

Posit 形式の CNN モデルにおける 精度評価

橋本大樹^{1, a)} 木山真人¹ 尼崎太樹¹ 飯田全広¹

概要： AI エッジコンピューティングにおいて、CNN（畳み込みニューラルネットワーク）の計算量やハードウェアリソースは、スマートフォンなどエッジデバイスの制約に反する大きさであることが多い。エッジデバイスで CNN を処理する手法の一つである量子化は、ネットワークモデルのパラメータを低精度の固定小数点へ変換しデータサイズを削減することにより、エッジデバイスで CNN を有効に処理することができる一方、認識精度が量子化前から低下してしまうというデメリットがある。新しい数値表現である Posit は、CNN の重みのような非常に小さい値の表現に適した特徴をもつ。本研究では、低精度表現への量子化による推論精度の低下を緩和するために、低精度 Posit を使用した CNN 推論を実装することで、Posit 形式によるパラメータ表現が推論精度に与える影響を評価する。32bit の浮動小数点で学習したモデルを 8bit の Posit に量子化して推論精度を測定した結果、従来の主流であった浮動小数点 8bit への量子化と比較して、推論精度の低下を小さく抑えることが可能であると分かった。

キーワード： 機械学習, ディープラーニング, 畳み込みニューラルネットワーク, 量子化, Posit

Investigating the Accuracy of CNN Models in Posit Format

DAIKI HASHIMOTO^{1, a)} MASATO KIYAMA¹
MOTOKI AMAGASAKI¹ MASAHIRO IIDA¹

Abstract: In AI edge computing, the amount of calculation and hardware resources of CNN (convolutional neural network) are often large, which is contrary to the restrictions of edge devices such as smartphones. Quantization, which is one of the methods for processing CNNs on edge devices, can effectively process CNNs on edge devices by converting network model parameters to low-precision fixed-point numbers and reducing the data size. On the other hand, there is a demerit that the recognition accuracy is lower than that before quantization. Posit, a new numerical representation, has features suitable for representing very small values such as CNN weights. In this study, we evaluate the effect of parameter representation in Posit format on inference accuracy by implementing CNN inference using low-precision Posit in order to mitigate the decrease in inference accuracy due to quantization to low-precision representation. .. As a result of measuring the inference accuracy by quantizing the model learned in 32-bit floating point into 8-bit Posit, it is possible to suppress the decrease in inference accuracy to a small extent compared to the conventional mainstream quantization to floating-point 8 bits. It turned out to be possible.

Keywords: Machine Learning , Deep Learning, Convolutional Neural Network Quantization , Posit number system

1. はじめに

近年、AI エッジコンピューティングと呼ばれる技術が注目されている。ディープラーニングなどを始めとする AI 技術は画像認識分野など様々な領域で研究されており、CNN (Convolutional Neural Network, 畳み込みニューラルネットワーク) は次々と新しいネットワークモデルが発表されている。エッジコンピューティングとは、ユーザ側に近い場所にある端末 (エッジ端末) で得た入力情報を、遠隔地のクラウドサーバに転送して処理するのではなく、エッジ端末内で処理を行うことによって、通信にかかる電力を節約したり遅延を低減し高いリアルタイム性を獲得することができる技術である。AI エッジコンピューティングとは、エッジ端末で得た情報を用いて AI 処理を行うことを言う。しかし CNN などの AI 技術は、計算にかかるコストが高く、通常 GPU などの計算能力が高いデバイスで処理される事が多い。

スマートフォンなどの電力性能やハードウェアリソースに制約があるデバイスで CNN を処理しようとする場合、CNN モデルに対して量子化という処理を施すことがある。量子化はデータのビット数をエッジデバイスで処理可能なサイズにまで削減することを言い、量子化を行う際には、量子化を施す前からいかに性能低下を防ぐかが重要となる。

CNN モデルに量子化を施す場合、クラウド側で浮動小数点 32bit で学習しパラメータ調整を行ったモデルを、低精度の固定小数点などハードウェアリソースが小さい数値表現に変換することが多い。しかし低精度固定小数点への変換は、CNN の推論精度を損なうというデメリットをもつ。

本研究では、低精度表現への量子化による推論精度の低下という問題に対し、Posit と呼ばれる表現を用いた CNN 推論の実装を行った。主な実装内容として、浮動小数点形式かた Posit 形式への量子化と、Posit 形式の CNN 推論処理の実装を行った。

