УДК 004.853, 004.55 https://doi.org/10.15407/pp2018.04.045

І.Ю. Гришанова, Ю.В. Рогушина

ТЕХНОЛОГІЧНІ РІШЕННЯ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ BIG DATA. МОВИ ПРОГРАМУВАННЯ

Розглянуто проблеми, що виникають в процесі застосування методів аналізу даних до Big Data. Проаналізовано сучасні мови програмування з точки зору ефективності їх застосування для розробки засобів машинного навчання (ML – Machine Learning) та прикладних програм, орієнтованих на Big Data. Проаналізовано основні типи задач машинного навчання, пов'язаних із здобуттям з Big Data відомостей, корисних для практичного застосування. Цей аналіз показує, що ці задачі вирішуються з використанням методів статистичної обробки та навчання нейромереж. Тому в програмних засобах, орієнтованих на рішення цих задач, доцільно мати відповідні бібліотеки. Наявність великого різноманіття алгоритмів ML, орієнтованих на різні типи вхідної інформації та знань, що за ними будуються, свідчить про потреби в спеціалізованих бібліотеках машинного навчання, які реалізують ці алгоритми. Ще одним важливим фактором для вибору інструментального середовища, в якому задачі ML вирішуються для Big Data, є швидкість обробки: це пов'язано з великими обсягами тих даних, що мають оброблятися. Зовнішні сервіси для ML та обробки Big Data, створені Google, Amazon тощо, значно спрощують процес розробки засобів інтелектуального аналізу даних для тих мов програмування, що підтримують використання таких сервісів. Таким чином, для створення експериментальних прототипів, що поєднують сучасні підходи до машинного навчання з елементами штучного інтелекту (ШІ), найбільш придатною мовою програмування є Python. Цей висновок підтверджують і світові результати опитувань розробників у сфері Data Sciences. Але інші мови програмування, проаналізовані в даній роботі, можуть бути навіть більш корисними за певних додаткових умов: приміром, C++ – для розробок, орієнтованих на специфічне апаратне або програмне забезпечення, а Java та Scala – для створення корпоративних застосувань.

Ключові слова: Big Data, інтелектуальний аналіз даних, машинне навчання

Вступ

Зростання обсягів інформації, що збирається у сучасному світі, і вимоги до її обробки і збереження роблять актуальним дослідження в області методів і алгоритмів аналізу великих наборів даних. Поки межею можливостей сучасних програмних застосувань, орієнтованих на обробку великих обсягів даних, є петабайтні набори і гігабайтні потоки даних. Але відповідно до тенденції розвитку науки і суспільства очікуються ще більші масштаби й обсяги даних. Подальше зростання кількості інформації та ускладнення її структури робить все більш актуальними проблеми машинного навчання (ML) та інтелектуального аналізу даних (Data Mining), які дозволяють здобувати з Big Data корисні та практично застосовні факти та знання.

Термін «великі дані» (Big Data) був введений у 2008 р. К. Лінчем – редактором журналу Nature [1]. Саме у цей час обсяги даних, що генерувалися в електронному вигляді, викликали потребу в розробці відповідної інфраструктури для їх збереження та обробки. Big Data – це дані, для яких характерні властивості, які називають «три V» – обсяг (volume), швидкість (velocity) і розманіття (variety). Відповідно до цих властивостей випливають проблеми: наявність великого обсягу є проблемним для засобів обробки; необхідність швидкої обробки та високе різноманіття форматів подання даних викликає складності аналізу й обробки.

Великі дані – це об'ємні, високошвидкісні та різноманітні інформаційні активи, які потребують економічно ефективних, інноваційних методів обробки інформації для поглибленого розуміння та прийняття рішень [2].

У роботі [3] висловлена гіпотеза про те, що виявлення закономірностей у великих масивах даних стає одним з інструментів дослідження й одним з методів здобуття нових знань у сучасних умовах. Якщо раніше поява нових фактів легко фіксувалася і ставала предметом дослідження, то в даний час проблемою є знаходження таких нових фактів у великих масивах даних і їх формалізація.

Аналіз даних великого обсягу потребує створення технологій та засобів реалізації високопродуктивних обчислень [4]. Програми, які орієнтовані на обробку великих обсягів, працюють з файлами даних обсягом від декількох терабайт до петабайта. На практиці ці дані надходять у різних форматах і часто бувають розподілені між декількома джерелами збереження інформації та неструктуровані або слабоструктуровані. Особливо слід виділити множинність форматів даних та їх неструктурованість або слабоструктурованість, що само по собі створює проблеми навіть при не дуже великому обсязі.

Data Mining

Data Mining – технології аналізу даних у базах або сховищах даних, яка базується на статистичних методах та призначена для виявлення заздалегідь невідомих закономірностей, а також для підтримки прийняття стратегічно важливих рішень. Ці технології спрямовані на одержання знань з даних.

Для *виявлення* (discovery) знань з даних використовують автоматизовані методи та засоби обробки інформації. За наявності даних великого обсягу саме цей напрямок забезпечує отримання нової та корисної інформації. Для виявлення знань застосують методи машинного навчання, які призначені для інтелектуального аналізу даних – Data Mining (більш детальний огляд задач Data Mining наведено в [5]). В Data Mining застосовують методи машинного навчання, призначені для використання отриманого раніше досвіду для вдосконалення подальшої поведінки комп'ютерної системи та методи статистичного аналізу. Машинне навчання базується на прикладній статистиці.

Сучасні методи інтелектуального аналізу даних є результатом еволюції підходів у двох напрямках: з одного боку – це інтелектуалізація статистичних методів, з іншого – побудова штучних систем за аналогією з біологічними об'єктами, які називають штучними нейронними мережами.

Глибоке навчання (Deep Learning) є окремим випадком машинного навчання. Воно дозволяє знайти рішення для таких проблем, як розпізнавання зображень та мовлення, переклад природної мови тощо, що не були задовільно вирішені за допомогою традиційних алгоритмів машинного навчання. Процес глибокого навчання по суті є процес побудови багатошарової нейронної мережі на основі стохастичного градієнтного спуску. Для її побудови потрібна велика кількість вхідної інформації, і тому таке навчання може ефективно використовуватися для обробки Big Data. Кожен наступний шар отримує на вході вихідні дані попереднього шару. При цьому ознаки організовані ієрархічно, ознаки більш високого рівня є похідними від ознак більш низького рівня. Це дозволяє вирішувати проблему обробки надто великого простору ознак, але оскільки внутрішні шари сховані, тому пояснення результатів глибокого навчання користувач не отримує.

