

評価関数の利用方法

評価関数はモデルの性能を測るために使われます。次のコードのように、モデルをコンパイルする際に `metrics` パラメータとして評価関数を渡して指定します。

```
model.compile(loss='mean_squared_error',
              optimizer='sgd',
              metrics=['mae', 'acc'])
```

```
from keras import metrics

model.compile(loss='mean_squared_error',
              optimizer='sgd',
              metrics=[metrics.mae, metrics.categorical_accuracy])
```

評価関数は**損失関数**とよく似ていますが、評価結果の値が訓練に直接使われることはありません。

渡す `metrics` パラメータには既存の評価関数の名前を引数に与えるか、自分で作った評価関数を渡す事ができます（**カスタマイズ**を参照してください）。

引数

- **y_true**: 真のラベル。Theano/TensorFlowのテンソル
- **y_pred**: 予測値。y_trueと同じshapeのTheano/TensorFlowのテンソル

戻り値

全データ点の平均値を表すスカラー。

利用可能な評価関数

binary_accuracy

```
binary_accuracy(y_true, y_pred)
```

categorical_accuracy

```
categorical_accuracy(y_true, y_pred)
```

sparse_categorical_accuracy

```
sparse_categorical_accuracy(y_true, y_pred)
```

top_k_categorical_accuracy

```
top_k_categorical_accuracy(y_true, y_pred, k=5)
```

カスタマイズ

`(y_true, y_pred)` を引数とし、各データ点に対してスカラを返す関数を評価関数として利用できます:

- **y_true**: 正解ラベル. Theano/TensorFlow テンソル
- **y_pred**: 予測. y_trueと同じ形状のTheano/TensorFlow テンソル

```
import keras.backend as K

def mean_pred(y_true, y_pred):
    return K.mean(y_pred)

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy', mean_pred])
```