データセット

CIFAR10 画像分類

10のクラスにラベル付けされた，50,000枚の32x32訓練用カラー画像，10,000枚のテスト用画像のデータセット.

使い方:

```
from keras.datasets import cifar10

(x_train, y_train), (x_test, y_test) = cifar10.load_data()
```

- **戻り値**:
  - 2つのタプル:
    - **x_train, x_test**: shape (num_samples, 3, 32, 32)または(num_samples, 32, 32, 3)のRGB画像データのuint8配列です．これはバックエンド設定の `image_data_format` が `channels_first` と `channels_last` のいずれなのかによって決まります.
    - **y_train, y_test**: shape (num_samples,) のカテゴリラベル(0-9の範囲の整数)のuint8配列.

CIFAR100 画像分類

100のクラスにラベル付けされた，50,000枚の32x32訓練用カラー画像，10,000枚のテスト用画像のデータセット.

使い方:

```
from keras.datasets import cifar100

(x_train, y_train), (x_test, y_test) = cifar100.load_data(label_mode='fine')
```

- **戻り値**:
  - 2つのタプル:
    - **x_train, x_test**: shape (num_samples, 3, 32, 32)または(num_samples, 32, 32, 3)のRGB画像データのuint8配列です．これはバックエンド設定の `image_data_format` が `channels_first` と `channels_last` のいずれなのかによって決まります.
    - **y_train, y_test**: shape (num_samples,) のカテゴリラベルのuint8配列.
- **引数**:
  - **label_mode**: "fine" または "coarse".

IMDB映画レビュー感情分類

感情 (肯定/否定) のラベル付けをされた，25,000のIMDB映画レビューのデータセット．レビューは前処理済みで，各レビューは単語のインデックス（整数）のシーケンスとしてエンコードされています．便宜上，単語はデータセットにおいての出現頻度によってインデックスされています．そのため例えば，整数"3"はデータの中で3番目に頻度が多い単語にエンコードされます．これによって"上位20個の頻出語を除いた，上位10,000個の頻出語についてのみ考える"というようなフィルタリング作業を高速に行うことができます．

慣例として，"0"は特定の単語を表さずに，未知語にエンコードされます．

使い方:

```
from keras.datasets import imdb

(x_train, y_train), (x_test, y_test) = imdb.load_data(path="imdb.npz",
                                                      num_words=None,
                                                      skip_top=0,
                                                      maxlen=None,
                                                      seed=113,
                                                      start_char=1,
                                                      oov_char=2,
                                                      index_from=3)
```

- **戻り値**:
  - 2つのタプル:
    - **x_train, x_test**: シーケンスのリスト，リストはインデックス（整数）．引数num_wordsに具体的な整数が与えられた場合，取り得るインデックスの最大値はnum_words-1となる．引数maxlenに具体的な数値が与えられた場合，シーケンスの最大長はmaxlenとなる．
    - **y_train, y_test**: 整数ラベル（1または0）のリスト．
- **引数**:
  - **path**: データをローカルに持っている場合 (`'~/.keras/datasets/' + path`)，cPickleフォーマットではこの位置にダウンロードされます．
  - **num_words**: 整数 または None． 指定された数値だけ上位の頻出語が対象となります．指定された数値より下位の頻出語はシーケンスデータにおいて`oov_char`の値で表現します．
  - **skip_top**: 整数．指定された数値だけ上位の頻出語が無視されます（シーケンスデータにおいて`oov_char`の値で表現します）．
  - **maxlen**: 整数．シーケンスの最大長．最大長より長いシーケンスは切り捨てられます．
  - **seed**: 整数．再現可能なデータシャッフルのためのシード．
  - **start_char**: この文字が系列の開始記号として扱われます． 0は通常パディング用の文字であるため，1以上からセットしてください．
  - **oov_char**: 整数． `num_words`か`skip_top`によって削除された単語をこの値で置換します．
  - **index_from**: 単語のインデックスはこのインデックス以上の数値が与えられます．

ロイターのニュースワイヤー トピックス分類

46のトピックにラベル付けされた，11,228個のロイター のニュースワイヤーのデータセット．IMDBデータセットと同様，各ワイヤーが一連の単語インデックスとしてエンコードされます（同じ慣例に基づく）．

使い方:

```
from keras.datasets import reuters

(x_train, y_train), (x_test, y_test) = reuters.load_data(path="reuters.npz",
                                                         num_words=None,
                                                         skip_top=0,
                                                         maxlen=None,
                                                         test_split=0.2,
                                                         seed=113,
                                                         start_char=1,
                                                         oov_char=2,
                                                         index_from=3)
```

仕様はIMDBデータセットのものに加えて，次のパラメータが追加されます:

- **test_split**: 浮動小数点数．テストデータとして使用するデータセットの割合．

このデータセットはシーケンスをエンコードに使われている単語インデックスを利用できます．

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

- **戻り値**: キーが単語（文字列），値がインデックス（整数）の辞書．
  例，`word_index["giraffe"]`は`1234`が返ります．
- **引数**:
  - **path**: データをローカルに持っていない場合（`'~/.keras/datasets/' + path`），この位置にダウンロードされます．

MNIST 手書き数字データベース

60,000枚の28x28，10個の数字の白黒画像と10,000枚のテスト用画像データセット．

使い方:

```
from keras.datasets import mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

- **戻り値**:
  - 2つのタプル:
    - **x_train, x_test**: shape (num_samples, 28, 28) の白黒画像データのuint8配列．
    - **y_train, y_test**: shape (num_samples,) のカテゴリラベル(0-9の整数)のuint8配列．
- **引数**:

- **path**: データをローカルに持っていない場合は(`'~/.keras/datasets/' + path`)，この位置にダウンロードされます．

Fashion-MNIST ファッション記事データベース

60,000枚の28x28，10個のファッションカテゴリの白黒画像と10,000枚のテスト用画像データセット．このデータセットはMNISTの完全な互換品として使えます．クラスラベルは次の通りです：

ラベル	説明
0	Tシャツ/トップス
1	ズボン
2	プルオーバー
3	ドレス
4	コート
5	サンダル
6	シャツ
7	スニーカー
8	バッグ
9	アンクルブーツ

使い方：

```
from keras.datasets import fashion_mnist

(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
```

- **戻り値**:
  - 2つのタプル:
    - **x_train, x_test**: shape (num_samples, 28, 28) の白黒画像データのuint8配列．
    - **y_train, y_test**: shape (num_samples,) のラベル(0-9の整数)のuint8配列．

ボストンの住宅価格回帰データセット

Carnegie Mellon大学のStatLib ライブラリのデータセット．

サンプルは，1970年代後半におけるボストン近郊の異なる地域の住宅に関する13の属性値を含みます． 予測値は，その地域での住宅価格の中央値（単位はk$）です．

使い方:

```
from keras.datasets import boston_housing

(x_train, y_train), (x_test, y_test) = boston_housing.load_data()
```

- **引数**:
  - **path**: ローカルに保存するパス (~/.keras/datasets).
  - **seed**: テストデータに分ける前にデータをシャッフルするためのシード.
  - **test_split**: テストデータとして使用するデータセットの割合.
- **返り値**: Numpy 配列のタプル: `(x_train, y_train), (x_test, y_test)`.