Нині технології машинного навчання широко застосовуються провідними розробниками програмного забезпечення. Приміром, компанія Google активно використовує ці технології, що дозволяє значно покращити функції його сервісів: розпізнавання мовлення в Google Now; машинний переклад, який використовує не тільки правила граматики, а також і попередній досвід; автоматичні короткі відповіді Smart Reply тощо.

Використання машинного навчання та методів ШІ значно підвищує якість роботи програм, тому, створюючи засоби інтелектуальної обробки даних, необхідно передбачати підтримку в них можливостей машинного навчання. Обираючи інструментальні засоби для розробки проекту, необхідно враховувати наступні фактори, які значним чином визначають ефективність їх застосування до даного класу задач:

- наявність бібліотек для статистичної обробки (саме на статистичних методах базується основна частина ML);
- наявність спеціалізованих бібліотек для побудови різноманітних нейронних мереж (для підтримки глибокого навчання);
- наявність бібліотек машинного навчання (це значно прискорює реалізацію

прикладних задач класифікації, кластеризації, прогнозування тощо) та реалізації алгоритмів ШІ;

- швидкість роботи (обробка Big Data потребує високопродуктивних обчислень);
- можливість обробки різних типів даних (Big Data можуть бути представлені у найрізноманітніших форматах) та неструктурованої та слабоструктурованої інформації;
- зручність застосування та розвинута інфраструктура інструментального середовища;
- підтримка спільноти користувачів та розробників, наявність вичерпної і актуальної документації та навчальних курсів.

Мови програмування, що використовуються для ML в Big Data

Аналіз публікацій у Web показує, що на даний час для масштабованої обробки даних у більшості випадків використовують наступні мови програмування: R, Python, Scala і Java. Деякі джерела включають у цей перелік ще мови, орієнтовані на розробки в галузі штучного інтелекту (ШІ) – Lisp і Prolog, та універсальні мови широкого вжитку – C/C++, PHP і навіть скриптову мову Javascript.

Особливості розробки застосувань з ML

В програмних застосуваннях з машинного навчання, задачі тренування та оперування (або виведення) *розділені*. Таким чином, для розробки застосування з тренування можливо використовувати одну мову, а для виведення – іншу. Це дозволяє підвищити операційну складність розроблюваних програм. Крім того, деякі мови мають більш швидкі бібліотеки для виконання окремих задач, і існує можливість їх використання через API. Таким чином, однією з важливих вимог є швидкість виконання.

Наступною характеристикою, яку виділяють спеціалісти з ML та Big Data, є *типізація змінних*. Можливість не вказувати явно тип змінних підвищує гнучкість програмного забезпечення (ПЗ), але це також підвищує шанс отримання помилок. Деякі спеціалісти вважають, що мови програмування такі, як Python, що мають динамічне виділення пам'яті (не типізовані змінні), не підходять для машинного навчання.

Для оцінювання придатності мови програмування для роботи з великими даними та реалізації функцій глибинного машинного навчання, визначимо вимоги, яким вона повинна відповідати.

1. Можливості роботи з масштабованими масивами даних.
2. Можливості роботи з існуючими бібліотеками ШІ.
3. Підтримка роботи з Deep Neural Networks (DNNs) або Deep Learning Frameworks.
4. Швидкість роботи.
5. Технологічна зрілість мови (наявність професійної спільноти, специфічних бібліотек, зручного інструментарію для розробки та тестування, можливості створення графічного інтерфейсу користувача).
6. Складність вивчення.
7. Економічна ефективність.

Розглянемо докладніше сформульовані вимоги. Зрозуміло, що для роботи з великими масивами даних, необхідно мати відповідний інструментарій, який дозволяє масштабування і не залежить від розміру, формату та місця зберігання. Крім того очевидно, що складні та вибагливі до швидкості та використання пам`яті алгоритми ШІ, роботи з натуральними мовами, неструктурованими даними, статистичні операції агрегування тощо мають бути реалізовані з максимальною оптимізацією.

Оскільки в більшості випадків досить просто і оптимально передавати процеси машинного навчання для виконання їх вже готовими нейромережами Deep Neural Networks (DNNs) або фреймворками Deep Learning Frameworks, вимога сумісності, наявності API, можливості працювати з відлагодженими алгоритмами DNNs та DLF є дуже важлива.

На даний час створено вже досить багато потужних багатофункціональних сервісів. У листопаді 2015 року корпорація Google запустила сервіс TensorFlow – без-

коштовне програмне забезпечення, яке має вільну ліцензію Apache 2.0. Це ПЗ добре відповідає сучасним вимогам машинного навчання і може використовуватися для проектів, пов`язаних з Deep Learning. Ця система – досить гнучка як для проведення досліджень, так і для використання машинного навчання в існуючих програмних продуктах для розширення можливостей.

TensorFlow – це відкрита програмна бібліотека для машинного навчання цілій низці задач, розроблена компанією Google для задоволення її потреб у системах, здатних будувати та тренувати нейронні мережі для виявлення та розшифровування образів та кореляцій, аналогічно до навчання й розуміння, які застосовують люди [6].

Для поширення ідей машинного навчання, компанія Google розробила курс відео-занять, де просто і доступно викладені принципи роботи з ПЗ Deep Learning, що представлено компанією для вільного використання (https://proglib.io/p/google-ml-recipes/). Розробники ПЗ можуть вільно використовувати цей сервіс: будувати мережі, завантажувати свої дані, робити їх аналіз тощо.

