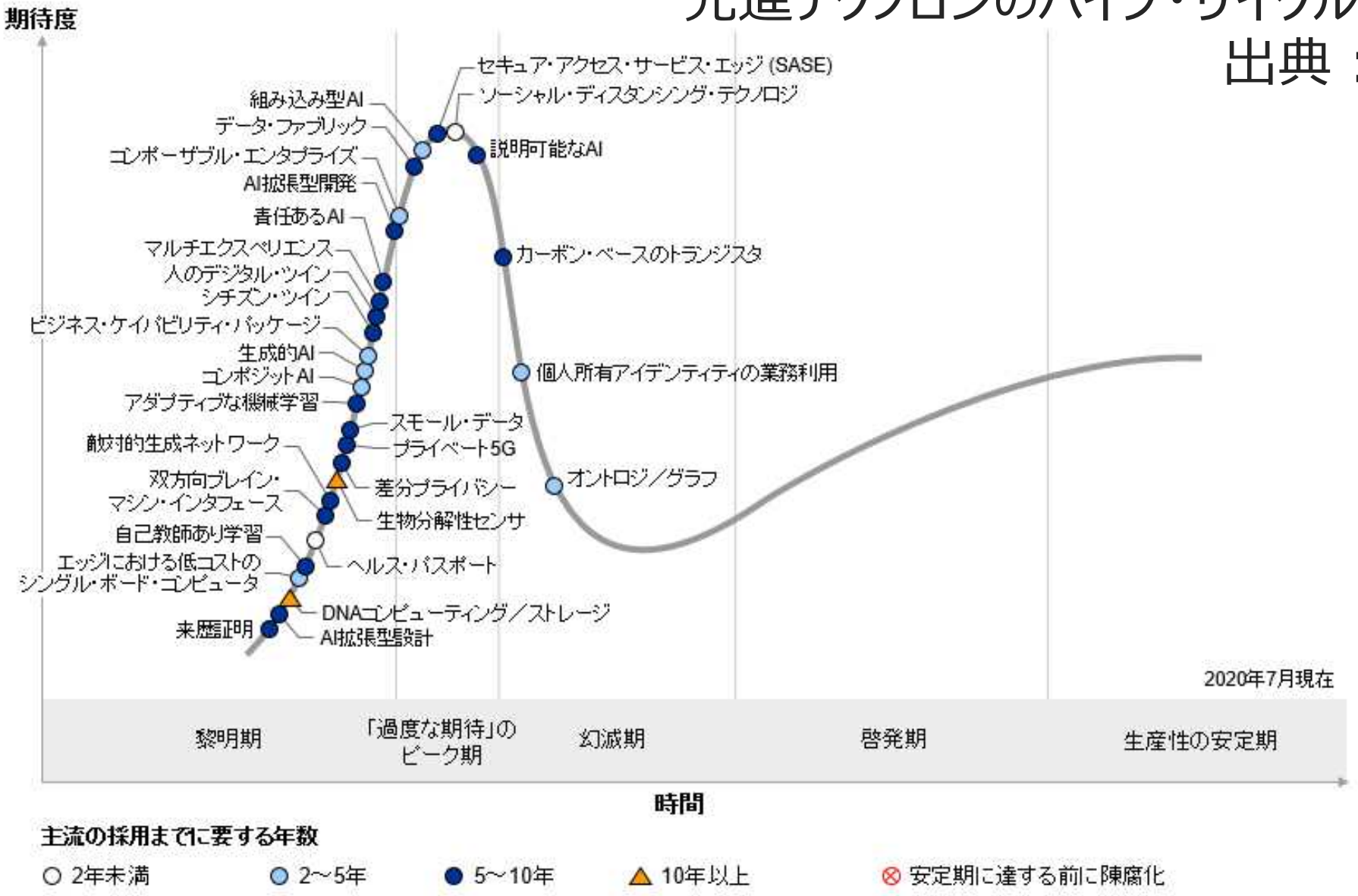
モバイルデバイスで AIモデルを動作させるための 量子化技術

熊本大学　大学院先端科学研究部
助教　木山 真人

2021年8月5日

ありとあらゆるところにAI

先進テクノロジのハイプ・サイクル：2020年
出典：ガートナー



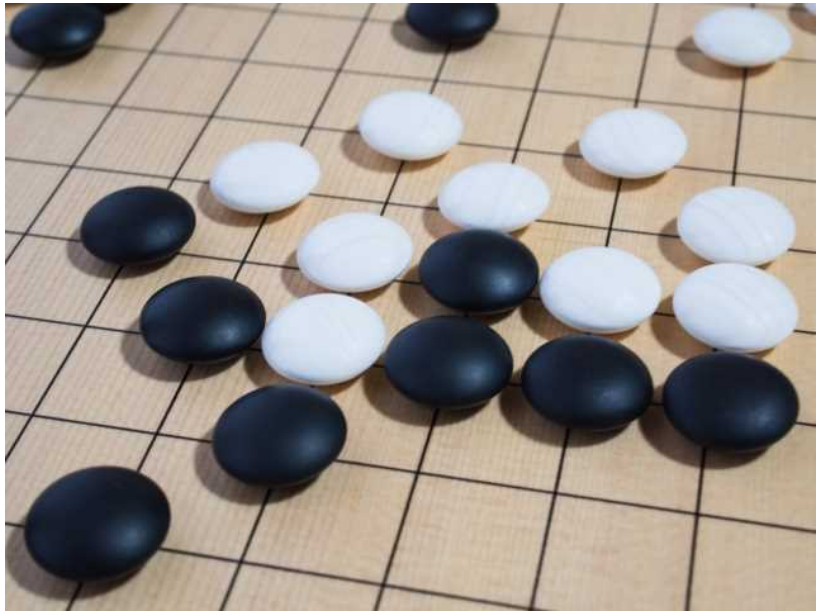
SoTAには大量のリソースが必要

GTP-3

- 最強の自然言語処理AI
- 学習に460万ドル・355年（GPUx1）

