

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІМ. Ю.М. ПОТЕБНІ  
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ**

**КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА  
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**Кваліфікаційна робота**

**другий (магістерський)**

(рівень вищої освіти)

на тему **Комп'ютерна система аналізу міжнародної економіки з  
використання машинного навчання**

Виконав: студент 2 курсу, групи 8.1212-іпз-2  
спеціальності 121 Інженерія програмного  
забезпечення

(код і назва спеціальності)

освітньої програми Інженерія програмного  
забезпечення

(код і назва освітньої програми)

І.В. Кириченко

(ініціали та прізвище)

Керівник доцент, А.І. Безверхий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ «Дісітел»

П.О. Лютий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя  
2023

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ**  
**ІМ. Ю.М. ПОТЕБНІ**  
**ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ**

Кафедра електроніки, інформаційних систем та програмного  
забезпечення

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 121 Інженерія програмного забезпечення \_\_\_\_\_  
(код та назва)

Освітня програма \_\_\_\_\_ Інженерія програмного забезпечення \_\_\_\_\_  
(код та назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри \_\_\_\_\_ Тетяна Критська \_\_\_\_\_

“ 02” вересня 2023 року

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

\_\_\_\_\_ Кириченко Ігор В'ячеславович \_\_\_\_\_

(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система аналізу міжнародної економіки з використанням машинного навчання

керівник роботи \_\_\_\_\_ доцент \_\_\_\_\_ Безверхий Анатолій Ігорович \_\_\_\_\_  
( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від \_\_\_\_\_ 09.10.2023, № 1577- С \_\_\_\_\_

2. Строк подання студентом кваліфікаційної роботи \_\_\_\_\_ 30.11.2023 \_\_\_\_\_

3. Вихідні дані магістерської роботи

- комплект нормативних документів ;
- технічне завдання до роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- дослідження проблеми розпізнавання мов та розробка методів її вирішення;
- створення програмного продукту та його опис;
- перелік вимог для роботи програми;
- дослідження поставленої проблеми та розробка висновків та пропозицій.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
слайдів презентації

## 6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 02.09.2023

## КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
	Аналіз предметної області	02.09-10.09.23	виконано
	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	11.09-12.09.23	виконано
	Аналіз проблематики	13.09-14.09.23	виконано
	Дослідження засобів та технологій прогнозування ВВП	15.09-20.09.23	виконано
	Дослідження основних типів існуючих нейронних мереж	21.09-26.09.23	виконано
	Узгодження подальших дій з науковим керівником	27.09-28.09.23	виконано
	Збір тренувального набору для нейронної мережі з відкритих джерел та його попередня підготовка	29.09-13.10.23	виконано
	Навчання моделі нейронної мережі за допомогою підготовленого тренувального набору	14.10-16.10.23	виконано
	Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження	17.10-19.10.23	виконано
0	Обробка отриманих результатів	20.10-09.11.23	виконано
1	Аналіз отриманих результатів	10.11-17.11.23	виконано
2	Оформлення звіту	17.11-27.11.23	виконано

Студент Кириченко І.В.  
( підпис ) (прізвище та ініціали)

Керівник роботи Безверхий А.І.  
( підпис ) (прізвище та ініціали)

**Нормоконтроль пройдено**

Нормоконтролер Скрипник І.А.  
( підпис ) (прізвище та ініціали)

## АНОТАЦІЯ

Сторінок: 82

Рисунків: 41

Таблиць: 0

Джерел: 14

Кириченко І.В. Комп'ютерна система аналізу міжнародної економіки з використанням машинного навчання: кваліфікаційна робота магістра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник А. І. Безверхий. Запоріжжя : ЗНУ, 2023. 82 с.

Мета і завдання дослідження полягають у вивченні можливостей нейронних мереж, а також у розробці комп'ютерної програми на мові Python яка б давала можливість прогнозувати ВВП країни на підставі наявних даних, зібраних у відповідний Dataset. Крім того, отримати можливість аналізу залежності отриманого результату від вихідних даних та вивчення причинно – наслідкових залежностей.

У процесі дослідження була розглянута проблема аналізу та прогнозування ВВП країн світу за допомогою комп'ютерної системи. Вивчені переваги використання нейронних мереж для прогнозування економічних показників. Реалізовано комп'ютерну систему на основі нейронної мережі для аналізу та прогнозування ВВП країни, що демонструє високу точність та адаптивність до змін економічних параметрів. Розглянуті переваги LSTM – нейронних мереж. На основі проведеного дослідження зроблено висновок про те, що комп'ютерна система на основі LSTM-нейронної мережі є ефективним інструментом для аналізу та прогнозування ВВП країни.

Ключові слова: нейронна мережа зворотного поширення помилки, глибинне навчання, часовий ряд, штучні нейронні мережи, Python, Keras, TensorFlow, набір даних, датасет, внутрішній валовий продукт (ВВП).

## SUMMARY

Pages: 82

Figures: 41

Tables: 0

Sources: 14

Kyrychenko I.V. Computer system for analyzing the international economy using machine learning: qualification work of the master of specialty 121 "Software Engineering" / Science head A.I. Bezverkhy. Zaporizhzhia : ZNU, 2023. 82 p.

The goal and objectives of the research are to explore the capabilities of neural networks and to develop a computer program in Python that would enable forecasting the GDP of a country based on available data collected in a corresponding dataset. Additionally, the aim is to analyze the dependency of the obtained results on input data and study cause-and-effect relationships.

During the research process, the issue of analyzing and forecasting the GDP of countries worldwide using a computer system was considered. The advantages of using neural networks for predicting economic indicators were studied. A computer system based on a neural network was implemented for the analysis and prediction of the GDP of countries worldwide, demonstrating high accuracy and adaptability to changes in economic parameters. The advantages of LSTM (Long Short-Term Memory) neural networks were discussed. Based on the conducted research, the conclusion was drawn that a computer system based on LSTM neural networks is an effective tool for the analysis and prediction of a country's GDP.

Keywords: backpropagation neural network, deep learning, time series, artificial neural networks, Python, Keras, TensorFlow, dataset, Gross Domestic Product (GDP).

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ПРОГНОЗУВАННЯ ВВП КРАЇНИ .....	13
1.1 Огляд проблеми прогнозування ВВП .....	13
1.2 Засоби та технології прогнозування ВВП .....	15
1.2.1 Економетрична модель .....	15
1.2.2 Модель векторної авторегресії (VAR) .....	16
1.2.3 Модель регресії .....	17
1.2.4 Факторні моделі .....	18
1.2.5 Нейронні мережі.....	19
1.3 Сфери застосування прогнозів ВВП .....	26
1.4 Переваги використання нейронних мереж в порівнянні з традиційними методами прогнозування ВВП.....	28
1.5 Висновки з розділу 1 .....	29
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ОСНОВНИХ ТИПІВ ІСНУЮЧИХ НЕЙРОННИХ МЕРЕЖ.....	30
2.1 Персептрон .....	30
2.2 Мережі з передачею зворотного поширення (Feedforward Neural Networks).....	31
2.3 Рекурентні нейронні мережі (Recurrent Neural Networks - RNN).....	33
2.4 Мережі згорткового типу (Convolutional Neural Networks - CNN).....	35
2.5 Мережі глибокого навчання (Deep Neural Networks – DNN).....	36
2.6 Самоорганізуючі карти (Self-Organizing Maps - SOM).....	38
2.7 Енкодер - декодер (Autoencoder).....	40

2.8	Мережі довготривалої пам'яті (Long Short-Term Memory - LSTM) .....	42	
2.9	Висновки з розділу 2 .....	45	
РОЗДІЛ 3 РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ ДЛЯ			
ПРОГНОЗУВАННЯ ВВП.....			46
3.1	Налаштування апаратного середовища.....	46	
3.2	Підготовка робочого DataSet-у.....	46	
3.3	Вибір програмного середовища, бібліотек та фреймворків.....	51	
3.4	Вибір архітектури проекту.....	66	
3.5	Висновки з розділу 3 .....	68	
РОЗДІЛ 4 МОДЕЛЮВАННЯ ТА АНАЛІЗ РОБОТИ НЕЙРОННОЇ МЕРЕЖІ			
.....			69
4.1	Реалізація моделі за допомогою бібліотеки scikit-learn, pandas та matplotlib.....	69	
4.2	Реалізація LSTM нейронної мережі.....	73	
4.3	Аналіз отриманих результатів.....	75	
4.4	Висновки з розділу 4 .....	77	
ВИСНОВКИ.....			79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....			81

## ВСТУП

### Актуальність теми

ВВП ключовий показник економічного зростання та загального стану країни, відіграє важливу роль в міжнародному порівнянні та внутрішньому моніторингу економіки. Є основним показником економічного розвитку країн на макрорівні, який характеризує сукупну ринкову вартість кінцевих товарів та послуг, вироблених підприємствами, організаціями та установами в поточному періоді на економічній території країни. Велике значення показник ВВП у розрахунку на одну особу має насамперед під час зіставлень з іншими країнами світу. Відслідковує економічні тренди та зміни на глобальному рівні, що є корисним для дослідників та аналітиків при аналізі тенденцій та розвитку світової економіки. Відіграє важливу роль в прийнятті рішень економічного характеру на рівні урядів та бізнесу, допомагаючи визначити пріоритети розвитку та алокувати ресурси. Корисний для оцінки розвитку та ефективності економічних політик, а також моніторингу стабільності та прогресу національної економіки. За допомогою статистичного аналізу можна визначити подальший економічний розвиток та чинники, що на нього впливають.

Необхідність такого аналізу та прогнозування на пряму виходить з такого показника, як Економічна безпека держави. Ця система є характеристикою стану економічної системи, оскільки сама система включає ряд підсистем, що відображають функціонування окремих сфер економіки: макроекономічну, інвестиційну, інноваційну, фінансову, соціальну, зовнішньоекономічну, енергетичну, продовольчу, демографічну. У свою чергу, Економічна Безпека є підсистемою системи вищого рівня – національної безпеки, що досягається таким рівнем розвитку і таким станом захищеності економіки, який в повній мірі забезпечує потреби держави і створює умови для інноваційного розвитку.

Для прийняття необхідних управлінських рішень з метою забезпечення досягнення максимально можливого потенціалу розвитку необхідно, перш за



все, оцінити поточний економічний стан регіону (країни, регіонів, видів економічної діяльності) і максимальний поріг потенційних можливостей при існуючих технологіях, рівнях зайнятості й обсягах виробничого капіталу. Правильний аналіз та прогнозування ВВП дає змогу передбачити тенденції розвитку економіки, що важливо для планування бюджетів, інвестицій та розвитку соціальних програм.

### **Мета і завдання дослідження**

полягає в дослідженні можливостей нейронних мереж для прогнозування ВВП країни.

### **Завдання дослідження:**

- проаналізувати наявні наукові дослідження проблематики прогнозування економічних показників;
- розглянути основні методи прогнозування часових рядів на базі мови Python;
- проаналізувати наявне програмне забезпечення для прогнозування економічних показників;
- розглянути фреймворки та бібліотеки для розробки нейронної мережі;
- розглянути та розробити потрібні дата-сети для навчання нейронної мережі;
- розробити комп'ютерну модель для прогнозування економічних показників в тому числі ВВП.

### **Об'єкт дослідження**

Процес комп'ютерного прогнозування економічних показників країни.

### **Предмет дослідження**

Система прогнозування економічних показників країни.

### **Методи дослідження**

Для вирішення поставленої задачі використовуються наступні методи дослідження:

1. аналіз існуючих рішень прогнозування часових рядів;
2. аналіз бібліотек глибокого навчання нейронних мереж;
3. аналіз та синтез отриманих знань та навичок у процесі дослідження проблеми прогнозування ВВП країни;
4. експериментування з використанням різних дата-сетів для навчання нейронної мережі.

### **Наукова новизна одержаних результатів**

Наукова новизна одержаних результатів дослідження полягає у тому, що для вирішення задачі прогнозування ВВП був використаний новий та ефективний підхід, а саме: була розроблена та навчена оптимальна модель LSTM - нейронної мережі..

### **Практичне значення одержаних результатів**

Практичне значення одержаних результатів дослідження полягає у тому, що була розроблена комп'ютерна система, яка здатна прогнозувати ВВП країни на підставі зібраних та оброблених статистичних даних, а також здатна показувати залежності між певними вхідними даними та результатом прогнозування.

### **Апробація одержаних результатів**

Результати дослідження були представлені на XVI науково-практичній конференції студентів, аспірантів, докторантів і молодих вчених Запорізького національного університету «Молода наука-2023» [1], а також на конференції здобувачів вищої освіти, аспірантів та молодих вчених Актуальні питання сталого науково-технічного та соціально-економічного розвитку регіонів України [2].

## Глосарій

*Глибинне (глибоке) навчання (анг. Deep learning)* – це сукупність методів машинного навчання (з вчителем, з частковим залучення вчителя, без вчителя та підкріпленням), що ґрунтуються на навчанні ознак з даних при використанні багатошарових нейронних мереж.

*Нейронні мережі зворотного поширення помилки (анг. Backpropagation)* - це тип навчання з учителем для багатошарових перцептронів (MLP) або штучних нейронних мереж. Зворотне поширення помилки є методом, що використовується для мінімізації функції втрат шляхом оновлення ваг нейронів на основі градієнту функції втрат відносно цих ваг.

*Набір даних або дата-сет (анг. Dataset)* – це інформація, яка використовується для навчання нейронної мережі, яка була попередньо певним чином оброблена та структурована.

*Часовий ряд* – це деяка послідовність визначених значень досліджуваної ознаки, тобто статистичного показника, що впорядкована у хронологічному порядку.

*Штучні нейронні мережі* – це математична модель, що побудована за принципом функціонування біологічних нейронних мереж, а саме мереж нервових клітин реального живого організму та складається з взаємопов'язаних нейронів, які є простими процесорами, що працюють паралельно.

*Python* - це потужна багатопарадигмова інтерпретована об'єктно - орієнтована мова комп'ютерного програмування, оптимізована для продуктивності програміста, легкої читаності коду та якості програмного забезпечення.

*Keras* — це відкритий фреймворк для глибинного навчання, розроблений з акцентом на швидке експериментування, що є надбудовою над бібліотекою TensorFlow.

*TensorFlow* — це відкрита бібліотека для машинного навчання. Вона може бути використана для широкого кола задач, але особлива увага приділяється навчанню та використанню глибоких нейронних мереж.

*Внутрішній валовий продукт (ВВП)* - сукупна вартість усіх товарів та послуг, створених на економічній території країни за певний проміжок часу з метою їх кінцевого використання.

# РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ПРОГНОЗУВАННЯ ВВП КРАЇНИ

## 1.1 Огляд проблеми прогнозування ВВП

В сучасному глобалізованому економічному середовищі всі країни взаємодіють і впливають одна на одну [3]. Тому прогнозування ВВП країни виявляється ключовим елементом в питанні стратегічного управління та розвитку. Внутрішній валовий продукт є не лише числовим показником економічного виробництва, але й важливим інструментом для передбачення, аналізу та визначення основних тенденцій у господарській сфері.

Задачею прогнозування ВВП є створення системи, яка може аналізувати великий обсяг економічних даних та передбачати їхній розвиток у майбутньому. Цей процес відіграє важливу роль у прийнятті стратегічних рішень на всіх рівнях - від бізнес-сектору та інвесторів до урядових інституцій. ВВП стає орієнтиром, який не лише відображає стан економіки країни, але і визначає її конкурентоспроможність, стабільність та соціальний розвиток.

Прогнозування ВВП має вирішальне значення також для бізнес-середовища. Підприємства використовують ці прогнози для оптимізації виробничих стратегій, враховуючи передбачувані зміни у рівні споживчого попиту та кон'юнктури ринку. Для інвесторів це є керованим дороговказом, який допомагає приймати обдумані рішення щодо розміщення капіталу та ризиків інвестицій.

Не менш важливою є роль прогнозування ВВП в економічному плануванні та формуванні державної політики країни. Уряд використовує ці дані для розробки стратегій, спрямованих на підтримання стабільності економіки, регулювання інфляції та забезпечення соціального розвитку. Прогноз ВВП слугуватиме важливим інструментом для визначення необхідних заходів для забезпечення сталого економічного зростання та благополуччя громадян.

Важливо відзначити, що прогноз ВВП необхідний не лише на внутрішньому рівні, але і в міжнародному контексті також. Країни використовують цю інформацію для управління економічними відносинами, укладення торгових угод та визначення своєї ролі та ваги на світовому ринку. Прогнози ВВП стають дієвим інструментом для визначення конкурентоспроможності країни та розробки стратегій залучення іноземних інвестицій.

У великому фінансовому світі прогноз ВВП є ключовим фактором для банків та інших фінансових установ. Вони використовують ці прогнози для оцінки ризиків, пов'язаних з кредитуванням, та розробки стратегій фінансового управління капіталом. Прогноз ВВП є сенсором для фінансового світу, що дозволяє адаптувати стратегії та мінімізувати ризики в умовах постійної зміни економічного середовища.

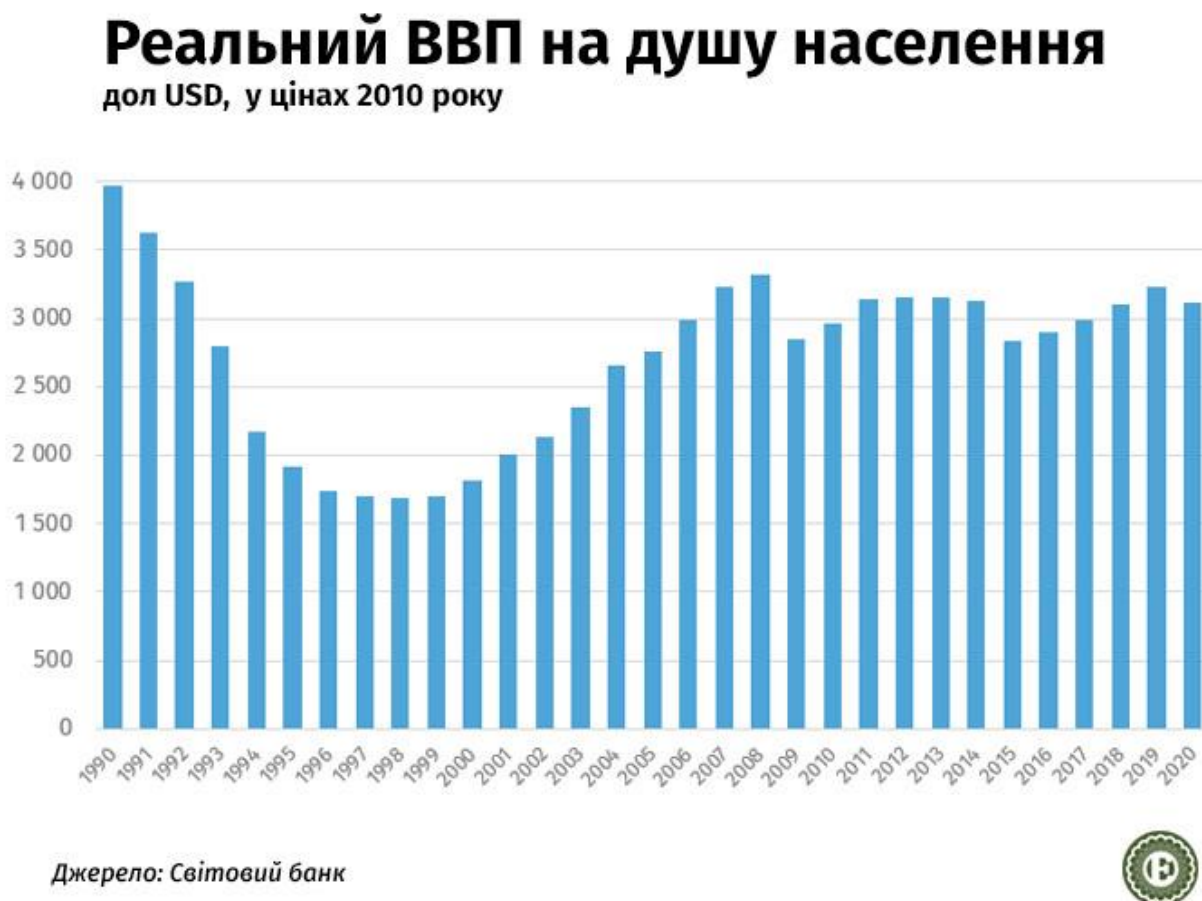


Рисунок 1 — Динаміка реального ВВП України з 1990 по 2020 рік

Отже, прогноз ВВП виступає не лише як статичний показник економічного розвитку, але й як динамічний інструмент, що формує основу для прийняття обґрунтованих рішень у бізнесі, політиці та фінансах. Важливість цього процесу полягає в його здатності відобразити складні економічні взаємодії та передбачити їхні наслідки, надаючи всім зацікавленим сторонам інструмент для ефективного управління та планування в умовах невизначеності та постійних змін.

