PythonとKerasによるディープラーニング

第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

19nm715n ZHANGYIHANG

3章について

- NNの中核的要素
- Kerasの紹介
- マシンのセットアップ
- NNを使った基本的な分類と回帰の問題の解決

ニューラルネットワークの訓練は次のオブジェクトに基づいて行われます。

- ネットワーク（モデル）として結合される層
- 入力データと対応する目的値
- 学習に使用されるフィードバックを定義する損失関数
- 学習の進め方を決定するオプティマイザ

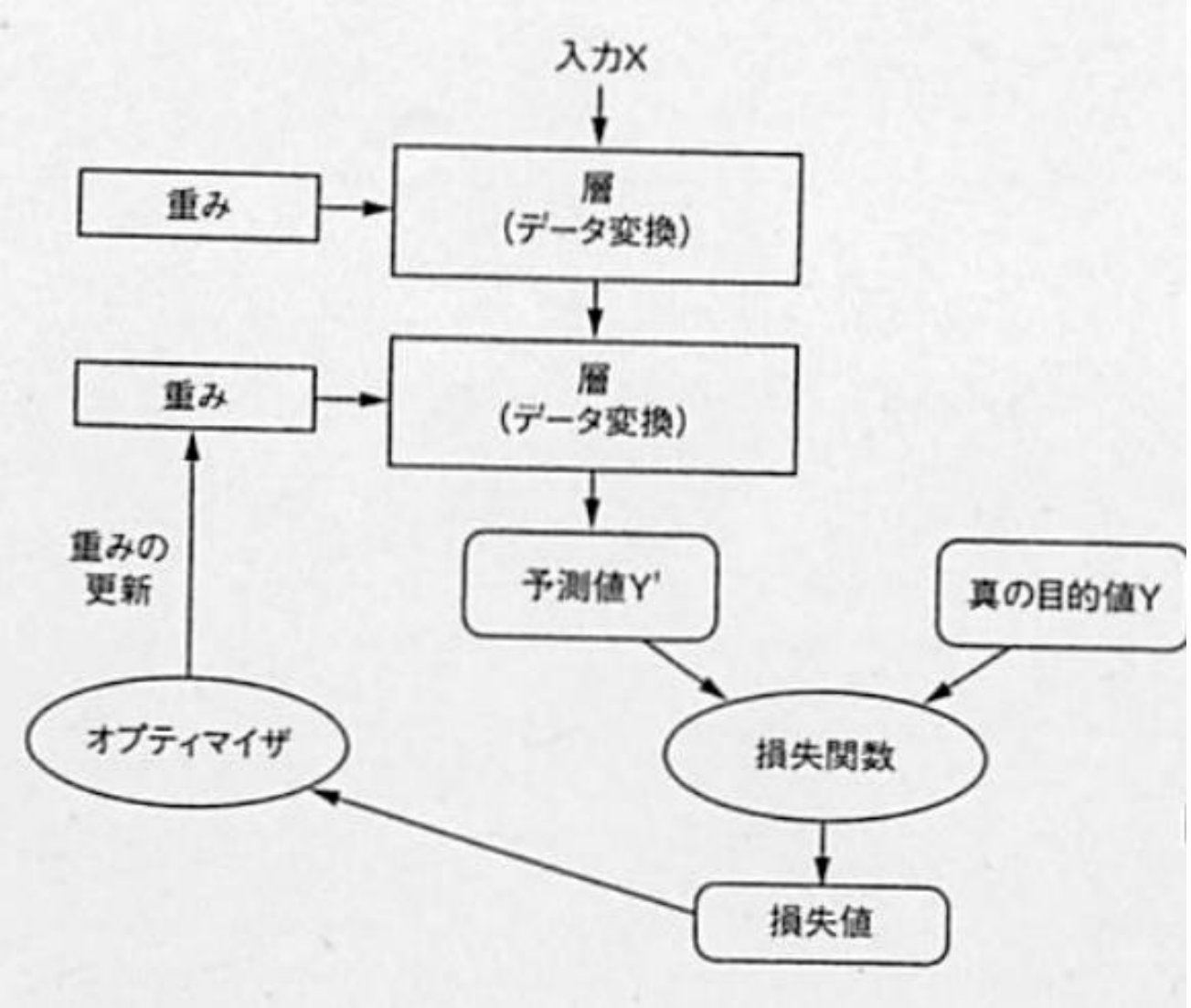


図3.1：ネットワーク、層、損失関数、オプティマイザの関係

3.1.1　層：ディープラーニングの構成要素

ニューラルネットワークの基本的なデータ構造は層です。層の状態は層の重みによって表されます。層の重みは、確率的勾配降下法に基づいて学習された１つ以上のテンソルであり、それぞれネットワークの知識の一部を含んでいます。

