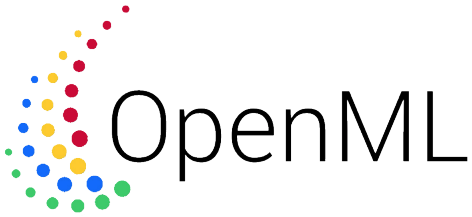
July 3, 2019

Software Transfer Document

Version 1.0.0



Members
Andrei Danila | 1034559
Bogdan Enache | 1035066
Gergana Goncheva | 1037010
Loïc Alexander Hijl | 1002745
Adrian-Stefan Mares | 0993873
Veselin Minev | 1014541
Thanh-Dat Nguyen | 1036672
Antoine Labasse Lutou Nijhuis | 1016657
Claudiu-Teodor Nohai | 1038442
Dragos Mihai Serban | 1033859
Tsvetan Zahariev | 1035269
Sonya Zarkova | 1034611

Project managers
Yuxuan Zhang
Stefan Tanja

Supervisor
Erik Luit
Ion Barosan

Customer
Joaquin Vanschoren

Abstract

This document represents the Software Transfer Document (STD) developed by OpenML Support Squad for the project "Sharing deep learning models". It contains all the necessary information required to establish a successful project transfer of Deep Learning (DL) Extension Library for OpenML to the client Joaquin Vanschoren. OpenML is a platform for sharing machine learning models, datasets, tasks and benchmarks. This STD shows in detail how to set up and use the DL Extension Library. This document complies with the Software Engineering Standard, defined by the European Space Agency (ESA) [1].

Contents

Document Status Sheet

Document Title: Software Transfer Document
Identification: STD/1.0.0
Version: 1.0.0
Authors: A.Danila, B.Enache, G.Goncheva, L.A.Hijl, A.Mares, V.Minev, T.Nguyen, A.L.L.Nijhuis, C.Nohai, D.M.Serban, T.Zahariev, S.Zarkova

Document History

Version	Date	Author(s)	Summary
0.0.1	25-04-2019	T. Zahariev	Created initial document
0.0.2	13-05-2019	L.A. Hijl	Setup Layout
0.0.3	17-05-2019	L.A. Hijl	Added chapters
0.0.4	05-06-2019	G. Goncheva, T. Zahariev	Added configuration item list and abstract
0.0.5	18-06-2019	S.Zarkova	Proofreading and corrections
0.0.6	26-06-2019	T.Zahariev	Added software problem problems, change and modification requests
0.1.0	26-06-2019	All authors	Fixed feedback from supervisor
1.0.0	30-06-2019	All authors	Final edit

1 | Introduction

1.1 Purpose

This document contains the build and installation procedures needed to install and use the DL Extension Library from the project "Sharing deep learning models", which represents an extension for the OpenML platform. The information presented here will be enough for researchers and machine learning specialists / enthusiasts to build and use the DL Extension Library. Furthermore, this document lists all of the items developed by OpenML Support Squad and their transfer to the client - Joaquin Vanschoren. Additionally, this STD provides a summary of the acceptance test in Acceptance Test Plan (ATP) document [2] along with any software problem reports, changes requests and modification reports that have arisen during the transfer phase.

The purpose of project "Sharing deep learning models" is to create an extension for the already existent OpenML Python Application Programming Interface (API) and provide a visualization module. The extension, called DL Extension Library, represents the biggest part of the product that OpenML Support Squad has to deliver. DL Extension Library will enable sharing deep learning models to the OpenML platform. The visualization module allows visualization of a model's structure and performance metrics in Dash.

1.2 Scope

OpenML is an open-source online machine learning platform for sharing and organizing data, machine learning algorithms and experiments. It allows people all over the world to collaborate and build on each other's latest findings, data or results. OpenML is designed to create a seamless network that can be integrated into existing environments, regardless of the tools and infrastructure used. It allows users to upload datasets, models (called OpenML flows) and tasks for OpenML flows. It further provides support for uploading runs that combine a model with a task for it, to allow users to compare results.

The contribution of OpenML Support Squad consists of two parts: creating the DL Extension Library and the visualization module.

