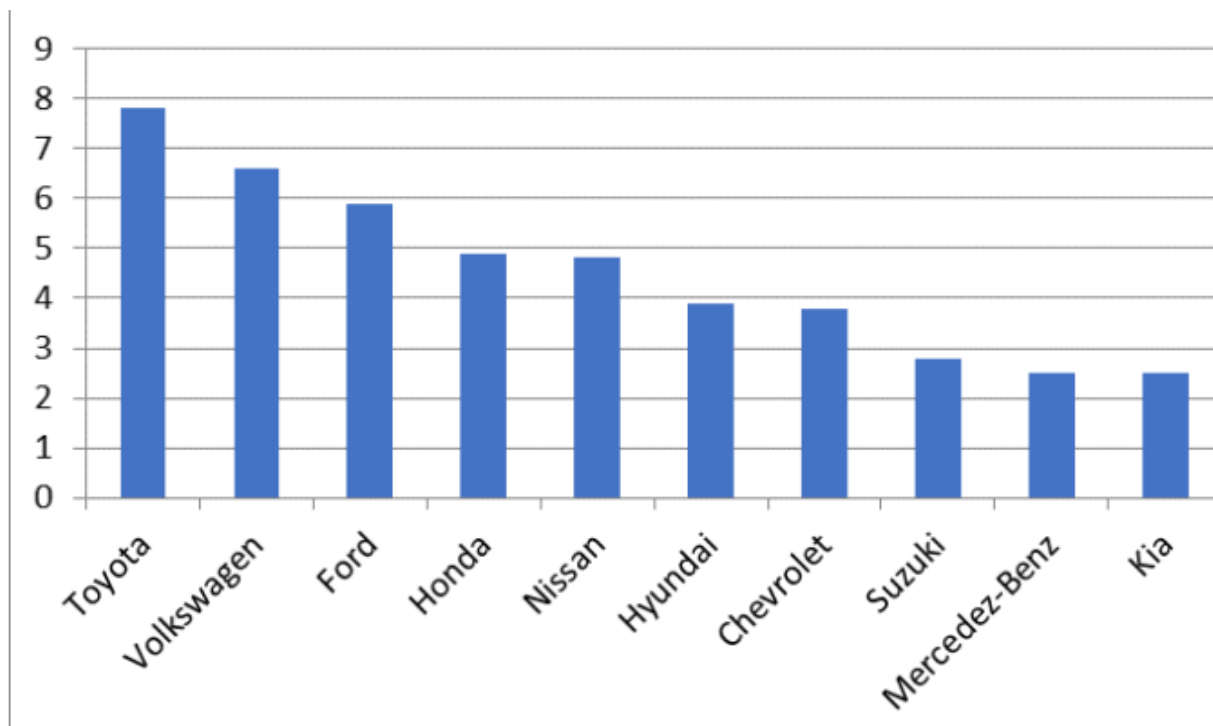


Рисунок 1.5 — Об'єм продажів автомобілів відомих марок

На основі проведеного дослідження світового автомобільного ринку, можна зробити такі висновки:

Розвиток автомобільної галузі – найважливіший показник стану національної економіки. Ця галузь відіграє виключно важливу роль в організації транспортного комплексу, виробництва та обігу продукції, промислового та сільськогосподарського секторів.

Зародження автомобільного ринку пов'язані з виробництвом першого автомобіля, введеного в експлуатацію, і посідає 80-ті роки XIX століття.

Автомобільний ринок різноманітний. Прийнято виділяти 4 основні сегменти: легкові автомобілі, LCV, вантажні автомобілі та автобуси. У межах окремих суб'єктів існує розширена класифікація автомобілів. Зі значною перевагою найбільший обсяг виробництва та споживання по світу припадає на легкові автомобілі.

Країнами з найрозвиненішими автомобільними ринками є Китай, США, Японія, Німеччина та Індія. Причому вони здобувають лідерство і з виробництва, і з споживання автомобілів. Країни займають ці позиції вже

багато років. Винятком є Індія, що відрізняється високими темпами зростання протягом останнього десятиліття.

50% ринку зосереджено в руках п'яти провідних компаній: Volkswagen Group, Toyota Group, Renault-Nissan Alliance, Hyundai-Kia, General Motors. Інша частина ринку розділена між безліччю невеликих компаній.

Успішні автомобільні компанії є альянсами і концернами, тобто об'єднання кількох автомобільних марок. Причому, як правило, автомобілі різні за характеристиками, що дозволяє зайняти велику частку на ринку і збільшити сферу впливу.

Для автомобільного ринку характерні угоди злиття та поглинання компаній, оскільки такий спосіб інтегрування дозволяє компаніям впоратися з фінансовими труднощами, зміцнити та покращити позиції в конкурентній боротьбі.

РОЗДІЛ 2

ПРОГНОЗУВАННЯ. ПОНЯТТЯ І МЕТОДИ.

2.1 Сутність прогнозування даних

Прогнозування — це процес створення прогнозів на основі минулих і поточних даних. Пізніше їх можна порівняти (вирішити) з тим, що відбувається. Наприклад, компанія може оцінити свій дохід у наступному році, а потім порівняти його з фактичними результатами, створивши фактичний аналіз відхилень. Прогноз — схожий, але більш загальний термін. Прогнозування може стосуватися конкретних формальних статистичних методів із застосуванням часових рядів, перехресних чи поздовжніх даних, або, альтернативно, менш формальних методів судження чи самого процесу прогнозування та вирішення. Використання може відрізнитися залежно від сфери застосування: наприклад, у гідрології терміни «прогноз» і «прогнозування» іноді зарезервовані для оцінок значень у певний конкретний майбутній час, тоді як термін «передбачення» використовується для більш загальних оцінок, таких як скільки разів відбуватимуться повені протягом тривалого періоду.

Ризик і невизначеність є центральними для прогнозування та передбачення; загалом вважається хорошою практикою вказувати ступінь невизначеності прогнозів. У будь-якому випадку дані повинні бути актуальними, щоб прогноз був максимально точним. У деяких випадках дані, які використовуються для прогнозування цікавої змінної, самі по собі є прогнозом.[1] Прогноз не слід плутати з бюджетом. Бюджети — це більш конкретні фінансові плани на фіксований термін, які використовуються для розподілу ресурсів і контролю, тоді як прогнози надають оцінки майбутніх фінансових показників, що забезпечує гнучкість і адаптацію до мінливих обставин. Обидва інструменти є цінними для фінансового планування та прийняття рішень, але вони виконують різні функції.

Прогнозування має застосування в багатьох галузях, де корисні оцінки майбутніх умов. Залежно від поля точність значно відрізняється. Якщо фактори, пов'язані з тим, що прогнозується, відомі та добре зрозумілі, а також є значна кількість даних, які можна використати, цілком імовірно, що кінцеве значення буде близьким до прогнозу. Якщо це не так або якщо на реальний результат впливають прогнози, то надійність прогнозів може бути значно нижчою.[2]

Зміна клімату та зростання цін на енергоносії призвели до використання Egain Forecasting для будівель. Це намагається зменшити енергію, необхідну для опалення будівлі, таким чином зменшуючи викиди парникових газів. Прогнозування використовується для планування попиту споживачів у повсякденному бізнесі виробничих і розподільчих компаній.

У той час як правдивість прогнозів щодо фактичної прибутковості акцій заперечується через посилення гіпотези ефективного ринку, прогнозування широких економічних тенденцій є звичайним. Такий аналіз надають як некомерційні групи, так і приватні комерційні установи.

Прогнозування коливань валютних курсів зазвичай досягається за допомогою поєднання діаграми та фундаментального аналізу. Суттєва відмінність між аналізом графіків і фундаментальним економічним аналізом полягає в тому, що чартисти вивчають лише цінову дію ринку, тоді як фундаменталісти намагаються шукати причини, що стоять за цією дією.[3] Фінансові установи об'єднують докази, надані їхніми дослідниками-фундаменталістами та чартистами, в одну замітку, щоб надати остаточний прогноз щодо відповідної валюти.[4]

Прогнозування використовувалося також для передбачення розвитку конфліктних ситуацій.[5] Прогнозисти проводять дослідження, які використовують емпіричні результати для вимірювання ефективності певних моделей прогнозування.[6] Проте дослідження показали, що немає великої різниці між точністю прогнозів експертів, обізнаних у конфліктній ситуації, і

прогнозів осіб, які знали набагато менше.[7] Подібним чином експерти в деяких дослідженнях стверджують, що рольове мислення – перебування на місці інших людей для прогнозування їхніх рішень – не сприяє точності прогнозу.[8]

Важливим, хоча часто ігнорованим аспектом прогнозування, є його взаємозв'язок із плануванням. Прогнозування можна описати як передбачення того, як виглядатиме майбутнє, тоді як планування передбачає, як має виглядати майбутнє.[6] Не існує єдиного правильного методу прогнозування. Вибір методу має ґрунтуватися на ваших цілях і ваших умовах (даних тощо).[9] Хорошим місцем для пошуку методу є відвідування дерева вибору.

Прогнозування можна застосовувати в багатьох ситуаціях:

- Управління ланцюгом постачання та планування попиту споживачів. Прогнозування можна використовувати в управлінні ланцюгом постачання, щоб переконатися, що потрібний продукт знаходиться в потрібному місці в потрібний час. Точне прогнозування допоможе роздрібним торговцям зменшити надлишкові запаси і таким чином підвищити маржу прибутку. Точне прогнозування також допоможе їм задовольнити попит споживачів. Дисципліна планування попиту, яку також іноді називають прогнозуванням ланцюга поставок, охоплює як статистичне прогнозування, так і процес консенсусу. Дослідження показали, що екстраполяції є найменш точними, тоді як прогнози прибутків компаній є найбільш надійними.[11]
- Економічне прогнозування
- Прогнозування землетрусів
- Знову прогнозувати
- Енергетичне прогнозування для інтеграції відновлюваної енергетики

- Фінансування від ризику дефолту за допомогою кредитних рейтингів і кредитних балів
- Прогнозування землекористування
- Ефективність гравця та команди у спорті
- Політичне прогнозування
- Прогнозування продукції
- Прогнозування продажів
- Технологія прогнозування
- Телекомунікаційне прогнозування
- Транспортне планування та прогнозування
- Прогнозування погоди, прогнозування повеней і метеорологія

Економічне прогнозування – це процес прогнозування економіки. Прогнози можна виконувати на високому рівні агрегування — наприклад, щодо ВВП, інфляції, безробіття чи бюджетного дефіциту — або на більш дезагрегованому рівні для конкретних секторів економіки чи навіть конкретних фірм. Економічне прогнозування є заходом для визначення майбутнього процвітання моделі інвестицій і є ключовим видом діяльності в економічному аналізі. Багато установ займаються економічним прогнозуванням: національні уряди, банки та центральні банки, консультанти та організації приватного сектора, такі як аналітичні центри, компанії та міжнародні організації, такі як Міжнародний валютний фонд, Світовий банк та ОЕСР. "Consensus Economics" збирає та компілює широкий спектр прогнозів. Деякі прогнози створюються щорічно, але багато оновлюються частіше.

Економіст, як правило, розглядає ризики (тобто події чи умови, які можуть призвести до того, що результат буде відрізнятись від початкових

оцінок). Ці ризики допомагають проілюструвати процес міркування, який використовувався для отримання остаточних прогнозних цифр. Економісти зазвичай використовують коментарі разом із інструментами візуалізації даних, такими як таблиці та діаграми, щоб передати свій прогноз.[1] Під час підготовки економічних прогнозів використовувалася різноманітна інформація, щоб підвищити точність.

Усе, починаючи від макроекономічних [2], мікроекономічних [3] ринкових даних майбутнього [4], машинного навчання (штучні нейронні мережі) [5] і досліджень поведінки людей [6], використовувалося для отримання кращих прогнозів. Прогнози використовуються для різних цілей. Уряди та підприємства використовують економічні прогнози, щоб допомогти їм визначити свою стратегію, багаторічні плани та бюджети на наступний рік. Аналітики фондового ринку використовують прогнози, щоб допомогти їм оцінити вартість компанії та її акцій.

Економісти вибирають, які змінні важливі для обговорюваного матеріалу. Економісти можуть використовувати статистичний аналіз історичних даних, щоб визначити очевидні зв'язки між окремими незалежними змінними та їхній зв'язок із залежною змінною, що досліджується. Наприклад, якою мірою зміни цін на житло вплинули на чистий капітал населення в цілому в минулому? Потім цю залежність можна використовувати для прогнозування майбутнього. Тобто, якщо очікується, що ціни на житло зміняться певним чином, який вплив це матиме на майбутню чисту вартість населення? Прогнози зазвичай базуються на вибіркових даних, а не на повній генеральній сукупності, що вносить невизначеність. Економіст проводить статистичні тести та розробляє статистичні моделі (часто використовуючи регресійний аналіз), щоб визначити, які співвідношення найкраще описують або передбачають поведінку досліджуваних змінних. Історичні дані та припущення про

майбутнє застосовуються до моделі для отримання прогнозу для певних змінних.[7]

Джерела прогнозів

Глобальний масштаб

Економічний прогноз — це двічі на рік ОЕСР аналіз основних економічних тенденцій і перспектив на наступні два роки.[8] МВФ двічі на рік публікує звіт «Перспективи розвитку світової економіки», який забезпечує повне глобальне охоплення[9]. МВФ і Світовий банк також готують Регіональний економічний прогноз для різних частин світу.[10]

Існують також приватні компанії, такі як The Conference Board і Lombard Street Research, які надають глобальні економічні прогнози.[11]

Прогнози США

Бюджетне управління Конгресу США (СВО) щорічно публікує звіт під назвою «Бюджет і економічні перспективи», який в основному охоплює наступний десятирічний період.[12] Члени Ради керуючих Федеральної резервної системи США також виступають з промовами, надають свідчення та випускають звіти протягом року, які висвітлюють економічні перспективи.[13][14] Регіональні федеральні резервні банки, такі як Федеральний резервний банк Сент-Луїса, також надають прогнози.[15]

Великі банки, такі як Wells Fargo і JP Morgan Chase, надають економічні звіти та інформаційні бюлетені.[16][17]

Європейські прогнози

Європейська комісія також щоквартально публікує вичерпні макроекономічні прогнози для своїх країн-членів – навесні, влітку, осені та взимку.[18]

Комбінування прогнозів

Прогнози з кількох джерел можна арифметично об'єднати, а результат часто називають консенсусним прогнозом. Приватні фірми, центральні банки та державні установи публікують великий обсяг прогнозної інформації, щоб

задовольнити високий попит на економічні прогнозні дані. Consensus Economics збирає макроекономічні прогнози, підготовлені різними прогнозистами, і публікує їх щомісяця. Журнал Economist також регулярно надає такий знімок для вузького кола країн і змінних.