На початку 2018 року Google представив новий продукт: Firebase Machine Learning Kit – набір бібліотек з машинного навчання. Цей набір дозволяє ефективно використовувати можливості машинного навчання в мобільних застосуваннях для Android та iOS. Бібліотека Firebase Machine Learning Kit дозволяє розробникам легко та з мінімальним кодом використовувати усі можливі високоточні, попередньо навчені глибокі моделі в своїх мобільних застосуваннях. Більшість моделей доступні як локально, так і в Google Cloud. На даний час моделі обмежені задачами, що пов`язані з комп'ютерним баченням, розпізнаванням обличь у реальному часі, оптичним розпізнаванням символів, скануванням штрих-кодів та виявленням об'єктів.

Крім Google, свій набір сервісів з машинного навчання випустили й інші лідери галузі ІТ: корпорація Амазон надало набір хмарних сервісів Amazon Machine Learning з докладною документацією з основ та прикладів використання машинного навчання (https://aws.amazon.com/ru/machine-learning/); Microsoft також запропонували Microsoft Azure Machine Learning – набір сервісів з машинного навчання (https://azure.microsoft.com/en-us/services/machine-learning/).

За наведеними в https://proglib.io/p/fast-machine-learning/ результатами тестування (на 2017 рік), використання Microsoft Azure Machine Learning значно перевищує Amazon: набагато швидше виконується ітерація, є можливість додавати/видаляти стовпці без необхідності повторного завантаження файлів. В наведеному прикладі, процес навчання, який на Amazon займає 11 хвилин, на Azure триває всього 23 секунди. Крім того зазначається, що в Microsoft Azure Machine Learning більш зручний інтерфейс. Однак, ця галузь дуже швидко розвивається і ситуація змінюється.

Проведене в травні 2018 року опитування серед профільних спеціалістів Інтернет-видання KDNuggets (https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html), що спеціалізується на Deep Learning, Machine Learning, Big Data, навпроти, показало падіння задоволеністю сервісами Microsoft, IBM, та зростання популярності сервісів від Google у порівнянні з 2017 роком.

Окрім вищезазначених сервісів від Microsoft, Google, IBM та Amazon, виділяють наступні популярні фреймворки:

- DIGITS: Програма Deep Learning візуалізації від NVIDIA. Вона надає графічний інтерфейс GUI для побудови та ревізії (reviewing) мережі DNNs (https://developer.nvidia.com/digits).

- Caffe: Фреймворк написаний на C++, він надає інтерфейс для Python (http://caffe.berkeleyvision.org/). Для початку роботи з ним не має необхідності нічого кодувати. Це зроблено дякуючи тому, що Caffe може бути сконфігуровано з Google Protobuf, текстовим форматом типу JSON (https://developers.google.com/protocol-buffers/docs/overview).

- Theano: Більшість Deep Learning Frameworks подібні одна до одної і мають досить несуттєві розбіжності, водночас як Theano (http://deeplearning.net/software/theano/) є однією з таких, що синтаксично відрізняється від Caffe. Theano – це бібліотека символьної математики, над функціями якої побудовані пакети, що реалізують можливості DNN.

- Torch: Torch (http://torch.ch/) використовується Facebook для дослідження DNN, він підтримується Google, NVIDIA, та іншими провідними компаніями, що займаються Deep Learning. Він використовує мову програмування Lua (http://www.lua.org/).

- TensorFlow: як Torch – інструмент Facebook, так TensorFlow (https://www.tensorflow.org/) – головний інструмент Google. Як більшість проектів Google, він базований на Python.

- cuDNN: це бібліотека NVIDIA (https://developer.nvidia.com/cudnn) для розпаралелювання навчання мережі DNN на графічних процесорах (GPU). Використання розпаралелювання на процесорах GPU надало величезний поштовх для розвитку нейромереж, надаючи можливість суттєвого пришвидшення процесу навчання.

- Convolutional Neural Networks (CNNs) (https://en.wikipedia.org/wiki/Convolutional_neural_network) – це набір блоків нейромережі DNN, що реалізує навчання на маленьких фрагментах зображення, які потім поєднуються з сусідніми, і так далі допоки не отримується початкове зображення. Таке розбиття на блоки знижує складність процесу навчання, що особливо важливо для обробки великих зображень.

Найбільш поширеними фреймворками на даний час вважаються Caffe та Tensorflow. Caffe підтримує Python та Matlab. Tensorflow підтримує Python та R. Також необхідно зазначити, що більшість менш популярних DNN фреймворків (типу Theano), беззаперечно надають підтримку мові Python. Це надає неабияку фору цій мові серед інших.

Важливим параметром у виборі мови програмування для глибокого навчання є швидкість виконання обчислень. Мова R була побудована як мова для статистичних обчислень, отже вона має вбудовану підтримку статистичної обробки та аналізу даних. За рахунок цих вбудованих функцій, R є більш швидкою для виконання статистичних задач. На противагу, Python використовує бібліотеки та фреймворки, що підключаються і тому є повільнішим.

Дуже важливим параметром є технологічна зрілість та поширеність мови програмування та інфраструктури. Як зазначалося, R найбільш підходить для статистичного аналізу. Python більше підходить для різних задач генерування: препроцесінг даних, пост-обробка результатів. Крім цього, Python є більш зручним для випадків, коли необхідна інтеграція машинного навчання з іншим програмним забезпеченням.

Підтримка співтовариства (Community Support). Враховуючи стрімкий розвиток технологій, коли нові релізи компіляторів, трансляторів, бібліотек, API виходять швидше, ніж створюється докладна документація, дуже важливо мати підтримку в певній професійній спільноті, що займається конкретним проектом. Більш поширені мови мають більші спільноти, водночас як рідкі екзотичні мови мають доволі обмежені співтовариства.

Складність початкового вивчення. Мова R є більш функціональна, водночас як Python – більш об`єктно-орієнтована. Отже, якщо ви знайомі з об`єктно-орієнтованим програмуванням, вивчення Python буде легшим, ніж R, і навпаки, при досвіді в функціональному програмуванні, R буде зручнішим. Таким чином, вибір мови суттєво залежить від попереднього досвіду розробника.

Не останнє місце займає параметр економічної ефективності. Як приклад, Matlab є комерційним проектом, використання якого потребує покупки ліцензії. Водночас, більшість мов є вільними проектами з відкритим кодом і не потребують оплати за їх використання. Однак, можли-

во, використання певних бібліотек, сервісів або фреймворків, теж може бути платним.