## **1.2 Засоби та технології прогнозування ВВП**

Для прогнозування, в тому числі і можливого прогнозування ВВП існує багато різних моделей. Всі вони мають свої переваги та недоліки, різної складності та з різною специфікою застосування. Як приклади можна навести деякі з них

### **1.2.1 Економетрична модель**

Економетрична модель — це математична структура, яка використовує економетричні методи для аналізу економічних явищ та встановлення статистичних відносин між різними змінними [4]. Основна мета економетричних моделей — дослідження та прогнозування економічних явищ на основі статистичних даних.

Економетричні моделі базуються на економічних теоріях та включають параметри, які оцінюються на основі доступних даних. Ці моделі можуть бути лінійними чи нелінійними, враховувати один або кілька економічних факторів. Наприклад, лінійна регресійна модель може використовуватися для визначення взаємозв'язків між ВВП та іншими факторами, такими як інвестиції чи споживання.

Економетричні моделі використовуються для тестування економічних гіпотез, визначення впливу факторів на економічні зміни, та прогнозування

майбутнього розвитку економіки. Прикладом може служити комплексна модель, яка досліджує зв'язки між основними економічними показниками в країні:

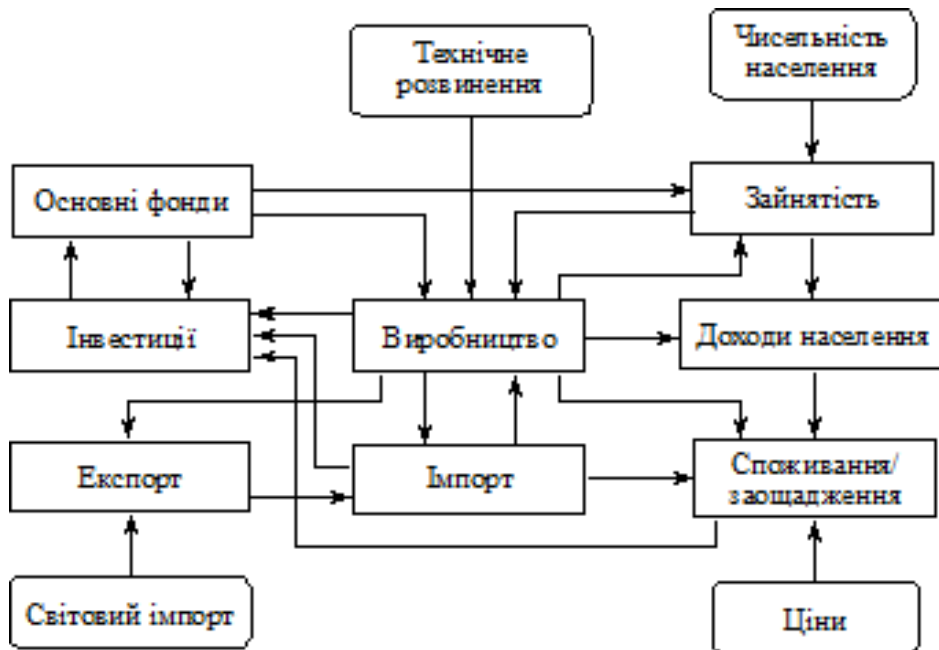


Рисунок 2 — Основні блоки змінних і зв'язки між ними в комплексній економетричній моделі

Використання економетричних моделей сприяє розумінню складних економічних взаємозв'язків, а отримані результати можуть служити основою для прийняття управлінських рішень та розробки політики в економіці.

### 1.2.2 Модель векторної авторегресії (VAR)

Модель векторної авторегресії - це статистична модель, яка використовується для аналізу та прогнозування взаємозв'язків між кількома часовими рядами [5]. У VAR кожна змінна розглядається як лінійна комбінація її власних попередніх значень та значень інших змінних.

Модель VAR може бути представлена наступним чином:

$$Y_t = c + A_1 Y_{t-1} + A_2 Y_{t-2} + \dots + A_p Y_{t-p} + \varepsilon_t,$$



де  $Y_t$  - вектор часових рядів,  $c$  - константа,  $A_1, A_2, \dots, A_p$  - матриці параметрів,  $p$  - порядок авторегресії,  $\varepsilon_t$  - вектор білих шумів.

Основна ідея VAR полягає в тому, що кожна змінна може бути взята участі в поясненні інших змінних у системі. Таким чином, VAR дозволяє враховувати взаємодії між різними економічними змінними. Прикладами таких змінних можуть бути ВВП, індекси споживчих цін, обсяги інвестицій тощо.

Наприклад, у VAR моделі для аналізу взаємозв'язку ВВП та індексу інфляції, кожен елемент вектору  $Y_t$  може представляти собою щомісячне спостереження за ВВП та індексом інфляції. Матриці  $A_1, A_2, \dots, A_p$  вказують на важливість і напрямок взаємозв'язків між цими змінними. Застосування VAR дозволяє отримати більш повне уявлення про динаміку економічних змін в часі та робити прогнози на основі минулих даних.

### 1.2.3 Модель регресії

Модель регресії - це статистичний інструмент, що використовується для вивчення взаємозв'язків між залежною змінною (яку ми намагаємося прогнозувати) і однією чи більше незалежними змінними (факторами, що можуть впливати на залежну змінну) [6]. Основною метою моделі регресії є оцінка та прогнозування значень залежної змінної на основі впливу незалежних змінних.

Модель регресії може бути лінійною або нелінійною. У лінійній моделі регресії взаємозв'язок між змінними виражається лінійною функцією, наприклад, у вигляді  $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \varepsilon$ , де  $y$  - залежна змінна,  $x_1, x_2$  - незалежні змінні,  $\beta_0, \beta_1, \beta_2$  - параметри моделі,  $\varepsilon$  - помилка.

Як приклад можна розглянути оцінку впливу кількості годин вивчення (незалежна змінна) на оцінки студента (залежна змінна), де лінійна модель регресії може виглядати як  $y = \beta_0 + \beta_1 x + \varepsilon$ .

Крім того, модель регресії використовується в економіці для прогнозування ВВП на основі різноманітних факторів, таких як інвестиції, споживання

та експорт. Оцінюючи ваги кожного фактору, економісти можуть намагатися передбачити майбутні зміни в економіці та приймати обґрунтовані рішення.

#### 1.2.4 Факторні моделі

Факторні моделі є одним із методів прогнозування економічних показників, таких як ВВП [7]. Ці моделі базуються на концепції факторів, які впливають на економічну динаміку, і дозволяють аналізувати взаємодію між різними факторами та їх вплив на економічний розвиток.

У факторних моделях враховуються різні чинники, такі як технологічний прогрес, споживчі витрати, інвестиції, експорт, імпорт, та інші економічні змінні. Модель розглядає їхні взаємодії та вагомість у формуванні економічних результатів.

Декілька прикладів факторних моделей в економіці:

Факторна модель споживання:

Модель враховує різні фактори, які впливають на споживання, такі як доходи, ціни, та інші. Вона може вказати, які чинники найбільше впливають на попит та як це впливає на ВВП.

Факторна модель інвестицій:

Враховує різні фактори, що впливають на рішення компаній і фірм здійснювати інвестиції. Це може включати рівень відсоткових ставок, наявність кредитів, і технологічний прогрес.

Факторна модель зовнішньої торгівлі:

Аналізує фактори, які впливають на обсяг експорту та імпорту, такі як обмеження торгівлі, курси валют, та світові економічні тенденції.

Факторна модель технологічного розвитку:

Вивчає вплив інновацій, досліджень та розвитку на економічний ріст. Факторами можуть бути витрати на дослідження та розвиток, наявність технологічного переваги тощо. Наприклад – схема факторної моделі виробництва валової продукції:

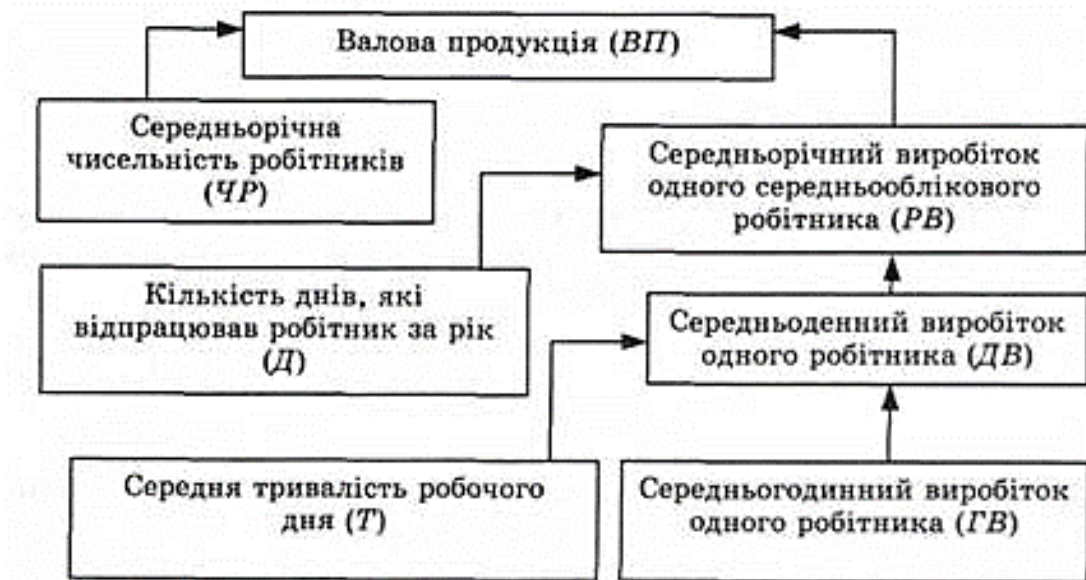


Рисунок 3 — Детермінована факторна модель виробництва валової продукції

Таким чином факторні моделі надають можливість оцінити вагомість та взаємодію різних факторів у формуванні економічних індикаторів.

### 1.2.5 Нейронні мережі

Нейромережі — це комп'ютерні системи, які моделюють структуру та функції мозку для вирішення завдань штучного інтелекту. Вони складаються зі штучних нейронів, які організовані в шари та взаємодіють за допомогою зв'язаних з'єднань. За своєю побудовою Нейромережі є дуже спрощеними аналогами біологічних систем чоловічого мозку. Для обробки інформації людський використовує спеціальні біологічні елементи – нейрони. Сукупність таких нейронів з'єднаних між собою в центральній нервовій системі та ганглії утворює біологічну нейронну мережу [8].

Біологічний нейрон сприймає вхідний сигнал із інших нейронів чи джерел інформації через дендрит, обробляє його в тілі сомі, та робить передачу вихідного сигналу через аксон до синапсу, який може або має бути з'єднаний

з дендритами наступного нейрона. Один біологічний нейрон може бути з'єднаний дендритами з більш як двадцять тисячами інших нейронів, для прийому від них вхідного сигналу, але на виході, зазвичай, наявний тільки один аксон.

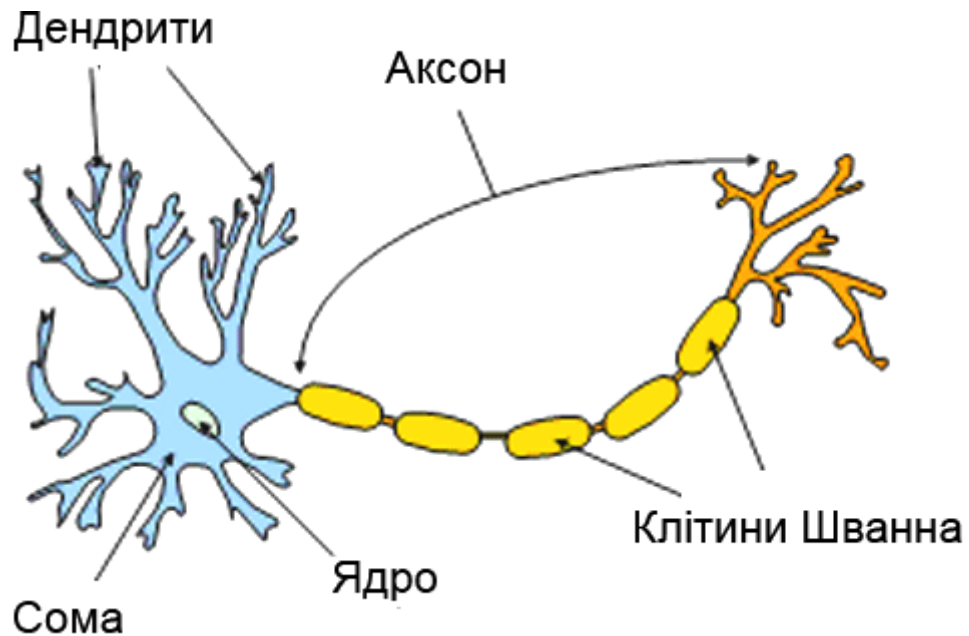


Рисунок 4 — Біологічний нейрон

Як і біологічний, штучний нейрон має здатність приймати на вхід сигнал від великої кількості нейронів, хоча і зазвичай їхня кількість не вимірюється десятками тисяч.

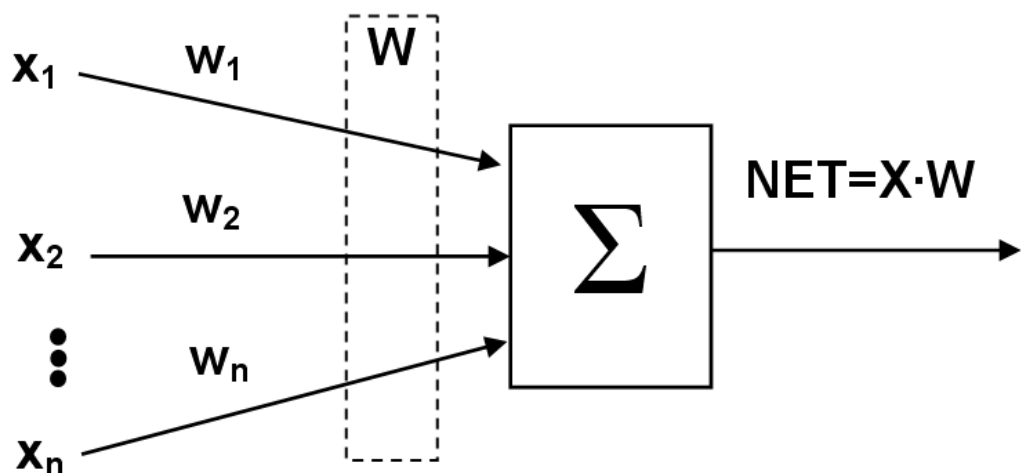


Рисунок 5 — Будова штучного нейрона

Штучний нейрон має так звані синаптичні ваги на входах, які визначають силу вхідних зв'язків. Тобто якщо це менш важливий зв'язок то він має меншу вагу та навпаки. Обробка інформації відбувається через суматор та функцію активації (передаточну функцію). Суматор перетворює зважені вхідні сигнали за формулою  $S = \sum_i w_i x_i$ . Після цього результат обробляється функцією активації за певною формулою  $Y = f(S)$  після чого подається на вихід нейрона.

Нейрони в штучних нейромережах можуть бути об'єднані в шари. Сигнал поступово передається із вхідних шарів до вихідних. То ж в більшості випадків, кожен нейрон вихідного і наступних шарів, окрім вихідного, з'єднаний із кожним нейроном наступного шару. Між вхідним та вихідним шарами можуть розміщуватися так звані приховані шари.

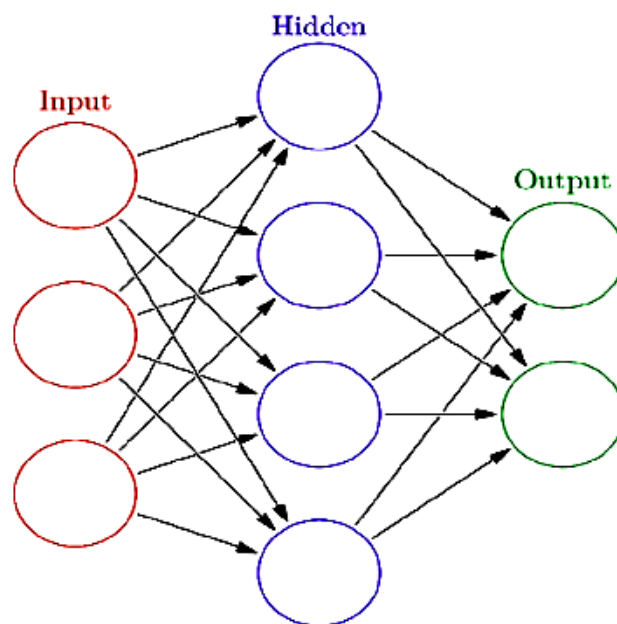


Рисунок 6 — Схема простої нейромережі з прихованим шаром

Розглянемо що таке функція активації або передавальна функція штучного нейрона. Це деяка функція, яка зазвичай відображає дійсні числа вхідного сигналу на числа з інтервалу  $(0,1)$  як вихідний сигнал. Однією із найпростіших функцій активації можна назвати ступінчасту функцію, що приймає значення 0 якщо  $Y$  не перевищує певного ліміту і 1. Проте в іншому випадку

$$Y(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

Існують функції із варіантами значень -1, 0, 1 тобто коли є  $x < 0$ ,  $x = 0$ ,  $x > 0$  відповідно.

