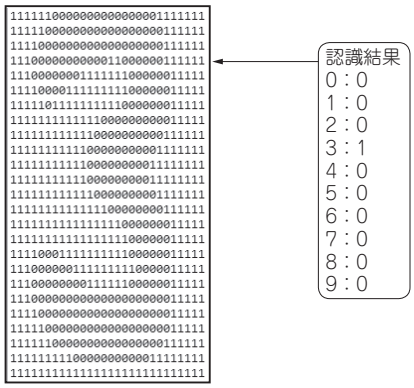
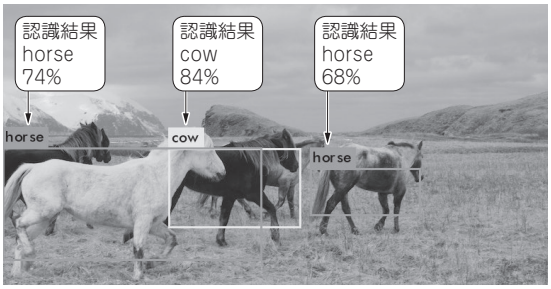
FPGA人工知能のポテンシャルを探る

第6回　バイナリじゃない量子化ニューラル・ネットワークQNNを試す

鈴木 量三朗



（a）1ビットBNN



（b）数ビットのQNN

図1　数ビット使う量子化ニューラル・ネットワークQNNは1ビットのBNNより精度を高めつつ使用するリソースも抑えめにできる

今回から，ザイリンクスの量子化ニューラル・ネットワーク（QNN：Quantized Neural Network）を取り上げます．

量子化ニューラル・ネットワークQNNの特徴

前回まで扱ってきたBNN（Binarized Neural Networks）と今回から扱うQNNでできることの違いを**図1**に示します．

● BNNは高速だったが精度に課題あり

BNNは2値化により高速化を図っていました．

FPGAでは浮動小数点を高速に扱うことに幾つも壁があります．2値化という考え方を導入することで，XNOR計算を並列に実行できるようになり，FPGAでの高速化を期待することができます．

その一方で精度を犠牲にすることになります．浮動小数点数を使用したネットワークと比べると適用可能な分野が限られてしまいます．

● BNNより高い精度をFPGA向きの演算で実現できる

量子化という技術を用いると計算精度を保ちつつ，使用されるリソースを抑えることができます．一般的なニューラル・ネットワーク用のエンジンでは8ビット程度で演算するようですが，ザイリンクスが用意するPYNQ上でのQNNは，FPGAに特化した形で実装されており簡易的な量子化を採用しています．

試してみる

具体的に実行するネットワークは，学習済みのYOLO（You Only Look Once）による物体検出（Object Detection）のプログラムです．

● サンプルはザイリンクスQNN

QNN-MO-PYNQ[1]は，ザイリンクスが呼ぶところの量子化のニューラル・ネットワークであるQNNを利用したニューラル・ネットワークです．

一般的な量子化は，8ビットをある範囲にマッピングさせることで実現することが多いようですが，ザイリンクスのQNNは，BNNの延長にあります．また，全ての処理をFPGAで行っているわけではなく，PythonとNumPyも利用しています．