**ЛЕКЦІЯ 1. ВВЕДЕННЯ В МАШИННЕ НАВЧАННЯ**

1. **Що таке машинне навчання?**

Машинне навчання (МН) лежить в основі деяких з найважливіших технологій, які ми використовуємо, від застосунків для перекладу до автономних транспортних засобів. У цьому курсі пояснюються основні концепції машинного навчання.

Машинне навчання пропонує новий спосіб вирішення проблем, відповіді на складні запитання та створення нового контенту. МН може передбачати погоду, оцінювати час у дорозі, рекомендувати пісні, автоматично заповнювати речення, резюмувати статті та створювати раніше невидані зображення.

По суті, машинне навчання — це процес навчання частини програмного забезпечення, яке називається моделлю, що дозволяє робити корисні прогнози або генерувати контент на основі даних.

Наприклад, припустимо, що ми хочемо створити застосунок для прогнозування опадів. Ми могли б використати або традиційний підхід, або підхід МН. Використовуючи традиційний підхід, ми створили б фізичне представлення атмосфери та поверхні Землі, обчисливши величезну кількість рівнянь гідродинаміки. Це неймовірно складно.

Використовуючи підхід МН, ми передавали моделі МН величезні обсяги даних про погоду, поки модель МН зрештою не вивчила математичний зв’язок між погодними умовами, що спричиняють різну кількість дощу. Потім ми передавали моделі поточні дані про погоду, і вона прогнозувала кількість опадів.

Перевірте себе:

Что такое «модель» в машинном обучении?

Модель — это часть компьютерного оборудования.

Модель — это математическое соотношение, полученное на основе данных, которые система машинного обучения использует для прогнозирования.

Модель — это уменьшенное представление того, что вы изучаете.

1. **Типи систем машинного навчання**

Системи МН потрапляють в одну або кілька наступних категорій в залежності від того, як вони навчаються робити прогнози або генерувати контент:

* Навчання з учителем
* Навчання без учителя
* Навчання з підкріпленням
* Генеративний ШІ
1. **Навчання з вчителем**

Моделі навчання з учителем можуть робити прогнози після перегляду великої кількості даних з правильними відповідями, а потім виявлення зв'язків між елементами даних, які дають правильні відповіді. Це схоже на те, як студент вивчає новий матеріал, вивчаючи старі екзаменаційні роботи, які містять як питання, так і відповіді. Як тільки студент натренується на достатній кількості старих екзаменаційних робіт, він буде добре підготовлений до здачі нового екзамену. Ці системи МН «контролюються» в тому сенсі, що людина надає системі МН дані з відомими правильними результатами.

Двома найбільш поширеними варіантами використання навчання з учителем є регресія та класифікація.

**Регресія**

Модель регресії передбачає числове значення. Наприклад, модель погоди, яка прогнозує кількість дощу в дюймах або міліметрах, є регресійною моделлю.

У таблиці нижче наведені додаткові приклади регресійних моделей:

| **Сценарій** | **Можливі входи** | **Числовий прогноз** |
| --- | --- | --- |
| Майбутня ціна будинку | Площа ділянки, поштовий індекс, кількість спалень та ванних кімнат, розмір ділянки, відсоткова ставка по іпотеці, ставка податку на нерухомість, витрати на будівництво та кількість будинків, що продаються в цьому районі. | Ціна будинку. |
| Майбутній час поїздки | Історичні умови дорожнього руху (зібрані зі смартфонів, датчиків дорожнього руху, служб таксі та інших навігаційних додатків), відстань до пункту призначення та погодні умови. | Час в хвилинах і секундах до прибуття в пункт призначення. |

**Класифікація**

Моделі класифікації передбачають ймовірність того, що щось належить до певної категорії. На відміну від регресійних моделей, вихідні дані яких є числом, моделі класифікації виводять значення, яке вказує на те, чи належить щось до певної категорії. Наприклад, моделі класифікації використовуються, щоб передбачити, чи є електронний лист спамом або чи зображено на фотографії кота.

