Зміст

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Вступ | | |  |
| 1 | Сучасні технічні засоби класифікації зображень | | |  |
|  | 1.1 | Методи машинного навчання |  | |
|  | 1.2 | Сучасні технології машинного навчання |  | |
|  | 1.3 | Алгоритми машинного навчання |  | |
|  | 1.4 | Особливості обробки багатовимірних даних |  | |
|  | 1.5 | Практичні приклади застосування машинного навчання у роботи із зображеннями |  | |
| 2 | Програмні засоби для машинного навчання | | |  |
|  | 2.1 | Порівняльні характеристики мов програмування у галузі |  | |
|  | 2.2 | Фреймворки, бібліотеки та API для машинного навчання |  | |
|  | 2.3 | Порівняння моделей TensorFlow для виявлення об’єктів на зображенні |  | |
|  | 2.4 |  |  | |
|  | 2.5 |  |  | |
| 3 | Практична частина | | |  |
|  | 3.1 | Налаштування системи для машинного навчання |  | |
|  | 3.2 | Підготовка даних |  | |
|  | 3.3 | Підготовка конфігураційних файлів |  | |
|  | 3.4 | Навчання моделі |  | |
|  | 3.5 | Оцінка точності |  | |
|  | Основні результати та висновки | | |  |
|  | Література | | |  |

Машинне навчання (ML) - це, безперечно, одна з найпопулярніших тем у розробці програмного забезпечення. Машинне навчання відкриває багато нових можливостей для розробників, власників програм і кінцевих споживачів. Від більшої персоналізації до більш розумних рекомендацій, вдосконалених функцій пошуку, інтелектуальних помічників і додатків, які можуть бачити, чути та реагувати - технологія ML може покращити додаток і досвід його використання різними способами.

Машинне навчання є підмножиною штучного інтелекту (AI). Це дає комп'ютерам можливість навчатися на даних і поступово підвищувати якість результату на особливих завданнях - все це не покладаючись на програмування на основі чітких правил. Алгоритми машинного навчання знаходять природні закономірності всередині даних і приймають на їх основі майбутні рішення.

**2 ПРОГРАМНІ ЗАСОБИ ДЛЯ МАШИННОГО НАВЧАННЯ**

**2.1 Порівняльні характеристики мов програмування у галузі**

Від вибору мови програмування для подальшої розробки напряму залежать такі фактори, як швидкодія, доступність готових бібліотек та фреймворків, а також кількість матеріалів для навчання та отримання підтримки зі сторони спільноти розробників.

Наразі найбільшою популярністю користуються такі мови програмування як Python, C/C++, Java, R та JavaScript. Кожна з них має свої недоліки та переваги, а також деякі особливості застосування.

Згідно з даними "Developer Economics", які було зібрано з більш ніж двох тисяч анкет data scientists та machine learning розробників, дослідники порівняли перші п'ять результатів у рейтингу за кількома критеріями, розглянемо ці дані на рис.1.

Рис. 1 Дані дослідження "Developer Economics"

По-перше, розглянемо загальну популярність мов для машинного навчання. Python є лідером у списку, при цьому 57% data scientists і ML розробників використовують його, а 33% - надають йому пріоритет під час розробки. Така ситуація склалася насамперед через великий прогрес фреймворків глибинного навчання на Python за останні 5 років, включаючи випуск TensorFlow та широкий вибір інших бібліотек.

Python часто порівнюється з R, але вони не можуть бути порівнянні з точки зору популярності: R є четвертим у загальному використанні (31%) і п'ятим у пріоритизації (5%). R фактично є мовою з найнижчим співвідношенням пріоритизації до використання серед п'яти лідерів (17%). Це означає, що в більшості випадків R є додатковою мовою, а не основним вибором. Це ж співвідношення для Python становить 58%, що є найбільшим серед п'яти мов-лідерів, а це свідчить про те, що тенденції використання Python є абсолютно протилежними тенденціям використання R. Окрім цього, Python також є основним вибором для більшості розробників.

