

算数からはじめる ディープ・ラーニング

ご購入はこちら

牧野 浩二, 西崎 博光

本特集で使う人工知能のフレームワーク Chainer を使うと、ニューラル・ネットワークの原理があまり分かっていなくても、ディープ・ラーニングのプログラムを作ることができます。しかし、原理が分かっていると、「どのような用途が最適なのか」や「どのようなネットワーク構造にしたらいいか」などが推測でき、よりうまくディープ・ラーニングを使うことができます。

フレームワークを使いこなすための人工知能の原理は、数式で説明されている場合が多いです。ですが数式であれこれと書かれても、なかなか理解しづらいですよ。

ここではディープ・ラーニングを手計算で体験することで、やっていること（原理）を理解しましょう。いきなり難しい計算をするのではなく、ステップ・アップしながら学んでいきます。そこで次のような進め方をします（図1）。

- (1) 1層のニューラル・ネットワーク
- (2) ディープ・ニューラル・ネットワーク
- (3) リカレント・ニューラル・ネットワーク
- (4) オート・エンコーダ
- (5) 畳み込みニューラル・ネットワーク

(3)～(5)はディープ・ラーニングの3大アルゴリズムです。(1)と(2)の練習をした後ならばどこから始めても構いません。画像向きということでよく登場する畳み込みニューラル・ネットワークを最後(次章)に持ってきたのは、独特のフィルタ処理(畳み込みやプーリング)が必要になるからです。

その1…

1層のニューラル・ネットワーク

● 1番シンプル! ここから

ここから少しだけ数式が出てきますが、足し算と掛け算をするだけなので安心してください。ニューラル・ネットワークは図2のように丸と矢印で表されます。この中の x_1 と x_2 が入力を表していて、 y が出力(答え)を表しています。矢印にも w_1 と w_2 という重みが付いています。また、 y には入力が1で重みが b の矢

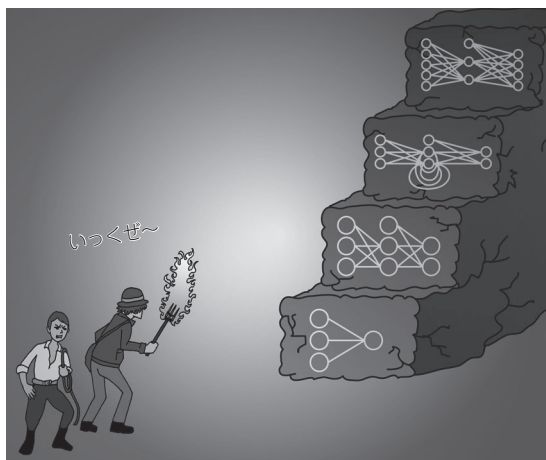


図1 算数を使ってステップ・バイ・ステップでディープ・ラーニングのメカニズムを理解する

印も付いています。これがあった方がニューラル・ネットワークの理論がうまく成り立つので、入力1で重みが b のものがいつも入るとっておきましょう。そして y の値は次の式で計算できます。

$$y = x_1 w_1 + x_2 w_2 + b$$

これは入力に対してそれぞれ決まった重みをかけて足し合わせるというものになります。

▶ 出力を求める

例えば重みが $w_1 = 0.3$, $w_2 = 0.8$, $b = 0.5$ と決まっているものとします。そこに入力として $x_1 = 1$, $x_2 = 0$ が与えられたとします。出力 y は次の計算で求められます。

$$y = 1 \times 0.3 + 0 \times 0.8 + 0.5 = 0.8$$

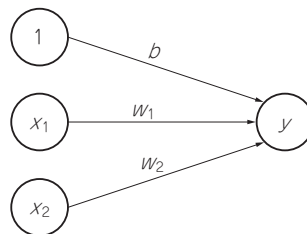


図2 最もシンプルなニューラル・ネットワーク