Економетричні дослідження продемонстрували, що використання попередніх помилок кожного початкового прогнозу для визначення ваг, призначених кожному прогнозу під час створення об'єднаного прогнозу, призводить до складеного набору прогнозів, які, як правило, дають менші середньоквадратичні помилки порівняно з будь-яким із цих прогнозів. індивідуальні вихідні прогнози.[19] Однак було виявлено, що вхід і вихід прогнозистів може мати суттєвий вплив на ефективність традиційних комбінованих методів у реальному часі.[20] Динамічний характер поєднання прогнозів і коригування методів зважування не є нейтральним.

2.2 Методи прогнозування даних

Процес економічного прогнозування подібний до аналізу даних і дає результати в оціночних значеннях ключових економічних змінних у майбутньому. Економіст застосовує методи економетрики в процесі прогнозування. Типові кроки можуть включати:

- **Обсяг:** ключові економічні змінні та теми для коментарів прогнозу визначаються на основі потреб аудиторії прогнозу.
- **Огляд літератури:** коментарі з джерел із перспективою підсумкового рівня, таких як МВФ, ОЕСР, Федеральна резервна система США та СВО, допомагають визначити ключові економічні тенденції, проблеми та ризики. Такі коментарі також можуть допомогти прогнозисту з його власними припущеннями, а також дати їм інші прогнози для порівняння.
- **Отримати вхідні дані:** історичні дані збираються щодо ключових економічних змінних. Ці дані містяться як у друкованих, так і в електронних джерелах, таких як база даних FRED або Євростат, які дозволяють користувачам запитувати історичні значення змінних, що цікавлять.
- **Визначення історичних зв'язків:** історичні дані використовуються для визначення зв'язків між однією чи декількома незалежними змінними та досліджуваною залежною змінною, часто за допомогою регресійного аналізу.
- **Модель:** історичні дані та припущення використовуються для розробки економетричної моделі. Моделі зазвичай застосовують обчислення до серії вхідних даних для створення економічного прогнозу для однієї або кількох змінних.

- Звіт: результати моделі включені у звіти, які зазвичай включають інформаційну графіку та коментарі, щоб допомогти читачеві зрозуміти прогноз.

Прогнозисти можуть використовувати обчислювальні моделі загальної рівноваги або динамічні стохастичні моделі загальної рівноваги. Останні часто використовуються центральними банками.

Методи прогнозування включають економетричні моделі, консенсус-прогнози, аналіз економічної бази, аналіз частки зсуву, модель витрат-виходу та модель Грінольда та Кронера. Див. також Прогнозування землекористування, Прогнозування еталонного класу, Транспортне планування та Розрахунок точності прогнозу попиту.

Світовий банк надає окремим особам і організаціям засоби для проведення власного моделювання та прогнозів за допомогою платформи iSimulate.[21]

Питання прогнозування

Точність прогнозу

Існує багато досліджень на тему точності прогнозів. Точність є одним з головних, якщо не основним, критерієм оцінки якості прогнозу. Деякі з наведених нижче посилань стосуються академічних досліджень точності прогнозів. Ефективність прогнозування залежить від часу, коли деякі зовнішні події впливають на якість прогнозу. Оскільки експертні прогнози, як правило, кращі за ринкові прогнози, ефективність прогнозу залежить від кількох факторів: моделі, політичної економії (тероризм), фінансової стабільності тощо.

На початку 2014 року ОЕСР провела самоаналіз своїх прогнозів[22]. «ОЕСР також виявила, що це було надто оптимістично для країн, які були найбільш відкритими для торгівлі та зовнішнього фінансування, які мали найжорсткіше регульовані ринки та слабкі банківські системи» згідно з Financial Times.[23]

В останні роки дослідження показали, що поведінкові упередження відіграють значну роль у впливі на точність прогнозів. Освіта та досвід роботи прогнозистів впливають на точність і сміливість їхніх прогнозів.[24] На точність прогнозування також впливає досвід прогнозіста з високими темпами інфляції.[25] Крім того, було показано, що політичні події, такі як тероризм, впливають на точність як експертних, так і ринкових прогнозів щодо інфляції та обмінних курсів.[26] Це підкреслює ряд зовнішніх факторів і упереджень, які слід враховувати під час оцінки точності прогнозів і прийняття обґрунтованих рішень.

Прогнози і Велика рецесія

Фінансово-економічна криза, яка вибухнула в 2007 році — можливо, найгірша з часів Великої депресії 1930-х років — не була передбачена більшістю прогнозистів, хоча багато аналітиків передбачали її протягом певного часу (наприклад, Стівен Роуч, Мередіт Вітні, Гері Шиллінг, Пітер Шифф, Марк Фабер, Нуріель Рубіні, Брукслі Борн і Роберт Шиллер).[27] Неспроможність більшості з них спрогнозувати «Велику рецесію» викликала пошук душі в професії. Сама королева Великої Британії Єлизавета запитала, чому «ніхто» не помітив, що кредитна криза наближається, і група економістів — експертів із бізнесу, міста, регуляторів, наукових кіл та уряду — спробувала пояснити це в листі.[28]

Це було не просто прогнозування Великої рецесії, а й прогнозування її впливу, коли було зрозуміло, що економісти стикаються з труднощами.

Наприклад, у Сінгапурі Сіті стверджувала, що країна переживе «найсерйознішу рецесію в історії Сінгапуру». Економіка зросла в 2009 році на 3,1%, а в 2010 році темпи зростання країни склали 15,2%.[29][30] Подібним чином Нуріель Рубіні передбачив у січні 2009 року, що ціна на нафту залишатиметься нижче 40 доларів протягом усього 2009 року. Однак до кінця 2009 року ціни на нафту становили 80 доларів США [31][32]. У березні 2009 року він передбачив, що S&P 500 цього року впаде нижче 600 і,

можливо, різко впаде до 200.[31][33] Він закотився на рівні понад 1115, що на 24% більше, ніж найбільше зростання за один рік з 2003 року.[34] У 2009 році він також передбачив, що уряд США візьме під контроль і націоналізує ряд великих банків; цього не сталося.[35][36] У жовтні 2009 року він передбачив, що золото «може перевищувати 1000 доларів, але воно не може піднятися на 20-30%»; він помилився, оскільки ціна на золото зросла протягом наступних 18 місяців, подолавши бар'єр у 1000 доларів до понад 1400 доларів. [36] Хоча в травні 2010 року він передбачив падіння фондового ринку на 20%, S&P фактично зріс приблизно на 20% протягом наступного року (навіть без урахування доходів від дивідендів).[37]

2.3 Якісні та кількісні методи

Методики якісного прогнозування є суб'єктивними, заснованими на думці та судженні споживачів і експертів; вони доречні, коли минулі дані недоступні. Зазвичай вони застосовуються до рішень середньої та довгострокової дії. Прикладами якісних методів прогнозування є [потрібне цитування] обгрунтована думка та судження, метод Дельфі, дослідження ринку та історична аналогія життєвого циклу.

Моделі кількісного прогнозування використовуються для прогнозування майбутніх даних як функції минулих даних. Їх доцільно використовувати, коли доступні минулі числові дані та коли розумно припустити, що деякі закономірності в даних, як очікується, збережуться в майбутньому. Ці методи зазвичай застосовуються до рішень короткого або середнього діапазону. Прикладами кількісних методів прогнозування є попит за останній період, прості та зважені ковзні середні за N-період, просте експоненціальне згладжування, прогнозування на основі моделі процесу Пуассона[14] та мультиплікативні сезонні індекси. Попередні дослідження показують, що різні методи можуть призвести до різного рівня точності прогнозування. Наприклад, було виявлено, що нейронна мережа GMDH має кращу ефективність прогнозування, ніж класичні алгоритми прогнозування, такі як Single Exponential Smooth, Double Exponential Smooth, ARIMA та нейронна мережа зворотного поширення.[15]

Середній підхід

У цьому підході передбачення всіх майбутніх значень дорівнюють середньому значенню минулих даних. Цей підхід можна використовувати з будь-якими даними, якщо доступні попередні дані. У позначенні часового ряду:

$$\widehat{y_{T+h|T}} = \underline{y} = (y_1 + \dots + y_T) / T$$

Незважаючи на те, що тут було використано позначення часових рядів, середній підхід також можна використовувати для перехресних даних (коли ми прогнозуємо неспостережувані значення; значення, які не входять до набору даних). Тоді прогноз для неспостережуваних значень є середнім спостережуваних значень.

Наївний підхід

Наївні прогнози є найбільш економічно ефективною моделлю прогнозування та забезпечують орієнтир, з яким можна порівнювати більш складні моделі. Цей метод прогнозування підходить лише для даних часових рядів.[16] Використовуючи наївний підхід, створюються прогнози, що дорівнюють останньому спостережуваному значенню. Цей метод досить добре працює для економічних і фінансових часових рядів, які часто мають моделі, які важко надійно й точно передбачити.[16] Якщо вважається, що часовий ряд має сезонність, сезонний наївний підхід може бути більш прийнятним, якщо прогнози дорівнюють значенню минулого сезону. У позначенні часових рядів:

$$\widehat{y_{T+h|T}} = y_T$$

2.4 Реляційні методи

Деякі методи прогнозування намагаються визначити основні фактори, які можуть вплинути на змінну, яка прогнозується. Наприклад, включення інформації про кліматичні моделі може покращити здатність моделі прогнозувати продажі парасольок. Моделі прогнозування часто враховують регулярні сезонні коливання. Окрім клімату, такі коливання також можуть бути пов'язані зі святами та звичаями: наприклад, можна передбачити, що продажі футбольного одягу студентів будуть вищими під час футбольного сезону, ніж у міжсезоння.[17]

Кілька неофіційних методів, які використовуються в причинно-наслідковому прогнозуванні, не покладаються виключно на результати математичних алгоритмів, а замість цього використовують судження прогнозіста. Деякі прогнози враховують минулі зв'язки між змінними: якщо одна змінна, наприклад, була приблизно лінійно пов'язана з іншою протягом тривалого періоду часу, може бути доречним екстраполювати такий зв'язок на майбутнє, не обов'язково розуміючи причини відносини.

Причинні методи включають:

- Регресійний аналіз включає велику групу методів для прогнозування майбутніх значень змінної з використанням інформації про інші змінні. Ці методи включають як параметричні (лінійні або нелінійні), так і непараметричні методи.

У статистичному моделюванні регресійний аналіз — це набір статистичних процесів для оцінки зв'язків між залежною змінною (яку часто називають змінною «результату» або «відповіді» або «міткою» на мові машинного навчання) та однією або кількома незалежними змінними (часто називають «предикторами», «коваріатами», «пояснювальними змінними» або «особливими ознаками»). Найпоширенішою формою регресійного аналізу є лінійна регресія, у якій знаходять лінію (або більш складну лінійну комбінацію), яка найбільше відповідає даним згідно з певним математичним

критерієм. Наприклад, метод звичайних найменших квадратів обчислює унікальну лінію (або гіперплощину), яка мінімізує суму квадратів різниць між справжніми даними та цією лінією (або гіперплощиною). З конкретних математичних міркувань (див. лінійну регресію) це дозволяє досліднику оцінити умовне очікування (або середнє значення сукупності) залежної змінної, коли незалежні змінні приймають заданий набір значень. Менш поширені форми регресії використовують дещо інші процедури для оцінки альтернативних параметрів розташування (наприклад, квантильна регресія або аналіз необхідних умов [1]) або оцінки умовного очікування для ширшої колекції нелінійних моделей (наприклад, непараметрична регресія).

Регресійний аналіз в основному використовується для двох концептуально різних цілей.

По-перше, регресійний аналіз широко використовується для передбачення та прогнозування, де його використання суттєво збігається з машинним навчанням.

По-друге, у деяких ситуаціях регресійний аналіз можна використовувати для висновку про причинно-наслідкові зв'язки між незалежними та залежними змінними. Важливо, що регресії самі по собі виявляють лише зв'язки між залежною змінною та набором незалежних змінних у фіксованому наборі даних. Щоб використовувати регресії для прогнозування або для висновку про причинно-наслідкові зв'язки, відповідно, дослідник повинен ретельно обґрунтувати, чому існуючі зв'язки мають прогностичну силу для нового контексту або чому зв'язок між двома змінними має причинно-наслідкову інтерпретацію. Останнє особливо важливо, коли дослідники сподіваються оцінити причинно-наслідкові зв'язки, використовуючи дані спостережень.[2][3]

Найбільш ранньою формою регресії був метод найменших квадратів, який був опублікований Лежандром у 1805 році [4] та Гауссом у 1809 році [5]. Лежандр і Гаусс застосували цей метод до проблеми визначення за

астрономічними спостереженнями орбіт тіл навколо Сонця (переважно комет, але пізніше також нещодавно відкритих малих планет). Гаусс опублікував подальший розвиток теорії найменших квадратів у 1821 році[6], включаючи версію теореми Гаусса-Маркова.

