

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Метод нейронних мереж оцінювання продуктивності бізнес-процесів.»

на здобуття освітнього ступеня бакалавра

зі спеціальності 124 Системний аналіз 0,

(код, найменування спеціальності)

освітньо-професійної програми Системний аналіз

(назва)

Кваліфікаційна робота містить результати власних досліджень.

Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

(підпис)

Захар ТУТОВ
Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр. ____

Захар ТУТОВ

Ім'я, ПРІЗВИЩЕ

Керівник: Ровіль НАФЄЄВ

науковий ступінь, Ім'я, ПРІЗВИЩЕ

вчене звання

Рецензент: Юлія БЕРЕЗОВСЬКА

науковий ступінь, Ім'я, ПРІЗВИЩЕ

вчене звання

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Бакалавр

Спеціальність 124 Системний аналіз

Освітньо-професійна програма Системний аналіз

ЗАТВЕРДЖУЮ

Завідувач кафедру ІСТ

Каміла СТОРЧАК

Каміла СТОРЧАК Ім'я, ПРІЗВИЩЕ

«___» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Тутов Захар Юрійович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Метод нейронних мереж оцінювання продуктивності бізнес-процесів.

керівник кваліфікаційної роботи Ровіль НАФЄЄВ,

(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій
від «___» _____ 2025 р. № ___

2. Строк подання кваліфікаційної роботи «___» _____ 2025 р.

3. Вихідні дані до кваліфікаційної роботи: бази даних, наукова та технічна література, експлуатаційна документація, нормативні документи.

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

1. Постановка завдання визначення потреби, вимог до функціональності та розроблення програмного забезпечення для прогнозування та оцінки бізнес-процесу.

2. Архітектура системи вибір типу алгоритму штучної нейронної мережі, для більш результатного прогнозування.

3. Реалізація та тестування програмного забезпечення та виведення вихідних даних для подальшого аналізу.

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання «___» _____ 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назви етапів Кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Ознайомлення з вимогами до роботи та складання календарного плану	15.02.2025 - 28.02.2025	Виконано
2	Аналіз та огляд предметної області	1.3.2025 - 14.03.2025	Виконано
3	Системний аналіз теорії основи бізнес-процесів та їх методи оптимізації	15.03.2025 - 29.03.2025	Виконано
4	Аналіз та огляд теоретичних основ штучного інтелекту. Аналіз наявних рішень оцінювання бізнес- процесів	29.03.2025 - 10.04.2025	Виконано
5	Розробка системи управління оцінюванням	11.04.2025 - 27.04.2025	Виконано
6	Розробка рекомендацій щодо інтеграції системи	28.04.2025 - 12.05.2025	Виконано
7	Завершення роботи, перевірка на плагіат, фінальне оформлення та подання роботи	13.05.2025 - 20.05.2025	Виконано

Здобувач(ка) вищої освіти _____
(підпис)

Захар ТУТОВ
(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи _____
(підпис)

Ровіль НАФЄЄВ
(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи складається зі вступу, трьох розділів, загальних висновків, списку джерел. Загальна кількість сторінок 87, наведено 77 рисунків та 2 таблиці. Список літературної інформації складається з 19 джерел.

Кваліфікаційна робота присвячена дослідженню методів бізнес-процесу та нейронних мереж, як побудовані та працюють штучні нейронні мережі та методи бізнес-процесів та їх застосування у бізнесі. А також яку важливу роль починає відігравати застосування штучних багат шарових нейронних мереж в оптимізації бізнес-процесів. Актуальність дослідження полягає в тому, що детально розглядається з чого будується сучасні бізнес процеси, їх приклади та де їх можна застосовувати.

Детально розглянуто з яких основ складається нейронна мережа, як працюють нейронні зв'язки та які є багат шарові збірки нейронних мереж. Наведені приклади багат шарових нейронних мереж, які використовуються у бізнесах. Здійснено системний аналіз чому нейронні мережі стають невід'ємною частиною сучасного менеджменту та які перспективи відкрилися для компаній, які активно починають користуватися ними. Розглянута ІТ-система стає заміною традиційним методам прогнозування. Вона допомагає бізнесу підлаштовуватися під нові виклики, такі, як зміни ринкових трендів та геополітичних криз.

Результати дослідження є практично значущим для бізнесу в цілому, завдяки цьому вони можуть оцінити свої витрати, а також оптимізувати та автоматизувати велику низку своїх процесів. Ця динамічна система адаптується при зміні та додавання нових даних.

КЛЮЧОВІ СЛОВА: УПРАВЛІННЯ, ІДЕНТИФІКАЦІЯ, ПАРЦЕПТРОН, МОДЕЛЮВАННЯ, ПРОДУКТИВНІСТЬ, РЕНТАБЕЛЬНІСТЬ, ОПТИМІЗАЦІЇ, ОСНОВИ, ШТУЧНИХ, НЕЙРОНИХ, TLU, БАГАТОШАРОВИХ, BPMN, IDEF.

ЗМІСТ

ВСТУП.....	8
 РОЗДІЛ 1 ТЕОРІЯ І ОСНОВИ БІЗНЕС-ПРОЦЕСІВ ТА ЇХ МЕТОДИ ОПТИМІЗАЦІЇ.....	 9
1.1 Поняття бізнес-процесу	9
1.2 Ідентифікація процесів.....	12
1.3 Моделювання візуальних процесів BPM та IDEF.....	14
1.4 Показники продуктивності у бізнес-процесі.....	16
1.5 Рентабельність інвестицій	23
1.6 Методи оптимізації.....	25
 РОЗДІЛ 2 ЗАСТОСУВАННЯ НЕЙРОННОЇ МЕРЕЖІ ЯК ІНСТРУМЕНТУ В: ОПТИМІЗАЦІЇ, ОЦІНЦІ ТА АНАЛІЗУ БІЗНЕС-ПРОЦЕСУ.....	 31
2.1 Основи штучних нейронних мереж.....	31
2.2 Архітектури багатошарових нейронних мереж	38
2.3 Приклади використання штучних нейронних мереж в бізнес.....	45
 РОЗДІЛ 3 ЗАСТОСУВАННЯ НЕЙРОННОЇ МЕРЕЖІ В: ОПТИМІЗАЦІЇ, ОЦІНЦІ ТА АНАЛІЗУ БІЗНЕС-ПРОЦЕС.....	 47
3.1 Побудова схеми шарів нейронної мережі та методологія графічного моделювання процесів IDEF0.....	47
3.1 Ціль використання нейронної мережі в оптимізації.....	50
3.2 Застосування нейронної мережі для оцінки та прогнозування.....	50
ВИСНОВКИ.....	70
ПЕРЕЛІК ПОСИЛАНЬ.....	71
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)	72

ВСТУП

У сучасному бізнес-середовищі, де конкуренція та динаміка ринку постійно зростають, ефективне управління процесами стає критичним фактором успіху. Традиційні методи аналізу, такі як статистичні моделі або експертні оцінки, часто виявляються недостатніми для обробки великих обсягів даних, врахування складних залежностей та оперативного реагування на зміни. Саме тут на допомогу приходить штучний інтелект, зокрема нейронні мережі, які відкривають нові можливості для оцінки та оптимізації продуктивності бізнес-процесів.

Актуальність цієї теми обумовлена стрімким розвитком технологій машинного навчання та зростанням обсягів даних, які генеруються підприємствами. Нейронні мережі, особливо багатoshарові перцептрони (MLP) та рекурентні мережі (LSTM), демонструють високу ефективність у задачах прогнозування, класифікації та аналізу часових рядів. Вони дозволяють виявляти складні закономірності, які залишаються непомітними при використанні класичних підходів. Практична значимість підтверджується кейсами компаній, які вже досягли значного зниження витрат, підвищення точності прогнозів та зростання прибутку завдяки цим технологіям. У майбутньому нейронні мережі стануть стандартним інструментом бізнес-аналітики, а їх впровадження — необхідною умовою для підтримки конкурентоспроможності.

Ця робота спрямована на те, щоб показати, як сучасні методи ШІ можуть перетворити дані на стратегічні рішення та забезпечити стійкий розвиток підприємств у цифрову епоху.

ТЕОРІЯ І ОСНОВИ БІЗНЕС-ПРОЦЕСІВ ТА ЇХ МЕТОДИ ОПТИМІЗАЦІЇ

1.1 Поняття бізнес-процесу

Основою управління бізнес-процесами є, перш за все, методи, які допомагають побудувати правильну та стабільну модель рішення проблем в організаціях у їх різних сферах діяльності. Швидше, ці структури дають спеціалістам контроль над виконанням роботи та забезпечують можливість для зміцнення та покращення відносин між організаціями та їх клієнтами.

Можна сказати, що це мистецтво та наука, яка будує та впроваджує великі ланцюги подій для оптимізації задач та слідкує за їх вирішенням. Звичайними прикладами можна назвати коли компанії скорочують або збільшують час або грошові витрати при виготовленні якогось продукту та перерозподіляють ресурси на інші потреби, а також реагують при вирішенні проблеми з товаром у клієнта.

Організації, в цілому, керуються цими методами, які ще називаються бізнес-процесами, будь-то комерційні, некомерційні, державні установи та урядові органи влади. У ці процеси включають такі приклади, котрі можна знайти в більшості організацій.

Наприклад:

«Від закупівлі до оплати»: цей тип процесу починається коли у якійсь організації виникає потреба у закупівлі якогось матеріалу або техніки, будь-то будівельна, інженерна чи приватна адвокатська компанія, а закінчується, коли послуги й потреби сплачені. Процеси від закупівлі до оплати має такі дії: отримання, котирування, затвердження покупки, вибір постачальника, оформлення замовлення на купівлю, отримання товару, перевірка та оплата рахунку.

Процес «Від закупівлі до оплати» також є двостороннім бо для нього існує відповідний процес «від котирування до готівки» з боку постачальника.

«Від котирування до готівки»- це процес встановлення ціну за послугу або продукт, яку оголошує продавець або покупець; кінцем є отримання грошей за

послугу. Цей приклад показує, що великі процеси можуть містити дрібні процеси якими неможливо знехтувати та їх треба враховувати при формуванні великих.

«Від проблеми до її вирішення»: суть цього метод в тому, щоб задовольнити потреби користувача після того, як він подав звіт, що при використанні або споживанні продукту він знайшов дефект. Цей процес триває до того поки обидві сторони не будуть задоволені, та поки не виправлять проблему. Таке можна зустріти в торгових та виробничих компаніях.

У кожному бізнес-процесі є свій життєвий цикл, який буде складатись з п'яти етапів: проектування процесів, моделювання, виконання, моніторинг та оптимізація. Кожна фаза життєвого циклу базується на попередній. Через це компанії, які дотримуються життєвого циклу, можуть вдосконалювати свої BPM процеси та зміцнювати їх під нові стратегічні задачі.

Проектування процесів - це перший та головний етап, бо з нього закладається основна BPM ініціатива, який відображає поточний стан ваших бізнес процесів, а також амбіцій і цілей, які ви прагнете досягнути. А саме головне, що ставить низку питань які потрібно поставити, включають:

Якими будуть наші цілі?

Який поточний стан нашого процесу?

Які проблемні точки є у нашому процесі?

Чи будуть нові процеси, які нам потрібно впровадити?

Етап проектування є життєво важливим, бо окреслює перспективи, вигоду та дає очікування для компанії. Також він дає цілісно подивитись на усю картину та визначити шляхи оптимізації.

Моделювання - другий процес, який використовується для чіткого аналізу своїх бізнес-процесів та допомагає визначитися з вхідними та вихідними даними кожного процесу. Вихідні дані - це дані, які отримуємо наприкінці процесу.

Наприклад: продукція або послуги.

Вхідні дані - це ті дані, які долучаються до процесу, наприклад інформація, матеріали або обладнання. Потім треба перейти до креслення моделі, а само до блок-схеми.

Блок-схема - це діаграма, яка слугує нам покроковою схемою сценарію процесу, яка показує нам, як працює система в різних сценаріях. Відобразивши схематично шляхи, компанія отримує чітке уявлення про те, яких вони повинні дотримуватись сценаріїв, та де їх можна покращити. Після цього починається етап автоматизації, який підвищує ефективність та зменшує кількість помилок та допомагає працівникам зосередитись на більш цінній діяльності. Під кінець, треба оцінити ті процеси, які ви сформували з точки зору раціональності, визначити чи підходять вони, чи є процеси важливими для нас, чи можна їх вилучити з процесу, якщо вигоди та витрати не задовільняють.

Виконання - третій етап в, якому повинно впроваджувати бізнес процес, який спланували та сформували для виконання. Перед тим, як впровадити в повному обсязі, треба його по частині вносити в систему та слідкувати за помилками, якщо вони потрапляють та переконатися, що все працює належним чином та дотримується нашої основної ідеї.

Моніторинг - четвертий етап на якому нам треба ефективно контролювати свої бізнес-процеси та для цього нам треба впровадити нові показники ефективності (KPI) для майбутнього відстеження. Вони мають залежність від типу бізнесу, середовища діяльності та конкретних процесів, які намагаємось покращити. При моніторингу показників ефективності ви можете також знайти не надійні та слабкі місця, які потім можна замінити. Це підводить до кінцевого етапу життєвого циклу управління бізнес-процесами.

Оптимізація - п'ятий етап, який дає можливість постійно вдосконалювати усі бізнес-процеси, які ви створили та впровадили. Головна мета роботи процесу – це здобути максимальний прибуток та перевершити бажаний результат. На цьому етапі потрібна ваше прагнення знаходити нові стежки оптимізації цього великого середовища, щоб збільшити прибутки, та зменшити витрати. Це є кінцевим етапом життєвого циклу бізнес-процесу.

1.2 Ідентифікація процесів

Ідентифікація процесів – це комплекс заходів, спрямованих на систематичне визначення набору бізнес-процесів компанії та встановлення чітких критеріїв для їх послідовного встановлення. Результатом ідентифікації процесів є архітектура процесу, яка представляє бізнес-процеси та їх взаємозв'язки.

Архітектура процесу слугує основою для визначення пріоритетів та обсягу проектів моделювання та редизайну процесів. Цей метод ідентифікації процесів, базується на двох фазах: призначенні та оцінці. Фаза призначення стосується визначення початкового списку процесів. Фаза оцінки розглядає відповідні критерії для визначення пріоритетів цих процесів та зосередження уваги на ключових.

Мало з тих організацій, які мають ресурси, необхідні для детального моделювання всіх своїх процесів, а також ретельного аналізу кожного з них та впровадження технологій автоматизації для підтримки кожного з цих процесів і, нарешті, постійного моніторингу ефективності всіх процесів у деталях. Навіть якби такі ресурси були доступні, витратити їх таким чином було б нераціонально. BPM не є безкоштовним. Таким чином, у кожній організації, що впроваджує BPM, вкрай важливо зосередити увагу на самих процесах. Деяким процесам потрібно встановити послідовність та задати правильний пріоритет, оскільки вони впливають та беруть на себе стратегічне значення для виживання організації.

Інші процеси можуть демонструвати суттєві проблеми, які слід вирішити заради всіх зацікавлених сторін. Іншими словами, у процесах, на яких організація повинна зосередитися, знаходяться в сферах, де або створюється велика цінність, або існують значні проблеми.

Деякі процеси можуть бути проблематичними в певний момент. Якщо проблеми були виявлені та вирішені за допомогою програми вдосконалення процесів, організація може обходитися лише періодичними перевірками протягом певного часу. Після того, як цей процес покращиться, а задоволеність клієнтів знову буде в межах бажаного діапазону, акцент може переміститися на процеси

оцінки ризиків, які важливі для довгострокової життєздатності та конкурентоспроможності компанії.

Поза динамікою часу, процеси, які можуть мати стратегічне значення для організації в певний момент, можуть втратити свою важливість з плином часу. Вимоги ринку можуть змінюватися, а нові правила або впровадження нових продуктів можуть обмежувати те, що колись було прибутковою діяльністю. Наприклад, поява нових конкурентів, які пропонують страхові поліси зі знижками через веб-канали, може підштовхнути відому компанію до переосмислення своїх процесів продажу страховок, щоб зробити їх більш компактними, швидшими та доступнішими в інтернеті. Щоб вирішити необхідність зосередження на підмножині ключових процесів, управлінська команда, аналітики процесів та власники процесів повинні мати відповіді на такі два питання:

Які процеси виконуються в організації та на яких з них, організація повинна зосередитись?

Іншими словами, організація, яка впроваджує ініціативи BPM, повинна мати карту своїх процесів, а також чіткі критерії для визначення того, які процеси мають вищий пріоритет. Однак, розуміння необхідне для визначення суми процесів, які потребують ретельного управління або вдосконалення. Збір знань та їх оновлення є саме метою ідентифікації процесів. Більш конкретно, ідентифікація процесів пов'язана з двома послідовними фазами: призначення та оцінка.

