УДК 629.33:681.51 DOI: 10.30977/BUL.2219-5548.2021.92.1.39

OVERVIEW OF PYTHON LIBRARIES FOR ANALYSIS GEOGRAPHICAL DATA

Pronin S.
Kharkiv National Automobile and Highway University

Abstract. *The article analyzes the possibility of using specialized libraries of the Python language for processing and analyzing data in geographic information systems. The article analyzes the main directions for the application of the methods of intelligent systems in the processing of geoinformation, and also considers the tools for the analysis.*
Keywords: *geographic information system, Python, machine learning.*

Introduction

Today, systems that link heterogeneous information to geographic and topographic data are widely used. Such systems are called geographic information systems (GIS). In these systems, issues related to the processing and analysis of information are of great importance. To solve this problem, at the present stage, various methods of artificial intelligence, statistical analysis, machine learning and work with "big data" are used. To apply these methods within the framework of programming languages, various specialized libraries have been developed that allow you to create custom applications.

The purpose of this work will be the choice of tools for data analysis in geo-information systems. The objectives of the research are the analysis of libraries for processing and analysis of geographic data. This article provides an overview of the corresponding Python toolkit.

Analysis of publications

When analyzing publications devoted to the application of the methods of intelligent systems for processing geographic information, the following works can be distinguished – [1–7].

Based on the analysis of these sources, it can be concluded that the intellectualization of a GIS is understood as the introduction of artificial intelligence methods and tools into its composition.

Also today, a rich toolkit has been developed for data mining and machine learning [8–11]. This makes it possible to create integrated systems for storing and structuring geo-information and systems for its analysis.

The introduction of artificial intelligence methods into GIS at various stages of data processing and analysis can be divided into several directions:

– search algorithms and data recognition. At this stage, the problem of recognizing structures and objects is being solved, as well as the implementation of intelligent search;

– algorithms for data analysis and interpretation. This stage involves the use of various heuristic algorithms for data management and analysis algorithms. At this stage, the analysis model is selected, the results are interpreted and the forecast function;

– creation of interfaces for providing information. It is a decision support system based on the results obtained when solving previous problems.

All of the above tasks can be solved with the help of the developed tools for creating intelligent systems [1–7].

Among the main tasks of intelligent systems are [12]:

– recognition – the ability of a system to classify an object or phenomenon with which it encounters in solving its problems;

– training (in terms of the scientific direction «machine learning») – the systematic training of algorithms and systems, as a result of which their knowledge or quality of work increases with the accumulation of experience;

– knowledge engineering is a scientific direction that studies algorithms and methods for modeling human knowledge that can be used in the process of training a system or in the process of logical inference;

– hese tasks can be used at any stage of the analytical process, both singly and in various combinations, depending on the task.

Application of artificial neural networks for solving the problem of image recognition and classification

A feature of building geoinformation systems is the need to divide the map into functional layers. To solve this problem, it is possible to use various methods of image recognition, which will allow you to select different objects on the

maps and divide them according to their functional purpose.

Modern image recognition systems are a set of special mathematical methods that allow you to find the desired object in the resulting image. The actual recognition problem can be divided into two subtasks [13]:

– selection of the desired object in the video stream;

– object classification (recognition).

Accordingly, methods for solving the problem can be divided into two groups: methods that allow you to select a fragment and methods that allow it to be recognized and classified.

To solve the first subproblem, various filtering methods are used, such as the Sobel operator, Laplace operator, Kenny's boundary detector, and others. These methods can be used if the task is to select the object we need [13].

Since after isolation, it is necessary to prolime recognition and classification, the use of methods of the first group is not sufficient. To solve this problem, various machine learning methods are used [13].

Among them, the most widespread today are artificial neural networks (ANNs).

The advantages of using ANN include:

– the ability to solve a wide range of tasks related to pattern recognition;

– can be applied to the recognition of any types of objects, both two-dimensional and linear;

– one network can recognize several images at once;

– the possibility of retraining or additional training in the process;

– the ability to work with noisy data;

– good scalability.