本論文では、第 2 章で Posit の数値表現としての特徴について述べ、第 3 章では本研究で実装した内容について説明し、第 4 章で評価結果について述べ第 5 章で考察を行った後、第 6 章で本論文をまとめる。

2. Posit

2.1 Posit のデータ構成

Posit は、J. Gustafson 氏が 2019 年に発表した新しい浮動小数点の表現形式である。Posit は、図 1 に示されるように、sign, regime, exponent, fraction の 4 種類のビットフィールドによって構成される。IEEE754 の浮動小数点との相違点として、数値のスケールリングをする指数部が、regime, exp の 2 種類のビットフィールドによって表現さ

れる点、また各ビットフィールドの長さは可変長であるという点が挙げられる。

Posit はユーザによって、ビット長 N と exp ビットサイズ es によって $\text{Posit} = \langle N, es \rangle$ として定義される。regime ビットは、Posit 特有のビットで 2 進数の並びによって特定の値を表現する。regime が表現する値は表 1 に示す通り、連続する 0 または 1 の個数によって決定される。regime ビットの終端は、連続する値の反転で決定される。exp ビットは 2 進数値 e を表し、 2^e を表す。fraction ビットは、 N からその他三種類のビットを除いた残りが割り当てられるため、最大 $N-es-3$ ビットである。fraction ビットは、常に値が 1 の隠しビットをもつ。

以上より Posit の値は、式 (2.1) によって表現される。

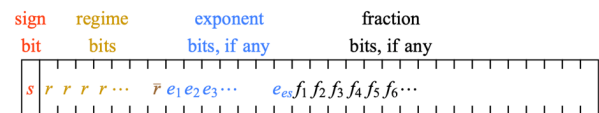


図 1 Posit のデータ構造

Binary	...	0001	001	01	10	110	1110	11110	...
regime	...	-3	-2	-1	0	1	2	3	...

表 1 regime ビットの示す値

$$(-1)^{\text{sign}} \times (2^{2^{\text{es}}})^{\text{regime}} \times 1.f \quad (2.1)$$

2.2 ダイナミックレンジ

Posit で表現された値の、パラメータ別に値の分布を表現したものを表 2 に示す。Posit の特徴として、パラメータ $\langle N, es \rangle$ によって表現する値のダイナミックレンジが変化するという特徴がある。データのビット数が同じである場合、 es サイズが大きい方が、絶対値の大きい数値まで表現することができるため、ダイナミックレンジが大きくなる。

Wordsize	$es = 1$	$es = 2$	$es = 3$
Posit8	$1.5 \times 10^{-2} \sim 64$	$2.4 \times 10^{-4} \sim 4.9 \times 10^3$	$5.8 \times 10^{-8} \sim 1.6 \times 10^7$
Posit12	$1.0 \times 10^{-3} \sim 1.0 \times 10^3$	$1.9 \times 10^{-6} \sim 2.0 \times 10^6$	$3.6 \times 10^{-12} \sim 4.0 \times 10^{12}$
Posit16	$6.1 \times 10^{-5} \sim 1.6 \times 10^4$	$4.0 \times 10^{-9} \sim 3.0 \times 10^8$	$1.6 \times 10^{-17} \sim 9.0 \times 10^{16}$
Posit32	$9.3 \times 10^{-10} \sim 1.0 \times 10^9$	$8.7 \times 10^{-19} \sim 1.2 \times 10^{18}$	$7.5 \times 10^{-37} \sim 1.3 \times 10^{36}$
Float32	$1.0 \times 10^{-38} \sim 3.0 \times 10^{37}$		

表 2 ダイナミックレンジの比較

1 熊本大学

Kumamoto University, chuou-ku, Kumamoto 860-0862, Japan

a) d_hashimoto@st.cs.kumamoto-u.ac.jp

Posit で表現された値の、10 進数との誤差の分布を表現したものを図 2 に示す。データに占める fraction ビットの割合が大きくなるほど値の大部分はレンジの中心、すなわち 0 近傍に集中するという特性上、0 近傍の数値は細かい刻みで表現が可能である。また、0 近傍の値は分布の中でも最も 10 進数からの変換誤差が小さい。しかしレンジの端の絶対値が大きい値は、データの大部分が regime ビットになり、数値の刻みが 2 の倍数またはべき乗に依存するため、表現の精度は低くなる。