AlphaGo

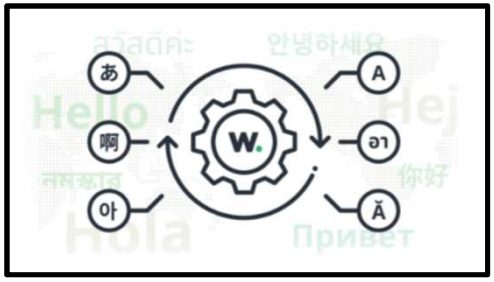
- 人間のプロ囲碁棋士に買ったAI
- 学習に3500万ドル



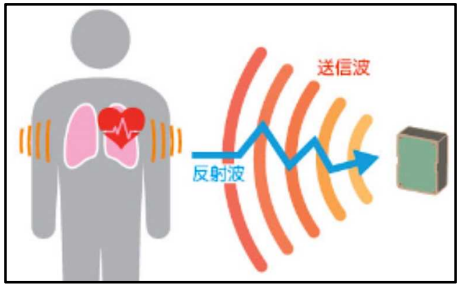
エッジAI

画像認識

音声処理

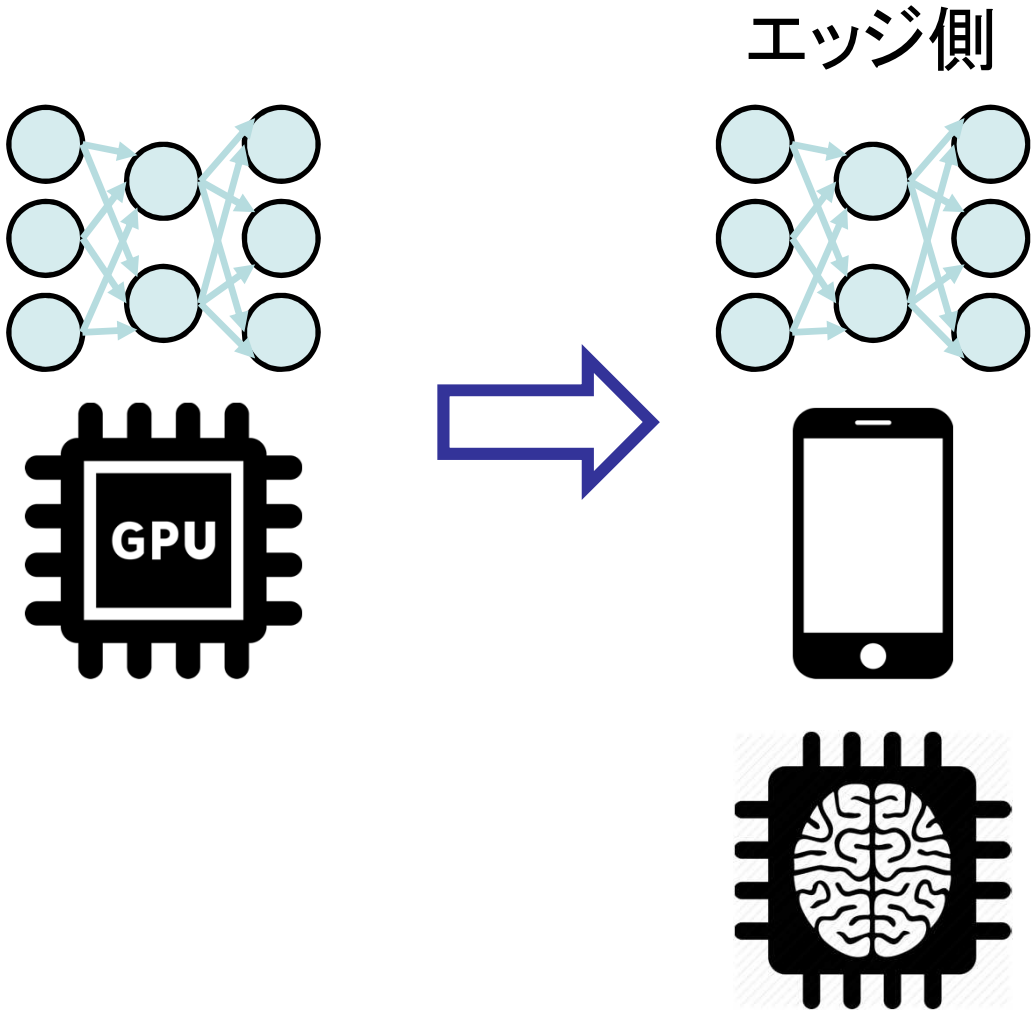


機械翻訳



生体信号処理

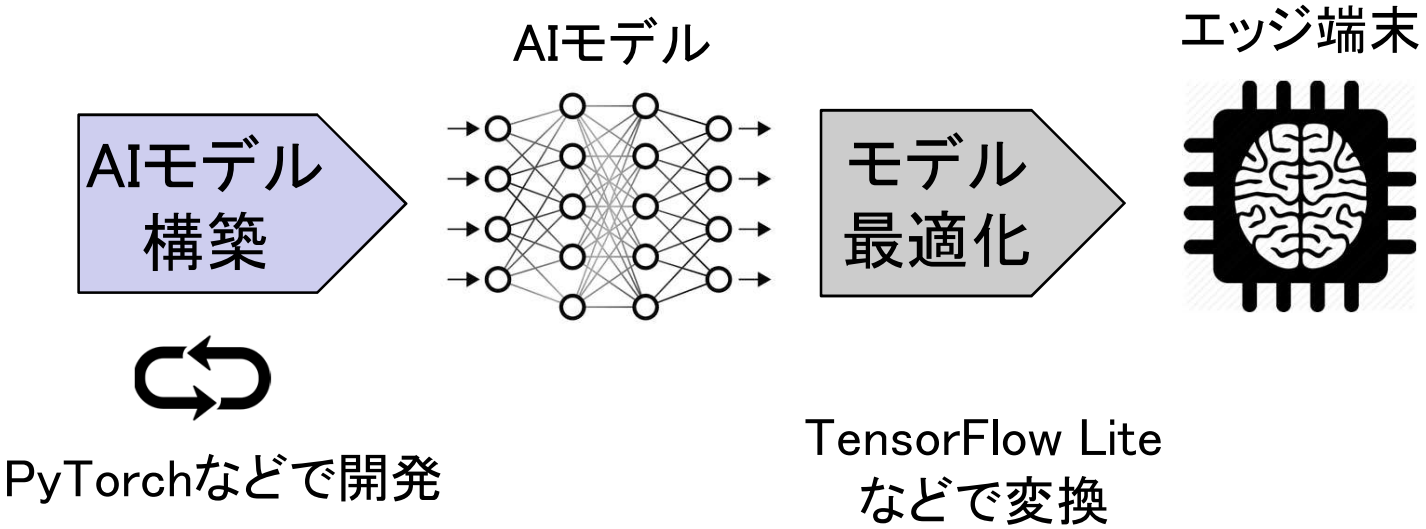
