

第3章 推論・学習の演算を並列に実行するアーキテクチャを動かす

元祖AI「単純パーセプトロン」の実装② Verilog HDLで記述する

本章では、第1章と第2章で紹介した「単純パーセプトロン」をVerilog HDLを用いてデジタル(RTL)実装してみます。

● 演算の流れ

ここで実装する単純パーセプトロンは、図1に示すように、入力 x_1, x_2 を受けた出力 z が所望の出力(S)と等しくなるように重み w_1, w_2, w_3 を更新(学習)していく装置です。

2入力1出力(すべて1ビット)の単純パーセプトロンに、論理和(OR)関数を学習させる計算例は次のとおりです。

- (1) 重みの初期値を与える
例： $(w_1, w_2, w_3) = (1, -2, 3)$
- (2) 入力を与える
例： $(x_1, x_2) = (0, 0)$
- (3) 入力と重みの積和演算の結果と活性化関数を用いて結果を出力する(図2)

- 積和演算結果： $y = x_1w_1 + x_2w_2 + 1 \cdot w_3 = 3$
- $f(y)$ をステップ関数とした場合の推論結果：
 $z = f(y) = 1$
- (4) 推論結果と教師信号を比較する(図3)
入力が(0, 0)なので、教師 S (所望の出力=入力の論理和)は0. 推論結果は1なので、教師と一致しない
- (5) 重みの更新を行う。
例： $(w_1, w_2, w_3) = (1, -2, 2)$ (w_3 だけ更新)
- (6) さまざまな入力を与えて、どのような入力に対しても理想の値が得られるように(2)~(5)の手順を繰り返す

単純パーセプトロン専用回路のスペック

ここでは、2入力1出力(各1ビット)の単純パーセプトロンの専用回路を考えてみましょう。まずは、スペックを考え、その後で必要なブロック・モジュールを考えていきます。まずは、次について決めておく必要があります。

● 活性化関数の種類

活性化関数(図1, 2の f)は、人工知能の推論出力(z)を決める関数です。一般的にはシグモイド関数や、ReLU関数などが用いられます。本章では、実装が簡単なステップ関数(入力が正の数であれば1, 負または0であれば0を出力する関数)を用いることにします。

● 学習率の値

図3に示した学習率 η は、学習の効率・性能を決定するハイパーパラメータです。通常は $0 < \eta \leq 1$ の範囲となります。ここでは、単純パーセプトロンの全ての演算を整数を行えるように $\eta = 1$ とします。1以下の少数(固定少数)とする場合は、演算回路全体にわたり小数点の位置を定義しなければならず、表現が複雑になります。

● 扱う値の符号・ビット幅

簡単化のため、図1~3に示した入力 x_1, x_2 と出力 z は1ビットとしましょう。また、重み(w_1, w_2, w_3)の値の範囲は、 $-8 \sim 7$ の離散整数値とします。4ビットの2の補数表現で扱うことができます。すると、入力と重みの積も $-8 \sim 7$ の範囲の整数となり4ビットで表されるので、それらの和である y は

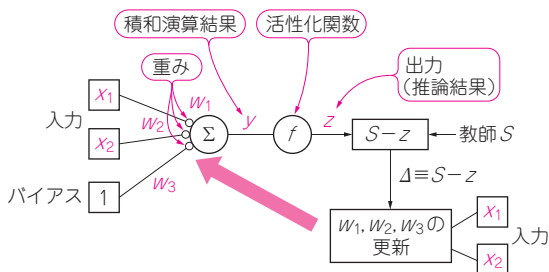


図1 本章では単純パーセプトロンをVerilog HDLを用いてデジタル実装してみる
すべて1ビットの単純パーセプトロンに、論理和(OR)関数を学習させる計算例

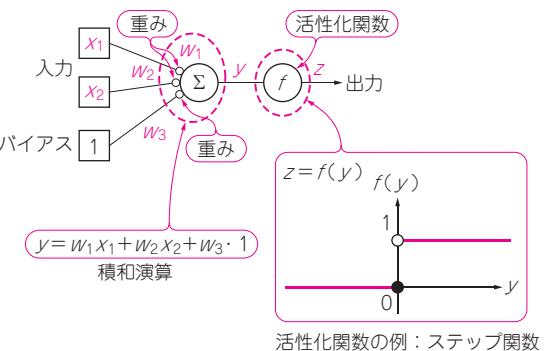


図2 推論の演算
入力と重みの積和演算の結果と活性化関数を用いて結果を出力する