

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

ЛЕКЦІЯ 2. Різновиди методів навчання з вчителем. Лінійна регресія

Львів -- 2025

Лекція зі штучного інтелекту 2025-02

Вступ

На цьому занятті ми розглянемо машинне навчання з вчителем (supervised learning), зосередившись на лінійній регресії як одному з фундаментальних алгоритмів цього підходу. Ми дослідимо основні концепції, методи оптимізації та практичні аспекти застосування цих технік.

Теми, що розглядаються

1. Машинне навчання з вчителем: концепція та принципи
2. Лінійна регресія як базовий алгоритм
3. Функція втрат та її роль у навчанні моделі
4. Градієнтний спуск як метод оптимізації
5. Гіперпараметри та їх налаштування
6. Практичні застосування лінійної регресії

Машинне навчання з вчителем

Типи машинного навчання

В залежності від того, як саме алгоритми машинного навчання можуть прийти до побудування прогнозів, розрізняють наступні типи навчання:

Навчання з вчителем (Supervised Learning)

Навчання з вчителем — це тип машинного навчання, при якому алгоритм навчається на розмічених даних. Розмічені дані — це дані, де для кожного вхідного прикладу (x) вже відомий правильний вихід або мітка (y).

Моделі навчання з вчителем здатні робити прогнози після аналізу великої кількості даних з правильними відповідями, виявляючи зв'язки між елементами даних, що призводять до цих відповідей. Такі системи машинного навчання називаються "з вчителем", оскільки зовнішнє джерело надає системі дані з відомими правильними результатами.

Основні типи задач навчання з вчителем:

- **Класифікація:** віднесення вхідних даних до певних категорій або класів (наприклад, спам/не спам, розпізнавання цифр)
- **Регресія:** прогнозування неперервних числових значень (наприклад, ціни будинку, температури)

Навчання без вчителя (Unsupervised Learning)

Навчання без вчителя працює з нерозміченими даними, де алгоритм самостійно шукає структуру або закономірності. Основні типи задач:

- **Кластеризація:** групування схожих об'єктів у кластери
- **Зменшення розмірності:** зменшення кількості змінних у даних, зберігаючи при цьому найважливішу інформацію
- **Виявлення аномалій:** ідентифікація незвичайних або відхилених зразків у даних

На ілюстрації нижче показано приклад кластеризації.



Навчання з підкріпленням (Reinforcement Learning)

Навчання з підкріпленням — це підхід, де агент навчається приймати рішення, взаємодіючи з середовищем. Агент отримує винагороди або покарання за свої дії, і його мета — максимізувати сумарну винагороду з часом.

Цей підхід особливо ефективний для:

- Ігор (шахи, Go, відеоігри)
- Робототехніки
- Автономних транспортних засобів
- Оптимізації ресурсів

Генеративний ШІ (Generative AI)

Генеративний ШІ — це підкатегорія машинного навчання, яка створює новий контент на основі навчальних даних. Ці моделі можуть генерувати:

- Текст (статті, діалоги, код)
- Зображення
- Аудіо (музика, мовлення)

- Відео

Генеративні моделі, такі як трансформери та генеративно-змагальні мережі (GANs), здатні створювати реалістичний контент, який може бути важко відрізнити від створеного людиною.

Основні концепції машинного навчання з вчителем

Машинне навчання з вчителем включає наступні ключові елементи:

- **Навчальні дані:** набір прикладів, де кожен приклад має вхідні ознаки (features) та відповідну мітку (label)
- **Модель:** математична структура, яка вивчає шаблони в даних
- **Функція втрат:** міра того, наскільки прогнози моделі відрізняються від фактичних значень
- **Алгоритм оптимізації:** метод мінімізації функції втрат шляхом коригування параметрів моделі



Процес машинного навчання з вчителем

Типовий процес машинного навчання з вчителем включає наступні етапи:

1. **Збір та підготовка даних:** отримання розмічених даних та їх попередня обробка
2. **Вибір моделі:** визначення типу моделі, яка найкраще підходить для вирішення задачі
3. **Навчання моделі:** використання алгоритму оптимізації для налаштування параметрів моделі
4. **Оцінка моделі:** перевірка ефективності моделі на тестових даних
5. **Налаштування гіперпараметрів:** оптимізація параметрів алгоритму навчання
6. **Прогнозування:** використання навченої моделі для прогнозування на нових даних

Лінійна регресія

Лінійна регресія — це один з найпростіших та найбільш фундаментальних алгоритмів машинного навчання з вчителем. Вона моделює лінійне відношення між вхідними змінними (ознаками) та вихідною змінною (міткою).

Математична модель лінійної регресії

У найпростішому випадку (з однією ознакою) лінійна регресія описується рівнянням:

$$\hat{y} = wx + b$$

де:

- \hat{y} — прогнозоване значення
- x — вхідна ознака
- w — вага (коефіцієнт нахилу)
- b — зміщення (перетин з віссю y)

Для багатовимірного випадку (з багатьма ознаками) рівняння розширюється до:

$$y = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

або у векторній формі:

$$y = \mathbf{w}^T \mathbf{x} + b$$

 Графічне представлення лінійної регресії

Мета лінійної регресії

Мета лінійної регресії — знайти такі значення параметрів \mathbf{w} та b , які мінімізують різницю між прогнозованими значеннями та фактичними значеннями у навчальних даних.

Функція втрат у лінійній регресії

Для оцінки якості моделі лінійної регресії використовується функція втрат. Найпоширенішою функцією втрат для лінійної регресії є середньоквадратична помилка (Mean Squared Error, MSE).