エッジAIでの問題点



様々な課題

- 消費電力は？
- どのくらいの精度？
- モデルは？ResNet18？

従来技術

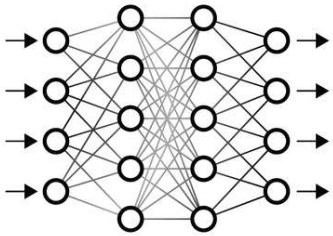


エッジ側での性能が出るまで

新技術の特徴と従来技術との比較

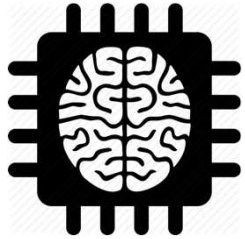
AIモデル構築

AIモデル



モデル最適化

エッジ端末



PyTorchなどで開発

DNNコンパイラで
より最適化

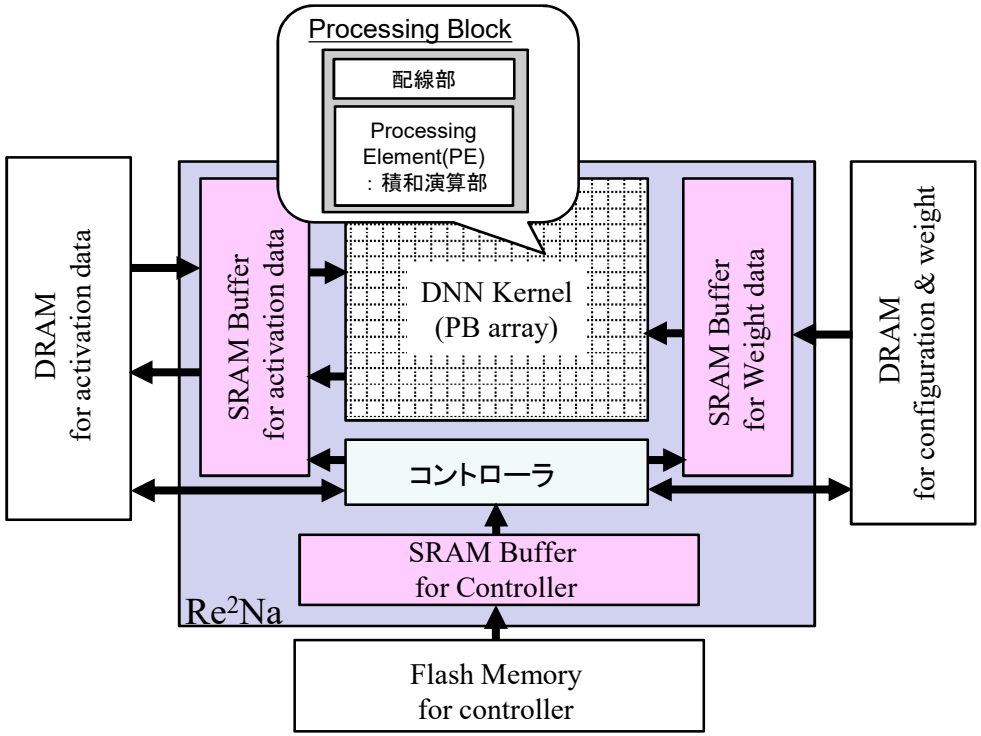
Re²Na
再構成可能なAIチップ

PyParchで量子化した結果
をエミュレーション

- AIプラットホーム
- ソフトからハードまで
- 素早く市場へ

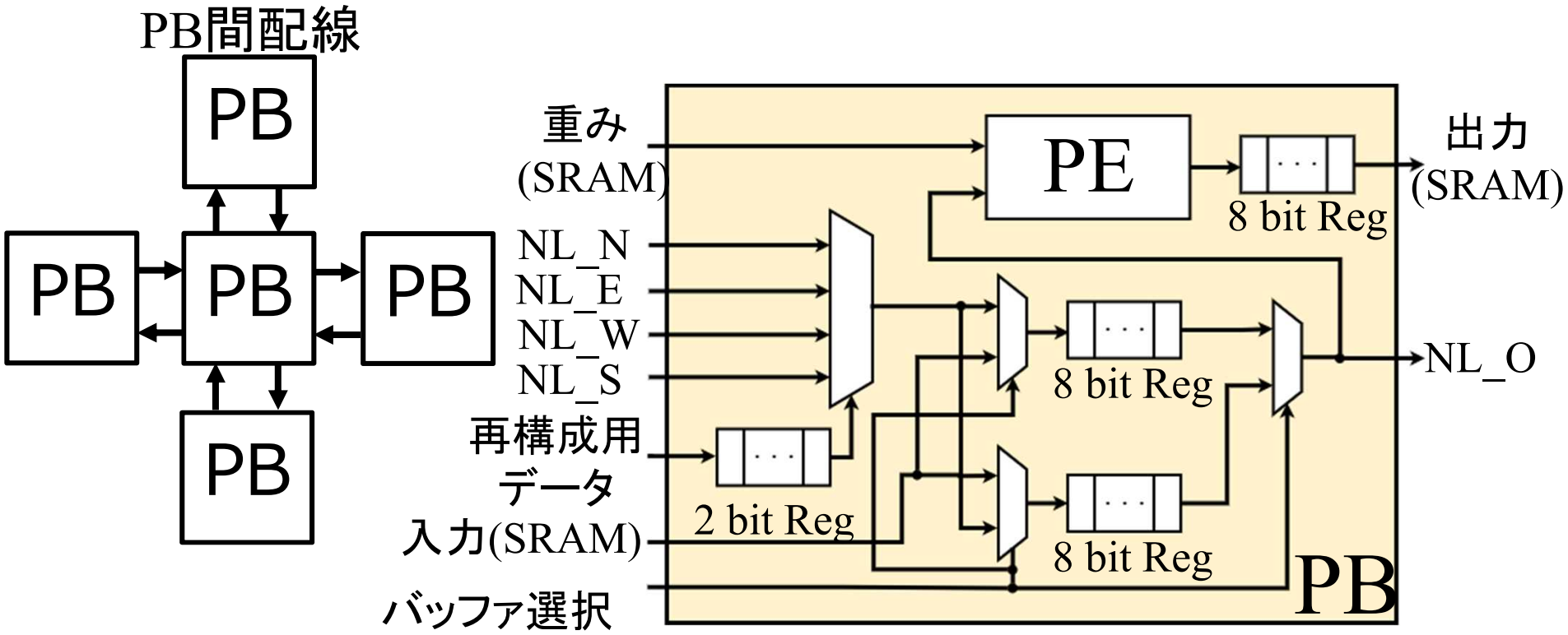
Re²Naのブロック図

- 構成
  - 64x64の積和演算アレイ＋1.5 MB SRAM
    - SRAMに収まらないデータは外部DRAMを使用
  - 畳込み層/全結合層はPB間接続を再構成することで実現