Термін «регресія» був введений Френсісом Гальтоном у 19 столітті для опису біологічного явища. Явище полягало в тому, що зріст нащадків високих предків має тенденцію до регресії до нормального середнього (це явище також відоме як регресія до середнього).[7][8] Для Галтона регресія мала лише це біологічне значення [9] [10], але пізніше Удні Юл і Карл Пірсон розширили його роботу до більш загального статистичного контексту [11] [12]. У роботі Юла та Пірсона передбачається, що спільний розподіл відповідних і пояснювальних змінних є Гаусовим. Це припущення було послаблено Р.А. Фішер у своїх роботах 1922 і 1925 років.[13][14][15] Фішер припустив, що умовний розподіл змінної відповіді є Гаусовим, але спільний розподіл не обов'язково є таким. У цьому плані припущення Фішера ближче до формулювання Гаусса 1821 року.

У 1950-х і 1960-х роках економісти використовували електромеханічні настільні «калькулятори» для обчислення регресії. До 1970 року іноді потрібно було до 24 годин, щоб отримати результат однієї регресії.[16]

Методи регресії продовжують бути областю активних досліджень. В останні десятиліття були розроблені нові методи надійної регресії, регресії, що включає корельовані відповіді, такі як часові ряди та криві зростання, регресії, в якій предиктор (незалежна змінна) або змінні відповіді є кривими, зображеннями, графіками або іншими складними об'єктами даних, регресійні методи, що враховують різні типи відсутніх даних, непараметрична регресія, байєсівські методи регресії, регресія, в якій прогностичні змінні вимірюються з помилкою, регресія з більшою кількістю предикторних змінних, ніж спостереження, і причинний висновок з регресією.

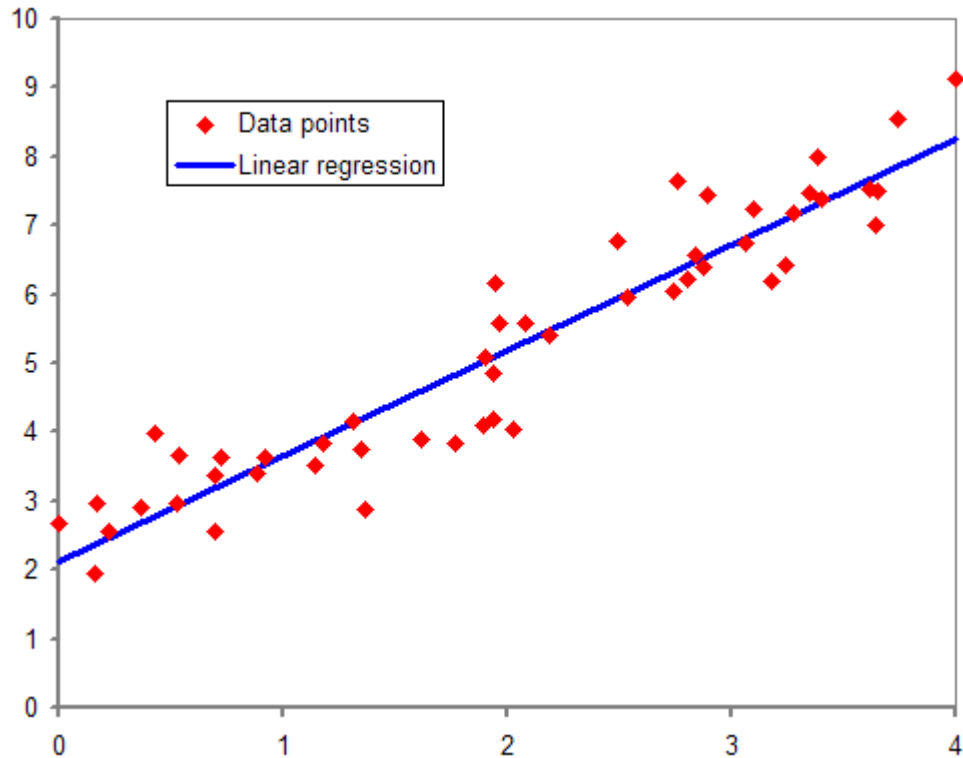


Рисунок 2.1 — Приклад графічного відображення лінійної регресії

- Авторегресійне ковзне середнє з екзогенними входами (ARMAX)[18]

Моделі кількісного прогнозування часто порівнюють одна з одною шляхом порівняння їх середньої квадратичної помилки у вибірці чи поза її межами, хоча деякі дослідники не радять цього.[19] Різні підходи до прогнозування мають різний рівень точності. Наприклад, в одному контексті було виявлено, що GMDH має вищу точність прогнозування, ніж традиційний ARIMA.[20]

У статистичному аналізі часових рядів моделі авторегресії та ковзного середнього (ARMA) забезпечують економний опис (слабко) стаціонарного стохастичного процесу в термінах двох поліномів, один для авторегресії (AR), а другий для ковзного середнього (MA). Загальна модель ARMA була описана в дисертації Пітера Уїттла 1951 року «Перевірка гіпотез в аналізі часових рядів» і була популяризована в книзі 1970 року Джорджа Е. П. Бокса та Гвіліма Дженкінса.

Дано часовий ряд даних X_t , модель ARMA є інструментом для розуміння і, можливо, прогнозування майбутніх значень у цій серії. Частина AR передбачає регресію змінної на її власні відсталі (тобто минулі) значення. Частина MA передбачає моделювання терміну помилки як лінійної комбінації термінів помилки, що виникають одночасно та в різний час у минулому. Модель зазвичай називають моделлю ARMA(p,q), де p — порядок частини AR, а q — порядок частини MA (як визначено нижче).

Моделі ARMA можна оцінити за допомогою методу Бокса–Дженкінса.

У статистиці, економетриці та обробці сигналів модель авторегресії (AR) є представленням типу випадкового процесу; як такий, він використовується для опису певних змінних у часі процесів у природі, економіці, поведінці тощо. Модель авторегресії визначає, що вихідна змінна лінійно залежить від своїх власних попередніх значень і від стохастичного терміну (недосконало передбачуваного терміну); таким чином, модель має форму стохастичного різницевого рівняння (або рекурентного співвідношення, яке не слід плутати з диференціальним рівнянням). Разом із моделлю ковзного середнього (MA) це окремий випадок і ключовий компонент більш загальної моделі авторегресії–ковзного середнього (ARMA) та авторегресійної інтегрованої ковзної середньої (ARIMA) моделей часових рядів, які мають складніший стохастичний структура; це також окремий випадок векторної авторегресійної моделі (VAR), яка складається з системи більш ніж одного пов'язаного стохастичного різницевого рівняння в більш ніж одній змінній випадковій змінній.

На відміну від моделі ковзного середнього (MA), авторегресійна модель не завжди є стаціонарною, оскільки може містити одиничний корінь.

РОЗДІЛ 3

ІНСТРУМЕНТИ ПРОГНОЗУВАННЯ СВІТОВОГО АВТОРИНКУ

3.1 Нейронні мережі

Нейронна мережа може означати або нейронну схему біологічних нейронів (іноді її також називають біологічною нейронною мережею), або мережу штучних нейронів або вузлів у випадку штучної нейронної мережі.[1] Штучні нейронні мережі використовуються для вирішення проблем штучного інтелекту (AI); вони моделюють зв'язки біологічних нейронів як ваги між вузлами. Позитивна вага відображає збудливий зв'язок, тоді як негативні значення означають гальмівні зв'язки. Усі вхідні дані модифікуються за вагою та підсумовуються. Ця діяльність називається лінійною комбінацією. Нарешті, функція активації контролює амплітуду вихідного сигналу. Наприклад, прийнятний діапазон вихідних даних зазвичай становить від 0 до 1, або він може бути від -1 до 1 .

Ці штучні мережі можна використовувати для прогнозного моделювання, адаптивного керування та додатків, де їх можна навчати за допомогою набору даних. Самонавчання в результаті досвіду може відбуватися в мережах, які можуть робити висновки зі складного та, здавалося б, непов'язаного набору інформації.[2]

Штучні нейронні мережі (ШНМ), які зазвичай просто називають нейронними мережами (НМ) або нейронними мережами [1], є

обчислювальними системами, натхненними біологічними нейронними мережами, які складають мозок тварин.[2]

ШНМ базується на сукупності з'єднаних одиниць або вузлів, званих штучними нейронами, які приблизно моделюють нейрони біологічного мозку. Кожне з'єднання, як синапси в біологічному мозку, може передавати сигнал до інших нейронів. Штучний нейрон отримує сигнали, потім обробляє їх і може сигналізувати підключеним до нього нейронам. «Сигнал» у з'єднанні є дійсним числом, а вихід кожного нейрона обчислюється деякою нелінійною функцією суми його входів. З'єднання називаються ребрами. Нейрони та ребра зазвичай мають вагу, яка коригується в міру навчання. Вага збільшує або зменшує силу сигналу під час підключення. Нейрони можуть мати такий поріг, що сигнал надсилається лише тоді, коли сукупний сигнал перетинає цей поріг.

Як правило, нейрони агреговані в шари. Різні шари можуть виконувати різні перетворення своїх вхідних даних. Сигнали проходять від першого шару (вхідний рівень) до останнього (вихідний рівень), можливо, після проходження шарів кілька разів.

Навчання

Нейронні мережі навчаються (або навчаються) шляхом обробки прикладів, кожен з яких містить відомі «вхідні дані» та «результат», утворюючи зважені за ймовірністю асоціації між ними, які зберігаються в структурі даних самої мережі. Навчання нейронної мережі з даного прикладу зазвичай проводиться шляхом визначення різниці між обробленим виходом мережі (часто передбаченням) і цільовим виходом. Ця різниця і є помилкою. Потім мережа коригує свої зважені асоціації відповідно до правила навчання та використовуючи це значення помилки. Послідовні коригування призведуть до того, що нейронна мережа буде виробляти результати, які все більше будуть схожі на цільовий результат. Після достатньої кількості цих

коригувань навчання можна припинити за певними критеріями. Це форма навчання під наглядом.

Такі системи «вчаться» виконувати завдання, розглядаючи приклади, як правило, без програмування з використанням правил для конкретних завдань. Наприклад, під час розпізнавання зображень вони можуть навчитися ідентифікувати зображення, на яких зображені коти, аналізуючи приклади зображень, які вручну позначені як «кіт» або «без кота», і використовуючи результати для ідентифікації котів на інших зображеннях. Вони роблять це без будь-яких попередніх знань про котів, наприклад, що у них є шерсть, хвости, вуса та котячі обличчя. Натомість вони автоматично генерують ідентифікаційні характеристики з прикладів, які вони обробляють.

Найпростішим типом прямої нейронної мережі (FNN) є лінійна мережа, яка складається з одного шару вихідних вузлів; входи подаються безпосередньо на виходи через серію ваг. Сума добутків ваг і вхідних даних обчислюється в кожному вузлі. Середньоквадратичні похибки між цими обчисленими виходами та заданими цільовими значеннями мінімізуються шляхом створення коригування ваг. Цей метод відомий понад два століття як метод найменших квадратів або лінійна регресія. Лежандр (1805) і Гаусс (1795) використовували його як засіб для знаходження хорошої грубої лінійної відповідності набору точок для передбачення руху планет.[3][4][5][6][7]

Вільгельм Ленц і Ернст Ізінг створили та проаналізували модель Ізінга (1925) [8], яка, по суті, є штучною рекурентною нейронною мережею (RNN), яка не навчається, і складається з нейроноподібних порогових елементів [6]. У 1972 році Шунічі Амарі зробив цю архітектуру адаптивною.[9][6] Його навчання RNN популяризував Джон Хопфілд у 1982 році.

Уоррен МакКаллох і Волтер Піттс [11] (1943) також розглядали ненавчальну обчислювальну модель для нейронних мереж [12]. Наприкінці 1940-х років Д. О. Хебб [13] створив гіпотезу навчання, засновану на

механізмі нервової пластичності, яка стала відомою як навчання Гебба. Фарлі та Веслі А. Кларк [14] (1954) вперше використали обчислювальні машини, які тоді називали «калькуляторами», для моделювання мережі Хебба. У 1958 році психолог Френк Розенблатт винайшов перцептрон, першу реалізовану штучну нейронну мережу [15] [16] [17] [18] за фінансової підтримки Управління військово-морських досліджень США [19].

Деякі кажуть, що дослідження зупинилися після того, як Мінські та Пейперт (1969), [20] виявили, що базові перцептрони не здатні обробляти схему виключного або, і що комп'ютерам бракує достатньої потужності для обробки корисних нейронних мереж. Однак на момент виходу цієї книги методи навчання багатошарових перцептронів (MLP) уже були відомі.

