ファイル更新日時→(2025/03/05 20:25)

GitHubバックアップ日時→(2025/03/05 20:25)

～はじめに～

このノートは「ゼロから作るDeepLearning ～Pythonで学ぶディープラーニングの理論と実装～」について学ぶ際に使用したノートです。

本を読んで気付いたこと、学んだことなどを書いてあります。

また、このファイルはフォルダごとGitHubにバックアップされてあります。(https://github.com/Rino-program?tab=repositories)

このPC上で誤ってファイルを消してしまっても基本は問題ありません。

～プログラミングについて(pythonインタープリタ)～

VSCodeを使用してJupyter Notebook(旧Ipython)でコードを書いています。

使用しているバージョンはMicrosoftstore版3.13.2を使用しています。

構文ハイライターとしてMagicPythonを使用しています。

～この文章を書いている理由～

本の内容がとても多いため混乱する事が多いと感じたから。

～第一章～

この章はpythonやモジュールの説明があるが、すでに基礎を学んでいるため今更学びなおす必要はない。

主に使用するモジュールはNumPy, Matplotlibの二つ。

NumPyはN次元配列など、Matplotlibはグラフの描画やデータの可視化に使われる。

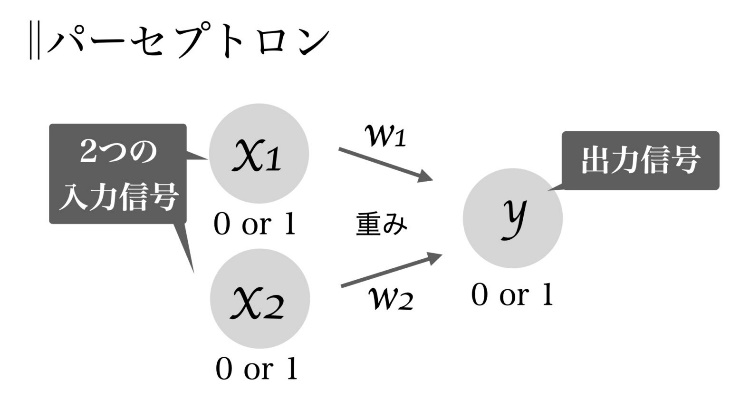
～第二章～

この章はパーセプトロンについて説明がある。

パーセプトロンとは複数の信号の入力を受け取り、一つの信号を出力する物。

一般的にはパーセプトロンの図ではこう表せる。（本も同様）

* 〇はニューロンまたはノード、矢印は重みを表す。



それぞれの重みには固有の値があります。

ニューロンでは送られてきた信号の総和が計算され、その総和が限界値を超えた場合のみ1を出力する。

本ではその限界値を閾値と呼び、記号として表す。

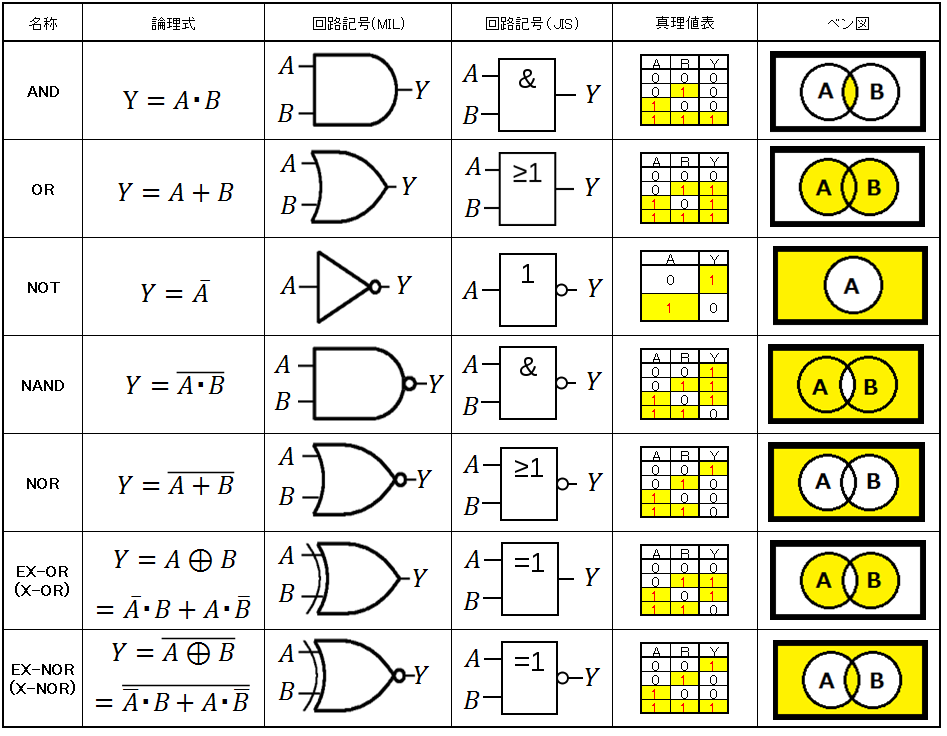
xは入力信号、wは重み、yは出力信号としたとき、式はこうなる。

この式では、入力と重みを掛け合わせ、足し合わせた値が閾値に対応しているかどうかを判定する。

式を見ればわかるが、yの値は0と1のみが出力されることとなる。

二章3.1までではこの重みや出力信号を変え、AND, NAND, ORゲートの3種類の論理回路を作成している。

(論理ゲートの入出力について)



