МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

ДВНЗ «УЖГОРОДСЬКИЙ НАЙЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ»

**ЗВIT**

**ПРАКТИЧНА РОБОТА**

З дисципліни «Інформаційно-технічний супровід »

**Багатошаровий перцептрон. Задача класифікації**

**Частина 1**

Виконав:

Ролич Михайло Михайлович,

студент 3-го курсу

Перевірив:

проф. Поліщук Володимир

Ужгород 2025

**Варіант:** 3 Багатошаровий перцептрон. Задача класифікації.

**Мета**:  
Розробити та реалізувати модель багатошарового перцептрона для класифікації об'єктів на основі їх числових характеристик із використанням інструментів штучного інтелекту в середовищі браузера.

**Багатошаровий перцептрон (MLP)** — це один із базових типів штучних нейронних мереж, який використовується для вирішення задач класифікації, регресії та інших типів передбачення. Він є прикладом глибокого навчання, оскільки містить принаймні один прихований шар між вхідним і вихідним шарами.

MLP побудований із трьох основних типів шарів:

* Вхідний шар: приймає набір числових ознак (фіч), які описують об'єкт. У задачі класифікації фруктів це, наприклад, вага, колір у форматі RGB, солодкість тощо.
* Приховані шари: один або декілька шарів, які виконують обчислення з вхідними даними, дозволяючи моделі виявляти складні залежності між фічами. У кожному прихованому шарі використовуються нейрони (вузли), які застосовують активаційні функції (наприклад, ReLU), щоб додати нелінійність у модель.
* Вихідний шар: повертає результат у вигляді вектора ймовірностей для кожного класу. У задачах класифікації часто використовується функція активації Softmax, яка забезпечує те, що сума ймовірностей дорівнює 1.

**Активаційні функції**

ReLU (Rectified Linear Unit) — функція, яка повертає 0 для всіх від’ємних значень і саму величину для додатних. Це проста, але дуже ефективна функція, яка дозволяє мережі вчитись складним зв’язкам між ознаками.

Softmax — функція, яка перетворює вихідні значення в ймовірності, дозволяючи визначити, до якого з можливих класів належить об’єкт. Застосовується у вихідному шарі класифікаторів із кількома класами.

**Процес класифікації**

У загальному випадку процес виглядає так:

1. Користувач вводить набір числових ознак об’єкта.
2. Модель передає ці дані через всі шари нейронної мережі.
3. Вихідний шар формує ймовірності належності до кожного з класів.
4. Клас із найвищою ймовірністю вважається передбаченим.

**One-hot кодування**

Оскільки машинне навчання працює з числами, мітки класів (наприклад, «яблуко», «банан», «виноград») кодуються у вигляді векторів. Наприклад:

* «Яблуко» → [1, 0, 0]
* «Банан» → [0, 1, 0]
* «Виноград» → [0, 0, 1]

Таке кодування дозволяє мережі працювати з категоріальними змінними та розрізняти класи під час навчання.

**TensorFlow.js**

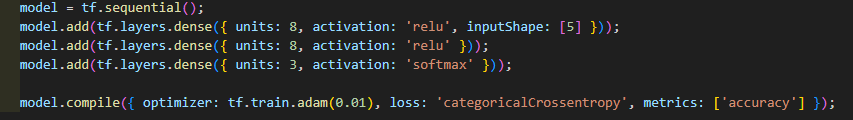
TensorFlow.js — це бібліотека для створення моделей машинного навчання безпосередньо у браузері. Вона дозволяє:

* будувати архітектуру моделі;
* тренувати модель на локальних даних без серверів;
* інтегрувати модель у вебінтерфейс;
* використовувати GPU браузера для прискорення обчислень.

Це дає можливість створювати інтерактивні ML-додатки, які працюють прямо в браузері, що особливо зручно для освітніх цілей або демонстрацій.