Метою “Фази призначення” є розуміння процесів, у яких бере участь організація, а також їхні взаємозв'язки. “Фаза оцінки” заснована на розумінні, встановленому на попередньому етапі, має на меті розробити пріоритети серед них для діяльності з управління цими процесами (моделювання, редизайн, автоматизація, моніторинг тощо). Зауважте, що жоден з цих етапів не стосується розробки детальних моделей та їх процесів.

Коли організація перебуває на самому початку своєї трансформації до системно орієнтованої організації, першим завданням є складання змістовного переліку існуючих процесів. Одна з труднощів виникає через ієрархічну природу бізнес-процесів: можна враховувати різні критерії для визначення того, які

ланцюги процесів можна розглядати, як незалежний бізнес-процес, а які як частину іншого процесу. Існують різні погляди на класифікацію бізнес-процесів (деякі дотримуються думки, що насправді в організації дуже мало процесів).

Наприклад: деякі дослідники стверджують, що існує лише два процеси - управління продуктовою лінійкою та управління циклом замовлень. Інші виділяють три основні процеси: розробка нового продукту, доставка продукту клієнтам та управління взаємовідносинами з клієнтами. Також треба підвести короткий опис процесів таких, як BPMN.

1.3 Моделювання візуальних процесів BPMN та IDEF.

Моделювання бізнес-процесів(BPMN) – це мова візуального моделювання для бізнес-аналізу та опису бізнес-процесів. Це відкритий стандарт нотації для графічних блок-схем, що використовуються для визначення робочих процесів та бізнес-процесів. Це популярна та інтуїтивно зрозуміла діаграма, яку легко зрозуміти всім зацікавленим сторонам в організації, включаючи бізнес-користувачів, бізнес-аналітиків, розробників програмного забезпечення та архітекторів даних. BPMN дозволяє нам фіксувати та документувати бізнес-процеси організації чітким та послідовним чином, що забезпечує залучення до процесу відповідних зацікавлених сторін, таких як власники процесів та бізнес-користувачі. Таким чином, команда може ефективніше реагувати на будь-які проблеми, виявлені в процесах. BPMN надає вичерпні та водночас змістовні нотації, які легко зрозуміти як технічним, так і нетехнічним зацікавленим сторонам. Моделювання бізнес-процесів надає важливі переваги компаніям та організаціям, таким як перелічені нижче.

Галузевий стандарт, розроблений консорціумом OMG, некомерційною галузевою групою, надає підприємствам можливість визначати та розуміти свої процедури за допомогою діаграм бізнес-процесів. Забезпечує стандартну нотацію, яку легко зрозуміти всім зацікавленим сторонам бізнесу. Долає комунікаційний розрив, який часто виникає між проектуванням та впровадженням бізнес-процесів.

Простий у вивченні та достатньо потужний, щоб зобразити потенційні складнощі бізнес-процесу .

Розуміння того, як працює бізнес, є першим і найважливішим кроком удосконалення бізнес-процесів. Модель та нотація бізнес-процесів (BPMN) забезпечують графічне представлення бізнес-процесів, яке може легко зрозуміти будь-хто, від бізнес-аналітика до зацікавленої сторони; це допомагає в аналізі та вдосконаленні бізнес-процесів. Будь-який процес, описаний за допомогою BPMN, представлений, як низка кроків діяльності, які виконуються послідовно або одночасно відповідно до певних бізнес-правил. Приклад за методом та модель нотації BPMN.

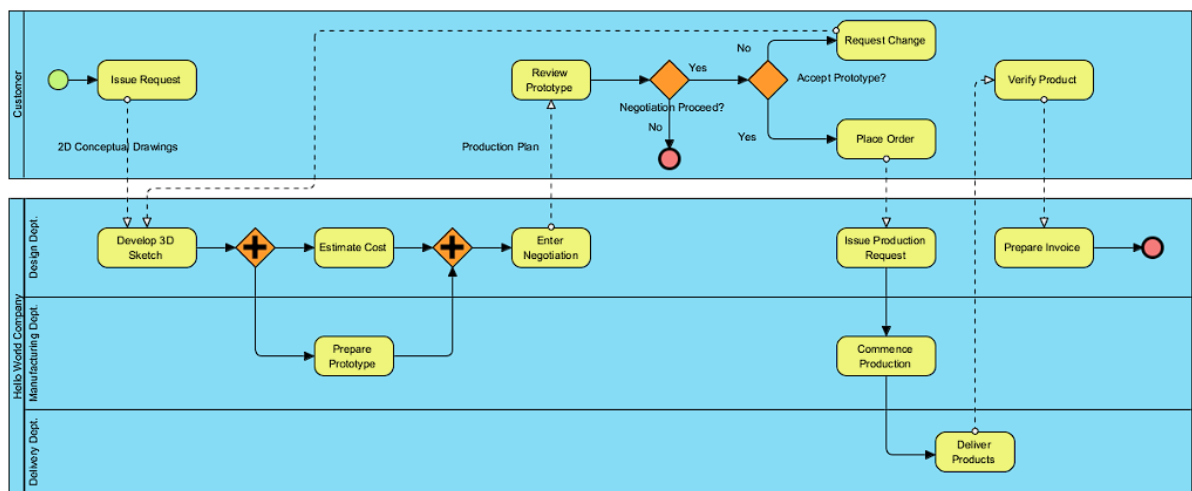


Рисунок: 1.1 Приклад за методом та модель нотації BPMN.

У BPMN процеси описуються за допомогою діаграм з низкою графічних елементів. Таке візуальне представлення полегшує користувачам розуміння логіки процесу. BPMN була розроблена, в першу чергу, для проектування та читання як простих, так і складних діаграм бізнес-процесів. Для цього стандарт BPMN класифікує графічні елементи за категоріями: в результаті елементи легко розпізнаються користувачами, які працюють з діаграмами бізнес-процесів.

IDEF (Інтегроване визначення) — це методологія графічного моделювання процесів, яка застосовується для проектування систем, розробки програмного забезпечення, функціонального аналізу даних, імітаційного моделювання, об'єктно-орієнтованого аналізу та роботи з знаннями.

Методологія була розроблена Повітряними силами США у 1970-х роках, як стандарт для документування та аналізу бізнес-процесів. Сьогодні IDEF використовується як структурований підхід до аналізу діяльності підприємств, фіксації поточних процесів («як є») та моделювання роботи бізнес-підрозділів. Хоча спочатку IDEF створювався для виробничих потреб, зараз його застосовують у різних сферах, зокрема в розробці програмного забезпечення.

1.4 Показники продуктивності у бізнес-процесах

Що таке ключові показники ефективності? Ключові показники ефективності – це кількісні показники, що використовуються для оцінки довгострокової загальної ефективності компанії. Організації використовують ключові показники ефективності для відстеження прогресу в досягненні ключових бізнес-цілей. Ключові показники ефективності допомагають визначити стратегічний, фінансовий та операційний успіх компанії, особливо порівняно з успіхом інших компаній у тій самій галузі. Їх також можна використовувати для оцінки прогресу або досягнень порівняно з набором контрольних показників або минулими показниками. Ключові показники ефективності вимірюють успіх компанії відносно набору цілей, завдань або галузевих аналогів. Ключові показники ефективності (KPI) можуть мати фінансовий характер, включаючи чистий прибуток (або чистий дохід), обсяг продажів за вирахуванням певних витрат або коефіцієнти ліквідності (ліквідність та наявність грошових коштів).

Клієнтоорієнтовані показники ефективності зазвичай зосереджені на якості обслуговування клієнтів, задоволеності клієнтів та лояльності клієнтів. Ключові показники ефективності, орієнтовані на процеси, спрямовані на вимірювання та моніторинг операційної ефективності організації. Компанії зазвичай вимірюють та відстежують ключові показники ефективності за допомогою аналітичного програмного забезпечення та інструментів звітності. В організаціях ключові показники ефективності використовуються для оцінки результативності та прогресу в досягненні конкретних, вимірюваних цілей. Таким чином, вони

надають власникам та менеджерам уявлення про ефективність їхнього бізнесу або певного аспекту їхнього бізнесу на певний момент часу. Їх можна порівняти з заданою точкою відліку, іншими конкурентами в галузі; показники діяльності компанії з плином часу.

Ключові показники ефективності (KPI), також відомі, як ключові показники успіху (KPI), різняться залежно від компаній та галузей.

Наприклад: компанія-розробник програмного забезпечення, яка прагне стати найшвидше зростаючою компанією у своїй галузі, може розглядати річне зростання доходу, як ключовий показник ефективності. І навпаки, роздрібна мережа може приділяти більше уваги продажам в одному магазині, як найкращому ключовому показнику ефективності для вимірювання зростання. Ключові показники ефективності зосереджені на зборі, зберіганні, очищенні та узагальненні даних.

Дані KPI збираються та порівнюються з встановленими цілями. Результати цього порівняння потім аналізуються та використовуються для висновків про те, як поточні системи або нещодавні зміни в цих системах сприяють досягненню цілей відділу або організації. Це дозволяє керівництву знати, чи ефективні поточні системи, чи потрібно внести зміни для покращення результатів та досягнення майбутніх цілей.

Мета ключових показників ефективності полягає в лаконічному представленні результатів, щоб керівництво могло приймати більш обґрунтовані стратегічні рішення. Їх часто вимірюють за допомогою аналітичного програмного забезпечення та інструментів звітності. Інформація з ключових показників ефективності може мати фінансовий або нефінансовий характер і може стосуватися окремих відділів компанії або ефективності компанії в цілому. Компанії можуть використовувати ключові показники ефективності на трьох загальних рівнях. Бізнес: ключові показники ефективності для всього підприємства зосереджені на загальному стані та ефективності бізнесу. Цей тип ключового показника ефективності корисний для інформування керівництва про

операційну ситуацію в компанії в цілому. Однак вони часто недостатньо деталізовані, щоб дозволити приймати рішення. Ключові показники ефективності компанії часто викликають дискусії про те, чому певні відділи працюють добре або погано. У відділах з покращенням KPI ефективність більша, ніж у відділах з KPI загально корпоративного рівня, часто дають уявлення про те, чому трапляються певні результати. Компанії часто аналізують ключові показники ефективності (KPI) на рівні відділів, щоб краще зрозуміти результати KPI загалом у масштабах підприємства. Наприклад, якщо загальний обсяг продажів знижується, компанія може захотіти дослідити коефіцієнти конверсії або рівень задоволеності клієнтів у певних відділах.

Проект або підрозділ: якщо компанія хоче детальніше розглянути питання, вона може розглянути ключові показники ефективності на рівні проекту або підрозділу. Ці ключові показники ефективності часто потрібно запитувати у керівництва, оскільки вони вимагають дуже специфічних наборів даних, які не обов'язково легкодоступні. Наприклад, керівництво може захотіти провести опитування фокус-групи щодо потенційного запуску продукту. Як визначити ключові показники ефективності (KPI)?

Для бізнесу важливо створити ключові показники ефективності, які відповідають його бізнес-цілям та результатам. Це означає, що ключові показники ефективності є специфічними для бізнесу, а не просто набором загальних правил. Ось деякі міркування, які слід враховувати під час встановлення ключових показників ефективності. Встановіть чіткі цілі: KPI повинні бути пов'язані з основною бізнес-ціллю, пов'язаною з успіхом компанії, будь то фінансова, орієнтована на обслуговування клієнтів, маркетингова чи інша ціль. Обов'язково визначте, що таке успіх, і як його вимірювати. Якщо бізнес не дотримується цих кроків, він ризикує витратити дорогоцінний час і ресурси, досягнувши результатів, які не допомагають йому досягти своїх цілей.

Повідомлення ключових показників ефективності (KPI): Співробітники повинні розуміти та погоджуватися з KPI компанії. Тому важливо донести їх та

пояснити, чому цілі є важливими та як їх буде досягнуто. Працівники компанії можуть порушувати важливі питання та занепокоєння, на які необхідно звернути увагу, щоб забезпечити значущість ключових показників ефективності. Переглядайте свої ключові показники ефективності: регулярний перегляд ваших ключових показників ефективності гарантує їх актуальність. Бізнес-середовище постійно змінюється, тому вам може знадобитися адаптувати свої KPI до показників ефективності, розвитку та еволюції компанії.

Більшість KPI поділяються на чотири широкі категорії. Кожна категорія має свої характеристики, часові рамки та рівень бізнесу, який, ймовірно, її використовуватиме. Різні KPI також можуть використовуватися різними відділами в межах одного бізнесу.

Стратегічні KPI зазвичай є найвищого рівня. Ці типи KPI можуть вказувати на те, як працює компанія, хоча вони не надають багато інформації, окрім загального огляду. Керівники найчастіше використовують стратегічні KPI. Приклади включають рентабельність інвестицій, норму прибутку та загальний дохід компанії.

Операційні KPI зосереджені на обмежених часових рамках. Ці KPI вимірюють, як компанія працює місяць за місяцем, або іноді день за днем, шляхом аналізу різних процесів, сегментів або географічних місць. Операційні KPI часто використовуються керівництвом персоналу для аналізу питань, що виникають в результаті аналізу стратегічних KPI. Наприклад, якщо керівник помічає, що дохід компанії знизився, він може дослідити, які лінійки продуктів мають труднощі.

Функціональні KPI зосереджені на конкретних відділах або функціях у компанії. Наприклад: фінансовий відділ може щомісяця відстежувати, скільки нових постачальників вони реєструють у своїй системі бухгалтерського обліку. Відділ маркетингу вимірює, скільки клієнтів отримує кожна розсилка електронної пошти.

Ці типи ключових показників ефективності (КРІ) можуть бути стратегічними або операційними. Їх відрізняє те, що вони надають найбільшу цінність одній конкретній групі користувачів.

Випереджувальний та відсталий КРІ описують характер аналізованих даних на те, чи сигналізують вони про щось майбутнє, чи про щось, що вже відбулося. Випереджувальні КРІ вказують на зміни, які настають у майбутньому. Відсталий КРІ вказують на зміни, які вже відбулися. Прикладами цього є кількість понаднормових годин та норми прибутку для флагманського продукту. Кількість понаднормових годин може бути провідним КРІ, якщо компанія почне помічати погіршення якості виробництва. Як варіант, норми прибутку є результатом операційної діяльності та вважаються відсталим показником.

Фінансові ключові показники ефективності, зазвичай вони зосереджені на доході та нормі прибутку. Чистий прибуток є найнадійнішим показником. Цей показник являє собою суму доходу, яка залишається, як прибуток за певний період після врахування всіх витрат, податків та відсоткових платежів компанії за той самий період. Фінансові показники можна отримати з фінансової звітності компанії. Однак, внутрішнє керівництво може вважати кориснішим аналізувати різні показники, які є більш специфічними для аналізу проблем або аспектів компанії, які керівництво хоче проаналізувати. Наприклад, компанія може використовувати змінні витрати для перерахунку певних залишків на рахунках лише для внутрішнього аналізу. Приклади фінансових ключових показників ефективності КРІ:

коефіцієнти ліквідності: КРІ, які вимірюють, наскільки добре компанія керуватиме короткостроковими борговими зобов'язаннями на основі наявних у неї короткострокових активів. Також відомі як коефіцієнти поточної ліквідності, які ділять поточні активи на поточні зобов'язання.

коефіцієнти рентабельності: КРІ, які вимірюють, наскільки добре компанія генерує продажі, зберігаючи при цьому низькі витрати. Прикладом є чиста рентабельність.

коефіцієнти платоспроможності: KPI, які вимірюють довгострокове фінансове здоров'я компанії, оцінюючи, наскільки добре компанія зможе сплачувати довгострокові борги. Прикладом є співвідношення загальної заборгованості до загальних активів.

коефіцієнти оборотності: KPI, які вимірюють, як швидко компанія може виконати певне завдання. Наприклад, оборотність запасів вимірює, як швидко компанія може перетворити товар із запасів на товар для продажу. Компанії прагнуть збільшити оборотність, щоб швидше генерувати відтік витратних коштів, а потім повертати ці кошти від доходів.

Метрики клієнтського досвіду та ключові показники ефективності, орієнтовані на клієнта, зазвичай зосереджені на кожному клієнті, задоволеності клієнтів та утриманні клієнтів. Ці показники використовуються командами служби підтримки клієнтів для кращого розуміння послуг, які отримують клієнти. Приклади показників, орієнтованих на клієнта, включають:

Кількість нових запитів: підраховує запити на обслуговування клієнтів та вимірює кількість нових проблем, які виникають у клієнтів.