Among the shortcomings, one can single out the low speed of work, especially at the training stage, which, as a rule, is associated with the need to configure the network.

In general, setting up a neural network for recognition and classification will have the following steps:

1) formation of a training sample;

2) submitting training examples to the network input;

3) network training;

4) checking network operation on a test sample.

If, after the fourth stage, the network will stably solve the task, then it is considered trained and ready to work.

Toolkit for creating machine learning systems

The most common tools for working with big data analysis today are the R and Python programming languages, as well as a set of related machine learning and data manipulation libraries [8–11].

Programming language R – a programming language designed for statistical processing of data and working with graphics, but at the same time it is a free open source software environment developed by the GNU project. R has become widespread in areas where work with data is carried out. The main computing power of R lies in statistical analysis, but it also has extensive functionality for primary data analysis (plotting graphs and contingency tables) and mathematical modeling.

Programming language Python – a high-level general-purpose interpreted scripting language. When developing in the Python language, great attention is paid to the simplicity and clarity of syntax, which not only reduces the time spent learning its basics, but also increases the speed of development in general [8].

These are not all the advantages of this language, the main ones are:

– object-oriented;

– free distribution and broad support;

– cross-platform;

– advanced functionality.

The language is cross-platform thanks to its implementation in portable ANSI C, which allows programs written in Python to compile and run equally well on any platform where a compatible version of Python is installed.

The hybrid nature of Python combines the simplicity and convenience of scripting languages with the power of compiling languages to make Python a convenient tool for developing all kinds of applications. However, the greatest efficiency of the language is achieved when solving problems of data analysis and automation of processes. Python is widely used in research projects. The Python programming language has powerful built-in tools (built-in object types and dynamic typing, automatic memory management) and the ability to use external libraries and third-party utilities to solve more highly specialized tasks.

Python is used not only by individual users, but also by companies, including commercial use. For example:

– Google uses Python extensively in its search engine and to build its App Engine

framework. Also Google has developed a free library for machine learning – Tensor Flow;

– YouTube's video sharing service is largely implemented in Python – manufacturers of electronic devices and computer components (such as Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm, and IBM) use Python to test hardware;

– a geographic information systems company (Environmental Systems Research Institute) uses Python as a tool to customize its software products to the needs of the end user.

Library scikit-learn. Due to its widespread distribution, Python has gathered around itself an active community of developers who, within the framework of various projects, develop modules for highly specialized tasks [9].

The development of this library is one of the reasons for the popularization of the use of the Python language in the field of data analysis using machine learning methods.

The scikit-learn library provides implementations of a number of algorithms for both Supervised learning and Unsuoervised learning.

Scikit-learn is built on top of the SciPy (Scientific Python) stack, which includes:

1) NumPy – is a library for the Python programming language that supports large, multidimensional arrays and matrices with a large collection of high-level mathematical functions to work on these arrays.

NumPy is open source software with many members. NumPy is aimed at implementing the Python reference program, which is a bytecode interface optimizer. Mathematical algorithms written for this version of Python often run much slower than compound equivalents. NumPy partially solves the problem of slowness by providing multidimensional arrays and functions and operators that work efficiently on arrays, requiring you to rewrite some code, mostly internal loops with NumPy.

The main functionality of NumPy is its «ndarray» – the data structure for an n-dimensional array. These arrays have a strict memory view. Unlike the built-in Python list data structure, this array must have all elements of the same type. Such arrays can also view the memory buffers allocated by the C / C ++, Cython, and Fortran extensions to the CPython interpreter without having to copy the data around, providing compatibility with existing number libraries. NumPy has built-in support for ndarrays mapped to memory;

2) SciPy – an open source library of high-quality scientific tools for the Python programming language;

3) Matplotlib – Library in Python programming language for data visualization with two-dimensional (2D) graphics (3D graphics are also supported). The original images can be used as illustrations in publications. Matplotlib is a flexible as well as easily configurable package that, along with other libraries such as NumPy, SciPy and IPython, provides features similar to MATLAB. The package can support the following types of graphs and charts: scatter plot, line plot, histogram, bar chart, pie chart, contour plot, bar chart. stem plot, quiver fields and spectrograms;

4) IPython is an interactive shell for the Python programming language that provides advanced introspection, additional command syntax, code highlighting, and auto-completion Sympy is a library for working with symbolic computations;

5) Pandas implements various data structures and analysis.

