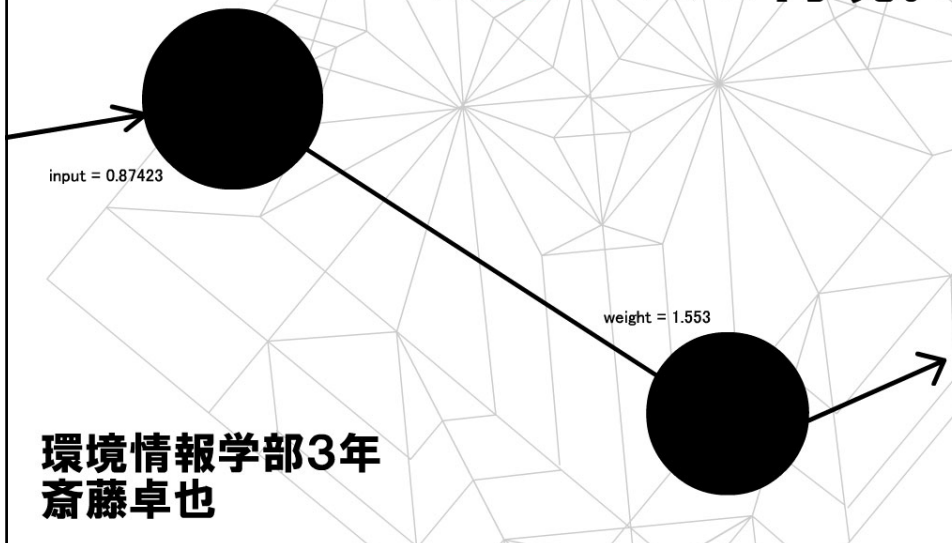


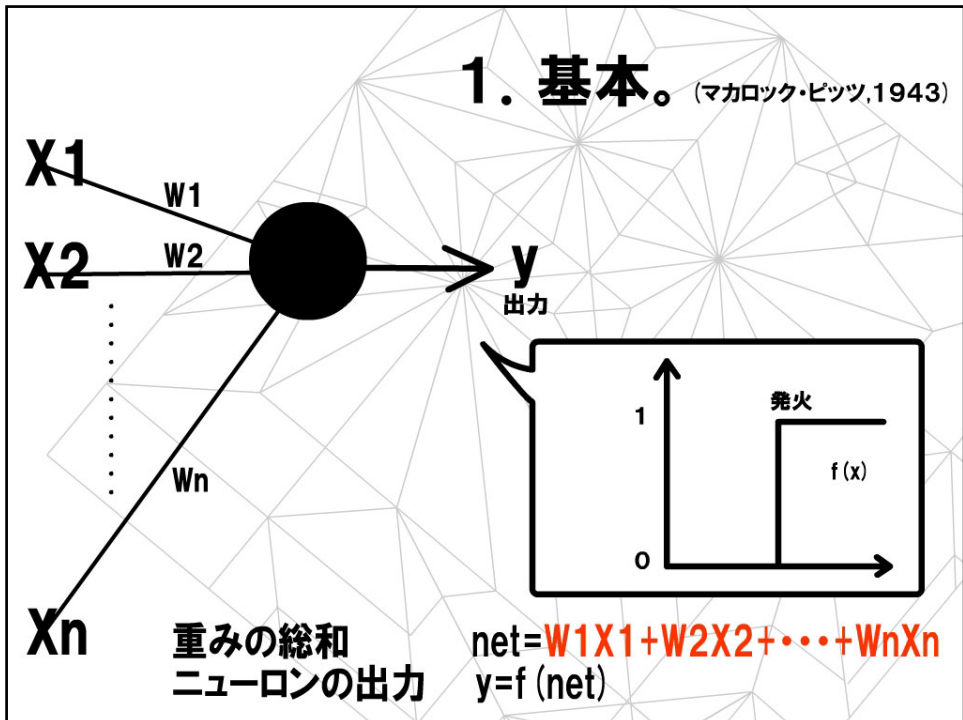
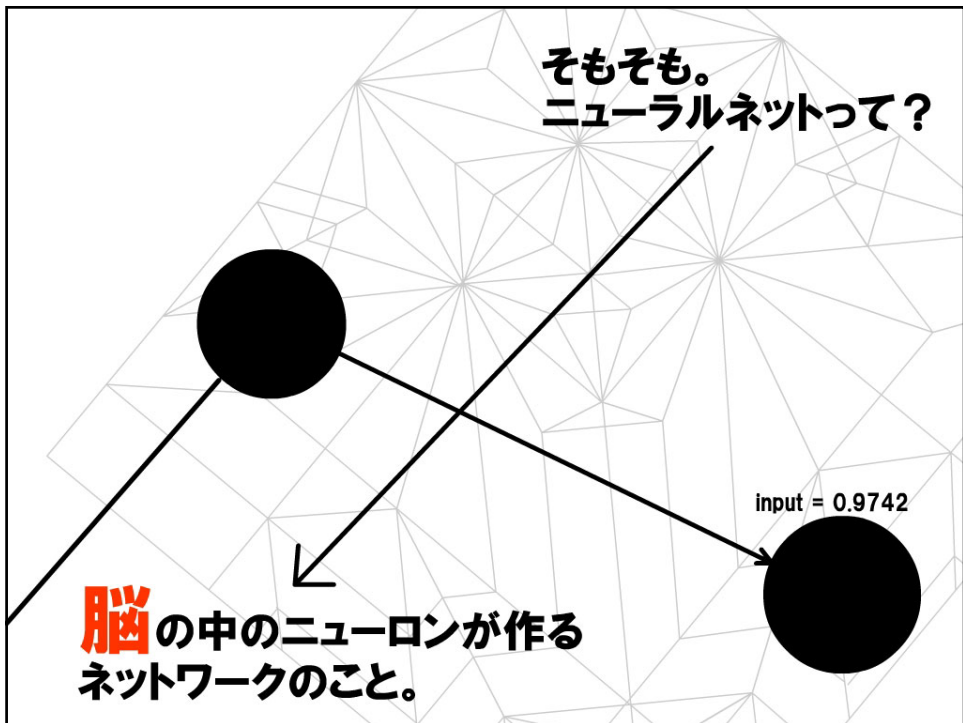
ニューラルネットワークの BESP上での再現。