Кількість вирішених запитів: підраховує кількість запитів, які були успішно оброблені. Порівнюючи кількість запитів з кількістю вирішень, компанія може оцінити свій рівень успішності в обробці запитів клієнтів.

Середній час вирішення: середній час, необхідний для допомоги клієнту з проблемою. Компанії можуть сегментувати середній час вирішення за різними запитами, наприклад, запити на технічні проблеми проти запитів на нові облікові записи. Середній час відповіді: середній час, необхідний агенту служби підтримки клієнтів для першого зв'язку з клієнтом після того, як клієнт подав запит. Хоча початковий агент може не мати знань чи досвіду для надання рішення, компанія може цінувати скорочення часу, протягом якого клієнт чекає на будь-яку допомогу.

Найкращий агент служби підтримки клієнтів: комбінація будь-яких вищезазначених показників, зіставлених представниками служби підтримки

клієнтів. Наприклад: окрім аналізу середнього часу відповіді по всій компанії, компанія може визначити трьох найшвидших та найповільніших респондентів.

Кількість проблем у різних типів запитів. Цей ключовий показник ефективності (KPI) може допомогти компаніям краще зрозуміти проблеми, які може мати клієнт (наприклад, веб-сайт компанії надавав неправильні або неточні вказівки), які компанія повинна вирішити.

Рейтинг задоволеності клієнтів: багато компаній можуть проводити опитування або анкетування, щоб зібрати додаткову інформацію про досвід клієнта, хоча це розпливчастий та неточний показник.

Метою метрик процесу є вимірювання та моніторинг операційної ефективності в організації. Ці ключові показники ефективності KPI аналізують, як виконуються завдання, та чи є проблеми з процесами, якістю або продуктивністю, або які потребують покращень.

Ці типи метрик найбільш корисні для компаній з повторюваними процесами, таких, як виробничі фірми або компанії в циклічних галузях. Приклади метрик ефективності процесу включають:

Ефективність виробництва: Часто вимірюється, як час виробництва для кожного етапу, поділений на загальний час обробки. Наприклад: компанія може прагнути витратити лише 2% свого часу на закупівлю сировини. Якщо вона виявляє, що це займає 5% від загального часу процесу, компанія знає, що цю область потребує покращення.

Загальний час циклу: загальна кількість часу, необхідна для завершення процесу від початку до кінця. Це можна перетворити на середній час циклу, якщо керівництво бажає проаналізувати процес протягом тривалого періоду.

Пропускна здатність: кількість вироблених одиниць, поділена на час виробництва на одиницю, що вимірює швидкість виробничого процесу.

Коефіцієнт помилок: загальна кількість помилок, поділена на загальну кількість вироблених одиниць. Компанія, яка прагне зменшити кількість відходів, може використовувати цей показник для розуміння кількості виробів, які не проходять перевірку контролю якості.

Показник якості: міра кількості вироблених виробів, які пройшли перевірку контролю якості. Поділивши кількість успішно завершених одиниць на загальну кількість вироблених одиниць, цей відсоток інформує керівництво про рівень успішності у дотриманні стандартів якості.

1.5 Рентабельність інвестицій

Рентабельність інвестицій (ROI) – це популярний показник прибутковості, який використовується для оцінки ефективності інвестицій. ROI виражається у відсотках і розраховується шляхом ділення чистого прибутку або збитку інвестицій на їх початкову вартість або витрати. ROI може використовуватися для порівняння фактичних даних та ранжування інвестицій у різні проєкти або активи. ROI не враховує період володіння або плин часу, тому може не враховувати альтернативні витрати інвестування в інші місця. Чи забезпечує щось хорошу рентабельність інвестицій ROI, слід порівнювати з іншими доступними можливостями.

Як розрахувати рентабельність інвестицій Формула рентабельності інвестицій виглядає наступним чином:

$$ROI = \frac{\text{Current Value of Investment} - \text{Cost of Investment}}{\text{Cost of Investment}}$$

Рисунок 1.2 Формула рентабельності.

«Поточна вартість інвестицій» означає надходження, отримані від продажу інвестицій, що становлять інтерес. Цей розрахунок включає такі фактори, як грошовий потік протягом усього життєвого циклу інвестиції та будь-якої понесеної витрати на обслуговування. Оскільки ROI вимірюється у відсотках, її

можна легко порівняти з дохідністю інших інвестицій, що дозволяє оцінювати різні типи інвестицій один з одним. Чому ROI є корисним показником?

ROI є популярним показником завдяки своїй універсальності та простоті. По суті, ROI можна використовувати, як рудиментарний показник прибутковості інвестицій. Це може бути рентабельність інвестицій в акції, рентабельність інвестицій, яку компанія очікує від розширення фабрики, або рентабельність інвестицій, отримана в результаті операції з нерухомістю. Сам розрахунок не надто складний і його відносно легко інтерпретувати для широкого спектра застосувань.

Якщо рентабельність інвестицій є позитивною, вона, ймовірно, варта уваги. Але якщо доступні інші можливості з вищою рентабельністю інвестицій, ці сигнали можуть допомогти інвесторам виключити або вибрати найкращі варіанти. Так само інвесторам слід уникати негативної рентабельності інвестицій, яка означає чистий збиток.

Наприклад, припустимо, що Джо інвестував 1000 доларів у Slice Pizza Corp. у 2017 році та продав акції за загальну суму 1200 доларів через рік. Щоб розрахувати рентабельність цих інвестицій, розділіть чистий прибуток 1200 доларів - 1000 доларів = 200 доларів на вартість інвестицій (1000 доларів), щоб отримати рентабельність інвестицій у розмірі 200 доларів/1000 доларів, або 20%. Маючи цю інформацію, можна порівняти інвестиції в Slice Pizza з будь-якими іншими проєктами. Припустимо, що Джо також інвестувала 2000 доларів США в Big-Sale Stores Inc. у 2014 році та продала акції на загальну суму 2800 доларів США у 2017 році. ROI (рентабельність інвестицій) активів Джо в Big-Sale становитиме 800 доларів США/2000 доларів США, або 40%. Порівнюючи ці інвестиції, також важливо враховувати кількість років, протягом яких кожна інвестиція була утримана. Приклади, подібні до прикладу Джо (вище), показують деякі обмеження використання ROI, особливо під час порівняння інвестицій. Хоча ROI другої інвестиції Джо була вдвічі більшою, ніж у першій інвестиції, час між покупкою та продажем Джо становив один рік для першої інвестиції, але три роки для другої. Джо могла відповідно скоригувати ROI багаторічної інвестиції.

Оскільки загальна ROI становила 40%, щоб отримати середньорічну ROI, Джо могла поділити 40% на 3, щоб отримати 13,33% річних. З цим коригуванням виявляється, що хоча друга інвестиція Джо принесла більший прибуток, перша інвестиція насправді була ефективнішим вибором.

ROI можна використовувати разом із нормою прибутковості (RoR), яка враховує часові рамки проєкту. Також можна використовувати чисту приведену вартість (NPV), яка враховує різницю у вартості грошей з часом через інфляцію.

Застосування NPV під час розрахунку RoR часто називають реальною нормою прибутковості. Що таке хороша ROI? Визначення того, що вважається «хорошою» ROI, має вирішальне значення для інвесторів, які прагнуть максимізувати свою прибутковість, одночасно керуючи ризиком. Хоча універсального орієнтира немає, кілька факторів впливають на те, що вважається кращим.

Схильність до ризику: Інвестори різняться у своїй готовності терпіти ризик. Ті, хто більше уникає ризику, можуть погодитися на нижчу ROI в обмін на більшу стабільність та передбачуваність своїх інвестицій. З іншого боку, інвестори, толерантні до ризику, можуть прагнути вищої ROI, але готові погодитися на більшу невизначеність та волатильність.

Тривалість інвестицій: Часовий горизонт інвестицій відіграє значну роль у визначенні того, що кваліфікується як хороша ROI.

1.6 Методи оптимізації.

Якщо в будь-якому з процесів виникають проблеми, то вирішуємо їх за допомогою впроваджених методів оптимізації. Методи оптимізації процесу допомагають підготувати їх до впровадження у власному бізнесі. Що таке оптимізація процесів?

Оптимізація процесів – це впровадження структурованих методів, стратегій, дисциплін та тактик для покращення певного процесу в рамках проєкту чи ініціативи. Це старий спосіб налаштування процесу, щоб він працював краще, ніж раніше. Пропуск, видалення або перезапис кроку процесу – це прості приклади

оптимізації робочого процесу. Оптимізацію процесів можна загалом визначити, як низку кроків, які включають до себе:

Визначення: це включає визначення процесу, який потребує оптимізації. Якщо процес занадто дорогий або не задовольняє клієнта, визначення першопричин є важливим для успіху будь-якого методу оптимізації.

Осмислення: як визначити процеси, які вам потрібно знати та над якими потрібно працювати. Питання тривалості процесу або кращого методу реалізації – це лише деякі з напрямків переосмислення запланованого процесу.

Впровадження: це крок впровадження до процесу. Під час аналізу результатів та внесення необхідних коригувань, щоб забезпечити їх належну роботу, враховуються наступні кроки:

Автоматизація: передбачає, автоматизацію процесу протягом усього робочого циклу з метою його оцінки на основі бажаних результатів, таких як зниження витрат, покращення продукту або запобігання помилок. Це стосується автоматизації, яка гарантує, що процес працює автоматично.

Моніторинг: Це стосується відстеження після успішної інтеграції. Це дозволяє виявити нові області для покращення або критичні проблеми, які необхідно вирішити, щоб проєкт або робочий процеси могли й надалі працювати без помилок. Які переваги методів оптимізації процесів?

Використання методів оптимізації процесів пропонує багато переваг, таких як: **Зниження ризиків:** Успішне впровадження методу може зменшити ризики, що виникають, коли неефективний процес залишається незмінним. Це допоможе вам краще виявляти та усувати потенційно нові ризики, перш ніж вони матимуть надто негативний вплив на ваш проєкт або робочий процес. **Забезпечення більшої узгодженості:** покращення узгодженості процесів дозволяє усунути зайву роботу, яка не служить жодній меті. Це може полегшити вам перехід між процесами без необхідності повторювати кроки чи дії. **Покращення якості:**

Метод оптимізації справді корисний, якщо ви хочете покращити якість свого процесу, проєкту чи робочого процесу. Якщо ви зможете досягти

високоякісних результатів, ви заощадите собі інвестиції в додаткові виправлення або коригувальні заходи.

Операційна оптимізація: Однією з головних переваг методу оптимізації є оптимізація ваших операційних процесів. Це дозволяє виконати всю операцію з мінімальними збоями. Це може дозволити всім учасникам працювати разом у злагодженому та спільному режимі. **Покращення управління ресурсами:** Якщо процес вимагає використання цінних ресурсів, метод оптимізації може допомогти вам їх зберегти. Ви можете зменшити кількість витрачених ресурсів та розподілити їх на кращі цілі та способи використання.

Економія коштів: Оптимізуючи критично важливий процес у вашому робочій системі, ви з більшою ймовірністю заощадите гроші та залишитеся в рамках свого бюджету. Якщо впровадити метод оптимізації успішно, він може збільшити ваші продажі та фінансові вигоди. **Підвищення продуктивності:** Виключивши неефективний процес або змінивши спосіб його роботи, ви можете підвищити загальну продуктивність вашого робочого процесу. Це може допомогти вам покращити доставлення та швидше впоратися з роботою.

Мінімізація майбутніх помилок: щойно ви краще оптимізуєте процес у своєму робочій системі, то помилки, ймовірно, траплятимуться рідше. Коли трапляється менше помилок, ви можете зосередити свої зусилля на інших процесах та більш важливих бізнес-питань. **Усунення проблем:** Усунення проблем дозволяє усунути критичні вузькі місця в робочому процесі, щоб процес міг продовжуватися безперебійно. З меншою кількістю проблем ви можете виконати свою роботу швидше та матимете час для вирішення інших накопичених процесів та завдань.

Також: Що таке автоматизація робочих процесів і як вона використовується? Ось вісім методів оптимізації процесів, які використовуються:

Метод оптимізації PDSA — це циклічна система, що складається з чотирьох частин, яка допомагає вам покращити якість процесів та оптимізувати ваш бізнес. Кроки цього методу такі: **Планування:** Першим кроком у цьому методі є чіткий опис та визначення успіхів, яких ви хочете досягти. **Впровадження:** Другий крок

полягає у тестуванні можливих змін у невеликому, обмеженому масштабі. Дослідження: Третій крок полягає у вивченні результатів методу та визначенні його корисності. Дія: Останнім кроком є впровадження змін у більших масштабах.

Метод процесного інтелекту Метод оптимізації процесного інтелекту – це комбінація методів, що включають елементи науки про дані. Цей метод передбачає вилучення даних із журналу подій, а потім аналіз дій членів вашої команди в організації шляхом вивчення кроків, які вони виконують для виконання завдання або завдань. Потім ви можете перетворити зібрані дані на корисну інформацію, допомагаючи іншим фахівцям виявляти проблеми та оптимізувати необхідні процеси.

Метод DMAIC Метод оптимізації DMAIC : Ця методологія працює як цикл кроків, що включає: Визначення: Перший крок полягає у визначенні процесів, які потребують оптимізації. Вимірювання: Другий крок полягає у вимірюванні та визначенні ефективності процесу. Аналіз: Третій крок – проаналізувати, як можна оптимізувати процес. Удосконалення: Четвертий крок – це удосконалення процесу. Моніторинг: П'ятий крок – це моніторинг майбутньої продуктивності нової або оновленої послуги.

Метод управління проектами ВРО. Метод оптимізації управління проектами ВРО: — складається з низки кроків, які включають: Ініціація: Першим кроком є авторизація фази, процесу або проекту як частини загальної ініціативи. Планування: Другий крок — визначити та обміркувати свої цілі, а потім вибрати найкращі дії, які допоможуть вам їх досягти. Виконання: Третій крок – координація та співпраця з людьми, які допоможуть вам виконати план або стратегію. Моніторинг: Четвертий крок, полягає у забезпеченні досягнення кожної поставленої мети шляхом регулярного моніторингу та вимірювання прогресу й результатів, а також вжиття коригувальних заходів за необхідності. Завершення: Останнім кроком є офіційне закриття проекту після того, як метод оптимізації досяг своєї мети.

Метод оптимізації фіксування потоку та створення цінності – це графічне представлення різних матеріалів, даних та інформації, що циркулюють у рамках проєкту чи ініціативи. Головна мета — надавати послуги та продукти клієнтам або кінцевим користувачам. Цей метод справді корисний для досягнення таких результатів: Виявлення та усунення марнування ресурсів. Отримайте цінну інформацію про процеси прийняття рішень та робочі процеси. Встановіть вимірні та реалістичні цілі для покращення процесів. Визначте області, які потребують покращення.

Метод оптимізації Кайдзен може працювати з методом PDCA. Це техніка, спрямована на регулярне вдосконалення всіх функцій компанії. До переваг цього методу, серед іншого, належать: сприяння швидшому отриманню результатів та заходи безпеки. Підвищити задоволеності роботи та продуктивності команди. Удосконалення всіх бізнес-процесів. Покращуйте якість продукції та задоволеності клієнтів.

Метод оптимізації DMADV має на меті допомогти вам підвищити показник якості. Цей метод вносить радикальні зміни та повністю замінює старий процес на новий. Переваги цього методу не обмежується: Збільшенням продажів і прибутку, покращенням відгуків та задоволеності клієнтів Зменшення кількості помилок, що виникають.

Метод оптимізації аналізу SIPOC корисний для впорядкування зібраних даних про клієнтів та продукти, що беруть участь у процесах. Цей метод може бути корисним з кількох причин, зокрема: Визначення всіх відповідних аспектів, які потребують покращення. Визначення складних проєктів та їх осмислення. Розуміння того, як має працювати процес: методів, які ви можете використовувати для покращення бізнес-процесів.

Щоб покращити процеси: Чітко визначте свої бізнес-цілі, оскільки саме на них ви зазвичай будете спиратись, щоб правильно застосувати свої сили на покращення процесів які вам дійсно потрібні. Ретельно плануйте свою стратегію, оптимізації та працюйте над покращенням результатів. Спробуйте передбачити певні результати обраного вами методу оптимізації, перш ніж розпочати його

застосування. Проведіть контрольоване дослідження нового технологічного процесу перед його впровадженням у більших масштабний, а потім проаналізуйте результати. Впроваджуйте та вдосконалюйте процеси, включаючи необхідні зміни, якщо потрібно. Продовжуйте відстежувати та контролювати результати вашої оптимізації. Спробуйте автоматизувати деякі аспекти процесу без шкоди для якості.

