

ДОСЛІДЖЕННЯ ЗАСОБІВ МАШИННОГО НАВЧАННЯ ІЗ ЗАЛУЧЕННЯМ МОВИ ПРОГРАМУВАННЯ JAVA

Анотація. В роботі проведено огляд програмного забезпечення, яке реалізує методи машинного навчання. Розглянуті можливості мови програмування Java для створення компонентів нейронних мереж.

Ключові слова: машинне навчання, нейронні мережі, Java, Neuroph, Nengo, Deeplearning4j, Datumbox Machine Learning Framework.

Вступ. На сьогоднішній час набули розповсюдження різноманітні методи машинного навчання. З одного боку, вони пов'язані з методами навчання нейронних мереж та видами топології їх архітектури, а з іншого боку, увібрали в себе методи математичної статистики.

Основними класичними задачами, які вирішуються за допомогою машинного навчання, являються класифікація, кластеризація, задачі прогнозування, зниження розмірності даних і їх візуалізація, відновлення щільності розподілу ймовірності за наборами даних інше.

На початкових стадіях створення систем машинного навчання велику роль відіграє розвідувальний аналіз та попередня підготовка даних (завдання підготовки ознак - feature engineering), котре включає, в тому числі, проведення обчислювальних експериментів за їх тестовою обробкою з використанням альтернативних програмних реалізацій різних методів машинного навчання [1]. У зв'язку з цим становить практичний інтерес актуальна інформація щодо наявності програмного забезпечення, яке реалізує ті чи інші методи машинного навчання, що дозволяє в ході досліджень оперативно вирішувати питання щодо вибору найбільш ефективного програмного інструменту. З іншого боку, може передбачатись використання деякого універсального програмного забезпечення, яке надає однаковий підхід до вирішення завдань, безпосередньо пов'язаних з реалізацією методів машинного навчання [2].

Для автоматизованого розпізнавання образів використовуються такі типи машинного навчання (machine learning):

- 1) навчання з учителем (supervised learning);
- 2) навчання без вчителя (unsupervised learning);
- 3) навчання з частковим залученням вчителя (semi-supervised learning).

Одним з методів машинного навчання є нейронні мережі, які використовують в задачах обробки інформації [3, 4]. Нейрона мережа представляє собою паралельну динамічну систему з топологією направленого графа, котра може отримувати вихідну інформацію шляхом реакції стану на вхідний вплив. Нейронні мережі вирішують широкий спектр задач: класифікація об'єктів, виявлення залежностей, розпізнавання образів, прогнозування, апроксимація, стискання даних. Універсальність нейронних мереж породжує задачу їх реалізації на різних мовах програмування з використанням передових технологій, що дозволяє застосовувати їх в будь-якій сфері від мобільних додатків до серверних.

Створюючи програми на більшості мов програмування, треба визначити, в якій операційній системі і на якому процесорі вони працюватимуть. Створюючи програми на мові Java, можна не замислюватися над цим. Java має власний набір машинно - незалежних бібліотек. Компілятор Java створює проміжний байт-код для віртуальної машини Java (Java Virtual Machine – JVM). Ядро віртуальної машини Java реалізовано практично для всіх типів комп'ютерів та операційних систем, тому файли байт-кодів вважаються незалежними від платформи[5].

На даній мові можна створювати як програмне забезпечення для використання в великих промислових системах (Java EE), так і додатки для платформи Android.

Визначення проблеми. Метою роботи є аналіз програмних засобів, реалізованих на мові програмування Java, для створення нейронних мереж, які застосовуються для розпізнавання зображень сигналів.

Основна частина. Для реалізації нейронної мережі на будь-якій мові програмування потрібно декілька складових:

- безпосередній опис об'єктів нейронної мережі, таких як нейрон, зв'язок, шар, входи і виходи мережі;

- представлення в конструкціях мови архітектури мережі: зв'язки між шарами, їх кількість і тип;
- програмні функції, які реалізують алгоритми навчання мережі, котрі можуть дозволити задати кількість епох навчання, встановити цільову погрішність;
- збереження і завантаження параметрів навченої мережі.

Для розв'язання задачі можна використовувати різні підходи. Перший припускає кодування усіх вузлів нейронної мережі і подальше їх використання. Другий полягає у використанні готових бібліотек і вимагає програмування тільки самої архітектури мережі. В даному випадку кожен елемент нейронної мережі може бути відображений своїм класом [6].

