

【要点まとめ】 中間層と出力層では利用される活性化関数が異なる。

- 恒等関数（恒等写像） 中間層で算出された値がそのまま出力される。（要はなにもしない関数） 回帰問題で利用されることがある。
- ソフトマックス関数 出力は0.0～1.0の間をとる。全ての入力を母数として扱うため、この関数の出力の総和は1.0となる。つまりそれはそのまま確率として使用することができる。

誤差関数は以下がよく用いられる。

- 二乗和誤差 誤差を二乗することで正負の問題を解決している。
- 交差エントロピー誤差 自然対数を利用している。

【実装演習】

- 今回はソフトマックス関数と二乗和誤差を実装してみる

```
# ソフトマックス関数
def softmax_func(x):
    c = np.max(x)
    exp_x = np.exp(x - c) # オーバーフロー対策
    sum_exp_x = np.sum(exp_x)
    return exp_x / sum_exp_x

# 二乗和誤差関数
def mean_squared_error_func(d, y):
    return 0.5 * np.sum((d-y)**2)
```

※ 参考資料：ゼロから作るDeepLearning

【実装演習考察】 ソフトマックス関数などの指数を扱う処理を実装する際は、特にオーバーフローなど処理の限界に気を遣う必要があることを知りました。 アルゴリズムを理解してそのまま実装するだけでなく、起こりうる問題も予想して、バグを埋め込まないよう注意したいと思います。