ЗАСТОСУВАННЯ НЕЙРОНОЇ МЕРЕЖІ ЯК ІНСТРУМЕНТУ В: ОЦІНЦІ, ОПТИМІЗАЦІЇ ТА АНАЛІЗУ-БІЗНЕС ПРОЦЕСУ

2.1 Основи штучних нейронних мереж.

Основою штучних нейронних мереж є принцип який розроблений вченим Френком Розенблатом у 1950-х і 1960-х роках та дав йому назву Перцептрон. Перцептрон вважається однією з найпростіших архітектур штучних нейронних мереж. Він використовується для двійкової класифікації. У той час для прогнозування широко використовувалися методи, такі як статистичне машинне навчання та традиційне програмування. Хоча модель перцептрона є однією з найпростіших форм штучних нейронних мереж, вона виявилася дуже ефективною у розв'язанні задач класифікації, що формує основу для розвитку штучного інтелекту та машинного навчання. Перцептрон складається з набору вхідних вузлів, які повністю з'єднані з набором вихідних вузлів(2.1).

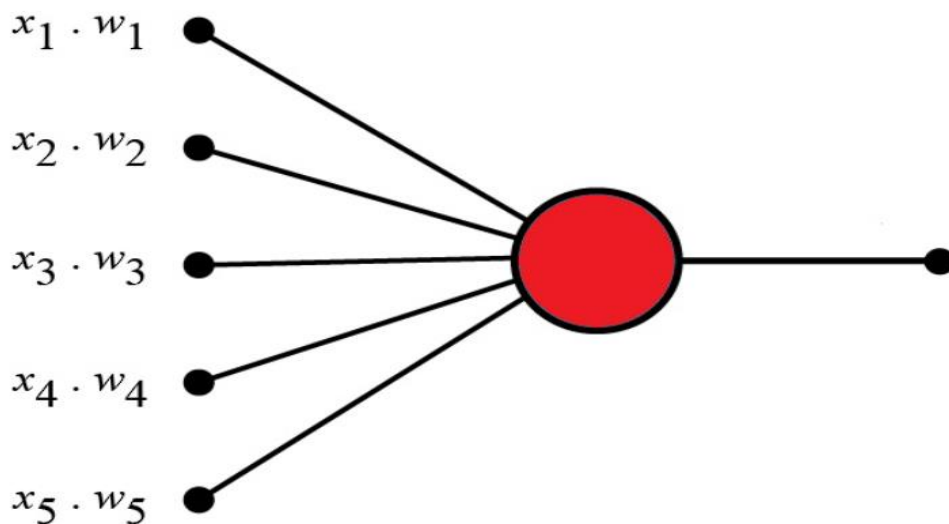


Рисунок 2.1 Структура перцептрона

Перцептрон використовує варіацію функціональних потоків, які називаються пороговими логічними одиницями. Функціональність: Перцептрон отримує кілька вхідних сигналів, кожен з яких представляє унікальний шаблон інформації. Вага: Кожному атрибуту присвоюється унікальна вага.

Ці параметри коригуються під час навчання, щоб знайти оптимальні значення.

Як працює перцептрон? Кожному вхідному вузлу перцептрона присвоюється вага, що вказує на важливість цього входу для визначення виходу. Вихід перцептрона обчислюється як зважена сума вхідних даних, яка потім проходить через функцію активації, щоб вирішити, чи спрацює перцептрон. Зважена сума обчислюється так:

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n = X^TW$$

Рисунок 2.2 Зважена сума

Крокова функція порівнює цю зважену суму з порогом. Якщо вхідний сигнал перевищує порогове значення, вихідний сигнал дорівнює 1; інакше – 0. Це найпоширеніша функція активації, що використовується в перцептронах, представлена кроковою функцією Хевісайда:

$$h(z) = \begin{cases} 0 & \text{if } z < \text{Threshold} \\ 1 & \text{if } z \geq \text{Threshold} \end{cases}$$

Рисунок 2.3 функція Хевісайда

Перцептрон складається з одного шару порогових логічних одиниць (TLU), причому кожна TLU повністю підключена до всіх вхідних вузлів

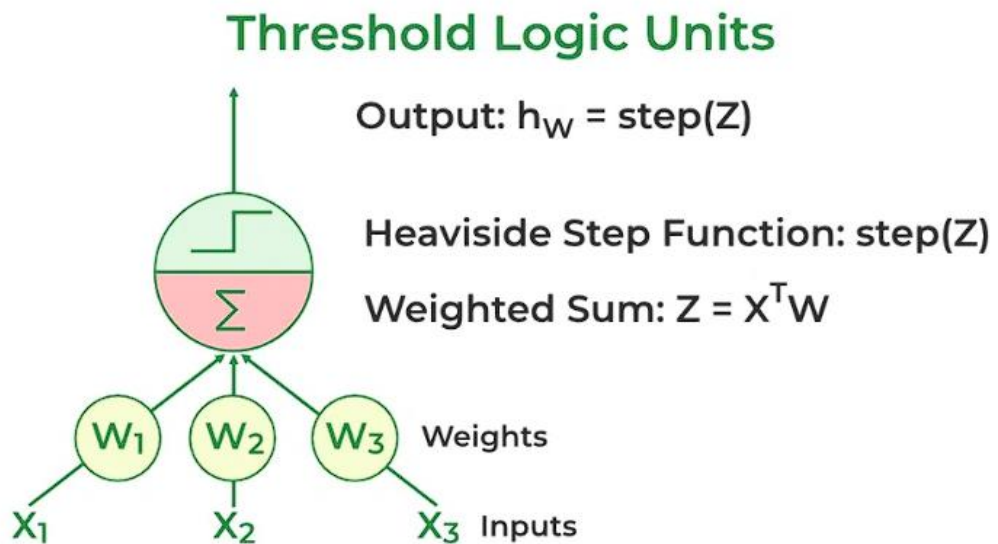


Рисунок 2.4 візуальний опис одиниць TLU

У повністю зв'язному шарі, також відомому як щільний шар, усі нейрони одного шару з'єднані з кожним нейроном попереднього шару.

Вихід повністю зв'язного шару обчислюється як:

$$f_{W,b}(X) = h(XW + b)$$

Рисунок 2.5 обчислення зв'язного шару.

де X – вхідні дані, W – вага для кожного вхідного нейрона, b – зміщення, а h – крокова функція. Під час навчання ваги перцептрона коригуються, щоб мінімізувати різницю між прогнозованим виходом і фактичним виходом. Це досягається за допомогою алгоритмів навчання з учителем, таких як правило дельта або правило навчання перцептрона. Формула оновлення ваг така:

$$w_{i,j} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

Рисунок 2.6 Формула оновлення ваг.

Де: W – вага між X_i вхідним та h_j -м вихідним нейроном, x_i – h_i -м вхідним значенням, y_j – фактичне значення, $\hat{y}_j$ – прогнозоване значення, η – швидкість навчання, що контролює, наскільки коригуються ваги.

Модель перцептрона приймає кілька вхідних даних з дійсними значеннями та видає один двійковий вихід. У моделі перцептрона кожен вхідний сигнал X і має вагу W і пов'язану з ним. Ваги вказують на важливість вхідних даних у процесі прийняття рішень. Вихідні дані моделі визначають за допомогою порогу W_0 : якщо зважена сума вхідних даних перевищує поріг W_0 , вихідні дані дорівнюватимуть 1, інакше вихідні дані дорівнюватимуть 0. Іншими словами, модель спрацює, якщо зважена сума перевищує поріг.

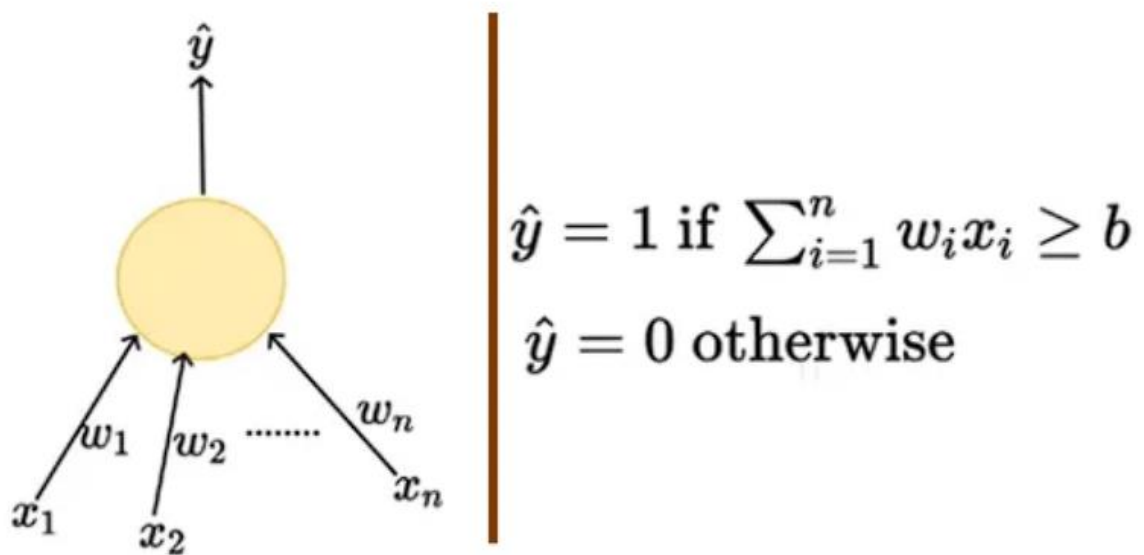


Рисунок 2.7 Модель перцептрона.

З математичного представлення можна сказати, що логіка встановлення порогу, що використовується перцептроном, є дуже жорсткою. Логіку жорсткого встановлення порогу на прикладі. Розглянемо процес прийняття рішення людиною, чи хоче вона придбати автомобіль, чи ні, на основі лише одного вхідного значення X_1 — зарплата, встановивши поріг $b(W_0) = -10$ та вагу $W_1 = 0,2$. Вихідні дані моделі перцептрона виглядатимуть, як показано на рисунку нижче.

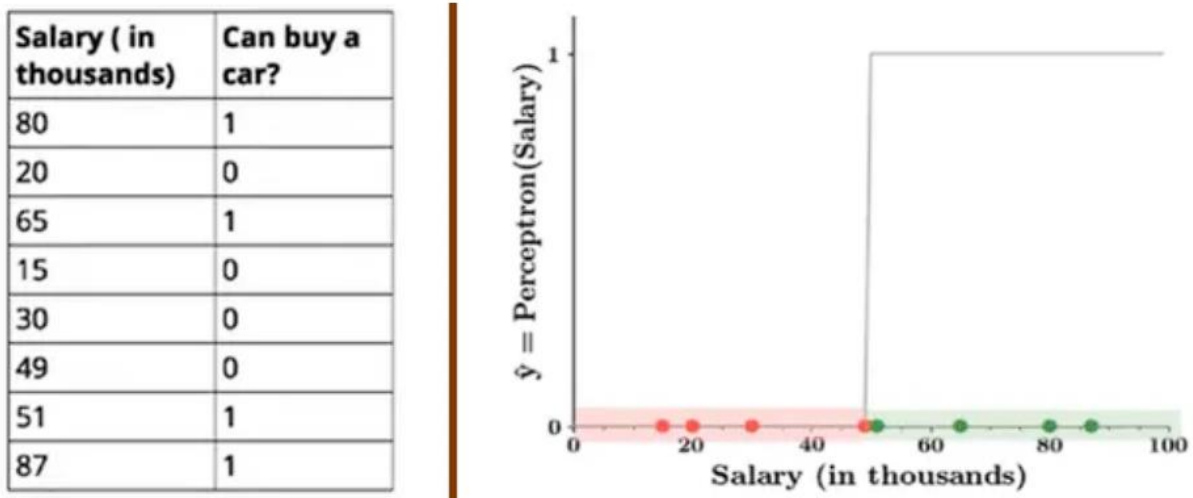


Рисунок 2.8 Вихідні дані моделі парцептрона

Червоні точки вказують на те, що людина не купила б машину, а зелені точки вказують на те, що людина хотіла б купити машину. Хіба не дивно, що людина з 50,1 тис. доларів купить машину, а хтось із 49,9 тис. доларів не купить машину? Невелика зміна вхідних даних перцептрона іноді може призвести до повного перемикання вихідних даних, скажімо, з 0 на 1. Така поведінка не є характеристикою конкретної обраної нами задачі або питомої ваги та обраного нами порогу. Це характеристика самого нейрона перцептрона, який поводить себе як крокова функція. Подолати цю проблему допомагає новий тип штучного нейрона, який називається сигмоподібним нейроном.

Сигмоїдний нейрон, де вихідна функція набагато плавніша, ніж у крокових функцій. У сигмоїдному нейроні невелика зміна вхідного сигналу викликає лише невелику зміну вихідного сигналу, на відміну від ступеневого вихідного сигналу. Існує багато функцій з S-подібною кривою, відомих як сигмоїдні функції. Найчастіше використовуваною функцією є логістична функція.

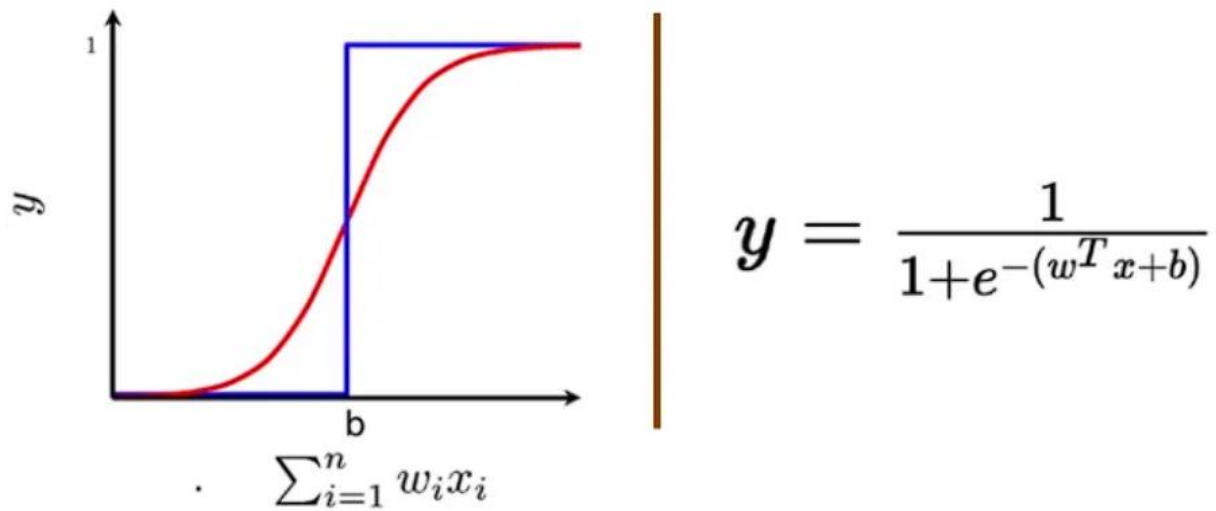


Рисунок 2.9 Сигмоїдний нейрон.

Більше немає різкого переходу на порозі b . Вихідний сигнал сигмоподібного нейрона не дорівнює 0 або 1. Натомість це дійсне значення між 0 і 1, яке можна інтерпретувати як ймовірність.

Входи сигмоїдного нейрона можуть бути дійсними числами, на відміну від логічних входів у MP Neuron, а вихід також буде дійсним числом між 0 та 1. У сигмоїдному нейроні треба регресувати зв'язок між X та Y з погляду ймовірності. Навіть якщо вихід знаходиться в діапазоні від 0 до 1, то все ще можна використовувати сигмоїдну функцію для завдань бінарної класифікації, вибравши певний поріг.

Алгоритм навчання параметрів w та b моделі сигмоподібного нейрона за допомогою алгоритму градієнтного спуску.