Мова R – інтерпретована об'єктно-орієнтована мова програмування високого рівня, орієнтована на виконання статистичних обчислень, аналізу та зображення даних у графічному вигляді, а також програмне середовище розробника або дослідника даних [7].

R – об'єктно-орієнтована мова програмування. Це означає, що теоретично будь-що може бути збережене як об'єкт R. Кожен об'єкт має свій клас, який описує, що містить цей об'єкт і, що кожна функція може з цими даними робити.

R підтримує широкий спектр статистичних і чисельних методів. Її можна розширювати за допомогою пакетів, які по суті є бібліотеками та призначені для підтримки специфічних функцій і для роботи у спеціальних областях застосування. Базовий набір R містить основний набір пакетів, але станом на 2013 рік доступно більш 4000 спеціалізованих пакетів.

R створювалася під впливом мови програмування S з семантикою успадкованою від Scheme. Незважаючи на певні принципові відмінності, більшість програм мовою S можуть працювати також в середовищі R. Її назва походить від першої літери імен її розробників – Роса Іхаки та Роберта Джентлмена (Оклендський Університет, Нова Зеландія).

Ще однією особливістю мови R є графічні можливості, що полягають у підтримці створення якісної графіки, яка може включати математичні символи.

R і її пакети, поширюються через CRAN (Comprehensive R Archive Network).

R розповсюджується безкоштовно за ліцензією GNU (General Public Licence) у вигляді вільно доступного вихідного коду або відкомпільованих бінарних версій для таких операційних систем, як Linux, FreeBSD, Microsoft Windows, Mac OS X, Solaris.

R використовує текстовий користувацький інтерфейс, однак існують різні графічні інтерфейси користувача, більшість з яких є комерційними.

R має потужні можливості для здійснення статистичного аналізу: ця мова підтримує такі широко вживані методи Data Mining, як лінійна і нелінійна регресія, аналіз часових рядів (серій), кластерний аналіз, а також класичні статистичні тести і багато іншого.

Більша частина стандартних функцій R написана мовою R, однак існує можливість підключати код, написаний на C та C++.

R – інтерпретована мова програмування. Це впливає на швидкість її роботи, але спрощує процес розробки прототипів застосувань.

R – це найбільш потужний безкоштовний програмний інструмент з дуже широким набором статистичних бібліотек. В 2013 році R став самим широко використовуваним в науковій літературі пакетом для статистичного аналізу. На сьогодні R фактично є стандартом в розробці застосувань в галузі статистики.

R надає дуже потужний і швидкий механізм для аналізу даних, однак орієнтація на математичну статистику і незвичний принцип програмування ускладнюють її вивчення, а суто академічні розробки не дозволяють проводити широкі практичні застосування та розробляти великі комерційні проекти. Існує багато онлайн курсів з вивчення цієї мови, в 2016 році вийшов безкоштовний курс на платформі Prometheus «Аналіз даних та статистичне виведення на мові R (https://courses.prometheus.org.ua/courses/IRF/Stat101/2016_T3/about).

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня із строгою динамічною типізацією [8], розроблена Гвідо ван Россумом в 1990 році. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі для всіх основних платформ.

Python підтримує різні парадигми програмування, у тому числі процедурну, функціональну, об'єктно-орієнтовану та аспектно-орієнтовану. Мова містить структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням.

Використання Python надає розробникам ПЗ в сфері ML наступні переваги:

- широкий набір бібліотек, які реалізують машинне навчання, алгоритми ШІ, роботи з нейромережами, сервісами крупних постачальників послуг DNNs та DLF;
- можливість використання в діалоговому режимі, що корисно для проведення експериментів з ML;
- наявність бібліотеки для семантичної обробки текстових документів. Бібліотека Gensim [9] – інструментарій з відкритим кодом для векторно-просторового та тематичного моделювання, який підтримує аналіз текстових документів для видобування семантичної структури, масштабовану статистичну семантику та пошук семантично схожих документів;
- велика кількість додаткових модулів широкого призначення (приміром, для створення графічного інтерфейсу);
- підтримка виконання математичних задач (обробка комплексних чисел, цілих чисел довільної величини);
- переносимість програм;
- потужне середовище розробки IDLE, що входить у стандартний дистрибутив;
- відкритий код (можливість редагувати його іншими користувачами).

До основних недоліків Python відносять низьку швидкодію, що може стати критичним при обробці великих обсягів інформації, та відсутність статичної типізації. Але слід відмітити, що бібліотека Gensim, яка ефективно реалізує ряд моделей для тематичного моделювання: латентно-семантичний аналіз, латентне розміщення Дірихле та ієрархічні процеси Дірихле, і за останні роки стала однією із стандартних бібліотек для моделі word2vec, після декількох етапів доопрацювання, нині працює швидше за первинний код, який написаний на C.

Java – універсальна об'єктно-орієнтована мова програмування, що сьогодні є однією з найпоширеніших [10]. Перша версія Java була розроблена Sun Microsystems (згодом у 2009 році компанія була поглинена Oracle) в 1996 році. Поточною версією (вересень 2018 року) є Java 11. Java застосовується для створення різноманітного програмного забезпечення: десктопних застосунків, Web-порталів, сервісів тощо, що можуть виконуватися на різних типах пристроїв: звичайних ПК, планшетах, смартфонах.

Синтаксис Java подібний до C/C++ і C#. Ключовою особливістю Java є те, що програмний код спочатку транслюється в спеціальний байт-код, який не залежить від платформи, а після цього виконується віртуальною машиною JVM (Java Virtual Machine), що працює на конкретному комп`ютері. У цьому полягає принципова відмінність Java від таких інтерпретованих мов, як PHP чи Perl.

Така архітектура є основою кроссплатформеності і апаратної переносимості програм на Java без перекомпіляції на різні платформи (Windows, Linux, Mac OS тощо), для яких існує своя реалізація віртуальної машини JVM.