C/C++ є не дуже далекими від Python, як у популярності (43%), так і за пріоритетами (19%). Java слідує за C/C++ та має 41% та 16% відповідно. Водночас JavaScript є п'ятим у використанні, з невеликим відривом від R за пріоритизацією (7%) і за популярністю (28%).

Респондентів питали також про інші мови, що використовуються в машинному навчанні, включаючи Julia, Scala, Ruby, Octave, MATLAB і SAS, але всі вони мали показники нижче 5% за пріоритетностю і нижче 26% від використання, тож не були включені у дослідження.

Дані показують, що найбільш вирішальним фактором при виборі мови для машинного навчання є тип проекту - область застосування. У розробників питали про 17 різних областей застосування, а також надавали можливість респондентам розповісти, що вони все ще вивчають варіанти, не активно працюючи в будь-якій області. У кінцевому дослідженні пропонуються три найпопулярніші області для кожної мови: ті, де розробники надають пріоритети кожній мові найбільше і найменше.

Науковці з машинного навчання, що працюють над аналізом настроїв тексту, надають найбільші пріоритети Python (44%) і R (11%), а найменші - JavaScript (2%) і Java (15%), ніж розробники, що працюють в інших областях. На відміну від Python, Java, має високий пріорітет серед тих, хто працює над мережевою безпекою, захистом від кібер-атак і виявленням шахрайства, двома областями, де Python є найменш пріоритетним. Алгоритми захисту мережі та виявлення шахрайства побудовані або використовуються в основному у великих компаніях, особливо у фінансових установах, де Java є найпопулярнішою. У напрямках, які менш орієнтовані на великі підприємства, такі як обробка природної мови (NLP) та аналіз настроїв, розробники обирають Python, який пропонує простіший і швидший спосіб побудови високоефективних алгоритмів, завдяки великій колекції спеціалізованих бібліотек та швидкій точці входу.

Штучний інтелект (AI) в іграх (29%) та керування роботами (27%) - це дві області, в яких C/C++ використовуються найбільшою мірою, що обумовлено необхідним рівенем контролю, високою продуктивністю та ефективністю. В цих областях мова програмування більш низького рівня, як C/C++, яка поставляється з дуже складними бібліотеками AI, є звичайним вибором, тоді як R, призначений для статистичного аналізу та візуалізації, вважається в основному неактуальним. AI в іграх (3%) і керування роботами (1%) - дві області, де R має найменший пріоритет, подібно до розпізнавання усної мови (3%).

Окрім аналізу настроїв, R також користується відносно високим пріоритетом - у порівнянні з іншими областями застосування - у біоінженерії та біоінформатиці (11%), а Java має в цих напрямках найгірші показники (13%). Враховуючи давнє використання R у біомедичній статистиці, як всередині, так і за межами академічних кіл, не дивно, що вона є однією з областей, де R використовується найбільше. Також, дані дослідження показують, що розробники, які є новими у галузі машинного навчання та досі вивчають варіанти, надають JavaScript більших пріоритетів, ніж іншим мовам (11%), а Java найменших - (13%). У багатьох випадках це розробники, які експериментують з машинним навчанням, використовуючи API для третіх сторін у веб-додатках.

По-друге, в даній області застосування, професійна підготовка також має вирішальне значення у виборі мови для машинного навчання. Найбільший пріоритет Python приділяється тими, для кого наука про дані та машинне навчання є основною професією або галуззю (38%). Те ж саме не можна сказати про R, який в основному має високий пріоритет від аналітиків даних і статистиків (14%), оскільки мова була спочатку створена для цих областей, замінивши S.

Front-end веб-розробники розширюють своє використання JavaScript для машинного навчання, 16% визначають його пріоритетним для цієї мети, залишаючись у стороні від складних та не пристосованих для цього напрямку C/C ++ (8%).

Ситуація серед інженерів вбудованих обчислювальних апаратів та електроніки склалася навпаки, тут C/C++ надають більшу перевагу ніж іншим мовам, уникаючи при цьому JavaScript, Java і R. Враховуючи інвестиції розробників в освоєння C/C++, для них не має сенсу використовувати мову, яка погіршить їх рівень контролю над програмою. До того ж, інженери часто працюють над подібними до їх галузі навчальними проектами, такими як проекти аналітики IoT, де апаратні засоби можуть змусити їх обрати саме більш низькорівневу мову програмування.