Find w and b such that:

$$\underset{w, b}{\text{minimize}} \mathcal{L}(w, b) = \sum_{i=1}^N (y_i - f(x_i))^2$$

Рисунок 2.10 Алгоритму градієнтного спуску

Мета алгоритму навчання полягає у визначенні найкращих можливих значень параметрів, щоб загальні втрати (квадратичні втрати помилки) моделі були максимально мінімізовані. Ось алгоритм навчання:

Initialise w, b

Iterate over data:

compute $\hat{y}$

compute $\mathcal{L}(w, b)$

$$w_{t+1} = w_t - \eta \Delta w_t$$

$$b_{t+1} = b_t - \eta \Delta b_t$$

till satisfied

Рисунок 2.11 Алгоритм навчання

Випадковим чином ініціалізуємо w та b . Потім перебираємо всі спостереження в даних, для кожного спостереження знаходимо відповідний прогнозований результат за допомогою сигмоподібної функції та обчислюємо квадрат втрат помилки. На основі значення втрат оновлюємо ваги таким чином, щоб загальні втрати моделі за нових параметрів були меншими за поточні втрати моделі.

$$\mathcal{L}(w, b) > \mathcal{L}(w + \eta \Delta w, b + \eta \Delta b)$$

Рисунок 2.11 Алгоритм навчання

Продовжуємо виконувати операцію оновлення, доки не будемо задоволені. «До задоволення» може означати будь-що з наступного: загальні втрати моделі стають нульовими. Загальні втрати моделі стають дуже малими, ближчими до нуля. Ітерація для фіксованої кількості проходів залежно від обчислювальної потужності.

2.2 Архітектури багат шарових нейронних мереж

Штучна нейронна мережа (ШНМ) – це модель машинного навчання, натхненна структурою та функцією взаємопов'язаної мережі нейронів людського мозку. Вона складається з взаємопов'язаних вузлів, які називаються штучними нейронами, організованих у шари. Інформація протікає через мережу, причому кожен нейрон обробляє вхідні сигнали та створює вихідний сигнал, який впливає на інші нейрони в мережі.

Багат шаровий перцептрон (БПМ) – це тип штучної нейронної мережі, що складається з кількох шарів нейронів. Нейрони в БПМ зазвичай використовують нелінійні функції активації, що дозволяє мережі вивчати складні закономірності в даних. БПМ є важливими в машинному навчанні, оскільки вони можуть вивчати нелінійні зв'язки в даних, що робить їх потужними моделями для таких завдань, як класифікація, регресія та розпізнавання образів.

Багат шаровий перцептрон: Багат шаровий перцептрон – це тип нейронної мережі прямого зв'язку, що складається з повністю пов'язаних нейронів з нелінійним типом функції активації. Він широко використовується для розрізнення даних, які не можна розділити лінійно. Багат шарові перцептрони широко використовуються в різних галузях, включаючи розпізнавання зображень, обробку природної мови та розпізнавання мовлення, серед інших. Їхня гнучкість в архітектурі та здатність апроксимувати будь-яку функцію за певних умов роблять їх фундаментальним будівельним блоком у глибокому навчанні та дослідженнях нейронних мереж. Деякі з його ключових концепцій такі:

Вхідний шар складається з вузлів або нейронів, які отримують початкові вхідні дані. Кожен нейрон представляє ознаку або вимір вхідних даних. Кількість нейронів у вхідному шарі визначається розмірністю вхідних даних.

У прихований шарів між вхідним та вихідним шарами може бути один або кілька шарів нейронів. Кожен нейрон у прихованому шарі отримує вхідні дані від усіх нейронів попереднього шару (або вхідного шару, або іншого прихованого шару) та створює вихідні дані, які передаються наступному шару. Кількість прихованих шарів та кількість нейронів у кожному прихованому шарі – це гіперпараметри, які необхідно визначити на етапі проєктування моделі.

Вихідний шар складається з нейронів, які створюють кінцевий вихід мережі. Кількість нейронів у вихідному шарі залежить від характеру завдання. У бінарній класифікації може бути один або два нейрони залежно від функції активації та представляє ймовірність належності до одного класу; тоді як у завданнях класифікації з кількома класами у вихідному шарі може бути кілька нейронів.

Ваги: Нейрони в сусідніх шарах повністю пов'язані один з одним. Кожен зв'язок має відповідну вагу, яка визначає силу зв'язку. Ці ваги вивчаються під час процесу навчання.

Нейрони зміщення: На додаток до вхідних та прихованих нейронів, кожен шар (крім вхідного шару) зазвичай містить нейрон зміщення, який забезпечує постійний вхідний сигнал для нейронів у наступному шарі. Нейрон зміщення має власну вагу, пов'язану з кожним зв'язком, яка також вивчається під час навчання. Нейрон зміщення, ефективно зміщує функцію активації нейронів у наступному шарі, дозволяючи мережі вивчати зміщення або зміщення на межі прийняття рішення. Регулюючи ваги, пов'язані з нейроном зміщення, MLP може навчитися контролювати поріг активації та краще відповідати навчальним даним.

Функція активації: Як правило, кожен нейрон у прихованих шарах та вихідному шарі застосовує функцію активації до своєї зваженої суми вхідних даних. Поширені функції активації включають сигмоподібну, тангенціальну, ReLU (випрямлену лінійну одиницю) та softmax. Ці функції вносять нелінійність у мережу, дозволяючи їй вивчати складні закономірності в даних.

Навчання за допомогою зворотного поширення: MLP навчаються за допомогою алгоритму зворотного поширення, який обчислює градієнти функції втрат відносно параметрів моделі та ітеративно оновлює параметри, щоб мінімізувати втрати.

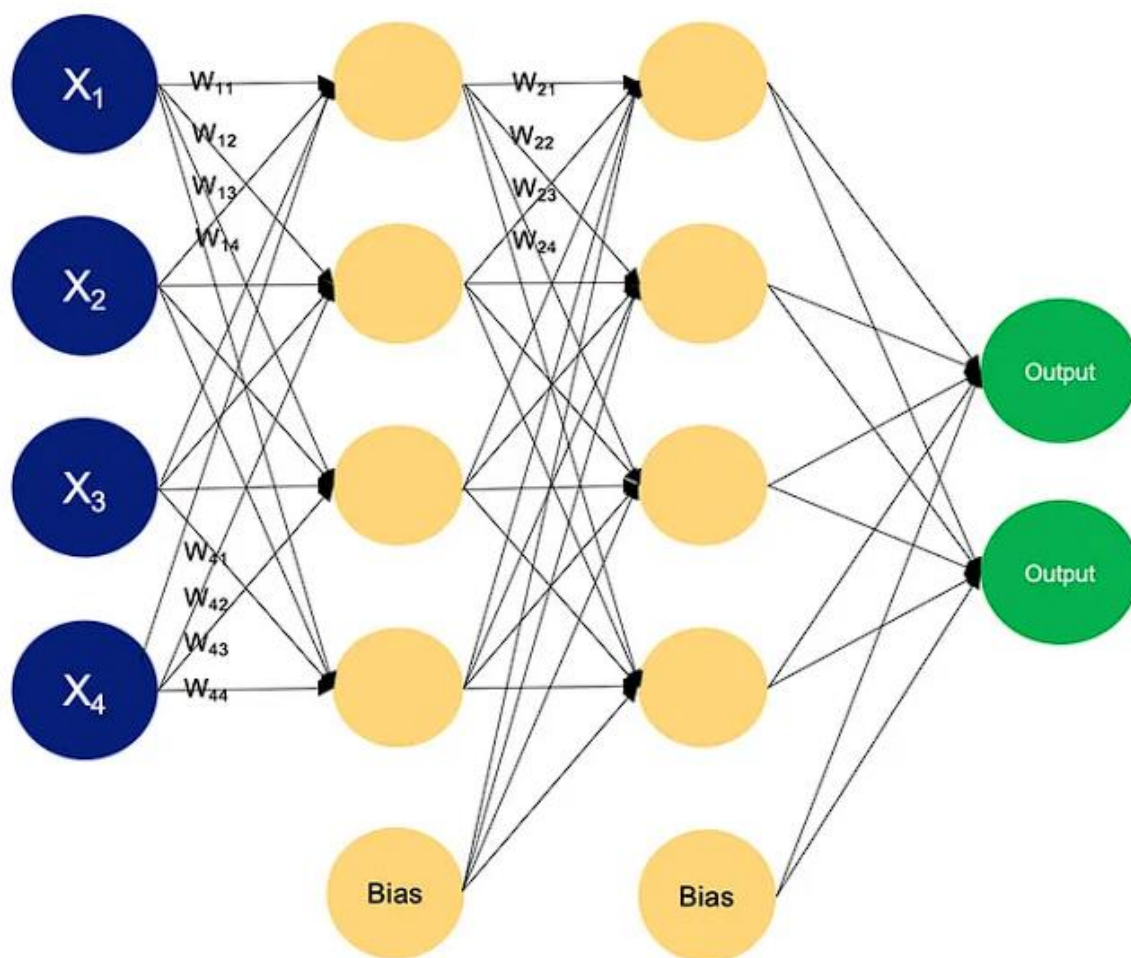


Рисунок 2.13 Алгоритм роботи з багатошаровим перцептроном.

У багатошаровому перцептроні нейрони обробляють інформацію крок за кроком, виконуючи обчислення, що включають зважені суми та нелінійні перетворення. Проходимо шар за шаром, щоб побачити магію, яка прихована всередині.

Градiєнтний спуск – це алгоритм оптимізації, який використовується для мінімізації функції шляхом коригування її параметрів. У контексті нейронних мереж він допомагає знаходити значення вагових коефіцієнтів, які мінімізують

функцію втрат. Цей процес допомагає нам виміряти, наскільки добре прогнози мережі відповідають фактичним даним.

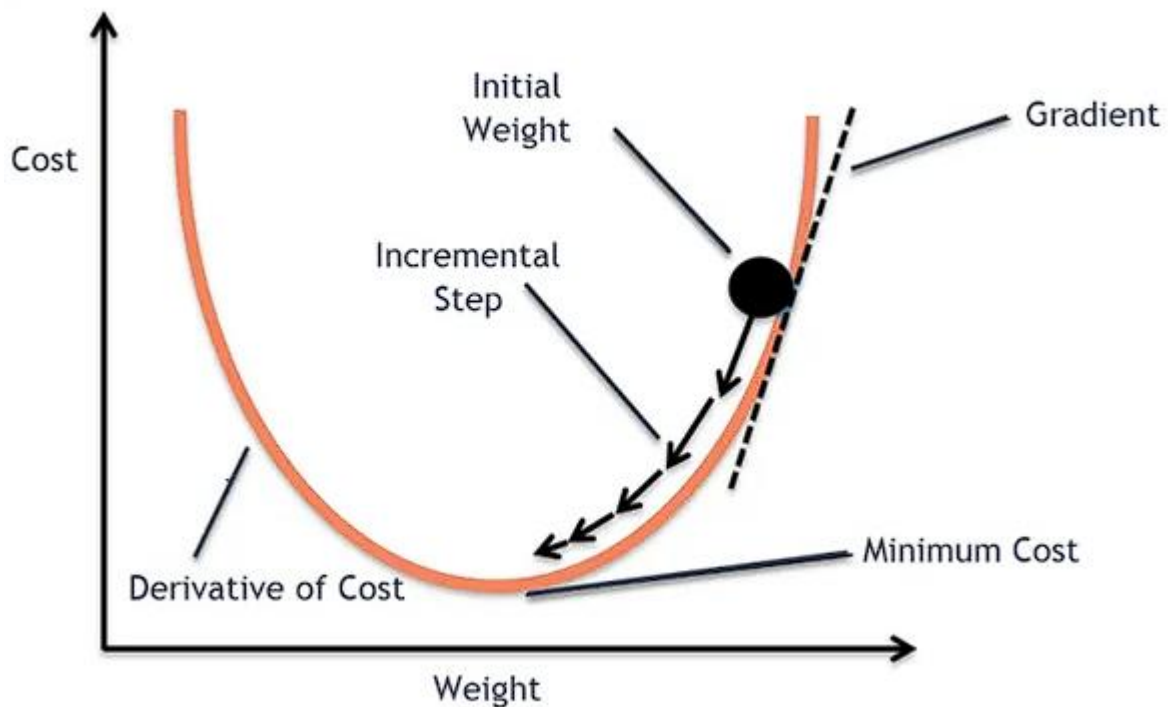


Рисунок 2.14 Алгоритм оптимізації градієнтного спуску

3 основні компоненти процесу градієнтного спуску:

Функція втрат

Градiєнт функції втрат

Швидкість навчання

Функція втрат

Функція втрат також відома як функція вартості або цільова функція.

Це математична функція, яка вимірює, наскільки добре прогнози моделі відповідають фактичним даним.

Вона кількісно визначає похибку між прогнозованими вихідними даними нейронної мережі та фактичними вихідними даними.

Метою будь-якого алгоритму машинного та глибокого навчання є мінімізація цієї функції втрат, тим самим підвищуючи точність прогнозів моделі.

Типові приклади включають середньоквадратичні помилки (MSE) для регресії та перехресні ентропійні втрати для класифікації.

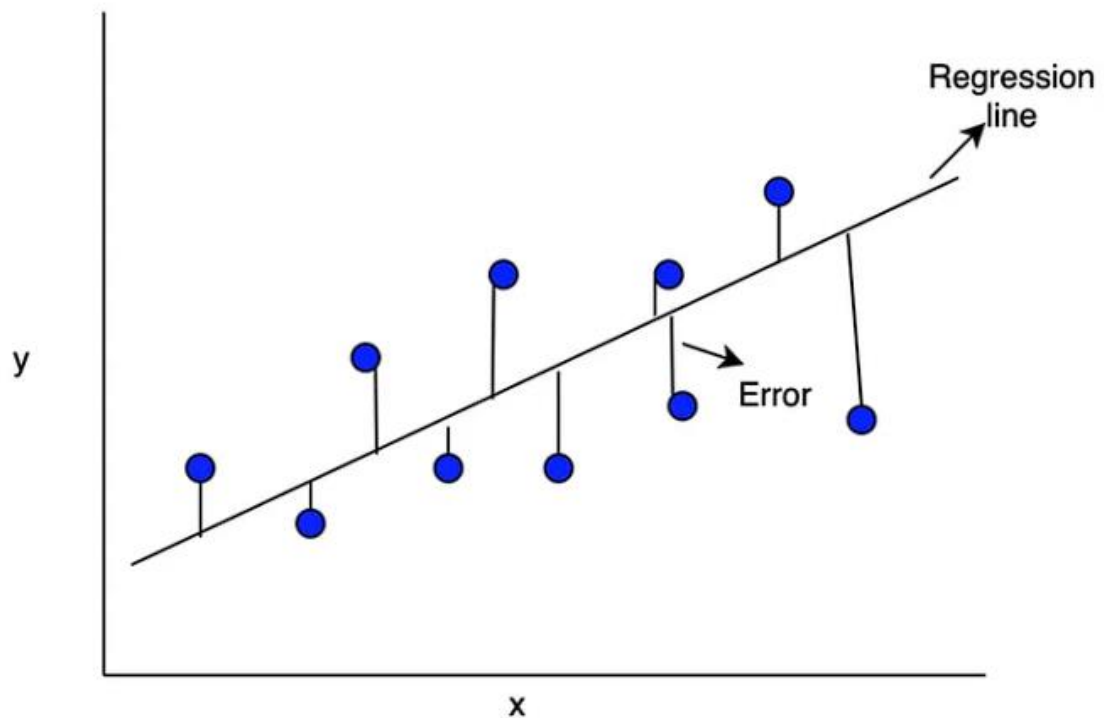


Рисунок 2.15 Графік середньоквадратичну помилки у MSE

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$$

Рисунок 2.16 MSE та градієнтний спуск

Градiєнт функції втрат відносно ваг вказує нам напрямок та швидкість зміни втрат. Він вказує напрямок найбільшого зростання функції втрат. Щоб мінімізувати втрати, потрібно рухатися в протилежному напрямку градієнта.

Він розраховується шляхом обчислення частинної похідної (нахилу функції в точці) функції втрат відносно кожного параметра в моделі (в першу чергу ваг та зміщення).

Зворотне поширення помилки – це метод, що використовується в глибокому навчанні для навчання штучних нейронних мереж, зокрема мереж прямого зв'язку. Він працює ітеративно, щоб налаштувати ваги та зміщення для мінімізації функції вартості.

На кожній епосі модель адаптує ці параметри, зменшуючи втрати, дотримуючись градієнта помилки. Зворотне поширення помилки часто використовує алгоритми оптимізації, такі як градієнтний спуск або стохастичний градієнтний спуск. Алгоритм обчислює градієнт за допомогою ланцюгового правила з математичного аналізу, що дозволяє йому ефективно переміщатися по складних шарах нейронної мережі для мінімізації функції вартості.