The DL Extension Library was built to provide support for the following deep learning libraries and specifications on the OpenML platform: Keras, PyTorch, Apache MXNet (MXNet) and Open Neural Network eXchange (ONNX). These libraries have already been created by their respective developers and, as such, are not part of this project and are not under the responsibility of OpenML Support Squad.

The visualization module was built to serve as a prototype which can be integrated into the website. It allows users to visualize their models and model results.

1.3 Definitions and Abbreviations

1.3.1 Definitions

DL Extension Library	The product to be created in order to convert the deep learning models and the model transfer specifications (i.e. ONNX and MLflow) into a format supported by the OpenML platform.
OpenML Python API	"A connector to the collaborative machine learning platform OpenML.org. The OpenML Python package allows to use datasets and tasks from OpenML together with scikit-learn and share the results online" [3].
OpenML flow	The representation of untrained machine learning models in the OpenML platform.
Apache MXNet	"Apache MXNet is an open-source deep learning software framework, used to train, and deploy deep neural networks. It is scalable, allowing for fast model training, and supports multiple programming languages (C++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl, and Wolfram Language)" [4].
Dash	"Dash is a user interface library for creating analytical web applications" [5].
Dataset	"Datasets are pretty straight-forward. They simply consist of a number of rows, also called instances, usually in tabular form" [3].
Deep learning model	A neural network with more than three layers. Deep learning is a subset of machine learning.
Joaquin Vanschoren	Client of this project. Founder of OpenML.
Keras	"Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano" [6].
Machine learning model	A model that performs a “task without explicit instructions, relying on patterns and inference” [7] through learning from data.
Neural Network	"Computing systems vaguely inspired by the biological neural networks that constitute animal brains." It is "a framework for many different machine learning algorithms" [8].
OpenML	An online machine learning platform for sharing and organizing data, machine learning algorithms and data experiments.
OpenML Support Squad	Name of the development team.
Project "Sharing deep learning models"	The project on which the OpenML Support Squad is working on.

Python Package Index	"The Python Package Index (PyPI) is a repository of software for the Python programming language. It helps in finding and installing software developed and shared by the Python community" [9].
PyTorch	"An open source deep learning platform that provides a seamless path from research prototyping to product deployment" [10].
Run	"A run is a particular flow, that is algorithm, with a particular parameter setting, applied to a particular task" [3].
Task	"A task consists of a dataset, together with a machine learning task to perform, such as classification or clustering and an evaluation method. For supervised tasks, this also specifies the target column in the data" [3].

1.3.2 Abbreviations

ADD	Architectural Design Document.
API	Application Programming Interface.
ATP	Acceptance Test Plan.
DL	Deep Learning.
ESA	European Space Agency.
MXNet	Apache MXNet.
ONNX	Open Neural Network eXchange.
SRD	Software Requirements Document.
STD	Software Transfer Document.
SUM	Software User Manual.
URD	User Requirements Document.
UTP	Unit Test Plan.

1.4 List of references

[1] Software Standardisation and Control. *ESA software engineering standards. 1991*.

[2] OpenML Support Squad. *DL Extension Library, ATP Acceptance Test Plan version 1.0.0*.

[3] OpenML. *OpenML Documentation*. URL: https://docs.openml.org/ (visited on 04/26/2019).

[4] Wikipedia. *Apache MXNet*. URL: https://en.wikipedia.org/wiki/Apache_MXNet (visited on 04/26/2019).

[5] Kyle Kelley. *Introducing Dash*. URL: https://medium.com/@plotlygraphs/introducing-dash-5ecf7191b503 (visited on 06/05/2019).

[6] Keras. *Keras homepage*. URL: https://keras.io/ (visited on 04/26/2019).

[7] Wikipedia. *Machine Learning*. URL: https://en.wikipedia.org/wiki/Machine_learning (visited on 04/26/2019).

[8] Wikipedia. *Artificial Neural Network*. URL: https://en.wikipedia.org/wiki/Artificial_neural_network (visited on 04/30/2019).

[9] *Python Package Index*. URL: https://pypi.org/ (visited on 05/20/2019).