The scikit-learn library consists of 35 modules, which can be subdivided into clustering modules, modules for evaluating the model and quantifying the quality of forecasts, modules for working with datasets (preprocessing, normalization), modules for working with features (extraction and identification of the most significant), modules that implement various algorithms for solving problems of classification and regression. Each module consists of classes and functions and solves problems such as:

– clustering – grouping of unallocated data;

– cross Validation - an assessment of the performance of the model on independent data;

– data sets (Datasets) – for storing test data sets and for generating data sets with certain properties for studying the behavioral properties of the model;

– dimensionality reduction – a set of algorithms for reducing the number of attributes for visualization and Feature Selection, for example, Principal Component Analysis;

– algorithmic compositions (Ensemble Methods) – a set of methods for combining predictions of several models;

– feature extraction – the process of defining attributes in data;

– feature selection – a set of algorithms for identifying significant attributes on the basis of which the model will be built;

– algorithm parameter optimization (Parameter Tuning) – methods to get the most effective output from the model;

– multiple learning (Manifold Learning) – an approach of non-linear reduction of data dimension.

Separately, it is necessary to highlight the methods that implement training with a teacher (Supervised Models).

This set of methods includes:

– generalized linear models;

– discriminate analysis methods;

– naive bayes classifier;

– neural networks;

– support vector machines;

– decision trees.

Library TensorFlow. An open software library for machine learning, developed by Google to solve problems of building and training a neural network in order to automatically find and classify images, reaching the quality of human perception. Designed to work with deep neural networks, such as convolutional and generative adversarial networks [10].

Calculations in TensorFlow are performed using data-flow graphs. In these graphs, the vertices are mathematical operations, while the edges are data that are usually represented as multidimensional arrays or tensors that are reported between these edges.

By opening the source code of the TensorFlow machine learning library, Google has simplified the process of building and deploying complex neural networks. TensorFlow does not allow every developer to benefit from the fruits of machine learning, but offers APIs for Python and C / C ++ that allow you to connect to the developer's program.

This type of machine learning is designed exclusively for research purposes, but thanks to open source software like TensorFlow, the company gets powerful tools to use its own data and process it in a cheap cloud environment. TensorFlow libraries significantly simplify the integration into applications of self-learning elements and functions of artificial intelligence, designed for speech recognition, computer vision or natural language processing. Of course, TensorFlow is not the only deep learning library, but like Google's search engine, it is considered the best in its class.

TensorFlow was conceived for Deep Learning, where the user builds the neural network architecture he needs. But the library also allows you to work with statistical algorithms for machine learning, although it does not provide them directly out of the box, that is, they also need to write yourself, and TensorFlow provides tools for this.

Library eo-learn. This is an open source Python library that uses images from artificial earth satellites to intelligently analyze data using Python machine learning libraries [11].

The library uses primitives from the numpy and shapely libraries to store and manipulate data from satellites.

Currently there are the following packages:

– eo-learn-core – the main package that implements the basic building blocks (EOPatch, EOTask and EOWorkflow) and commonly used functions;

– eo-learn-coregistration – a package that deals with joint registration of images;

– eo-learn-features – a set of utilities for retrieving and manipulating data properties;

– eo-learn-geometry – geometry package used for geometric transformation and transformation of vector and raster data;

– eo-learn-io – I / O packet that deals with obtaining data from Sentinel Hub services or local persistence and loading data;

– eo-learn-mask – package used for data masking and cloud mask computation;

– eo-learn-ml-tools – various tools that can be used before or after the machine learning process;

– eo-learn-visualization – visualization tools for the main elements of eo-learn.