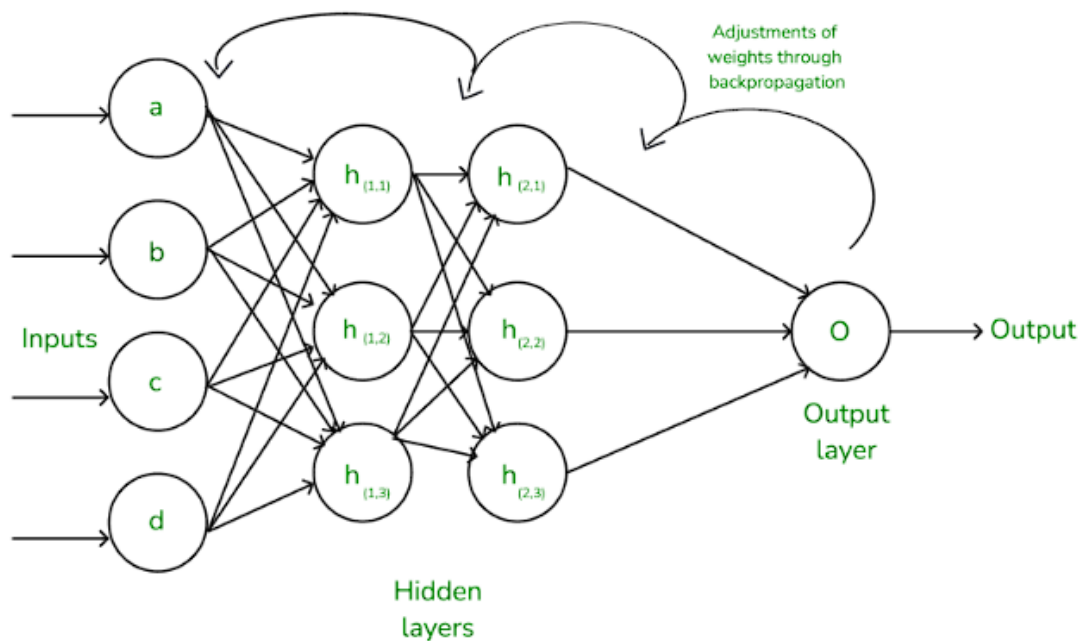


Рисунок 2.17 Зворотне поширення в нейронних мережах

Зворотне поширення відіграє вирішальну роль у тому, як нейронні мережі вдосконалюються з часом. Ось чому:

Ефективне оновлення ваг: Воно обчислює градієнт функції втрат відносно кожної ваги, використовуючи правило ланцюга, що дозволяє ефективно оновлювати ваги.

Масштабованість: Алгоритм зворотного поширення добре масштабується для мереж з кількома шарами та складними архітектурами, що робить глибоке навчання можливим.

Автоматизоване навчання: Завдяки зворотному поширенню процес навчання стає автоматизованим, і модель може налаштовуватися для оптимізації своєї продуктивності. Робота алгоритму зворотного поширення

Алгоритм зворотного поширення включає два основні кроки: прямий прохід та зворотний прохід. Як працює прямий прохід?

При прямому проході вхідні дані подаються у вхідний шар. Ці вхідні дані в поєднанні з їхніми відповідними вагами передаються на приховані шари. Наприклад, у мережі з двома прихованими шарами (h_1 та h_2 , як показано на рис. (a)) вихід з h_1 служить вхідними даними для h_2 . Перед застосуванням функції активації до зважених вхідних даних додається зміщення. Кожен прихований шар, обчислює зважену суму (а) вхідних даних, а потім застосовує функцію активації, таку як ReLU (випрямлена лінійна одиниця), для отримання вихідного значення (о). Вихідні дані передаються на наступний шар, де функція активації, така як softmax, перетворює зважені вихідні дані на ймовірності для класифікації.

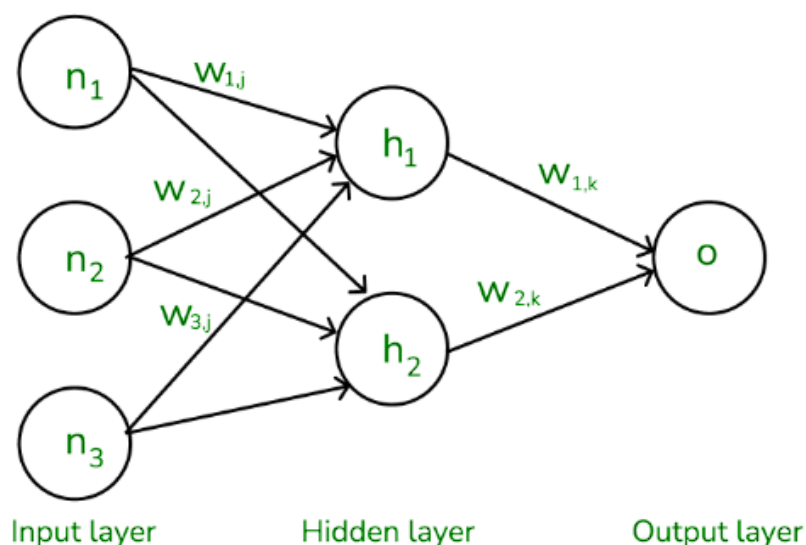


Рисунок 2.18 Взаємодії між нейронними шарами

2.3 Приклади використання штучних нейронних мереж в бізнесі

Google відома, як найвидатніша та найдосконаліша компанія в галузі штучного інтелекту та машинного навчання. Основною причиною цього, ймовірно, є сума грошей, яку компанія витратила на придбання стартапів. За даними CB Insights, Google витратив більше, ніж будь-хто інший.

Google повністю використовує переваги нейронних мереж, глибокого навчання та штучного інтелекту. Google запустив кілька чат-ботів на основі штучного інтелекту, які відповідають на повідомлення за вас, наприклад, у вигляді складнішого автоматичного електронного листа з відповідями в різних контекстах, включаючи Skype, Slack та прямі повідомлення Twitter.

Але найсильнішою стороною Google у цій галузі, ймовірно, є спектр хмарних послуг, які вона пропонує розробникам, включаючи інструменти машинного навчання Google Cloud AI.

Baidu Ще однією компанією, яка була дуже активною на ринку, є китайський пошуковий гігант Baidu. Компанія особливо зацікавлена в обробці природної мови з метою розробки працездатної пошукової системи з голосовою активацією.

Одним із багатьох придбань Baidu в галузі машинного навчання стала Kitt.ai, яка, як кажуть, має портфоліо чат-ботів та голосових програм. Фінансові умови цієї конкретної угоди не розголошуються, але, за даними CB Insights, Baidu вважається десятим найбільшим інвестором у придбання у світ

Apple може існувати думка, що Apple запізнилася з машинним навчанням, але це, ймовірно, неправда, особливо враховуючи, що саме вона першою запустила голосового помічника на смартфоні. Мільйони людей розмовляють із Siri, і Apple прагне розширити застосування цього розмовного помічника через свій новий пристрій для розумного дому або динамік HomePod. Apple також активно бере участь у придбаннях, поступаючись лише Google.

Однією з найбільш помітних покупок стала Lattice Data, яка має систему машинного навчання для перетворення неструктурованих даних, таких як випадковий текст і зображення, на структуровані дані.

Microsoft за даними CB Insights, Microsoft фактично була третьою за величиною інвестором у придбання за останні кілька років. Компанія добре і серйозно вийшла на інтернет-ринок, особливо після придбання LinkedIn за двадцять шість мільярдів доларів кілька років тому.

Але, ймовірно, найзначнішим придбанням, яке Microsoft здійснила у сфері машинного навчання, стала Maluuba, яка, за словами технологічного гіганта, має «одну з найвражаючих у світі дослідницьких лабораторій глибокого навчання для розуміння природної мови».

HubSpot Кожен, хто знайомий з HubSpot, ймовірно, вже знає, що компанія давно є одним з перших, хто використовує новітні технології, і компанія довела це ще раз, коли оголосила про придбання фірми машинного навчання Kemvi.

Прогнозний скоринг лідів — це лише одне з багатьох потенційних застосувань штучного інтелекту та машинного навчання. HubSpot планує використовувати технологію Kemvi в низці застосувань, зокрема, інтегруючи технологію машинного навчання DeepGraph та обробки природної мови від Kemvi у свою внутрішню систему управління контентом.

ЗАСТОСУВАННЯ НЕЙРОННОЇ МЕРЕЖІ В: ОЦІНЦІ ТА АНАЛІЗУ- БІЗНЕС ПРОЦЕСУ

3.1 Побудова схеми шарів нейронної мережі та методологія графічного моделювання процесів IDEF0.

Схема шарів нейронної мережі (Рисунок 3.1) дає нам наступне розуміння: Архітектура моделі Відображає структуру мережі, включаючи кількість шарів, їх тип (вхідний, прихований, вихідний) та послідовність зв'язків між ними. Глибина мережі Показує кількість прихованих шарів, що вказує на складність моделі (наприклад, "малі" vs "глибокі" мережі). Розмірність даних Відображає зміну розмірності даних при проходженні через шари (наприклад, зменшення розмірності у випадку згорткових шарів). Типи шарів Дозволяє ідентифікувати використовувані шари (наприклад, повнозв'язні, згорткові, рекурентні, пулінгові тощо). Потік інформації Візуалізує напрямок руху даних (feedforward, рекурентні зв'язки, skip-connections тощо). Параметризація Дає уявлення про кількість параметрів (ваг) у кожному шарі, що впливає на складність моделі. Специфіка обробки Показує спеціалізовані шари (наприклад, embedding шари для обробки тексту, batch normalization тощо). Така схема є важливим інструментом для аналізу, налагодження та пояснення роботи нейронної мережі.

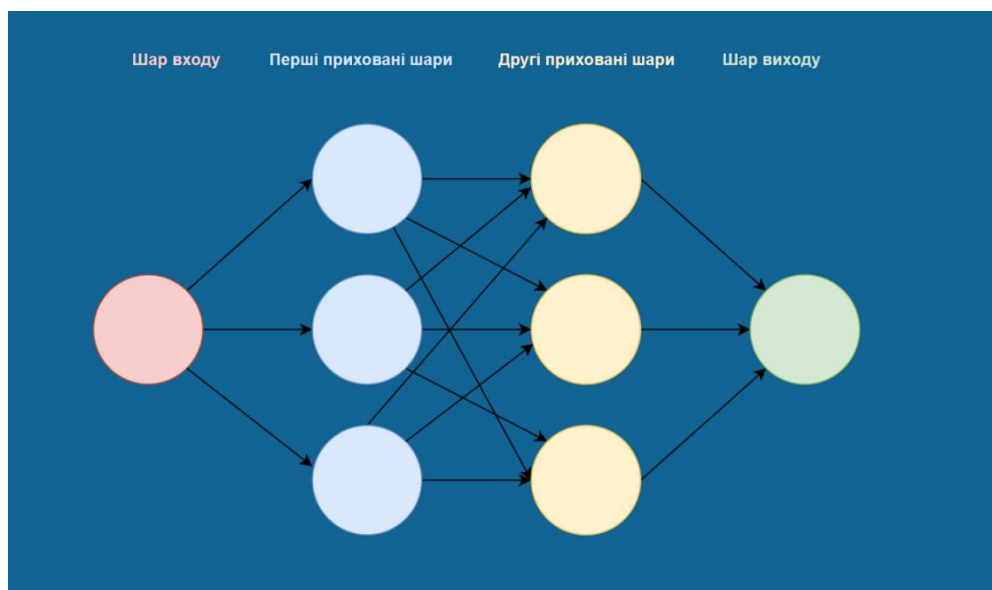


Рисунок 3.1 Схема шарів нейронної мережі.

IDEF0 — це потужний інструмент для структурованого аналізу, оптимізації та візуалізації будь-яких процесів. Він особливо корисний там, де важлива чіткість, контроль і документування діяльності. Як інструмент моделювання процесів дає наступні можливості:

Візуалізація бізнес-процесів Чітко відображає вхідні дані, вихідні результати, механізми (ресурси) та управлінські впливи для кожного процесу. Допомогає структурувати складні процеси у вигляді ієрархії діаграм (від загального до деталей).

Аналіз та оптимізація процесів Дозволяє виявити надлишкові операції, "вузькі місця" та неефективність. Дає змогу порівнювати "як є" (поточний стан) і "як має бути" (бажаний стан).

Стандартизація та документування Забезпечує уніфікований опис процесів для всіх учасників (від керівництва до виконавців). Створює чітку документацію для навчання персоналу або сертифікації (наприклад, за ISO).

Підтримка прийняття рішень Допомогає оцінити взаємозв'язки між підрозділами та наслідки змін. Використовується для планування впровадження ІТ-систем (наприклад, ERP, CRM).

Застосування в різних сферах Спочатку розроблялася для виробництва, але зараз застосовується в: ІТ (розробка ПЗ, системний аналіз), Логістиці, Фінансах, Державному управлінні.



Рисунок 3.2 Діаграма IDEF0 рівень A0.

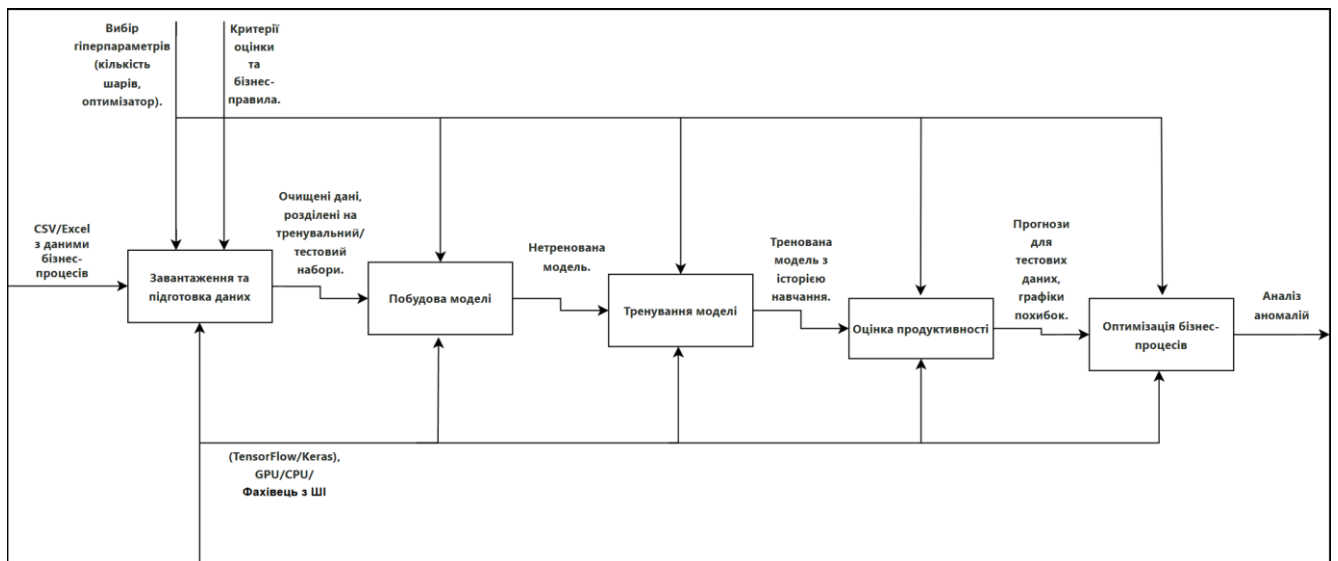


Рисунок 3.3 Діаграма IDEF0 A1-A5.

3.2 Ціль використання нейронної мережі в оцінці та оптимізації.

Суттю використання нейронної мережі у цьому бізнес процесі є оцінка та оптимізація ланцюгів постачання та управління запасами. У цій задачі регресії метою є прогнозування результату безперервного значення, такого як ціна або ймовірність. Порівняйте це із задачею класифікації, де метою є вибір класу зі списку класів. Використовуємо набір даних Auto MPG де демонструється, як створюються моделі для прогнозування та оцінки. Для цього надаємо моделям опис багатьох автомобілів того періоду. Цей опис включає такі атрибути, як кількість циліндрів, об'єм, потужність та вага.

3.3 Оцінка та оптимізація ланцюгів за допомогою нейронної мережі.

Імпортуємо бібліотеки в середу розробки.

```
!pip install -q seaborn
```

Рисунок 3.4 Інсталяція бібліотеки Seaborn.

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
```

Рисунок 3.5 Імпортуємо бібліотеки.

Робимо видрук NumPy легшими для читання.

```
np.set_printoptions(precision=3, suppress=True)
```

Рисунок 3.6 Робимо видрук легшими для читання.

Імпортуємо високорівневий API Keras із бібліотеки TensorFlow

Імпортуємо шари нейронних мереж із модуля keras.

Шари - це будівельні блоки нейронних мереж з яких складається LSTM модель

```
import tensorflow as tf

from tensorflow import keras
from tensorflow.keras import layers

print(tf.__version__)
```

Рисунок 3.7 Імпортуємо API Keras та Шари мережі із бібліотеки TensorFlow.

Набір даних Auto MPG доступний у репозиторії машинного навчання UCI.

Спочатку завантажуюмо та імпортуєте набір даних за допомогою pandas.

Набір даних містить кілька невідомих значень

```
url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data'
column_names = ['MPG', 'Cylinders', 'Displacement', 'Horsepower', 'Weight',
                 'Acceleration', 'Model Year', 'Origin']

raw_dataset = pd.read_csv(url, names=column_names,
                           na_values='?', comment='\t',
                           sep=' ', skipinitialspace=True)
```

Рисунок 3.8 Завантажуємо данні.