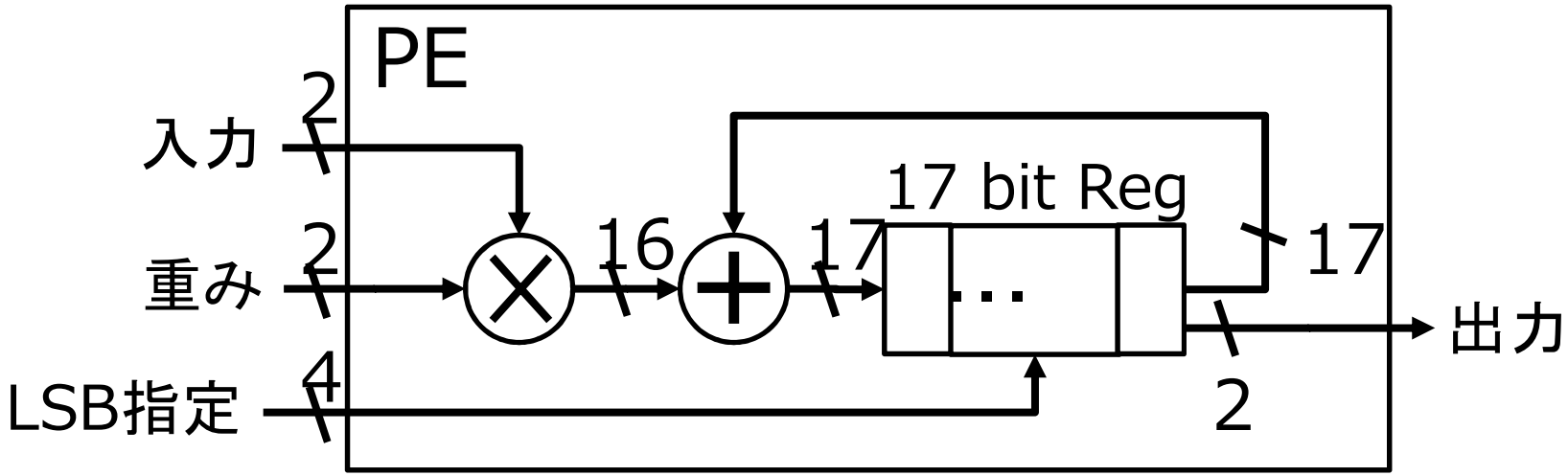
PBの構成

- PE：積和演算とReLU処理
- 配線部：入力，重み，出力を制御
  - 再構成時にPB間配線を切り替え
  - 2つのレジスタで入力をダブルバッファリング



PEの構成

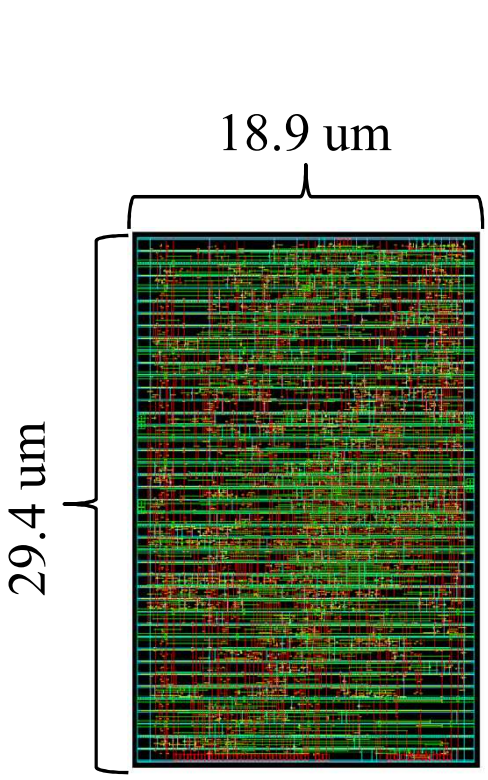
- 演算には5サイクル必要
- MAC演算用に17bitのレジスタを使用
- 演算結果出力時，8 bitに量子化
  - 指定したLSBから8 bitを切り出す
  - 同時にReLUを適用



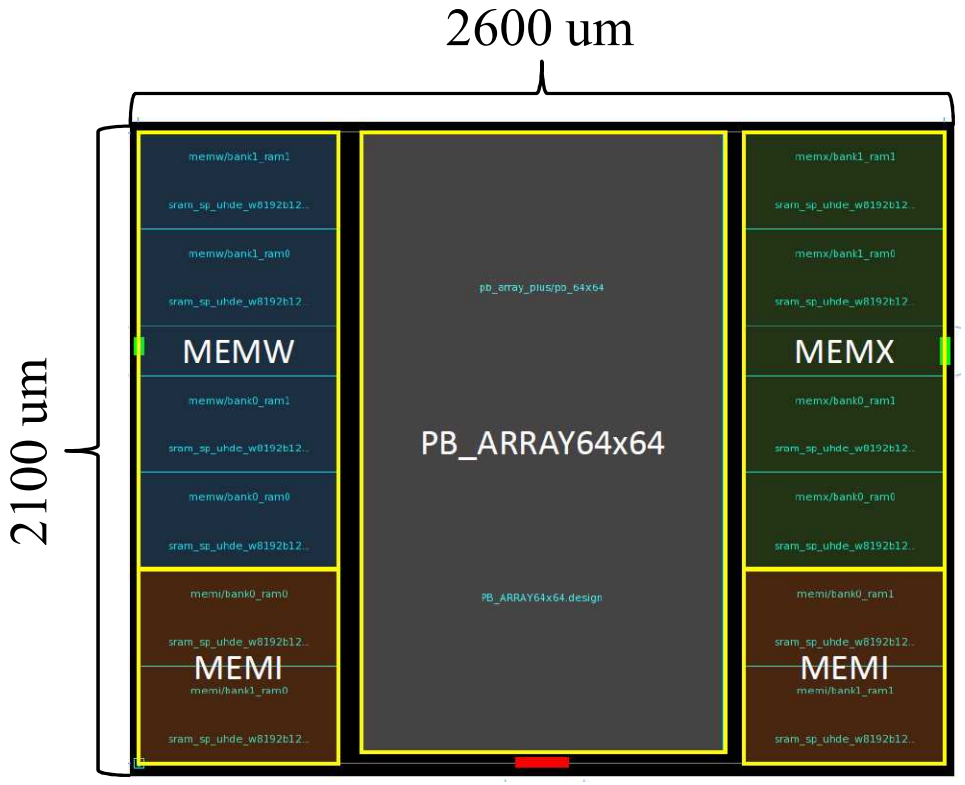
LSB(Least Significant Bit)：最下位ビット

IP設計

- TSMC 22nm ULL(Ultra Low Leak)
  - 電源電圧0.8v，8層メタル



PB



IP全体

軽量化技術

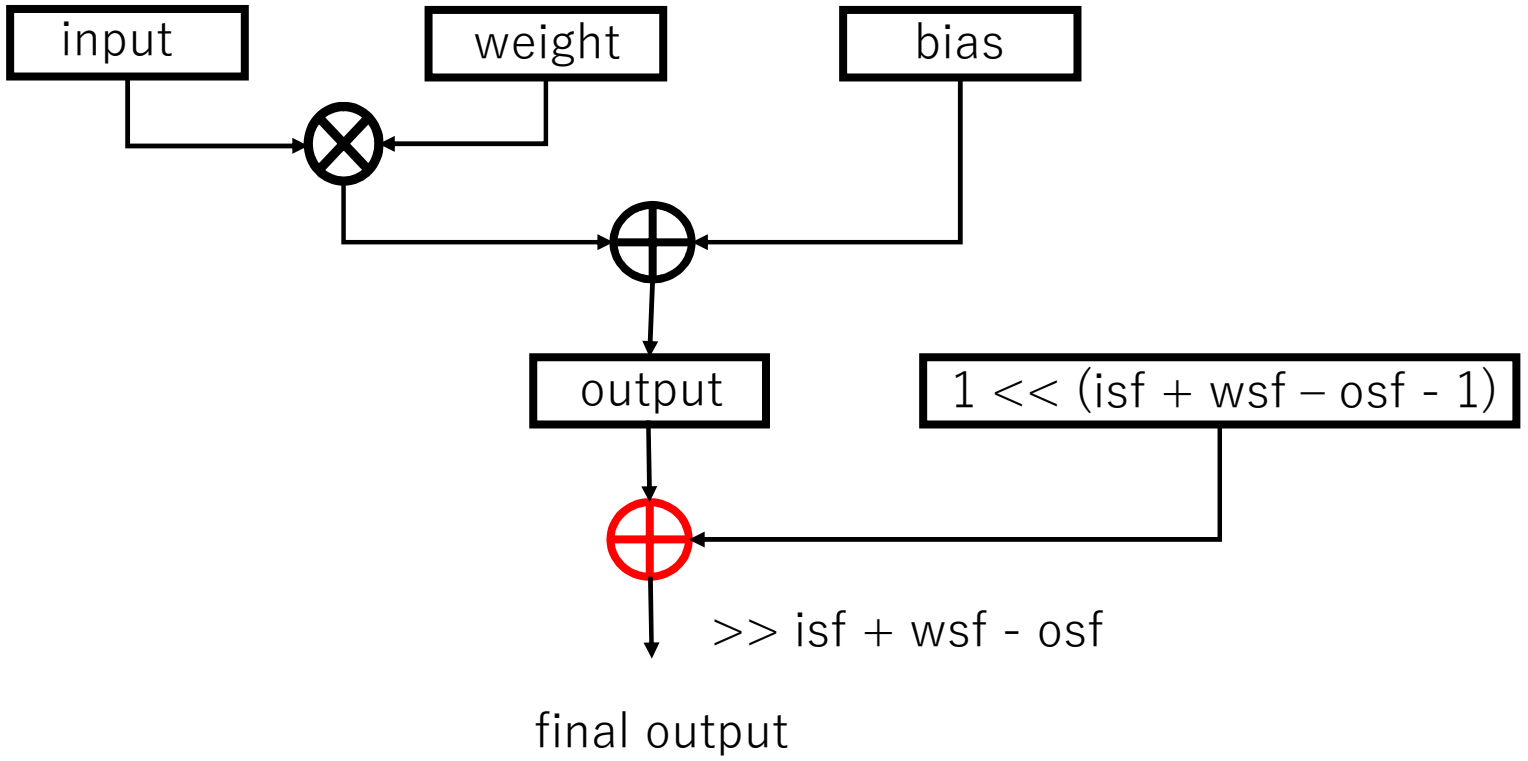
- 枝刈り
  - 重要度の低いニューロンを削減
- 蒸留
  - 元々のモデルを模倣するモデルを作成
- 量子化
  - 数値データの精度を下げる