環境情報学部3年
齋藤卓也

本日のお品書き。

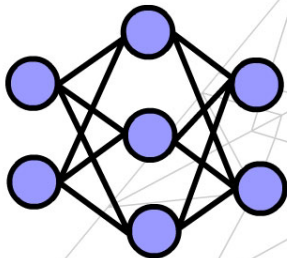
1. ニューラルネットワークとは？
2. アルゴリズムの説明
(簡単に！)
3. モデルの説明
4. 演習



重み?

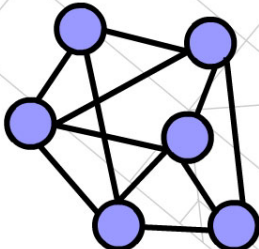


モデルのおおまかな分類



階層型ネットワーク

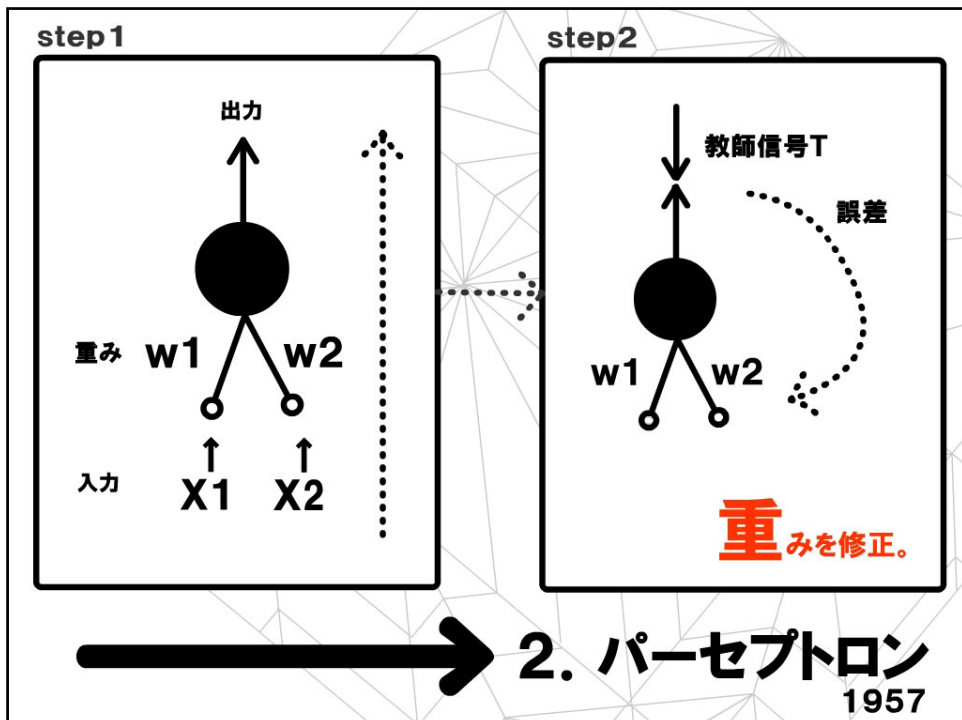
← フィードバックなし。
学習し、重みを調整する。



非階層型ネットワーク

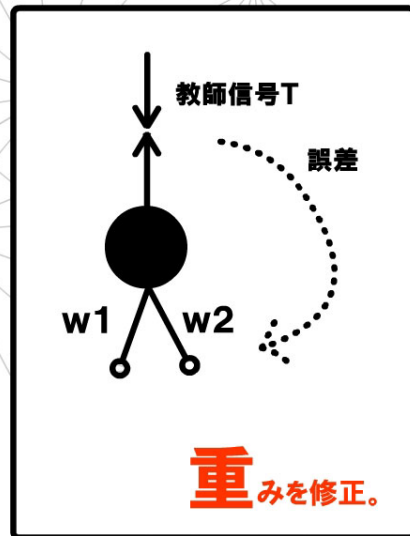