Для Java, більшість складають розробники front-end комп`ютерних додатків, яких значно більше, ніж серед інших мов (21%). Розробники на підприємствах, як правило, використовують Java у всіх проектах, включаючи машинне навчання. Вплив компанії в цьому випадку також видно з третього фактора, який сильно співвідноситься з визначенням мовних пріоритетів - причиною для входу в машинне навчання. Найбільш поширеною причиною для вибору Java (27%) є пропозиція або вимога від компанії. Це найменш бажана мова (14%) серед тих, хто потрапив у ML тільки через власний інтерес, на відміну від Python (38%).

R обирають, у більшості випадків (7%), через те, що він використовується під час навчання в учбових закладах.

C/C++ є більш пріоритетним для тих, хто хоче покращити свої існуючі програми/проекти з машинним навчанням (20%) і менше для тих, хто сподівається створити нові висококонкурентні програми на основі машинного навчання (14%). Ці показники вказують на те, що C/C++ використовується в основному в інженерних проектах, а також у програмах IoT або AR/VR, які, швидше за все, вже написані на C/C++, до яких додаються функціональні можливості, що підтримуються ML. При створенні нової програми з нуля - особливо з використанням NLP для чат-ботів - немає особливої причини використовувати C/C++.

Нарешті, розробники, які потрапили в машинне навчання, щоб збільшити свої шанси на забезпечення високооплачуваних проектів, віддають перевагу JavaScript більше, ніж іншим мовам (8%).

Важливим фактором для вибору мови програмування є її швидкодія. Лідером у цьому питанні є C/C++. На другому місці перебуває Java. Ці мови є компільованими, тобто код зводиться у байт-код, перш ніж зберігатися як виконуваний файл. Натомість, R, JS та Python є інтерпретованими мовами, тобто код зберігається в тому ж форматі, в якому його було введено. Компільовані програми зазвичай працюють швидше, ніж інтерпретовані, оскільки інтерпретовані програми повинні переводитися у машинні команди кожного разу, коли вони виконуються. Проте для Python існує розширення Cython - яке, по суті, еквівалентне Python зі статичним типізацією, що дозволяє розробникам значно збільшити швидцість виконання програми.

Найбільша кількість бібліотек та пакетів інструментів насьогодні є у Python та R. Python також є мовою з найменшою точкою входу, тобто для написання програм на Python не потрібно глибоких знань цієї мови, що робить її найкращим вибором для початківця у ML. У цьому проекті також використано Python.

**2.2 Фреймворки, бібліотеки та API для манинного навчання**

Фреймворки машинного навчання постійно розвиваються. Штучний інтелект у поєднанні з правильною системою глибокого навчання збільшив загальний розмір того, чого можуть досягти різні підприємства, не виходячи за межі своїх предметних областей. Крім того, з постійно зростаючою кількістю організацій, які сподіваються розширити спектр своїх послуг та можливостей, необхідно щоб будь-яка така компанія асимілювала як машинне навчання, так і прогностичні дослідження. Розглянемо найполпулярніші фреймворки для машинного навчання.

*TensorFlow*

TensorFlow — це відкрита програмна бібліотека для машинного навчання, здатна вирішувати цілу низку задач, розроблена компанією Google для використання всередині компанії, до того, як його було випущено як відкрите програмне забезпечення 9 листопада 2015 року. Він може працювати на декількох центральних та графічних процесорах (включно з додатковими розширеннями CUDA для обчислень загального призначення на графічних процесорах).

Рис. 2 Компанії, що застосовують TensorFlow

В даний час TensorFlow займає перше місце в списку фреймворків для машинного навчання. Більшість розробників використовують TensorFlow, оскільки він має велику спільноту підтримки і багато вбудованих функцій. Цей фреймворк використовується такими компаніями, як Google, Airbus, Airbnb, Coca-Cola, AMD, Nvidia, IBM та іншими (рис. 2), здебільшого через надзвичайно адаптовану інженерну структуру.