Conclusions

In geographic information systems, issues related to the processing and analysis of information is of great importance. To solve this problem, it is possible to use various methods of artificial intelligence, statistical analysis, machine learning and work with «big data». To apply these methods within the framework of programming languages, various specialized libraries have been developed that allow you to create your own applications. Among them, the most promising are artificial neural networks (ANNs).

For practical implementation, it is possible to use tools for data mining systems and machine learning. This makes it possible to create integrated systems for storing and structuring geographic information and systems for its analysis.

The most common tools for working with big data analytics today are the Python programming language, as well as a collection of machine learning and data libraries such as scikit-learn, TensorFlow, and the eo-learn specialized library.

References

1. Глотов А. А. Геоинформационное моделирование эволюции долинно-речных ландшафтов Воронежской области: автореф. дис. на соискание ученой степени кандидата географических наук. Воронеж, 2013. 24 с.

2. Гаврилова Т. А., Муромцев Д. И. Интеллектуальные технологии в менеджменте: инструменты и системы: учебное пособие. 2-е издание. Санкт-Петербург: Высшая школа менеджмента, 2008. 488 с.
3. Ивакин Я. А. Интеллектуализация ГИС. Методы на основе онтологий. LAP Lambert Academic Publishing, 2010. 322 с.
4. Савиных В. П., Цветков В. Я. Развитие методов искусственного интеллекта в геоинформатике. Транспорт Российской Федерации. 2010. № 5. С. 41−43.
5. Kendal S. L., Green M. An introduction to knowledge engineering. London: Springer, 2007. 287 p.
6. McKeown David M. The role of artificial intelligence in the integration of remotely sensed data with geographic information systems. Pittsburgh, 1986. 36 p.
7. Popovich V. Intelligent GIS Conceptualization. Information Fusion and Geographic Information Systems, Lecture Notes in Geoinformation and Cartography. 2014. P. 17−44.
8. Documentation Python.org. [Электронный ресурс]. https://www.python.org/.
9. Scikit-learn. Machine Learning in Python. [Электронный ресурс]. http://scikit-learn.org.
10. TensorFlow [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/.
11. Eo-learn [Электронный ресурс]. https://eo-learn.readthedocs.io/en/latest/.
12. Russel Stuart J. Artificial Intelligence. A modern approach. New Jersey, 1995. 932 p.
13. Форсайт Д., Понс Же. Компьютерное зрение. Современный подход. Москва: Вильямс, 2004. 928 с.

References

1. Glotov A. A. Geoinformacionnoe modelirovanie evolyucii dolinno-rechnyh landshaftov Voronezhskoj oblasti: avtoreferat dissertacii kandidata geograficheskih nauk [Geoinformation modeling of evolution of valley-river landscapes of the Voronezh region: abstract of the dissertation of the candidate of geographical sciences]. Voronezh, 2013. 24 s.[in Russian]
2. Gavrilova T. A., Muromcev D. I. Intellektual'nye tekhnologii v menedzhmente: instrumenty i sistemy: uchebnoe posobie. 2-e izdanie. Sankt-Peterburg: Vysshaya shkola menedzhmenta, 2008. 488 s. [in Russian].
3. Ivakin YA. A. Intellektualizaciya GIS. Metody na osnove ontologij [GIS Intellectualization. Ontology based methods]. LAP Lambert Academic Publishing, 2010. 322 s. [in Russian]
4. Savinyh V. P., Cvetkov V. YA. Razvitie metodov iskusstvennogo intellekta v geoinformatike [The development of artificial intelligence methods in geoinformatics]. Transport Rossijskoj Federacii. 2010. № 5. S. 41−43.
5. Kendal S. L., Green M. An introduction to knowledge engineering. London: Springer, 2007. 287 p.
6. McKeown David M. The role of artificial intelligence in the integration of remotely sensed data with geographic information systems. Pittsburgh, 1986. 36 p.
7. Popovich V. Intelligent GIS Conceptualization. Information Fusion and Geographic Information Systems, Lecture Notes in Geoinformation and Cartography. 2014. P. 17−44.
8. Documentation Python.org. [Elektronij resurs]. - https://www.python.org/.
9. Scikit-learn. Machine Learning in Python. [Elektronij resurs] . - http://scikit-learn.org
10. TensorFlow [Elektronij resurs]. – Regim dostupa: https://www.tensorflow.org/.
11. Eo-learn [Elektronij resurs]. https://eo-learn.readthedocs.io/en/latest/.
12. Russel Stuart J. Artificial Intelligence. A modern approach. New Jersey, 1995. 932 p.
13. Forsajt D., Pons Zhe. Kompyuternoe zrenie. Sovremennyj podhod. Moskva: Vilyams, 2004. 928 s.

Pronin Sergey Viktorovich, Ph.D., Associate Professor, Department of Computer Technologies and Mechatronics, psv59777@gmail.com,
tel. 057-707-37-43
Kharkov National Automobile and Highway University, st. Yaroslav the Wise, 25, Kharkov, 61002, Ukraine.

Аналіз бібліотек мови python з метою оцінювання географічних даних

Анотація. Нині *значного поширення отримали системи, які містять різноманітну інформацію щодо географічних і топографічних даних. Такі системи називаються геоінформаційними системи (ГІС). За їхньою допомого вирішують питання, пов'язані з обробленням і аналізом інформації. Для вирішення цього завдання на сучасному етапі застосовують різноманітні методи штучного інтелекту, статистичного аналізу, машинного навчання та роботи з «великими даними». Для застосування цих методів на основі мов програмування розроблені різноманітні спеціалізовані бібліотеки, що дозволяють створювати призначені для користувача програми. Метою роботи є вибір інструментарію для аналізу даних у геоінформаційних системах. Завданнями дослідження є аналіз бібліотека для оброблення й аналізу географічних даних. У статті аналізують відповідний інструментарій мови Python. На основі аналізу конкретних джерел можна дійти висновку, що інтелектуалізацією ГІС є впровадження до її складу методів та інструментів штучного інтелекту. Також на сьогодні розроблено велику кількість*

інструментарію для інтелектуального аналізу даних і машинного навчання. Це дає можливість для створення інтегрованих систем зберігання та структуризації геоінформації і систем її аналізу. Особливістю побудови геоінформаційних систем є необхідність поділу карти за функціональними верствами. Для вирішення цього завдання можна використовувати різноманітні методи розпізнавання образів, що дозволить виокремити на картах різноманітні об'єкти та поділити їх за функціональним призначенням. Сучасні системи розпізнавання образів є набором спеціальних математичних методів, які дозволяють в отриманому зображенні знайти потрібний об'єкт. Найбільш розповсюдженими на сьогодні є штучні нейронні мережі (ШНМ). До переваг застосування ІНС належать:

− можливість вирішення великого кола завдань, пов'язаних із розпізнаванням образів;

− можливість використання будь-яких типів об'єктів (як двовимірних, так і лінійних);

– одна мережа може розпізнавати одночасно декілька образів;

– можливість навчання в процесі роботи;

– можливість роботи з зашумленими даними;

– маштабованість.

Найбільш поширеними інструментами для роботи з аналізом великих даних на сьогоднішні є мови програмування R і Python, а також набір пов'язаних з цими мовами бібліотек машинного навчання і роботи з даними, зокрема бібліотека scikit-learn і нова, але вже досить популярна спеціалізована бібліотека для роботи з геоданими eo-learn.

Все це дає можливість для створення систем аналізу на основі застосування бібліотек машинного навчання

***Ключові слова**: геоінформаційна система, інтелектуальна система, машинне навчання.*

Пронін Сергій Вікторович, к.т.н., доцент кафедри комп'ютерних технологій і мехатроніки, psv59777@gmail.com, тел. 057-707-37-43
Харківський національний автомобільно-дорожній університет, вул. Ярослава Мудрого, 25, м. Харків, 61002, Україна.