Для функціонування нейронної мережі необхідно використання функції активації, яка нормалізує дані, а також фактично є логічним ядром нейрона: вона дозволяє отримати вихідні значення сигналу по вхідному. Найчастіше використовують наступні функції активації [7]:

- гіперболічний тангенс $f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$; (1)

- сігмоїдна функція $f(x) = \frac{1}{1 + e^{-x}}$. (2)

Нейронні мережі найчастіше складаються з великої кількості шарів. Після створення необхідних нейронів вони групуються в масив або список з нейронів і формують архітектуру нейронної мережі. Після створення нейронної мережі необхідно провести її навчання.

Навчання проводиться на тренувальних наборах (сетах) – послідовності даних, якими може оперувати нейронна мережа. Обробка тренувального сету може проходити в декілька етапів – епох. На кожній з епох нейронна мережа обробляє повний набір даних. Метою навчання є зниження помилки, яка відображає розбіжність між очікуваними і отриманими результатами. Помилка визначається на кожній епосі і при успішному навчанні повинна ставати меншою [7]. Для обчислення помилки використовуються різні способи [7]:

- Mean Squared Error (MSE) $mse = \frac{1}{N} \sum_{i=1}^N (t_i - a_i)^2$; (3)

- Root MSE $\sqrt{mse} = \sqrt{\frac{1}{N} \sum_{i=1}^N (t_i - a_i)^2}$. (4)

Кожен із способів має свої особливості, але вони усі реалізують однакову логіку: чим ближче вихідний вектор до очікуваного, тим менше похибка. [7].

Проведемо огляд основних бібліотек, які дозволяють спростити програмування нейронних мереж на мові Java.

Фреймворк Neuroph [8]. Neuroph – це легкий фреймворк, написаний на мові Java. Він дозволяє створювати мережі з традиційними архітектурами:

- адалайн (Adaptive Linear Neuron - адаптивний лінійний нейрон);
- багат шаровий персептрон з алгоритмом зворотного розповсюдження помилки;
- мережа Хопфілда;
- мережа Кохонена;
- мережа радіально-базисних функцій
- двостороння асоціативна пам'ять (нейронна мережа Коско);
- мережа Хебба;
- згортоква нейронна мережа..

З методів машинного навчання в Neuroph реалізовані кластеризація методом k-середніх та метод найближчих сусідів (класифікація і регресія).

Neuroph складається з двох компонентів: логіки нейронних мереж і графічного додатку Neuroph Studio, який дозволяє конфігурувати мережу без написання коду. У графічному режимі потрібно вибрати тип мережі, її параметри, задати навчальну вибірку. Після цього стає доступним перегляд структури отриманої мережі, з'являється можливість перевірки якості навчання мережі.

Основною областю застосування пакета є розпізнавання образів. Neuroph є бібліотекою Java-класів (таким чином, створені в ньому нейронні мережі можна безпосередньо використовувати в програмах на Java), доповнену графічним середовищем Neuroph Studio для створення і навчання нейромереж.

Крім графічного інтерфейсу програмісту надається доступ до класів бібліотеки. Конфігурування мережі відбувається в об'єктно-орієнтованому стилі.

Використання Neuroph Studio може спростити навчання теорії нейронних мереж, а бібліотека Neuroph дозволяє легко конфігурувати

необхідну архітектуру мережі. Дану бібліотеку можна розширювати за рахунок додавання нових типів мереж і способів навчання.

Deeplearning4j [9] – бібліотека, яка реалізує алгоритми глибокого навчання (deep learning algorithms). Глибоке навчання – це окремий клас задач, в який входить, наприклад, розпізнавання образів на зображеннях. Для глибокого навчання потрібно більше обчислювальних ресурсів, ніж для класичних нейронних мереж. Бібліотека *Deeplearning4j* написана на Java, працює на Java Virtual Machine. На відміну від *Neuroph* складні обчислення в *DL4J* здійснюються окремим модулем, який працює незалежно від JVM, що дозволяє прискорити роботу бібліотеки. Обчислення можуть також проводитися на розподілених ресурсах і на графічних прискорювачах. Вся конфігурація проводиться за допомогою програмного коду.

Пакет *Nengo* (Neural Engineering Object) [10] орієнтований на моделювання великомасштабних нейронних мереж і працює як під управлінням скриптів, так і за допомогою графічного інтерфейсу. Останній дозволяє створювати і з'єднувати ансамблі нейронів засобами візуального моделювання. На відміну від традиційного нейронного моделювання в *Nengo* не потрібно використовувати правило навчання, щоб знайти вагові коефіцієнти нейронної мережі. Замість цього в *Nengo* для пошуку ваг використовується Neural Engineering Framework (NEF). Крім того, в *Nengo* використовується адаптований формалізм теорії управління, що дозволяє реалізувати засобами пакета складні динамічні системи.

