Лабораторна робота 1.1: Лінійна регресія з однією змінною

Мета лабораторної роботи:

- Імплементуєте модель лінійної регресії з однією змінною;
- Обчислите функцію вартості (cost function) ;
- Реалізуєте алгоритм градієнтного спуску для знаходження оптимальних параметрів;
- Візуалізуєте результати та процес навчання;

Виконання:

1. Вибір варіанту завдання:

Для виконання завдання було обрано П'ЯТИЙ варіант, відповідно до розподілу студентів на сайті.

2. Підготовака даних:

Імпортування необхідних бібліотек:

- numpy для роботи з математичними операціями, функціями лінійної алгебри та іншого;

- matplotlib для побудови графіків.

Початкові дані відповідно до варіанту:

Опис: Залежність балів за тест (0-100) від кількості годин навчання.

Набір даних:

```
x_train = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])  # години навчання
y_train = np.array([30, 40, 45, 55, 60, 70, 75, 85, 90, 95])  # бали за тест
```

Рекомендовані параметри для початку:

- w_init = 7
- b_init = 30
- alpha = 0.01
- iterations = 1000

Візуалізація даних вхідних результатів:

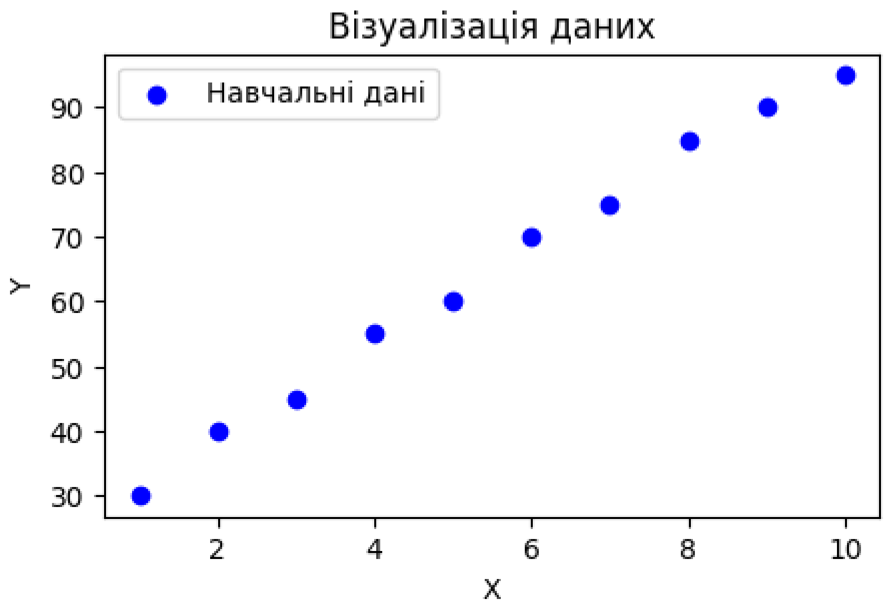

Рисунок 1 — Результат візуалізації

3. Реалізація функцій

Реалізація функції обчислення вихідних значень моделі `compute_model`:

Функція `compute_model(x, w, b)` реалізує лінійну модель прогнозування за формулою:

$$f_{w,b}(x^{(i)}) = wx^{(i)} + b$$

Вона має три аргументи:

- x — вхідні дані (масив розміром m);
- w — ваговий коефіцієнт;
- b — зміщення.

Спочатку визначається кількість елементів у x, після чого створюється масив нулів y_pred для збереження прогнозованих значень. Далі у циклі кля кожного елемента обчислюється прогноз за лінійною формулою і зберігається в y_pred. Наприкінці функція повертає масив прогнозів.

Реалізація функції обчислення вартості `compute_cost`:

Функція compute_cost(y, y_pred) обчислює значення функції вартосі, яке показує наскільки добре модель передбачає результати. Вона реалізує середньоквадратичну помилку поділену на 2 і розраховується за формулою:

$$J(w,b) = \frac{1}{2m} \sum_{i=0}^{m-1} \left(f_{w,b}(x^{(i)}) - y^{(i)}\right)^2 \quad ,$$

де:

- y — фактичні значення;
- y_pred — передбачені значення моделі;
- m — кількість прикладів у наборі даних.