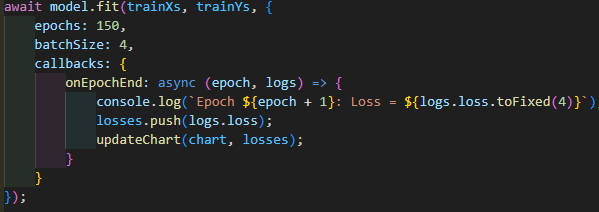
**1. Створення та компіляція моделі**

У ході реалізації моделі багатошарового перцептрона використовувалась бібліотека TensorFlow.js. Нижче наведено ключові фрагменти коду:



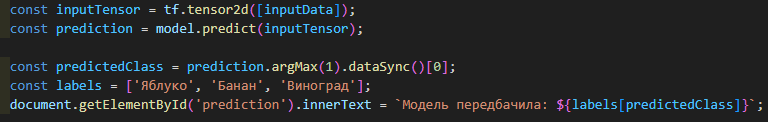
У моделі використано два приховані шари з функцією активації ReLU та вихідний шар із функцією Softmax, що відповідає задачі багатокласової класифікації.

**2. Навчання моделі з побудовою графіка втрат**

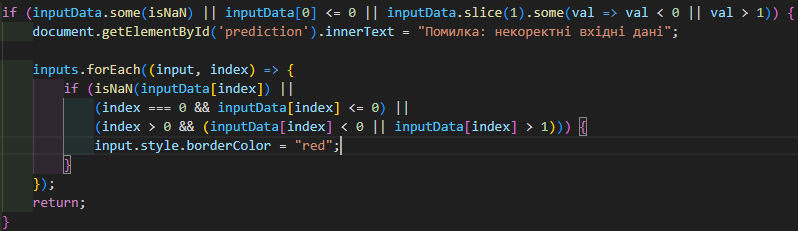


Під час навчання на кожній епосі оновлюється графік втрат для візуального відстеження якості навчання.

