

[Docs](#) » [レイヤー](#) » [Advanced Activationsレイヤー](#)

## LeakyReLU

[\[source\]](#)

```
keras.layers.LeakyReLU(alpha=0.3)
```

ユニットがアクティブでないときに微少な勾配を可能とするRectified Linear Unitの特別なバージョン:  $f(x) = \alpha * x \text{ for } x < 0$ ,  $f(x) = x \text{ for } x \geq 0$ .

### 入力のshape

任意. このレイヤーをモデルの最初のレイヤーとして利用する場合, `input_shape` というキーワード引数 (サンプル数の軸を含まない整数のタプル) を指定してください.

### 出力のshape

入力のshapeと同じ.

### 引数

- `alpha`: 0以上の浮動小数点数. 負の部分の傾き.

### 参考文献

- [Rectifier Nonlinearities Improve Neural Network Acoustic Models](#)

## PReLU

[\[source\]](#)

```
keras.layers.PReLU(alpha_initializer='zeros', alpha_regularizer=None, alpha_constraint=None,
```

Parametric Rectified Linear Unit:  $f(x) = \alpha * x \text{ for } x < 0$ ,  $f(x) = x \text{ for } x \geq 0$ , `alphas` はxと同じshapeを持つ学習対象の配列です.

### 入力のshape

任意. このレイヤーをモデルの最初のレイヤーとして利用する場合, `input_shape` というキーワード引数 (サンプル数の軸を含まない整数のタプル) を指定してください.

### 出力のshape

入力のshapeと同じ.

- `alpha_initializer` : 重みを初期化する関数.
- `alpha_regularizer` : 重みを正則化する関数.
- `alpha_constraint` : 重みに対する制約.
- `shared_axes` : 活性化関数で共有する学習パラメータの軸. 例えば, incoming feature mapsが, 出力shapeとして `(batch, height, width, channels)` を持つ, 2Dコンボリューションからなるもので, 空間全体で各フィルターごとに一組しかパラメータを持たない場合、そのパラメータを共有したければ, `shared_axes=[1, 2]` とセットして下さい.

## 参考文献

- [Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification](#)

## ELU

[\[source\]](#)

```
keras.layers.ELU(alpha=1.0)
```

Exponential Linear Unit:  $f(x) = \alpha * (\exp(x) - 1.)$  for  $x < 0$ ,  
 $f(x) = x$  for  $x \geq 0$ .

## 入力のshape

任意. このレイヤーをモデルの最初のレイヤーとして利用する場合, `input_shape` というキーワード引数 (サンプル数の軸を含まない整数のタプル) を指定してください.

## 出力のshape

入力のshapeと同じ.

## 引数

- `alpha` : 負の部分のscale.

## 参考文献

- [Fast and Accurate Deep Network Learning by Exponential Linear Units \(ELUs\)](#)

## ThresholdedReLU

[\[source\]](#)

```
keras.layers.ThresholdedReLU(theta=1.0)
```

Thresholded Rectified Linear Unit:  $f(x) = x$  for  $x > \theta$   $f(x) = 0$  otherwise.

任意. このレイヤーをモデルの最初のレイヤーとして利用する場合, `input_shape` というキーワード引数 (サンプル数の軸を含まない整数のタプル) を指定してください.

### 出力のshape

入力のshapeと同じ.

### 引数

- **theta** : 0以上の浮動小数点数. 活性化する閾値.

### 参考文献

- [Zero-Bias Autoencoders and the Benefits of Co-Adapting Features](#)

---

## Softmax

[\[source\]](#)

```
keras.layers.Softmax(axis=-1)
```

Softmax関数.

### 入力のshape

任意. このレイヤーをモデルの最初のレイヤーとして利用する場合, `input_shape` というキーワード引数 (サンプル数の軸を含まない整数のタプル) を指定してください.

### 出力のshape

入力のshapeと同じ.

### 引数

- **axis** : 整数, softmax正規化が適用される軸.

---

## ReLU

[\[source\]](#)

```
keras.layers.ReLU(max_value=None)
```

Rectified Linear Unit activation function.

### 入力のshape

任意. このレイヤーをモデルの最初のレイヤーとして利用する場合, `input_shape` というキーワード引数 (サンプル数の軸を含まない整数のタプル) を指定してください.

## 出力のshape

入力のshapeと同じ.

## 引数

- `max_value` : 浮動小数点数, 最大の出力値.