[10] PyTorch. *PyTorch homepage*. URL: https://pytorch.org/ (visited on 04/26/2019).

[11] OpenML Support Squad. *DL Extension Library, User Requirements Document (URD) User Requirement Document version 1.0.0.*

[12] OpenML Support Squad. *DL Extension Library, Software Requirements Document (SRD) Software Requirement Document version 1.0.0.*

[13] OpenML Support Squad. *DL Extension Library, Architectural Design Document (ADD) Architectural Design Document version 1.0.0.*

[14] OpenML Support Squad. *DL Extension Library, Software User Manual (SUM) Software User Manual version 1.0.0.*

[15] OpenML Support Squad. *DL Extension Library, Unit Test Plan (UTP) User Test Plan version 1.0.0.*

2 | Build procedure

The DL Extension Library from the project "Sharing deep learning models" is built entirely on the scripting language Python 3.5/3.6/3.7. Furthermore, the DL Extension Library does not contain any components that need compilation or building, so all relevant information on how to install and set it up for use is described in the following chapter 3: "Installation Procedure".

The DL Extension Library is part of the OpenML Python API that will be imported by the users in their Python project. The OpenML Python API handles all interactions with the OpenML platform. Whenever necessary, the OpenML Python API calls the DL Extension Library in order to convert the deep learning model to an OpenML flow, the format supported by the OpenML platform. The OpenML Support Squad is not responsible for the development and maintenance of the OpenML Python API. The OpenML Support Squad has not modified the OpenML Python API since it is outside of the scope of the project "Sharing deep learning models".

The visualization module was built using Dash and the OpenML Python API. In order to start the local Dash website, the `visualizer.py` needs to be compiled and run. The visualizer.py is part of the OpenML Python API and can be found inside this library. To open the local website, use the IP and port provided in the terminal, after running `visualizer.py`.

3 | Installation Procedure

This chapter describes the installation procedures for all four separate extensions of the DL Extension Library i.e. Keras, PyTorch, ONNX, MXNet and the Dash visualization tool.

3.1 Keras extension library installation

We assume that you have the latest version of `Python Package Index (PyPI)` and `Python 3.5/3.6/3.7`.

1. Open a terminal
2. Execute: `pip install https://codeload.github.com/adriansmares/openml-deeplearning/zip/master`
3. Execute: `pip install https://codeload.github.com/adriansmares/openml-deeplearning/zip/master#egg=openml[dlkeras]`

3.2 PyTorch extension library installation

We assume that you have the latest version of `Python Package Index (PyPI)` and a 64 bit version of `Python 3.5/3.6/3.7`.

1. Open a terminal
2. Execute: `pip install https://codeload.github.com/adriansmares/openml-deeplearning/zip/master`
3. Execute: `pip install https://codeload.github.com/adriansmares/openml-deeplearning/zip/master#egg=openml[dltorch]`

3.3 ONNX extension library installation

We assume that you have the latest version of `Python Package Index (PyPI)` and `Python 3.5/3.6/3.7`.

1. Open a terminal
2. Execute: `pip install https://codeload.github.com/adriansmares/openml-deeplearning/zip/master`
3. Execute: `pip install https://codeload.github.com/adriansmares/openml-deeplearning/zip/master#egg=openml[dlonnx]`

3.4 MXNet extension library installation

We assume that you have the latest version of `Python Package Index (PyPI)` and `Python 3.5/3.6/3.7`.

1. Open a terminal
2. Execute: `pip install https://codeload.github.com/adriansmares/openml-deeplearning/zip/master`
3. Execute: `pip install https://codeload.github.com/adriansmares/openml-deeplearning/zip/master#egg=openml[dlmxnet]`

3.5 Dash visualization installation

We assume that you have the latest version of `Python Package Index (PyPI)` and `Python 3.5/3.6/3.7`.