テンソルのフォーマットやデータ処理の種類はさまざまであり、それらに適している層もそれぞれ異なります。

データ	**処理方法**
（２）ベクトルデータ	密結合された層によって処理
（３）シーケンスデータ	リカレント層によって処理
（４）画像データ	２次元の畳み込み層によって処理

ディープラーニングの層はLEGOブロックとして考えることができます。Kerasでのモデルの構築は、互換性のある層をつなぎ合わせて有益なデータ変換パイプラインを形成します。

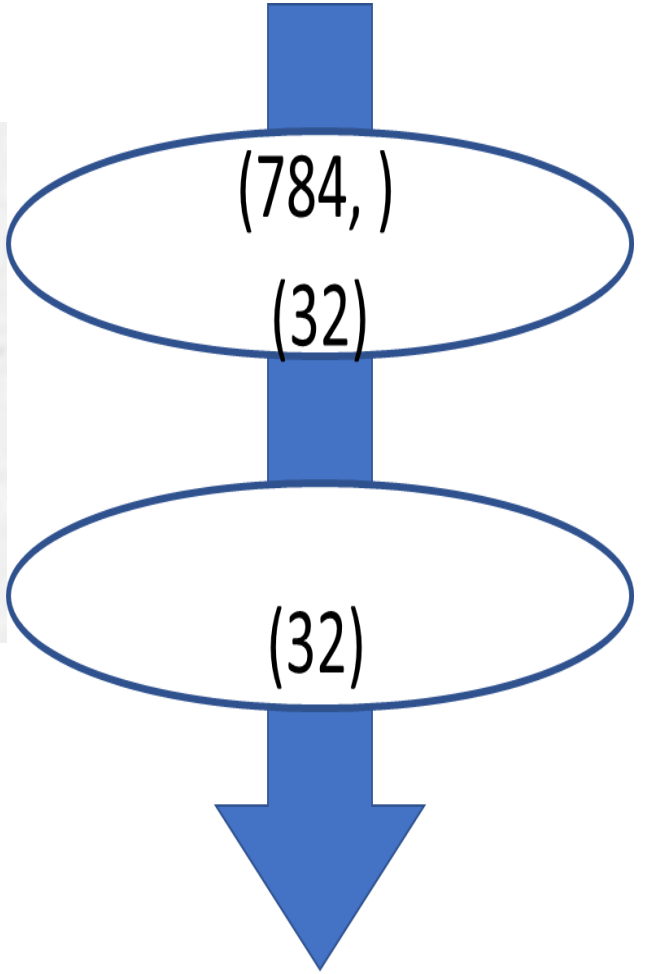
```
from keras import models
from keras import layers

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

モデルを作ります

層を作ります

自動的に推察します。



3.1.2　モデル：複数の層からなるネットワーク

　ディープラーニングモデルは複数の層からなる有向非巡回グラフです。これを形成する前に、色々な種類のネットワークトポロジが待ち構えています。以下はいくつのトポロジの例：

- ２分岐ネットワーク
- マルチヘッドネットワーク
- インセプションブロック

　これらのトポロジは仮説空間を定義します。ネットワークトポロジの選択により、「仮説空間」は入力データを出力データにマッピングする一連のテンソル演算に絞り込まれます。

　正しいネットワークアーキテクチャの選び出し方法は、科学というよりも、選択術の方が適当です。本物のニューラルネットワークアーキテクトを目指すなら、実践しなければなりません。

3.1.3　損失関数とオプティマイザ：学習プロセスを設定するための鍵

ネットワークアーキテクチャが定義されたら、

次の2つの選択を行わなければなりません。

- 損失関数
  訓練中に最小化する数量。タスクの成功の目安となる尺度であり、
  目的関数とも呼ばれます。

- オプティマイザ
  損失関数に基づいてネットワークをどのように
  更新するのかを決定します。
  確率的勾配降下法の一種を実装します。

  問題を正しく評価して適切な目的関数を選択することは非常に重要です。目的関数の微小な問題が副作用を生むこともあります。

損失関数選択例

二値分類	**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**
回帰問題	**平均二乗誤差（MSE）**
系列学習	**CTC(connecitionist Temporal Classification)**