

PythonとKerasによる ディープラーニング

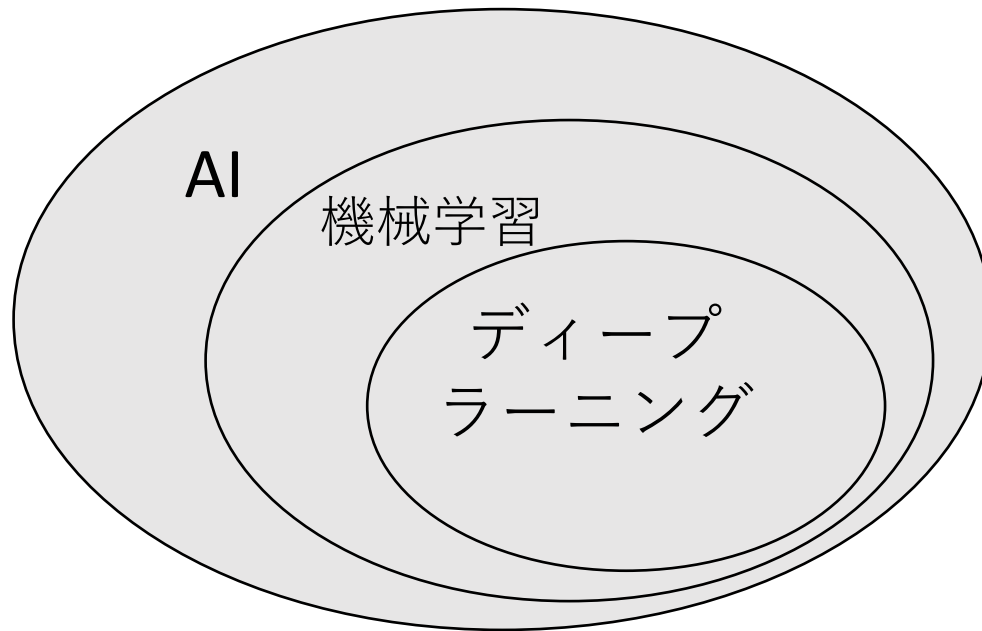
第1章 ディープラーニングとは何か

1.1 AI, 機械学習, ディープラーニング

17T4014Y 伊藤 陽樹

AI、機械学習、ディープラーニングとは

- 本来ならば人が行う知的な作業を自動化する取り組み



1950~1980年代のAI

- Symbolic AI

- 知識を操作するのに十分な大量のルールを明示的に定義して、手作業で組み込むアプローチ.
- チェスの試合など、明確に定義された論理的な問題を解くのに適していた.
- しかし、より複雑でファジーな問題(画像分類、音声認識、言語の翻訳など)を解くのは不可能.

機械学習

- Symbolic AIに代わる新しいアプローチ.
- データと、そのデータから期待される答えを入力すると、ルールを出力.
- 明示的にプログラムされるのではなく、訓練 (train)される.
- タスクに関連するサンプルを大量に与えると、それらのサンプルから統計的な構造を抽出し、最終的にそのタスクを自動化するためのルールの生成が可能になる.

機械学習のアルゴリズムが「何をするのか」

- 機械学習: 期待されるもののサンプルを与えると、データ処理タスクを実行するためのルールが抽出される.
- 機械学習を実行するためには、次の3つのものが必要.
 - ✓ 入力データ点
 - 音声認識タスク: 人の会話を録音した音声
 - 画像タグ付けタスク: 写真
 - ✓ 期待される出力の例
 - 音声認識タスク: 音声ファイルから人が起こした原稿
 - 画像タグ付けタスク: “dog”や“cat”といったタグ
 - ✓ アルゴリズムがよい仕事をしたかどうかを評価する方法
 - 結果はアルゴリズムの動作方法を調整するためのフィードバックとして使用. 「学習」と呼んでいるのは、この調整ステップのこと.

機械学習のアルゴリズムが「何をするのか」

- 機械学習のモデルは、入力データを意味のある出力に変換する.
- これは、既知の入力と出力のサンプルから「学習」するプロセス.

したがって、

- 「データを意味のある方法で変換すること」が機械学習とディープラーニングでは主な課題になる.
- つまり、機械学習は、期待される出力に近づくために、与えられた入力データから有益な**表現**を学習する.

