Open for Innovation
KNIME

KNIME Deep Learning Integration Installation Guide

KNIME AG, Zurich, Switzerland
Version 3.7 (last updated on 2022-01-11)

Table of Contents

Introduction

This document describes how to install the KNIME Deep Learning Integrations.

These integrations bring deep learning capabilities to KNIME Analytics Platform, which allow you to read, create, edit, train, and execute deep neural networks within KNIME Analytics Platform.

KNIME Deep Learning Integrations

Three different deep learning libraries have been integrated:

KNIME Keras Integration

The KNIME Keras Integration utilizes the Keras deep learning framework to enable users to read, write, train, and execute Keras deep learning networks within KNIME. Furthermore, you can also build custom deep learning networks directly in KNIME via the Keras layer nodes.

KNIME Tensor Flow Integration

The KNIME TensorFlow Integration provides access to the powerful machine learning library TensorFlow* within KNIME. It enables you to read, write, train, and execute TensorFlow networks directly in KNIME. You can also convert your Keras networks to TensorFlow networks with this extension for even greater flexibility.

* TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc.

KNIME Deeplearning4j Integration

The KNIME Deeplearning4j Integration integrates the Deeplearning4j library into KNIME, which provides deep learning capabilities in Java. Within KNIME this means you can read, write, train, execute, and build Deeplearning4j networks.

KNIME Keras Integration Installation

This section describes how to install the KNIME Keras Integration for use with KNIME Analytics Platform.

Similar to the KNIME Python Integration, the KNIME Keras Integration internally uses an existing Python installation, which is installed alongside of KNIME and depends on certain Python packages that have to be installed. In this document we show how to setup the necessary Python installation and how to install the KNIME Keras Integration.

Python Installation

1. KNIME executes Keras in a local Python installation, which has to be set up manually. If you have already set up your Python installation, you can skip this step. We **highly recommend** setting up a `conda` environment as described in our Python Installation Guide. In this guide you can either follow the Quickstart Section, or follow the Full Installation Section if you'd like to have a detailed walkthrough.

   > ❗ The KNIME Keras Integration only supports Python 3. Therefore it is necessary to set up Python 3.

2. Additionally to the Python packages installed in the previous step, the KNIME Keras Integration depends on further packages. If you are using Anaconda, they can be installed with a single command. The required packages are `h5py=2.8`, `tensorflow-mkl=1.12` (`tensorflow=1.12` for GPU), and `keras=2.2.4` (`keras-gpu=2.2.4` for GPU). For the CPU version use the command:

   ```
   conda install --name <your_env_name> h5py=2.8 tensorflow-mkl=1.12 keras=2.2.4
   ```

   and for the GPU version the command:

   ```
   conda install --name <your_env_name> h5py=2.8 tensorflow=1.12 keras-gpu=2.2.4
   ```

   where `<your_env_name>` is the name of your conda environment.

   A general description about how to install further Python packages using Anaconda can be found here.

The packages `keras` and `keras-gpu` are only available with newer conda versions (minimum conda version: 4.3.30). If you have an older version, you can update `conda` using the command `conda update conda`.

In case you run into problems within KNIME due to missing Keras dependencies or if you want to double check that everything was set up correctly, here is a list of Keras dependencies of Keras (these should have been installed automatically):

- h5py (version: 2.8)
- numpy (version: 1.15)
- pyyaml (version: 3.13)
- scipy (version: 1.1)
- six (minimum version: 1.11)
- tensorflow or tensorflow-gpu (version: 1.12)

If you are using Anaconda, you can check whether these dependencies are installed by running:

```
conda list -n <your_env_name>
```

where `<your_env_name>` is the name of your conda environment.

Installing the KNIME Keras Integration

The KNIME Keras Integration extension can be installed using the KNIME Analytics Platform Update Site where it is listed under KNIME Labs Extensions. Alternatively, enter *keras* into the search box.

If Keras is not displayed as an available deep learning back end, you may need to restart KNIME Analytics Platform.

After you have set up Python and the KNIME Keras Integration is installed, you are ready to go.

Extensions

For the KNIME Deep Learning Integration there are the following extensions:

- *KNIME Image Processing - Deep Learning Extension*: Support for Images from KNIME Image Processing. The extension can be found in the Trusted Community Contributions Update Site.

GPU Support

Keras is able to accelerate deep learning models using a compatible NVIDIA® GPU via TensorFlow. Most of the required dependencies for GPU (i.e. CUDA® and cuDNN) will be automatically installed by Anaconda when installing the `conda` packages `tensorflow=1.12` and `keras-gpu=2.2.4`. The only additional requirement that needs to be installed manually is the latest version of the NVIDIA® GPU driver.