Недоліком Java є відносно низька швидкодія: програми, що виконуються віртуальною машиною Java, працюють повільніше за скомпільований машинний код (наприклад, написаними на C++, який при компіляції часто оптимізується під виконання на певному програмному і апаратному забезпеченні). Проте за останнє десятиріччя розробники віртуальної машини значно пришвидшили цей процес, тому нині програми на Java не надто поступаються аналогам на C++. Крім того, вважається, що програмування на Java – досить складний процес, кількість кодів для промислової програми завелика.

Java підтримує поліморфізм, спадкування, статичну типізацію. Об’єктно-орієнтований підхід дозволяє вирішити задачі з побудови великих, але гнучких, масштабованих і розширюваних додатків. Підтримується великий набір спеціалізованих бібліотек (приміром, Weka, Mahout) [11].

Weka [12] – середовище для розробки методів машинного навчання і застосування їх до реальних даних. Це вільне програмне забезпечення, написане на Java. Воно надає прямий доступ до бібліотеки алгоритмів.

MOA (Massive On-Line Analysis) [13] – бібліотека з відкритим кодом, на-

писана на Java та призначена для машинного навчання і збору інформації в потоках даних у режимі реального часу. Вона містить набір алгоритмів машинного навчання: для регресійного аналізу, класифікації, виявлення аномалій, кластеризації, рекомендаційних систем, та інструменти для оцінки їх за часом і в питанні використання пам'яті.

Deeplearning4 (https://deeplearning4j.org/) – бібліотека з відкритим кодом, написана на Java і Scala, призначена для об'єднання глибоких нейронних мереж і глибокого навчання для бізнес-середовища. Бібліотека дозволяє працювати з Hadoop, вирішуючи задачі розпізнавання мовлення і тексту, для виявлення аномалій у даних часових рядів.

MALLET (MAchine Learning for Languag Toolkit) (http://mallet.cs.umass.edu/) – бібліотека з відкритим кодом, розроблена для обробки тексту з застосуванням машинного навчання. Вона підтримує статистичну обробку природної мови, кластеризацію, класифікацію документів, інформаційний пошук, моделювання тощо. В бібліотеці реалізовано широкий спектр алгоритмів ML – наївний байєсівський алгоритм, дерево прийняття рішень, метод максимуму ентропії тощо.

ELKI (Environment for Developing KDD-Applications Supported by Index Structures) (https://elki-project.github.io/) – бібліотека, що забезпечує середовище для розробки KDD (Knowledge discovery in databases – здобуття знань з баз даних), що підтримуються індексними структурами. Бібліотека орієнтована на неконтрольовані методи в кластерному аналізі і виявлення аномалій. Для досягнення високої продуктивності і масштабованості ELKI пропонує структури індексації даних, такі як R*-дерева (такі структури застосовуються для індексації просторової інформації), що можуть забезпечити значне збільшення продуктивності.

Scala – мультипарадигмова мова програмування, що поєднує властивості об'єктно-орієнтованого та функціонального програмування [14]. Вона створена на початку 2000-их років у Федеральній політехнічній школі міста Лозанна (у Швейцарії). Багато концепцій Scala запозичено з Java і C#.

Scala сумісна із існуючими програмами мовою Java, тобто код Scala може викликатися із Java-програм і навпаки. Програми Scala виконуються на віртуальній машині Java (JVM) за умови приєднання до дистрибутиву файлу scala-library.jar. Інтеграція з Java дозволяє компілювати код, написаний на Scala, для JVM, а також використовувати всі java-бібліотеки. Scala поєднує статичну типізацію, об'єктно-орієнтоване програмування і функціональний підхід. Основна відмінність Scala від Java – наявність лямбда-виразів, монад та інших елементів функціонального програмування.

Для вивчення Scala рекомендується вивчити спочатку Java, тому що ці мови програмування часто перетинаються між собою і використовують загальні технології. Навчитися на Scala досить складно, але це пов'язано скоріше з тим, що на ній потрібно вирішувати більш складні задачі, пов'язані з Big Data і великими системами.

Переваги Scala: підтримка функціонального програмування та синтаксис простіший, ніж у Java. Недоліки: важкий для розуміння код, повільна робота компілятора.

За відгуками спеціалістів, що працюють зі Scala, це досить потужна і зручна мова, вона містить багато корисних функцій програмування, таких як співставлення з зразками (патернами проектування) і потребує значно менше коду, ніж стандартна Java.

Зазначимо, що Hadoop MapReduce, HDFS написано на Java. Storm, Kafka та Spark працюють на JVM (в Clojure та Scala).

C++ – компільована мова програмування високого рівня, розроблена Б. Страуструпом в 1979 році, що підтримує кілька парадигм програмування: об'єктно-орієнтовану, узагальнену та процедурну [15]. C++ базується на мові C. У 1990-х роках C++ стала однією з найуживаніших мов програмування загального призначення.

C++ використовується для системного програмування, розробки програмного забезпечення, драйверів, потужних серверних та клієнтських програм, а також для створення відеоігор. Можливості цієї мови дозволяють програмувати на низькому рівні, працювати з пам'яттю, адресами, портами (на апаратному рівні – hardware) та досягати оптимізації та швидкодії за рахунок використання низькорівневих функцій. Програми на C++ розробляють для різних платформ і систем з використанням особливостей конкретного апаратного забезпечення.

Головна перевага C++ – швидкість виконання: програма перетворюється в код, який при компіляції оптимізується під конкретне апаратне забезпечення. На цій мові створюють апаратно-прив`язане ПЗ, критичне за швидкістю.

На C++ написано багато надшвидких бібліотек, які реалізують генетичні алгоритми, нейронні мережі, обробку сигналів реального часу та критичних систем.

До явних недоліків C++ можна віднести необхідність дрібного кодування низького рівня, що ускладнює розробку великих систем.

LISP (LISP, від англ. LISt Processing – «обробка списків») – сімейство високорівневих мов програмування загального призначення, що базуються на представленні програми системою лінійних списків символів. LISP розроблено в кінці 1950-их у Масачусетському Технологічному Інституті для дослідження проблем штучного інтелекту та для рішення задач не чисельного характеру. Ця мова вважається другою після Fortran найстаршою високорівневою мовою програмування.