データから表現を学習する

- 表現とは、データを表す(エンコードする)ためにデータを別の角度から捉える方法.
- 例: カラー画像 ← RGBフォーマットかHSVフォーマット
- ある表現では難しいタスクが、別の表現では簡単になることがある.
- 機械学習モデルの本質は、入力データに適した表現(分類タスクなど、データを現在のタスクにより適したものにする変換)を見つけ出すことにある.

機械学習における「学習」

- よりよい表現を自動的に検索するプロセス.
 - 機械学習のアルゴリズムは、予め定義された一連の演算(仮説空間)を一通り検索するだけ.(創造力を働かせることはない)
 - 技術的な定義:
機械学習とは「フィードバックのガイダンスに基づいて、予め定義された仮説空間内で入力データの有益な表現を検索する」こと.
- 音声認識から自動運転まで、広い範囲にわたる知的なタスクの解決が可能になる.

ディープラーニング

- 新しいスタイルの機械学習. 連続する層の学習に重点が置かれる.
 - 表現の層がたくさん重なっている階層型の表現.
 - ディープラーニングの「ディープ」は、「連続する表現の層」という概念を指す.
 - こうした階層型の表現を(ほぼ)ニューラルネットワークと呼ばれるモデルで学習する.

まとめると、

- 機械学習とは、画像などの入力からラベル“cat”などの目的値へのマッピングのようなもの。
 - ▶ マッピングは、入力値と目的値の様々なサンプルを観測することで実現。
- ディープニューラルネットワークでは、入力値から目的値へのマッピングが単純なデータ変換(層)を深く積み重ねていくことによって実現され、そうしたデータ変換がサンプルから学習される。

学習の仕組み

- **重み(パラメータ):** 層が入力データに対して何を行うかに関する指定。
 - 技術的には、層によって実装される変換は重みによってパラメータ化される。

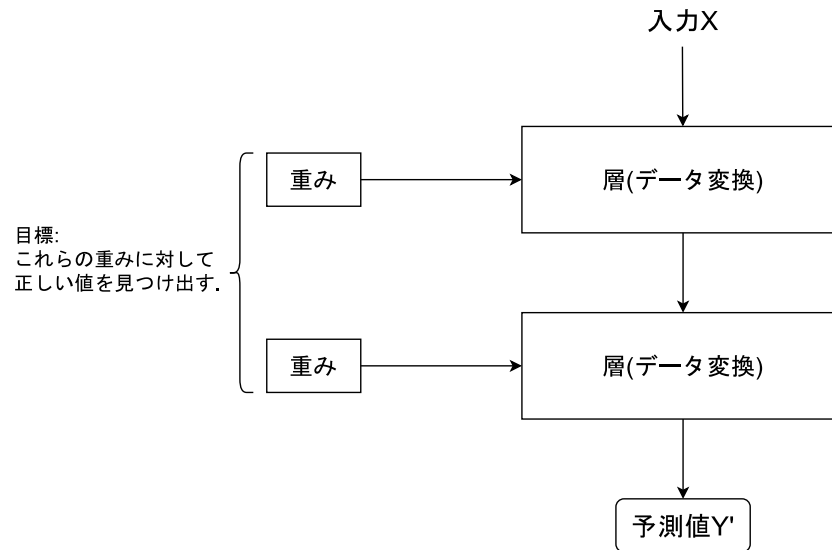


図: ニューラルネットワークはその重みによってパラメータ化される

学習の仕組み

- この場合の学習とは、入力値を関連する目的値に正しくマッピングするための重みの値を、ネットワーク内の全ての層にわたって見つけ出すこと。
- ニューラルネットワークの出力を制御するには、その出力が期待されているものからどのくらいかけ離れているか計測できなければいけない。
 - これを計測するのがネットワークの**損失関数(目的関数)**。
 - 損失関数は、ネットワークの予測値と真の目的値から損失率を計算することで、そのサンプルでのネットワークの性能を補足。