**3. Валідація та передбачення**

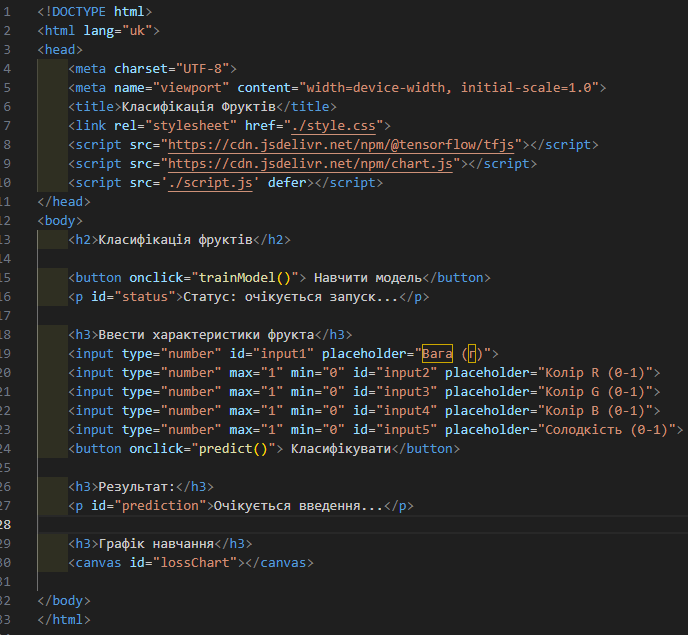
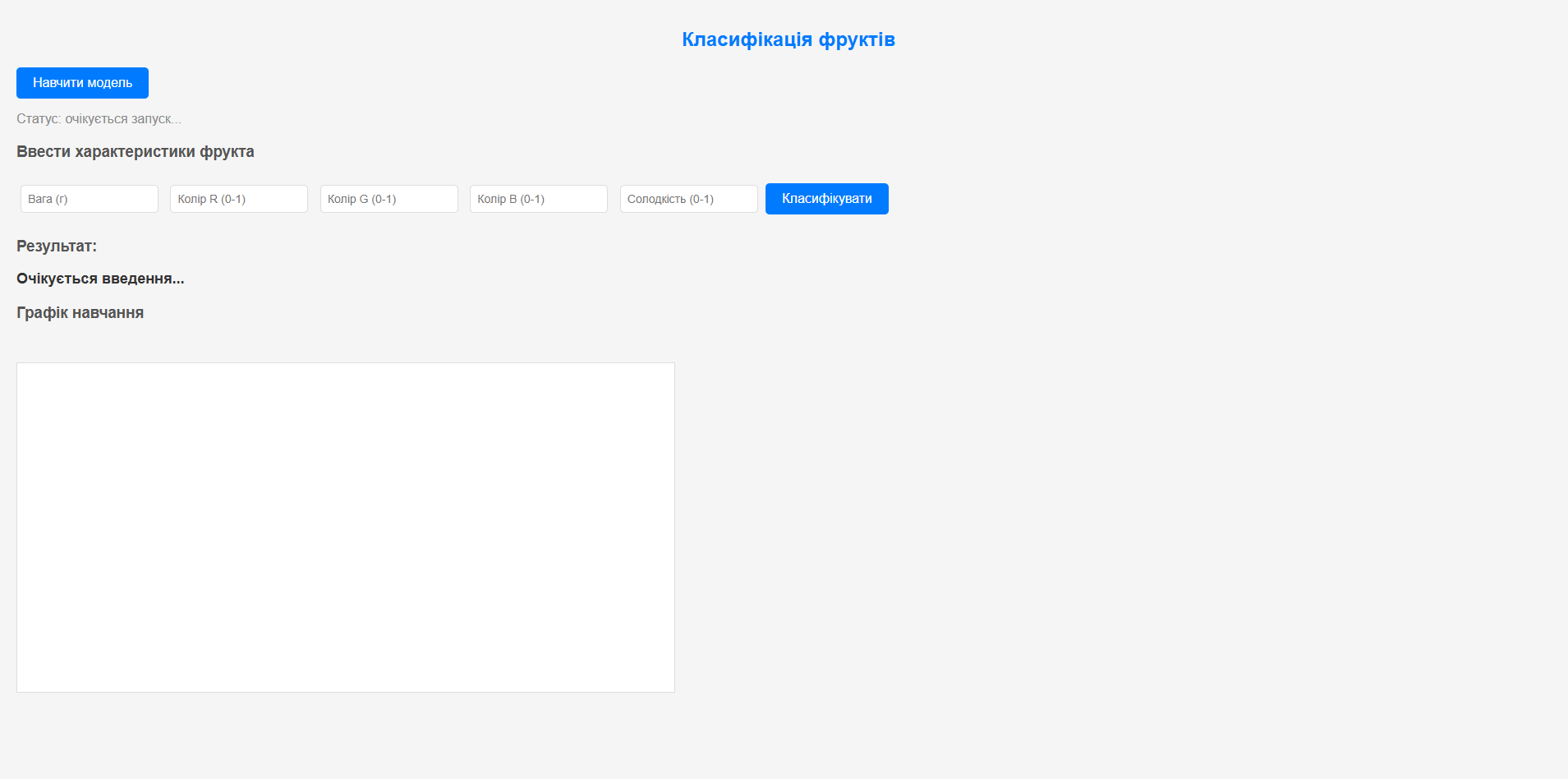
****

Користувач вводить 5 числових ознак, після чого модель класифікує об'єкт як один із класів: яблуко, банан, або виноград.



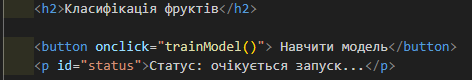
**Реалізація веб-інтерфейсу для класифікації фруктів**

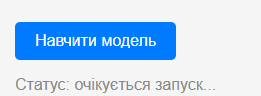
Для реалізації інтерфейсу користувача було створено веб-сторінку, яка дозволяє вводити характеристики фруктів та здійснювати їх класифікацію за допомогою натренованої моделі. Далі наведено фрагмент HTML-коду сторінки, який описує структуру інтерфейсу.



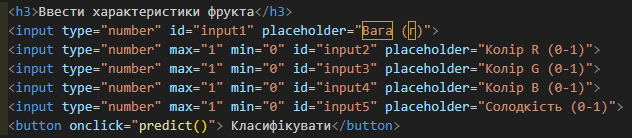
**Заголовок та кнопки для навчання моделі та класифікації**

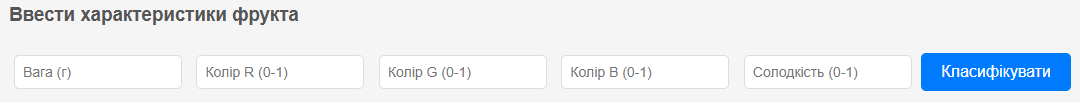
Кнопка "Навчити модель" ініціює процес навчання нейронної мережі. Після натискання з'являється повідомлення про статус навчання.





**Поля для введення характеристик фруктів**





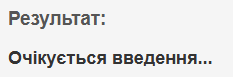
Ці поля дозволяють користувачеві ввести числові значення для характеристики фрукта. Вони включають:

* Вага (г)
* Колір у форматі RGB (в межах 0-1)
* Солодкість (0-1)

Після введення даних користувач натискає кнопку **"Класифікувати"**, щоб модель здійснила класифікацію фрукта.

**Виведення результату класифікації**

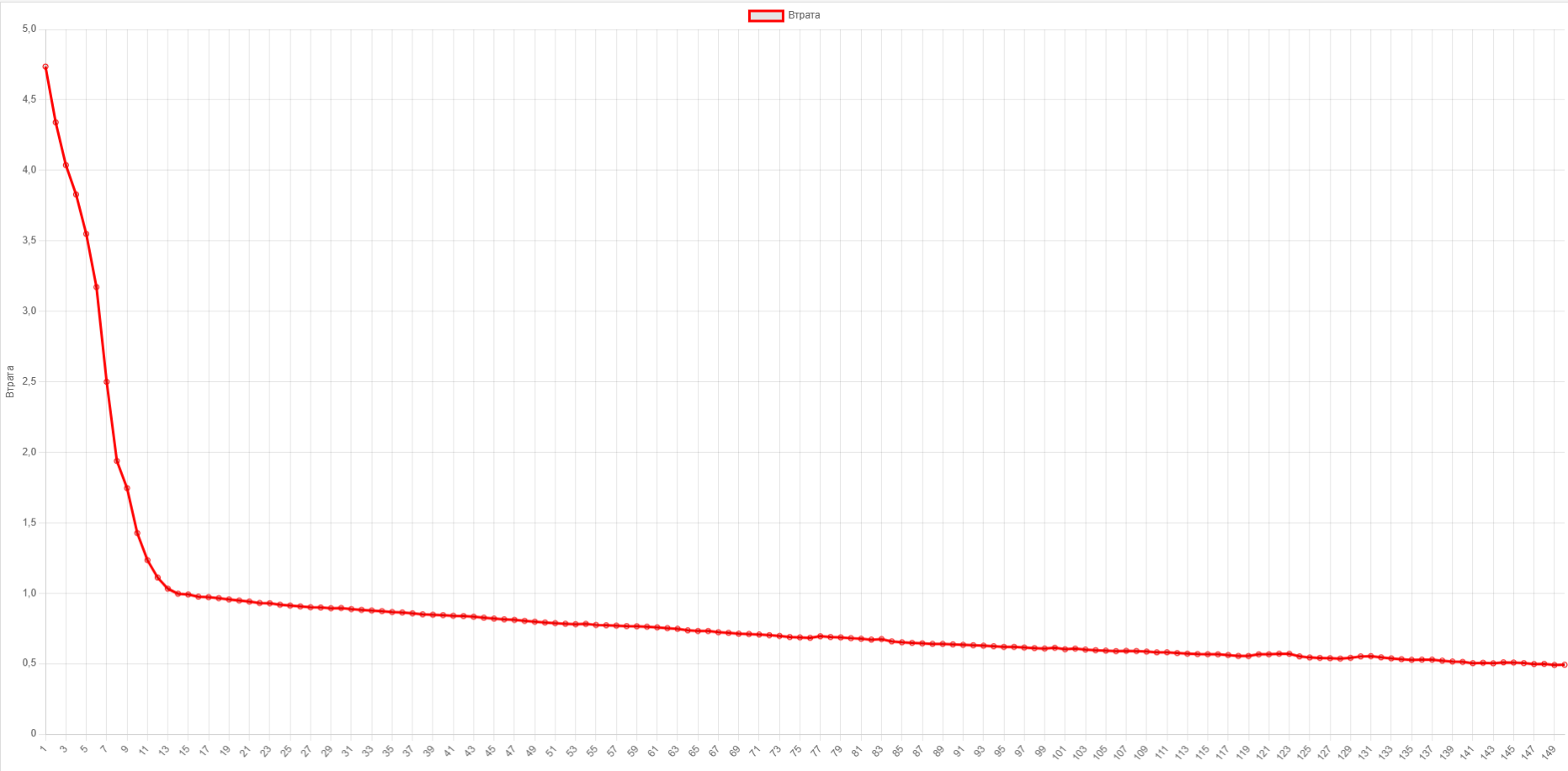
****

****

Цей блок відображає результат класифікації, де після обробки вхідних даних модель передбачить тип фрукта (яблуко, банан або виноград).

**Графік втрат під час навчання**





**Пояснення роботи інтерфейсу**

Навчання моделі: При натисканні кнопки "Навчити модель" запускається процес навчання моделі на попередньо заданих даних, використовуючи TensorFlow.js. Під час навчання на екрані оновлюється статус та графік втрат, що допомагає відслідковувати ефективність моделі.

Класифікація фрукта: Користувач вводить характеристики фрукта, такі як вага, колір у форматі RGB та солодкість, а потім натискає кнопку "Класифікувати". Модель передбачає клас фрукта на основі введених даних і відображає результат на екрані.

Графік втрат: Графік на сторінці надає візуальне відображення процесу навчання моделі. Це дозволяє побачити, як зменшується втрата під час тренування та оцінити, наскільки добре модель навчається.

**Висновки**

1. Розробка моделі класифікації фруктів: У ході виконання завдання була розроблена модель класифікації фруктів на базі багатошарового перцептрона (MLP). Модель була навчена на невеликому наборі даних, що включають такі характеристики фруктів, як вага, колір (RGB) та солодкість. Використано стандартні активаційні функції, такі як ReLU для прихованих шарів і Softmax для вихідного шару, що дозволяє моделі передбачати ймовірність належності до одного з трьох класів: яблуко, банан чи виноград.
2. Використання TensorFlow.js: Для реалізації моделі було використано бібліотеку TensorFlow.js, що дозволяє створювати та тренувати нейронні мережі безпосередньо в браузері. Цей підхід дозволяє розробляти інтерактивні веб-додатки з використанням машинного навчання без потреби у серверній частині, що значно спрощує розгортання і використання моделей у реальних умовах.
3. Інтерактивний веб-інтерфейс: Створено простий і зрозумілий веб-інтерфейс для взаємодії з моделлю. Користувач може ввести характеристики фрукта, після чого система класифікує фрукт і відображає результат. Також передбачено виведення графіка навчання, що допомагає візуалізувати процес тренування моделі і оцінити її ефективність.
4. Процес навчання та класифікації: Модель успішно навчається на заздалегідь підготовлених даних, знижуючи рівень втрат в процесі навчання. Після завершення навчання система здатна точно класифікувати фрукт за введеними характеристиками. Під час навчання виводиться інформація про поточні втрати, що дозволяє відстежувати ефективність оптимізації.
5. Перспективи розвитку: Подальше вдосконалення моделі передбачає розширення набору даних для навчання, що дозволить підвищити точність класифікації. Також можна реалізувати більш складні моделі глибокого навчання для обробки більшої кількості ознак та кращого розпізнавання фруктів. Використання більш складних наборів даних дозволить зробити модель більш універсальною та здатною до роботи з іншими типами класифікацій.
6. Висока ефективність TensorFlow.js: Використання TensorFlow.js для навчання моделей безпосередньо в браузері показало високу ефективність. Це дозволяє не лише здійснювати навчання безпосередньо на клієнтських машинах, але й використовувати графічні процесори для прискорення обчислень. Цей підхід може бути корисним у випадках, коли необхідно створити інтерактивні програми для навчання та демонстрацій.

Таким чином, реалізована модель класифікації фруктів демонструє можливості застосування машинного навчання для вирішення реальних завдань класифікації, а також показує, як можна інтегрувати такі моделі в веб-додатки для зручного користування.