

活性化関数の使い方

活性化関数は `Activation` レイヤー, または全てのフォワードレイヤーで使える引数 `activation` で利用できます.

```
from keras.layers.core import Activation, Dense

model.add(Dense(64))
model.add(Activation('tanh'))
```

上のコードは以下と等価です :

```
model.add(Dense(64, activation='tanh'))
```

要素ごとに適用できるTensorFlow/Theano/CNTK関数を活性化関数に渡すこともできます:

```
from keras import backend as K

def tanh(x):
    return K.tanh(x)

model.add(Dense(64, activation=tanh))
model.add(Activation(tanh))
```

利用可能な活性化関数

softmax

```
softmax(x, axis=-1)
```

Softmax関数

引数

- `x`: テンソル.
- `axis`: 整数. どの軸にsoftmaxの正規化をするか.

戻り値

テンソル. softmax変換の出力.

Raises

- `ValueError`: `dim(x) == 1` のとき.

elu

```
elu(x, alpha=1.0)
```

selu

```
selu(x)
```

Scaled Exponential Linear Unit. (Klambauer et al., 2017)

引数

- **x**: 活性化関数を適用するテンソルか変数.

参考文献

- [Self-Normalizing Neural Networks](#)

softplus

```
softplus(x)
```

softsign

```
softsign(x)
```

relu

```
relu(x, alpha=0.0, max_value=None)
```

tanh

```
tanh(x)
```

sigmoid

```
sigmoid(x)
```

hard_sigmoid

```
hard_sigmoid(x)
```

linear

```
linear
```

より高度な活性化関数

単純なTensorFlow/Theano/CNTK関数よりも高度な活性化関数 (例: 状態を持てるlearnable activations) は, **Advanced Activation layers**として利用可能です. これらは, **keras.layers.advanced_activations** モジュールにあり, **PReLU** や **LeakyReLU** が含まれます.