PyParch

Accuracyを正確に調査できるフレームワーク

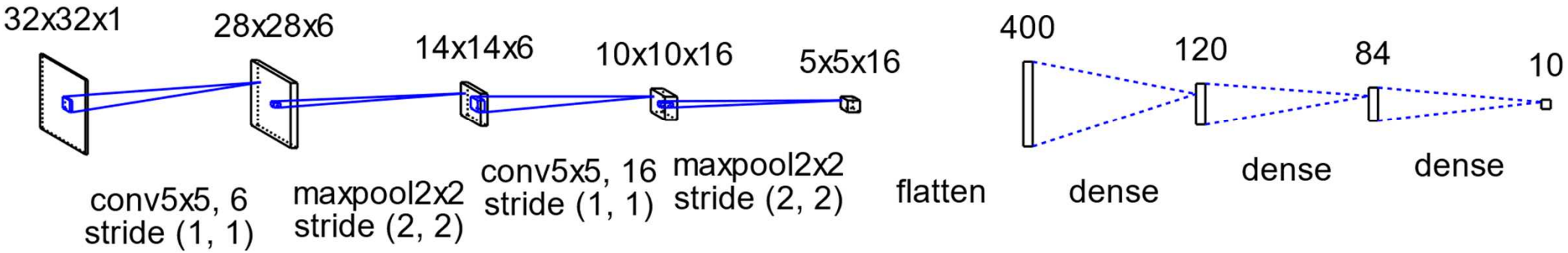
- PyTorch上に作られたフレームワーク
- PyTorchと同じように使え、機能追加が容易
- 推論はすべて整数(int64)で行う
  - **正確なエミュレーションが可能**
- クラスを再実装(PyTorchでは整数を計算に使えないため)
  - QConv2d、QLinear、QReLU
  - QMaxPool2dP

ハードウェアでの演算方法

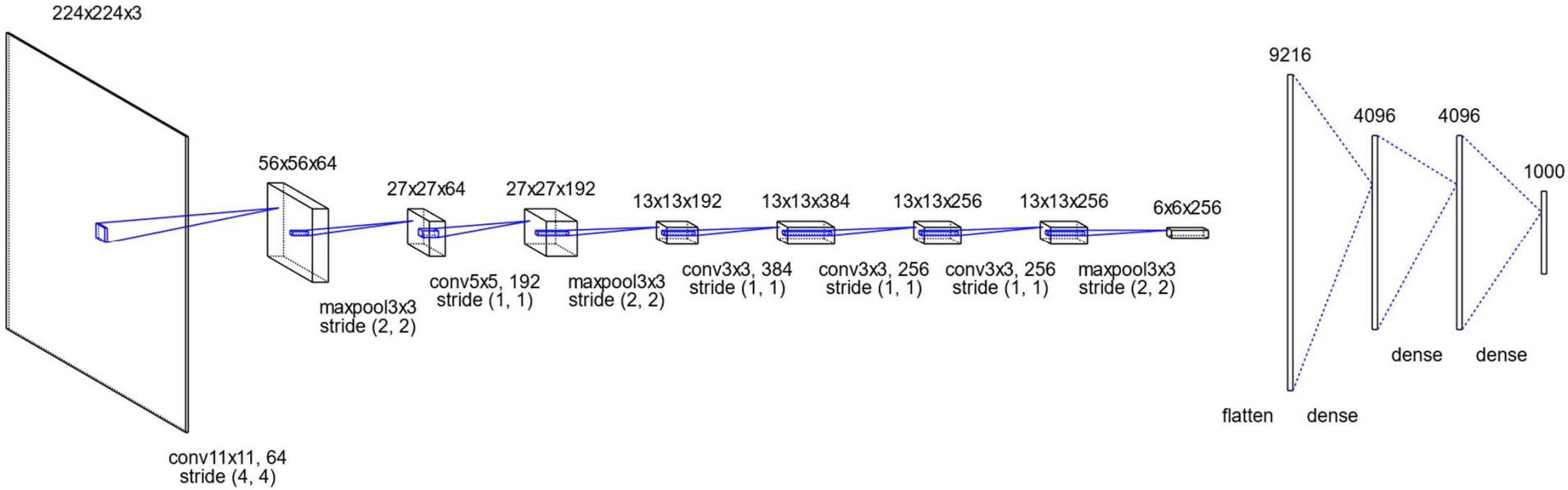


動作確認で使ったモデル

LeNet-5(MNIST)