Реалізація функціїобчислення градієнту `compute_gradient:`

Функція compute_gradient(x, y, y_pred) обчислює градієнти похибки за параметрам моделі. Вона використовує алгоритм градієнту спуску для оновленя параметрів з метою мінімізувати функції вартості.

- x — масив вхідних значень;
- y — масив цільових значень;
- y_pred — масив прогнозованих значень;
- m — кількість прикладів у наборі.

Реалізація функції градієнтного спуску `gradient_descent:`

Функція реалізує алгоритм градієнтного спуску для знаходження оптимальних значень параметрів лінійної моделі. Відбувається покрокове зменшення значення функції вартості, змінюючи параметри моделі у напрямку протилежному до градієнта. Розраховується за формулами:

$$w = w - \alpha \frac{\partial J(w,b)}{\partial w}$$

$$b = b - \alpha \frac{\partial J(w,b)}{\partial b}$$

де:

$$\frac{\partial J(w,b)}{\partial w} = \frac{1}{m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)}) x^{(i)}$$

$$\frac{\partial J(w,b)}{\partial b} = \frac{1}{m} \sum_{i=0}^{m-1} (f_{w,b}(x^{(i)}) - y^{(i)})$$

4. Запуск градієнтного спуску

Виконаємо функцію gradient_decent(x_train, y_train, w_init, b_init, alpha, iterations), де:

- x_train — список вхідних даних;
- y_train — список цільових значень;
- w_init — початкове значення коеф. ваги;
- b_init — початкове значення зміщення;
- alpha — швидкісь навчання;
- iteration — кількість ітерацій градієнтного спуску.

Результат виконання функції:

- Ітерація 0: cost = 9.5

- Ітерація 100: cost = 3.324131197031415
- Ітерація 200: cost = 2.5680608072991307
- Ітерація 300: cost = 2.0714835784844103
- Ітерація 400: cost = 1.7453380931882176
- Ітерація 500: cost = 1.5311299672276777
- Ітерація 600: cost = 1.3904408475059664
- Ітерація 700: cost = 1.2980380578390722
- Ітерація 800: cost = 1.2373491036746551
- Ітерація 900: cost = 1.197489385468655
- Ітерація 1000: cost = 1.1713100393695746

5. Аналіз та візуалізація процесів

Прогнозування:

y_pred = compute_model(x_train, w_opt, b_opt)

print(y_pred) #30, 40, 45, 55, 60, 70, 75, 85, 90, 95

Результат:

[32.22067926,
39.4257,
46.63072074,
53.83574147,
61.04076221,
68.24578295,
75.45080368,
82.65582442,
89.86084515,
97.06586589]

Візуалізація:

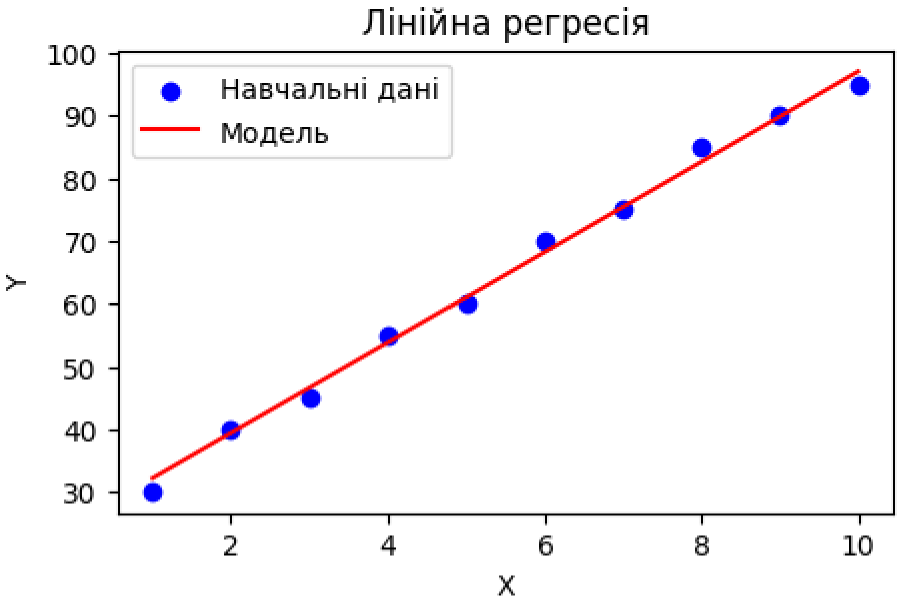

Рисунок 2 – Результат візуалізації

Зміна функції вартості під час навчання:

Рисунок 3 – Графік зміни функції вартості

Питання для самоперевірки:

1. Що таке лінійна регресія та коли її доцільно застосовувати?