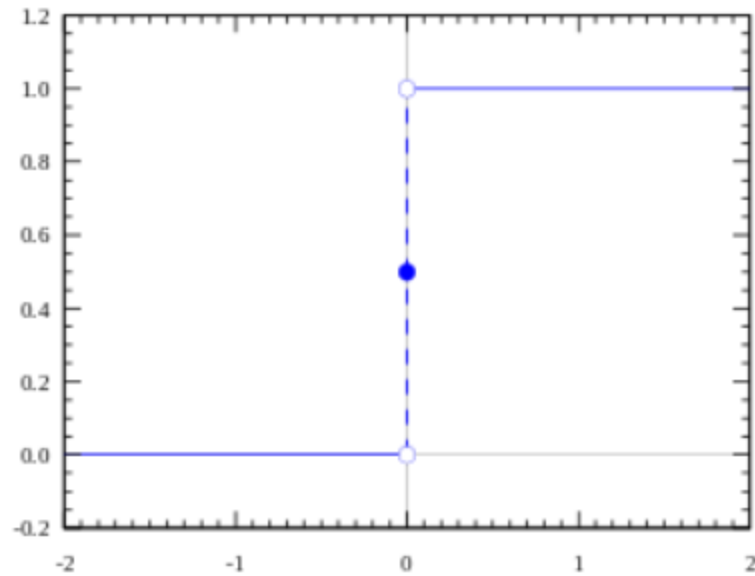


Рисунок 7 — Графік ступінчастої функції

Більшість нейромереж використовує сигмоїдні функції активації, наприклад, логістичну криву,

$$Y(x) = \frac{e^x}{e^x + 1}$$

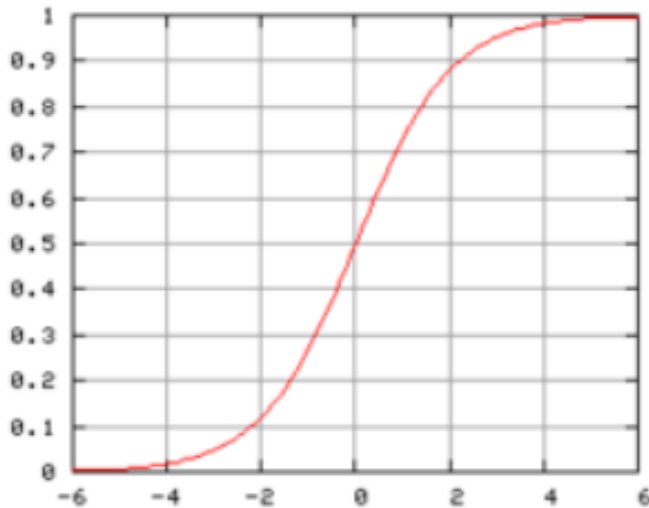


Рисунок 8 — Графік логістичної кривої

Взагалі існує багато різних функцій, докладно на яких ми не будемо зосереджуватись, але можемо привести їх довідково. Як наприклад серед інших використовують також гіперболічний тангенс, може бути нормована експоненційна функція, або функція ReLU або випрямленого лінійного вузла  $Y(x) = \max(0, x)$ , (1.7) що повертає  $x$ , якщо  $x$  додатній і  $0$  в іншому випадку, тож на практиці в деяких випадках може активуватися достатньо не часто.

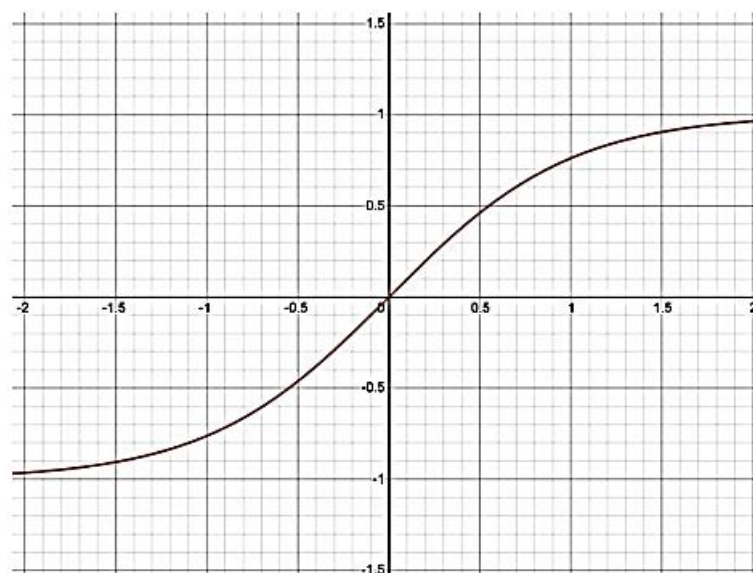


Рисунок 9 — Графік гіперболічного тангенсу

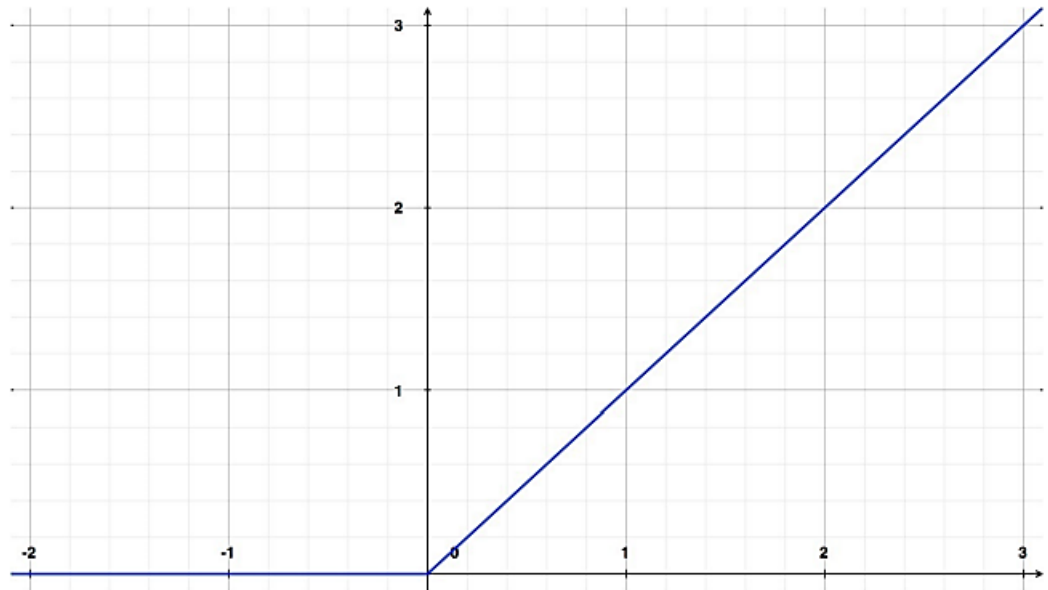


Рисунок 10 — Графік функції ReLU

Вочевидь, що постає питання вибору оптимальної конфігурації нейромережі. Що до вибору кількості шарів нейромережі та кількості нейронів у прихованих шарах необхідно пам'ятати наступне: максимальна кількість зв'язків у мережі буде обмежена об'ємом навчальної вибірки, якщо збільшуватися кількість зв'язків між нейронами то це збільшує можливості мережі до навчання. При нормальній або схожій швидкості навчання кращою слід вважати ту мережу, яка дає меншу похибку наближення, при близьких значеннях похибки наближення. Перевагу слід віддати мережі, яка навчається швидше, час, необхідний для навчання мережі буде завжди пропорційним складності нейромережі.

Для того, що б її було можливо застосовувати, нейромережу потрібно спочатку навчити. Це називають підлаштування параметрів мережі [9]. Навчання нейромережі включає в себе визначення параметрів (ваги та зсуви) моделі з використанням вхідних даних та відповідей (вихідних даних). Основні методи навчання включають:

- Навчання з учителем (Supervised Learning). В цьому випадку модель навчається на основі пар вхідних та відповідних вихідних даних.



Алгоритм спробує знайти оптимальну функцію відображення між вхідними та вихідними даними. Цей метод застосовується для класифікація та регресії.

- Ненавчане навчання (Unsupervised Learning). Ця модель навчається без використання відповідей. Зазвичай вона виявляє приховані структури або закономірності в наборі даних. Можна застосовувати для кластеризації, зменшення розмірності, генерації.

- Півнавчання (Semi-Supervised Learning). Подібна модель навчається на невеликому наборі даних з відповідями та на великому наборі даних без відповідей. Використовується для роботи з великими обсягами невідомих даних. Може застосовуватись у випадках класифікація та регресії.

- Підсилене навчання (Reinforcement Learning). Модель навчається взаємодіючи з оточенням та отримує винагороду або покарання на основі вибору дій. Агент вивчає оптимальні стратегії для досягнення мети. Застосовується для керування роботами, гри.

- Навчання з підтримкою вчителя (Transfer Learning). В цьому варіанті ми навчаємо модель на одному завданні, а отримані знання застосовуємо для поліпшення виконання на іншому, схожому завданні. Цей метод доцільно застосовувати тоді, коли є невеликий обсяг маркованих даних для нового завдання.

- Навчання за допомогою градієнтного спуску (Gradient Descent). Це метод оптимізації, коли оновлюються ваги моделі в напрямку, протилежному градієнту функції втрати. Існують різні варіанти, такі як стохастичний градієнтний спуск та міні-пакетний градієнтний спуск. Може застосовуватись до будь яких видів навчання.

- І, нарешті, навчання з ансамблем (Ensemble Learning). Тобто це використання кількох моделей для рішення одного завдання. Результат об'єднується для отримання кращої ефективності. Також може застосовуватись для всіх видів навчання.

Є і певні проблеми, які виникають при навчанні. Так при використанні тренувальних даних і навчанні з учителем виникає проблема вираховування та мінімізації похибки на основі саме тренувальних, а не реальних даних. Це призводить до так званої проблеми перенавчання. Сама проблема полягає в тому що коли збільшується складності моделі, то це хоча дає перевагу в збільшенні точності. Проте в деяких випадках складність може стати зовеликою і призвести до того, що модель не буде здатна до насправді точного прогнозування реальних даних, в той час як на тренувальних вона показуватиме дуже високу точність. Ключова причина появи проблеми перенавчання є зовелика кількість параметрів в моделі, що не відповідає наявному числу спостережень. Щоб знайти цю проблему використовують розподіл вихідних даних на тренувальні або навчальні та тестові. А вже після цього навчають нейромережу. Та якщо в процесі навчання похибка перестає зменшуватись або починає зростати то навчання припиняється. Для вирішення цієї проблеми використовують методи перехресного затвердження або крос-валідації, ранню зупинку, регуляризацію та порівнюють модель із результатами інших мереж за зміни параметрів мережі. Після цих експериментів із вирішення проблеми проводиться запуск моделі на тестовій вибірці для кінцевої оцінки її точності на реальних даних.

Загалом нейромережі володіють універсальністю та можливістю адаптації до різних завдань, що робить їх потужним інструментом у багатьох галузях, включаючи науку, медицину, фінанси та технології проте потребують уважного підходу при виборі моделі та її навчанні.

### **1.3 Сфери застосування прогнозів ВВП**

Прогнози ВВП відіграють важливу роль в різних галузях, допомагаючи розуміти та адаптуватися до економічних тенденцій та змін. Ось деякі докладні сфери застосування прогнозів ВВП:

- Економічне планування:

Уряди та підприємства використовують прогнози ВВП для розробки стратегій та планів розвитку. Це допомагає визначити необхідні ресурси, обсяг інвестицій та ризики.

- Макроекономічна політика:

Центральні банки та урядові органи використовують прогнози ВВП для прийняття рішень щодо грошової політики, податкової політики та інших макроекономічних заходів.

- Інвестиційне планування:

Підприємства та інвестори використовують прогнози ВВП для оцінки перспектив економічного розвитку та визначення потенційних областей для інвестицій.

- Фінансове планування:

Компанії використовують прогнози ВВП для розрахунку прибутку, планування бюджету та визначення фінансових цілей.

- Зовнішньоекономічна діяльність:

Країни та підприємства використовують прогнози ВВП для аналізу та планування міжнародної торгівлі та іноземних інвестицій.

- Соціальна політика:

Прогнози ВВП грають ключову роль у визначенні соціальних програм, рівня зайнятості та боротьби з бідністю.

- Фінансовий ринок:

Інвестори на фінансовому ринку використовують прогнози ВВП для управління портфелем та прийняття рішень щодо інвестицій у фондовий та облігаційний ринок.

- Страхування та ризиковий менеджмент:

Компанії використовують прогнози ВВП для оцінки ризиків та розробки стратегій управління ризиками.

Прогнози ВВП є ключовим інструментом для прийняття управлінських та стратегічних рішень в різних секторах економіки, допомагаючи суб'єктам господарювання адаптуватися до змін у економічному середовищі.

#### **1.4 Переваги використання нейронних мереж в порівнянні з традиційними методами прогнозування ВВП**

Нейронні мережі в сучасному економічному аналізі виступають ключовим інструментом для прогнозування ВВП, забезпечуючи значущий крок у напрямку точніших та адаптивних моделей [9]. Ми можемо докладніше розглянути переваги використання нейронних мереж у порівнянні з традиційними методами прогнозування.

Нейронні мережі, і особливо глибокі нейронні мережі (ГНМ), володіють надзвичайною гнучкістю та здатністю адаптуватися до складних економічних змін. Вони виявляються ефективними в умовах факторів що змінюються та в умовах непередбачуваних обставин.

Традиційні методи можуть стикається з обмеженнями у виявленні та моделюванні нелінійних взаємозв'язків, в той же час нейронні мережі виявляються відмінними, бо набагато краще справляються з обробкою нелінійних залежностей. Вони надійно враховують складні та неоднорідні ефекти, що впливають на ВВП.

Глибокі нейронні мережі здатні враховувати багатofакторність моделей, вони дозволяють інтегрувати значну кількість вхідних змінних, забезпечуючи можливість розглядати широкий спектр економічних показників. Це включає такі аспекти, як рівень інвестицій, споживання, зовнішньоекономічна діяльність, та інші.

Великі обсяги економічних даних давно стали стандартом у сучасному світі. Інколи їх обсяг стає таким, що його майже неможливо обробити чи проаналізувати. Але нейронні мережі здатні ефективно обробляти ці обсяги та

враховувати різноманітність інформації, що робить їх ідеальними для сучасних завдань аналізу.

Нейронні мережі здатні до роботи з послідовностями. Рекурентні нейронні мережі (RNN) використовуються для роботи з часовими рядами, враховуючи динаміку економічних процесів та забезпечуючи гнучку систему прогнозування.

Не завжди стандартна модель може розпізнати складні зв'язки. В той час як нейронні мережі виявляються ефективними в розпізнаванні та моделюванні складних економічних взаємозв'язків між різними групами змінних, що забезпечує здатність враховувати та аналізувати вплив різних аспектів економіки на ВВП.

Ще однією з особливостей нейронних мереж є універсальність їхнього застосування. Нейромережі можуть бути успішно використані в різних економічних галузях, включаючи фінанси, торгівлю, виробництво та послуги. Ця універсальність дозволяє їм знаходити застосування в різноманітних сценаріях прогнозування ВВП.

Але, хоча ми і зазначили ці переваги, важливо враховувати, що нейронні мережі не є універсальним рішенням для всіх завдань. Їхнє використання вимагає уважного налаштування та надання відповідних даних для навчання, але з правильним підходом вони можуть значно покращити точність та прогностичну здатність моделей прогнозування ВВП.

## **1.5 Висновки з розділу 1**

В даному розділі були розглянуті наступні питання:

1. Розглянута проблематика необхідності прогнозування економічних показників, зокрема Внутрішнього валового продукту країни (ВВП);
2. Зроблено огляд наявних сучасних методів рішення цієї задачі;
3. Окреслено сфери використання результатів прогнозування ВВП;
4. Доведено пріоритетність такого прогнозування за допомогою використання нейронних мереж.

## РОЗДІЛ 2 ДОСЛІДЖЕННЯ ОСНОВНИХ ТИПІВ ІСНУЮЧИХ НЕЙРОННИХ МЕРЕЖ

### 2.1 Персептрон

Персептрон — це проста форма штучного нейрона. Була введена Френком Розенблаттом в 1957 році [12], та стала основою для розвитку штучних нейронних мереж та машинного навчання взагалі. Персептрон допоміг вирішити проблеми бінарної класифікації та є попередником більш складних архітектур. Розглядається як базове поняття.

Персептрон складається з вхідного шару, прихованого шару та вихідного шару. Кожен вхід має вагу, яку регулюється під час навчання. Нейрони в прихованому шарі використовують функцію активації для обчислення вагової суми вхідних сигналів. Вихідний шар використовує функцію активації для визначення вихідного значення.

Процес персептрона можна розділити на чотири етапи:

Вхід – на цьому етапі персептрон отримує вхідні дані, які мають числові значення. Кожен вхід має свою вагу, яка представляє собою важливість цього вхідного сигналу.

Вагова сума - тут для кожного нейрона в прихованому шарі обчислюється вагова сума, яка представляється як скалярний добуток вхідних значень та їх ваг.

Функція активації - обчислена на минулому етапі вагова сума передається через функцію активації, яка визначає вихід нейрона. Популярною функцією активації є порогова функція Хевісайда або сигмоїда.

Навчання - ваги персептрона піддаються корекції під час навчання. Цей процес включає в себе порівняння вихідного значення з очікуваним значенням та відповідне оновлення ваг.

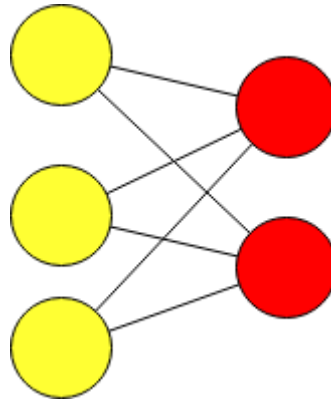


Рисунок 11 — Одношаровий перцептрон

Перцептрони використовуються для вирішення задач бінарної класифікації, де вони можуть визначати, до якого класу належить вхідний об'єкт. Однак їхні обмеження полягають в тому, що вони не можуть вирішувати складні проблеми, такі як нелінійна роздільність. Складніші архітектури, такі як мережі з передачею зворотного поширення, розроблені для вирішення цих проблем та виявлення більш складних залежностей в даних.

## 2.2 Мережі з передачею зворотного поширення (Feedforward Neural Networks)

Мережі з передачею зворотного поширення є однією з найпоширеніших та основних форм нейронних мереж у галузі машинного навчання. Вони визначаються відсутністю циклічних зв'язків між нейронами, це означає, що дані переміщуються лише в одному напрямку, від вхідного шару до вихідного. Виділяють наступні основні характеристики та принципи роботи цих мереж.

FNN складаються з трьох основних шарів: вхідного, прихованого і вихідного. Вхідний шар приймає вхідні дані, прихований шар оброблює ці дані, а вихідний шар генерує кінцевий результат.

