engineering proceedings

MDPI

Proceeding Paper

Development of a Server for the Implementation of Data Processing Pipelines and ANN Training †

Brais Galdo *, Daniel Rivero and Enrique Fernandez-Blanco

Faculty of Computer Science, CITIC, University of A Coruna, 15071 A Coruna, Spain; daniel.rivero@udc.es (D.R.); enrique.fernandez@udc.es (E.F.-B.)

* Correspondence: brais.cgaldo@udc.es

† Presented at the 4th XoveTIC Conference, A Coruña, Spain, 7–8 October 2021.

Abstract: Data processing and the use of machine learning techniques make it possible to solve a wide variety of problems. The great disadvantage of using this type of technology is the enormous amount of computation involved. This is why we have tried to develop an architecture that makes the best possible use of the resources available on each machine. The growth of cloud computing and the rise of virtualization techniques have led to a development that allows these tasks to be carried out in a more optimized way.

Keywords: machine learning; Artificial Neural Networks; deep learning; data processing; web server; virtualization; docker

Citation: Galdo, B.; Rivero, D.; Fernandez-Blanco, E. Development of a Server for the Implementation of Data Processing Pipelines and ANN Training. *Eng. Proc.* **2021**, *7*, 38. https://doi.org/10.3390/engproc2021007038

Academic Editors: Joaquim de Moura, Marco A. González, Javier Pereira and Manuel G. Penedo

Published: 18 October 2021

1. Introduction

The use of data processing techniques [1] and machine learning [2] is based on trying to detect patterns in a set of data in order to provide an estimate on the data. This technology is experiencing a great boom due to the optimization of the different algorithms and the notable increase in the computational capacity of the different systems.

Both database processing and the training of machine learning models are very complex and computationally expensive tasks. When this is added to the processing of very large databases or the development of complex models, it is common to have specific hardware to speed up these tasks. Otherwise, this task would take a long time to be performed on a conventional computer, even breaking some of its components due to the stress caused by computational volume.

In addition, defining different processing pipelines or different models can be very complex for people who are not experts in the field. To alleviate these deficiencies, there are tools that allow this task to be carried out visually. This would be the case of Weka [3], which allows performing these tasks in a simple way. However, this application does not allow its execution on different machines.

With these points in mind, namely ease of use and scalability, the architecture of a distributed system for database processing and training of machine learning models is proposed. In this way, the resources of the machine on which the different processes are executed will be specifically dedicated to this task.

2. Materials and Methods

The boom of the different virtualization technologies [4] makes them ideal for the construction of a system of this style. They allow the developed system to be independent of the machine on which it is executed, which provides great versatility and flexibility. In addition, these technologies allow an exclusive use of the resources, allowing them to contain only the necessary modules. One of the most powerful and versatile technologies in this field is *Docker* [5], which allows an easy definition of systems with their characteristics to be taken into account. This, in addition to efficiency when carrying out the tasks, would

provide greater security since the machine will only contain the services necessary to perform the task entrusted to it.

The architecture developed must also allow the management of different users and databases. This is not such a costly task, so it will be included in the same module to optimize the architecture resources.

3. Results

Thanks to this architecture, load balancing [6] of the different training and data processing processes can be carried out exclusively. This implies that the nodes will be activated on demand and will have all the resources dedicated to the work they want to perform without taking into account other functionalities such as user authentication or the management of the different files that would consume a series of resources unnecessarily. Likewise, the architecture of the system would be as shown in the Figure 1.

It is necessary to mention that the current development is based on the ANN technique, which allows the implementation of deep learning models [7].

This architecture can be divided into a front-end part based on an MVC pattern [8] and a back-end part composed of three large modules. These modules are divided according to their expected workload. Firstly, there is a Data Processing module [1], whose objective is to perform the operations indicated by the user on the data. Secondly, a model training module [2] has been detected, which is in charge of generating the models indicated by the user and performing the training with the required database. Finally, a Facade module [9] is needed, in charge of acting as a facade and performing the less expensive operations such as user management and management of the different files on the server.