LISP орієнтовано на обробку символьної інформації. Ця мова дуже зручна для розробки лінгвістичних програм, особливо для обробки природномовних текстів. Розробники використовують LISP для багатьох класичних проектів ШІ. У вигляді списків зручно представляти алгебраїчні вирази, графи, множини, правила виведення і багато інших складних об'єктів. LISP підтримує символьне програмування, має швидкий інструментарій з прототипування, широкі можливості з розширювання і багато варіантів трансляторів.

Prolog (від "PROgramming in LOGic") – декларативна мова логічного програмування загального призначення, розроблена в 1972 році Аланом Кольмерое та Філіпом Русселем для вирішення задач з області штучного інтелекту та математичної лінгвістики. Мета створення цієї мови програмування – поєднати використання логіки з представленням знань. Prolog базується на логіці диз'юнктів Хорна, що є підмножиною логіки предикатів першого порядку.

Prolog є однією із найстарших мов логічного програмування, хоча він значно менш популярний за імперативні мови. Він використовується в системах обробки природних мов, дослідженнях ШІ: експертних системах, онтологічному аналізі і інших предметних областях, для яких використання логічної парадигми є природним.

Логічне програмування – основна парадигма Prolog, але пізніші реалізації, наприклад, Visual Prolog, підтримують об'єктно-орієнтоване чи кероване подіями програмування, іноді навіть з елементами імперативного стилю.

Структура програми на Prolog відрізняється від структури програми, написаної процедурною мовою. Prolog-програма – це набір правил і фактів. Рішення задачі досягається інтерпретацією цих правил і фактів. При цьому користувачу не потрібно забезпечувати детальну послідовність інструкцій, щоб указати, яким чином здійснюється керування ходом обчислень на шляху до результату. Замість цього він тільки визначає можливі рішення задачі і забезпечує програму фактами і правилами, що дозволяють їй відшукати необхідне рішення.

На сьогодні Prolog – одна з найпопулярніших мов програмування для доведення теорем, побудови експертних систем, обробки природномовних текстів. Її широко використовують в дослідницькій роботі та освіті, але промислове програмування на ній вважається складним, оскільки не всі компілятори підтримують модулі, а також існують проблеми сумісності між системами модулів основних компіляторів.

Prolog реалізовано практично для усіх відомих операційних систем і платформ, приміром, для Unix, Windows, iOS і для мобільних платформ. Існує кілька безкоштовних та комерційних реалізацій, що забезпечують створення зручних графічних інтерфейсів користувача.

PHP (http://php/net) – скриптова мова програмування, розроблена як інструмент для створення динамічних Web-сторінок і роботи з базами даних. Ця мова забезпечує генерацію HTML-сторінок на стороні Web-сервера. Зараз PHP є однією з найпоширеніших мов, що використовуються у розробці Web-застосувань, – вона фактично є стандартом для стеку LAMP (Linux, Apache, MySQL, PHP), що підтримується переважною більшістю хостинг-провайдерів.

Нині PHP загалом використовується для створення Web-застосувань, але вона придатна і для створення звичайних GUI-додатків (використовується зв'язування PHP GUI-бібліотекою GTK) чи CLI-додатків.

Основна реалізація PHP, розроблена PHP Group, є вільним програмним забезпеченням і поширюється на умовах ліцензії PHP License.

Головними недоліками мови вважають недостатню швидкість (робота інтерпретатора) та непередбачуваність нових версій, а також проблеми в безпеці, надійності, цілісності та передбачуваності.

Але PHP постійно розвивається, її нові версії передбачають реалізацію алгоритмів машинного навчання, роботу з API популярних сервісів DNNs та фреймворками Deep Learning Frameworks та з Big Data. Переваги PHP – простота синтаксису, досить багата функціональність. Наявність ядра і модулів, що підключаються, збільшують її можливості.

PHP-ML Machine Learning library for PHP – бібліотека для машинного навчання мовою PHP, яка містить алгоритми з аналізу даних, обробки багатошарових нейронних мереж, препроцесінг, видобування ознак тощо [17]. В цій бібліотеці реалізовано методи класифікації (приміром, метод *к*-найближчих сусідів), регресії, кластеризації тощо. Для використання бібліотеки PHP-ML необхідно мати PHP версії 7.1 або вище. PHP-ML використовує ліцензію MIT Licence.

PHP FANN – стандартне базове розширення PHP версій 5.2.0 і вище і бібліотеки libfann 2.1.0 і вище [17] реалізує в PHP багатошарову штучну нейронну мережу з підтримкою повнозв'язних та неповнозв'язних мереж. Розширення включає фреймворк для керування навчальними вибірками. Це розширення просте в використанні, гнучке, має добру документацію та швидко працює.

Базові функції: навчання мережі, операції із структурою нейронної мережі FANN (створення, копіювання, видалення), операції із структурою навчальних даних (створення, запис-читання, копіювання, видалення, об'єднання тощо), керування процесом навчання.

Рекомендації від світового товариства Data Science

Згідно опитування, яке було проведено видавництвом KDNuggets [18] в травні 2018, Python визнаний беззаперечним лідером у використанні його в Data Science and Machine Learning. Важливими чинниками цього вибору називають широкий вибір бібліотек і факт того, що це досить легка мова як для вивчення, так і для роботи. Крім того необхідно зауважити, що всі основні академічні курси з комп'ютерних наук, аналізу даних та машинного навчання, використовують Python для демонстрації і тестування прикладів. Таким чином, Python фактично став мовою, на якій розмовляють вчені з даних та машинного навчання.

Основні причини, за якими Python обирають більшість розробників:

- проста та високорівнева мова, яка дозволяє трьома рядками виконати складний процес;
- гнучкість;
- наявність розвинених бібліотек, зокрема Tensorflow, Keras і Theano;
- зрозумілість мови для вивчення завдяки традиційній об`єктній орієнтованості;

- широка підтримка професійних спільнот;
- універсальність;
- зручність інструментарія для розробника;
- швидке створення прототипів і побудова програм;
- підтримка модульності;
- швидке тестування.

У таблиці приведені результати опитування видавництвом KDNuggets світової спільноти спеціалістів, які працюють з великими даними та машинним навчанням.

