Datasets

CIFAR10 small image classification

Dataset of 50,000 32x32 color training images, labeled over 10 categories, and 10,000 test images.

Usage:

```
from keras.datasets import cifar10

(X_train, y_train), (X_test, y_test) = cifar10.load_data()
```

- **Return:**
  - 2 tuples:
    - **X_train, X_test**: uint8 array of RGB image data with shape (nb_samples, 3, 32, 32).
    - **y_train, y_test**: uint8 array of category labels (integers in range 0-9) with shape (nb_samples,).

CIFAR100 small image classification

Dataset of 50,000 32x32 color training images, labeled over 100 categories, and 10,000 test images.

Usage:

```
from keras.datasets import cifar100

(X_train, y_train), (X_test, y_test) = cifar100.load_data(label_mode='fine')
```

- **Return:**
  - 2 tuples:
    - **X_train, X_test**: uint8 array of RGB image data with shape (nb_samples, 3, 32, 32).
    - **y_train, y_test**: uint8 array of category labels with shape (nb_samples,).
- **Arguments:**
  - **label_mode**: "fine" or "coarse".

IMDB Movie reviews sentiment classification

Dataset of 25,000 movies reviews from IMDB, labeled by sentiment (positive/negative). Reviews have been preprocessed, and each review is encoded as a sequence of word indexes (integers). For convenience, words are indexed by overall frequency in the dataset, so that for instance the integer "3" encodes the 3rd most frequent word in the data. This allows for quick filtering operations such as: "only consider the top 10,000 most common words, but eliminate the top 20 most common words".

As a convention, "0" does not stand for a specific word, but instead is used to encode any unknown word.

Usage:

```
from keras.datasets import imdb

(X_train, y_train), (X_test, y_test) = imdb.load_data(path="imdb.pkl",
                                                      nb_words=None,
                                                      skip_top=0,
                                                      maxlen=None,
                                                      test_split=0.1)
```

- **Return:**
  - 2 tuples:
    - **X_train, X_test**: list of sequences, which are lists of indexes (integers). If the nb_words argument was specific, the maximum possible index value is nb_words-1. If the maxlen argument was specified, the largest possible sequence length is maxlen.
    - **y_train, y_test**: list of integer labels (1 or 0).
- **Arguments:**
  - **path**: if you do have the data locally (at `'~/.keras/datasets/' + path`), if will be downloaded to this location (in cPickle format).
  - **nb_words**: integer or None. Top most frequent words to consider. Any less frequent word will appear as 0 in the sequence data.
  - **skip_top**: integer. Top most frequent words to ignore (they will appear as 0s in the sequence data).
  - **maxlen**: int. Maximum sequence length. Any longer sequence will be truncated.
  - **test_split**: float. Fraction of the dataset to be used as test data.
  - **seed**: int. Seed for reproducible data shuffling.

Reuters newswire topics classification

Dataset of 11,228 newswires from Reuters, labeled over 46 topics. As with the IMDB dataset, each wire is encoded as a sequence of word indexes (same conventions).

Usage:

```
from keras.datasets import reuters

(X_train, y_train), (X_test, y_test) = reuters.load_data(path="reuters.pkl",
                                                         nb_words=None,
                                                         skip_top=0,
                                                         maxlen=None,
                                                         test_split=0.1)
```

The specifications are the same as that of the IMDB dataset.

This dataset also makes available the word index used for encoding the sequences:

```
word_index = reuters.get_word_index(path="reuters_word_index.pkl")
```

- **Return:** A dictionary where key are words (str) and values are indexes (integer). eg. `word_index["giraffe"]` might return `1234`.
- **Arguments:**
    - **path**: if you do have the index file locally (at `'~/.keras/datasets/' + path`), if will be downloaded to this location (in cPickle format).

MNIST database of handwritten digits

Dataset of 60,000 28x28 grayscale images of the 10 digits, along with a test set of 10,000 images.

Usage:

```
from keras.datasets import mnist

(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

- **Return:**
    - 2 tuples:
        - **X_train, X_test**: uint8 array of grayscale image data with shape (nb_samples, 28, 28).
        - **y_train, y_test**: uint8 array of digit labels (integers in range 0-9) with shape (nb_samples,).
- **Arguments:**
    - **path**: if you do have the index file locally (at `'~/.keras/datasets/' + path`), if will be downloaded to this location (in cPickle format).