Перший MLP глибокого навчання був опублікований Олексієм Григоровичем Івахненком і Валентином Лапою в 1965 році як груповий метод обробки даних. [21][22][23] Перший MLP глибокого навчання, навчений стохастичним градієнтним спуском [24], був опублікований у 1967 році Шунічі Амарі [25] [6]. У комп'ютерних експериментах, проведених учнем Амарі Сайто, п'ятирівневий MLP з двома модифікованими шарами навчився використовувати корисні внутрішні представлення для класифікації нелінійно роздільних класів шаблонів. [6]

Самоорганізуючі карти (SOM) були описані Теуво Кохоненом у 1982 році. [26][27] SOM — це нейрофізіологічні [28] нейронні мережі, які вивчають низьковимірні представлення високовимірних даних, зберігаючи топологічну структуру даних. Вони навчаються за допомогою змагального навчання. [26]

Архітектура згорткової нейронної мережі (CNN) із згортковими шарами та шарами зменшення дискретизації була представлена Куніхіко Фукусімою в 1980 році. [29] Він назвав це неокогнітроном. У 1969 році він також представив функцію активації ReLU (випрямлена лінійна одиниця) [30] [6]. Випрямляч став найпопулярнішою функцією активації для CNN і

глибоких нейронних мереж загалом.[31] CNN стали важливим інструментом комп'ютерного зору.

Алгоритм зворотного поширення є ефективним застосуванням правила ланцюга Лейбніца (1673)[32] до мереж диференційованих вузлів.[6] Він також відомий як зворотний режим автоматичної диференціації або зворотного накопичення, завдяки Сеппо Ліннайнімаа (1970).[33][34][35][36][6] Термін «помилки зворотного поширення» був введений у 1962 році Френком Розенблатом [37] [6], але він не мав реалізації цієї процедури, хоча Генрі Дж. Келлі [38] і Брайсон [39] базувалися на динамічному програмуванні. безперервні попередники зворотного поширення [21] [40] [41] [42] вже в 1960-61 роках у контексті теорії керування [6]. У 1973 році Дрейфус використав зворотне поширення для адаптації параметрів контролерів пропорційно до градієнтів помилок.[43] У 1982 році Пол Вербос застосував зворотне розповсюдження до MLP у спосіб, який став стандартним.[44][40] У 1986 році Румельхарт, Хінтон і Вільямс показали, що зворотне розповсюдження вивчає цікаві внутрішні представлення слів як векторів ознак, коли навчається передбачати наступне слово в послідовності.[45]

Нейронна мережа із затримкою часу (TDNN) Алекса Вайбеля (1987) поєднала згортки, розподіл ваги та зворотне поширення.[46][47] У 1988 році Wei Zhang та ін. застосував зворотне поширення до CNN (спрощений Neocognitron зі згортковими взаємозв'язками між шарами ознак зображення та останнім повністю підключеним шаром) для розпізнавання алфавіту.[48][49] У 1989 році Yann LeCun та ін. навчив CNN розпізнавати рукописні поштові індекси в пошті.[50] У 1992 році Хуан Венг та ін. запровадили максимальне об'єднання для CNN. щоб допомогти з інваріантністю найменшого зсуву та толерантністю до деформації для сприяння розпізнаванню 3D-об'єктів.[51][52][53] LeNet-5 (1998), 7-рівневий CNN від Yann LeCun та ін.,[54] який класифікує цифри, був застосований

кількома банками для розпізнавання рукописних чисел на чеках, оцифрованих у зображеннях 32x32 пікселів.

З 1988 року [55] [56] використання нейронних мереж трансформувало сферу передбачення структури білка, зокрема, коли перші каскадні мережі були навчені на профілях (матрицях), створених численними вирівнюваннями послідовностей [57].

У 1980-х роках зворотне поширення не працювало добре для глибоких FNN і RNN. Щоб подолати цю проблему, Юрген Шмідхубер (1992) запропонував ієрархію RNN, попередньо навчених по одному рівню за допомогою самоконтрольованого навчання.[58] Він використовує прогнозне кодування для вивчення внутрішніх представлень у кількох самоорганізованих масштабах часу. Це може істотно полегшити подальше глибоке навчання. Ієрархію RNN можна згорнути в єдину RNN шляхом дистиляції мережі чанкера вищого рівня в мережу автоматизатора нижчого рівня.[58][6] У 1993 році чанкер вирішив завдання глибокого навчання, глибина якого перевищувала 1000.[59]

У 1992 році Юрген Шмідхубер також опублікував альтернативу RNN [60], яка зараз називається лінійним трансформатором або трансформатором з лінеаризованим самоувагою [61] [62] [6] (за винятком оператора нормалізації). Він вивчає внутрішні прожектори уваги: [63] повільна прямонаправлена нейронна мережа вчиться за допомогою градієнтного спуску контролювати швидкі ваги іншої нейронної мережі через зовнішні продукти самогенерованих шаблонів активації FROM і TO (які тепер називаються ключем і значенням для себе). -увага).[61] Це швидке відображення ваги уваги застосовується до шаблону запиту.

Сучасний трансформер був представлений Ашишем Васвані та ін. al. у своїй статті 2017 року «Увага — це все, що вам потрібно». [64] Він поєднує це з оператором softmax і матрицею проєкції [6]. Трансформери все частіше стають моделлю вибору для обробки природної мови.[65] Багато сучасних

великих мовних моделей, таких як ChatGPT, GPT-4 і BERT, використовують його. Трансформатори також все частіше використовуються в комп'ютерному зорі.[66]

У 1991 році Юрген Шмідгубер також опублікував змагальні нейронні мережі, які змагаються одна з одною у формі гри з нульовою сумою, де виграш однієї мережі є програшем іншої.[67][68][69] Перша мережа є генеративною моделлю, яка моделює розподіл ймовірностей за моделями вихідних даних. Друга мережа вчиться шляхом градієнтного спуску передбачати реакцію середовища на ці шаблони. Це було названо «штучною цікавістю».

У 2014 році цей принцип був використаний у генеративній змагальній мережі (GAN) Ієном Гудфеллоу та іншими.[70] Тут реакція навколишнього середовища дорівнює 1 або 0 залежно від того, чи є перший вихід мережі в даному наборі. Це можна використовувати для створення реалістичних дипфейків.[71] Чудова якість зображення досягається за допомогою StyleGAN (2018) Nvidia [72] на основі прогресивного GAN Тєро Карраса, Тімо Айли, Самулі Лайне та Яакко Лехтїнена [73]. Тут генератор GAN вирощується від малого до великого у пірамідальний спосіб.

Дипломна робота Зеппа Гохрейтера (1991) [74] була названа «одним із найважливіших документів в історії машинного навчання» його керівником Юргеном Шмідгубером [6]. Гохрайтер визначив і проаналізував проблему зникаючого градієнта[74][75] і запропонував повторювані залишкові зв'язки для її вирішення. Це призвело до появи методу глибокого навчання під назвою довготривала короткочасна пам'ять (LSTM), опублікованого в Neural Computation (1997).[76] Рекурентні нейронні мережі LSTM можуть вивчати завдання «дуже глибокого навчання»[77] з довгими шляхами розподілу кредитів, які вимагають спогадів про події, які відбулися за тисячі дискретних часових кроків до цього. "Ванільний LSTM" із забутими воротами був представлений у 1999 році Феліксом Герсом, Шмідхубером та

Фредом Каммінсом. LSTM стала найбільш цитованою нейронною мережею 20 століття.[6] У 2015 році Рупеш Кумар Срівастава, Клаус Грефф і Шмідхубер використали принцип LSTM для створення мережі Highway, прямої нейронної мережі із сотнями шарів, набагато глибшої, ніж попередні мережі.[79][80] 7 місяців потому, Каймін Хе, Сян'ю Чжан; Shaoqing Ren і Jian Sun виграли конкурс ImageNet 2015 із варіантом мережі Highway із відкритим або без воріт під назвою Residual neuro network [81]. Це стала найбільш цитованою нейронною мережею 21 століття.[6]

Розвиток дуже великомасштабної інтеграції (VLSI) метал-оксид-напівпровідник (MOS) у формі додаткової технології MOS (CMOS) дозволив збільшити кількість MOS-транзисторів у цифровій електроніці. Це забезпечило більше процесорної потужності для розробки практичних штучних нейронних мереж у 1980-х роках.[82]

Ранні успіхи нейронних мереж включали прогнозування фондового ринку, а в 1995 році (здебільшого) безпілотний автомобіль.[a][83]

Джеффри Хінтон та ін. (2006) запропонував вивчати представлення високого рівня, використовуючи послідовні шари двійкових або реальних прихованих змінних з обмеженою машиною Больцмана [84] для моделювання кожного шару. У 2012 році Нг і Дін створили мережу, яка навчилася розпізнавати концепти вищого рівня, такі як коти, лише дивлячись зображення без міток.[85] Попереднє навчання без нагляду та збільшення обчислювальної потужності графічних процесорів і розподілених обчислень дозволили використовувати більші мережі, зокрема в проблемах розпізнавання зображень і візуального зображення, що стало відомим як «глибоке навчання».[86]

Ciresan та його колеги (2010) [87] показали, що, незважаючи на проблему зникнення градієнта, графічні процесори роблять зворотне поширення можливим для багаторівневих нейронних мереж прямого зв'язку [88]. У період між 2009 і 2012 роками ШНМ почали вигравати нагороди в

змаганнях з розпізнавання зображень, наближаючись до людського рівня виконання різних завдань, спочатку в розпізнаванні образів і розпізнаванні рукописного тексту.[89][90] Наприклад, двонаправлена та багатовимірною довготривала короткочасна пам'ять (LSTM)[91][92] Graves et al. виграв три конкурси з розпізнавання рукописного тексту в 2009 році, не маючи жодних попередніх знань про три мови, які потрібно вивчати.[91][92]

Сірсан і його колеги створили перші пристрої розпізнавання шаблонів для досягнення конкурентоспроможних/надлюдських показників [93] на таких тестах, як розпізнавання дорожніх знаків (IJCNN 2012).

ШНМ почалися як спроба використати архітектуру людського мозку для виконання завдань, у яких звичайні алгоритми мали невеликий успіх. Незабаром вони переорієнтувалися на покращення емпіричних результатів, відмовившись від спроб залишатися вірними своїм біологічним попередникам. ШНМ мають здатність вивчати та моделювати нелінійності та складні зв'язки. Це досягається тим, що нейрони з'єднуються різними схемами, що дозволяє вихідним сигналам одних нейронів стати входом інших. Мережа утворює орієнтований, зважений граф.[94]

Штучна нейронна мережа складається з імітованих нейронів. Кожен нейрон з'єднаний з іншими вузлами за допомогою зв'язків, як біологічне з'єднання аксон-синапс-дендрит. Усі вузли, з'єднані зв'язками, отримують деякі дані та використовують їх для виконання певних операцій і завдань над даними. Кожна ланка має вагу, що визначає силу впливу одного вузла на інший [95], дозволяючи вагам вибирати сигнал між нейронами.

Штучні нейрони

ШНМ складаються зі штучних нейронів, які концептуально походять від біологічних нейронів. Кожен штучний нейрон має вхідні дані та видає один вихід, який можна надіслати багатьом іншим нейронам.[96] Вхідні дані можуть бути значеннями ознак вибірки зовнішніх даних, наприклад зображень чи документів, або вони можуть бути виходами інших нейронів.

Вихідні дані кінцевих вихідних нейронів нейронної мережі виконують завдання, наприклад, розпізнають об'єкт на зображенні.

Щоб знайти вихід нейрона, ми беремо зважену суму всіх входів, зважених за вагами з'єднань від входів до нейрона. Ми додаємо зміщення до цієї суми.[97] Ця зважена сума іноді називається активацією. Ця зважена сума потім пропускається через (зазвичай нелінійну) функцію активації для отримання результату. Початковими входними даними є зовнішні дані, наприклад зображення та документи. Кінцеві результати виконують завдання, наприклад, розпізнають об'єкт на зображенні.[98]

Організація

Нейрони зазвичай організовані в кілька шарів, особливо при глибокому навчанні. Нейрони одного шару з'єднуються тільки з нейронами безпосередньо попереднього і наступного шарів. Рівень, який отримує зовнішні дані, є входним. Шар, який дає кінцевий результат, є вихідним шаром. Між ними є нуль або більше прихованих шарів. Використовуються також одношарові та нешарові мережі. Між двома шарами можливі кілька шаблонів з'єднання. Вони можуть бути «повністю з'єднані», коли кожен нейрон одного шару з'єднується з кожним нейроном наступного шару. Вони можуть бути об'єднаними, коли група нейронів одного шару з'єднується з одним нейроном наступного шару, таким чином зменшуючи кількість нейронів у цьому шарі.[99] Нейрони лише з такими зв'язками утворюють орієнтований ациклічний граф і відомі як мережі прямого зв'язку.[100] Крім того, мережі, які дозволяють з'єднання між нейронами на тому самому або попередніх рівнях, відомі як рекурентні мережі.[101]

Навчання

Навчання — це адаптація мережі для кращого виконання завдання шляхом врахування вибіркового спостережень. Навчання передбачає коригування ваг (і необов'язкових порогів) мережі для підвищення точності результату. Це досягається шляхом мінімізації виявлених помилок. Навчання

є завершеним, якщо перевірка додаткових спостережень не зменшує коефіцієнт помилок. Навіть після навчання частота помилок зазвичай не досягає 0. Якщо після навчання частота помилок надто висока, мережу зазвичай потрібно перепроектувати. Практично це робиться шляхом визначення функції витрат, яка періодично оцінюється під час навчання. Поки його продуктивність продовжує знижуватися, навчання триває. Вартість часто визначається як статистична величина, значення якої можна лише приблизно визначити. Вихідні дані насправді є числами, тому, коли помилка низька, різниця між результатом (майже напевно кіт) і правильною відповіддю (кіт) невелика. Навчання намагається зменшити загальну кількість відмінностей між спостереженнями. Більшість моделей навчання можна розглядати як пряме застосування теорії оптимізації та статистичної оцінки.[94][103]

Швидкість навчання визначає розмір коригувальних кроків, які робить модель для коригування помилок у кожному спостереженні.[104] Висока швидкість навчання скорочує час навчання, але з меншою кінцевою точністю, тоді як нижча швидкість навчання займає більше часу, але з потенціалом для більшої точності. Такі оптимізації, як Quickprop, насамперед спрямовані на прискорення мінімізації помилок, тоді як інші вдосконалення в основному намагаються підвищити надійність. Щоб уникнути коливань всередині мережі, таких як чергування ваг з'єднання, і покращити швидкість конвергенції, удосконалення використовують адаптивну швидкість навчання, яка відповідно збільшується або зменшується.[105] Концепція імпульсу дозволяє зважити баланс між градієнтом і попередньою зміною так, щоб коригування ваги певною мірою залежало від попередньої зміни. Імпульс, близький до 0, підкреслює градієнт, тоді як значення, близьке до 1, підкреслює останню зміну.

