FPGA人工知能のポテンシャルを探る

第5回　カメラで取り込んだ手書き文字のAI画像認識

鈴木 量三朗

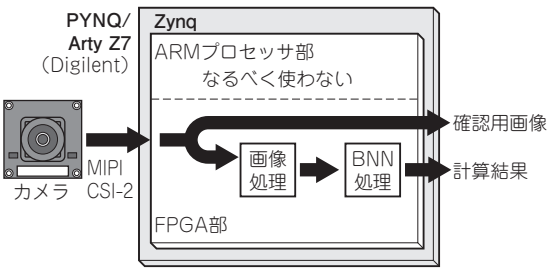

図1　カメラからのデータを直接取り込んでなるべくCPUの介在なしにBNNによる分類をする

本連載ではFPGAで威力を発揮するといわれているニューラル・ネットワークBNN（Binarized Neural Networks）のポテンシャルを探ります．今回は，カメラから取り込んだ画像を対象としてBNNによる手書き文字の認識を行います（**図1**）．

● 目標…CPUの介在なしにBNNによる分類を行う

PYNQと呼ばれるFPGAボードの実行環境では，PythonやJupyter Notebookを使ってBNNを試すことができます（連載第1回，2018年6月号を参照）．画像はUSBカメラやHDMI，静止画像など，Linuxで用意できる入力ソースを利用可能です．

Pythonを利用すると結果をグラフ化できるなど，使い勝手は非常に良くなります．しかしBNN自身のデバッグ環境としては，PYNQという評価ボードの上ではなく，PC上の方が格段に上となるでしょう．

前回（第4回，2018年9月号），高位合成ツールVivado HLSのテストベンチを作成し，PC上で画像処理の確認をするような環境を作りました．その処理と合わせて動作し，BNNのデバッグができる環境を今回は作っていきます．同時にCPUがなるべく介在しないような構造に変更していきます．

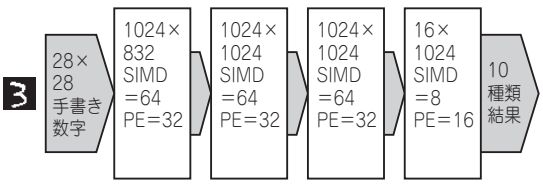

図2　PYNQボードで動かしたBNNネットワーク

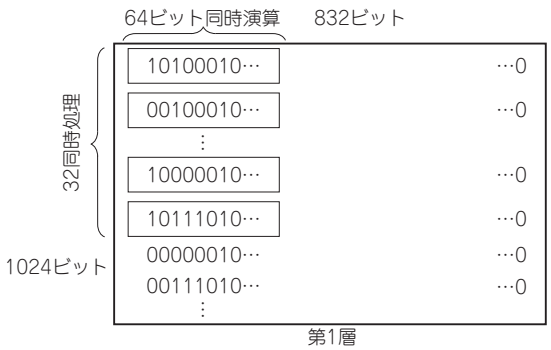

図3　FPGAで処理を並列化する

カメラ入力への対応

● BNNのパラメータ

PYNQ-BNNのネットワークの構成（モデル）を改めて見ていきましょう（**図2**）．

第1層では，入力は28×28を64で切り上げた832ビットで，出力は1024ビットです．

通常のニューラル・ネットワークであれば浮動小数点数を使用した行列計算をします．BNNのデータは2値化された値であり，それらをXNORを使って832×1024の行列演算に相当する計算をし，結果として得られたビットの数を足します．

このとき，FPGAの特徴を生かして処理を並列化します（**図3**）．ソースコードでは，#pragmaで指定し