

Deep Learning の基礎知識

KaiRA

この冊子は主に KaiRA の 2018 年度の会誌と 2019 年度の会誌がもとになっています。

2018 年度の会誌は無料で、2019 年度の会誌は 200 円でそれぞれ [KaiRA ホームページ](#)
[の Docs ページ](#)からダウンロードできます。

それらの会誌はこの冊子より長くなりますが、初歩から丁寧に詳しく Deep Learning
について解説しているので、興味のある方はぜひご覧ください!

目次

1. 人工知能・機械学習・深層学習	・・・1
2. ニューラルネットワークと層構造	・・・1
3. 学習	・・・2
4. 損失関数	・・・2
5. 勾配法	・・・3
6. 誤差逆伝播法	・・・3

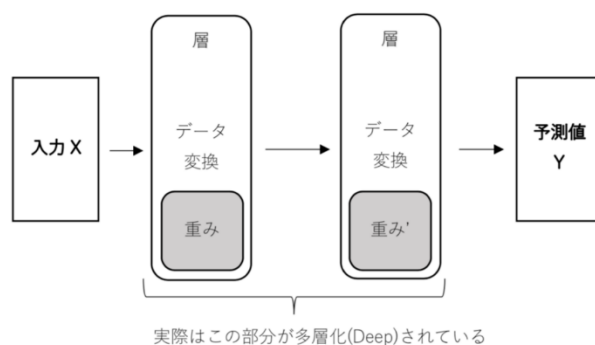
1. 人工知能・機械学習・深層学習

近年、とりわけ alpha Go が現れてきてから人工知能(AI)や機械学習(Machine Learning)、深層学習(Deep Learning)といった単語を耳にします。しかし多くの人はそれぞれの違いを把握していないと思います。なので、初めにこれらについて簡単に説明します。

AI は「人間のように知的な作業を行う計算機」というイメージが一般的です。また、計算機だけでなく、そのような学術的分野としても使われます。Machine Learning はその分野の中の一つの分野です。Machine Learning の大きな特徴としてはシステムに人力で加工した大量のデータを与えることで、訓練し、その問題の解決方法を学習させるという点です。Machine Learning ではこの加工、特徴量エンジニアリングが肝となってきます。AI は単純に、例えば条件分岐だけで、人力で人間っぽくさせれば出来てしまい、Machine Learning はそれをある程度自動的にやってくれるのです。そして、Deep Learning は機械学習の中の一つの手法です。Machine Learning では特徴量エンジニアリングを人力で行う必要がありましたが、Deep Learning はそれも自動化してくれるのです。KaiRA では主に Deep Learning についての勉強会を行っています。

2. ニューラルネットワークと層構造

さて、前節で説明した Deep Learning のアルゴリズムの要がニューラルネットワークモデルです。ニューラルネットワークの模式図を図 1 に示します。



重み
の正しい値を見つけ出すことが目標

図 1 ニューラルネットワークの模式図(KaiRA NF2018 会誌より)

ニューラルネットワークの処理の 1 層あたりの流れは、入力データに重みをかけたり、バイアスを足したりすることで、線形変換を行い、その後活性化関数による非線形変換を行いうものとなります。この処理を何回も繰り返す、つまり多層化することでより複雑なタスクを可能にしているのです。

3. 学習

Deep learning における学習とは、ひとことで言えば「ニューラルネットワークにおける重みパラメータを調整すること」です。Deep learning では、この「学習」を自動で行うように設定することで、人間が詳細な設計をせずに目的とする複雑な処理を行うことができるようになっていきます。この調整は、理想的なモデルの特徴と現在のモデルの特徴との「ずれ」が小さくなるように重みパラメータの値を変更することで行われます。この「ずれ」とは、具体的には次に説明する損失関数の出力のことです。自動で重みパラメータを調整すると言っても、この「ずれ」を表す損失関数の中身や重みパラメータの値の変更の仕方など人間が設定する部分は複数あり、これらをうまく設定できるかどうかは Deep learning のモデルの精度に直結します。

