Deep Learningの基本

2017/2/14

ディープラーニングをいきなり勉強するのは難しい

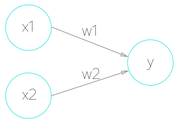
学習の順序

パーセプトロン
↓
単層ニューラルネットワーク
↓
ディープラーニング
(多層ニューラルネットワーク)

パーセプトロン

複数の信号を入力し、一つの信号を出力する

信号は流れる(=1)か流れない(=0)の2値



x1, x2は入力信号、w1, w2は重み、yは出力信号

ニューロン or ノード

入力信号はニューロンに送られる際、重みを乗算(w1x1, w2x2)

この総和がある閾値(θ)を
超えたときのみyは1を出力する　＝ ニューロンが発火する

$$y = \begin{cases} 0 & (w1x1 + w2x2 <= theta) \\ 1 & (w1x1 + w2x2 > theta) \end{cases}$$

重みは各信号の重要性を表している

パーセプトロンで単純な論理回路を作ってみる

ANDゲート

x1	x2	y
0	0	0
1	0	0
0	1	0
1	1	1

上記の表を満たすようなw1, w2, thetaを見つける
そのようなパラメータは無数にある
例えば(w1, w2, theta) = (0.2, 0.2, 0.3)でも満たせる

```
0 * 0.2 + 0 * 0.2 = 0   <= 0.3, y = 0
1 * 0.2 + 0 * 0.2 = 0.2 <= 0.3, y = 0
0 * 0.2 + 1 * 0.2 = 0.2 <= 0.3, y = 0
1 * 0.2 + 1 * 0.2 = 0.4 >  0.3, y = 1
```

同様にNAND, ORも作れる

上記のANDゲートをPythonで実装

```
def AND (x1, x2):
    w1, w2, theta = 0.2, 0.2, 0.3
    if x1 * w1 + x2 * w2 <= theta:
        return 0
    else
        return 1
```

バイアスの導入

theta = -bとする

$$y = \begin{cases} 0 & (b + w1x1 + w2x2 <= 0) \\ 1 & (b + w1x1 + w2x2 > 0) \end{cases}$$

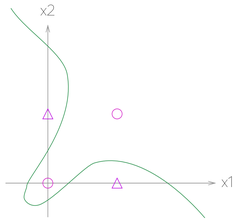
bをバイアスと呼ぶ
バイアスは発火のしやすさを表す
(bが小さいほど発火しにくい)

パーセプトロンの限界

XORゲート

x1	x2	y
0	0	0
1	0	1
0	1	1
1	1	0

どうやってもXORゲートを満たすw1, w2, bを見つけることはできない

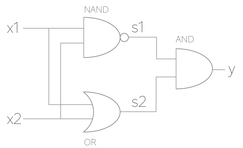


上の図の○と△を分けるには非線形に領域を分割する必要があるが、
パーセプトロンは線形に分けることしかできない。

多層パーセプトロン

パーセプトロンでは領域の非線形な分割はできないが、パーセプトロンの層を重ねることによって実現できる。

XORゲート



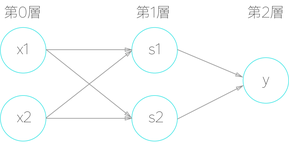
真理値表

x1	x2	s1	s2	y
0	0	1	0	0
1	0	1	1	1
0	1	1	1	1
1	1	0	0	0

Pythonによる実装

```
def XOR (x1, x2):
    s1 = NAND(x1, x2)
    s2 = OR(x1, x2)
    y = AND(s1, s2)
    return y
```

パーセプトロン



ニューロンは3層になっているが重みを持つ層は2層であるため、
2層のパーセプトロンと呼ぶ。
(文献によっては3層と呼ぶこともある)