KNIME TensorFlow Integration Installation

This section describes how to install the KNIME TensorFlow Integration for use with KNIME Analytics Platform.

Installation

The KNIME TensorFlow Integration can be installed using the KNIME Analytics Platform Update Site where it is listed under KNIME Labs Extensions. Alternatively, enter *tensorflow* into the search box.

If TensorFlow is not displayed as an available deep learning back end, you may need to restart KNIME Analytics Platform.

Advanced

If you want to use TensorFlow models in DL Python nodes with custom Python scripts, you need a Python installation alongside KNIME. This Python installation has the same requirements as the KNIME Keras Integration. In order to install Python and the necessary dependencies, please refer to the Python Installation Section.

GPU Support

TensorFlow is able to accelerate deep learning models using a compatible NVIDIA® GPU. For detailed instructions on how to set up GPU support on your system, please refer to the official documentation. A list of supported GPU devices is also shown on the TensorFlow documentation page.

Due to limitations by TensorFlow, the GPU support for the KNIME TensorFlow Integration can't be used on Mac. Only Linux and Windows are supported.

The versions of the required software for the GPU support of TensorFlow must be compatible with the installed TensorFlow version. The KNIME Keras Integration uses TensorFlow version `1.12`.

For TensorFlow `1.12`, the following NVIDIA® software must be installed on your system:

- NVIDIA® GPU drivers - CUDA® 9.0 requires 384.x or higher.
- CUDA® Toolkit - TensorFlow supports CUDA® 9.0. Detailed installation instructions can be found here.
- cuDNN - (version >= 7.2) - select cuDNN v7.4.1 (Nov 8, 2018), for CUDA® 9.0. Detailed installation instructions can be found here.

KNIME Deeplearning4j Installation

This section explains how to install KNIME Deeplearning4j Integration to be used with KNIME Analytics Platform.

Installation

The KNIME Deeplearning4j Integration can be installed using the KNIME Analytics Platform Update Site where it is listed under KNIME Labs Extensions.

GPU Support

The KNIME Deeplearning4j Integration supports acceleration of deep learning models using a compatible NVIDIA® GPU. For the GPU support, CUDA® Toolkit 8.0 must be installed on your system. Detailed installation instructions can be found here.

Known Issues

Older versions of KNIME Analytics Platform, used together with older versions of the KNIME

Deep Learning Integrations dependencies, may lead to errors. These errors can be fixed by updating KNIME Analytics Platform and the KNIME Deep Learning Integrations to the latest versions. Furthermore, if you are using Python, please make sure to use the recommended conda package versions. The following issues are currently known:

- *Node fails with error* ***AttributeError: '[..]' object has no attribute 'inbound_nodes'*** *at the bottom of a Python traceback in the KNIME log (KNIME 3.5.x only).*

  Keras version 2.1.3 introduced breaking changes that were adapted in KNIME version 3.6.0. Please upgrade KNIME to version 3.6.0 or downgrade Keras to version 2.1.2 or below (minimum version: 2.0.7).

- *Node fails with error* ***UnicodeEncodeError: 'ascii' codec can't encode character [..] in position [..]: ordinal not in range(128)*** *at the bottom of a Python traceback in the KNIME log.*

  This error may occur when using Keras version 2.1.2 to load a Keras network that was saved using an older Keras version. Make sure not to use Keras 2.1.2 in such cases.

- *Node fails with both an error* ***SystemError: unknown opcode*** *and a warning* ***XXX lineno: [..], opcode: [..]*** *in the KNIME log.*

  This is a Python related error that occurs when loading a Keras network containing a Lambda expression (e.g. within a Lambda layer) that was saved using a different Python version. Make sure you use the same Python version for saving and loading the same network.

- *DL Keras Network Learner fails with error* ***AttributeError: 'int' object has no attribute 'dtype'*** *at the bottom of a Python traceback in the KNIME log when the Clip norm option is enabled in the node dialog and Keras (TensorFlow) is the selected back end.*

  This is a TensorFlow related error that only occurs in very specific situations. Try to use a different Keras back end to work around this issue.

- *DL Python Network Executor scripting node outputs wrong numerical values when using Flatbuffers serialization library (KNIME 3.6.1 and older).*

  Flatbuffers in KNIME does not support float32 data at the moment. We recommend using Apache Arrow as serialization library instead. This option can be changed in KNIME via *File* → *Preferences* → *KNIME* → *Python* → *Serialization library*.

Open for Innovation
KNIME

KNIME AG
Technoparkstrasse 1
8005 Zurich, Switzerland
www.knime.com
info@knime.com