Хоча можна визначити функцію вартості *ad hoc*, часто вибір визначається бажаними властивостями функції (такими як опуклість) або

тому, що він виникає з моделі (наприклад, у імовірнісній моделі апостериорна ймовірність моделі може використовуватися як зворотна вартість).

3.2 Інструментальні засоби розробки нейронних мереж

Python часто порівнюють з іншими інтерпретованими мовами, такими як Java, JavaScript, Perl, Tcl або Smalltalk. Порівняння з C++, Common Lisp і Scheme також можуть бути повчальними. У цьому розділі я коротко порівнюю Python з кожною з цих мов. Ці порівняння зосереджені лише на мовних питаннях. На практиці вибір мови програмування часто диктується іншими обмеженнями реального світу, такими як вартість, доступність, навчання та попередні інвестиції чи навіть емоційна прив'язаність. Оскільки ці аспекти дуже різноманітні, розглядати їх для цього порівняння здається марною тратою часу.

Java

Зазвичай очікується, що програми на Python працюватимуть повільніше, ніж програми на Java, але їх розробка потребує набагато менше часу. Програми на Python зазвичай у 3-5 разів коротші за еквівалентні програми на Java. Цю різницю можна пояснити вбудованими високорівневими типами даних Python і його динамічною типізацією. Наприклад, програміст на Python не витрачає час на оголошення типів аргументів або змінних, а потужні поліморфні типи списків і словників Python, для яких розширена синтаксична підтримка вбудована прямо в мову, знаходять застосування майже в кожній програмі на Python. Через типізацію під час виконання, час виконання Python має працювати більше, ніж Java. Наприклад, коли обчислюється вираз $a+b$, він повинен спочатку перевірити об'єкти a і b , щоб дізнатися їхній тип, який невідомий під час компіляції. Потім він викликає відповідну операцію додавання, яка може бути перевантаженим методом, визначеним користувачем. Java, з іншого боку, може виконувати ефективно додавання цілих чисел або з плаваючою комою, але вимагає оголошення змінних для a і b і не дозволяє перевантажувати оператор $+$ для екземплярів визначених користувачем класів.

З цих причин Python набагато краще підходить як «склеюча» мова, тоді як Java краще характеризується як мова низькорівневої реалізації. Фактично, обидва разом утворюють чудове поєднання. Компоненти можна розробляти на Java та комбінувати для створення програм на Python; Python також можна використовувати для створення прототипів компонентів, доки їх дизайн не буде «зміцнений» у реалізації Java. Для підтримки цього типу розробки розробляється реалізація Python, написана на Java, яка дозволяє викликати код Python з Java і навпаки. У цій реалізації вихідний код Python перекладається на байт-код Java (за допомогою бібліотеки часу виконання для підтримки динамічної семантики Python).

Javascript

«Об'єктна» підмножина Python приблизно еквівалентна JavaScript. Як і JavaScript (і на відміну від Java), Python підтримує стиль програмування, який використовує прості функції та змінні без участі у визначеннях класів. Однак для JavaScript це все, що є. Python, з іншого боку, підтримує написання набагато більших програм і краще повторне використання коду завдяки справжньому об'єктно-орієнтованому стилю програмування, де класи та успадкування відіграють важливу роль.

Perl

Python і Perl походять із схожого фону (скрипти Unix, які обидва давно переросли), і мають багато подібних функцій, але мають різну філософію. Perl наголошує на підтримці звичайних програмно-орієнтованих завдань, напр. завдяки вбудованим регулярним виразам, функціям сканування файлів і створення звітів. Python наголошує на підтримці загальних методологій програмування, таких як проектування структури даних та об'єктно-орієнтоване програмування, і заохочує програмістів писати читабельний (і, отже, підтримуваний) код, надаючи елегантну, але не надто загадкову нотацію. Як наслідок, Python наближається до Perl, але рідко перемагає його

в початковій області застосування; однак Python має застосовність далеко за межами ніші Perl.

Tcl

Як і Python, Tcl можна використовувати як мову розширення програми, а також як окрему мову програмування. Однак Tcl, який традиційно зберігає всі дані у вигляді рядків, слабкий щодо структур даних і виконує типовий код набагато повільніше, ніж Python. Tcl також не має функцій, необхідних для написання великих програм, таких як модульні простори імен. Таким чином, хоча «типова» велика програма, що використовує Tcl, зазвичай містить розширення Tcl, написані на C або C++, які є специфічними для цієї програми, еквівалентну програму Python часто можна написати на «чистому Python». Звичайно, розробка на чистому Python набагато швидша, ніж написання та налагодження компонента C або C++. Було сказано, що однією з переваг Tcl є інструментарій Tk. Python прийняв інтерфейс Tk як стандартну бібліотеку компонентів GUI.

Tcl 8.0 вирішує проблему швидкості, надаючи компілятор байт-коду з обмеженою підтримкою типів даних і додаючи простори імен. Однак це все ще набагато більш громіздка мова програмування.

Smalltalk

Можливо, найбільша відмінність між Python і Smalltalk полягає в більш «масовому» синтаксисі Python, який дає йому перевагу в підготовці програмістів. Як і Smalltalk, Python має динамічний тип і зв'язування, і все в Python є об'єктом. Однак Python відрізняє вбудовані типи об'єктів від визначених користувачем класів і наразі не дозволяє успадковувати вбудовані типи. Стандартна бібліотека типів даних колекції Smalltalk є більш досконалою, тоді як бібліотека Python має більше можливостей для роботи з реальністю Інтернету та WWW, як-от електронна пошта, HTML та FTP.

Python має іншу філософію щодо середовища розробки та розповсюдження коду. Там, де Smalltalk традиційно має монолітний «образ

системи», який включає як середовище, так і програму користувача, Python зберігає як стандартні модулі, так і модулі користувача в окремих файлах, які можна легко змінити або розповсюдити за межами системи. Одним із наслідків є те, що існує більше ніж один варіант для приєднання графічного інтерфейсу користувача (GUI) до програми на Python, оскільки GUI не вбудовано в систему.

C++

Майже все, що сказано для Java, також стосується C++, навіть більше того: коли код Python зазвичай у 3-5 разів коротший за еквівалентний код Java, він часто в 5-10 разів коротший за еквівалентний код C++! Анекдотичні дані свідчать про те, що один програміст на Python може закінчити за два місяці те, що два програмісти на C++ не можуть виконати за рік. Python блищить як зв'язуюча мова, яка використовується для об'єднання компонентів, написаних мовою C++.

Common Lisp і Scheme

Ці мови близькі до Python за своєю динамічною семантикою, але настільки різні за своїм підходом до синтаксису, що порівняння стає майже релігійним аргументом: відсутність синтаксису Lisp є перевагою чи недоліком? Слід зазначити, що Python має інтроспективні можливості, подібні до можливостей Lisp, і програми Python можуть створювати та виконувати фрагменти програми на льоту. Зазвичай властивості реального світу є вирішальними: Common Lisp є великим (в усіх сенсах), а світ Scheme фрагментований між багатьма несумісними версіями, де Python має єдину, безкоштовну, компактну реалізацію.

Отже, можна зробити висновок, що для поставлених задач Python є найкращим з можливих рішень.

Python — це широко використовувана, інтерпретована, об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою, яка використовується для програмування загального призначення. Це скрізь,

і люди щодня використовують численні пристрої на базі Python, усвідомлюють вони це чи ні.

Python був створений Гвідо ван Россумом і вперше випущений 20 лютого 1991 року. Хоча ви знаєте пітона як велику змію, назва мови програмування Python походить від старого комедійного серіалу BBC під назвою «Летючий цирк Монті Пайтона».

Однією з дивовижних особливостей Python є те, що це фактично робота однієї людини. Зазвичай нові мови програмування розробляють і публікують великі компанії, в яких працює багато професіоналів, і через правила авторського права дуже важко назвати когось із людей, які беруть участь у проекті. Python є винятком.

Звичайно, Гвідо ван Россум не розробляв і розвивав усі компоненти Python самостійно. Швидкість, з якою Python поширився по всьому світу, є результатом безперервної роботи тисяч (часто анонімних) програмістів, тестувальників, користувачів (багато з них не IT-спеціалісти) та ентузіастів, але слід сказати, що дуже Перша ідея (зерно, з якого пророс Python) прийшла до однієї голови – Гвідо.

Python підтримується Python Software Foundation, некомерційною членською організацією та спільнотою, яка займається розробкою, вдосконаленням, розширенням і популяризацією мови Python та її середовища.

Python є всюдисущим, і люди щодня використовують численні пристрої на платформі Python, усвідомлюють вони це чи ні. На Python написані мільярди рядків коду, що означає майже необмежені можливості для повторного використання коду та навчання на добре розроблених прикладах. Більше того, існує велика й дуже активна спільнота Python, яка завжди рада допомогти.

Існує також кілька факторів, які роблять Python чудовим для вивчення:

- Його легко вивчити – час, необхідний для вивчення Python, менший, ніж для багатьох інших мов; це означає, що можна швидше розпочати фактичне програмування;
- Він простий у використанні для написання нового програмного забезпечення – часто можна писати код швидше, використовуючи Python;
- Його легко отримати, встановити та розгорнути – Python є безкоштовним, відкритим і багатоплатформним; не всі мови можуть цим похвалитися.

Сильні сторони Python:

Легко читати та писати: Python має простий та зрозумілий синтаксис, який легко читати та писати, що робить його популярним вибором як для початківців, так і для досвідчених програмістів. Код, написаний на Python, часто є лаконічним і легким для розуміння, що сприяє швидшій розробці та зменшує зусилля з обслуговування.

Велика стандартна бібліотека: Python постачається з великою стандартною бібліотекою, яка надає широкий спектр модулів для таких завдань, як файловий ввід/вивід, регулярні вирази, маніпулювання даними, мережа тощо. Це робить Python універсальною мовою та дозволяє розробникам використовувати існуючий код, що може значно пришвидшити розробку.

Багата екосистема сторонніх бібліотек: Python має величезну екосистему сторонніх бібліотек, які пропонують додаткові функції майже для будь-якої області застосування. Такі бібліотеки, як NumPy для чисельних обчислень, Pandas для аналізу даних, Django для веб-розробки та TensorFlow для машинного навчання, широко використовуються та сприяють популярності Python серед розробників.

Кросплатформна сумісність: Python є кросплатформною мовою, що означає, що код Python, написаний в одній операційній системі, часто можна запускати в іншій без змін. Це робить Python придатним для розробки програм, які повинні працювати на різних платформах, включаючи Windows, macOS, Linux тощо.

Сильна спільнота та підтримка: Python має велике й активне співтовариство розробників, користувачів і учасників, які надають широкую підтримку через форуми, списки розсилки, онлайн-ресурси та конференції. Ця активна спільнота сприяє розвитку та вдосконаленню Python, постійно розробляючи нові бібліотеки, інструменти та фреймворки, а також надаючи цінні відгуки розробникам мови.

Слабкі сторони Python:

Продуктивність: Python — це інтерпретована мова, яка може бути повільнішою порівняно з скомпільованими мовами, такими як C++ або Java, особливо для завдань, що потребують інтенсивного обчислення. Хоча Python пропонує такі бібліотеки, як NumPy, для чисельних обчислень, які оптимізують продуктивність, він може бути не найкращим вибором для програм, які вимагають високої продуктивності та оптимізації низького рівня.

Глобальне блокування інтерпретатора (GIL): Глобальне блокування інтерпретатора (GIL) Python є обмеженням, яке обмежує можливість досягнення справжнього паралелізму в багатоядерних системах. Це означає, що навіть якщо Python підтримує багатопотоковість, він може не забезпечити оптимальної продуктивності в ситуаціях, коли паралельна обробка є критичною.

Споживання пам'яті: Python може споживати більше пам'яті порівняно з іншими мовами завдяки динамічному набору тексту та механізмам збирання сміття. Це може не підійти для програм із жорсткими обмеженнями

пам'яті або для розробки програмного забезпечення, яке має працювати на пристроях з обмеженими ресурсами.

Мобільні та вбудовані системи: Python може бути не найкращим вибором для розробки програм для мобільних пристроїв або вбудованих систем, де критичними факторами є пам'ять, обчислювальна потужність і час автономної роботи. Накладні витрати інтерпретатора Python і його великий обсяг пам'яті можуть обмежити його використання в таких середовищах.

Упаковка та розгортання: Упаковка та розповсюдження програм Python іноді може бути складним завданням, особливо коли ви маєте справу зі складними залежностями або розповсюджуєте програми як окремі виконувані файли. Управління залежностями, забезпечення сумісності та розгортання програм Python може потребувати додаткових зусиль і може бути не таким простим, як у деяких інших мовах.

