

自作AIステップ2… ちょっとディープなのを作る

ご購入はこちら

牧野 浩二, 西崎 博光

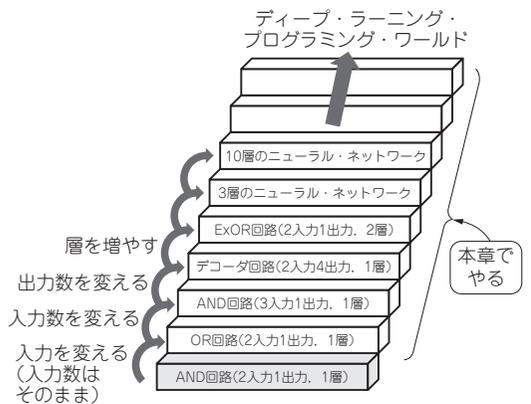


図1 前章で解説した基本ディープ・ラーニング・プログラミング (1段階目) をふまえているいくつかの改造していく

前章ではAND回路を例に、Chainerを使ったディープ・ラーニングのプログラミングを体験しました。このAND回路のディープ・ラーニングができれば、一部を変えるだけでいろいろなタイプのディープ・ラーニングのプログラムになります。

本章を読み進めると、少しずつ階段を上るようにプログラムを変えていながら、いつの間にか10層のディープ・ラーニングのプログラムができるようになります。難しいと思っていたディープ・ラーニングを自力でできるようになりましょう。

2段階目… 入力と出力の関係だけを変えてみる

筆者提供プログラムはor.pyです。以下に出てくる**.pyも含めて本誌ウェブ・ページからダウンロードできます。

<http://www.cqpub.co.jp/interface/download/contents.htm>

● OR回路で試す…入出力の数はANDのまま

図1の階段を1段昇ります。入力の数、出力の数を変えずに、入力と出力の関係だけを変えるためにOR回路を実現してみます。OR回路とは図2で表される

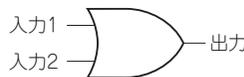


図2 OR回路…前章のAND回路と同じく2入力1出力

表1 OR回路…入力に対する出力の値

状態	入力1	入力2	出力
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

リスト1 OR回路をChainerで学習したときの出力

epoch	main/loss	validation main /main/loss	validation /accuracy	main/accuracy	elapsed_time
1	0.323303	0.323001	1	0.75	0.452863
2	0.323001	0.322701	0.75	0.75	0.806641
3	0.322701	0.322404	0.75	0.75	1.15456
...					
999	0.151991	0.15189	1	1	420.523
1000	0.15189	0.15179	1	1	420.909
input: [0. 0.], result: 0					
input: [0. 1.], result: 1					
input: [1. 0.], result: 1					
input: [1. 1.], result: 1					

電子回路の素子で、AND回路と同じ2入力1出力の回路です。入力に対する出力の値は表1のようになっています。

前章リスト1のand.pyからの変更点は、前章リスト1の27行目の出力だけです。

▶変更前

```
trainy = np.array([0, 0, 0, 1],
dtype=np.int32)
```

▶変更後

```
trainy = np.array([0, 1, 1, 1],
dtype=np.int32)
```

実行するとリスト1の表示が得られます。このプログラムの構造はAND回路と同じようにdotコマンドで変換し、displayコマンドで確認できます。前章図8と全く同じ画像が表示されます。

● やってみよう!

問題 (1)：最初の一步が分かっていることはとても重要です。Chainerで実際に作成し、動作させてみ