Середньоквадратична помилка (MSE)

MSE обчислюється як середнє значення квадратів різниць між прогнозованими та фактичними значеннями:

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

де:

- N — кількість прикладів
- y_i — фактичне значення для i -го прикладу
- \hat{y}_i — прогнозоване значення для i -го прикладу

Квадрат різниці використовується з кількох причин:

- Він перетворює від'ємні різниці на додатні значення
- Він надає більшої ваги великим помилкам
- Він створює гладку функцію, яку легше оптимізувати

 Графік функції втрат

Інтерпретація функції втрат

Функція втрат створює "ландшафт" у просторі параметрів моделі. Наша мета — знайти найнижчу точку цього ландшафту, яка відповідає оптимальним значенням параметрів.

Гرادієнтний спуск

Градiєнтний спуск — це ітеративний алгоритм оптимізації, який використовується для знаходження мінімуму функції втрат шляхом поступового руху в напрямку, протилежному градієнту функції.

Принцип роботи градієнтного спуску

1. Починаємо з деяких початкових значень параметрів (зазвичай випадкових або нульових)
2. Обчислюємо градієнт функції втрат відносно кожного параметра
3. Оновлюємо параметри, рухаючись у напрямку, протилежному градієнту
4. Повторюємо кроки 2-3, доки не досягнемо збіжності

Математично, оновлення параметрів відбувається за формулою:

$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} \text{MSE} \quad b := b - \alpha \frac{\partial}{\partial b} \text{MSE}$$

де α — швидкість навчання (learning rate), яка визначає розмір кроку в напрямку, протилежному градієнту.

 Ілюстрація градієнтного спуску

Варіанти градієнтного спуску

Існує кілька варіантів градієнтного спуску:

1. **Пакетний градієнтний спуск (Batch Gradient Descent)**: використовує всі навчальні приклади для обчислення градієнта на кожній ітерації
2. **Стохастичний градієнтний спуск (Stochastic Gradient Descent, SGD)**: використовує один випадковий приклад для обчислення градієнта на кожній ітерації
3. **Міні-пакетний градієнтний спуск (Mini-batch Gradient Descent)**: використовує підмножину навчальних прикладів для обчислення градієнта на кожній ітерації

Проблеми градієнтного спуску

При використанні градієнтного спуску можуть виникнути наступні проблеми:

1. **Вибір швидкості навчання**: занадто велика швидкість може призвести до розбіжності, а занадто мала — до повільної збіжності
2. **Локальні мінімуми**: алгоритм може застрягти в локальному мінімумі (хоча для лінійної регресії функція втрат є опуклою і має лише один глобальний мінімум)
3. **Плато**: області з майже нульовим градієнтом, де навчання сповільнюється

Гіперпараметри в лінійній регресії

Гіперпараметри — це параметри алгоритму навчання, які встановлюються перед початком процесу навчання, на відміну від параметрів моделі, які визначаються під час навчання.

Основні гіперпараметри в лінійній регресії

1. **Швидкість навчання (learning rate)**: визначає розмір кроку в градієнтному спуску
2. **Кількість ітерацій**: максимальна кількість кроків градієнтного спуску
3. **Регуляризація**: параметр, який контролює складність моделі для запобігання перенавчанню
 - L1-регуляризація (Lasso)
 - L2-регуляризація (Ridge)

4. **Розмір міні-пакету:** кількість прикладів, що використовуються для обчислення градієнта в міні-пакетному градієнтному спуску

Налаштування гіперпараметрів

Для знаходження оптимальних значень гіперпараметрів використовуються різні методи:

1. **Пошук по сітці (Grid Search):** перебір всіх можливих комбінацій гіперпараметрів з заданого набору значень
2. **Випадковий пошук (Random Search):** випадковий вибір комбінацій гіперпараметрів з заданого діапазону
3. **Байєсівська оптимізація:** послідовний вибір гіперпараметрів на основі результатів попередніх експериментів



Вплив гіперпараметрів на модель

- **Швидкість навчання:** занадто велика швидкість може призвести до розбіжності, а занадто мала — до повільної збіжності
- **Регуляризація:** сильна регуляризація може призвести до недонавчання, а слабка — до перенавчання
- **Кількість ітерацій:** занадто мала кількість може призвести до недонавчання, а занадто велика — до перенавчання або надмірних обчислювальних витрат

Практичні застосування лінійної регресії

Лінійна регресія широко застосовується в різних галузях:

1. **Економіка та фінанси:** прогнозування цін, аналіз попиту та пропозиції
2. **Медицина:** прогнозування результатів лікування, аналіз факторів ризику
3. **Маркетинг:** аналіз ефективності рекламних кампаній
4. **Інженерія:** моделювання фізичних процесів
5. **Соціальні науки:** аналіз взаємозв'язків між різними факторами

Висновок

У цій лекції ми розглянули основи машинного навчання з вчителем, зосередившись на лінійній регресії як одному з фундаментальних алгоритмів. Ми дослідили математичну модель лінійної регресії, функцію втрат, градієнтний спуск як метод оптимізації та роль гіперпараметрів у навчанні моделі.

Лінійна регресія, незважаючи на свою простоту, є потужним інструментом для моделювання лінійних залежностей та служить основою для розуміння більш складних алгоритмів машинного навчання. У наступних лекціях ми розглянемо інші алгоритми машинного навчання з вчителем та їх застосування для вирішення різноманітних задач.