The Journal of Open Source Software

DEEPaaS API: a REST API for Machine Learning and Deep Learning models

Álvaro López García[1]

1 Institute of Physics of Cantabria (IFCA), Spanish National Research Council (CSIC) and University of Cantabria (UC)

DOI: 10.21105/joss.01517

Software

- Review
- Repository
- Archive

Submitted: 23 April 2019
Published: 25 October 2019

Summary

Arguably, Python is one of the most widely used programming languages in data analytics, data mining, artificial intelligence, machine learning and deep learning (Nguyen et al., 2019). DEEPaaS API is a Python package that provides a REST API (Fielding, 2000) that can be used to easily expose the underlying model functionality over HTTP. Developers can plug their applications into DEEPaaS API with minimal requirements and modifications, allowing for easier interaction with the underlying functionality, both for training, validating and testing.

Motivation

Scientists developing machine learning models do not have an easy and common way to share their developed applications with their colleagues or with anybody interested in using them. The whole model (i.e. the code and any configuration assets needed) can be shared, but this requires that the receptors of the model need to have enough knowledge to execute it.

The approach of exposing a model over the network following a server-oriented architecture is being increasingly adopted in Science (Foster, 2005; Zorrilla & García-Saiz, 2013) as they allow to encapsulate different functionalities (like a machine learning model) as a service that will be exploited by other users. In this way, users do not need to deal with model deployment and configuration, but there is a lack of a standard method to expose them, thus scientists need to develop their own methods and tools.

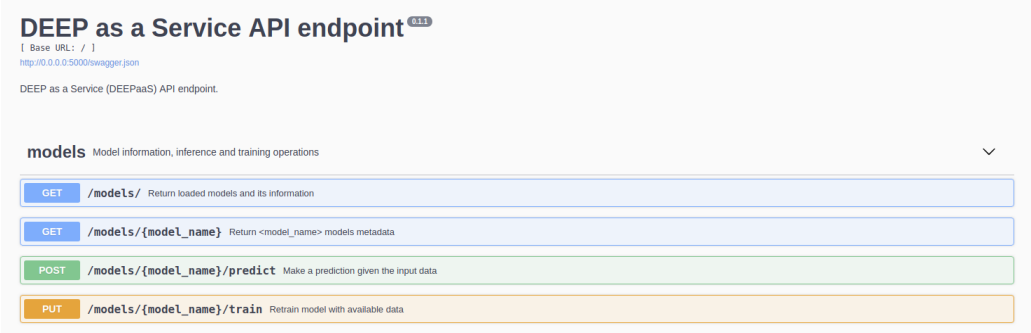
DEEPaaS API aims to alleviate this problem by exposing an API, following the OpenAPI specification that can be easily coupled with a Python-based module, exposing the most common functionality of machine learning and deep learning models. Moreover, DEEPaaS API allows the deployment of the model on top of OpenWhisk, providing an easy way to deploy the model following a serverless or Function as a Service approach.

Overview

DEEPaaS API is written in Python, is compatible with Python 2 and Python 3 versions. In order to integrate a model with DEEPaaS API, we have tried to be as less intrusive as possible. Our approach is based on the dynamic loading of the model trough Python's setuptools entrypoints. As such, developers need to define the corresponding entry points in the `deepaas.model` namespace, with the required functions to be implemented in the namespace documented in the DEEPaaS API documentation. We also provide an abstract

base class that users can extend, but this is not required in order to integrate it. Moreover, DEEPaaS API is not restricted to a simple model, as it allows to load several of them, dynamically creating the different endpoints and routes.

DEEPaaS API is based on Flask-RESTPlus (Haustant, 2018), thus it provides an OpenAPI compatible specification (OpenAPI Initiative, 2018), as well as a documentation endpoint that can be utilized as a simple interface to the model, as shown in the following Figure:



We provide HTTP methods to expose the most common tasks to be performed on a machine learning or deep learning models, such as `train/` for training a model, `predict/` to perform inferences or predictions or `validate/` to perform the validation of the model.

Acknowledgements

This software has been developed within the DEEP-Hybrid-DataCloud (Designing and Enabling E-infrastructures for intensive Processing in a Hybrid DataCloud) project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 777435.

References

Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* (PhD thesis). University of California, Irvine.

Foster, I. (2005). Service-oriented science. *Science*, *308*(5723), 814–817. doi:10.1126/science.1110411

Haustant, A. (2018). Flask-restplus. Retrieved from https://flask-restplus.readthedocs.io/en/stable/

Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., Heredia, I., Malík, P., et al. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artificial Intelligence Review*. doi:10.1007/s10462-018-09679-z

OpenAPI Initiative. (2018). OpenAPI specification. Retrieved from https://github.com/OAI/OpenAPI-Specification

Zorrilla, M., & García-Saiz, D. (2013). A service oriented architecture to provide data mining services for non-expert data miners. *Decision Support Systems*, *55*(1), 399–411. doi:10.1016/j.dss.2012.05.045