Таблиця

Software	2018 %	% 2018 vs 2017
Python	65.6 %	11 %
R	48.5 %	– 14 %

В травні 2017 року [19] видання Towards Data Science провело опитування серед більше ніж 2 тисяч спеціалістів з Data Science та машинного навчання на тему які мови програмування вони використовують і над якими проектами працюють. Оскільки опитування проводили спеціалісти з даних, то за результатами опитування були побудовані декілька моделей з метою визначити найбільш важливі фактори, які впливають на вибір мови програмування для нового проекту. Порівняння топ-5 мов і результатів показало, що не існує простої відповіді на питання «яка мова краща». Все залежить від багатьох чинників: яку систему планується побудувати, який базис мають розробники, і чому було обрано саме машинне навчання як рішення задачі.

Найбільш популярною мовою на даний час вважається Python – 57 % вчених з даних і розробників машинного навчання використовують його, 33 % віддають цій мові перевагу.

Python часто порівнюють з R, але ці мови з погляду на рейтинг популярності досить далекі. R посідає 4-те місце в рейтингу з використання (31 %) та 5 позицію у питанні «якій мові ви віддасте перевагу при виборі для нового проекту» (5 %). R займає останнє місце – тільки 17 % розробників, які її використовували, віддадуть їй перевагу в новому проекті. Це говорить про те, що в більшості випадків R є доволі додатковою мовою. У Python цей показник 58 %, найвищий з мов, що чітко показує тренди використання. Отже Python є як найбільш вживаним, так і мова, якій більшість віддає перевагу при запуску нового проекту.

Друге місце за використанням (44 %) і за пріоритетністю (19 %) після Python займає C/C++. Третє місце посідає Java.

Мови машинного навчання типу Julia, Satlab, SAS попадають за межі 5 % в пріоритетності та використанні.

Аналіз результатів опитування показав, що при виборі мови програмування для машинного навчання, найбільш важливим є тип проекту – предметна область застосування.

Науковці з машинного навчання, які працюють в області аналізу емоцій віддають перевагу Python (44 %) та R (11 %) більше, ніж іншим мовам, що відрізняється від точки зору розробників з інших галузей.

Java обирають частіше, коли задачі пов`язані з безпекою мереж, кібер-атаками і розпізнаванням шахрайських дій. У даній галузі Python має останнє місце. Загалом, безпека мереж і алгоритми розпізнавання шахрайських дій розробляються і використовуються у великих організаціях, частіше в фінансових закладах, де Java завжди використовувалась для внутрішніх розробок. В областях, які менш орієнтовані на корпорації, типу обробки природної мови (natural language processing) та аналіз емоцій, розробники обирають Python, який надає простий і швидкий засіб для побудови високопродуктивних алгоритмів. Ця простота і швидкість отримується завдяки існуванню широкому набору спеціалізованих бібліотек.

В розробці комп'ютерних ігор (29 %) і пересувних роботів (27 %) – двох предметних областях (ПрО), де найпоширенішим є C/C++, яка надає високу продуктівність та ефективність, використання

алгоритмів ШІ розширює можливості таких програм. Логічно, що реалізацію алгоритмів ШІ варто створювати також на C/C++, для чого вже існують високопродуктивні бібліотеки.

Мові R найбільше віддають перевагу в сфері біоінженерії та біоінформатиці (11%), оскільки раніше її вже активно використовували в біомедичній статистиці в учбових закладах. Саме тому ця мова найбільш пріоритетна в цій галузі.

Крім проблематики тієї задачі, для якої розробляється ПЗ, на вибір мови програмування для машинного навчання є професійний досвід розробників. В залежності від свого попереднього професійного досвіду, розробники виділяють 5 мов. Python виділяють першим більшість з тих, для кого Data Science є першою спеціалізацією або областю вивчення (38 %). Це говорить про те, що Python на даний час став складовою частиною області Data Science і мовою спілкування (стандартом де-факто) для вчених і спеціалістів з даних. Цього не можна сказати про R, який частіше обирають спеціалісти з аналітики даних та статистики (14 %).

Відповідно, спеціалісти в C/C++ віддали перевагу своїй мові (8 %). Інженери і технічні спеціалісти, які використовують C/C++ для низькорівневого програмування контролерів та інших комп`ютерних елементів (embedded programming), віддають перевагу своїй мові і більше побоюються таких мов, як Java та R. Логічно, що вони частіше працюють на проектах, пов`язаних з низькорівневим машинним навчанням на апаратному рівні. Це проекти в галузі класифікації зображень, пересування роботів тощо.

Розробники, які працюють на Java, здебільшого займаються розробкою фронт-енд застосувань для стаціонарних комп`ютерів. Це пов`язано з здебільшого корпоративними застосуваннями. Корпоративні розробники схильні використовувати Java в усіх проектах, включаючи і машинне навчання (пріоритетність обрали 21 %). Однак, зазначимо, що Java є доволі складна мова програмування, що потребує великого часу для вивчення.

На відмінність від Java, Python є простою мовою, швидкою у вивченні, яку можна швидко застосувати для експериментів, щоб швидко розібратися в машинному навчанні.

Мови C/C++ для машинного навчання обирають користувачі, які бажають розширити свої існуючі проекти використанням машинного навчання (20 %) і зовсім рідко – для побудови нових застосувань (14 %).

Фронт-енд розробники, що працюють з Javascript, розширюють функціонал існуючих веб-застосувань та створюють нові, підключаючись за допомогою API до сервісів машинного навчання. Як приклад, це візуалізація роботи алгоритму машинного навчання на веб сторінці.

Проведений аналіз показав, що поняття «краща мова програмування для машинного навчання» не існує. Вибір мови програмування залежить від задачі, типу програми, галузі, з якої ви прийшли і для чого ви використовуєте машинне навчання. В більшості випадків розробники портирують алгоритми машинного навчання в мову, яка вже їм відома, особливо якщо це стосується задачі вдосконалення вже існуючих проектів, як наприклад, інженерні проекти для C/C++ або задача візуалізації для Javascript.