AlexNet(ImageNet)



Accuracyの変化

LeNet-5

Bit width	Quantized float	Fixed point
8	0.9923	0.9923
7	0.9925	0.9925
6	0.9915	0.9915
5	0.9902	0.9902
4	0.9743	0.9743
3	0.8775	0.8775

AlexNet

Bit width: 8

Accuracy

Top1:

Q:0.55768

F: 0.5204

Top5:

Q:0.78566

F: 0.7648

特別な機能を組み込む

必要なCarry bit数の調査

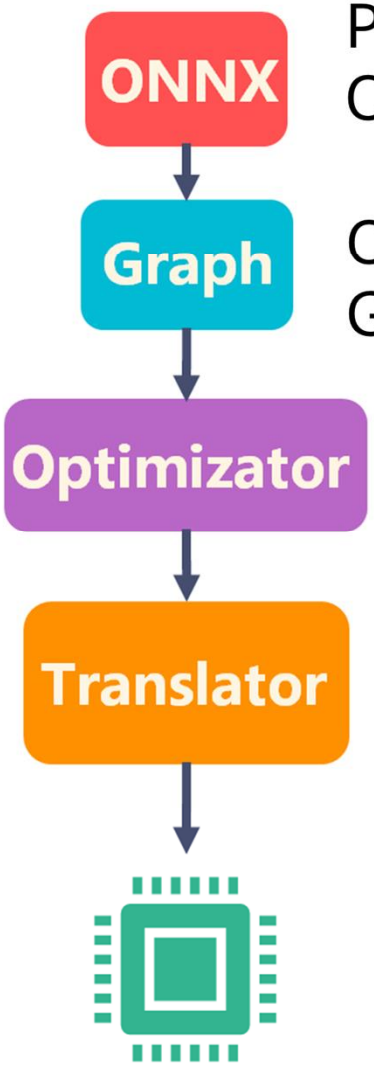
Layer	Carry bit
conv1	1
conv2	2
fc1	3
fc2	1
fc3	1

必要な$carry\ bit$の数
$\log_2$ addの数

演算方法の効果

Bit width	Add 0.5	None
8	0.9923	0.9918
7	0.9925	0.9918
6	0.9915	0.9905
5	0.9902	0.9647
4	0.9743	0.1851
3	0.8755	0.1009
2	0.2191	0.1009

DNNコンパイラ（in Rust）



Pytorchで学習したモデルを
ONNX形式、量子化モデル用に独自形式(tane)

ONNX形式からモデルの構造や重みを抽出し
Graphとして保存

Graphから不要なノードを削除、融合

各形式で実行(ndarray, opencl)

Pytorchで2.43秒が1.93秒へ

想定される用途

- 最新AIを即座にエッジ側で実行
- 事前に精度見積もりができる

実用化に向けた課題

- ハードウェアはIP設計まで終わっているので、あとは実際にチップ作成
- ソフトウェアの基本部分は作成済み。簡単なモデルは動作するがより複雑なモデルに対応する必要がある。

企業への期待

- ハードからソフトまで開発しているところが強み
- エッジ側で具体的なアプリケーションの実行を考えている企業との共同研究を希望

本技術に関する知的財産権

- 発明の名称：演算装置、及び演算方法
- 出願番号：特願2020-193718
- 出願人：熊本大学
- 発明者：木山真人、尼崎太樹、飯田全広

本技術に関する知的財産権

- 発明の名称 ：活性化関数演算方法、および活性化関数演算処理装置
- 出願番号 ：特願2020-167815
- 出願人 ：熊本大学
- 発明者 ：尼崎太樹、上村斗真、中原康宏、飯田全広

本技術に関する知的財産権

- 発明の名称：ニューラルネットワークの回路及び
  ニューラルネットワーク演算方法
- 出願番号　：特願2019-196326
- 出願人　：熊本大学
- 発明者　：尼崎太樹、飯田全広、中原康宏、
  千竈純太郎

お問い合わせ先

熊本大学
熊本創生推進機構・イノベーション推進部門・
リサーチ・アドミニストレーター・和田　翼

ＴＥＬ　096-342-3247
ＦＡＸ　096-342-3300
e-mail　liaison@jimu.kumamoto-u.ac.jp