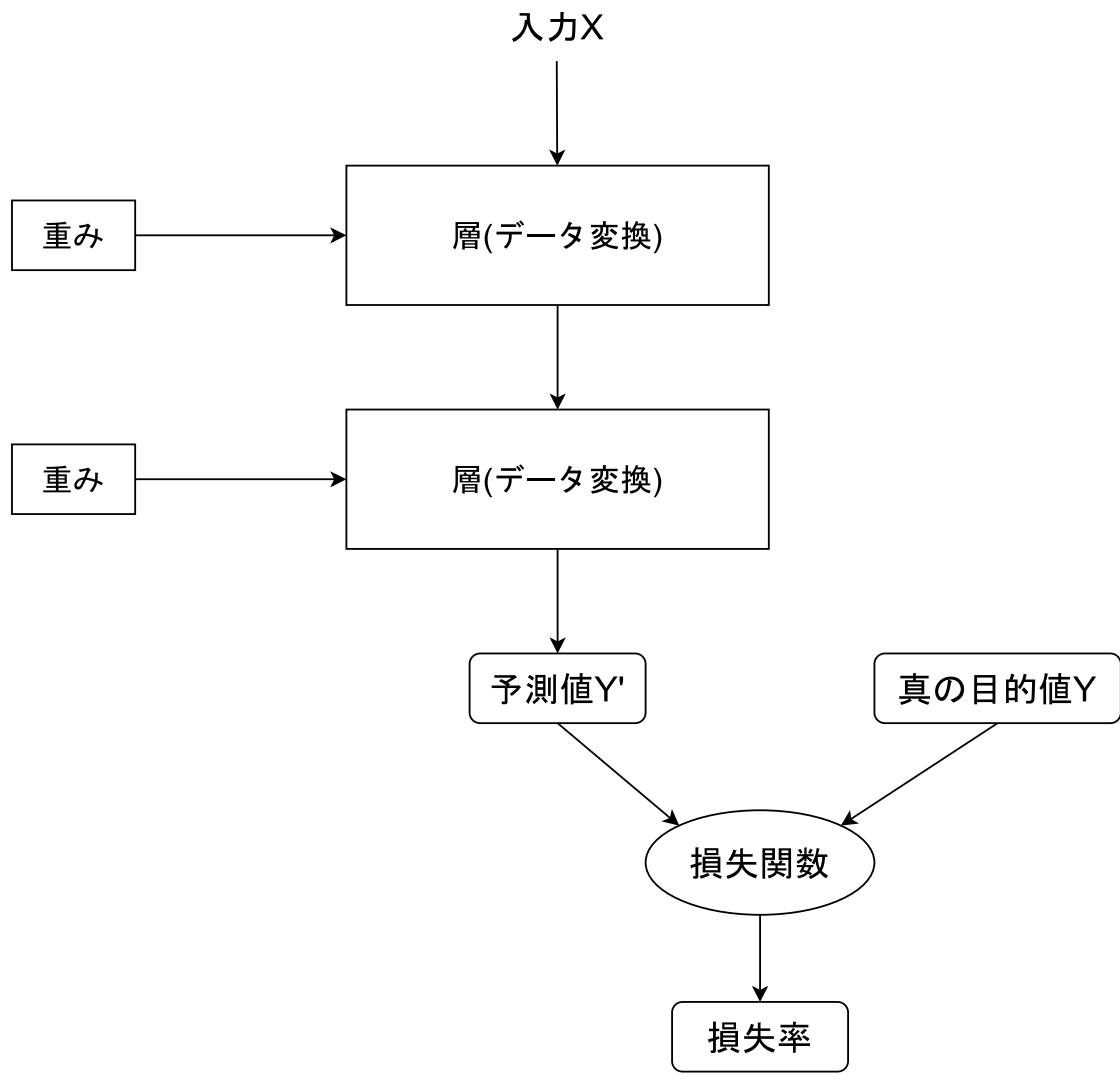


図: ネットワークの出力の品質を評価する損失関数

学習の仕組み

- ディープラーニングの基本的な原理は、損失率をフィードバックとして使用することで重みの値を少しずつ調整していく。
 - 重みを調整するのは**オプティマイザ**.
 - オプティマイザは**バックプロパゲーション(誤差逆伝播法)**と呼ばれるアルゴリズムを実装する.
- ネットワークの重みは乱数値で初期化され、ネットワークの出力は理想的な出力からかけ離れている.
- しかし、ネットワークがサンプルを処理するたびに、重みが正しい方向に向かって少しずつ調整される(**訓練ループ**).
- 訓練ループを十分な回数繰り返すと、損失関数を最小化する重みの値が生成される.