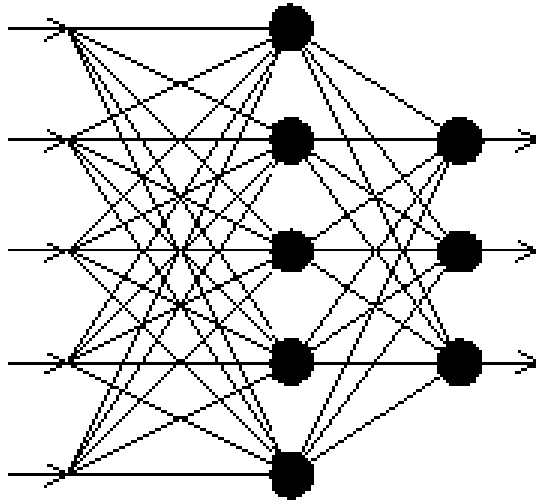


Рисунок 12 — Багатошаровий перцептрон

Кожен зв'язок між нейронами має вагу, яка представляє собою важливість цього зв'язку. Крім того, додається біас до вагової суми, щоб додати нейронам додаткову гнучкість. Під біасом мається на увазі константного члена, який додається до зваженої суми вхідних значень в кожному нейроні. Біас дозволяє нейронам виявляти гнучкість та адаптуватися до різних умов, що робить їх більш потужними та загальними в інтерпретації даних. Біас додається до суми зважених вхідних значень перед застосуванням функції активації. Це дозволяє нейрону вирізнитися навіть у випадку, коли всі вхідні значення рівні нулю. Без біасу мережа могла б бути обмеженою у своїй здатності апроксимувати різноманітні функції та залежності.

Біас допомагає моделі адаптуватися та узагальнювати на нові дані, що робить його важливим елементом в багатьох нейронних мережах та інших модулях машинного навчання.

Кожен нейрон у FNN має функцію активації, яка визначає його вихід на основі зваженої суми вхідних сигналів. Популярні функції активації включають сигмоїдальну, гіперболічну тангенс та ReLU [14]. Використовують алгоритми навчання, такі як зворотне поширення помилки, для оптимізації ваг і



покращення точності передбачень. Зворотне поширення включає в себе розповсюдження помилки від вихідного до вхідного шару з метою коригування ваг та біасів.

FNN застосовуються в різних областях, таких як класифікація, регресія, розпізнавання образів, обробка природної мови та багато інших завдань. Вони використовуються для розв'язання проблем, де важлива обробка статичних даних. Є досить простими та ефективними для багатьох завдань, але вони можуть бути обмеженими в обробці послідовностей та динамічних даних. Для більш складних завдань рекурентні нейронні мережі або згорткові нейронні мережі можуть бути більш підходящими.

FNN залишається важливим інструментом у світі нейромереж і є основою для більш складних архітектур, таких як глибокі нейронні мережі. Їх простота та ефективність дозволяють успішно використовувати їх для багатьох завдань у машинному навчанні.

### **2.3 Рекурентні нейронні мережі (Recurrent Neural Networks - RNN)**

Рекурентні нейронні мережі (RNN) представляють собою клас нейронних мереж, розроблений для роботи з послідовностями даних, де важливо враховувати контекст та залежності між різними частинами введення [14]. Одна з ключових особливостей RNN полягає в тому, що вони мають внутрішні зв'язки, які дозволяють їм зберігати попередні стани та використовувати їх для обробки нового введення.

Основні компоненти RNN:

Нейрони та Ваги - як і в інших нейронних мережах, RNN мають нейрони, які обробляють інформацію, та ваги, які регулюють сили зв'язків між нейронами.

Введення та Вихід - RNN приймає введення на кожному кроці часу, а вихід генерується. Це дозволяє їм обробляти послідовності будь-якої довжини та запам'ятовувати інформацію про попередні кроки.

Стани Нейрона (Cell State) - кожен нейрон у RNN має свій внутрішній стан, який слугує для зберігання попередньої інформації та передачі її наступним крокам.

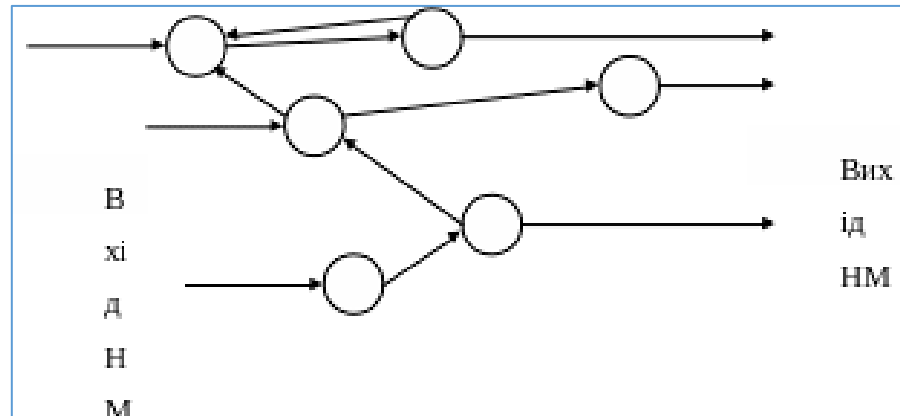


Рисунок 13 — Фрагмент рекурентної нейронної мережі

На кожному кроці часу RNN отримує введення та поточний стан, генеруючи вихід і оновлюючи свій стан для використання на наступних кроках. Це створює можливість засвоювати взаємозв'язки та контекст в послідовностях, які не завжди доступні для стандартних нейронних мереж.

Незважаючи на свою потужність, RNN можуть стикатися з проблемою зниклого та вибухаючого градієнту, коли градієнт може стати дуже малим або дуже великим, ускладнюючи навчання на великій кількості кроків.

RNN використовуються для обробки природної мови, генерації тексту, музичного моделювання, аналізу часових рядів та інших завдань, де важлива послідовність даних. Вони є потужним інструментом в галузі машинного навчання, здатним до ефективного моделювання та прогнозування на основі послідовностей даних. Незважаючи на труднощі, які вони можуть зустріти, вони залишаються ключовим інструментом в обробці послідовностей.

## 2.4 Мережі згорткового типу (Convolutional Neural Networks - CNN)

Мережі згорткового типу є досить потужним інструментом у галузі машинного навчання, зокрема та особливо в обробці та аналізі зображень. Ці мережі здатні ефективно виявляти та розпізнавати патерни, подібні до тих, що розпізнаються людським зором. Вони визначають новий рівень продуктивності в обробці великого обсягу візуальної інформації. Розглянемо їхню структуру, основні компоненти та застосування [12].

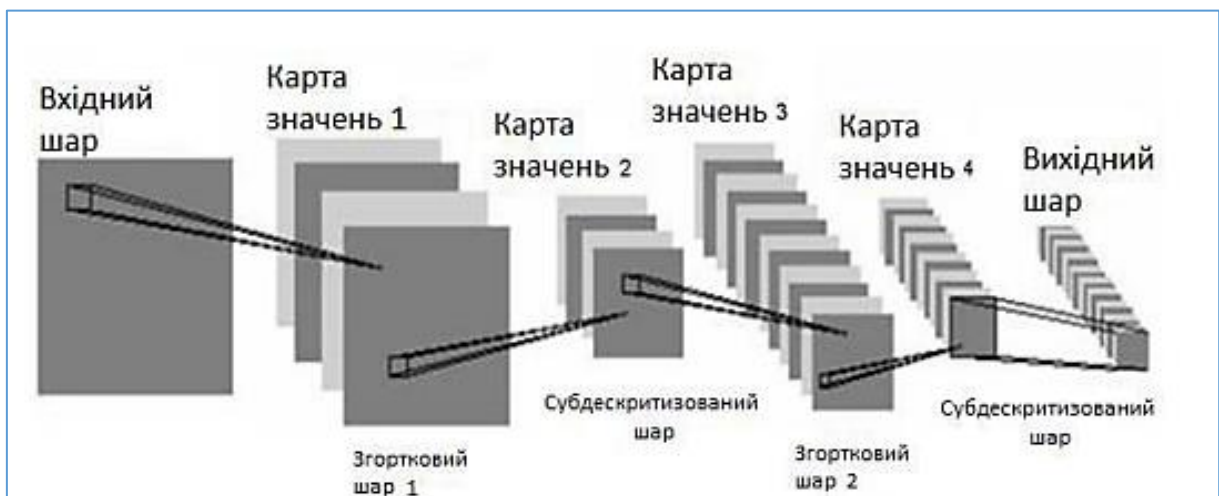


Рисунок 14 — Структура згорткової мережі

Структура CNN: CNN складається з трьох основних компонентів: вхідного шару, прихованих шарів та вихідного шару. Однак основна відмінність від звичайних мереж — використання шарів згортки та підсумовування.

Шари згортки (Convolutional Layers), використовують фільтри для виявлення локальних патернів у вхідних даних. Це дозволяє мережі автоматично вчитувати характеристики, такі як краї та текстури.

Шари підсумовування (Pooling Layers) зменшують просторові розміри представлень та кількість параметрів у мережі. Найпоширенішою технікою є максимальне підсумовування, де обирається максимальне значення в заданому регіоні.

Повністю зв'язані шари (Fully Connected Layers) відповідають за зіставлення вихідних патернів з класами та прийняття рішення про класифікацію.

Основними компонентами CNN є фільтри та ядро згортки. Фільтри виявляють певні характеристики у вхідних даних, а ядро згортки використовується для локальної взаємодії з областями вхідних даних.

CNN успішно використовуються в багатьох сферах, включаючи розпізнавання обличчя (можуть виявляти та класифікувати обличчя на зображеннях), розпізнавання об'єктів (CNN може ефективно розпізнавати об'єкти на зображеннях, що корисно в автономних автомобілях, медичних дослідженнях тощо), класифікація зображень (використовується для класифікації об'єктів на зображеннях), розпізнавання рукописного тексту (допомагають в розпізнаванні та класифікації написаного тексту).

Таким чином CNN відкриває широкі перспективи в обробці зображень, забезпечуючи високий рівень точності та швидкість в різних завданнях машинного зору. Завдяки їхній здатності ефективно використовувати локальні залежності та автоматично вивчати корисні функції, CNN стали ключовим елементом багатьох успішних застосувань в галузі комп'ютерного зору.

## **2.5 Мережі глибокого навчання (Deep Neural Networks – DNN)**

Мережі глибокого навчання (DNN) представляють собою різновид нейронних мереж, які включають в себе багато прихованих шарів, навіть декілька десятків або сотень. Ці мережі є ключовим елементом глибокого навчання і використовуються для вирішення складних завдань, таких як класифікація, регресія, аналіз зображень, обробка природної мови та інші.

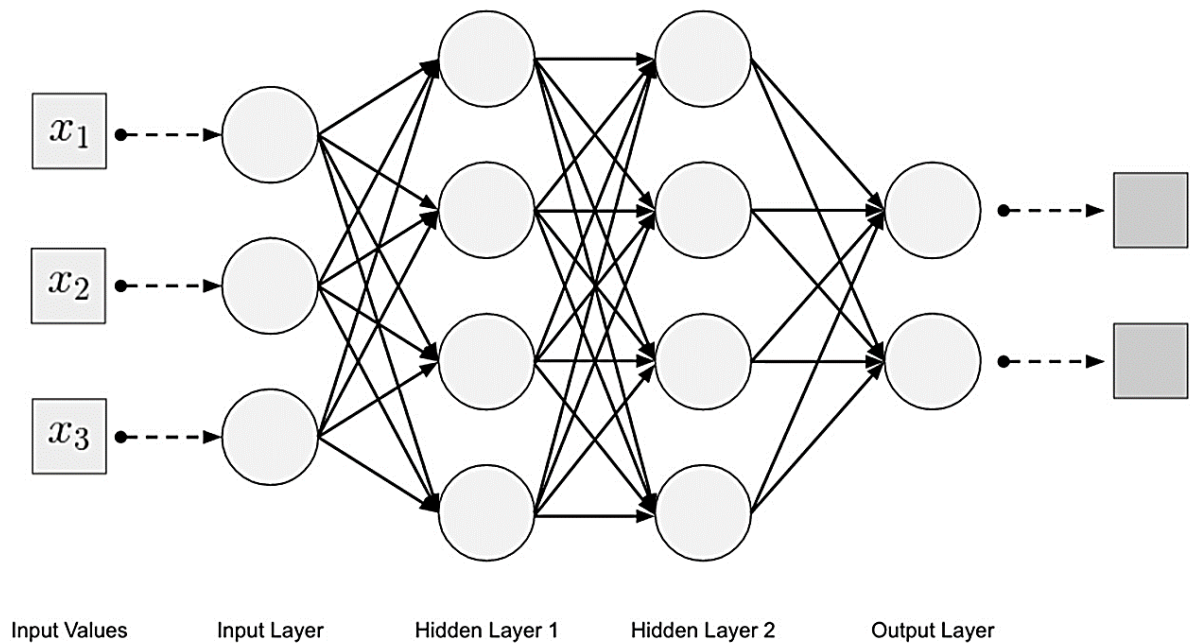


Рисунок 15 — Базова архітектура моделей глибокого навчання

Головна ідея глибокого навчання полягає в тому, щоб дозволити моделі автоматично вивчати корисні представлення вхідних даних на різних рівнях абстракції. DNN відрізняються від традиційних нейронних мереж тим, що вони мають значний обсяг шарів та параметрів, що робить їх дуже ефективними для виявлення складних закономірностей в даних.

Однією з ключових рис DNN є глибока структура, яка дозволяє їм виявляти високорівневі абстракції вхідних даних. Кожен шар може розглядатися як абстракція певного рівня, інтегруючи більш високорівневі концепції для здійснення більш точних та складних прогнозів.

Глибокі нейронні мережі навчаються за допомогою методу зворотного поширення помилок, який вимагає великої кількості даних для тренування. Процес тренування полягає в налаштуванні ваг та параметрів моделі таким чином, щоб максимізувати її здатність визначати закономірності у вхідних даних.

Однією з переваг DNN є їхня здатність розпізнавати складні та абстрактні залежності, такі як обличчя на фотографіях або семантичне значення слів

в тексті. Це робить їх особливо ефективними для завдань, де важливі внутрішні представлення даних.

Проте, тренування DNN може вимагати значних обчислювальних ресурсів та тривати тривалий час. Крім того, може виникнути проблема перенавчання, коли модель надто точно відповідає тренувальним даним і не здатна ефективно генералізувати на нові дані.

Загалом, глибокі нейронні мережі відіграють ключову роль у досягненнях глибокого навчання, їх використання розширюється у багатьох галузях, від обробки зображень та мовного процесування до біотехнологій та фінансів.

## 2.6 Самоорганізуючі карти (Self-Organizing Maps - SOM)

Самоорганізуючі карти — це вид нейронних мереж, розроблений фінським вченим Теуво Кохоненом в 1980-х роках. SOM відзначаються здатністю до безнавчального вивчення, упорядкування та візуалізації великих обсягів даних у низькорозмірних просторах.

Основний принцип SOM полягає в тому, щоб відобразити багатовимірні дані на двовимірну або тривимірну сітку нейронів, зберігаючи при цьому топологічну структуру вхідного простору. В процесі навчання SOM вчаться розподіленню простору так, щоб схожі дані вихідного простору відображалися на схожих областях SOM.

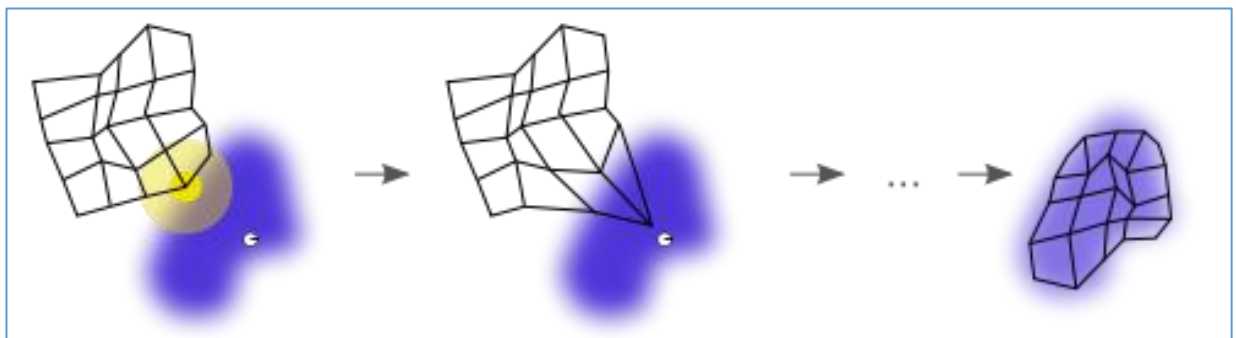


Рисунок 16 — Навчання самоорганізуючої карти

Архітектура SOM складається з великої кількості вузлів, відомих як нейрони чи вузли карти. Кожен нейрон має вагові вектори, які ініціалізуються випадковим чином. Під час навчання вагові вектори змінюються, спрямовуючись до зон вхідного простору з високою концентрацією даних.

Процес навчання SOM включає дві основні фази: конкурентний та кооперативний етапи. Під час конкурентного етапу кожен нейрон змагається за вагову вектор для введеного сигналу, визначаючи нейрон, який є найбільш схожим з вхідним сигналом. Після цього настає кооперативний етап, де вагові вектори обраного нейрона та його сусідніх нейронів оновлюються, здійснюючи адаптацію до вхідних даних.

Цікаві аспекти та властивості самоорганізуючих карт (SOM), що важливі для розуміння їхнього функціонування та застосувань:

Топологічна збереженість: SOM здатні зберігати топологічну структуру вхідного простору в своєму вихідному представленні. Це означає, що близькі за значенням вхідні дані будуть також близькими у вихідному представленні.

Безнавчальне вивчення: Однією з ключових характеристик SOM є їхня здатність до безнавчального вивчення, що означає, що вони можуть самостійно адаптуватися до структури вхідних даних без учителя.

Геометрична структура: SOM можуть мати різні геометричні форми, такі як прямокутні, шестикутні, або інші. Обрана геометрія впливає на їхню здатність виявляти та відображати структуру вхідних даних.

Низькорозмірний вихідний простір: SOM можуть проектувати великі обсяги вхідних даних на низькорозмірний вихідний простір, що полегшує візуалізацію та аналіз даних.

Інтерпретація вихідних векторів: Вихідні вектори SOM можуть інтерпретуватися як функції активації нейронів у вихідному шарі, що може допомагати в розумінні, як нейрони реагують на вхідні стимули.

SOM використовуються для вирішення завдань кластеризації, асоціативного вивчення, аналізу даних та візуалізації великих обсягів інформації. Їх

ефективність виявляється у тому, що вони здатні зберігати топологічну структуру вхідних даних у вихідному просторі, навіть при обробці великих та складних наборів даних. SOM застосовуються у багатьох галузях, включаючи аналіз зображень, обробку природних мов, економічний аналіз та багато інших.

## 2.7 Енкодер - декодер (Autoencoder)

Енкодер-декодер, відомий також як autoencoder, представляє собою тип нейронної мережі, який зазвичай використовується для зменшення розмірності даних та вивчення ефективного кодування. Ця архітектура стала ключовою в області представлення даних та компресії інформації.

Структура Енкодера-Декодера наступна - Енкодер та Декодер — це дві частини мережі. Енкодер відповідає за перетворення вхідних даних у менший обсяг, який називається кодом або представленням. Декодер, з іншого боку, відповідає за відновлення вихідних даних з цього коду.

Енкодер використовується для витягування ключових ознак або представлення, яке зберігає суттєву інформацію про вхідні дані. Ця операція може зменшити розмірність даних, зберігаючи при цьому важливі характеристики.