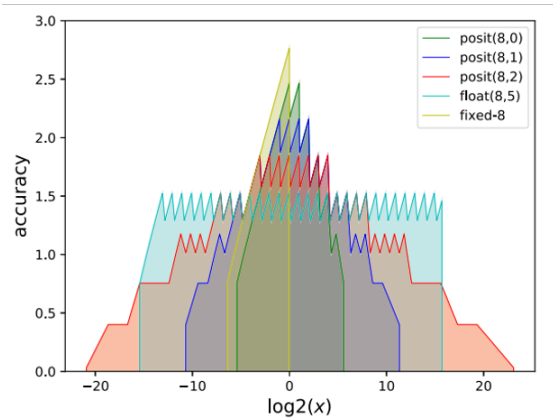


図 2 Posit の 10 進数変換精度

2.3 Posit と機械学習

Posit で表現される値は、0 近傍になるほど数値間のハミング距離は小さくなり、細かい刻みで絶対値の小さな小数も表現可能である。また、値が小さくなるほど、基となった 10 進数との変換誤差が小さいという特徴を持つ。

機械学習のトレンドである、畳み込みニューラルネットワーク（以下、CNN（Convolutional Neural Network））の火付け役となった、AlexNet の畳み込み層第 3 層の重みの分布と、 $\langle N, es \rangle = \langle 8, 1 \rangle$ の Posit の値の分布を示したものを図 3 に示す。

CNN の重みの分布と Posit の値の分布は大部分が一致しており、また、CNN の推論精度は重みの表現における精度にある程度依存するため、研究者の間では CNN の重み表現として Posit を使用する試みが行われている。

図 3 にも見られるように、ニューラルネットワークの重みの分布は 0 を中心としてその周辺の値が最も密である。したがって、これらの値をいかに正確に表現するかが、ネットワークモデルの認識精度に大きく影響する。一般的には浮動小数点形式の 32 ビットの情報を用いることで、非常に高い認識精度を達成することが可能であるが、スマートフォンをはじめとするエッジデバイスでは 32 ビットのデータ量は処理するには大きすぎるという問題がある。しかしデータの量子化を行うことによってデータ量を削減しようとすると、ニューラルネットワークの重みの中でも最も重要

な 0 近傍の値における表現精度の低下を招き、結果としてモデルの認識精度を低下させることにつながる。

Posit は同じビット数の浮動小数点形式と比較して、表現できる値のレンジの広さでは劣るものの、0 近傍での値においては精度において優位であるという特徴を持つ。ゆえに、認識精度をある程度担保しつつ、データを量子化する先の数値表現として注目されている。

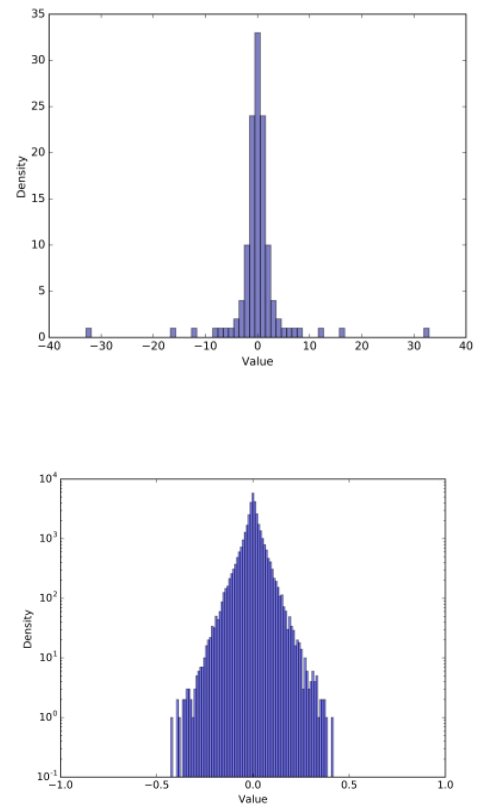


図 3 Posit（上）と AlexNet 第 3 層の重み（下）の分布

3 実装

本研究では、Posit の数値表現としての特徴に注目し、CNN モデルに対してデータ形式を Posit に変換して実装を行った。関連研究との差別化として、同一の実装を複数の CNN モデルに対して行うことにより、異なる特徴をもつモデルに対して量子化を行う場合それぞれどのような影響を及ぼすのかを評価した。特に、使用した CNN モデルについては、エッジでの Posit 実装を想定し、パラメータ数がエッジ向けに最適化されている軽量なモデルを選択した。