Datumbx Machine Learning Framework [11] - фреймворк, реалізований на мові Java, орієнтований на швидку розробку додатків для машинного навчання і різних статистичних додатків. Основна частина проекту використовує функції Java 8 та застосовує Maven Project Structure. Крім того, фреймворк можна використовувати в якості бібліотеки, яка надає класи для вирішення наступних завдань машинного навчання: регресія, класифікація, кластеризація, зниження розмірності, тематичне моделювання (topic modeling), виділення ознак (feature selection) і створення рекомендаційних систем. Ось деякі з реалізованих в *Datumbx Machine Learning Framework* алгоритмів: лінійний дискримінантний аналіз; метод максимальної ентропії; наївний байєсовський класифікатор; метод опорних векторів; бутстреп-агрегація (беггінг); Adaboost (адаптивний бустінг); метод k-

середніх; ієрархічна кластеризація; змішані моделі процесу Діріхле; Softmax-регресія; ординальне логістична регресія; лінійна регресія; покрокова регресія; аналіз головних компонент.

Encog [12] - фреймворк, присвячений машинному навчанню, доступний на мовах Java, .Net і C ++. Фреймворк підтримує метод опорних векторів, штучні нейронні мережі (зокрема, мережі Кохонена), генетичні алгоритми та генетичне програмування, а крім того, байєсовські мережі і приховані марковські моделі. Більшість алгоритмів навчання в Encog є багатопоточними і масштабуються на багатоядерних апаратних засобах. Encog може також використовувати графічний інтерфейс користувача для подальшого скорочення часу обробки даних.

Висновки. Проведено огляд програмного забезпечення, яке реалізує методи машинного навчання на мові програмування Java, і, яке можна використовувати для розпізнавання зображень.

Одним із методів машинного навчання є нейронні мережі. В роботі аналізуються різні підходи до їх реалізації. Кожен з підходів має свої переваги і сфери застосування. Програмування з нуля добре лягає на об'єктну модель і дозволяє врахувати особливості конкретної мережі, щоб використовувати тільки ті елементи, які необхідні. *Neuroph* представляє графічний інтерфейс і спрощує створення архітектури мережі, даючи можливість реалізувати більшість відомих типів нейронних мереж. *Deeplearning4j* дозволяє вирішувати завдання глибокого навчання, а також оптимізує обчислення за рахунок використання розподілених ресурсів. *Datumbbox Machine Learning Framework* можна використовувати в якості бібліотеки, яка надає класи для вирішення завдань класифікації і кластеризації. Вибір підходу обумовлює задача, яку треба розв'язати.

ЛИТЕРАТУРА

1. Coelho L.P., Richert W. Building Machine Learning Systems with Python. Packt Publishing, 2015. 302 p.
2. Machine learning // Wikipedia, the free encyclopedia: internet-portal. Available at: https://en.wikipedia.org/wiki/Machine_learning.
3. Software // Machine Learning Open Source Software. Available at: <http://www.mloss.org/software>
4. Misiti J. Awesome Machine Learning // GitHub: public repository. Available at: <https://github.com/josephmisiti/awesome-machine-learning> .
5. Васильев А. Н. Java. Объектно-ориентированное программирование. — СПб.: Питер, 2011. — 400 с.
6. Тархов Д. А. Нейросетевые модели и алгоритмы. Справочник. — М.: Радиотехника, 2014. — 352 с.
7. Хайкин Саймон. Нейронные сети: полный курс, 2-е издание.: Пер. с англ. — М.: Издательский дом «Вильямс», 2006
8. Neuroph // Neuroph Java Neural Network Framework: official website. Available at: <http://neuroph.sourceforge.net/>
9. Deeplearning4j – Open-source, distributed deep learning for the JVM. DL4J Deep Learning for Java: official website. Available at: <http://deeplearning4j.org/>
10. Nengo // The Nengo Neural Simulator: official website. Available at: <http://nengo.ca/>
11. Datumbox Machine Learning Framework // Datumbox: official website. Available at: <http://www.datumbox.com/machine-learning-framework>
12. Encog Machine Learning Framework // Heaton Research: official website. Available at: <http://www.heatonresearch.com/encog/>.