

[Docs](#) » [レイヤー](#) » [レイヤーラッパー](#)

TimeDistributed

[\[source\]](#)

```
keras.layers.wrappers.TimeDistributed(layer)
```

このラッパーにより、入力のすべての時間スライスにレイヤーを適用できます。

入力は少なくとも3次元である必要があります。 インデックスの次元は時間次元と見なされます。

例えば、32個のサンプルを持つバッチを考えます。 各サンプルは16次元で構成される10個のベクトルを持ちます。 このバッチの入力のshapeは `(32, 10, 16)` となります (`input_shape` はサンプル数の次元を含まないため、 `(10, 16)` となります) 。

このとき、10個のタイムスタンプのレイヤーそれぞれに `Dense` を適用するために、 `TimeDistributed` を利用できます:

```
# as the first layer in a model
model = Sequential()
model.add(TimeDistributed(Dense(8), input_shape=(10, 16)))
# now model.output_shape == (None, 10, 8)

# subsequent layers: no need for input_shape
model.add(TimeDistributed(Dense(32)))
# now model.output_shape == (None, 10, 32)
```

出力のshapeは `(32, 10, 8)` です。

`TimeDistributed` は `Dense` だけでなく任意のレイヤーに使えます。 例えば、 `Conv2D` に対して:

```
model = Sequential()
model.add(TimeDistributed(Conv2D(64, (3, 3)),
                              input_shape=(10, 299, 299, 3)))
```

引数

- `layer`: レイヤーインスタンス。

Bidirectional

[\[source\]](#)

```
keras.layers.wrappers.Bidirectional(layer, merge_mode='concat', weights=None)
```

RNNのBidirectionalなラッパー。

引数

- **layer:** `Recurrent` のインスタンス.
- **merge_mode:** RNNのforwardとbackwardの出力同士を組み合わせる際のモード. {'sum', 'mul', 'concat', 'ave', None}のいずれか. Noneの場合, 出力はリストになります.

Raises

- **ValueError:** `merge_mode` 引数が不正な場合.

例

```
model = Sequential()
model.add(Bidirectional(LSTM(10, return_sequences=True), input_shape=(5, 10)))
model.add(Bidirectional(LSTM(10)))
model.add(Dense(5))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer='rmsprop')
```