本章では、学習済みモデルに対して 8 ビット量子化を行い、いくつかの CNN モデルを Posit 形式に変換して推論を行う過程について述べる。

3.1 Posit 形式への量子化

本節では、Float 形式の CNN モデルの Posit 形式への量子化について述べる。CNN モデルの学習には、機械学習用フレームワークの Pytorch を使用した。Pytorch は Float から Posit への量子化へ対応していないため、Posit への量子化方法の実装が必要である。本研究では、量子化のために Github で公開されているライブラリである「QPytorch」を使用した。QPytorch は、Pytorch の量子化用ライブラリであり、Float32 から 8 ビットの Posit (以下、Posit8) など、データ幅を削減した形式への相互変換が可能である。QPytorch はいくつかの Posit パラメータにのみ対応しており、8bit の Posit は、es は 0, 1, 2 の 3 通りが対応している。

3.2 Posit 形式の CNN 処理

本節では、Posit 形式に量子化を施した CNN の処理について述べる。

Posit 形式での処理を行うためには、CNN の層の入出力を Posit 形式に変換しなければならない。今回推論に使用した CNN モデルは、Pytorch で 32 ビット Float (以下、Float32) を用いて学習済みのモデルである。学習済みモデルに含まれる重みは学習過程で調整された Float32 の値であるため、CNN の入力層で 8 ビットの Posit (以下、Posit8) に変換した。

CNN の処理の大部分は、入力と重みと呼ばれるパラメータの積和演算である。Pytorch は Posit 形式で表現された値の積和演算に対応していないため、本研究では、Posit 形式の値を演算時に Float 形式に再変換して演算を処理することで推論を実装した。

図 4 に、Float32 と Posit8 で表現された値のダイナミックレンジと、変換元となった 10 進数との表現誤差をグラフにしたものを示す。

図 4 より、Float32 は Posit8 に比べ遥かに広いダイナミックレンジがあり、また、すべての領域において Posit8 より高い精度で数値を表現することができる。Float32 は Posit8 に比べダイナミックレンジが広いと、Float32 から Posit8 に量子化を施すと、多くのビットに含まれる情報が削減される。Posit8 で表現された値を Float32 に再変換する場合、表現能力が低い形式から高い形式に変換するため、Float32 への変換による誤差は小さいと考えられる。よって、Posit8 の積和演算および結果は、Float32 での積和演算でシミュレートすることが可能であると考えられる。

Float32 の値で積和演算した結果は、再度 Posit8 に変換され、層の出力となり次層における入力として再利用される。以上により、Posit8 形式 CNN モデルを用いた推論処理を行うことができる。

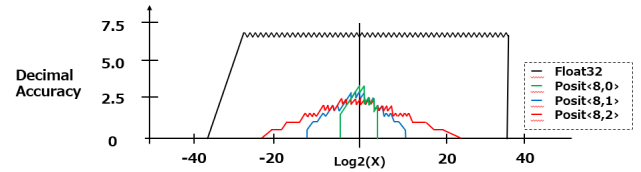


図 4 Float32 と Posit8 の比較

4 評価

4.1 評価の概要

本節では、Posit8 形式での CNN を用いて推論処理を行い、精度評価を行った概要について述べる。

評価に使用した CNN モデルは、AlexNet, VGG16, MobileNetV2, ResNet50, ResNet152 を選択した。それぞれのモデルは、Pytorch 環境で構築したものを使用した。学習および推論のデータセットには ImageNet の画像 5 万枚を使用し、1000 分類を行った時の精度で評価を行った。Posit のパラメータは、ビット長 N は 8 ビットで固定し、exp ビットのサイズを表す es の値を 0~2 の範囲で変化させた。

4.2 評価結果

Posit8 形式に量子化した CNN モデルを用いて量子化を行った結果を表 3 と表 4 に示す。

Posit8 を用いた量子化は、es の値によって推論精度が前後したが、いずれの CNN モデルにおいても、最も精度が落ちなかったのは es = 1 の場合であった。この傾向は AlexNet のようなモデルに含まれる重みの数が多いものから、ResNet50 のような重みの数が少ないものまで同じであった。

今回推論精度の測定に使用したモデルについて、それぞれの計算層に含まれるパラメータである重みの個数を表 5 に示す。ニューラルネットワークはモデルの層の深さなどの要因によってモデルに含まれる重みの数が決まっており、一般的にこの重みの数が多いほど計算コストが高くなる傾向にある。

表 3 Top-1 の精度

Model	Float32	int8	Posit		
			es = 0	es = 1	es = 2
AlexNet	55.57	50.89	36.71	52.30	52.08
VGG16	71.15	67.66	40.17	66.77	66.22
mobileNetv2	70.00	42.21	28.05	62.33	54.51
ResNet-50	76.15	51.16	49.86	72.81	49.72
ResNet-152	78.31	55.49	0.35	75.55	75.39

