

# Python と Keras によるディープラーニング

3章 入門：ニューラルネットワーク

3.1 ニューラルネットワークの構造

17T4028N 菊田尚樹

# ニューラルネットワークの構造

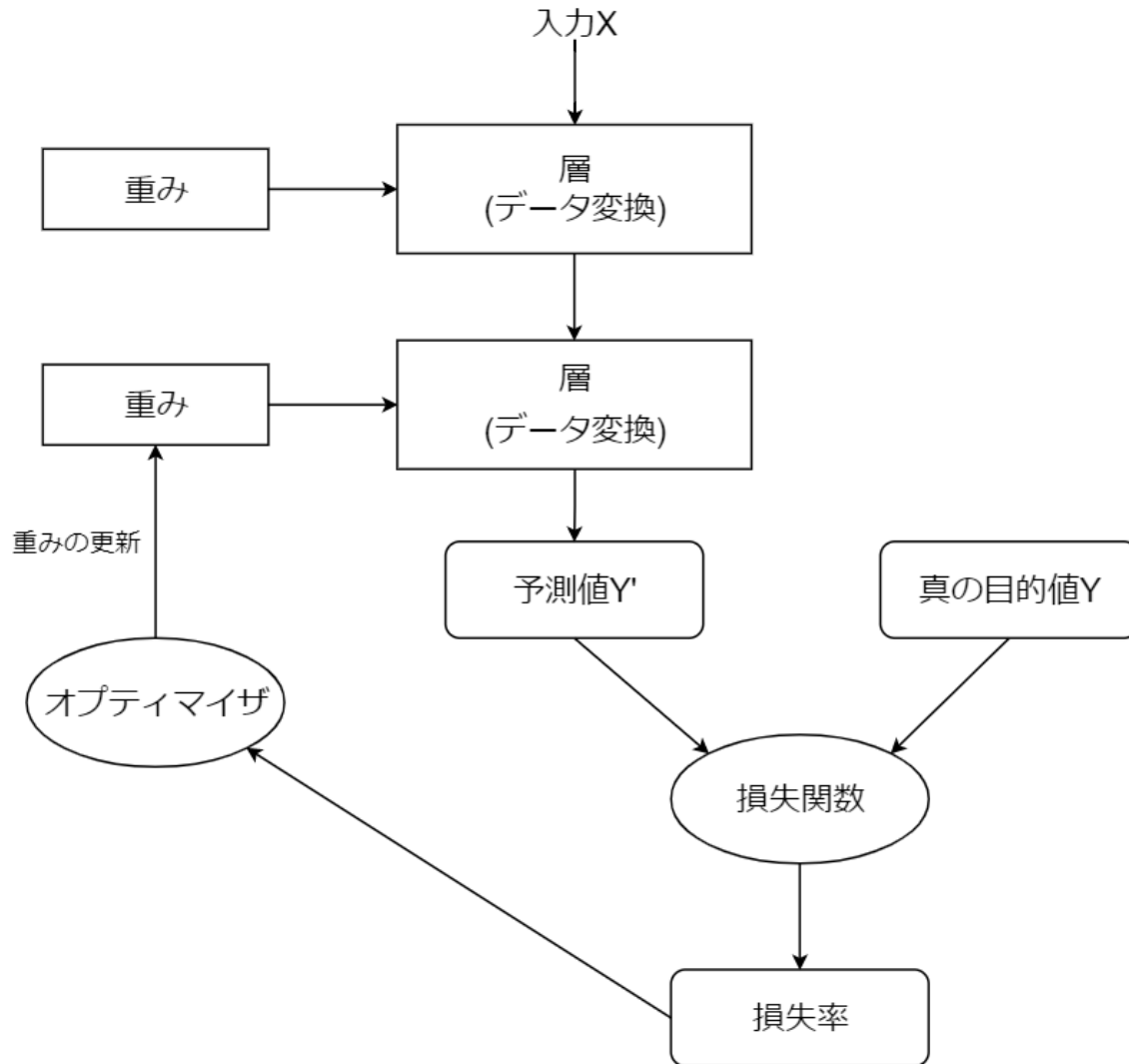
## 以前の章の復習

ニューラルネットワークで出てくるオブジェクト

- 層
- 入力データとそれに対応する目的値
- 予測値と目的値から損失率を計測する損失関数
- 損失率を用いて重みを調整するオプティマイザ

# ニューラルネットワークの構造

以前の章の復習 オブジェクトの関係図



# 層について

- 層はデータを処理するモジュール
- 1つ以上のテンソルを入力とし受け取り、1つ以上のテンソルを出力する
- 層の状態は重みによってあらわされる

# テンソルと層

テンソルのフォーマット・データ処理の種類は様々。  
それらに適した層もそれぞれ異なる。

テンソル	テンソルの具体例	適した層
2次元 (samples, features)	個人情報のデータセット 人数：1000 要素：名前, 性別, 年齢, 住所 テンソル：(1000, 4)	全結合層(Dense)
3次元 (samples, timesteps, features)	株価のデータセット (一分ごとに現在の株価と最高値と最低値を格納する) データ数：360件(360日分) 取引時間：390分 テンソル：(360, 3, 390)	リカレント層
4次元 (samples, height, width, channels)	画像のデータセット データ数：10000件 データサイズ：300×250 チャンネル数：3 (RGB) テンソル：(10000, 300, 250, 3)	2次元の畳み込み層 (Conv2D)

