第2章 ハードウェアAIの要素「乗算器」はANDゲート1個で作れる

元祖AI「単純パーセプトロン」の実装① 確率的コンピューティングを使う

● ブレッドボード上のロジックICと，Arduinoで人工知能を体験する

ディジタル乗算器の内部では，基本的には小学校で習う2けた以上のかけ算と同じこと（九九による部分積とけた上がり・部分積の和の計算）が2進数で行われています．高速なものは大量の部分積を一気に加算する高速加算器のお化けのような巨大な構造を持っています．半導体チップ上の面積は限られているので，巨大なディジタル乗算器を大量に搭載して並列乗算を行うことは一般的ではなく，GPUなどの専用デバイスを活用します．

乗算を，原理的にはANDゲート単体で行うことができる方法が存在します．それが「確率的コンピューティング」です．歴史を遡ってみると，その源流は1960年代にあるようです［文献(1)］．これをうまく利用すれば，ブレッドボード上でも本格的な人工知能や，乗算を多用するその他の回路を組める可能性があります．

本稿ではその可能性を探るために，ブレッドボード上のロジックICと，Arduinoを動かしながら乗算や重み付き加算の基礎を学んだり，確率的コンピューティングを体験したります．具体的には，次の内容を解説します．

(1) 確率的コンピューティングによる乗算と重み付き加算のしくみ
(2) ブレッドボード上の1つの論理ゲートで乗算または重み付き加算ができることを体感
(3) 確率的コンピューティングを用いて人工知能の卵とも言うべき「単純パーセプトロン」を実装

単純パーセプトロンは最も簡単な人工知能です．その基本演算である「積和演算」は最先端の人工知能と同じです．確率的コンピューティングによる単純パーセプトロン実装を経て，より大規模な人工知能実装への可能性を感じていただければと思います．

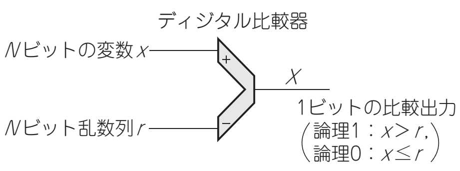

図1　変数のエンコード方法
変換したい変数xと一様乱数列rの大小をディジタル比較器で比較することで，変数の大きさを時系列Xに変換する（単位時間当たりの論理1の数がxに比例）

確率的コンピューティングの基礎知識

■ エンコードとデコードの基本

確率的コンピューティングは，行いたい演算を確率事象の演算に対応させる相似計算です．これを行うには，まず演算したい数値を1ビットの時系列信号に変換（エンコード）します．次に必要な演算を行い，その演算が完了したら演算結果（1ビットの時系列信号）をもとの数値表現に戻す（デコードする）必要があります．その手法について説明します．

● エンコード…数値を1ビットの時系列に変換する

図1に示すのは，ディジタル比較器です．ディジタル回路の場合は，本図に示すように変換したい変数x（“N” ビット，正の値）と，それと同じビット数の一様乱数列r［線形帰還シフト・レジスタ（LFSR）などでつくった疑似乱数］を用意します．それらの大小をこの比較器で比較します．

ここで，変換したい“N”ビット変数xと乱数列rを1に規格化した値変数をそれぞれ$\bar{x}$（$\equiv x/2^N$），$\bar{r}$（$\equiv r/2^N$）で表すと，この比較器の出力Xは確率$\bar{x}$で論理1となる（確率$\bar{r}=1-\bar{x}$で出力が論理0となる）2値の時系列信号となります．例えば，$\bar{x}$が0.1であれば，$\bar{r}$は確率0.9で0.1よりも大きい値となることから，確率0.1でXは論理1です．この変換（変数xと乱数列rの比較）を，ここでは「エンコード」と呼ぶことにします．通常表記の数値xを，確率的コンピューティングの世界で扱う時系列X（確率$\bar{x}$で論理1となる時系列）にエンコードしたのです．これが確率的コンピューティングを行うための入口（最初の準備）になります．

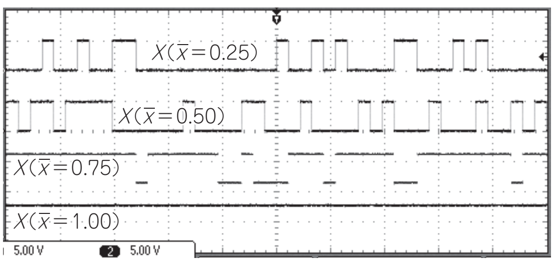

図2　変数のエンコード例
Arduinoで確率$\bar{x}$の値が異なる4種類の時系列Xを生成してオシロスコープで観測したもの．縦軸は4種類の時系列X（5V/div），横軸は時間（100ms/div）