二章3.2以降では新たにバイアスを導入している。

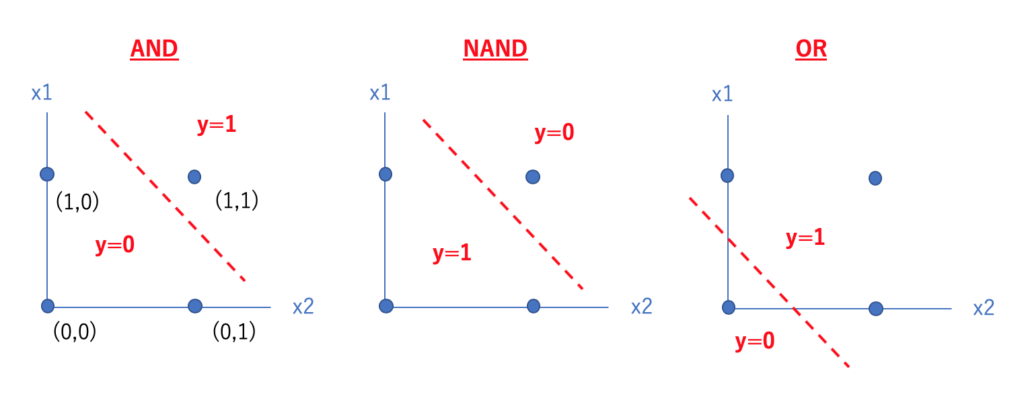
この本ではを-bとして表している。(一般的にもこう表す事が多い)導入した式はこうなる。

重みは、各信号の重要性をコントロールする要素として働く。重みが重いほどその信号に対する重要度が増します。

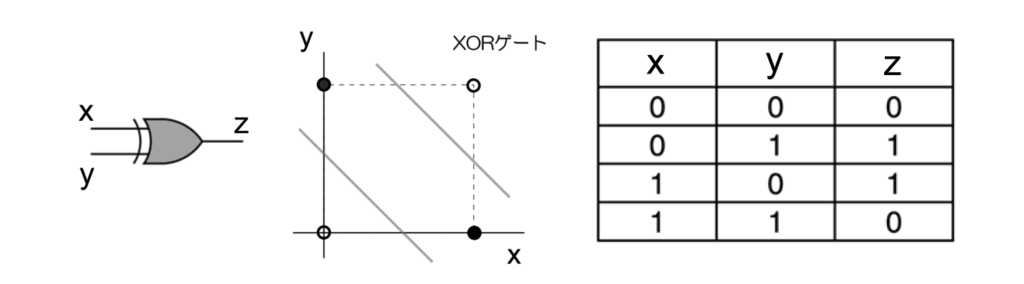
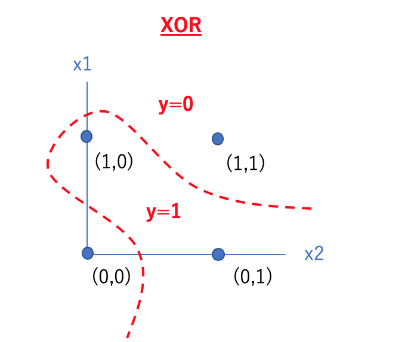
二章4ではパーセプトロンの限界について書いている。

この章ではパーセプトロンの限界として、XORゲートを使用して説明している。

XORでは他の論理ゲートとは違う点がる。例えば、



AND, NAND, ORは0と1の境界線が一本の直線で表せる事ができるのに対し、XORでは、

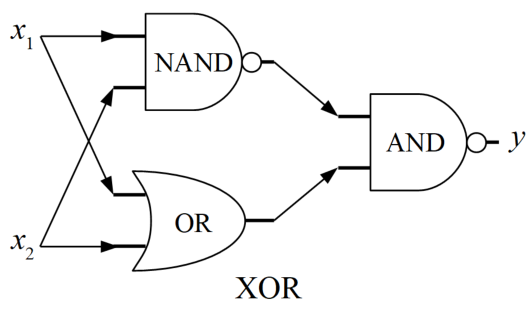


このように直線で表す事ができなかったり一本の直線で表す事ができなかったりするためである。

そこで、パーセプトロンを何層か重ねることで実現する事ができます。

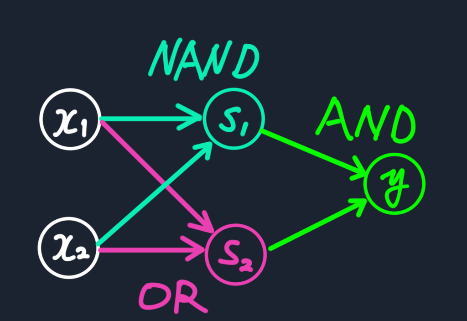
まず、パーセプトロンは一層だけではなく何層にも重ねられます。

つまり論理ゲートを組み合わせて作る事ができるXORゲートは多層パーセプトロンで実現する事ができます。



このように表す事でXORゲートを実現する事ができます。

パーセプトロンで表すと、



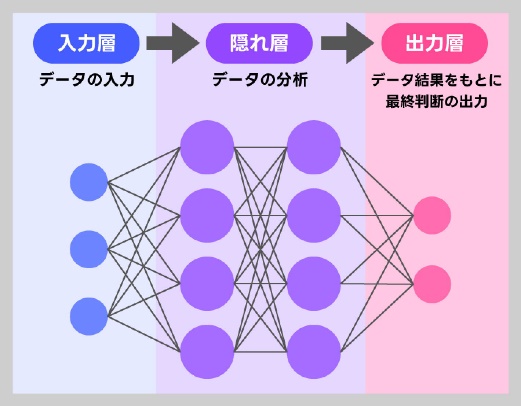
このようになります。見た通りですね。

このような多層パーセプトロンは複雑な回路を組む事も出来ます。ex)加算器, 2→10進数エンコーダーなど

そして、コンピューターも作る事ができます。

NANDゲートだけでコンピューターが組めるともいわれているらしい。（テトリスも作る事が可能とのこと）

～第三章～ニューラルネットワーク



ニューラルネットワークについて