表 4 Top-5 の精度

Model	Float32	int8	Posit		
			es = 0	es = 1	es = 2
AlexNet	78.42	75.66	96.83	77.63	77.28
VGG16	89.43	88.51	79.15	89.12	88.75
mobileNetv2	89.29	67.09	54.09	85.43	79.15
ResNet-50	92.87	74.63	75.60	91.70	75.60
ResNet-152	94.96	78.14	12.25	93.31	93.23

表 5 モデルに含まれる重みの数

モデル名	重みの数
AlexNet	61,100,840
VGG16	138,357,544
MobileNetV2	3,504,872
ResNet-50	25,557,032
ResNet-152	60,192,808

それぞれの CNN モデルの計算層に含まれるパラメータである重みの数と処理に用いるデータサイズの積を横軸に、推論精度を縦軸に取り、32 ビットと 8 ビットのそれぞれの場合の推論精度の測定結果をグラフに表したものを、図 5 に示す。グラフ中のシンボルは、四角形がベースラインとなる Float32 のときの精度、星型が Posit8、丸形が固定小数点 8 ビットである。

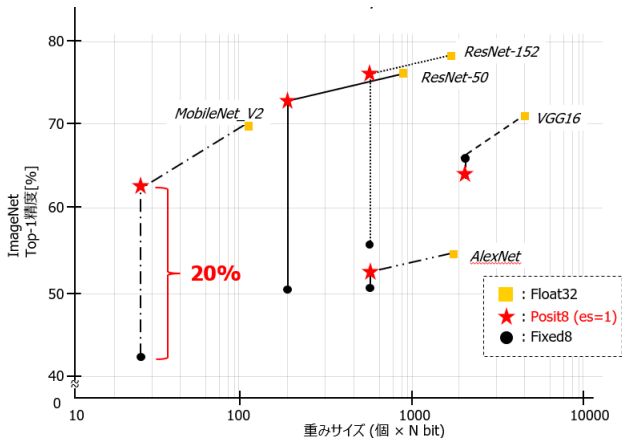


図 5 モデル別の重みサイズと推論精度の比較

5 考察

本節では得られた結果から、Posit 形式に量子化した CNN モデルを用いた推論についての考察を行う。

量子化した 3 種類の Posit8 のうち、もっとも推論精度が高いものは es が 1 の Posit8 (Posit<8, 1>) であった。いずれのモデルにおいても、Posit<8, 0> がもっとも精度が低く、Posit<8, 2> が 2 番目に精度が高い結果となった。Posit<8, 0> は、Fixed8 よりも低い推論精度となった。

また、図 5 のグラフより、Posit8 を用いた推論は、Fixed8 を用いた推論と比較したとき、推論精度の低下をより抑えていると言える。中でも特に MobileNetV2 のような、モデルに含まれるモデルが小さいモデルは Fixed8 に量子化した場合、20%以上の精度低下が見られたのに対し、Posit8 へ量子化した場合精度低下は 5%以下に抑えられた。

これらの理由について考察する。まず、今回量子化に使用した表現のそれぞれのダイナミックレンジを表 5.1 に示す。

表 6 ダイナミックレンジの比較

形式	ダイナミックレンジ
Posit <8, 0>	$1.5 \times 10^{-2} \sim 64$
Fixed 8	$3.9 \times 10^{-3} \sim 128$
Posit <8, 1>	$2.4 \times 10^{-4} \sim 4.1 \times 10^3$
Posit <8, 2>	$5.8 \times 10^{-4} \sim 1.6 \times 10^7$

ダイナミックレンジを比較すると、Posit<8, 0>のレンジは Fixed8 のレンジより狭い。Posit<8, 0>が Fixed8 より精度が下がってしまった原因として、ダイナミックレンジで劣るために表現できる値の桁数で劣ってしまったことが上げられる。

次に、Posit<8, 2>での精度が Posit<8, 1>での精度に劣ってしまった理由について考察する。図 5. 1 に Posit<8, 1>と Posit<8, 2>のダイナミックレンジに対する数値表現の精度を示す。図 5. 1 に見られるように、Posit<8, 1>の方がダイナミックレンジでは劣るものの、0 付近の値については Posit<8, 2>に精度で優れる値域が存在する。図 5. 2 に Posit<8, 1>と Posit<8, 2>の精度の差が大きかったモデルである、ResNet-50 の重みの分布図を示す。