Підсумовуючи, Python є популярною та потужною мовою програмування з численними сильними сторонами, такими як простота, велика стандартна бібліотека, розгалужена екосистема сторонніх бібліотек, сумісність між платформами та сильна спільнота. Однак у нього також є слабкі сторони щодо продуктивності, глобального блокування інтерпретатора (GIL), споживання пам'яті, придатності для мобільних/вбудованих систем і проблем з пакуванням/розгортанням.

РОЗДІЛ 4

РОЗРОБКА СИСТЕМИ ПРОГНОЗУВАННЯ СВІТОВОГО АВТОРИНКУ

4.1 Вимоги до технічного і програмного забезпечення

Розроблюване програмне забезпечення є легким для запуску і не потребує особливих вимог, проте є перелік обов'язкових умов, які необхідно виконати для нормального функціонування додатку:

- операційна система Windows XP і вище;
- інтерпретатор Python 3.7 або вище;
- встановлені бібліотеки, такі як:
 - pandas;
 - numpy;
 - sklearn;
 - matplotlib;
- 2 ГБ і більше оперативної пам'яті.

4.2 Первинне тренування моделі

Первинним етапом створення нейронної мережі є створення відповідних датасетів для тренування. Нижче показано механізм викоремлення необхідних даних з загального датасету.

```
df = pd.read_csv("dataset.csv")
x = df.iloc[:, :4]
y = df.iloc[:, 4:5]
```

Після створення датасетів необхідно створити і навчити моделі (в даному випадку створено 3 моделі: без додавання вагів, з додаванням і модель, що піддавалася редукції). Процес створення і навчання перших двох моделей показано нижче.

```
model_1 = LinearRegression()
model_1.fit(x.values, y.values)
pred(model_1, 'Non-weighted model')

sample_weight = np.ones(36) * 20
model_2 = LinearRegression()
model_2.fit(x.values, y.values, sample_weight)
pred(model_2, 'Weighted model')
```

Для кофмортного повторного використання програмного коду функціонал створення прогнозу і підрахунку середньоквадратичної помилки було винесено в окрему функцію.

```
def pred(model, title):
    Xp = [26.6100, 26.5722, 26.4244, 26.3503]
    y_pred = [26.3314]
    Xp = np.reshape(Xp, (1, -1))
    predictions = model.predict(Xp)
    print('\n', title)
    print('Prediction data: [26.6100, 26.5722, 26.4244, 26.3503]')
```

```
print('Needed result: [26.3314]')  
print('Result: ', predictions)  
print('MSE: %.8f' % mean_squared_error(y_pred, predictions))  
print('Weights', model.coef_)
```

4.3 Редукція моделі

Для підвищення точності моделі було вирішено провести редукцію моделі.

Редукція нейронної мережі відноситься до процесу спрощення або скорочення структури моделі з метою зменшення її обчислювальних витрат та розміру. Цей підхід може бути корисним у різних випадках, наприклад, для покращення швидкості навчання, зниження кількості параметрів моделі, прискорення передачі даних або покращення узагальнюючої здатності.

Існує кілька методів редукції нейронних мереж, які можуть застосовуватись в залежності від конкретної задачі та вимог. Одним із найпоширеніших методів є прунінг (відкидання), який полягає у видаленні зайвих нейронів або зв'язків з моделі, що мають незначний вплив на результати. Прунінг може бути глобальним, коли він проводиться після тренування всієї моделі, або локальним, коли прунінг виконується під час процесу навчання.

Ще одним методом редукції є квантизація, яка полягає у зменшенні кількості бітів, використовуваних для представлення ваг та активацій моделі. Замість використання плаваючої точки можуть використовуватись цілі числа або обмежені розрядності, що дозволяє зменшити обчислювальні витрати.

Також існують методи компресії моделі, такі як використання спеціальних архітектур, які дозволяють спільно використовувати пам'ять та обчислювальні ресурси для кількох нейронів, або використання методів приближеного обчислення, які зменшують точність обчислень на користь швидкодії.

Застосування редукції нейронної мережі може допомогти знизити обчислювальні вимоги моделі, зменшити її розмір для ефективного розгортання на пристроях з обмеженими ресурсами або покращити продуктивність та швидкодію моделі без втрати якості результатів.

Для проведення редукції нейронної мережі було проведено процедуру перевірки впливу кожного нейронного зв'язку на загальну середньоквадратичну помилку і видалено зв'язки, які негативно впливали на результат. Процедура редукції зображена нижче.

```
def reduction mdl, sample_weights):
    model = mdl
    Xp = [26.6100, 26.5722, 26.4244, 26.3503]
    y_pred = [26.3314]
    Xp = np.reshape(Xp, (1, -1))
    predictions = model.predict(Xp)
    etalon_MSE = mean_squared_error(y_pred, predictions)
    current_mse = mean_squared_error(y_pred, predictions)
    for i in range(36):
        temp = sample_weights[i]
        sample_weights[i] = 0
        model.fit(x.values, y.values, sample_weight)
        prediction_test = model.predict(Xp)
        temp_mse = mean_squared_error(y_pred, prediction_test)
        if temp_mse > current_mse:
            sample_weights[i] = temp
        else:
            current_mse = temp_mse
    return model
```

4.4 Прогнозування і відображення результатів

Для створення графічного відображення отриманих результатів створено дві окремі функції, які відповідають за:

- створення графіків порівняння результатів прогнозів:

```
def do_charts_values(model_unr, model_r):
    Xp = [[26.6100, 26.5722, 26.4244, 26.3503],
          [27.9783, 28.0087, 28.0096, 28.0576],
          [26.6931, 26.6457, 26.6752, 26.6455],
          [26.9290, 26.8601, 26.8928, 27.0029],
          [26.9298, 26.8102, 26.7317, 26.7452]
         ]
    y_pred = [26.3314, 28.0642, 26.6504, 27.0247, 26.7264]
    predicts_unr = []
    predicts_r = []
    i = 0
    for el in Xp:
        el = np.reshape(el, (1, -1))
        predicts_unr.append(model_unr.predict(el).tolist()[0][0])
    for elem in Xp:
        elem = np.reshape(elem, (1, -1))
        predicts_r.append(model_r.predict(elem).tolist()[0][0])

    fig = plt.figure(figsize=(10, 10))
    ax1 = plt.subplot()
    ax1.plot((1, 2, 3, 4, 5), (y_pred), label='Real values')
    ax1.legend()
    ax2 = plt.subplot()
    ax2.plot((1, 2, 3, 4, 5), (predicts_unr), label='Regular values')
    ax2.legend()
```

```

ax3 = plt.subplot()
ax3.plot((1, 2, 3, 4, 5), (predicts_r), label='Reduced values')
ax3.legend()
plt.title('Compare values of predictions')
plt.show()

```

- створення графіків порівняння середньоквадратичної помилки:

```

def do_charts_MSE(model_unr, model_r):
    Xp = [[26.6100, 26.5722, 26.4244, 26.3503],
           [27.9783, 28.0087, 28.0096, 28.0576],
           [26.6931, 26.6457, 26.6752, 26.6455],
           [26.9290, 26.8601, 26.8928, 27.0029],
           [26.9298, 26.8102, 26.7317, 26.7452]
          ]
    y_pred = [26.3314, 28.0642, 26.6504, 27.0247, 26.7264]
    predicts_unr = []
    predicts_r = []
    i = 0
    j = 0

    for el in Xp:
        el = np.reshape(el, (1, -1))
        predicts_unr.append(mean_squared_error([y_pred[i]],
model_unr.predict(el)))
        i += 1
    for elem in Xp:
        elem = np.reshape(elem, (1, -1))
        predicts_r.append(mean_squared_error([y_pred[j]],
model_r.predict(elem)))
        j += 1

```

```
print(predicts_unr)
print(predicts_r)
fig = plt.figure(figsize=(10, 10))
ax2 = plt.subplot()
ax2.plot((1, 2, 3, 4, 5), (predicts_unr), label='Regular NN MSE values')
ax2.legend()
ax3 = plt.subplot()
ax3.plot((1, 2, 3, 4, 5), (predicts_r), label='Reduced NN MSE values')
ax3.legend()
plt.title('Compare MSE of predictions')
plt.show()
```

4.5 Аналіз результатів роботи

Функції програмного продукту створювалися таким чином, щоб вони виводили на консоль проміжні результати і можна було бачити динаміку роботи системи і явні результати редукції.

В першу чергу система виводить на екран результати прогнозів для кожної з трьох описаних вище моделей і середньоквадратичну помилку для них (рис. 3.1)

```
Non-weighted model
Prediction data: [26.6100, 26.5722, 26.4244, 26.3503]
Needed result: [26.3314]
Result: [[26.25299684]]
MSE: 0.00614705
Weights [[-0.07342321 -0.03869003 -0.65654982  1.73938164]]

Weighted model
Prediction data: [26.6100, 26.5722, 26.4244, 26.3503]
Needed result: [26.3314]
Result: [[26.25299684]]
MSE: 0.00614705
Weights [[-0.07342321 -0.03869003 -0.65654982  1.73938164]]

Reduced model
Prediction data: [26.6100, 26.5722, 26.4244, 26.3503]
Needed result: [26.3314]
Result: [[26.32484591]]
MSE: 0.00004296
Weights [[-0.01913052 -0.02511482 -0.4843013  1.49165757]]
```

Рисунок 4.1 — Вивід результатів роботи на консоль

Для більш детального розуміння проведеної роботи і її результатів було створено функції побудови графіків порівняння, які дають змогу оцінити наглядно різницю в прогнозах і середньоквадратичних помилках між звичайною нейронною мережею і тією мережею, яка пройшла процедуру редукції.

Графік порівняння прогнозів зображено на рисунку 4.2.

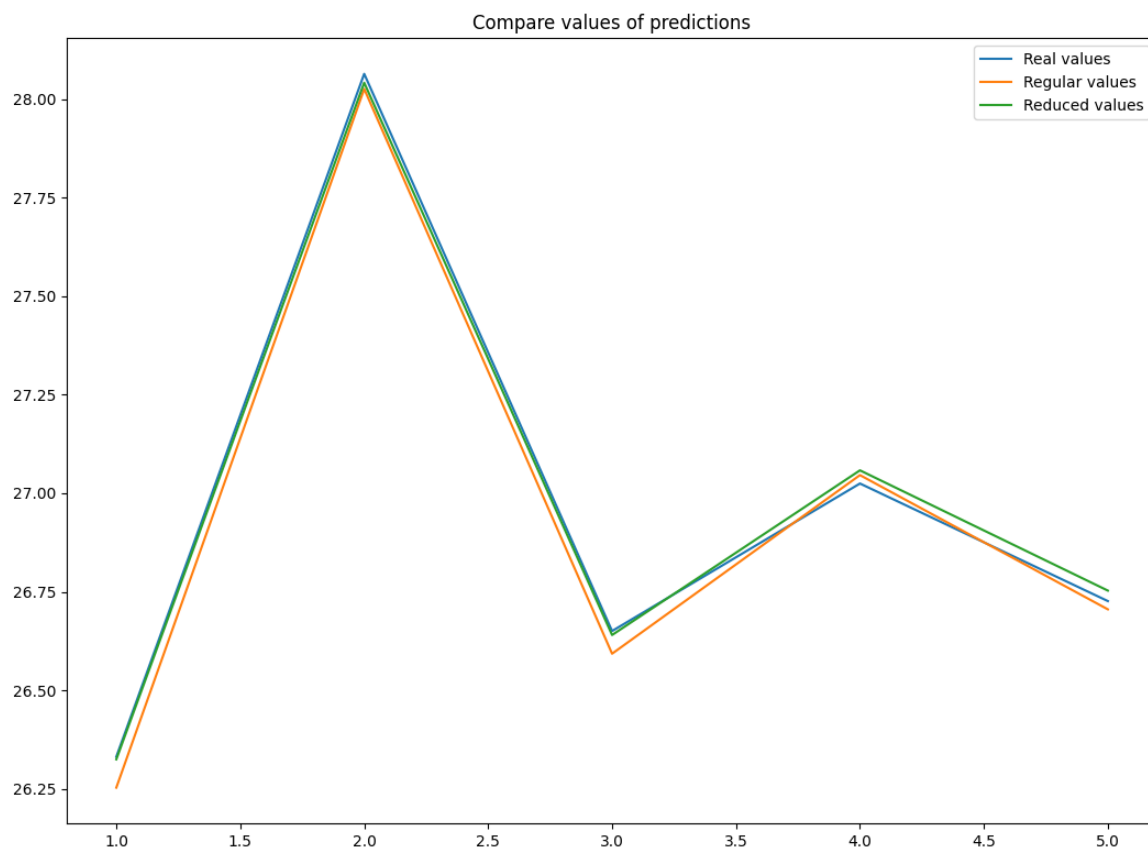


Рисунок 4.2 — Графік прогнозів моделей

Графік порівняння середньоквадратичних помилок зображено на рисунку 4.3.