```
dataset = raw_dataset.copy()
dataset.tail()
```

Рисунок 3.9 Очистимо дані, бо набір даних містить кілька невідомих значень.

```
dataset.isna().sum()
```

Рисунок 3.10 Знаходимо дані які містять кілька невідомих значень.

```
dataset = dataset.dropna()
```

Рисунок 3.11 Видаляємо строки з Nan.

Стовпець « Origin» є категоріальним, а не числовим. Тому наступним кроком є одноразове кодування значень у стовпці за допомогою `pd.get_dummies`. Якщо не вказати тип даних за допомогою аргументу `dtype`, то отримаємо логічні значення, що призведе до помилок під час нормалізації при створенні екземпляра об'єкта `Tensor`, якщо значення ознак не будуть приведені до уніфікованого типу під час передачі масиву в `tf.keras.layers.Normalization.adapt()`. Об'єкти `Tensor` повинні містити уніфіковані типи даних.

```
dataset['Origin'] = dataset['Origin'].map({1: 'USA', 2: 'Europe', 3: 'Japan'})
```

Рисунок 3.12 Задаємо порядок країнам.

```
dataset = pd.get_dummies(dataset, columns=['Origin'], prefix='', prefix_sep='')
dataset.tail()
```

Рисунок 3.13 Підготовка даних до перетворення 'Origin' перед навчанням моделі.

Розділяємо дані на навчальний та тестовий набори. Де будемо використовувати тестовий набір у оцінці наших моделей.

```
train_dataset = dataset.sample(frac=0.8, random_state=0)
test_dataset = dataset.drop(train_dataset.index)
```

Рисунок 3.14 Тренування дата сету.

Переглянемо спільний розподіл кількох пар стовпців з навчального набору.

У верхньому рядку зазначено, що паливна економічність (MPG) є функцією всіх інших параметрів. Інші рядки вказують на те, що вони є функціями один одного.

```
sns.pairplot(train_dataset[['MPG', 'Cylinders', 'Displacement', 'Weight']], diag_kind='kde')
```

Рисунок 3.15 Візуалізація взаємозв'язків: MPG, циліндри, об'єм двигуна та вага.

Також перевіримо загальну статистику. Зверніть увагу, що кожна функція охоплює дуже різний діапазон:

```
train_dataset.describe().transpose()
```

Рисунок 3.16 Перевірка статистики.

Відокремлюємо ознаки від міток

Відокремлюємо цільове значення — «мітку» — від ознак. Ця мітка — це значення, яке ви будете навчати модель прогнозувати.

```
train_features = train_dataset.copy()
test_features = test_dataset.copy()

train_labels = train_features.pop('MPG')
test_labels = test_features.pop('MPG')
```

Рисунок 3.17 Навчання моделі прогнозування.

Нормалізація

У таблиці статистики легко побачити, наскільки відрізняються діапазони кожної ознаки:

```
train_dataset.describe().transpose()[['mean', 'std']]
```

Рисунок 3.18 Нормалізація.

Гарною практикою є нормалізація ознак, які використовують різні масштаби та діапазони.

Одна з причин важливості цього полягає в тому, що ознаки множаться на ваги моделі. Отже, масштаб вихідних даних та масштаб градієнтів залежать від масштабу вхідних даних.

Хоча модель може сходитися без нормалізації ознак, нормалізація робить навчання набагато стабільнішим.

Примітка: Нормалізація окремих ознак не має переваг — це зроблено тут для простоти.

Шар нормалізації `tf.keras.layers.Normalization` — це простий та зрозумілий спосіб додати нормалізацію об'єктів до нашої моделі.

Першим кроком є створення шару:

```
normalizer = tf.keras.layers.Normalization(axis=-1)
```

Рисунок 3.19 Створення шару нормалізації.

Потім, підігнавши стан шару попередньої обробки до даних, викликавши функцію

```
normalizer.adapt(np.array(train_features))
```

Рисунок 3.20 Обчислення середнього значення дисперсії.

Обчисліть середнє значення та дисперсію і збережемо їх у шарі:

```
print(normalizer.mean.numpy())
```

Рисунок 3.21 Зберігання інформації у шарі.

Коли викликається шар, він повертає вхідні дані, причому кожна ознака незалежно нормалізована:

```

first = np.array(train_features[:1])

with np.printoptions(precision=2, suppress=True):
    print('First example:', first)
    print()
    print('Normalized:', normalizer(first).numpy())

```

Рисунок 3.22 Нормалізація та вивід першого прикладу з тренувальних даних.

Лінійна регресія

Перш ніж створювати модель глибокої нейронної мережі, почніть з лінійної регресії, використовуючи одну або кілька змінних.

Лінійна регресія з однією змінною. Почнемо з лінійної регресії з однією змінною, щоб передбачити «MPG» на основі «Horsepower». Навчання моделі за допомогою `tf.keras` зазвичай починається з визначення архітектури моделі. Використовуємо ознаки модель `tf.keras.Sequential`, яка представляє послідовність кроків. У нашій моделі лінійної регресії з однією змінною є два кроки: Нормалізуйте вхідні характеристики «Horsepower» за допомогою шару попередньої обробки `tf.keras.layers.Normalization`. Застосовуємо лінійне перетворення ($y=mx+b$) для отримання першого виходу за допомогою лінійного шару (`tf.keras.layers.Dense`). Кількість вхідних даних може бути встановлена аргументом `input_shape` або автоматично під час першого запуску моделі. Спочатку створіть масив NumPy, що складається з характеристик «Horsepower». Потім створіть екземпляр `tf.keras.layers.Normalization` та підберіть його стан до даних про кінські сили:

```

horsepower = np.array(train_features['Horsepower'])

horsepower_normalizer = layers.Normalization(input_shape=[1,], axis=None)
horsepower_normalizer.adapt(horsepower)

```

Рисунок 3.23 Лінійна регресія.

Побудуємо послідовну модель Keras:

```
horsepower_model = tf.keras.Sequential([
    horsepower_normalizer,
    layers.Dense(units=1)
])

horsepower_model.summary()
```

Рисунок 3.24 Будуємо послідовну модель Keras.

Ця модель передбачить «MPG» на основі «Horsepower».

Запустіть ненавчену модель на перших десяти значеннях «Horsepower». Результат буде невдалим, але зверніть увагу, що він має очікувану форму (10, 1):

```
horsepower_model.predict(horsepower[:10])
```

Рисунок 3.25 Запуск ненавченої моделі.

Після побудови моделі налаштуйте процедуру навчання за допомогою методу Keras `Model.compile`. Найважливішими аргументами для компіляції є втрати та оптимізатор, оскільки вони визначають, що буде оптимізовано (`mean_absolute_error`) та як (за допомогою `tf.keras.optimizers.Adam`).

```
horsepower_model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.1),
    loss='mean_absolute_error')
```

Рисунок 3.26 Процедура навчання за допомогою методу Keras `Model.compile`.

Використовуємо Keras `Model.fit` для виконання навчання протягом 100 епох:

```
%%time
history = horsepower_model.fit(
    train_features['Horsepower'],
    train_labels,
    epochs=100,
    verbose=0,
    validation_split = 0.2)
```

Рисунок 3.27 Навчання протягом 100 епох

Візуалізуємо прогрес навчання моделі, використовуючи статистику, що зберігається в об'єкті історії:

```
hist = pd.DataFrame(history.history)
hist['epoch'] = history.epoch
hist.tail()
```

Рисунок 3.28 Візуалізуємо прогрес навчання моделі.

```
def plot_loss(history):
    plt.plot(history.history['loss'], label='loss')
    plt.plot(history.history['val_loss'], label='val_loss')
    plt.ylim([0, 10])
    plt.xlabel('Epoch')
    plt.ylabel('Error [MPG]')
    plt.legend()
    plt.grid(True)
    plot_loss(history)
```

Рисунок 3.29 Візуалізуємо прогрес навчання моделі.

Зберемо результати тестового набору для подальшого використання:

```
test_results = {}

test_results['horsepower_model'] = horsepower_model.evaluate(
    test_features['Horsepower'],
    test_labels, verbose=0)
```

Рисунок 3.30 Зберігання результатів тестового набору.

Оскільки це регресія з однією змінною, легко розглядати прогнози моделі як функцію вхідних даних:

```
x = tf.linspace(0.0, 250, 251)
y = horsepower_model.predict(x)
def plot_horsepower(x, y):
    plt.scatter(train_features['Horsepower'], train_labels, label='Data')
    plt.plot(x, y, color='k', label='Predictions')
    plt.xlabel('Horsepower')
    plt.ylabel('MPG')
    plt.legend()
plot_horsepower(x, y)|
```

Рисунок 3.31 Прогнози моделі як функція вхідних даних.

Лінійна регресія з множинними вхідними даними

Можемо використовувати майже ідентичний підхід для прогнозування на основі кількох вхідних даних. Ця модель досі використовує ту саму формулу $y=mx+b$, але з такими особливостями:

Знову створюємо двоетапну послідовну модель Keras, де першим шаром буде нормалізатор (`tf.keras.layers.Normalization(axis=-1)`), який ви визначили раніше та адаптували до всього набору даних:

```
linear_model = tf.keras.Sequential([
    normalizer,
    layers.Dense(units=1)
])
```

Рисунок 3.32 Створення двоетапної послідовної моделі.

Коли викликаємо `Model.predict` для пакета вхідних даних, він створює вихідні дані `units=1` для кожного прикладу

```
linear_model.predict(train_features[:10])
```

Рисунок 3.33 Створення вихідних даних `units=1` для кожного прикладу.

Коли викликаємо модель, будуть побудовані її вагові матриці — перевіримо, чи ваги ядра (m в $y=mx+b$) мають форму (9, 1):

```
linear_model.layers[1].kernel
```

Рисунок 3.34 Отримує доступ до ваги шару нейромережі.

Налаштуємо модель за допомогою Keras `Model.compile` та навчіть її за допомогою `Model.fit` протягом 100 епох:

```
linear_model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.1),
    loss='mean_absolute_error')
```

Рисунок 3.35 Налаштуємо модель протягом 100 епох.

```
%%time  
history = linear_model.fit(  
    train_features,  
    train_labels,  
    epochs=100,  
    verbose=0,  
    validation_split = 0.2)
```

Рисунок 3.36 Криві навчання лінійної регресійної моделі.

Використання всіх вхідні дані у цій регресійній моделі досягає значно меншої помилки навчання та перевірки, ніж `horsepower_model`, яка мала один вхідний дані:

```
plot_loss(history)
```

Рисунок 3.37 Побудова графіка.

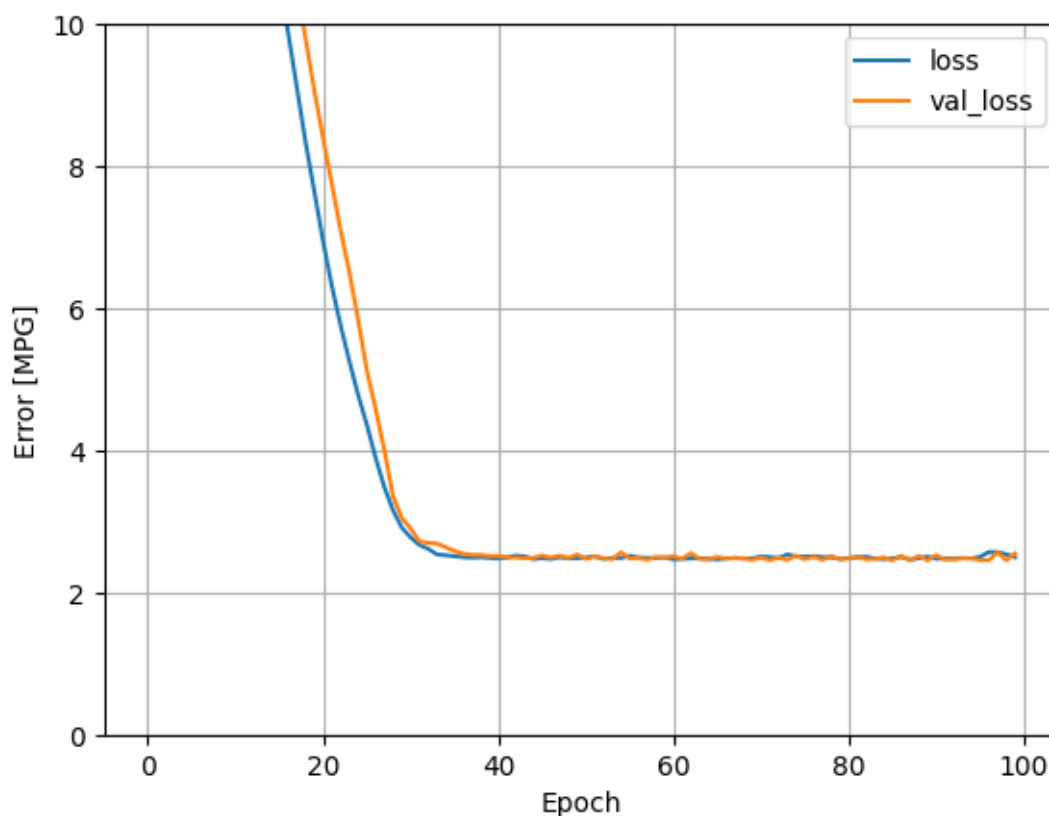


Рисунок 3.38 Графік результату.

Зберемо результати тестового набору для подальшого використання:

```
test_results['linear_model'] = linear_model.evaluate(
    test_features, test_labels, verbose=0)
```

Рисунок 3.39 Зберемо результати тестового набору.

Регресія з глибокою нейронною мережею (DNN)

Тут реалізуємо моделі DNN з одним входом та кількома входами.

Код в основному однаковий, за винятком того, що модель розширена, щоб включити деякі "приховані" нелінійні шари. Назва "прихований" тут просто означає, що не пов'язаний безпосередньо з входами або виходами.

Ці моделі міститимуть кілька шарів більше, ніж лінійна модель:

Шар нормалізації, як і раніше (з `horsepower_normalizer` для моделі з одним входом та нормалізатором для моделі з кількома входами).

Два приховані, нелінійні, щільні шари з нелінійністю функції активації ReLU (relu).

Лінійний щільний шар з одним виходом. Обидві моделі використовуватимуть однакову процедуру навчання, тому метод компіляції включено до функції

```
def build_and_compile_model(norm):
    model = keras.Sequential([
        norm,
        layers.Dense(64, activation='relu'),
        layers.Dense(64, activation='relu'),
        layers.Dense(1)
    ])

    model.compile(loss='mean_absolute_error',
                  optimizer=tf.keras.optimizers.Adam(0.001))
    return model
```

Рисунок 3.40 Регресія з глибокою нейронною мережею (DNN).

Регресія з використанням DNN та одного вхідного значення

Модель DNN лише з «horsepower» як вхідним значенням та horsepower_normalizer (визначеним раніше) як шаром нормалізації:

```
dnn_horsepower_model = build_and_compile_model(horsepower_normalizer)
```

Рисунок 3.41 Модель DNN лише з «horsepower».

Ця модель має значно більше параметрів для навчання, ніж лінійні моделі:

```
dnn_horsepower_model.summary()
```

Рисунок 3.42 Побудова графіка.

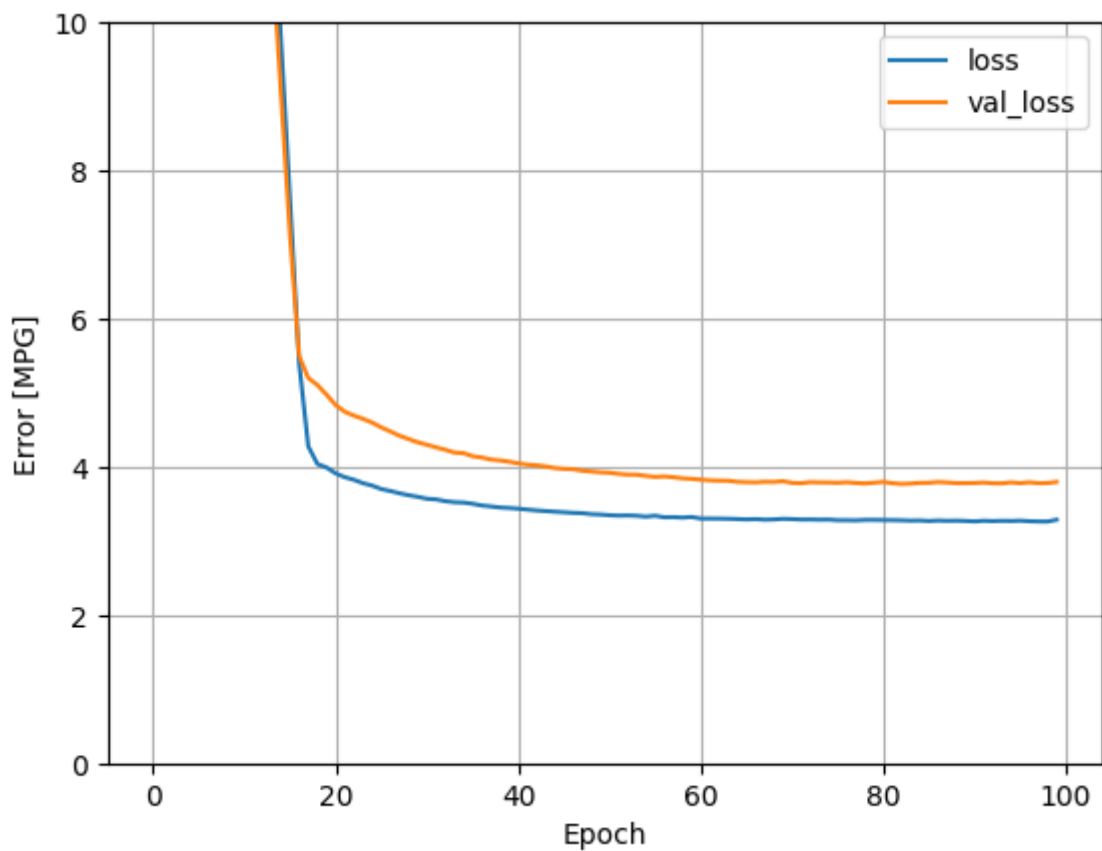


Рисунок 3.43 Графік результату.

Навчіть модель за допомогою Keras Model.fit:

```
%%time
history = dnn_horsepower_model.fit(
    train_features['Horsepower'],
    train_labels,
    validation_split=0.2,
    verbose=0, epochs=100)
```

Рисунок 3.44 Навчаємо модель за допомогою.

Ця модель працює трохи краще, ніж лінійна одно вхідна horsepower_model:

```
plot_loss(history)
```

Рисунок 3.45 Побудування графіку.

Якщо побудуємо прогнози як функцію від «Кінських сил», то ми повинні помітити, як ця модель використовує нелінійність, що забезпечується прихованими шарами:

```
x = tf.linspace(0.0, 250, 251)
y = dnn_horsepower_model.predict(x)
plot_horsepower(x, y)
```

Рисунок 3.46 Застосовуємо модель для нових даних та будуємо графік.

Зберіть результати тестового набору для подальшого використання:

```
test_results['dnn_horsepower_model'] = dnn_horsepower_model.evaluate(
    test_features['Horsepower'], test_labels,
    verbose=0)
```

Рисунок 3.47 Зберігаємо результати тестового набору

Регресія з використанням глибокої нейронної мережі (DNN) та кількох вхідних даних.

Повторимо попередній процес, використовуючи всі вхідні дані. Продуктивність моделі дещо покращується на наборі даних для перевірки.

```
dnn_model = build_and_compile_model(normalizer)
dnn_model.summary()
%%time
history = dnn_model.fit(
    train_features,
    train_labels,
    validation_split=0.2,
    verbose=0, epochs=100)
```

Рисунок 3.48 Використанням глибокої нейронної мережі.

Зберіть результати на тестовому наборі:

```
test_results['dnn_model'] = dnn_model.evaluate(test_features, test_labels, verbose=0)
```

Рисунок 3.49 Результати на тестовому наборі.

Продуктивність

Оскільки всі моделі навчені, можемо переглянути продуктивність їхнього тестового набору:

```
pd.DataFrame(test_results, index=['Mean absolute error [MPG]']).T
```

Рисунок 3.50 Перегляд продуктивності тестового набору.

Таблиця результати тестування моделі регресії 3.1

№	Назва моделі	Середня абсолютна похибка
1	horsepower_model	3.651240110397339
2	linear_mode	2.432298421859741
3	dnn_horsepower_model	2.956965208053589
4	dnn_model	1.7068039178848267

Ці результати відповідають похибці валідації, що спостерігалася під час навчання.

Тепер можемо робити прогнози за допомогою `dnn_model` на тестовому наборі, використовуючи `Keras Model.predict`, та переглянути втрати:

```
test_predictions = dnn_model.predict(test_features).flatten()

a = plt.axes(aspect='equal')
plt.scatter(test_labels, test_predictions)
plt.xlabel('True Values [MPG]')
plt.ylabel('Predictions [MPG]')
lims = [0, 50]
plt.xlim(lims)
plt.ylim(lims)
_ = plt.plot(lims, lims)
```

Рисунок 3.51 Прогнозування за допомогою DNN моделі.

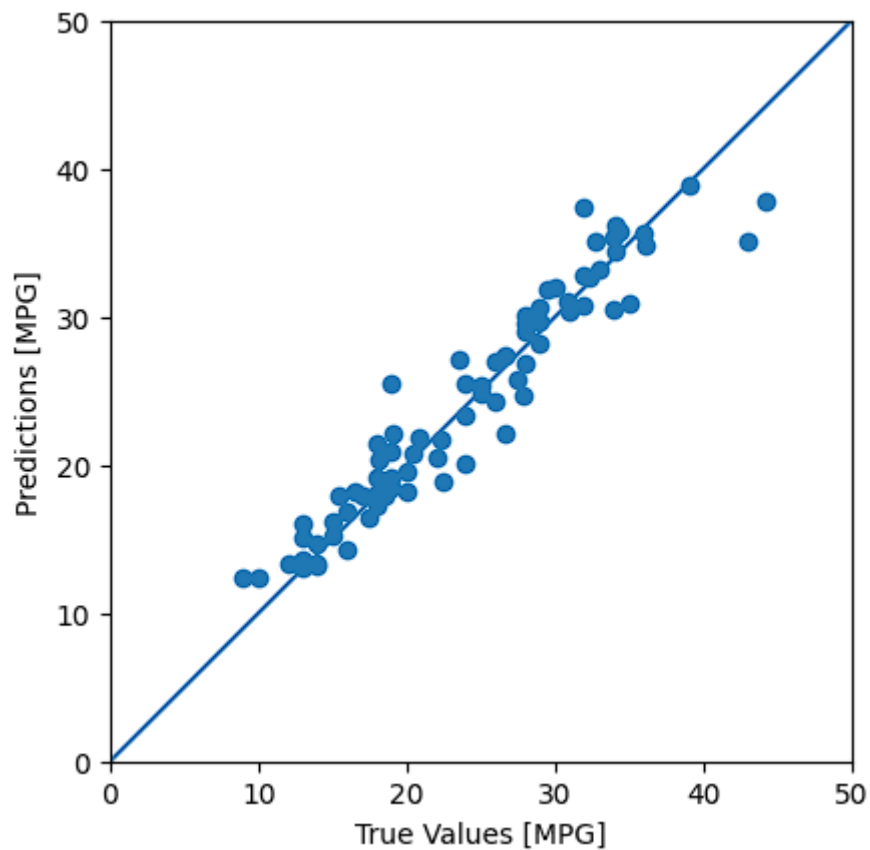


Рисунок 3.52 Справжні цінності проти прогнозів.

Тепер перевіримо розподіл помилок:

```
error = test_predictions - test_labels
plt.hist(error, bins=25)
plt.xlabel('Prediction Error [MPG]')
_ = plt.ylabel('Count')
```

Рисунок 3.53 Перевірка розподілу помилок.

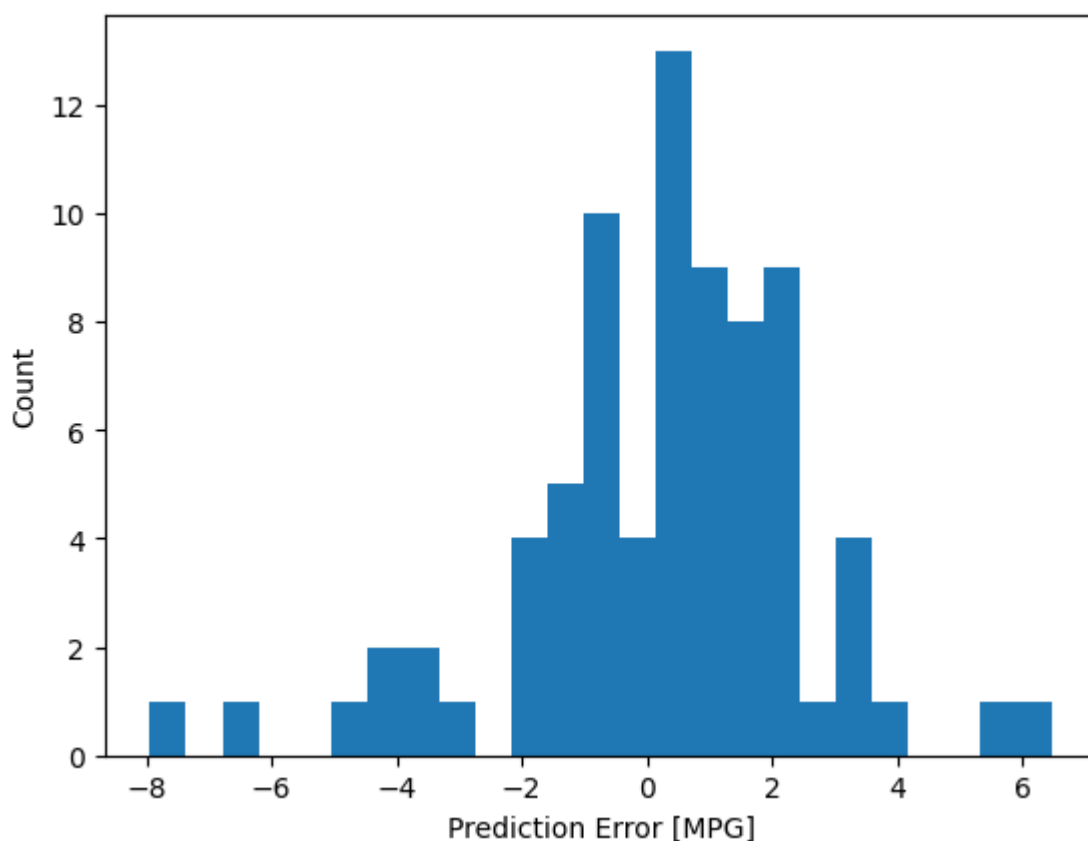


Рисунок 3.54 Помилки прогнозування.

Збережемо її для подальшого використання за допомогою Model.save:

```
dnn_model.save('dnn_model.keras')
```

Рисунок 3.55 Зберігання її для подальшого використання.

Якщо перезавантажити модель, вона видасть ідентичний результат:

```
reloaded = tf.keras.models.load_model('dnn_model.keras')

test_results['reloaded'] = reloaded.evaluate(
    test_features, test_labels, verbose=0)
pd.DataFrame(test_results, index=['Mean absolute error [MPG]']).T
```

Рисунок 3.56 Перезавантаження моделі.

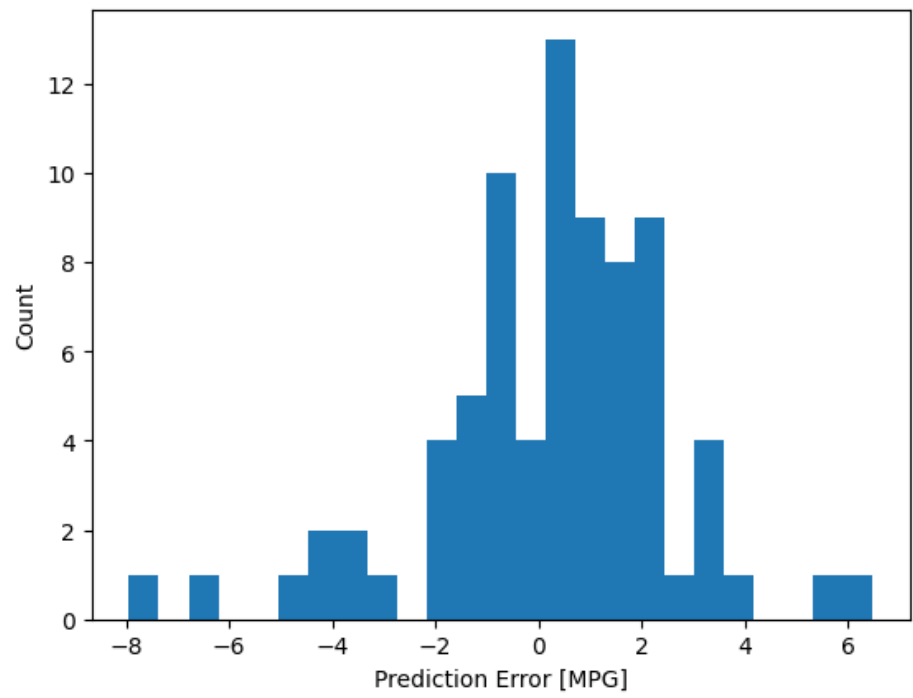


Рисунок 3.57 Перезавантажені помилки прогнозування.

Таблиця повторного проходження тестування моделі регресії.3.2

№	Назва моделі	Середня абсолютна похибка
1	horsepower_model	3.651240110397339
2	linear_mode	2.432298421859741
3	dnn_horsepower_model	2.956965208053589
4	dnn_model	1.7068039178848267
5	reloaded	1.7068039178848267

ВИСНОВКИ

Проведене дослідження у цій кваліфікаційній роботі доводить та підтверджує, що роль штучних нейронних мереж починає відігравати все більш важливу роль в аналізі та оцінці бізнес-процесів та їх ланцюгах. Таким чином, у сучасному бізнес-середовищі, традиційні методи не достатні, тому більшість компаній приходять до застосування штучних багатошарових нейронних мереж. У цій роботі розгорнута та застосована багатошарова нейронна мережа, яка працює ефективно з часовими рядами та послідовними даними. В першу чергу це потужний інтелектуальний інструмент адаптивного прогнозування, який діє у складних умовах бізнес-даних. Традиційні методи дають низьку точність та не надають довгостроковий прогноз, а також коли дані мають сезонність та тренди.

Особливу цінність ця кваліфікаційна робота надасть компаніям, які потребують швидкого та точного інструмента динамічного прогнозування, що критично важливо для: пандемії, геополітичної кризи та зміни ринкових трендів. Також скоротить їм витрати, зменшивши їх надлишкові запаси, та виявить можливі аномалії у транзакціях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) https://repository.dinus.ac.id/docs/ajar/Fundamentals_of_Business_Process_Management_1.pdf
- 2) <https://freshtech.global/blog/how-to-classify-and-optimize-business-processes>
- 3) <https://www.visual-paradigm.com/guide/bpmn/what-is-bpmn/>
- 4) <https://www.investopedia.com/terms/k/kpi.asp>
- 5) <https://www.investopedia.com/terms/r/returnoninvestment.asp>
- 6) <https://www.investopedia.com/terms/a/abandon-rate.asp>
- 7) <https://www.indeed.com/career-advice/career-development/process-optimization-methods>
- 8) <https://www.geeksforgeeks.org/what-is-perceptron-the-simplest-artificial-neural-network/>
- 9) https://www.w3schools.com/ai/ai_perceptrons.asp
- 10) <https://medium.com/data-science/sigmoid-neuron-deep-neural-networks-a4cd35b629d7>
- 11) <https://medium.com/@nlunge786/a-deep-architecture-multi-layer-perceptron-164bc5ff3842>
- 12) <https://medium.com/@anmoltalwar/gradient-descent-72b8d138cabe>
- 13) <https://www.geeksforgeeks.org/backpropagation-in-neural-network/>
- 14) <https://www.linkedin.com/pulse/real-world-industry-use-cases-neural-network-devendra-kanade>
- 15) https://d2l.ai/chapter_recurrent-modern/lstm.html
- 16) <https://www.tensorflow.org/tutorials/keras/regression>
- 17) <https://github.com/tensorflow/docs/blob/master/site/en/tutorials/keras/regression.ipynb>
- 18) <https://www.edrawsoft.com/what-is-idef.html?srsId=AfmBOorh1RHBeJ0uocBRCA3tRHFEAnjSKn72Hk9CsPUao1sJmXcy9hbm>
- 19) <https://colab.google/>