# 層の動的構築

層の追加を行う場合

```
from keras import models
from keras import layers

model = models.Sequential()
model.add(layers.Dense(32, input_shape=(784, )))
model.add(layers.Dense(32))
```

モデルに追加される層は入力側の層の形状に合わせて動的に構築される。

(二つ目の層にてinput\_shapeパラメータへの引数をしなくてもよい)

# 損失関数の選択

適切な損失関数を選択することが重要。

損失関数とタスクの成功との関連性があまいだと望んだ動きにはならない。

ただ、一般的な問題には代表的な損失関数がある。

(例)

二値分類→二値の交差エントロピー

多クラス分類→多クラス交差エントロピー

回帰→平均二乗誤差

系列学習→CTC

基本的に既存の損失関数を使えばよい。

損失関数を独自に開発するのはまったく新しい研究課題に取り組んでいる場合。

# 交差エントロピー

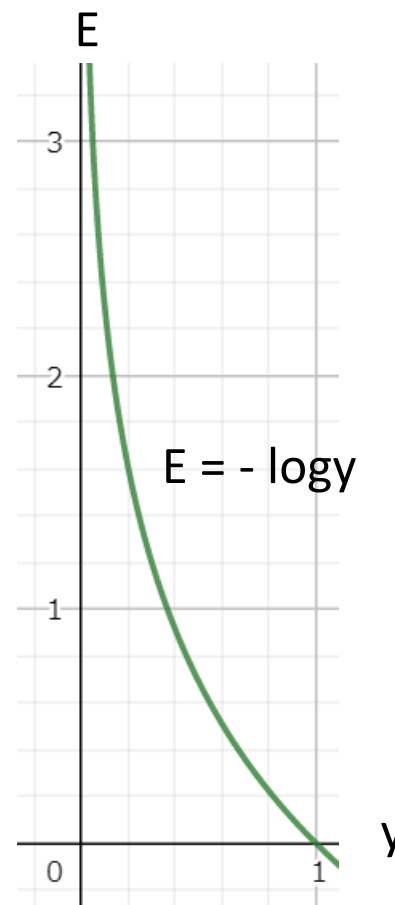
分類問題に用いられる

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

$t_k$ は正解ラベルのみ1 その他は0

→正解のときの項だけ見ればよい

$y_k$ は出力(0~1)



出力が正解に近づくほど交差エントロピーは小さく  
遠ざかるほど大きくなる