Це найпростіший та найпоширеніший метод машинного навчання, що використовують для моделювання залежності між однією чи декількома незалежними ознаками і однією цільовою змінною.

Доцільно застосовувати, коли видно, що є лінійний зв'язок між змінними та коли потрібно знайти швидке та просте рішення.

2. Поясніть значення параметрів w і b у моделі лінійної регресії.

w — коефіцієнт нахилу, показує наскільки сильно зміниться цільова змінна, якщо змінюється незалежна ознака.

b — зміщення, точка, де пряма перетинає вісь y.

Ці значення використовуються для мінімізації помилки між прогнозуванням і реальними значеннями.

3. Що таке функція вартості та яку роль вона відіграє в навчанні моделі?

Це математична функція, що вимірює, наскільки добре модель передбачає дані. Відіграє роль «вимірювача помилки» з метою зробити її якомога меншою.

4. Як працює алгоритм градієнтного спуску?

Алгоритм крок за кроком змінює параметри моделі так, щоб зменшити функцію вартості.

5. Як вибір швидкості навчання (alpha) впливає на процес навчання?

Швидкість навчання визначає наскільки сильно змінюються параметри моделі на кожній ітерації градієнтного спуску. Якщо швидкість навчання занадто мала — швидкість змінення моделі дуже повільна; якщо занадто велика — параметри змінюються занадто різко, а функція зменшення вартості почати збільшуватись або вийти з-під контролю.

6. Що таке коефіцієнт детермінації (R^2) і як його інтерпретувати?

Це покажчик, який вказує наскільки добре модель «вгадує» реальні дані. Якщо коефіцієнт детермінації наближається до 1 — передбачення моделі майже сходяться з реальними значеннями, чим менше значення — тим менше результат моделі відповідає дійсності.

7. Які переваги та недоліки лінійної регресії з одним змінним?

Переваги: легка та швикда модель, яка добре підходить для простих задач, де одна змінна впливає на іншу, леко візуалізувати.

Недоліки: не підходить для нелінійної залежності, не враховує інші можливі влпиви, чутлива до помилок та «нестандартних» значень даних.

8. Як визначити, що модель була успішно навчена?

Успішність навчання моделі можна визначити за декількома параметрами:

- За допомогою мінімальної функціїї вартості: якщо функція вартості зменшується поступово і стабілізуєтсья на низькому рівні, то можна вважати що модель добре адаптувалася до даних;

- Визначити коефіцієнт детермінації: якщо він наближається до 1 — це вказує на те, що модель добре пояснює варіативність даних;

- Перевірити на тестовому наборі даних та візуалізувати отримані значення: модель має добре прогнозувати значення для даних, що не використовувались при навчанні.

9. Які проблеми можуть виникнути під час навчання моделі?

- Вибір невідповідної швидкості навчання;
- Переобучення або недообучення моделі;
- Велика кількість взаємозалежних змін;
- Наявність «аномайй» або «шуму»;
- Неадекватність лінійної моделі;
- Невідповідні дані для градієнту спуску.

10. Як можна покращити точність моделі лінійної регресії?

- Покращити якість вхідних даних;
- Розширити набір ознак;
- Оптимізація параметрів;
- Збільшити кількість вхідних якісних даних.

Висновки: під час виконання роботи було реалізовано модель лінійної регресії з однією зміною. Було підготовлено та візуалізовано дані, реалізовано функції обрахунку моделі, вартості, градієнтів та функцію градієнтного спуску, виконано навчання моделі для знаходження оптимальних параметрів. Було проаналізовано вплив параметрів моделі на точність її передбачень, застосовано алгоритм градієнтного спуску для зменшення функції вартосі і знаходження найкращих значень параметрів, проведено візуалізацію результатів та тестування швидкості навчання шляхом зміни значення швидкості навчання.