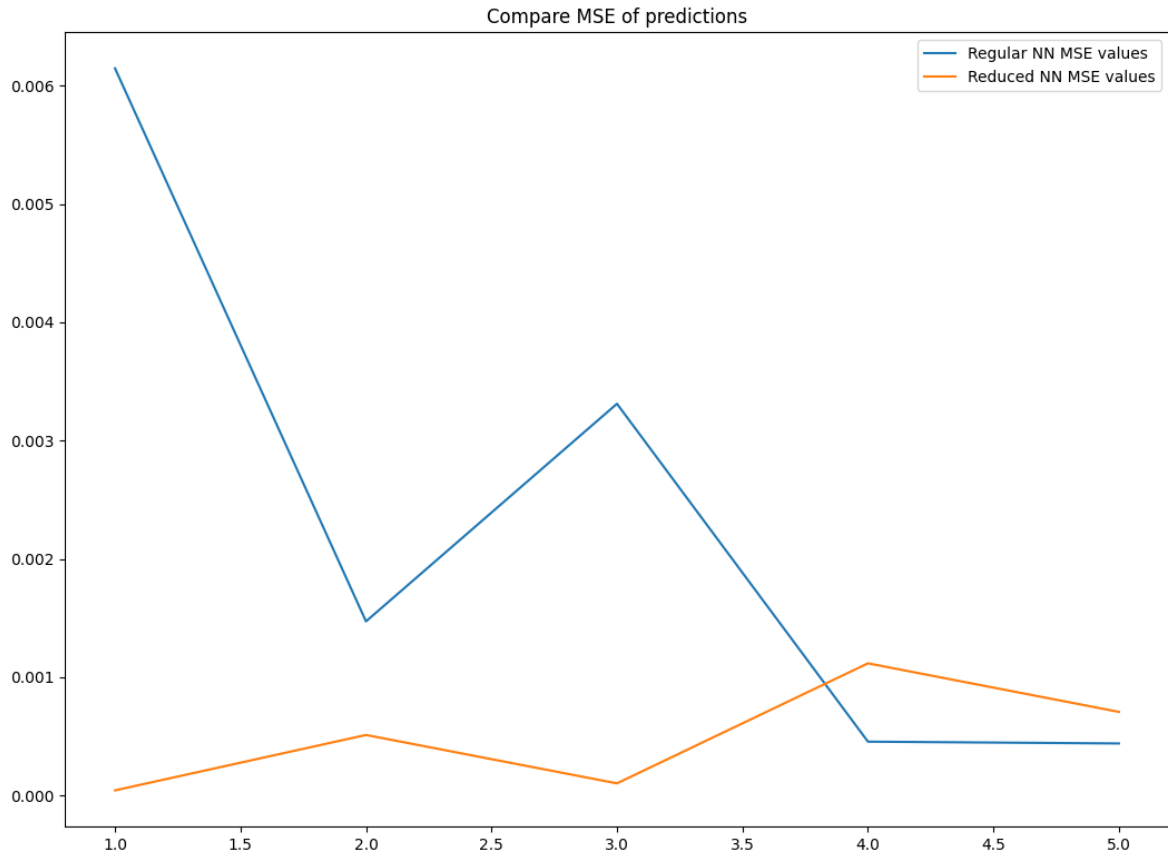


Рисунок 4.3 — Графік середньоквадратичних помилок

Як видно з графіків, редукція не створює мережу, яка краща за стандарту у всіх випадках, проте падіння середньоквадратичної помилки на більшості проміжків може вважатися позитивним результатом, адже середня похибка для всього датасету значно знижується за рахунок її зниження для більшості ситуацій.

За рахунок редукції було підвищено точність прогнозування нейронної мережі, функціоналом якої є прогнозування даних світового авторинку.

ВИСНОВКИ

Під час роботи були виконані наступні завдання:

Завдання дослідження:

1. Здійснений огляд літератури, визначені основні теоретичні підходи та методи аналізу даних автомобільного ринку.
2. Зібрані та систематизовані наявні дані про автомобільний ринок, включаючи статистику продажів, ціни, виробництво, витрати на маркетинг та інші фактори, що впливають на ринок.
3. Провести статистичний аналіз даних для виявлення взаємозв'язків та тенденцій на автомобільному ринку.
4. Розроблені моделі прогнозування на основі зібраних даних та використовуючи підходи машинного навчання та статистичного аналізу.
5. Здійснено валідацію та перевірку точності розроблених моделей прогнозування за допомогою історичних даних та порівняння прогнозів з реальними результатами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hardesty, Larry (14 April 2017). "Explained: Neural networks". MIT News Office. Retrieved 2 June 2022.
2. Yang, Z.R.; Yang, Z. (2014). *Comprehensive Biomedical Physics*. Karolinska Institute, Stockholm, Sweden: Elsevier. p. 1. ISBN 978-0-444-53633-4. Archived from the original on 28 July 2022. Retrieved 28 July 2022.
3. Mansfield Merriman, "A List of Writings Relating to the Method of Least Squares"
4. Stigler, Stephen M. (1981). "Gauss and the Invention of Least Squares". *Ann. Stat.* 9 (3): 465–474. doi:10.1214/aos/1176345451.
5. Bretscher, Otto (1995). *Linear Algebra With Applications* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.
6. Schmidhuber, Juergen (2022). "Annotated History of Modern AI and Deep Learning". arXiv:2212.11279 [cs.NE].
7. Stigler, Stephen M. (1986). *The History of Statistics: The Measurement of Uncertainty before 1900*. Cambridge: Harvard. ISBN 0-674-40340-1.
8. Brush, Stephen G. (1967). "History of the Lenz-Ising Model". *Reviews of Modern Physics*. 39 (4): 883–893. Bibcode:1967RvMP...39..883B. doi:10.1103/RevModPhys.39.883.
9. Amari, Shun-Ichi (1972). "Learning patterns and pattern sequences by self-organizing nets of threshold elements". *IEEE Transactions*. C (21): 1197–1206.
10. Hopfield, J. J. (1982). "Neural networks and physical systems with emergent collective computational abilities". *Proceedings of the National Academy of Sciences*. 79 (8): 2554–2558. Bibcode:1982PNAS...79.2554H. doi:10.1073/pnas.79.8.2554. PMC 346238. PMID 6953413.

11. McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics*. 5 (4): 115–133. doi:10.1007/BF02478259.
12. Kleene, S.C. (1956). "Representation of Events in Nerve Nets and Finite Automata". *Annals of Mathematics Studies*. No. 34. Princeton University Press. pp. 3–41. Retrieved 17 June 2017.
13. Hebb, Donald (1949). *The Organization of Behavior*. New York: Wiley. ISBN 978-1-135-63190-1.
14. Farley, B.G.; W.A. Clark (1954). "Simulation of Self-Organizing Systems by Digital Computer". *IRE Transactions on Information Theory*. 4 (4): 76–84. doi:10.1109/TIT.1954.1057468.
15. Haykin (2008) *Neural Networks and Learning Machines*, 3rd edition
16. Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization in the Brain". *Psychological Review*. 65 (6): 386–408. CiteSeerX 10.1.1.588.3775. doi:10.1037/h0042519. PMID 13602029. S2CID 12781225.
17. Werbos, P.J. (1975). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*.
18. Rosenblatt, Frank (1957). "The Perceptron—a perceiving and recognizing automaton". Report 85-460-1. Cornell Aeronautical Laboratory.
19. Olazaran, Mikel (1996). "A Sociological Study of the Official History of the Perceptrons Controversy". *Social Studies of Science*. 26 (3): 611–659. doi:10.1177/030631296026003005. JSTOR 285702. S2CID 16786738.
20. Minsky, Marvin; Papert, Seymour (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press. ISBN 978-0-262-63022-1.
21. Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". *Neural Networks*. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.

- 22.Ivakhnenko, A. G. (1973). Cybernetic Predicting Devices. CCM Information Corporation.
- 23.Ivakhnenko, A. G.; Grigor'evich Lapa, Valentin (1967). Cybernetics and forecasting techniques. American Elsevier Pub. Co.
- 24.Robbins, H.; Monro, S. (1951). "A Stochastic Approximation Method". The Annals of Mathematical Statistics. 22 (3): 400. doi:10.1214/aoms/1177729586.
- 25.Amari, Shun'ichi (1967). "A theory of adaptive pattern classifier". IEEE Transactions. EC (16): 279–307.
- 26.Kohonen, Teuvo; Honkela, Timo (2007). "Kohonen Network". Scholarpedia. 2 (1): 1568. Bibcode:2007SchpJ...2.1568K. doi:10.4249/scholarpedia.1568.
- 27.Kohonen, Teuvo (1982). "Self-Organized Formation of Topologically Correct Feature Maps". Biological Cybernetics. 43 (1): 59–69. doi:10.1007/bf00337288. S2CID 206775459.
- 28.Von der Malsburg, C (1973). "Self-organization of orientation sensitive cells in the striate cortex". Kybernetik. 14 (2): 85–100. doi:10.1007/bf00288907. PMID 4786750. S2CID 3351573.
- 29.Fukushima, Kunihiko (1980). "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position" (PDF). Biological Cybernetics. 36 (4): 193–202. doi:10.1007/BF00344251. PMID 7370364. S2CID 206775608. Retrieved 16 November 2013.
- 30.Fukushima, K. (1969). "Visual feature extraction by a multilayered network of analog threshold elements". IEEE Transactions on Systems Science and Cybernetics. 5 (4): 322–333. doi:10.1109/TSSC.1969.300225.
- 31.Ramachandran, Prajit; Barret, Zoph; Quoc, V. Le (16 October 2017). "Searching for Activation Functions". arXiv:1710.05941 [cs.NE].

32. Leibniz, Gottfried Wilhelm Freiherr von (1920). *The Early Mathematical Manuscripts of Leibniz: Translated from the Latin Texts Published by Carl Immanuel Gerhardt with Critical and Historical Notes* (Leibniz published the chain rule in a 1676 memoir). Open court publishing Company.
33. Linnainmaa, Seppo (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors (Masters) (in Finnish). University of Helsinki. pp. 6–7.
34. Linnainmaa, Seppo (1976). "Taylor expansion of the accumulated rounding error". *BIT Numerical Mathematics*. 16 (2): 146–160. doi:10.1007/bf01931367. S2CID 122357351.
35. Griewank, Andreas (2012). "Who Invented the Reverse Mode of Differentiation?". *Optimization Stories. Documenta Mathematica, Extra Volume ISMP*. pp. 389–400. S2CID 15568746.
36. Griewank, Andreas; Walther, Andrea (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Second Edition*. SIAM. ISBN 978-0-89871-776-1.
37. Rosenblatt, Frank (1962). *Principles of Neurodynamics*. Spartan, New York.
38. Kelley, Henry J. (1960). "Gradient theory of optimal flight paths". *ARS Journal*. 30 (10): 947–954. doi:10.2514/8.5282.
39. "A gradient method for optimizing multi-stage allocation processes". *Proceedings of the Harvard Univ. Symposium on digital computers and their applications*. April 1961.
40. Schmidhuber, Jürgen (2015). "Deep Learning". *Scholarpedia*. 10 (11): 85–117. Bibcode:2015SchpJ..1032832S. doi:10.4249/scholarpedia.32832.
41. Dreyfus, Stuart E. (1 September 1990). "Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure". *Journal of Guidance, Control, and Dynamics*. 13 (5): 926–928. Bibcode:1990JGCD...13..926D. doi:10.2514/3.25422. ISSN 0731-5090.

42. Mizutani, E.; Dreyfus, S.E.; Nishio, K. (2000). "On derivation of MLP backpropagation from the Kelley-Bryson optimal-control gradient formula and its application". Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium. IEEE: 167–172 vol.2. doi:10.1109/ijcnn.2000.857892. ISBN 0-7695-0619-4. S2CID 351146.
43. Dreyfus, Stuart (1973). "The computational solution of optimal control problems with time lag". IEEE Transactions on Automatic Control. 18 (4): 383–385. doi:10.1109/tac.1973.1100330.
44. Werbos, Paul (1982). "Applications of advances in nonlinear sensitivity analysis" (PDF). System modeling and optimization. Springer. pp. 762–770. Archived (PDF) from the original on 14 April 2016. Retrieved 2 July 2017.
45. David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams , "Learning representations by back-propagating errors Archived 8 March 2021 at the Wayback Machine," Nature', 323, pages 533–536 1986.
46. Waibel, Alex (December 1987). Phoneme Recognition Using Time-Delay Neural Networks. Meeting of the Institute of Electrical, Information and Communication Engineers (IEICE). Tokyo, Japan.
47. Alexander Waibel et al., Phoneme Recognition Using Time-Delay Neural Networks IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume 37, No. 3, pp. 328. – 339 March 1989.
48. Zhang, Wei (1988). "Shift-invariant pattern recognition neural network and its optical architecture". Proceedings of Annual Conference of the Japan Society of Applied Physics.
49. Zhang, Wei (1990). "Parallel distributed processing model with local space-invariant interconnections and its optical architecture". Applied Optics. 29 (32): 4790–7. Bibcode:1990ApOpt..29.4790Z. doi:10.1364/AO.29.004790. PMID 20577468.

50. LeCun et al., "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, 1, pp. 541–551, 1989.
51. J. Weng, N. Ahuja and T. S. Huang, "Cresceptron: a self-organizing neural network which grows adaptively Archived 21 September 2017 at the Wayback Machine," *Proc. International Joint Conference on Neural Networks*, Baltimore, Maryland, vol I, pp. 576–581, June 1992.
52. J. Weng, N. Ahuja and T. S. Huang, "Learning recognition and segmentation of 3-D objects from 2-D images Archived 21 September 2017 at the Wayback Machine," *Proc. 4th International Conf. Computer Vision*, Berlin, Germany, pp. 121–128, May 1993.
53. J. Weng, N. Ahuja and T. S. Huang, "Learning recognition and segmentation using the Cresceptron Archived 25 January 2021 at the Wayback Machine," *International Journal of Computer Vision*, vol. 25, no. 2, pp. 105–139, Nov. 1997.
54. LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner (1998). "Gradient-based learning applied to document recognition" (PDF). *Proceedings of the IEEE*. 86 (11): 2278–2324. CiteSeerX 10.1.1.32.9552. doi:10.1109/5.726791. S2CID 14542261. Retrieved 7 October 2016.
55. Qian, Ning, and Terrence J. Sejnowski. "Predicting the secondary structure of globular proteins using neural network models." *Journal of molecular biology* 202, no. 4 (1988): 865-884.
56. Bohr, Henrik, Jakob Bohr, Søren Brunak, Rodney MJ Cotterill, Benny Lautrup, Leif Nørskov, Ole H. Olsen, and Steffen B. Petersen. "Protein secondary structure and homology by neural networks The α -helices in rhodopsin." *FEBS letters* 241, (1988): 223-228
57. Rost, Burkhard, and Chris Sander. "Prediction of protein secondary structure at better than 70% accuracy." *Journal of molecular biology* 232, no. 2 (1993): 584-599.