Для тих, хто тільки починає вивчати програмування і машинне навчання, Python буде найкращим вибором: просте використання і великий набір спеціалізованих бібліотек, але для роботи в великих компаніях, можливо, краще обрати Java або Scala.

Висновки

Вищенаведений аналіз основних задач машинного навчання, пов’язаних з обробкою Big Data, з метою здобуття з них корисних для практичного застосування відомостей, показав доцільність застосування для цього засобів статистичної обробки та роботи з нейромережами. Одночасно наявність великого різноманіття алгоритмів ML, орієнтованих на різні типи вхідної інформації та знань, що за ними будуються, свідчить про потреби в спеціа-

лізованих бібліотеках, що реалізують ці алгоритми.

На даний час існує багато он-лайн сервісів з машинного навчання, а також бібліотек і фреймворків, які можливо застосовувати в своїх розробках. Великі переваги мовам програмування надає можливість використовувати зовнішні хмарні сервіси для збереження та обробки великих даних.

Ще одним важливим фактором для вибору інструментального середовища, в якому вирішуються задачі ML, є швидкість обробки: це пов'язано з великими обсягами тих даних, що мають оброблятися.

Таким чином, для створення експериментальних прототипів, що поєднують сучасні підходи до машинного навчання з елементами штучного інтелекту, найбільш придатною мовою програмування є Python. Цей висновок підтверджують і результати опитувань розробників в сфері Data Sciences. Але інші мови програмування, проаналізовані в даній роботі, можуть бути навіть більш корисними за певних додаткових умов: приміром, для розробок, орієнтованих на специфічне програмне забезпечення або на створення корпоративних застосувань.

Література

1. Lynch C. Bigdata: How do your data grow? *Nature*. 2008. Vol. 455, N 7209. P. 28–29.
2. Gandomi A., Haider M. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management.* 35 (2). 2015. P. 137–144.
3. The Fourth Paradigm: Data-Intensive Scientific Discovery. 2009. http://research.microsoft.com/enus/collaboration/fourthparadigm.
4. Чехарин Е.Е. Большие данные: большие проблемы. *Перспективы науки и образования*, № 3 (21), 2016.
5. Гладун А.Я., Рогушина Ю.В. Data Mining: пошук знань в даних. К.:ТОВ "ВД "АДЕФ-Україна", 2016. 452 с.
6. TensorFlow. https://www.tensorflow.org/get_started/get_started.
7. The R Project for Statistical Computing. https://www.r-project.org.
8. Python. – https://www.python.org.
9. Gensim. – https://radimrehurek.com/gensim/.
10. Java. https://www.oracle.com/technetwork/java/index.html.
11. Топ 5 библиотек машинного обучения для Java. https://javarush.ru/groups/posts/254-top-5-bibliotek-mashinnogo-obuchenija-dlja-java.
12. Weka. https://www.cs.waikato.ac.nz/ml/weka/index.html.
13. MOA – Massive On-Line Analysis. https://moa.cms.waikato.ac.nz/.
14. The Scala Programming Language. https://www.scala-lang.org.
15. The Features of C++ as a Language. http://www.cplusplus.com/ info/description/.
16. PHP-ML. https://php-ml.readthedocs.io/en/latest/.
17. Fast Artificial Neural Network или FANN. – http://php.net/manual/ru/book.fann.php.
18. Python eats away at R: Top Software for Analytics, Data Science, Machine Learning in 2018: Trends and Analysis. – https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html.
19. Voskoglou C. What is the best programming language for Machine Learning? – https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7.

References

1. Lynch C. Bigdata: How do your data grow? *Nature*. 2008. Vol. 455, N 7209. P. 28–29.
2. Gandomi A., Haider M. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management.* 35 (2). 2015. P. 137–144.
3. The Fourth Paradigm: Data-Intensive Scientific Discovery. 2009. http://research.microsoft.com/enus/collaboration/fourthparadigm.
4. Cheharin E.E. Big data: big problems. *Perspectives of sciences and education*, 2016. N 3 (21). [in Russian].
5. Gladun A.Y., Rogushina J.V. Data Mining: retrieval of knowlegde into data. K.: ADEF-Ukraine, 2016. 452 p. [in Ukrainian].
6. TensorFlow. https://www.tensorflow.org/get_started/get_started.
7. The R Project for Statistical Computing. https://www.r-project.org.
8. Python. – https://www.python.org.
9. Gensim. – https://radimrehurek.com/gensim/.
10. Java. https://www.oracle.com/technetwork/java/index.html.

11. Top 5 libraries of machine learning for Java. – https://javarush.ru/groups/posts/254-top-5-bibliotek-mashinnogo-obuchenija-dlja-java. [in Russian].
12. Weka. https://www.cs.waikato.ac.nz/ml/weka/index.html.
13. MOA – Massive On-Line Analysis. https://moa.cms.waikato.ac.nz/.
14. The Scala Programming Language. https://www.scala-lang.org.
15. The Features of C++ as a Language. http://www.cplusplus.com/ info/description/.
16. PHP-ML. https://php-ml.readthedocs.io/en/latest/.
17. Fast Artificial Neural Network или FANN. – http://php.net/manual/ru/book.fann.php.
18. Python eats away at R: Top Software for Analytics, Data Science, Machine Learning in 2018: Trends and Analysis. – https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html.
19. Voskoglou C. What is the best programming language for Machine Learning? – https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7.

Одержано 05.10.2018

Про авторів:

Гришанова Ірина Юріївна,
науковий співробітник.
Кількість наукових публікацій в українських виданнях – 17.
Кількість наукових публікацій в зарубіжних виданнях – 3.
http://orcid.org/0000-0003-4999-6294.

Рогушина Юлія Віталіївна,
кандидат фізико-математичних наук,
старший науковий співробітник.
Кількість наукових публікацій в українських виданнях – 140.
Кількість наукових публікацій в зарубіжних виданнях – 30.
Індекс Хірша – 3.
http://orcid.org/0000-0001-7958-2557.

Місце роботи авторів:

Інститут програмних систем
НАН України,
03181, Київ-187,
проспект Академіка Глушкова, 40.
Тел.: 066 550 1999.

E-mail: i26031966@gmail.com,
ladamandraka2010@gmail.com,