TensorFlow впроваджує різноманітні інструменти, які широко використовуються. TensorBoard використовується для візуалізації інформації щодо системи, він демонструє процеси тренування та оцінки моделі, служить для швидкого оформлення нових розрахунків та тестів.

TensorFlow забезпечує прикладний програмний інтерфейс для Python, а також для C++, Haskell, Java та Go, а також доступний для 64-розрядних Linux, macOS, Windows, та для мобільних обчислювальних платформ, включно з Android та iOS.

*Caffe*

Caffe — це система для глибинного навчання, первинно розроблена Янці Дзя, як частина його докторської праці в Каліфорнійському університеті в Берклі, наразі у цього проекту багато учасників, і його розміщено на GitHub. Вона є відкритою, з ліцензією BSD та написана мовою C++ з інтерфейсом для Python.

Caffe підтримує багато різних типів архітектур глибинного навчання, орієнтованих на класифікацію та сегментування зображень. Вона підтримує конструкції згорткових та рекурентних нейронних мереж, а також довгої короткочасної пам’яті та повноз'єднаних нейронних мереж. Caffe підтримує прискорення на основі графічного процесора із застосуванням Cudа Nvidia.

*Torch*

Torch — відкрита бібліотека для машинного навчання, система для наукових обчислень та мова сценаріїв на основі мови програмування Lua. Пропонує широкий спектр алгоритмів для глибинного машинного навчання і використовує мову сценаріїв LuaJIT та реалізацію мовою C в основі.

Основним пакетом бібліотеки Torch є одноіменний компонент torch. Він забезпечує гнучкий N-вимірний масив, або тензор, який підтримує основні процедури для індексування, розшаровування, транспозиції, приведення типів, зміни розмірів, розподілення зберігання та клонування. Цей об'єкт використовується більшістю інших пакетів, і відтак є центральним об'єктом бібліотеки. Тензор також підтримує математичні операції, такі як max, min, sum, статистичні розподіли, такі як рівномірний, нормальний та поліноміальний, та операції основних підпрограм лінійної алгебри, такі як скалярний добуток, матрично-векторне множення, матрично-матричне множення, матрично-векторний скалярний добуток та матричний скалярний добуток.

У складі бібліотеки також є пакет nn, який застосовують для побудови нейронних мереж. Його розділено на модульні об'єкти, які мають спільний інтерфейс Module. Модулі мають методи, які дозволяють їм виконувати пряме та зворотне поширення. Модулі можна з'єднувати, застосовуючи модулеві компонувальники, щоб створювати складні, підігнані під задачу графи. Простіші модулі, такі як Linear, Tanh і Max, складають модулі основних складових. Цей модульний інтерфейс забезпечує автоматичне диференціювання градієнтів першого порядку.

Torch використовує група дослідження штучного інтелекту компанії Facebook, IBM, Яндекс та Дослідницький інститут Ідіап. Torch було розширено для використання під Android і iOS. Його використовували для відтворення апаратних реалізацій потоків даних, подібних тим, що відбуваються у нейронних мережах.

*Theano*

Theano - це ще одна популярна бібліотека Python для чисельних обчислень і схожа на NumPy. Theano дозволяє ефективно визначити, оптимізувати та оцінити математичні вирази, що включають багатовимірні масиви.

Те, що відрізняє Theano, полягає в тому, що він користується перевагами GPU комп'ютера. Це дає змогу робити обчислення інтенсивних даних до 100 разів швидше, ніж при роботі на CPU. Швидкість Теано робить її особливо цінною для глибокого навчання та інших складних завдань.

Остаточний випуск бібліотеки Theano минулого року - 2017, версія 1.0.0 з великою кількістю нових можливостей, змін інтерфейсу та поліпшень.

Sources

1) "Speed/accuracy trade-offs for modern convolutional object detectors."

Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z,

Song Y, Guadarrama S, Murphy K, CVPR 2017