← フィードバックあり。
学習なし。
nクイーン問題や
ナイトツアー問題。

では、学習モデルの発展
について説明しましょう。



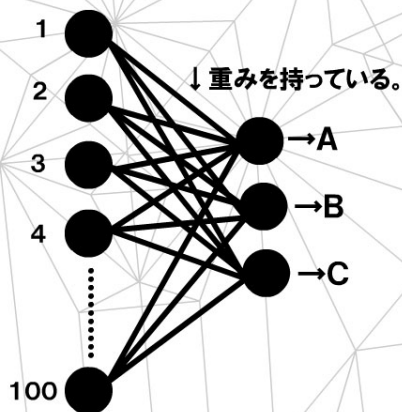
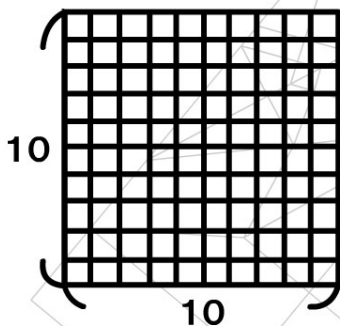
→ 誤差が大きいほど、
修正も大幅。

→ 学習モデル。
(文字認識など)

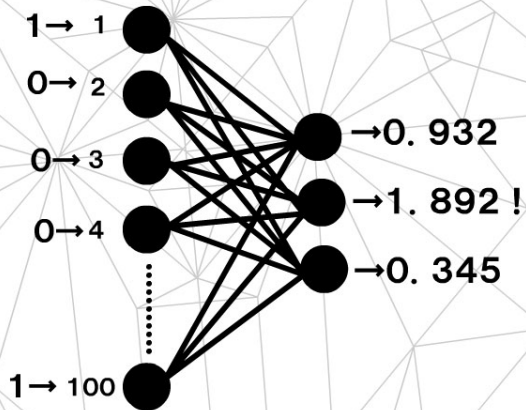
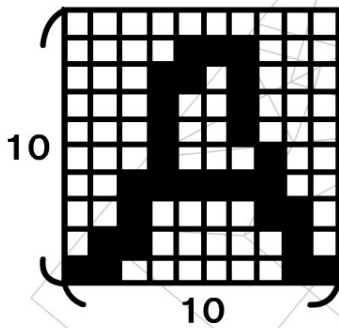


→ 2. パーセプトロン
1957

文字認識はこんな感じ。



文字認識はこんな感じ。



→ B ?