В свою чергу Декодер відновлює вхідні дані з отриманого коду. Його завдання полягає в реконструкції оригінального вхідного сигналу з отриманої абстракції.

В чому суть Автоенкодера - Автоенкодер старається зробити реконструкцію якнайточнішою, максимізуючи подібність між вхідними та відновленими даними. Ця функціональність допомагає вивчати внутрішнє представлення, яке затребуване для відтворення вхідних даних.



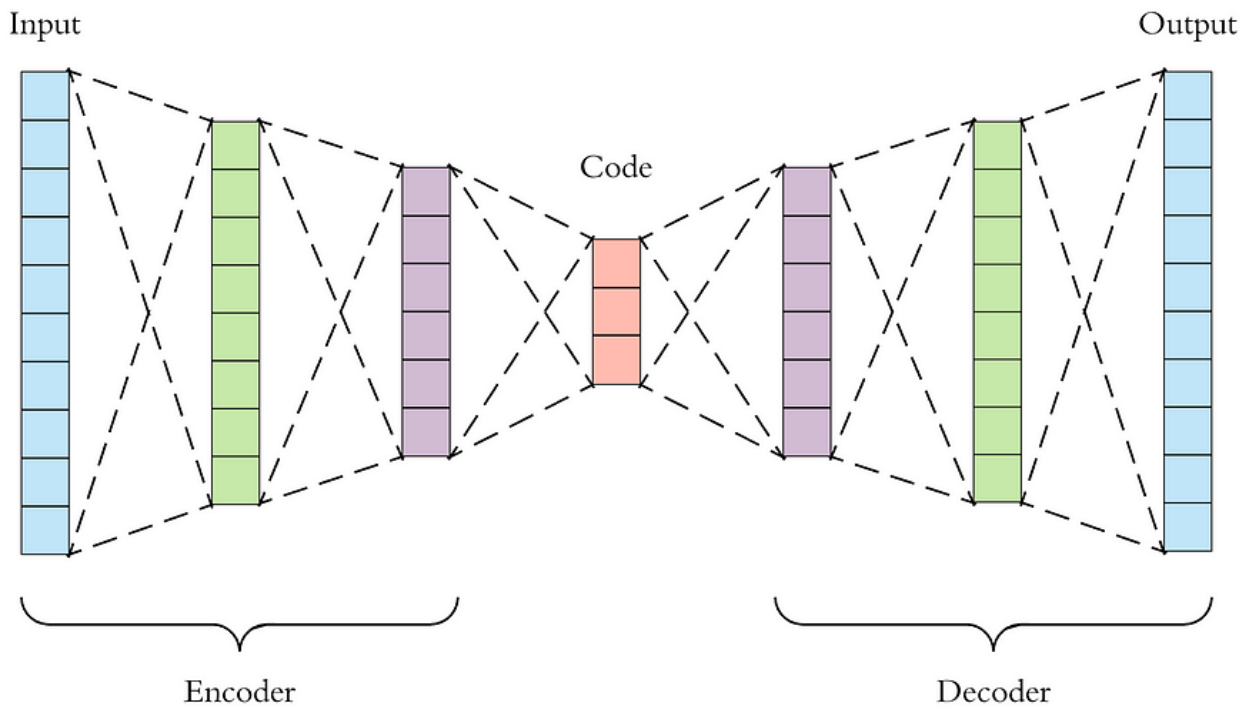


Рисунок 17 — Схема роботи Енкодера-декодера

Якщо намагатися коротко представити це:

**Енкодер:**

Вхідний шар: Приймає вхідні дані.

Прихований шар (код): Використовується для витягування ключових ознак та створення представлення.

**Декодер:**

Прихований шар: Отримує код і використовує його для створення внутрішнього представлення.

**Вихідний шар:** Повертає реконструйовані дані.

Одним із ключових застосувань енкодера-декодера є зменшення розмірності даних. Це корисно в ситуаціях, коли потрібно представити велику кількість ознак або зменшити обсяг даних для подальшого аналізу. Крім того Автоенкодери можуть використовуватися для навчання без нагляду, коли намагаєтьсямо вивчити репрезентацію даних без чітких міток класів, що робить їх

ефективними для роботи з невирішеними проблемами. Завдяки цьому Енко-дери-Декодери часто використовуються у глибокому навчанні, включаючи генерацію зображень, анаморфоз та інші завдання.

В цілому, енкодери-декодери є потужним інструментом у сфері глибокого навчання, зокрема для зменшення розмірності даних та отримання їхнього представлення. Вони дозволяють ефективно вивчати інформацію та використовувати її для різних завдань.

## **2.8 Мережі довготривалої пам'яті (Long Short-Term Memory - LSTM)**

Мережі довгострокової пам'яті (LSTM) є розширенням рекурентних нейронних мереж (RNN), призначеним для вирішення проблеми зниклого градієнта та забезпечення моделі здатності вивчати та враховувати довгострокові залежності в даних. Цей тип мережі особливо корисний при роботі з послідовністю даних, таких як мовлення, музика, часові ряди тощо.

LSTM-модулі розроблені спеціально, щоб уникнути проблеми довготривалої залежності, запам'ятовуючи значення як на короткі, так і довгі проміжки часу [11]. Це можна пояснити тим, що LSTM-модуль не використовує функцію активації всередині своїх рекурентних компонентів. Таким чином, значення, що зберігається, не розмивається в часі і градієнт не зникає при використанні методу зворотного поширення помилки в часі при тренуванні мережі.

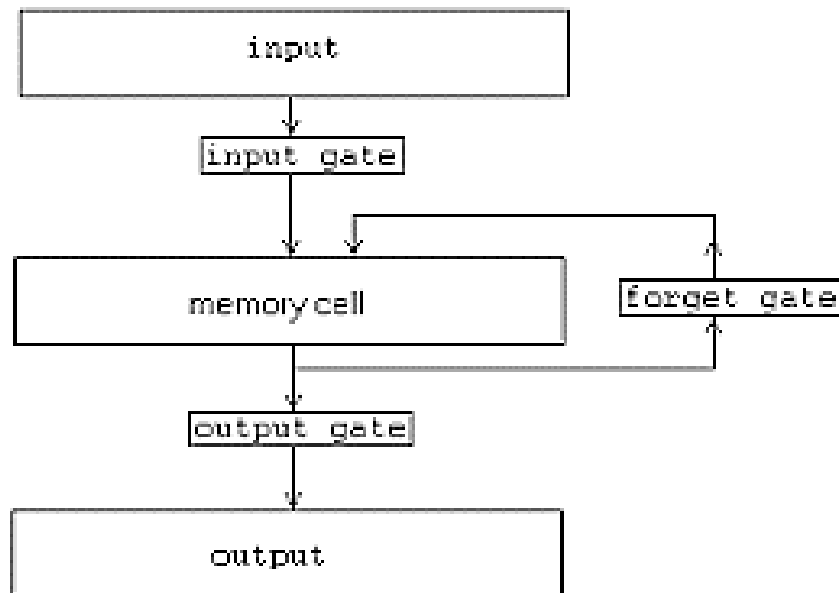


Рисунок 18 — Принципова схема LSTM

Ключові компоненти LSTM-модуля: стан комірки та різні фільтри (gates). Про стан комірки можна говорити як про пам'ять мережі, яка передає відповідну інформацію по всьому ланцюжку модулів. Таким чином, навіть інформація з ранніх тимчасових кроків може бути отримана на пізніших, нівелюючи ефект короткочасної пам'яті. Отже, ключовою особливістю є використання комірок пам'яті та механізму воріт (фільтри) для регулювання потоку інформації через мережу.

#### 1. Входи, виходи та приховані стани:

LSTM має вектор входів, який представляє поточний вхід до мережі. Кожен елемент вектору відповідає певному аспекту вхідних даних. Є також вектор виходів, який представляє вихідні дані. Прихований стан є внутрішнім станом мережі, який зберігає попередню інформацію.

#### 2. Комірки пам'яті:

Головною інновацією LSTM є використання комірок пам'яті, які можуть зберігати інформацію на тривалий час. Це дозволяє мережі зберігати та використовувати важливі взаємозв'язки в довгострокових залежностях.

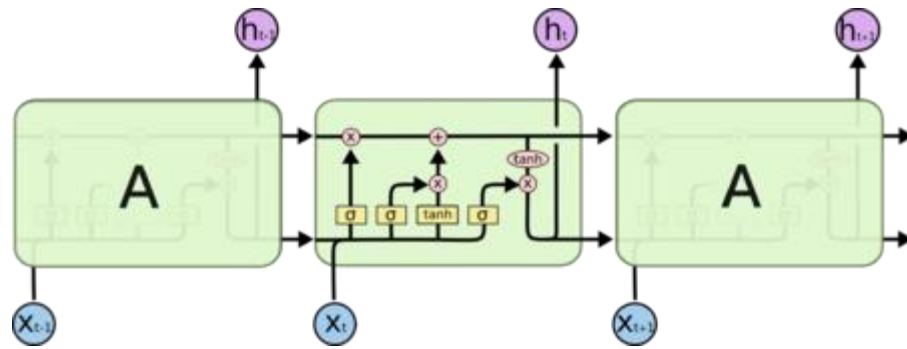


Рисунок 19 — Схема шарів LSTM

### 3. Ворота:

LSTM включає три типи воріт: ворота забування (forget gate), ворота входу (input gate) та ворота виходу (output gate). Вони регулюють потік інформації в, з та з комірки пам'яті відповідно. Це дозволяє мережі вирішувати, яку інформацію тримати, додавати або витіснити.

#### 3.1 Ворота забування (Forget Gate):

Визначає, яку інформацію треба забути з комірки пам'яті на підставі поточного входу та попереднього стану.

#### 3.2 Ворота входу (Input Gate):

Визначає, яку інформацію треба додати до комірки пам'яті, враховуючи поточний вхід та попередній стан.

#### 3.3 Ворота виходу (Output Gate):

Визначає, яку частину інформації з комірки пам'яті вивести як вихід мережі.

Комірка пам'яті здатна утримувати інформацію протягом тривалого часу, що робить LSTM особливо корисним для вирішення завдань, де важливі довгострокові залежності. Загалом, використання LSTM дозволяє нейронній мережі ефективно враховувати довгострокові залежності в послідовностях даних, забезпечуючи високий рівень гнучкості та точності у вирішенні різних завдань.

## 2.9 Висновки з розділу 2

В цьому розділі були розглянуті основні типи нейронних мереж та зроблені наступні висновки:

1. Існує дуже багато типів нейронних мереж, які мають як свої переваги, так і свої недоліки;
2. Визначено, що для рішення поставленої задачі прогнозування краще використовувати модель на базі мережі довготривалої пам'яті (LSTM).

## РОЗДІЛ 3 РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ ДЛЯ ПРОГНОЗУВАННЯ ВВП

### 3.1 Налаштування апаратного середовища

Для вирішення поставленої задачі було обрано ноутбук Lenovo з наступними основними характеристиками:

Назва ОС Майкрософт Windows 11 Home

Версія 10.0.22621 Збірка 22621

Тип x64-based PC

Процесор AMD Ryzen 5 5600H with Radeon Graphics, 3301 МГц, ядер 6, логічних процесорів 12

Відеокарта NVIDIA GeForce GTX 1650

Установлена фізична пам'ять (ОЗП) 16,0 ГБ

Усього віртуальної пам'яті 21,1 ГБ

Розмір файлу довантаження 7,25 ГБ

### 3.2 Підготовка робочого DataSet-у

Правильний вибір датасету та його оптимізація грають ключову роль у створенні надійної та точної моделі прогнозування.

Створення датасету для прогнозування ВВП вимагає доступу до відповідних економічних даних, які мають бути достовірними та перевіреними. Наводимо кілька ресурсів та джерел, де можна знайти верифіковані дані для створення датасету для прогнозування ВВП:

**Світовий банк (World Bank).** World Bank Data: World Bank надає доступ до широкого спектру економічних індикаторів та статистики для багатьох країн.

The screenshot shows the World Bank DataBank interface. The main content is a table titled 'Preview' for the indicator 'GDP (current US\$)'. The table lists data for 266 countries from 2015 to 2022. The countries listed include Ukraine, United Arab Emirates, United Kingdom, United States, Uruguay, Uzbekistan, Vanuatu, and Venezuela, RB. The table also includes a 'World' row at the bottom. The interface includes a search bar, a 'Preview' tab, and a 'Download options' menu.

	2015	2016	2017	2018	2019	2020	2021	2022
Ukraine	91,030,968,761	93,355,868,977	112,090,503,81	130,891,086,68	153,883,045,52	156,617,719,78	199,765,856,764.5	160,502,739,23
United Arab Emirates	370,275,469,57	369,255,326,23	390,516,804,03	427,049,432,15	417,989,721,74	349,473,015,33	415,021,590,687.5	507,534,921,71
United Kingdom	2,934,857,946,2	2,699,659,680,5	2,683,488,510,5	2,878,152,147,3	2,857,057,847,9	2,704,609,160,0	3,122,480,345,924.5	3,070,667,732,3
United States	18,206,020,741	18,695,110,842	19,477,336,549	20,533,057,312	21,380,976,119	21,060,473,613	23,315,080,560,000.0	25,462,700,000
Uruguay	57,680,328,702	57,480,788,380	65,006,047,680	65,203,071,110	62,048,585,618	53,666,908,479	61,412,268,248.9	71,177,146,197
Uzbekistan	86,196,264,741	86,138,288,633	62,081,322,741	52,870,108,217	60,283,503,705	60,224,701,295	69,600,614,988.6	80,391,853,884
Vanuatu	730,870,805.2	780,889,605.9	880,061,831.0	914,736,985.4	936,526,267.6	909,421,043.5	971,636,097.9	983,582,864.6
Venezuela, RB	..	..	..	..	..	..	..	..
World	730 780 340 071	757 096 004 172	804 282 676 410	810 476 473 64	824 265 367 071	846 616 760 161	966 437 600 717.0	808 003 370 061

Рисунок 20 — Сторінка Світового Банку

## Міжнародний Валютний Фонд (International Monetary Fund - IMF).

IMF Data: також надає доступ до своєї бази даних, яка включає економічні дані для різних країн.

The screenshot shows the IMF Data website. The main content is a section titled 'IMF FINANCES'. It includes a search bar, a navigation menu, and a main content area with 'SDR Rates Information' and 'Query Tools'. The 'SDR Rates Information' section includes 'CURRENCY AMOUNTS', 'INTEREST RATES', and 'EXCHANGE RATES'. The 'Query Tools' section includes 'Financial Data (cross country)', 'Financial Data (by country)', 'Exchange Rates', 'Interest Rates', 'Daily SDR Valuation', and 'Weekly SDR Interest Rate Calculation'. The page also features a 'GENERAL INFORMATION' section with 'FINANCIAL DATA', 'IMF LENDING', 'FINANCIAL REPORTS', and 'RESOURCES'.

Рисунок 21 — Сторінка Міжнародного Валютного Фонду

**Національні Статистичні Служби.** Багато країн мають національні статистичні служби, які публікують офіційні дані щодо економічних показників, включаючи ВВП. Наприклад, для США це Bureau of Economic Analysis (BEA).

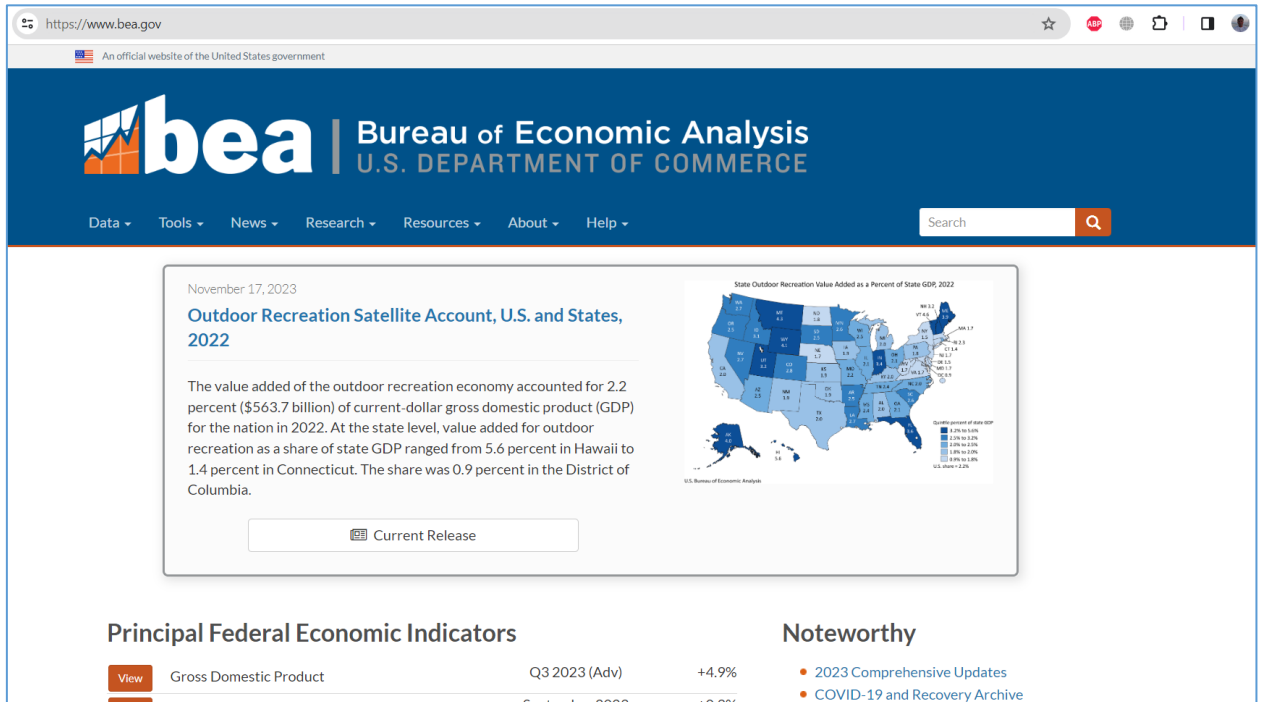


Рисунок 22 — Сторінка Бюро Економічного Аналізу США

### Організація економічного співробітництва та розвитку (OECD):

OECD Data: OECD надає дані щодо економічних показників для своїх членів та інших країн.