1. Open a terminal.
2. Execute: `git clone https://github.com/adriansmares/openml-deeplearning.git`
3. Execute: `pip install keras.`
4. Execute: `pip install mxnet==1.5.0b20190619.`
5. Execute: `pip install onnx==1.2.1.`
6. Execute the command for installing PyTorch according to your preferences. The command can be found on the following website https://pytorch.org/ .
7. Execute: `pip install pydot.`
8. Execute: `pip install onnxmltools.`
9. Download Graphviz from the following link: https://graphviz.gitlab.io/download/ according to your preferences and install it
10. Ensure that dot.exe (from Graphviz) is added to the system PATH.
11. Run the `visualizer.py` to launch the local Dash website.

4 | Configuration Item List

In this chapter all items delivered by the OpenML Support Squad are listed. All documents related to the project are provided in a PDF format. The software is delivered in the form of a source code, which is also part of the Github project of OpenML Python API.

4.1 Documentation

The following documents are delivered:

- User Requirements Document (URD) [11]
- Software Requirements Document (SRD) [12]
- Architectural Design Document (ADD) [13]
- Software Transfer Document (STD) [this document]
- Software User Manual (SUM) [14]

4.2 Test plans

- Acceptance Test Plan (ATP) [2]
- Unit Test Plan (UTP) [15]

4.3 Software

- Keras extension package
- PyTorch extension package
- ONNX extension package
- MXNet extension package
- Dash visualization package

5 | Acceptance Test Report Summary

This section describes the outcome of the acceptance test.

Description: All tests described in the ATP have been executed during the acceptance test, which was performed on Friday, 21^{th} of June 2019, from 10:10 until 12:00. All acceptance tests passed. At the night before the acceptance test Dash had an update which caused a dependency issue. This was fixed immediately during testing by downgrading the related dependency of Dash to the version before the update. With the exception of the just explained issue, no problems occurred during the test session. The people present during the acceptance test were the customer, the supervisor, the project managers and the OpenML Support Squad. A test report has been made in section 5 of the ATP [2] to show the verdict for every test performed during the acceptance test.

6 | Software Problem Reports

This chapter lists any problems that have arisen during the transfer phase.

6.1 Visualization issues

During the transfer phase, a single problem occurred in the visualization module - a dependency issue. Dash had released an update at the night before the acceptance test. Thus, the visualization was unable to start with the given release. The issue was fixed by downgrading the version of Dash to the one before the latest update.

7 | Software Change Requests

This section is left empty, since no change requests arose during the transfer phase.

8 | Software Modification Reports

This section is left empty, since no change requests arose during the transfer phase.

Bibliography

[1] Software Standardisation and Control. *ESA software engineering standards. 1991.*

[2] OpenML Support Squad. *DL Extension Library, ATP Acceptance Test Plan version 1.0.0.*

[3] OpenML. *OpenML Documentation.* URL: https://docs.openml.org/ (visited on 04/26/2019).

[4] Wikipedia. *Apache MXNet.* URL: https://en.wikipedia.org/wiki/Apache_MXNet (visited on 04/26/2019).

[5] Kyle Kelley. *Introducing Dash.* URL: https://medium.com/@plotlygraphs/introducing-dash-5ecf7191b503 (visited on 06/05/2019).

[6] Keras. *Keras homepage.* URL: https://keras.io/ (visited on 04/26/2019).

[7] Wikipedia. *Machine Learning.* URL: https://en.wikipedia.org/wiki/Machine_learning (visited on 04/26/2019).

[8] Wikipedia. *Artificial Neural Network.* URL: https://en.wikipedia.org/wiki/Artificial_neural_network (visited on 04/30/2019).

[9] *Python Package Index.* URL: https://pypi.org/ (visited on 05/20/2019).

[10] PyTorch. *PyTorch homepage.* URL: https://pytorch.org/ (visited on 04/26/2019).

[11] OpenML Support Squad. *DL Extension Library, URD User Requirement Document version 1.0.0.*

[12] OpenML Support Squad. *DL Extension Library, SRD Software Requirement Document version 1.0.0.*

[13] OpenML Support Squad. *DL Extension Library, ADD Architectural Design Document version 1.0.0.*

[14] OpenML Support Squad. *DL Extension Library, SUM Software User Manual version 1.0.0.*

[15] OpenML Support Squad. *DL Extension Library, UTP User Test Plan version 1.0.0.*