図 6 において黄色のハイライトをしている値域は、Posit<8, 1>が Posit<8, 2>に精度で勝る値域である。図 7 では、同じレンジをハイライトしており、この値域に含まれる重みは、ResNet-50 の全ての重みの 91%に相当する。つまり、Posit<8, 1>は ResNet-50 の重みの 91%を Posit<8, 2>より高い精度で表現できたと言える。CNN の重みの表現精度の高さは推論精度の高さにつながるため、Posit<8, 1>が最も推論精度が高かったと考えることができる。

また、MobileNetV2やResNetではFixed8とPosit8の精度において大きな差が出たが、VGG16のような重みの数が多いモデルでは精度の差は小さかった。CNNの推論精度は重みの表現精度に左右されるため、重みの母数が少ない場合一つの重みの表現精度の差が全体の表現精度に占める影響が大きくなるため、MobileNetV2などでは精度に大きな差が生まれたと考えられる。

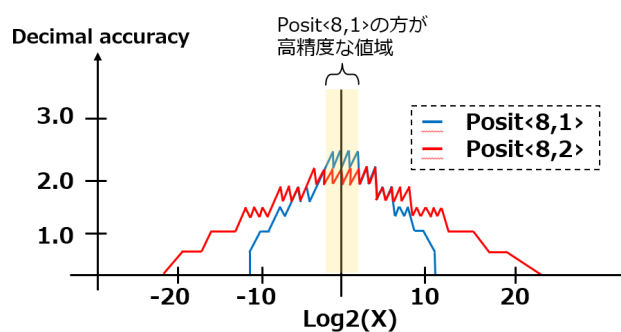


図6 Posit<8,1>とPosit<8,2>の精度の比較

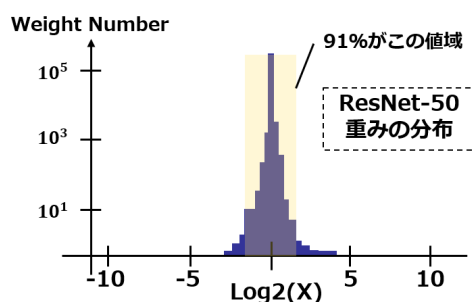


図7 ResNet-50の重みの分布

6 まとめ

Posit8に量子化する場合、Posit<8, 1>はFloat32からの精度低下を10%未満に抑えることが可能であった。低精度固定小数点へCNNモデルを量子化すると推論精度が低下するという問題に対し、Posit<8, 1>は精度低下の低減に有効な表現であることがわかった。

また、Positのパラメータであるesは大きな値を選択すれば、表現できる値のダイナミックレンジが広がるもののCNNの推論においてはesが大きいPositが必ず推論精度で秀でるとは限らないので、Positのビット長とその他の条件に応じて最適なパラメータを探索することが必要である。

7 参考文献

[1] John L. Gustafson, Isaac T. Yonemoto, "Beating Floating Point at its Own Game", *upercomputing Frontiers and Innovations*, 2017, Pages 71-86

[2] Manish Kumar Jaiswal, "PACoGen: A Hardware Posit Arithmetic Core Generator", *ISSRE '11: Proceedings of the 2011 IEEE 22nd International Symposium on Software Reliability Engineering*, November 2011, Pages 120–129

[3] Zachariah Carmichael, Hamed F. Langroudi, Char Khazanov, Jeffrey Lillie, John L. Gustafson, Dhiresha Kudithipudi, "Deep Positron: A Deep Neural Network Using the Posit Number System", *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019, Pages 1421-1426

[4] Jinming Lu, Chao Fang, Mingyang Xu, Jun Lin, Zhongfeng Wang, "Evaluations on Deep Neural Networks Training Using Posit Number System", *IEEE Transactions on Computers*, 1 Feb 2021, vol.70 no.2 Pages 174-187

[5] Jinming Lu, Siyuan Lu, Zhisheng Wang, Chao Fang, Jun Lin, Zhongfeng Wang, Li Du, "Training Deep Neural Networks Using Posit Number System", *2019 32nd IEEE International System-on-Chip Conference (SOCC)*, Singapore, 2019, Pages 62-67

[6] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, December 2012, Pages 1097-1105

[7] Karen Simonyan, Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, 2015, Pages 730-734

[8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, Pages 4510-4520

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, Pages 770-778

[10] 斎藤康毅, "ゼロから作る Deep Learning", 株式会社オライリージャパン, 東京, 2016