

【要点まとめ】 活性化関数には10以上の種類がある。ステップ関数、シグモイド関数、Relu関数が特に有名。

- ステップ関数 入力値が0以下であれば出力値は0、入力値が0より大きければ出力値は1となる関数 線形分離可能なものしか学習できず、あまり用いられない。
- シグモイド関数 入力値に反応して0～1の値を出力する 勾配消失問題が引き起こされることがわかったので、あまり用いられない
- Relu関数 入力値が0以下であれば0を、それ以上であればその値を出力する 最もナウい関数

【実装演習】

- 活性化関数3種類の実装

```
# ステップ関数
def step_func(x):
    y = x > 0
    return y.astype(np.int)

# シグモイド関数
def sigmoid_func(x):
    return 1 / (1 + np.exp(-x))

# Relu関数
def relu_func(x):
    return np.maximum(0, x)
```

※ 参考資料：ゼロから作るDeepLearning

【実装演習考察】 3種類の有名な活性化関数を実装してみました。説明を聞いたり、入出力のグラフを見ると複雑に感じますが、実際に実装するととてもシンプルに書けることがわかりました。numpyなどのライブラリを使用する前提ですが、デファクトスタンダードなこれらのライブラリをもっと使用して ある程度のアルゴリズムは直感的に実装していけるよう訓練が必要だと感じました。