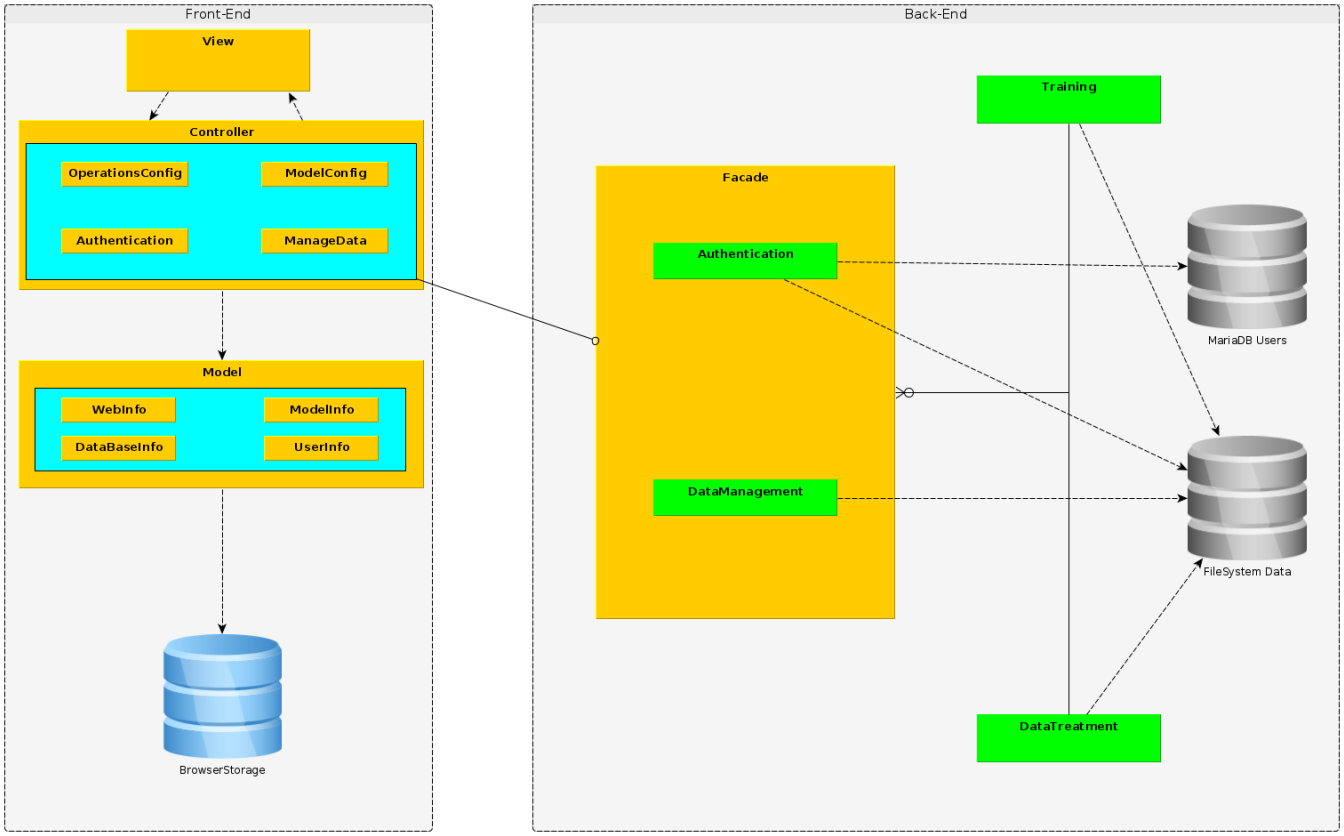

Figure 1. Data processing and model training server architecture.

A possible implementation of this architecture can be found on the GitHub repository https://github.com/braiscgaldo/NIR-Lab-2.0 (accessed on 3 July 2021).

4. Discussion

A scheme has been defined for a server capable of performing the data processing and model training tasks in a distributed and on-demand manner. This offers a number of advantages over other systems such as Weka. The latter performs these tasks in a single instance, which causes the resources of the machine in which it is executed to be depleted due to the fact that it must manage all the functionalities present in the system.

This approach offers the possibility of running on cloud services such as AWS [10], Azure [11], or Google Cloud [12]. This architecture enables the replication of nodes as needed for the execution of data processing or model training in a unique way, which offers a great advantage over desktop applications whose only source of computational power is the computer itself.

5. Future Work

This project presents numerous avenues for future work. One of these possible developments is motivated by the extension of the type of machine learning models. It would be straightforward to extend the set of models composed only of ANN to other algorithms such as SVM, KNN, RF, or LDA.

It is also necessary to highlight the possibility of interactive data processing, visualizing at each point how the different variables defined by the user behave.

Author Contributions: Conceptualization, B.G.; methodology, B.G.; software, B.G.; validation, B.G.; formal 64 analysis, B.G.; investigation, B.G.; resources, B.G., D.R., E.F.-B.; data curation, B.G.; writing—original draft preparation, 65 B.G., D.R., E.F.-B.; writing—review and editing, B.G., D.R., E.F.-B.; visualization, B.G.; supervision, B.G., D.R., E.F.-B.; project 66 administration, B.G.; funding acquisition, B.G., D.R., E.F.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the Galician goverment and EFRD funds (ED431G/01).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: An implementation could be found in https://github.com/braiscgaldo/NIR-Lab-2.0 (accessed on 3 July 2021).

Acknowledgments: The authors would like to thank the support from RNASA-IMEDIR group.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RNA	Redes de Neuronas Artificiales
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
RF	Random Forest
LDA	Linear Discriminant Analysis

References

1. Brandt, S.; Brandt, S. *Data Analysis*; Springer: Berlin/Heidelberg, Germany, 1998.
2. Mohammed, M.; Khan, M.B.; Bashier, E.B.M. *Machine Learning: Algorithms and Applications*; CRC Press: Boca Raton, FL, USA, 2016.
3. Bouckaert, R.R.; Frank, E.; Hall, M.A.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. WEKA—Experiences with a Java Open-Source Project. *J. Mach. Learn. Res.* **2010**, *11*, 2533–2541.
4. Xing, Y.; Zhan, Y. Virtualization and cloud computing. In *Future Wireless Networks and Information Systems*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 305–312.
5. Anderson, C. Docker [software engineering]. *IEEE Softw.* **2015**, *32*, 102-c3. [CrossRef]
6. Cardellini, V.; Colajanni, M.; Yu, P.S. Dynamic load balancing on web-server systems. *IEEE Internet Comput.* **1999**, *3*, 28–39. [CrossRef]
7. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.

8. Krasner, G.E.; Pope, S.T. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *J. Object Oriented Program.* **1988**, *1*, 26–49.
9. Schmidt, D.C.; Stal, M.; Rohnert, H.; Buschmann, F. *Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects*; John Wiley & Sons: Hoboken, NJ, USA, 2013; Volume 2.
10. Amazon, E. Amazon Web Services. 2015; p. 39. Available online: http://aws.amazon.com/es/ec2/ (accessed on 11 November 2020).
11. Chappell, D. Introducing the Azure Services Platform. White Paper, October 2008; p. 1364. Available online: http://www.davidchappell.com (accessed on 13 November 2020).
12. Geewax, J.J.J. *Google Cloud Platform in Action*; Manning: Shelter Island, NY, USA, 2018.