自由ヶ丘高等学校
ＡＩ研究チーム（魚橋憲司　安永匠）

Python によるＡＩの研究

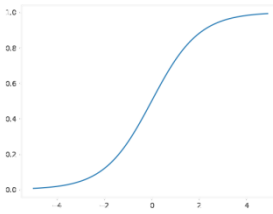
参考:斎藤　康毅　著
「ゼロから作る DeepLearning」

1．シグモイド関数の特性について

シグモイド関数の式は $f(x)=\frac{1}{1+e^{-x}}$であり、e はネイピア数と言って２．７１８３・・・という数字である。

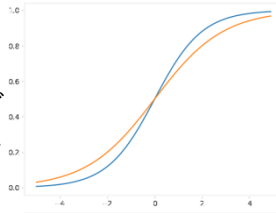
e の部分がよくわからないので、ここを変えたらどうなるのかを考えてみた。

右のグラフはただのシグモイド関数である。



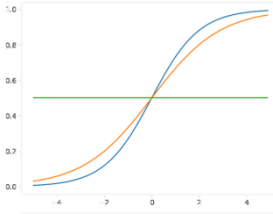
①e の部分を２にしてみる。
（sig2 と名付ける）

オレンジが新しい関数である。シグモイド関数と比べて傾きが緩やかになった。



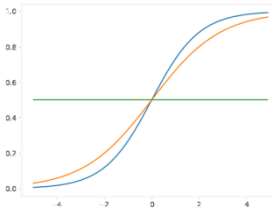
②e の部分を１にしてみる。

緑が新しい関数である。縦軸の 0.5 の場所に真横に伸びる直線ができた。

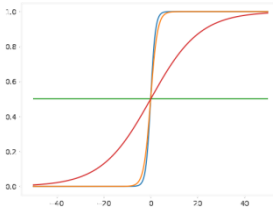


③e の部分を１．１にしてみる。
(sig1.1 と名付ける)

ピンクが新しい関数である。右上がりの直線のように見えた。しかし、Ｘの範囲を－５０～５０にすると、シグモイド関数のようなグラフになっていた。

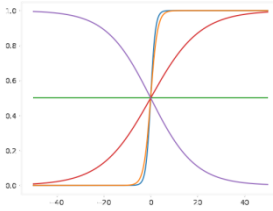


このように、e の部分を１に近づけると傾きが緩やかになっていき、１にすると真横の直線になることが分かった。



④e の部分を０．９にしてみる。
右下がりになった。

このように、e の部分を１よりも小さくすると。右下がりになってゆくことが分かった。



では、シグモイド関数の代わりに sig2 や sig1.1 が使えないか考える。シグモイド関数と sig2 の実行速度を測ってみた。

```
def speed(f):
    x = np.arange(-10, 10, 0.000001)
    t1 =time.time()
    y = f(x)
    t2 =time.time()
    print(t2-t1)
speed(sigmoid)   # 0.6935763359069824
speed(sig2)      # 1.5600404739379883
```

シグモイド関数はおよそ 0.69 秒で sig2 はおよそ１．56 秒だった。AI では計算で時間がかかるため実行速度は速い方がいい。したがって、シグモイド関数を使った方がいいということが分かった。

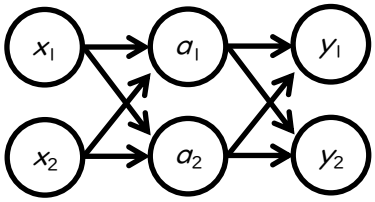
2. DeepLearning の様子をグラフで見てみたい

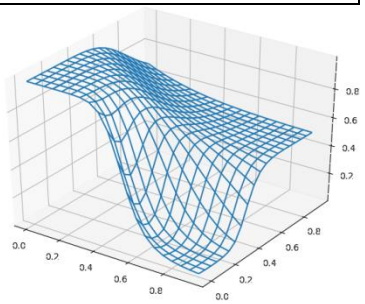
１つのニューロンに注目するとシグモイド関数が使われているので出力が滑らかに変化することは分かったが、２つ以上のニューラルネットワークだと出力がどのように変化しているのかが気になった。

そこで、下の図のようなニューラルネットワークを作り、２つの入力から、出力 y_1 がどのように変化するのかを３Ｄグラフで表してみた。

・重み、バイアスの値（重みはランダムで与えた）

```
W１    [[  5.07739125  -3.52745036]
       [-15.10745431  -4.08228085]]
ｂ１    [0. 0.]
W２    [[ -6.17089246  -1.09504476]
       [ 16.07542025 -20.02472672]]
ｂ２    [0. 0.]
```





簡単なニューラルネットワークだけど、結構複雑な形になった。さらにニューロンを増やすともっと複雑な形になることがわかった。

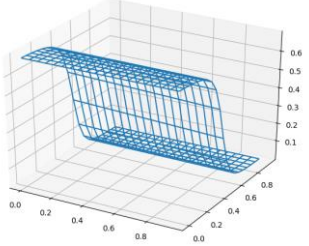
3. 学習した出力をグラフで見てみたい

ニューラルネットワークを使えば隠れ層が１層であっても画像の認識ができる。今回はその原理が知りたかったので、理想形を考えた。隠れ層は１層とした。

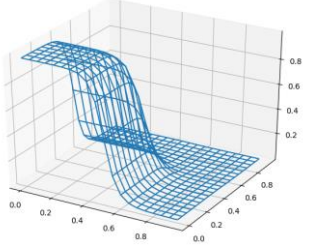
・入力データ
　x_1，x_2：０から１までのランダムな値
・教師データ
　t_1：0($0\leqq x_1<0.5$，$0\leqq x_2<0.5$)
　t_2：1($0\leqq x_1<0.5$，$0.5\leqq x_2<1$)
　t_3：2($0.5\leqq x_1<1$，$0\leqq x_2<0.5$)
　t_4：3($0.5\leqq x_1<1$，$0.5\leqq x_2<1$)

（予想）・・・　出力 y_1 は($0\leqq x_1<0.5$，$0\leqq x_2<0.5$)の範囲で１に近い値になる。

（結果）・・・　100 エポック学習させた。



ニューロンの数が１つの場合



ニューロンの数が２つの場合

だいたい予想した通りの結果が確認できた。ニューロンの数が１つの場合は１直線に沿ったシグモイド関数のような形になって、誤差が大きめになることが分かった。