FPGA人工知能のポテンシャルを探る

第7回　量子化QNN物体認識で必要になるPython浮動小数点演算

鈴木 量三朗

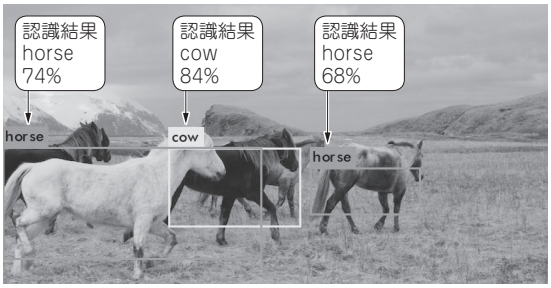

図1　FPGA向きQNNの物体認識では浮動小数点演算が必要

前回から，ザイリンクスの量子化ニューラル・ネットワーク（QNN[注1]：Quantized Neural Network）を取り上げています．参照している実装[(1)]ではPythonで浮動小数点の畳み込み演算をしています．

最終的に全てをFPGAで処理するため，今回はPythonによる処理の概略をつかみます．

FPGA物体認識で必要なもの

● 画像向きなCNNを使いたい

物体検出（Object Detection）プログラムYOLO（You Only Look Once，前回紹介，**図1**）は，いわゆるCNN（Convolutional Neural Network）の1つです．畳み込みと呼ばれる，画像処理ではおなじみの技術を使っています．特にFPGAではエッジ検出などで取り扱われることが多い技術です．

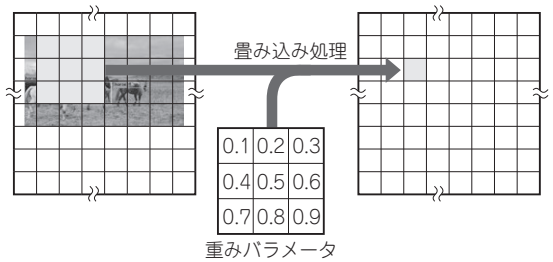

図3　CNNではフレームで評価

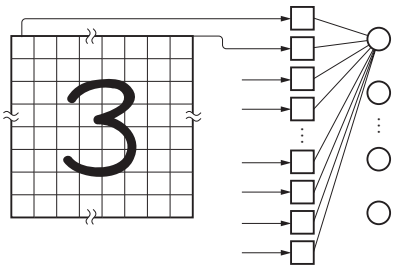

図2
BNNの手書き数字認識ではフラットに評価

BNN（Binarized Neural Networks）の手書き数字認識でも画像を扱っていましたが縦横を無視してフラットにして処理をしていました（**図2**）．一方，CNNは，画像フレームを保存するかのように処理を進めていきます（**図3**）．その点は画像処理，とりわけ物体認識には都合の良い方式だといえます．

そこで今回はFPGAによる物体認識にYOLOを使います．

● 精度が必要なためPythonで浮動小数点演算を行う

YOLOは，正確なクラス分けをするために計算精度が必要になります．

紹介したPYNQ上の実装[注2]では，精度の必要な入口と出口に関してはPython（とNumPy）とdarknetを利用して浮動小数点数の計算をしました．中間の全てのレイヤをFPGAに特化した形で実装したQNNという技術を使いました．1ビットの重み情報と3ビットのバイアス値の情報を使って並列に計算するという独自のアプローチのものでした．レイヤは8層で最新のYOLO v3の100を超える層と比べるとディープではあ

注1：QNNと言えば量子ニューラル・ネットワーク（Quantum Neural Network）を指すことが多い．ザイリンクスのQNNは意味が異なるので注意．

注2：正確にはtinier YOLO．本家のtiny YOLOをさらにコンパクトにするために，ザイリンクスが独自に実装したもの[(1)]．