4. 損失関数

損失関数とは、ひとことで言えば「理想的なモデルの特徴と現在のモデルの特徴とのずれを出力する関数」のことです。「学習」で述べたように、Deep learning のモデルはこの「ずれ」が小さくなるように重みパラメータを調整します。Deep learning のモデルはあくまで一つの大きな演算装置であるので、学習させるためには「ずれ」を数値化する必要があります。そのときに使われるのが「損失関数」です。

例として次の2式のようなものがあります。

$$E = \frac{1}{2} \sum_k (y_k - t_k)^2 \quad (1)$$

$$E = - \sum_k t_k \log y_k \quad (2)$$

E はずれ(Error)、 t_k は理想的なモデルの出力(正解)、 y_k は現在のモデルの出力(予想)です。

(1)式は2乗和誤差、(2)式は交差エントロピー誤差と言います。2乗和誤差は気温予想など、連続的な値から数値を予想する場合(回帰問題)によく使われ、交差エントロピー誤差は画像分類など、入力データからその属性を予想する場合(分類問題)によく使われます。

損失関数の出力をもとに次に説明する勾配法や誤差逆伝播法を用いることで、損失関数の出力が小さくなるようにモデルを学習させることができます。

5. 勾配法

損失関数の値が小さくなるようにパラメータの値を変更していくことを「学習」と呼びます。問題はそれをどのように行うかです。例えば $y=x^2$ のようにグラフを描いて一目で最小値がわかるようなら楽なのですが、実際はそうはいきません。変数の数も多いためグラフを描くことも難しいからです。そこで勾配情報を使って損失関数の値を小さくしようとするのが勾配法です。損失関数 L をあるパラメータ θ で微分した値を使って計算していきます。

これはよく山登りに例えられます。あなたは今山にいて現在地がわからなくなったとします。しかし今いる場所の坂の傾き(勾配)は感覚的にわかるかと思います。従ってより高い方向に登っていくようにすればいつかは頂上にたどり着くでしょう。勾配法は山登りとは違って損失関数の小さい場所を目指すものですが、現在の勾配情報を基に場所を変更していくというイメージは共通しています。

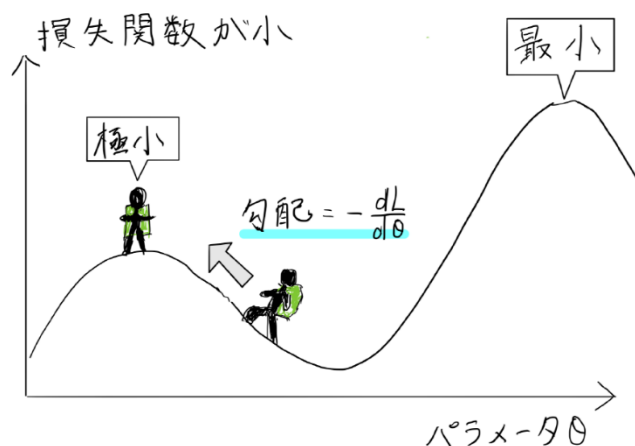


図2 山登りで例えられる勾配法(KaiRA NF2019 会誌より)

2. 誤差逆伝播法

上の勾配法をプログラムで実装する上で、誤差逆伝播法という手法が使われます。端的に言えば、誤差逆伝播法とは複雑な関数を簡単な演算に分割して微分をし、その値を伝播していく手法です。そもそもプログラム(数値微分)で複雑な微分を正確に計算するのは難しく、時間もかかります。しかし誤差逆伝播法を使うことで、局所的な演算について微分すれば良いことになり、これは正確に求めることができます。また、局所的な演算の共通する部分があれば繰り返し使えるため、効率も高まります。ニューラルネットワークはパラメータがとても多いため、このような「正確性」と「高速性」がかなり重要視されているのです。

KaiRA の NF2018 の会誌では具体例を計算グラフを使って説明しています。詳しく知りたい方は是非ダウンロードして頂き、ご覧ください。