ニューラルはすごい！

- 最適解を見つけていくモデルなので、コンピュータの苦手なあいまいな領域を扱える利点。(文字認識など)
- このモデルは現実の人間の脳を正確に表したものではありません。

生物ニューラルネット → 人工ニューラルネット
→ もともとは神経回路網の機能を解明を目指したものだが、情報工学の広い分野に適用可能である。



されどもっ。



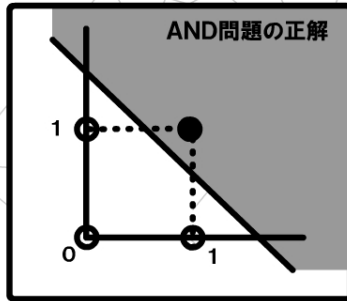
**XOR問題(非線形問題)が
解けない!**

↓ パーセプトロンの限界。

XOR問題とは？

補

→まずはAND、ORから。



○ = 0 ● = 1

AND問題

1 1 → 1
1 0 → 0
0 1 → 0
0 0 → 0

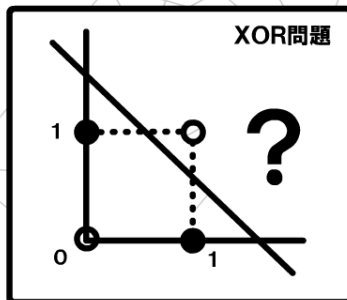
OR問題

1 1 → 1
1 0 → 1
0 1 → 1
0 0 → 0

→直線で正解を得れる。
=線形問題

XOR問題とは？

補



○ = 0 ● = 1

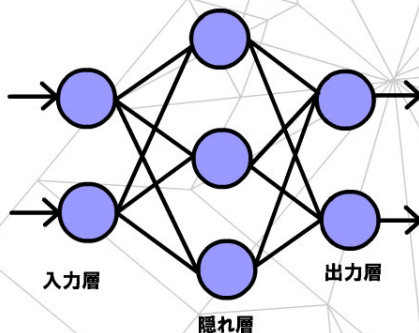
XOR問題

1 1 → 0
1 0 → 1
0 1 → 1
0 0 → 0

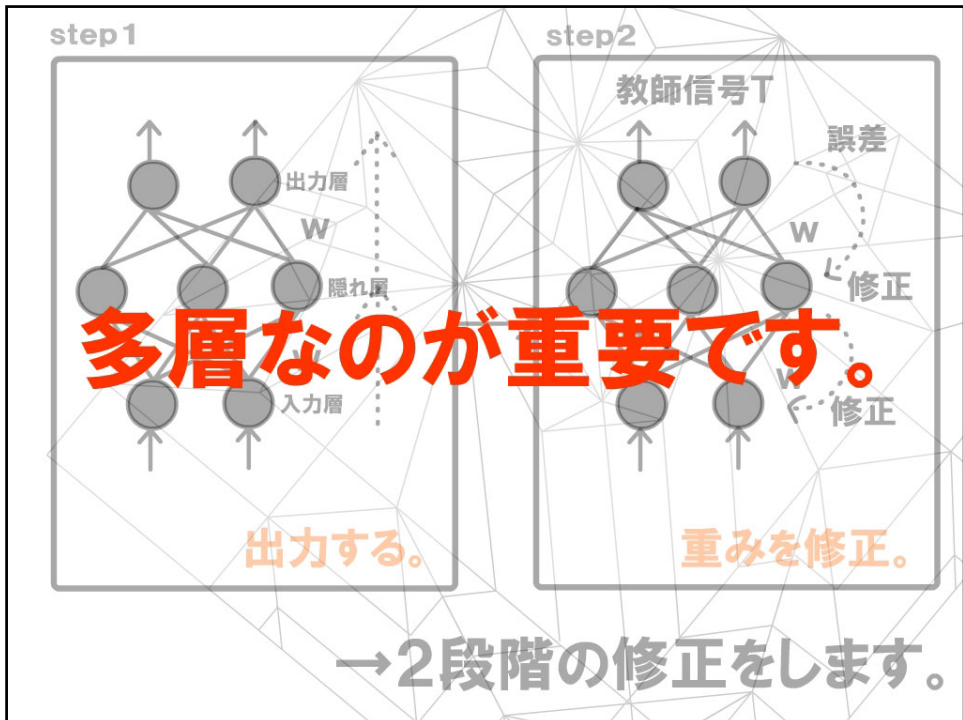
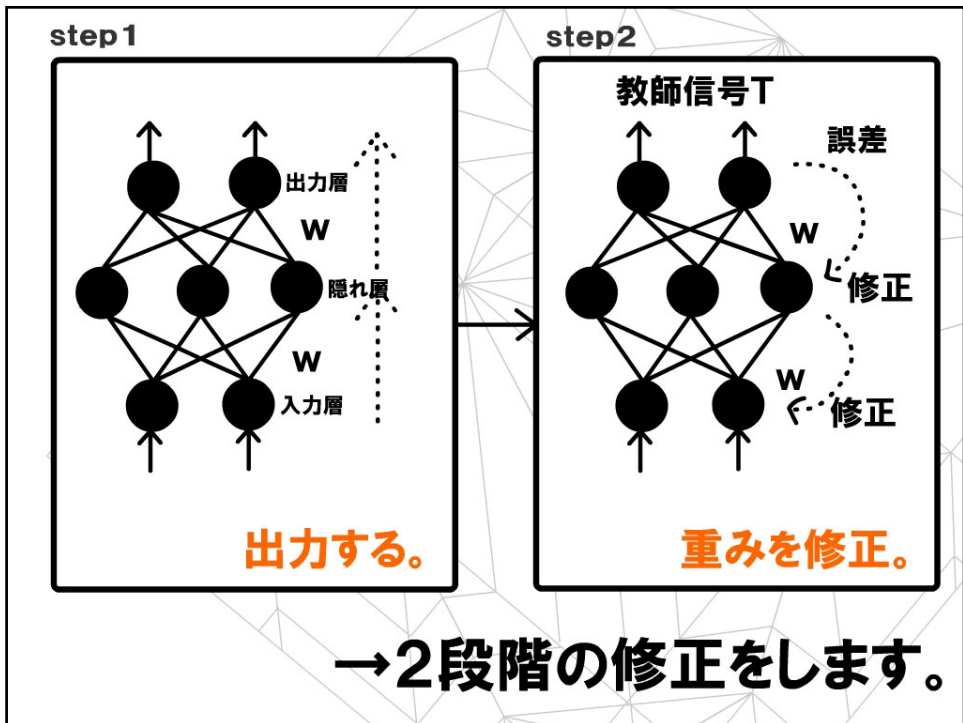
→直線では正解できない。
=非線形問題

冬...。

3. バックプロパケーション (誤差逆伝播学習)