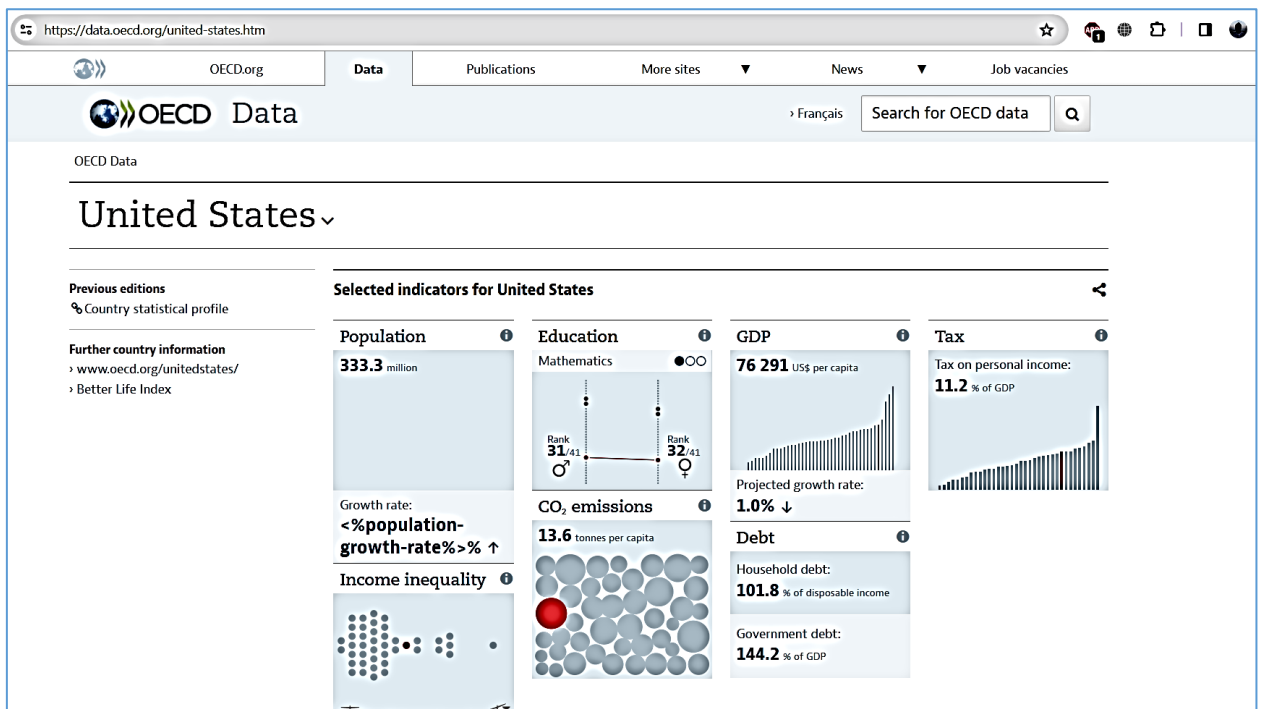


Рисунок 23 — Сторінка Організації економічного співробітництва та розвитку

**Національні Банки.** Часто національні банки також публікують економічні дані та статистику для своїх країн. У випадку цієї роботи наводиться приклад сторінки Bank of America, так як сам Датасет створюється на основі економічних даних США.

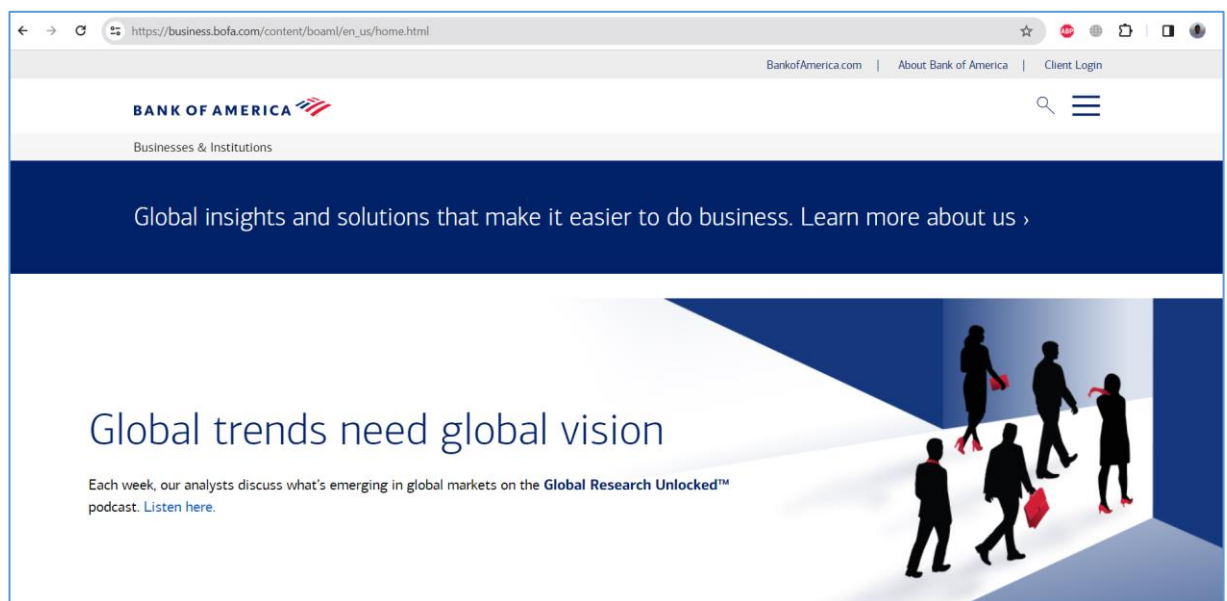


Рисунок 24 — Сторінка Bank of America

**Інші Економічні Організації.** Інші міжнародні та регіональні економічні організації можуть також надавати важливі дані для прогнозування ВВП.

Окрім того можна користуватись готовими датасетами, які потребують мінімальної подальшої обробки, якщо є довіра до джерела та автору інформації.

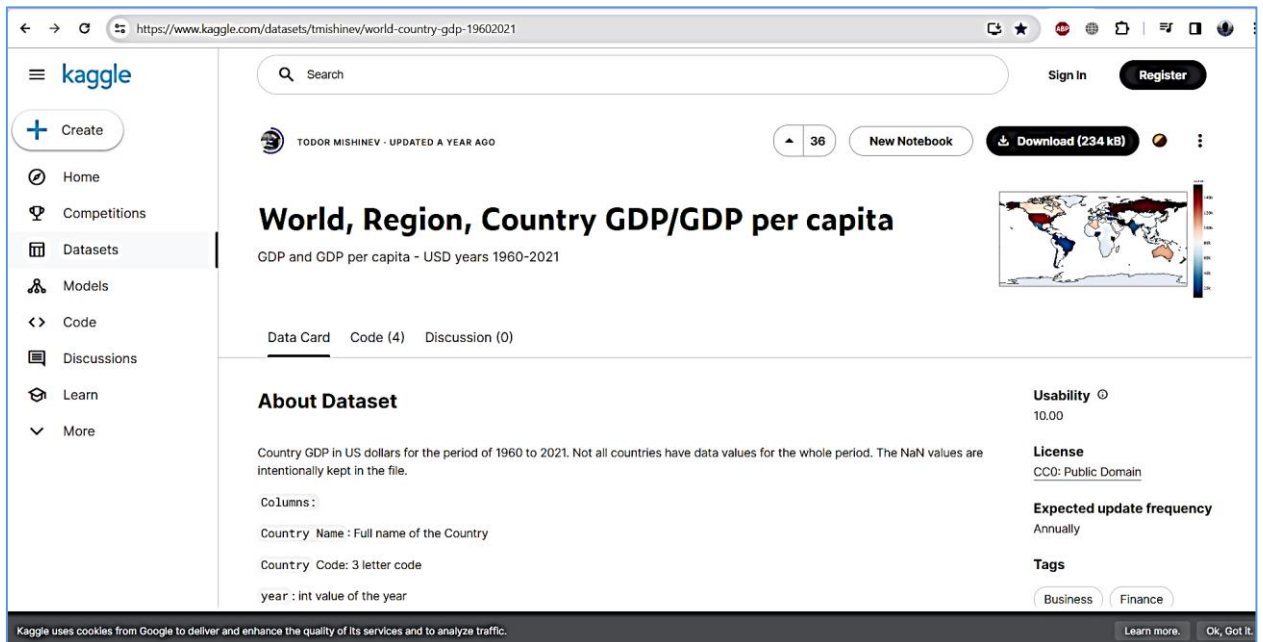


Рисунок 25 — Сторінка сайту датасетів Kaggle

Ще деякі основні принципи вибору датасету у відповідності до поставленої задачі:

1. Для точного прогнозування ВВП важливо враховувати географічний обсяг датасету. Деякі моделі можуть бути специфічними для певних регіонів, тому важливо вибрати датасет, який відображає економічні особливості конкретного регіону чи країни. В нашому випадку це будуть, як вже було зазначено, Сполучені Штати Америки. Чому саме США. Це єдина країна, яка спромоглася надати Світовому Банкові (саме інформація цього джерела стала базовою при створенні датасету) практично вичерпний об'єм економічних даних у тривалому періоді.

2. Датасет повинен містити різноманітні типи даних, необхідних для прогнозування ВВП. Це можуть бути дані про виробництво, зайнятість, інвестиції, споживчі витрати, торгівельний баланс та інші економічні показники. Розмаїття даних дозволяє моделі краще усвідомити економічну динаміку. Створений датасет містить 455 різних економічних показників за 51 рік з 1970 по 2021 роки.

Проблему відсутніх дані яких не вистачало з різних причин, таких як технічні проблеми, зміни методів збору даних чи економічні кризи було вирішено шляхом обробки відсутніх даних, використовуючи такі методи, як інтеполяція та заміна значень на основі статистичних показників (за можливостю). Надалі ці дані було оптимізовано до одного масштабу та типу даних, що мало поліпшити швидкість навчання моделі та забезпечити більш стабільні результати.

Далі датасет було поділено на навчальний та тестовий набори у стандартному співвідношенні 80/20. Процес валідації моделі буде описано пізніше на етапі реалізації моделі.

### **3.3 Вибір програмного середовища, бібліотек та фреймворків**

Для рішення поставленої задачі було вибрано наступні інструменти:

- Програмне середовище: Anaconda, jupyter notebook, spyder;
- Фреймворкі: TensorFlow, Keras;
- Бібліотеки: scikit-learn, NumPy, Pandas та Matplotlib.

На етапі реалізації проекту виникли проблеми з тим, що відеокарта обраного ноутбуку (а іншого достатньо швидкого комп'ютеру просто немає) не підтримує роботу з TensorFlow. Для вирішення цієї проблеми було вирішено скористатись можливостями Anaconda.

#### **Anaconda**

Anaconda – це дистрибутив мов програмування Python і R, який містить в собі набір популярних вільних бібліотек, об'єднаних проблематикою науки

про дані та машинне навчання. Основною метою є надати єдиним узгодженим пакетом найбільш затребуваних колом користувачів, що займаються рішенням питань машинного навчання та науки про дані тематичних модулів (таких як NumPy, SciPy, Astropy та інших) з вирішенням залежностей і конфліктів, що виникають, які неминучі при одиночній установці. Дистрибутив містить понад 2,0 тисячі модулів.

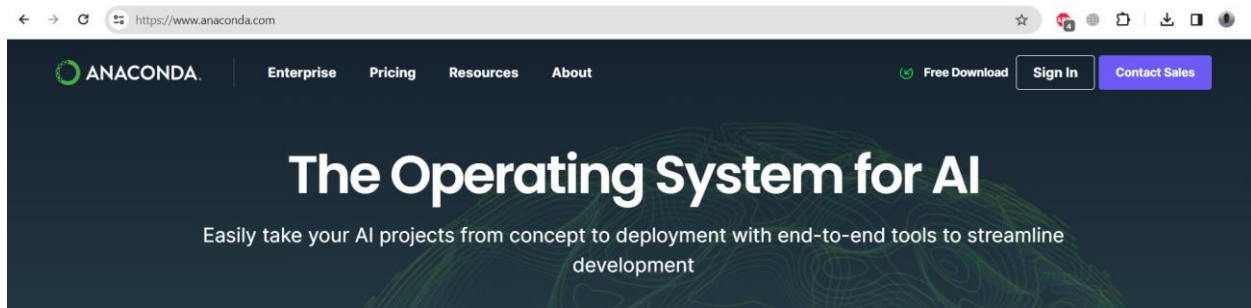


Рисунок 26 — Головна сторінка офіційного сайту Anaconda

Основні риси Anaconda:

**Пакунковий менеджер (Conda):** Anaconda включає в себе пакунковий менеджер Conda, який дозволяє легко встановлювати, оновлювати та керувати пакунками та їх залежностями. Conda дозволяє швидко налаштувати робоче середовище з необхідними бібліотеками.

**Середовища (Environments):** За допомогою Anaconda можна створювати віртуальні середовища, що дозволяє ізолювати проекти та їхні залежності. Це корисно для уникнення конфліктів між версіями бібліотек у різних проектах.

**Широкий вибір бібліотек:** Anaconda поставляється з великою кількістю популярних бібліотек для науки про дані, таких як NumPy, Pandas, Matplotlib, SciPy, Scikit-learn та інші. Це робить Anaconda ідеальним вибором для роботи з даними та розвитку машинного навчання.

**Cross-Platform:** Anaconda підтримується на різних операційних системах, таких як Windows, macOS, і Linux, що робить його універсальним і доступним для багатьох користувачів.

Spyder та Jupyter Notebooks: Anaconda містить інтегроване середовище розробки Spyder та підтримує використання Jupyter Notebooks, що полегшує вивчення, тестування та візуалізацію результатів.

## Spyder

Spyder — це потужне наукове середовище, написане на Python для Python і розроблене вченими, інженерами та аналітиками даних. Він має унікальне поєднання розширених функцій редагування, аналізу, налагодження та профілювання комплексного інструменту розробки з дослідженням даних, інтерактивним виконанням, глибокою перевіркою та чудовими можливостями візуалізації наукового пакету.

Крім того, Spyder пропонує вбудовану інтеграцію з багатьма популярними науковими пакетами, зокрема NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy тощо.

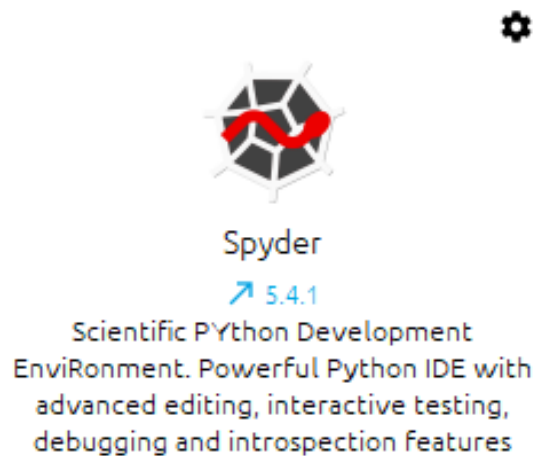


Рисунок 27 — Іконка Spyder в Anaconda Navigator

Основні характеристики Spyder:

- Редактор коду:

Spyder має вбудований редактор коду з можливістю виділення синтаксису, автодоповненням, перевіркою наявності помилок та іншими корисними функціями.

- IPython консоль:

Інтерактивна консоль IPython вбудована безпосередньо в середовище Spyder, що дозволяє вам виконувати код по чергово та отримувати результати в режимі реального часу.

- Об'єктна інспекція:

Spyder надає можливість переглядати значення змінних та об'єктів в режимі виконання, що спрощує налагодження та розробку.

- Простеження за виконанням коду:

Інструменти простеження за виконанням коду дозволяють вам крок за кроком виконувати програму та перевіряти проміжні результати.

- Менеджер змінних та середовища:

Spyder дозволяє легко переглядати та керувати змінними та середовищем роботи.

- Підтримка пакетів та наукових бібліотек:

Spyder входить до складу дистрибутиву Anaconda, який містить велику кількість наукових бібліотек та інструментів для роботи з даними.

- Візуальне програмування:

Spyder надає можливість створювати візуальні графічні елементи для програмування та візуалізації даних.

- Інтеграція з Jupyter:

Інтеграція з Jupyter дозволяє використовувати ноутбуки Jupyter безпосередньо в Spyder.

- Окремо треба відмітити де що:

Крім багатьох вбудованих функцій, Spyder можна розширити ще за допомогою плагінів сторонніх виробників. Spyder також можна використовувати як бібліотеку розширення PyQt5, дозволяючи розвивати її функціональність і вбудовувати її компоненти, такі як інтерактивна консоль або розширений редактор, у власне програмне забезпечення.

## Jupyter Notebook

Jupyter Notebook це інтерактивне середовище для програмування та аналізу даних, яке дозволяє об'єднувати код, текстові описи, графіки та інші мультимедійні елементи в одному документі. Він підтримує багато мов програмування, таких як Python, R, і Julia, і дозволяє вам створювати та ділитися документами, що містять живий код, різноманітні візуалізації, текстові пояснення та інші елементи.

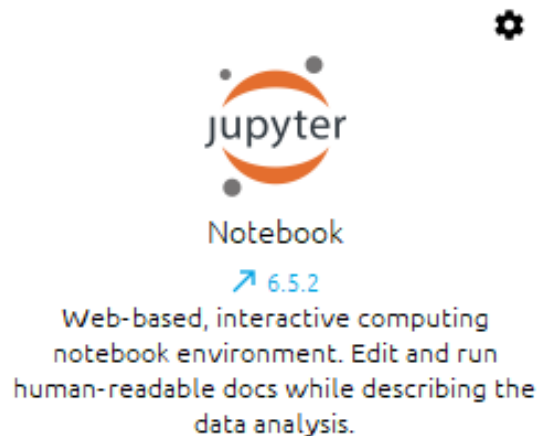


Рисунок 28 — Іконка Jupyter Notebook в Anaconda Navigator

Основні можливості Jupyter Notebook:

- Кодові Комірки:

Розробник можете вводити та виконувати код в окремих комітках. Це дозволяє виконувати код по частинам і виводити результати між ними.

- Текстові Комірки:

Розробник можете додавати текстові комірки, використовуючи Markdown або HTML, щоб надати пояснення до свого коду, описати алгоритми, чи вставити зображення.

- Візуалізація:

Jupyter підтримує вбудовані візуалізації, які можуть бути відображені прямо в документі. Таким чином можливо використовувати бібліотеки, такі як Matplotlib, Seaborn, Plotly для створення графіків та діаграм.

- Інтерактивні Віджети:

Jupyter надає можливості для створення інтерактивних віджетів, які дозволяють розробнику маніпулювати параметрами та спостерігати за змінами в реальному часі.

- Підтримка Різних Мов:

Хоча Jupyter спочатку був розроблений для мови програмування Python, тепер він підтримує багато інших мов, таких як R, Julia, Scala та інші.

Jupyter Notebook поширено в області аналізу даних, машинного навчання, наукових досліджень і навчання. Він забезпечує інтерактивну роботу з кодом і дозволяє зберігати результати в одному документі, що полегшує роботу та спільну роботу над проектами.

### **TensorFlow**

TensorFlow — це відкрите програмне забезпечення для чисельних обчислень, яке спеціально призначене для розвитку та навчання глибоких нейронних мереж. Воно розроблене командою Google Brain і стало однією з найпопулярніших бібліотек для машинного навчання та глибокого навчання.



Рисунок 29 — Логотип TensorFlow

Основні характеристики TensorFlow:

- Граф обчислень (Computational Graphs):

TensorFlow використовує графи обчислень для представлення обчислень. Граф обчислень - це набір взаємозв'язаних операцій у вигляді графа. Ко-



жен вузол у графі представляє операцію, а ребра - дані, що проходять між операціями. Це дозволяє TensorFlow автоматично оптимізувати та розподіляти обчислення для ефективного використання ресурсів.

- Тензори:

Основними об'єктами у TensorFlow є тензори. Тензор - це багатовимірний масив, який може бути константою або змінюваною змінною.

- Можливості обчислення на багатьох пристроях:

TensorFlow підтримує обчислення на різних пристроях, таких як CPU, GPU, а також на спеціалізованих апаратних пристроях, таких як Tensor Processing Units (TPU).

