

演習： 1-2.活性化関数

```
In [6]: import numpy as np
from common import functions
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('ggplot')
```

```
In [7]: def print_vec(text, vec):
        print(f'{text} : \n{vec}')
```

中間層の活性化関数

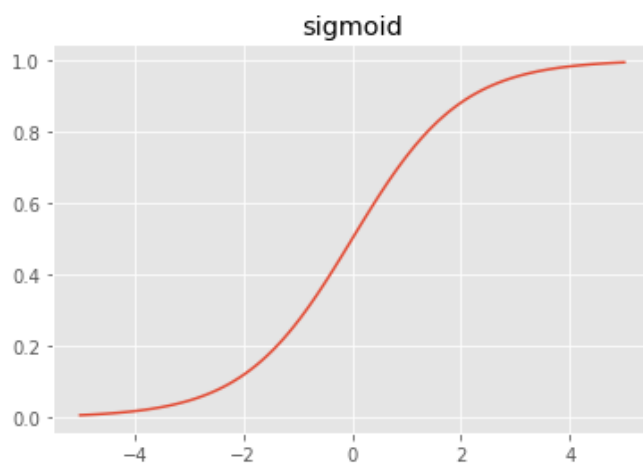
```
In [8]: def sigmoid(x):
        """シグモイド関数（ロジスティック関数）"""
        return 1/(1 + np.exp(-x))

        def relu(x):
            """ReLU関数"""
            return np.maximum(0, x)

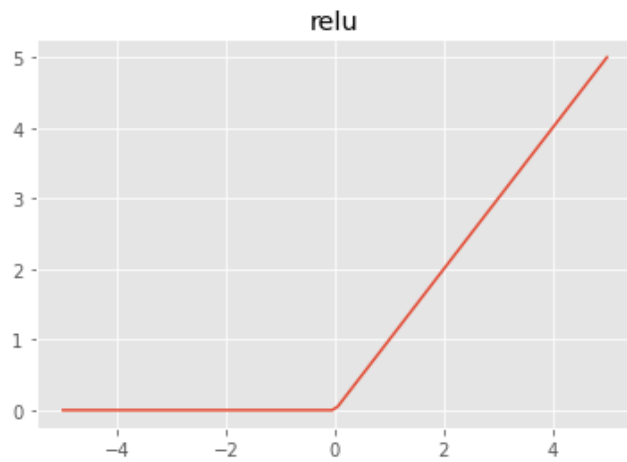
        def step_function(x):
            """ステップ関数（閾値0）"""
            return np.where( x > 0, 1, 0)
```

活性化関数の形状を描画する

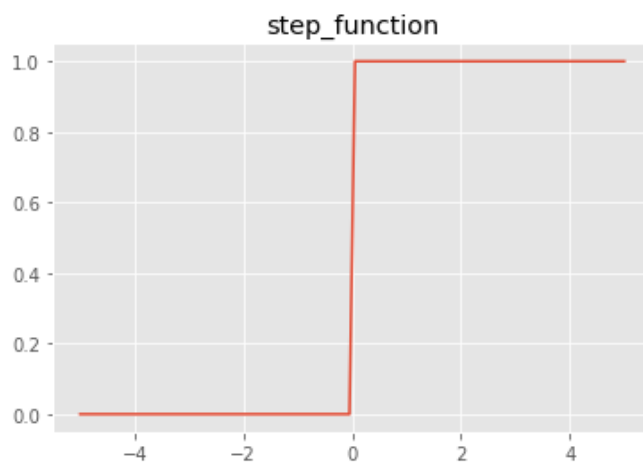
```
In [16]: x = np.linspace(-5, 5, 100)
y = sigmoid(x)
plt.plot(x, y)
plt.title('sigmoid')
plt.show()
```



```
In [17]: x = np.linspace(-5, 5, 100)
y = relu(x)
plt.plot(x, y)
plt.title('relu')
plt.show()
```



```
In [18]: x = np.linspace(-5, 5, 100)
y = step_function(x)
plt.plot(x, y)
plt.title('step_function')
plt.show()
```



順伝播（単層・複数ユニット）

```
In [9]: # 重み
W = np.array([[0.1], [0.2]])
print_vec("重み", W)

# バイアス
b = 0.5
print_vec("バイアス", b)

# 入力値
x = np.array([2, 3])
print_vec("入力", x)

# 総入力
u = np.dot(x, W) + b
print_vec("総入力", u)
```

```
重み :
[[0.1]
 [0.2]]
バイアス :
0.5
入力 :
[2 3]
総入力 :
[1.3]
```

```
In [22]: # 中間層出力(ReLU)
z = relu(u)
print_vec("中間層出力(ReLU)", z)
```

```
中間層出力(ReLU) :
[1.3]
```

```
In [23]: # 中間層出力(sigmoid)
z = sigmoid(u)
print_vec("中間層出力(sigmoid)", z)
```

```
中間層出力(sigmoid) :
[0.78583498]
```

```
In [24]: # 中間層出力(step_function)
z = step_function(u)
print_vec("中間層出力(step_function)", z)
```

```
中間層出力(step_function) :
[1]
```

順伝播（単層・複数ユニット）

```
In [63]: # 重み
W = np.array([
    [0.1, 0.2, 0.3],
    [0.2, 0.3, 0.4],
    [0.3, 0.4, 0.5],
    [0.4, 0.5, 0.6]
])
print_vec("重み", W)

# バイアス
b = np.array([0.1, -0.5, 0.8])
print_vec("バイアス", b)

# 入力値
x = np.array([-2.0, -1.0, 4.0, -1.0])
print_vec("入力", x)

# 総入力
u = np.dot(x, W) + b
print_vec("総入力", u)
```

```
重み :
[[0.1 0.2 0.3]
 [0.2 0.3 0.4]
 [0.3 0.4 0.5]
 [0.4 0.5 0.6]]
バイアス :
[ 0.1 -0.5  0.8]
入力 :
[-2. -1.  4. -1.]
総入力 :
[ 0.5 -0.1  1.2]
```

```
In [64]: # 中間層出力(ReLU)
z = relu(u)
print_vec("中間層出力(ReLU)", z)
```

```
中間層出力(ReLU) :
[0.5 0.  1.2]
```

```
In [65]: # 中間層出力(sigmoid)
z = sigmoid(u)
print_vec("中間層出力(sigmoid)", z)
```

```
中間層出力(sigmoid) :
[0.62245933 0.47502081 0.76852478]
```

```
In [66]: # 中間層出力(step_function)
z = step_function(u)
print_vec("中間層出力(step_function)", z)
```

```
中間層出力(step_function) :
[1 0 1]
```

考察：

- ReLUは、入力が0以上なら入力値をそのまま出力。入力が0未満なら0を出力。
- sigmoidは、入力が0以上なら0.5以上の値を出力。最大は1。入力が0未満なら0.5未満の値を出力。最小は0。
- step_functionは、入力が0より大きいと1、0以下だと0を出力する。

In []: