

# 損失関数

## ゆってぃー

# 二乗和誤差

- $y_1, y_2$ それぞれで二乗和誤差を出力し、結果を考察しよう

```
import numpy as np

def sum_squared_error(y, t):
    return 0.5 * np.sum((y - t) ** 2)

# 教師データ
t = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
# ニューラルネットワークの出力
y1 = [0.1, 0.05, 0.6, 0.0, 0.05, 0.1, 0.0, 0.1, 0.0, 0.0]
y2 = [0.1, 0.05, 0.1, 0.0, 0.05, 0.1, 0.0, 0.6, 0.0, 0.0]
```

# 交差エントロピー誤差

$$E = - \sum_k t_k \log y_k$$

# 交差エントロピー誤差

- $y_1, y_2$ で交差エントロピー誤差を出力し、結果を考察しよう

```
import numpy as np

def cross_entropy_error(y, t):
    delta = 1e-7
    return -np.sum(t * np.log(y + delta))

# 教師データ
t = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
# ニューラルネットワークの出力
y1 = [0.1, 0.05, 0.6, 0.0, 0.05, 0.1, 0.0, 0.1, 0.0, 0.0]
y2 = [0.1, 0.05, 0.1, 0.0, 0.05, 0.1, 0.0, 0.6, 0.0, 0.0]
```

# [バッチ対応版] 交差エントロピー誤差

- バッチ処理に対応した交差エントロピー誤差を実装しよう
  - 以下の数式を参考にしてください
  - データが1つの場合と、データがバッチとしてまとめられて入力される場合の両方に対応するように実装してください

$$E = -\frac{1}{N} \sum_n \sum_k t_{n,k} \log y_{n,k}$$