- Вбудовані інструменти для візуалізації:

TensorFlow надає інструменти для візуалізації графів обчислень та моніторингу навчання моделей.

- Високорівневі API для легшого використання:

TensorFlow надає високорівневі API, такі як Keras, які спрощують розробку моделей та роблять код більш зрозумілим та лаконічним.

- Модульність та розширюваність:

TensorFlow є модульним та розширюваним, що дозволяє розробникам використовувати лише ті частини, які їм необхідні.

- Спільнота та підтримка:

TensorFlow має велику та активну спільноту, що допомагає розвивати та вдосконалювати цю бібліотеку, яка широко використовується в індустрії та дослідницьких проєктах для розв'язання завдань машинного навчання, включаючи класифікацію, регресію, виявлення об'єктів, обробку мови, генерацію тексту та багато інших.

## **Keras**

Keras - це високорівневий інтерфейс для роботи з нейронними мережами, який базується на інших бібліотеках, таких як TensorFlow та Theano. Він спрощує процес розробки нейронних мереж, роблячи його більш доступним

для початківців та зменшуючи кількість коду, необхідного для вирішення типових завдань машинного навчання.

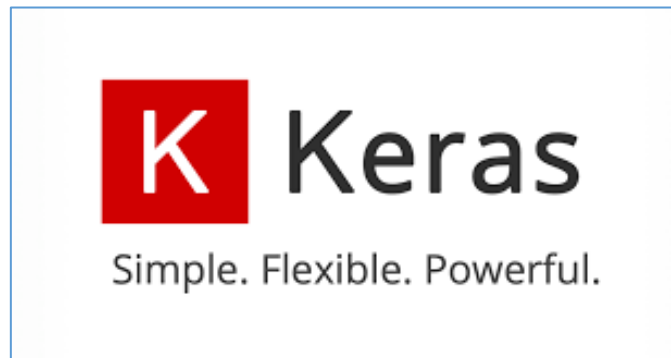


Рисунок 30 — Логотип Keras

Основні риси Keras:

- Простота використання:

Keras має простий та легкий інтерфейс, що дозволяє швидко створювати та експериментувати з різними моделями нейронних мереж.

- Модульність:

Його архітектура є модульною, що дозволяє легко додавати нові шари, функції активації та інші компоненти.

- Підтримка різних бекендів:

Keras може використовувати різні бекенди, такі як TensorFlow, Theano або Microsoft Cognitive Toolkit (CNTK). Це дозволяє розробнику вибрати підходящий бекенд відповідно до ваших потреб та умов.

- Зручність для початківців та експертів:

Keras достатньо простий для початківців, але і гнучкий для досвідчених користувачів. Це зроблено за рахунок того, що розробник можете легко визначати власні шари та функції активації, або використовувати готові.

- Вбудовані функції для обробки даних:

Keras має вбудовані інструменти для обробки даних, такі як завантажувачі даних, аугментація та інші.

Зворотна сумісність:

Keras інтегровано в TensorFlow, що робить його стандартним інтерфейсом для високорівневого програмування в нейромережах для цієї бібліотеки.

Основна концепція Keras - це модель, яка визначається за допомогою послідовності шарів. Кожен шар може бути розглянутий як будівельний блок мережі. Шари можуть бути додані до моделі за допомогою простого інтерфейсу. Таким чином Keras - це потужний інструмент для створення та експериментів з нейронними мережами, який вигідно використовувати як початківцям, так і досвідченим дослідникам в галузі машинного навчання.

### **scikit-learn**

Scikit-learn (Scientific Kit for Learning) — це відкрита бібліотека для мови програмування Python, яка надає ефективні інструменти для аналізу даних і машинного навчання. Ця бібліотека створена на основі інших популярних пакетів для наукових обчислень у Python, таких як NumPy, SciPy і matplotlib. Основна мета scikit-learn — надати простий і ефективний інтерфейс для розв'язання завдань класифікації, регресії, кластеризації та інших задач машинного навчання.



Рисунок 31 — Логотип scikit-learn

Основні можливості scikit-learn включають:

- Класифікація і регресія:

Scikit-learn надає реалізації різноманітних алгоритмів класифікації і регресії, таких як метод опорних векторів (SVM), наївний Баєсівський класифікатор, лінійна регресія та інші.

- Кластеризація:

Бібліотека включає в себе алгоритми кластеризації, такі як k-середніх, ієрархічна кластеризація та інші.

- Вимірювання якості моделей:

Scikit-learn має набір інструментів для оцінки якості моделей, таких як метрики точності, точності, відзиву, F1-мери та інші.

- Вибір моделей та параметрів:

Бібліотека надає інструменти для автоматичного вибору моделей та оптимізації параметрів моделей.

- Обробка даних:

Scikit-learn допомагає у підготовці даних для аналізу та машинного навчання, зокрема виконання операцій нормалізації, стандартизації та інші операції.

- Введення/виведення даних:

Легко завантажувати дані з різних форматів та джерел і зберігати моделі для подальшого використання.

- Пайплайни для обробки даних:

Scikit-learn дозволяє збудовувати складні пайплайни для обробки даних та побудови моделей.

- Вбудовані набори даних:

Включає в себе деякі стандартні набори даних для практичного тестування та навчання.

- Найпоширеніші класи об'єктів у scikit-learn:

Моделі: Представляють алгоритми машинного навчання, які можуть бути навчені на даних та використовуватися для прогнозування.

Метрики: Визначають якість та ефективність моделей на основі вхідних даних.

Інструменти для обробки даних: Містять функції для обробки, нормалізації та видалення шуму з даних.

Scikit-learn — це потужний інструментарій для розробки та впровадження моделей машинного навчання в різних областях, включаючи науку про дані, фінанси, біологію та інші галузі.

### **NumPy**

NumPy (Numerical Python) - це потужна бібліотека для мови програмування Python, яка надає підтримку для роботи з масивами, векторами та матрицями, а також велику колекцію математичних функцій для їх обробки. NumPy є однією з основних бібліотек для наукового обчислення в Python і служить базою для багатьох інших бібліотек, таких як SciPy, Pandas та інші.



Рисунок 32 — Логотип NumPy

Основні характеристики NumPy:

- Масиви (Arrays):

Основний об'єкт у NumPy - це масив (`numpy.ndarray`). Масиви в NumPy подібні до списків в Python, але забезпечують ефективніше використання пам'яті та операції векторизації (операції, які виконуються одночасно для всіх елементів масиву):

Створення масиву

```
arr = np.array([1, 2, 3, 4, 5])
```

- Операції векторизації:

NumPy дозволяє виконувати операції на всьому масиві, не потрібно використовувати цикли.

### Векторизована операція додавання

```
arr = arr + 1
```

- Широкий спектр математичних функцій:

NumPy містить багато вбудованих математичних функцій, таких як `sin`, `cos`, `exp`, `sqrt` та інші.

### Використання математичної функції

```
np.sin(arr)
```

- Індксація та зрізи:

NumPy підтримує потужні механізми для індексації та зрізів масивів, що дозволяє вам легко отримувати доступ до певних елементів чи підмасивів.

Зріз масиву

```
arr_slice = arr[1:4]
```

- Радіоматричні операції:

NumPy надає можливість виконувати операції між масивами різних розмірностей, використовуючи техніку розширення масивів.

Радіоматричне додавання

```
a = np.array([1, 2, 3])
```

```
b = np.array([4, 5, 6])
```

```
result = a + b
```

- Лінійна алгебра:

NumPy містить функції для виконання операцій лінійної алгебри, таких як обчислення детермінанту, обернення матриць, розв'язання систем лінійних рівнянь та інших.

Обернення матриці

```
matrix = np.array([[1, 2], [3, 4]])
```

```
inv_matrix = np.linalg.inv(matrix)
```

- Зчитування та запис файлів:

NumPy має інструменти для зчитування та запису даних у файлових форматах, таких як CSV.

Зчитування даних з CSV файлу

```
data = np.genfromtxt('data.csv', delimiter=',')
```

## Pandas

Pandas — програмна бібліотека, написана для мови програмування Python для маніпулювання даними та їхнього аналізу. Вона, зокрема, пропонує структури даних та операції для маніпулювання чисельними таблицями та часовими рядами. pandas є вільним програмним забезпеченням.



Рисунок 33 — Логотип Pandas

Основні структури даних в Pandas:

- DataFrame:

Це основна структура даних в Pandas. Це двовимірна таблична структура даних, схожа на таблицю бази даних чи електронну таблицю Excel. Кожен стовпець у DataFrame - це об'єкт типу Series, а DataFrame сам може бути розглянутий як словник об'єктів Series.

- Series:

Series - це одновимірний масив, який може містити будь-який тип даних. У вас може бути серія чисел, рядків, чи навіть інших об'єктів Python.

Основні можливості Pandas:

- Читання та запис даних:

Pandas може легко читати дані з різноманітних джерел, таких як CSV, Excel, SQL, HTML, JSON та інші. Також, ви можете зберігати дані у різних форматах.

```
import pandas as pd
# Читання даних з CSV файлу
```

```
df = pd.read_csv('file.csv')
# Запис даних у Excel файл
df.to_excel('output.xlsx', index=False)
```

- Індксація та вибірка даних:

Розробник може легко вибирати дані з DataFrame за допомогою індксів, а також використовувати різні методи фільтрації та вибірки.

```
# Вибірка стовпця
column_data = df['Column_Name']
# Вибірка рядка за індксом
row_data = df.loc[0]
# Фільтрація за умовою
filtered_data = df[df['Column_Name'] > 10]
```

- Операції з даними:

Pandas надає різні операції для обробки та зміни даних, такі як об'єднання, групування, сортування, агрегація та інші.

```
# Об'єднання DataFrame
merged_df = pd.concat([df1, df2])
# Групування та агрегація
grouped_data = df.groupby('Category').mean()
# Сортування
sorted_df = df.sort_values(by='Column_Name')
```

- Обробка пропущених значень:

Pandas надає зручні інструменти для роботи з пропущеними значеннями, такі як їх заповнення або видалення.

```
# Заповнення пропущених значень
df.fillna(value, inplace=True)
# Видалення рядків з пропущеними значеннями
df.dropna(inplace=True)
```



## Matplotlib

Matplotlib - це бібліотека для створення графіків у мові програмування Python. Вона надає об'єктно-орієнтований API для вбудованих графіків у різних форматах. За допомогою Matplotlib можна створювати різні типи графіків, включаючи лінійні графіки, стовпчасті діаграми, колові діаграми, гістограми та багато інших.



Рисунок 34 — Логотип matplotlib

Основні компоненти Matplotlib:

- Figure (Фігура):

Фігура є контейнером для всіх об'єктів графіку. Вона може містити один або декілька графіків (axes).

- Axes (Осі):

Вісі - це об'єкти, що фактично містять графічні елементи, такі як лінії, польоти, текст і т.д. В одній фігурі може бути кілька вісей.

- Axis (Осі):

Осі визначають область деякої фігури та визначають масштаб для даних, що виводяться.

- Artist (Митець):

Більшість об'єктів в Matplotlib є митецькими об'єктами. Це може бути все, від тексту та ліній до колірних карт і візуалізацій.

Matplotlib надає два стилі API для створення графіків:

- MATLAB-стиль:

Використовується функціональний підхід, подібний до того, як графіки створюються в MATLAB. Це дозволяє швидко створювати прості графіки.

- Об'єктно-орієнтований стиль:

Використовується більш об'єктно-орієнтований підхід, де об'єкти фігури та вісей створюються явно і контролюються.

Matplotlib також підтримує велику кількість різноманітних параметрів та можливостей для вдосконалення та налаштування графіків. Окрім того, з Matplotlib можна також створювати анімації, 3D-графіки та інші розширені візуалізації.

### 3.4 Вибір архітектури проекту

Для побудови моделі глибокого навчання було використано бібліотеку Keras з використанням архітектури Sequential. Тобто спочатку створюється об'єкт моделі типу Sequential. Сам Sequential є класом у Keras, який дозволяє лінійно додавати шари (layers) до моделі. Далі кожен новий шар буде автоматично приєднаний до попереднього.

```
# Define LSTM model
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(X_train_reshaped.shape[1], X_train_reshaped.shape[2])))
model.add(LSTM(50))
model.add(Dense(25))
model.add(Dense(1))

# Compile the model with a custom learning rate
model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')
model.summary()
```

Model: "sequential\_4"

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 1, 50)	101200
lstm_3 (LSTM)	(None, 50)	20200
dense_2 (Dense)	(None, 25)	1275
dense_3 (Dense)	(None, 1)	26

```
=====  
Total params: 122701 (479.30 KB)  
Trainable params: 122701 (479.30 KB)  
Non-trainable params: 0 (0.00 Byte)
```

Рисунок 35 — Архітектура моделі глибокого навчання

LSTM(50): Додає шар LSTM із 50 нейронами. `return_sequences=True`: Параметр, який вказує, чи повертати повну послідовність вихідних значень для кожного кроку часу. В нашому випадку, оскільки слідує ще один LSTM шар, ми встановлюємо `return_sequences=True`, щоб передавати повні послідовності. `input_shape=(X_train_reshaped.shape[1], X_train_reshaped.shape[2])`: Визначає форму вхідних даних. У цьому випадку, вказується, що вхідні дані мають форму (кількість\_кроків\_часу, кількість\_ознак). Далі додається другий LSTM шар із 50 нейронами.

Далі додається Dense шар з 25 нейронами та ще один вихідний Dense шар з одного нейрону.

Взагалі, коли додається Dense шари після LSTM шару, це означає, що модель може навчитися вагам і зсувам (biases), щоб виразити більше абстрактних та складних патернів у вхідних даних. Dense шари виконують операції лінійного перетворення (включаючи ваги та зсуви), а функція активації додає нелінійність, що допомагає моделі у вивченні складніших функцій.

Кількість нейронів у кожному Dense шарі є гіперпараметром, який можна налаштовувати під час експериментів для досягнення кращої продуктивності моделі на конкретній задачі. Таким чином, вихідні ознаки, які отримані від LSTM шару, проходять через два Dense шари для отримання більш складної та абстрактної репрезентації перед передачею до останнього Dense шару, який формує окремий вихід.

Модель компілюється з оптимізатором `adam` та функцією втрати `mean_squared_error`. Оптимізатор `adam` використовується для навчання моделі, а `mean_squared_error` вказує, як буде оцінюватися помилка між прогнозованими та фактичними значеннями під час навчання.

Отже, загальна архітектура моделі включає в себе два LSTM-шари для обробки часових послідовностей та два Dense-шари для взаємодії зі згенерованими ознаками та виводом прогнозів.

### 3.5 Висновки з розділу 3

В цьому розділі були розглянуті основні етапи підготовки до реалізації поставленої задачі та зроблені наступні висновки:

1. Задачу можливо реалізувати сучасними програмними та технічними засобами;
2. Визнано необхідність ретельного використання виключно верифікованих джерел інформації для підготовки робочого датасету для подальшого аналізу. До того ж необхідно уважно підходити до достатності даних.
3. Вибрані наступні фреймворки та бібліотеки для роботи: програмне середовище: Anaconda, jupyter notebook, spyder. Фреймворки: TensorFlow, Keras. Бібліотеки: scikit-learn, NumPy, Pandas та Matplotlib.
4. Вибрана LSTM – мережа з додаванням шарів Dense, архітектурою Sequential та функцією активації ReLu.

## РОЗДІЛ 4 МОДЕЛЮВАННЯ ТА АНАЛІЗ РОБОТИ НЕЙРОННОЇ МЕРЕЖІ

### 4.1 Реалізація моделі за допомогою бібліотеки `scikit-learn`, `pandas` та `matplotlib`.

Для можливості порівняльного аналізу спочатку було реалізовано модель за допомогою лінійної регресії.

Перш ніж проводити прогнозування було зроблено аналіз на те, як дані дата сету корелюють з цільовим показником. За для чого:

Кореляція між змінними та цільовим показником може вказувати на ступінь лінійної залежності між ними. Це може впливати на якість та ефективність моделі прогнозування. Яка вона буває:

Позитивна кореляція: Якщо змінна має позитивну кореляцію з цільовим показником, це означає, що, в середньому, зі збільшенням значення змінної цільовий показник також збільшується. У такому випадку модель може легше виявити та використовувати цю залежність для прогнозування.

Від'ємна кореляція: Якщо змінна має від'ємну кореляцію з цільовим значенням, це означає, що зі збільшенням значення змінної цільове значення зменшується. Аналогічно, модель може використовувати цю залежність для прогнозування.

Слабка кореляція: Якщо кореляція слабка або немає, це може вказувати на відсутність лінійної залежності між змінною та цільовим значенням. У таких випадках прогнозування може бути більш важким, інші методи моделювання, такі як нелінійні моделі, можуть бути більш ефективними.

Мультиколінеарність: Якщо дві чи більше змінних сильно корелюють між собою, це може призводити до проблеми мультиколінеарності, яка може ускладнити оцінку ваг моделі.

Кореляція не завжди вказує на причинно-наслідковий зв'язок, і важливо враховувати інші аспекти дослідження та моделювання для забезпечення адекватності та точності прогнозів.

Лістинг 1 код для оцінки кореляції даних датасету з цільовим показником ВВП.

```

path_to_file =
r'C:\Education\Диплом\UsaStatistics_1970.xlsx'
data = pd.read_excel(path_to_file, index_col=0,
sheet_name='DataTr')
features = data.drop(['ВВП (поточні долари США)'],
axis=1)
target = data['ВВП (поточні долари США)']
correlations = features.corrwith(target)
high_corr_features = correlations[correlations >
0.9].index
significant_correlations =
data[high_corr_features].corrwith(target)
significant_correlation_table =
pd.DataFrame(significant_correlations, columns=['Кореляція
з ВВП'])

```

	A	B
1		Кореляція з ВВП
2	Скоригований чистий національний дохід (поточні долари США)	0,999785633
3	Скоригований чистий національний дохід на душу населення (поточні долари США)	0,99697637
4	Скориговані заощадження: споживання основного капіталу (поточний дол. США)	0,998602536
5	Скориговані заощадження: витрати на освіту (поточні дол. США)	0,989097387
6	Коефіцієнт вікової залежності, старих (% населення працездатного віку)	0,912015645
7	Авіап перевезення, вантажні перевезення (млн. тонно-км)	0,96798288
8	Широка грошова маса (поточний LCU)	0,977167434
9	Урожайність зернових (кг/га)	0,967976077
10	Збори за використання інтелектуальної власності, платежі (ВоР, поточні долари США)	0,980623175
11	Комісії за використання інтелектуальної власності, квитанції (ПБ, поточні долари США)	0,978968602
12	Хімічна промисловість (% доданої вартості у виробництві)	0,960961985
13	Викиди CO2 від транспорту (% від загального спалювання палива)	0,941252283
14	Експорт комерційних послуг (поточні дол. США)	0,98001496
15	Імпорт комерційних послуг (поточні долари США)	0,98630108
16	Оплата праці працівників (поточний LCU)	0,984480245
17	Комп'ютерні, комунікаційні та інші послуги (% імпорту комерційних послуг)	0,969558509
18	Індекс споживчих цін (2010 = 100)	0,9783349
19	Індекс рослинництва (2014-2016 = 100)	0,94848435
20	Митні та інші імпорتنі збори (поточний LCU)	0,905925008
21	Внутрішній кредит, наданий фінансовим сектором (% ВВП)	0,974812877
22	Внутрішній кредит приватному сектору (% ВВП)	0,946034067
23	Виробництво електроенергії з відновлюваних джерел, за винятком гідроелектростанцій (% від загального обсягу)	0,938426665
24	Виробництво електроенергії з відновлюваних джерел, за винятком гідроелектроенергії (кВт-год)	0,94435664
25	Витрати (поточний LCU)	0,971478075
26	Експорт як імпортна здатність (постійна LCU)	0,990178896
27	Експорт товарів і послуг (ПБ, поточний долар США)	0,984693179
28	Експорт товарів і послуг (постійна LCU)	0,987297886

Рисунок 37 — Частина отриманого результату в табличній формі



Рисунок 38 — Теплова карта кореляції

Для подальшої роботи було видалено дані, які мають коефіцієнт кореляції від 0,99%.

Тепер основні етапи роботи коду:

Лістинг 2 Після зазначення шлях до файлу Excel, який містить датасет, зчитуються дані з аркуша "DataTr" у форматі pandas DataFrame.

```
path_to_file
='C:\Education\Диплом\UsaStatistics_1970.xlsx'
data = pd.read_excel(path_to_file, index_col=0,
sheet_name='DataTr')
```

Лістинг 3 Визначається матриця ознак (features) та вектор цільової змінної (target) Дані розділяються на навчальний (до 2020 року), тестовий (2020 рік) та майбутній (2021 рік) набори.

```

features = data.drop(['ВВП (поточні долари США)'],
axis=1)
target = data['ВВП (поточні долари США)']
train_data = data[data.index < 2020]
test_data = data[data.index == 2020]
future_data = data[data.index == 2021]

```

Лістинг 4 Ініціалізується та навчається модель лінійної регресії на навчальних даних, здійснюється прогноз ВВП на тестовому наборі (2020 рік) та майбутньому наборі (2021 рік). Робиться перевірка на відсутність пропущених даних датасету. Формується DataFrame, який містить фактичні значення та прогноз ВВП для 2021 року. Використовуються об'єднані дані для навчання моделі лінійної регресії для прогнозу ВВП на 2022 рік.

```

model = LinearRegression()
model.fit(X_train, y_train)
y_pred_test = model.predict(X_test)
if y_test.isnull().any():
    print("У тестовому наборі є недійсні значення в
цільовій змінній.")
    y_test = y_test.dropna()
    X_test = X_test.loc[y_test.index]
    y_pred_test = model.predict(X_test)
y_pred_future = model.predict(X_future)
combined_data_2021 = pd.concat([future_data,
pd.Series(y_pred_future, index=future_data.index, name='ВВП
(поточні долари США) прогноз']], axis=1)
X_combined = combined_data_2021.drop(['ВВП (поточні
долари США)'], axis=1)
y_combined = combined_data_2021['ВВП (поточні долари
США) прогноз']

```



```

model_combined = LinearRegression()
model_combined.fit(X_combined, y_combined)

```

Лістинг 5 Здійснюється прогноз ВВП на 2022 рік, виводяться фактичні та прогнозовані значення, а також будується графік.

```

X_2022 = future_data.drop(['ВВП (поточні долари США)'],
axis=1)
y_pred_2022 = model.predict(X_2022)
print(f'Прогноз ВВП на 2022 рік: {y_pred_2022[0]}')
actual_2021 = combined_data_2021['ВВП (поточні долари
США)'].iloc[0]
print(f'Фактичне ВВП на 2021 рік: {actual_2021}')

```

## 4.2 Реалізація LSTM нейронної мережі

Лістинг 6 Зчитування даних з Excel-файлу, що містить датасет та вивід перших рядків та інформацію про дані. Розділення даних на ознаки та цільову змінну.

```

data = pd.read_excel(path_to_file, sheet_name='DataTr')
print(data.head())
print(data.info())
features = data.drop('ВВП (поточні долари США)',
axis=1)
target = data['ВВП (поточні долари США)']

```

Лістинг 7 Застосування логарифму до цільової змінної та масштабування ознак та цільової змінної

```

target_log = np.log1p(target)
feature_scaler = StandardScaler().fit(features)

```

```

features_scaled = feature_scaler.transform(features)
target_scaler = MinMaxScaler()
target_scaled =
target_scaler.fit_transform(target_log.values.reshape(-1,
1))

```

**Лістинг 8 Розділення даних на навчальні та тестові набори. LSTM очікує дані у форматі [samples, time steps, features], тому робимо реформування даних.**

```

X_train, X_test, y_train, y_test =
train_test_split(features_scaled, target_scaled,
test_size=0.2, random_state=42)
X_train_reshaped = np.reshape(X_train,
(X_train.shape[0], 1, X_train.shape[1]))
X_test_reshaped = np.reshape(X_test, (X_test.shape[0],
1, X_test.shape[1]))

```

### **Лістинг 9 Створення та навчання моделі LSTM**

```

model = Sequential()
model.add(LSTM(50, return_sequences=True,
input_shape=(X_train_reshaped.shape[1],
X_train_reshaped.shape[2])))
model.add(LSTM(50))
model.add(Dense(25))
model.add(Dense(1))
model.compile(optimizer=Adam(learning_rate=0.001),
loss='mean_squared_error')
model.fit(X_train_reshaped, y_train, epochs=250,
batch_size=32)

```

Лістинг 10 Прогнозування та зворотне масштабування даних, оцінка та виведення метрик.

```

y_pred_scaled = model.predict(X_test_reshaped)
y_pred_log =
target_scaler.inverse_transform(y_pred_scaled)
y_pred = np.expml(y_pred_log)
mse = mean_squared_error(y_test_actual.flatten(),
y_pred.flatten())
mae = mean_absolute_error(y_test_actual.flatten(),
y_pred.flatten())
r2 = r2_score(y_test_actual.flatten(),
y_pred.flatten())

```

Лістинг 11 Прогноз на майбутні дані та виведення результату.

```

future_data = pd.read_excel(path_to_file,
sheet_name='NewData1')
future_data_scaled =
feature_scaler.transform(future_data)
future_data_reshaped = np.reshape(future_data_scaled,
(future_data_scaled.shape[0], 1,
future_data_scaled.shape[1]))
y_pred_scaled_future =
model.predict(future_data_reshaped)
y_pred_future_log =
target_scaler.inverse_transform(y_pred_scaled_future)
y_pred_future = np.expml(y_pred_future_log)

```

### 4.3 Аналіз отриманих результатів

Результат роботи моделі лінійної регресії виглядає наступним чином:

Прогноз ВВП на 2022 рік: 23315442259319.78  
 Фактичне ВВП на 2021 рік: 23315080560000

Рисунок 39 — Вивід результатів прогнозування в порівнянні з фактичними даними за 2021 рік.

Таким чином ми бачимо, що прогноз ВВП на 2022 рік майже не відрізняється від фактичного ВВП на 2021 рік, що може свідчити про недосконалість прогнозу за допомогою моделі лінійної регресії.

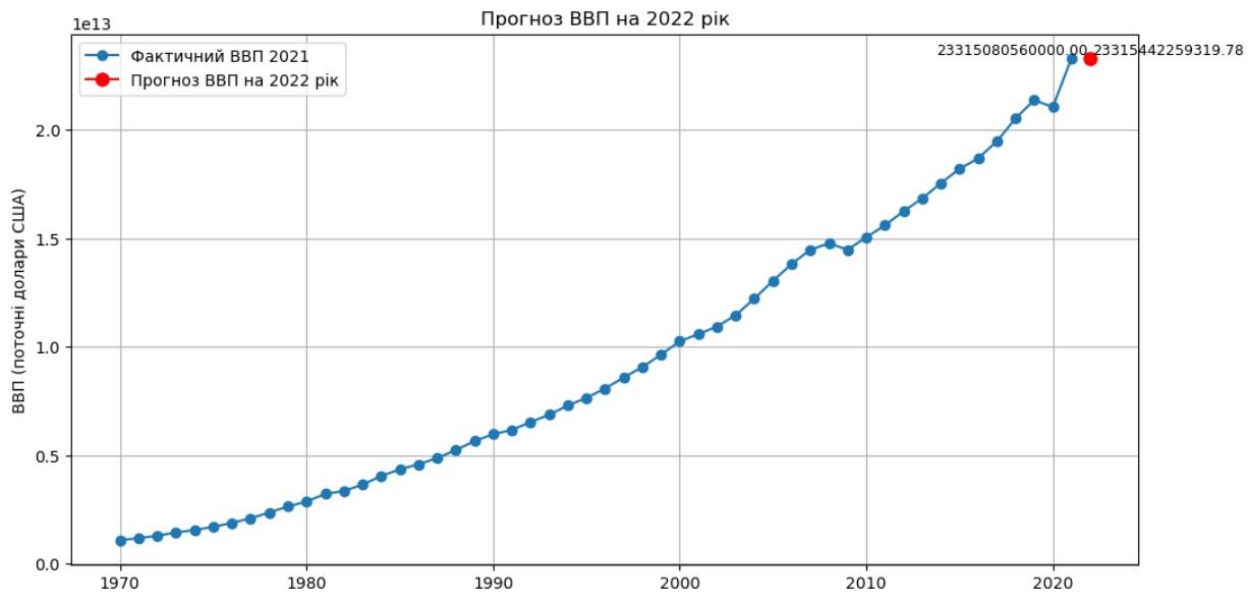


Рисунок 40 — Виведення результатів прогнозування в порівнянні з фактичними даними за 2021 рік в графічній формі.

Результат роботи LSTM - моделі виглядає наступним чином:

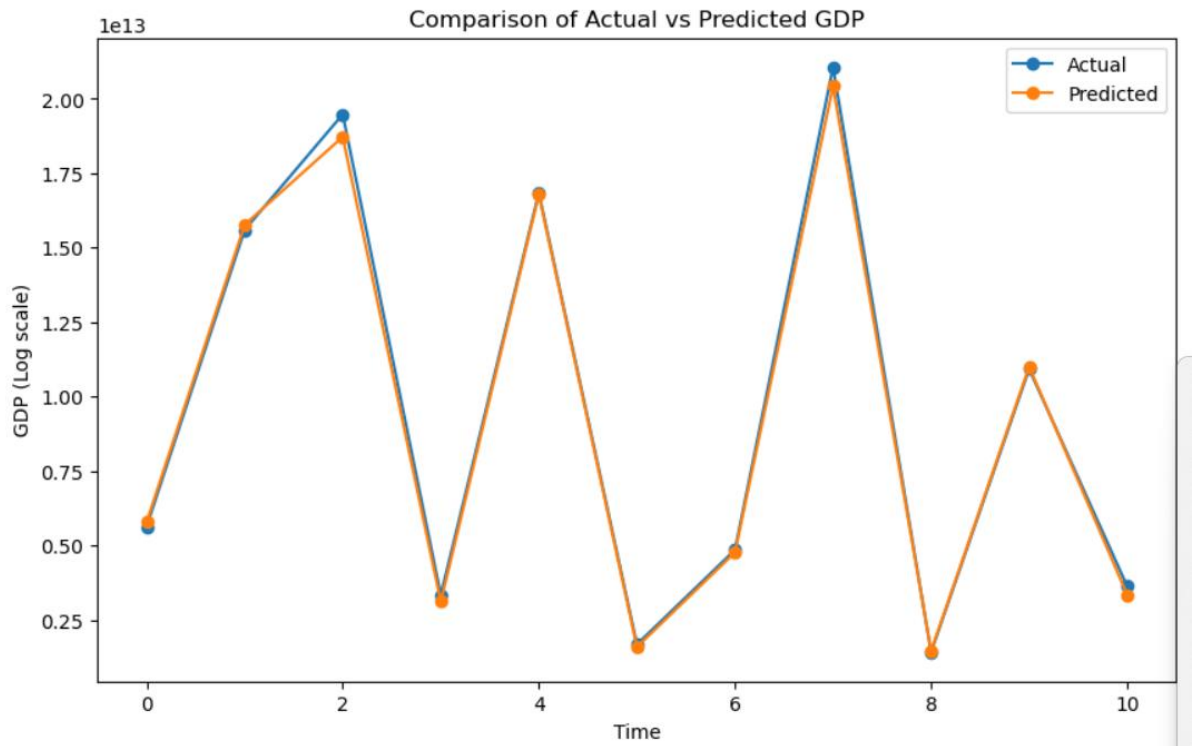


Рисунок 40 — графічний вивід співпадіння прогнозованих та фактичних даних

Прогноз на 2022 рік:		
Рік	Predicted	
0 2022	2.501343e+13	
фактичний ВВП: 2.546270e+13		

Рисунок 41 — фактичний та розрахований ВВП

При цьому  $R^2$ , або коефіцієнт детермінації (Coefficient of Determination) на текстових даних склав 0.9819

#### 4.4 Висновки з розділу 4

В цьому розділі були розглянуті основні етапи підготовки до реалізації поставленої задачі та зроблені наступні висновки:

1. Створена модель лінійної регресії для порівняльного аналізу з результатом наступного моделювання;

2. Проведено кореляційний аналіз даних, що використовуються а дата-сеті;
3. Створена модель LSTM – нейронної мережі для реалізації задачі прогнозування ВВП;
4. Показано, що прогноз за допомогою LSTM – нейронної мережі є більш наближеним до фактичного показника прогнозованого року, а ніж це прогнозує модель лінійної регресії.

## ВИСНОВКИ

1. В кваліфікаційній роботі розглянута проблематика необхідності прогнозування економічних показників, зокрема Внутрішнього валового продукту країни (ВВП). Зроблено огляд наявних сучасних методів рішення цієї задачі. Окреслено сфери використання результатів прогнозування ВВП. Доведено пріоритетність такого прогнозування за допомогою використання нейронних мереж.

2. Досліджені основні типи та архітектури нейронних мереж. Зроблені висновки, що мережі довгострокової пам'яті (LSTM), які розширенням рекурентних нейронних мереж (RNN), призначені для вирішення проблеми, притаманних іншим мережам, як наприклад проблема зниклого градієнта та забезпечує моделі здатність вивчати та враховувати довгострокові залежності в даних. То ж цей тип мережі особливо корисний при роботі з послідовністю даних, таких як, наприклад, часові ряди які нам і потрібні.

3. Вивчені основні умови, які слід пред'являти до датасетів, на підставі яких робляться прогнози. Особлива увага має приділятися джерелам отримання інформації, а саме верифікованих даних та їх достатності. Для вирішення завдання кваліфікаційної роботи було вибрано мову Python, та наступні бібліотеки та фреймворки, що доступні для цієї мови, а саме фреймворки TensorFlow, Keras та бібліотеки: scikit-learn, NumPy, Pandas та Matplotlib.

4. Проведено кореляційний аналіз даних, що використовуються а датасеті. Проведено навчання двох моделей нейронних мереж для прогнозування ВВП (лінійної регресії та LSTM – мережі), налаштоване середовище для навчання, підготовлений відповідний завданню датасет та успішно виконано навчання обох моделей.

5. Було визначено, що LSTM – мережа має переваги перед лінійною регресією завдяки своїй більшій гнучкості та здатності працювати з нелінійними

залежностями, що підтверджено практичними результатами порівняння отриманих прогнозних даних з реальними контрольними даними. Однак, LSTM – мережа є значно складнішою моделлю, а тому потребує дуже уважного відношення до налаштувань.

6. Поставлена проблема прогнозування ВВП Сполучених Штатів Америки була повністю вирішена з допустимою точністю.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кириченко І.В., магістрант, Безверхий А. І., доцент — науковий керівник. Комп'ютерна система аналізу і прогнозування ВВП країн світу за допомогою нейронної мережи ЗБІРНИК наукових праць студентів, аспірантів, докторантів і молодих вчених «МОЛОДА НАУКА-2023», Запоріжжя : Запорізький національний університет, 2023. С. 92-94.
2. Кириченко І.В., магістрант, Безверхий А. І., доцент — науковий керівник. Комп'ютерна система аналізу міжнародної економіки з використанням машинного навчання. III Всеукраїнської науково-практичної конференції за участю молодих науковців «АКТУАЛЬНІ ПИТАННЯ СТАЛОГО НАУКОВО-ТЕХНІЧНОГО ТА СОЦІАЛЬНО-ЕКОНОМІЧНОГО РОЗВИТКУ РЕГІОНІВ УКРАЇНИ». Запоріжжя : Запорізький національний університет, 2023. С. ??.
3. Лугінін О.Є. Статистика національної та міжнародної економіки : навч. посіб. / О.Є. Лугінін, С.В. Фомішин. - Львів : Новий Світ-2000, 2014.-471 с.
4. Лугінін, О.Є. Економетрія: навч. посібник / О.Є. Лугінін; – 2-е вид., перероб. і доп. – Київ: ЦНЛ, 2008. – 278 с.
5. Sims C. Money, income and causality / C. Sims // American Economic Review, 1972. — № 652. — P. 540—542.
6. Літнарівич Р.М. Побудова і дослідження математичної моделі за джерелами експериментальних даних методами регресійного аналізу. Навчальний посібник, МEGY, Рівне, 2011.-140 с.
7. Peña, D., Poncela, P. Nonstationary dynamic factor analysis. Journal of Statistical Planning and Inference, No. 136, pp. 1237–1257. DOI: <https://doi.org/10.1016/j.jspi.2004.08.020>, 2006.
8. Borisyuk R. Forecasting the 2005 general election: A neural network approach. The British Journal of Politics and International Relations. 2005. February 7. PP. 199-209

9. Уоссермен Ф. Нейрокомп'ютерна техніка: Теорія та практика. Мір, 1992. 184 с.
10. Charu Aggarwal. Neural Networks and Deep Learning: A Textbook, Springer, p. 512, 2018
11. Gers F. A., Eck D., Schmidhuber J. Applying LSTM to time series predictable through time-window approaches. Neural Nets WIRN Vietri01. Springer, London. 2002. p.p. 193-200.
12. Zeiler M. D., Fergus R. Visualizing and understanding convolutional networks. European conference on computer vision. Springer, Cham, 2014. P. 818-833.
13. He K., Zhang X., Ren S., Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision. 2015. P. 1026-1034.
14. Rumelhart D. E., Hinton G. E., Williams R. J. Learning representations by back-propagating errors. Nature. 1986. Vol. 323, No 6088. P. 533-536.

**Декларація  
академічної доброчесності  
здобувача ступеня вищої освіти ЗНУ**

Я, Кириченко Ігор В'ячеславович, студент 2 курсу форми навчання денної, Інженерного навчально-наукового інституту ім. Ю.М. Потебні ЗНУ, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти se22m-08@stu.zsea.edu.ua

- підтверджую, що написана мною кваліфікаційна робота на тему **«Комп'ютерна система аналізу міжнародної економіки з використання машинного навчання»** відповідає нормам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь який спосіб, у тому числі за допомогою інтернет-систем, а також на архівування моєї роботи в базі даних цієї системи.

Дата 30.11.2023 Підпис \_\_\_\_\_ Кириченко Ігор В'ячеславович  
(студент)

Дата 30.11.2023 Підпис \_\_\_\_\_ Безверхий Анатолій Ігорович  
(науковий керівник)