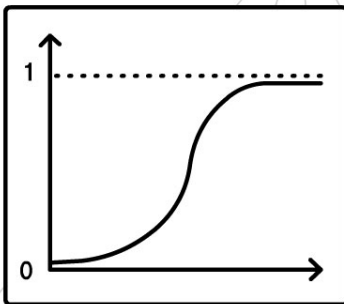
パーセプトロンを**多**層にすることで
非線形問題を解決。



そもそもBPはいかにして可能か？

—アルゴリズムの説明—

→ BPはいかにして可能か？

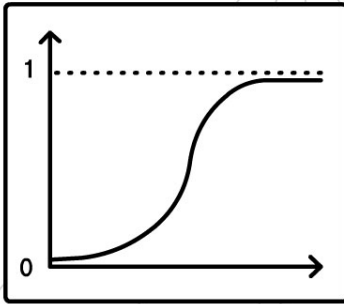


シグモイド関数を用いる。

$$f(x) = \frac{1}{1 + e^{-x}}$$

シグモイド関数を使うことによって
多層のパーセプトロンの誤差修正が可能になった。

→ BPはいかにして可能か？



微分ができる！

- 微分は傾きを知ること。
- グラフの傾きを知ることによって、どのくらい修正すればよいか分かる。

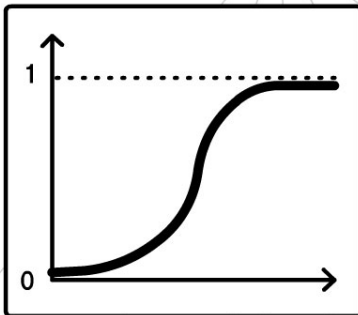
$$f(x) = \frac{1}{1 + e^{-x}}$$

シグモイド関数

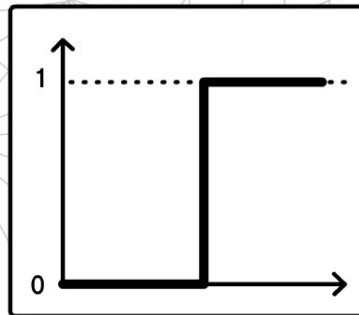
$$f'(x) = X(1 - X)$$

微分した値
とても単純な式に。

→ BPはいかにして可能か？



シグモイド関数



マカロック・ピッツ

→ **グラフが似ている。**

ゆるやかなしきい値として。

→ BPはいかにして可能か？

修正値 $\delta =$

$$\underbrace{\alpha}_{\text{学習係数}} \times \text{誤差} \times \underbrace{\text{出力} \times (1 - \text{出力})}_{f'(x) \text{ を微分した値}} \times \underbrace{\text{入力}}_{\text{自分への}}$$

上の式が重みに足されていき、重みが修正される。

→ BPはいかにして可能か？

修正値 $\delta =$

$$\underbrace{\alpha}_{\text{学習係数}} \times \text{誤差} \times \underbrace{\text{出力} \times (1 - \text{出力})}_{f'(x) \text{ を微分した値}} \times \underbrace{\text{入力}}_{\text{自分への}}$$

上の式が重みに足されていき、重みが修正される。

誤差が大きければ大きいほど、
修正の値は大きくなる。

→ BPはいかにして可能か？

修正値 $\delta =$

$$\underbrace{\alpha}_{\text{学習係数}} \times \text{誤差} \times \underbrace{\text{出力} \times (1 - \text{出力})}_{f'(x) \text{ を微分した値}} \times \underbrace{\text{入力}}_{\text{自分への}}$$

上の式が重みに足されていき、重みが修正される。

微分した値

傾きにより修正の具合を変えるため。

→ BPはいかにして可能か？

修正値 $\delta =$

$$\underbrace{\alpha}_{\text{学習係数}} \times \text{誤差} \times \underbrace{\text{出力} \times (1 - \text{出力})}_{f'(x) \text{ を微分した値}} \times \underbrace{\text{入力}}_{\text{自分への}}$$

上の式が重みに足されていき、重みが修正される。

**入力された値にも
影響を受ける。**

→ BPはいかにして可能か？

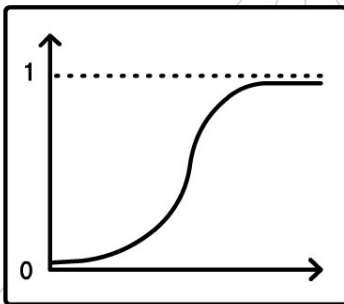
修正値 $\delta =$

$$\underbrace{\alpha}_{\text{学習係数}} \times \text{誤差} \times \underbrace{\text{出力} \times (1 - \text{出力})}_{f(x) \text{ を微分した値}} \times \underbrace{\text{入力}}_{\text{自分への}}$$

上の式が重みに足されていき、重みが修正される。

学習係数 α
全体を調整するため。

→ BPはいかにして可能か？

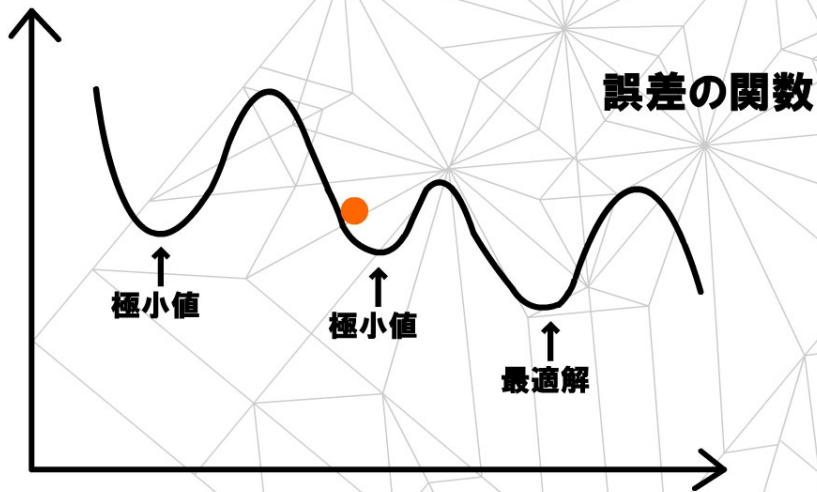


シグモイド関数を用いる。

- 微分が可能に。
- 傾きから誤差の修正具合を知る。
- 微分が容易である。
- マカロックピッツのグラフと酷似

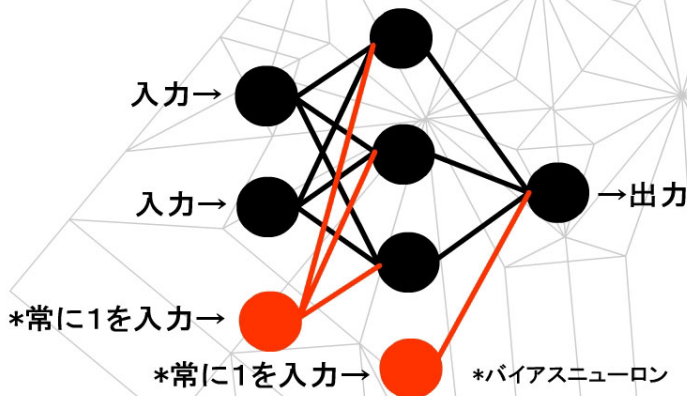
$$\underbrace{\alpha}_{\text{学習係数}} \times \text{誤差} \times \underbrace{\text{出力} \times (1 - \text{出力})}_{f(x) \text{ を微分した値}} \times \underbrace{\text{出力}}_{\text{一つ前の}}$$

→ローカルミニマムを乗り越えよ！



最適な**学**習係数を知らなくては！

→ローカルミニマムを乗り越えよ！



バイアスニューロンでより早く最適解を！

最適解をすばやく探すために。



- **適度な学習係数**

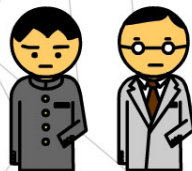
小さすぎると、極小値にはまる。

大きすぎると、振動して収束しなくなる。

- **中間層の数を変える**

どうすれば早くできるか考えてみる。

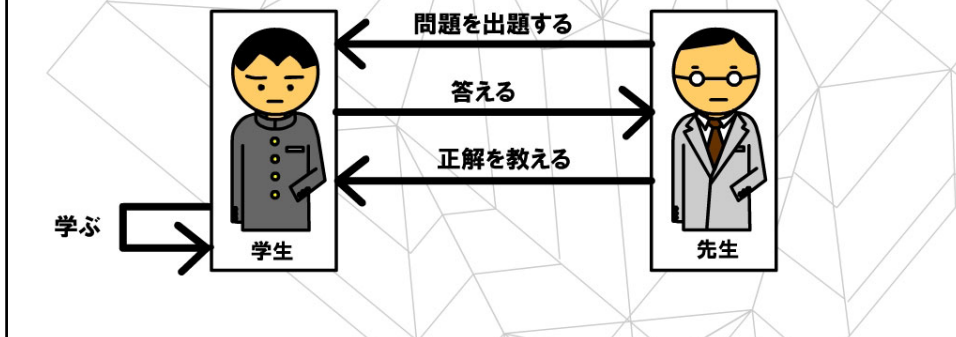
モデルの説明



先生が生徒にAND, OR, XOR問題を教えるモデル。

Teacher AgentとStudent Agentが登場します。

先生は、問題を出題し答えを生徒におしえます。
生徒は学習していきます。



A screenshot of a simulation titled '学習の様子' (Learning Status) for an XOR problem. The top part shows a dashed box containing a 'Student' and a 'Teacher' agent. The Student's 'NeuralNetwork' component is in a state 'waitingForCorrectAnswer'. The Teacher's 'Teach' component is in a state 'waitingForReply=>waitingForTime'. Communication is shown with arrows labeled 'neuralnetwork_ClassRoom' and 'neuralng_正解だよ。' (Correct answer). Below this, there are two windows. The left window is a plot showing a signal that starts with high-frequency oscillations and then settles into a steady state. The right window shows a neural network diagram with three input nodes (labeled 1, 0, 1) and one output node. The output node is currently displaying '0', which is crossed out with a red 'X', and a '1' is shown below it, indicating the correct output for the given inputs.