Моделі класифікації поділяються на дві групи: бінарна класифікація та мультикласова класифікація.

Моделі бінарної класифікації виводять значення з класу, який містить лише два значення, наприклад модель, яка виводить або дощ, або без дощу.

Моделі мультикласової класифікації виводять значення з класу, який містить більше двох значень, наприклад модель, яка може виводити дощ, град, сніг або слякоть.

Перевірте себе:

Якщо б ви хотіли використовувати модель машинного навчання для прогнозування споживання енергії комерційними будівлями, який тип моделі ви б використали?

Класифікація

Регресія

1. **Навчання без вчителя**

Моделі навчання без вчителя (без нагляду) роблять прогнози, отримуючи дані, які не містять правильних відповідей. Мета моделі навчання без нагляду — виявити значущі закономірності серед даних. Іншими словами, у моделі немає підказок щодо того, як класифікувати кожен фрагмент даних, замість цього вона повинна створювати власні правила.

Широко використовувана модель навчання без нагляду використовує метод, який називається кластеризацією. Модель знаходить точки даних, які розмежовують природні групи.

Кластеризація відрізняється від класифікації тим, що категорії не визначаються вами. Наприклад, неконтрольована модель може групувати набір погодних даних на основі температури, виявляючи сегментації, які визначають пори року. Потім ви можете спробувати назвати ці кластери на основі вашого розуміння набору даних.

 

Рис. 1 Результат кластеризації

Перевірте своє розуміння

Що відрізняє контрольований підхід від неконтрольованого?

Неконтрольований підхід знає, як маркувати кластери даних.

При контрольованому підході надаються дані, що містять правильну відповідь.

Контрольований підхід зазвичай використовує кластеризацію.

1. **Навчання з підкріпленням**

Моделі навчання з підкріпленням роблять прогнози, отримуючи винагороди або штрафи залежно від дій, виконуваних у середовищі. Система навчання з підкріпленням генерує політику, яка визначає найкращу стратегію для отримання максимальної винагороди.

Навчання з підкріпленням використовується для навчання роботів виконувати завдання, такі як пересування по кімнаті, а також для таких програм, як AlphaGo, для гри в го.

Що таке навчання з підкріпленням, або Reinforcement learning? Майже те ж саме, що й навчання з учителем, але в ролі "вчителя" виступає справжнє або віртуальне середовище. Давайте "на пальцях": пам'ятайте популярний спосіб навчити плавати? Вивезти в човні на середину озера і викинути на глибину з повчанням: «Жити хочеш - випливеш».

Цей самий принцип діє в навчанні з підкріпленням: робота кидають в якийсь лабіринт, з якого він сам повинен знайти вихід. У процесі пошуку робот отримує від зовнішнього середовища інформацію про те, де виходу немає, таким чином вивчає навколишній світ і вчиться знаходити шлях до виходу.

Нагородою за успішно виконане завдання є можливість взятися за нове, а також набрані в процесі виконання бали. Чим ефективніше виконана задача, тим більше нараховується балів.

Навчання з підкріпленням застосовується у випадках коли потрібно вибрати кращий варіант серед багатьох або досягти складної мети за безліч ходів. Алгоритми підкріплення, які включають в себе глибоке навчання, можуть перемогти чемпіонів світу в грі Go, починаючи з базового розуміння правил гри і тренуючись від партії до партії.

Таким чином, це вже штучний інтелект в дії: машина намагається вирішити задачу різними способами, помиляється, вчиться на своїх помилках, покращує показники. Цей метод використовують у першу чергу там, де потрібно навчити машину виживати в реальному середовищі.

1. **Генеративний ШІ**

Генеративний Штучний Інтелект (ШІ) — це клас моделей, які створюють контент на основі користувацького вводу. Наприклад, генеративний ШІ може створювати унікальні зображення, музичні композиції та жарти; він може резюмувати статті, пояснювати, як виконати завдання, або редагувати фотографії.