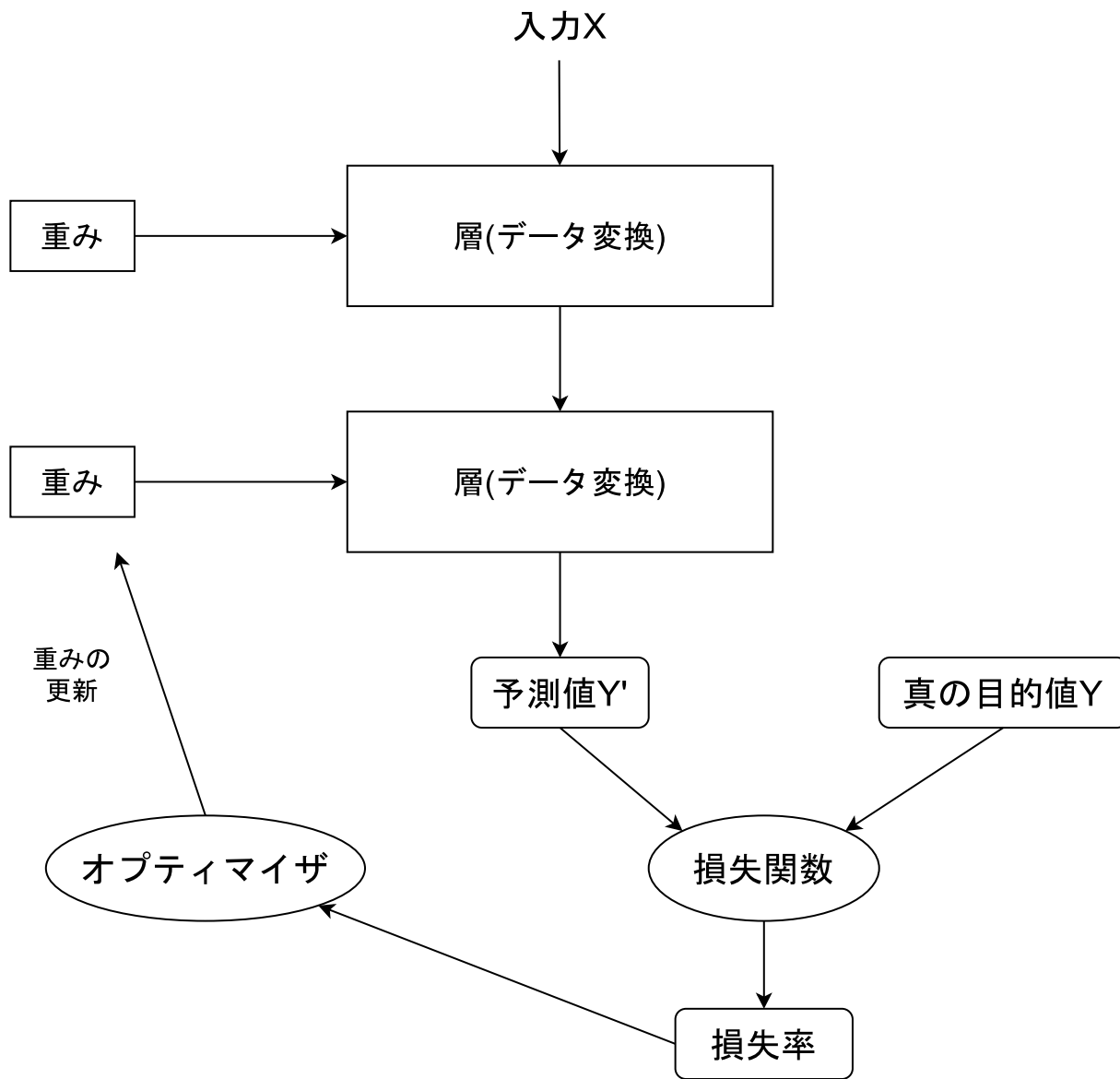


図: 損失率は重みを調整するためのフィードバックとして使用される