58. Schmidhuber, Jürgen (1992). "Learning complex, extended sequences using the principle of history compression" (PDF). *Neural Computation*. 4 (2): 234–242. doi:10.1162/neco.1992.4.2.234. S2CID 18271205.
59. Schmidhuber, Jürgen (1993). Habilitation Thesis (PDF).
60. Schmidhuber, Jürgen (1 November 1992). "Learning to control fast-weight memories: an alternative to recurrent nets". *Neural Computation*. 4 (1): 131–139. doi:10.1162/neco.1992.4.1.131. S2CID 16683347.
61. Schlag, Imanol; Irie, Kazuki; Schmidhuber, Jürgen (2021). "Linear Transformers Are Secretly Fast Weight Programmers". *ICML 2021*. Springer. pp. 9355–9366.
62. Choromanski, Krzysztof; Likhoshesterov, Valerii; Dohan, David; Song, Xingyou; Gane, Andreea; Sarlos, Tamas; Hawkins, Peter; Davis, Jared; Mohiuddin, Afroz; Kaiser, Lukasz; Belanger, David; Colwell, Lucy; Weller, Adrian (2020). "Rethinking Attention with Performers". arXiv:2009.14794 [cs.CL].
63. Schmidhuber, Jürgen (1993). "Reducing the ratio between learning complexity and number of time-varying variables in fully recurrent nets". *ICANN 1993*. Springer. pp. 460–463.
64. Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N.; Kaiser, Lukasz; Polosukhin, Illia (12 June 2017). "Attention Is All You Need". arXiv:1706.03762 [cs.CL].
65. Wolf, Thomas; Debut, Lysandre; Sanh, Victor; Chaumond, Julien; Delangue, Clement; Moi, Anthony; Cistac, Pierrick; Rault, Tim; Louf, Remi; Funtowicz, Morgan; Davison, Joe; Shleifer, Sam; von Platen, Patrick; Ma, Clara; Jernite, Yacine; Plu, Julien; Xu, Canwen; Le Scao, Teven; Gugger, Sylvain; Drame, Mariama; Lhoest, Quentin; Rush, Alexander (2020). "Transformers: State-of-the-Art Natural Language Processing". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

- Processing: System Demonstrations. pp. 38–45.
doi:10.18653/v1/2020.emnlp-demos.6. S2CID 208117506.
- 66.He, Cheng (31 December 2021). "Transformer in CV". Transformer in CV. Towards Data Science.
- 67.Schmidhuber, Jürgen (1991). "A possibility for implementing curiosity and boredom in model-building neural controllers". Proc. SAB'1991. MIT Press/Bradford Books. pp. 222–227.
- 68.Schmidhuber, Jürgen (2010). "Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990-2010)". IEEE Transactions on Autonomous Mental Development. 2 (3): 230–247. doi:10.1109/TAMD.2010.2056368. S2CID 234198.
- 69.Schmidhuber, Jürgen (2020). "Generative Adversarial Networks are Special Cases of Artificial Curiosity (1990) and also Closely Related to Predictability Minimization (1991)". Neural Networks. 127: 58–66. arXiv:1906.04493. doi:10.1016/j.neunet.2020.04.008. PMID 32334341. S2CID 216056336.
- 70.Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). Generative Adversarial Networks (PDF). Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680. Archived (PDF) from the original on 22 November 2019. Retrieved 20 August 2019.
- 71."Prepare, Don't Panic: Synthetic Media and Deepfakes". witness.org. Archived from the original on 2 December 2020. Retrieved 25 November 2020.
- 72."GAN 2.0: NVIDIA's Hyperrealistic Face Generator". SyncedReview.com. 14 December 2018. Retrieved 3 October 2019.

- 73.Karras, Tero; Aila, Timo; Laine, Samuli; Lehtinen, Jaakko (1 October 2017). "Progressive Growing of GANs for Improved Quality, Stability, and Variation". arXiv:1710.10196.
- 74.S. Hochreiter., "Untersuchungen zu dynamischen neuronalen Netzen Archived 2015-03-06 at the Wayback Machine," Diploma thesis. Institut f. Informatik, Technische Univ. Munich. Advisor: J. Schmidhuber, 1991.
- 75.Hochreiter, S.; et al. (15 January 2001). "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies". In Kolen, John F.; Kremer, Stefan C. (eds.). *A Field Guide to Dynamical Recurrent Networks*. John Wiley & Sons. ISBN 978-0-7803-5369-5.
- 76.Hochreiter, Sepp; Schmidhuber, Jürgen (1 November 1997). "Long Short-Term Memory". *Neural Computation*. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. ISSN 0899-7667. PMID 9377276. S2CID 1915014.
- 77.Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". *Neural Networks*. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003.
- 78.Gers, Felix; Schmidhuber, Jürgen; Cummins, Fred (1999). "Learning to forget: Continual prediction with LSTM". *9th International Conference on Artificial Neural Networks: ICANN '99*. Vol. 1999. pp. 850–855. doi:10.1049/cp:19991218. ISBN 0-85296-721-7.
- 79.Srivastava, Rupesh Kumar; Greff, Klaus; Schmidhuber, Jürgen (2 May 2015). "Highway Networks". arXiv:1505.00387 [cs.LG].
- 80.Srivastava, Rupesh K; Greff, Klaus; Schmidhuber, Juergen (2015). "Training Very Deep Networks". *Advances in Neural Information Processing Systems*. Curran Associates, Inc. 28: 2377–2385.
- 81.He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016). *Deep Residual Learning for Image Recognition*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, US:

- IEEE. pp. 770–778. arXiv:1512.03385. doi:10.1109/CVPR.2016.90. ISBN 978-1-4673-8851-1.
- 82.Mead, Carver A.; Ismail, Mohammed (8 May 1989). Analog VLSI Implementation of Neural Systems (PDF). The Kluwer International Series in Engineering and Computer Science. Vol. 80. Norwell, MA: Kluwer Academic Publishers. doi:10.1007/978-1-4613-1639-8. ISBN 978-1-4613-1639-8. Archived (PDF) from the original on 6 November 2019. Retrieved 24 January 2020.
- 83.Domingos, Pedro (22 September 2015). "chapter 4". *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books. ISBN 978-0465065707.
- 84.Smolensky, P. (1986). "Information processing in dynamical systems: Foundations of harmony theory.". In D. E. Rumelhart; J. L. McClelland; PDP Research Group (eds.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol. 1. pp. 194–281. ISBN 978-0-262-68053-0.
- 85.Ng, Andrew; Dean, Jeff (2012). "Building High-level Features Using Large Scale Unsupervised Learning". arXiv:1112.6209 [cs.LG].
- 86.Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). *Deep Learning*. MIT Press. Archived from the original on 16 April 2016. Retrieved 1 June 2016.
- 87.Cireşan, Dan Claudiu; Meier, Ueli; Gambardella, Luca Maria; Schmidhuber, Jürgen (21 September 2010). "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition". *Neural Computation*. 22 (12): 3207–3220. arXiv:1003.0358. doi:10.1162/neco_a_00052. ISSN 0899-7667. PMID 20858131. S2CID 1918673.
- 88.Dominik Scherer, Andreas C. Müller, and Sven Behnke: "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition Archived 3 April 2018 at the Wayback Machine," In 20th International

- Conference Artificial Neural Networks (ICANN), pp. 92–101, 2010. doi:10.1007/978-3-642-15825-4_10.
- 89.2012 Kurzweil AI Interview Archived 31 August 2018 at the Wayback Machine with Jürgen Schmidhuber on the eight competitions won by his Deep Learning team 2009–2012
- 90."How bio-inspired deep learning keeps winning competitions | KurzweilAI". www.kurzweilai.net. Archived from the original on 31 August 2018. Retrieved 16 June 2017.
- 91.Graves, Alex; Schmidhuber, Jürgen (2009). "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks" (PDF). In Koller, D.; Schuurmans, Dale; Bengio, Yoshua; Bottou, L. (eds.). Advances in Neural Information Processing Systems 21 (NIPS 2008). Neural Information Processing Systems (NIPS) Foundation. pp. 545–552. ISBN 9781605609492.
- 92.Graves, A.; Liwicki, M.; Fernandez, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. (May 2009). "A Novel Connectionist System for Unconstrained Handwriting Recognition" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 31 (5): 855–868. CiteSeerX 10.1.1.139.4502. doi:10.1109/tpami.2008.137. ISSN 0162-8828. PMID 19299860. S2CID 14635907. Archived (PDF) from the original on 2 January 2014. Retrieved 30 July 2014.
- 93.Ciresan, Dan; Meier, U.; Schmidhuber, J. (June 2012). Multi-column deep neural networks for image classification. 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3642–3649. arXiv:1202.2745. Bibcode:2012arXiv1202.2745C. CiteSeerX 10.1.1.300.3283. doi:10.1109/cvpr.2012.6248110. ISBN 978-1-4673-1228-8. S2CID 2161592.

94. Zell, Andreas (2003). "chapter 5.2". Simulation neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.). Addison-Wesley. ISBN 978-3-89319-554-1. OCLC 249017987.
95. Artificial intelligence (3rd ed.). Addison-Wesley Pub. Co. 1992. ISBN 0-201-53377-4.
96. Abbod, Maysam F. (2007). "Application of Artificial Intelligence to the Management of Urological Cancer". *The Journal of Urology*. 178 (4): 1150–1156. doi:10.1016/j.juro.2007.05.122. PMID 17698099.
97. Dawson, Christian W. (1998). "An artificial neural network approach to rainfall-runoff modelling". *Hydrological Sciences Journal*. 43 (1): 47–66. doi:10.1080/02626669809492102.
98. "The Machine Learning Dictionary". www.cse.unsw.edu.au. Archived from the original on 26 August 2018. Retrieved 4 November 2009.
99. Ciresan, Dan; Ueli Meier; Jonathan Masci; Luca M. Gambardella; Jurgen Schmidhuber (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification" (PDF). *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Two*. 2: 1237–1242. Archived (PDF) from the original on 5 April 2022. Retrieved 7 July 2022.
100. Zell, Andreas (1994). *Simulation Neuronaler Netze [Simulation of Neural Networks]* (in German) (1st ed.). Addison-Wesley. p. 73. ISBN 3-89319-554-8.
101. Miljanovic, Milos (February–March 2012). "Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction" (PDF). *Indian Journal of Computer and Engineering*. 3 (1).
102. Lau, Suki (10 July 2017). "A Walkthrough of Convolutional Neural Network – Hyperparameter Tuning". *Medium*. Archived from the original on 4 February 2023. Retrieved 23 August 2019.

103. Kelleher, John D.; Mac Namee, Brian; D'Arcy, Aoife (2020). "7-8". Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies (2nd ed.). Cambridge, MA. ISBN 978-0-262-36110-1. OCLC 1162184998.
104. Wei, Jiakai (26 April 2019). "Forget the Learning Rate, Decay Loss". arXiv:1905.00094 [cs.LG].
105. Li, Y.; Fu, Y.; Li, H.; Zhang, S. W. (1 June 2009). The Improved Training Algorithm of Back Propagation Neural Network with Self-adaptive Learning Rate. 2009 International Conference on Computational Intelligence and Natural Computing. Vol. 1. pp. 73–76. doi:10.1109/CINC.2009.111. ISBN 978-0-7695-3645-3. S2CID 10557754.
106. Huang, Guang-Bin; Zhu, Qin-Yu; Siew, Chee-Kheong (2006). "Extreme learning machine: theory and applications". *Neurocomputing*. 70 (1): 489–501. CiteSeerX 10.1.1.217.3692. doi:10.1016/j.neucom.2005.12.126. S2CID 116858.
107. Widrow, Bernard; et al. (2013). "The no-prop algorithm: A new learning algorithm for multilayer neural networks". *Neural Networks*. 37: 182–188. doi:10.1016/j.neunet.2012.09.020. PMID 23140797.
108. Ollivier, Yann; Charpiat, Guillaume (2015). "Training recurrent networks without backtracking". arXiv:1507.07680 [cs.NE].
109. Hinton, G. E. (2010). "A Practical Guide to Training Restricted Boltzmann Machines". Tech. Rep. UTML TR 2010-003. Archived from the original on 9 May 2021. Retrieved 27 June 2017.
110. ESANN. 2009.[full citation needed]
111. Bernard, Etienne (2021). Introduction to machine learning. Champaign. p. 9. ISBN 978-1579550486. Retrieved 22 March 2023.
112. Bernard, Etienne (2021). Introduction to machine learning. Champaign. p. 12. ISBN 978-1579550486. Retrieved 22 March 2023.

113. Bernard, Etienne (2021). Introduction to Machine Learning. Wolfram Media Inc. p. 9. ISBN 978-1-579550-48-6.
114. Ojha, Varun Kumar; Abraham, Ajith; Snášel, Václav (1 April 2017). "Metaheuristic design of feedforward neural networks: A review of two decades of research". Engineering Applications of Artificial Intelligence. 60: 97–116. arXiv:1705.05584. Bibcode:2017arXiv170505584O. doi:10.1016/j.engappai.2017.01.013. S2CID 27910748.