Генеративний ШІ може приймати різні входи і створювати різні виходи, такі як текст, зображення, аудіо та відео. Він також може брати і створювати їх комбінації. Наприклад, модель може використовувати зображення як вхідні дані і створювати зображення та текст як вихідні дані або використовувати зображення та текст як вхідні дані і створювати відео як вихідні дані.

Ми можемо обговорювати генеративні моделі за їхніми входами та виходами, які зазвичай записуються як «тип входу» до «типу виходу». Наприклад, нижче наведено неповний список деяких входів і виходів для генеративних моделей:

* Текст у текст
* Перетворення тексту в зображення
* Перетворення тексту в відео
* Перетворення тексту в код
* Перетворення тексту в мову

У наступній таблиці наведено приклади генеративних моделей, їхні входи та приклад можливих виходів:

| **Модель** | **Вхід** | **Приклад виводу** |
| --- | --- | --- |
| Текст у текст | Кто придумал гонку Ле-Ман? | Гонка 24 години Ле-Ман була організована в 1923 році. Ідея цієї гонки принадлежит Жану-Луї Шевроном і Луї-Роже Пюжолом. Ось кілька ключевых фактов про гонку: |
| текст в зображення | Інопланетний восьминіг пропливає через портал і читає газету**.**  | Изображение мультяшного осьминога. |
| текста у код | Напишіть цикл Python, який перебирає список чисел і друкує прості числа. | for number in numbers:  # Check if the number is prime.  is\_prime = True  for i in range(2, number):    if number % i == 0:        is\_prime = False        break  # If the number is prime, print it.  if is\_prime:    print(number) |
| зображення в текст | Изображение фламинго. | Це фламінго. Вони обітають в Карибському морі |

Як працює генеративний ІІ? На високому рівні генеративні моделі вивчають закономірності в даних з метою створення нових, але схожих даних. Генеративні моделі виглядають наступним чином:

Юмористи, які навчаються наслідувати інших, спостерігаючи за поведінкою та стилем мовлення людей.

Художники, які навчаються малювати в певному стилі, вивчаючи безліч картин в цьому стилі.

Кавер-групи, які навчаються звучати як певна музична група, слухаючи багато музики цієї групи.

Для отримання унікальних і творчих результатів генеративні моделі спочатку навчаються за допомогою неконтрольованого підходу, при якому модель навчається імітувати дані, на яких вона навчена. Іноді модель додатково навчається з використанням контрольованого навчання або навчання з підкріпленням на конкретних даних, пов'язаних із завданнями, які модель може бути запропонована виконати, наприклад, підвести підсумок статті або відредагувати фотографію.

Генеративний штучний інтелект — це швидко розвиваюча технологія, в якій постійно з'являються нові варіанти використання. Наприклад, генеративні моделі допомагають компаніям покращувати зображення продуктів електронної комерції, автоматично видаляючи відволікаючий фон або покращуючи якість зображень з низьким роздільною здатністю.

1. **Основні етапи машинного навчання**

Основні етапи машинного навчання включають кілька ключових кроків, що забезпечують ефективне навчання моделі. Нижче наведені основні етапи:

1. **Збір даних**
Перший крок полягає в зборі відповідних даних для задачі, яку потрібно вирішити. Дані можуть бути зібрані з різних джерел, таких як бази даних, веб-сайти, сенсори або дані користувачів.
2. **Підготовка та обробка даних**
Зібрані дані часто містять пропущені значення, шум чи нерелевантні параметри, тому їх необхідно обробити: видалити аномальні значення, заповнити пропуски, нормалізувати або масштабувати дані для подальшого навчання.
3. **Вибір ознак**
На цьому етапі обираються значущі ознаки (параметри) для моделі, які найбільше впливають на результат. Цей процес допомагає зменшити обсяг даних, видаляючи зайві чи малозначущі ознаки.
4. **Розподіл даних на навчальну та тестову вибірки**
Дані розбиваються на дві вибірки: **навчальну**, яку використовують для навчання моделі, та **тестову**, яка дозволяє оцінити її точність. Іноді також виділяють **валідаційну вибірку** для налаштування параметрів моделі.
5. **Вибір і навчання моделі**
Обирається відповідний алгоритм машинного навчання (наприклад, лінійна регресія, дерева рішень, нейронні мережі) для вирішення задачі. Модель навчається на основі навчальної вибірки, шукаючи залежності між ознаками та результатом.
6. **Оцінка моделі**
Навчену модель перевіряють на тестових даних. Обчислюються метрики оцінки, такі як точність, середньоквадратична помилка, F-мір тощо, щоб визначити, наскільки добре модель прогнозує на нових даних.
7. **Тонке налаштування моделі**
Можливе вдосконалення моделі шляхом налаштування її гіперпараметрів, вибору нових ознак або заміни алгоритму. Тонке налаштування допомагає досягти кращих результатів на основі валідаційної вибірки.
8. **Розгортання моделі**
Після успішного навчання та оцінки модель можна інтегрувати у виробниче середовище, де вона буде використовуватися для прогнозування нових даних або автоматизації процесів.
9. **Моніторинг та оновлення моделі**
Після розгортання модель потребує постійного моніторингу, оскільки дані з часом можуть змінюватися. За необхідності модель перенавчається або налаштовується для підтримки її точності.

Кожен з цих етапів є важливим і сприяє тому, щоб модель машинного навчання працювала максимально ефективно в реальних умовах.

1. **Бібліотеки Python для машинного навчання**

**1. scikit-learn**

* **Опис**: Найпопулярніша бібліотека для класичного машинного навчання.
* **Особливості**: Включає алгоритми для класифікації, регресії, кластеризації, зменшення розмірності, оцінки моделей тощо.
* **Використання**: Легко інтегрується з іншими бібліотеками, такими як NumPy і pandas.

**2. TensorFlow**

* **Опис**: Відкрита бібліотека для глибокого навчання, розроблена компанією Google.
* **Особливості**: Підтримує нейронні мережі, конволюційні та рекурентні моделі. Має власну платформу для розгортання моделей на мобільних та веб-додатках.
* **Використання**: Має модуль Keras для спрощення створення моделей.

**3. Keras**

* **Опис**: Високоурівнева бібліотека для глибокого навчання, що працює поверх TensorFlow.
* **Особливості**: Простий і зрозумілий API для швидкої розробки нейронних мереж.
* **Використання**: Ідеальна для швидкого прототипування.

**4. PyTorch**

* **Опис**: Відкрита бібліотека для глибокого навчання, розроблена Facebook.
* **Особливості**: Інтуїтивно зрозумілий інтерфейс, динамічне обчислення графів, що робить його зручним для досліджень.
* **Використання**: Зростаюча популярність у академічному середовищі.

**5. pandas**

* **Опис**: Бібліотека для обробки та аналізу даних.
* **Особливості**: Забезпечує структури даних, які спрощують роботу з табличними даними.
* **Використання**: Важлива для попередньої обробки даних.

**6. NumPy**

* **Опис**: Бібліотека для наукових обчислень.
* **Особливості**: Забезпечує підтримку великих багатовимірних масивів і матриць, а також великий набір математичних функцій.
* **Використання**: Основна бібліотека для числових обчислень у Python.

**7. Matplotlib**

* **Опис**: Бібліотека для візуалізації даних.
* **Особливості**: Дозволяє створювати графіки, діаграми та інші візуальні елементи.
* **Використання**: Ідеально підходить для візуалізації результатів моделей.

**8. Seaborn**

* **Опис**: Бібліотека для візуалізації даних, побудована на основі Matplotlib.
* **Особливості**: Пропонує більш привабливі графіки та зручніші для користувача функції.
* **Використання**: Легко інтегрується з pandas для візуалізації даних.