Keras: The Python Deep Learning library

You have just found Keras.

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.*

Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

Read the documentation at Keras.io.

Keras is compatible with: **Python 2.7-3.6**.

Guiding principles

- **User friendliness.** Keras is an API designed for

human beings, not machines. It puts user experience front and center. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.

- **Modularity.** A model is understood as a sequence or a graph of standalone, fully-configurable modules that can be plugged together with as few restrictions as possible. In particular, neural layers, cost functions, optimizers, initialization schemes, activation functions, regularization schemes are all standalone modules that you can combine to create new models.
- **Easy extensibility.** New modules are simple to add (as new classes and functions), and existing modules provide ample examples. To be able to easily create new modules allows for total expressiveness, making Keras suitable for advanced research.
- **Work with Python**. No separate models configuration files in a declarative format. Models are described in Python code, which is compact, easier to debug, and allows for ease of extensibility.

Getting started: 30 seconds to Keras

The core data structure of Keras is a **model**, a way to organize layers. The simplest type of model is the `Sequential` model, a linear stack of layers. For more complex architectures, you should use the Keras functional API, which allows to build arbitrary graphs of layers.

Here is the `Sequential` model:

```
from keras.models import Sequential

model = Sequential()
```

Stacking layers is as easy as `.add()`:

```
from keras.layers import Dense

model.add(Dense(units=64, activation='relu',
input_dim=100))
model.add(Dense(units=10, activation='softmax'))
```

Once your model looks good, configure its learning process with `.compile()`:

```
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])
```

If you need to, you can further configure your optimizer. A core principle of Keras is to make things reasonably simple, while allowing the user to be fully in control when they need to (the ultimate control being the easy extensibility of the source code).

```
model.compile(loss=keras.losses.categorical_crossent


optimizer=keras.optimizers.SGD(lr=0.01,
momentum=0.9, nesterov=True))
```

You can now iterate on your training data in batches:

```
# x_train and y_train are Numpy arrays --just
like in the Scikit-Learn API.
model.fit(x_train, y_train, epochs=5,
batch_size=32)
```

Alternatively, you can feed batches to your model manually:

```
model.train_on_batch(x_batch, y_batch)
```

Evaluate your performance in one line:

```
loss_and_metrics = model.evaluate(x_test, y_test,
batch_size=128)
```

Or generate predictions on new data:

```
classes = model.predict(x_test, batch_size=128)
```

Building a question answering system, an image classification model, a Neural Turing Machine, or any other model is just as fast. The ideas behind deep learning are simple, so why should their implementation be painful?

For a more in-depth tutorial about Keras, you can check out:

- Getting started with the Sequential model
- Getting started with the functional API

In the examples folder of the repository, you will find more advanced models: question-answering with memory networks, text generation with stacked LSTMs, etc.

Installation

Before installing Keras, please install one of its backend engines: TensorFlow, Theano, or CNTK. We recommend the TensorFlow backend.

- TensorFlow installation instructions.
- Theano installation instructions.
- CNTK installation instructions.

You may also consider installing the following **optional dependencies**:

- cuDNN (recommended if you plan on running Keras on GPU).
- HDF5 and h5py (required if you plan on saving Keras models to disk).
- graphviz and pydot (used by visualization utilities to plot model graphs).

Then, you can install Keras itself. There are two ways to install Keras:

- **Install Keras from PyPI (recommended):**

```
sudo pip install keras
```

If you are using a virtualenv, you may want to avoid using sudo:

```
pip install keras
```

- **Alternatively: install Keras from the GitHub source:**

First, clone Keras using `git`:

```
git clone https://github.com/keras-team/keras.git
```

Then, `cd` to the Keras folder and run the install command:

```
cd keras
sudo python setup.py install
```

Configuring your Keras backend

By default, Keras will use TensorFlow as its tensor manipulation library. Follow these instructions to configure the Keras backend.

Support

You can ask questions and join the development discussion:

- On the Keras Google group.
- On the Keras Slack channel. Use this link to request an invitation to the channel.

You can also post **bug reports and feature requests**

(only) in GitHub issues. Make sure to read our guidelines first.

Why this name, Keras?

Keras (κέρας) means *horn* in Greek. It is a reference to a literary image from ancient Greek and Latin literature, first found in the *Odyssey*, where dream spirits (*Oneiroi*, singular *Oneiros*) are divided between those who deceive men with false visions, who arrive to Earth through a gate of ivory, and those who announce a future that will come to pass, who arrive through a gate of horn. It's a play on the words κέρας (horn) / κραίνω (fulfill), and ἐλέφας (ivory) / ἐλεφαίρομαι (deceive).

Keras was initially developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System).

> *"Oneiroi are beyond our unravelling --who can be sure what tale they tell? Not all that men look for comes to pass. Two gates there are that give passage to fleeting Oneiroi; one is made of horn, one of ivory. The Oneiroi that pass through sawn ivory are deceitful, bearing a message that will not be fulfilled; those that come out through polished horn have truth behind them, to be accomplished for men who see them."* Homer, Odyssey 19. 562 ff (Shewring translation).

Next □