2016 年 3 月 24 日 情報システム基盤学専攻中間発表資料

ニューラルネットワークアクセラレータに関する研究

発表者：　1553004 大場 百香
所属：　高性能コンピューティング学講座　本多・三輪研究室
指導教員：　三輪 忍 准教授、本多 弘樹 教授、多田 好克 教授

1　はじめに

画像認識や音声認識の分野で広く使われているニューラルネットワーク (NN) は、近年大規模化に伴い、その計算に要する時間や消費エネルギーが問題となっている。

この問題を解決するために多種のハードウェア・アクセラレータ（DaDianNao[1]、TrueNorth[2]) が開発がされてきたが、これまでのアクセラレータは以下で述べるように限られた機能しかサポートしておらず、用途が非常に限定的であった。

例えば、NN のニューロンモデルは様々な種類があるが、従来アクセラレータは 1 つのニューロンモデルの計算しかサポートしていない。また、NN のシミュレーションでは学習に要する時間も膨大になっているが、ほとんどのアクセラレータは認識処理のみに特化し、学習処理をサポートしていない。

そこで本研究では、ユーザによるニューロンモデルや学習アルゴリズムの自由な変更を許すことによって、任意の NN 計算を高速化かつ低消費電力化するアクセラレータの設計を目的とする。

2　ニューラルネットワーク

ニューラルネットワークは人間の脳内ニューロンの仕組みをモデル化したネットワークである。各ニューロンの出力は一般に、以下のようにして計算される。まず、ニューロン i の膜電位 u_i は、ニューロン j の出力値 N_j とニューロン i, j 間の結合重み w_{ij} と閾値 θ_i を用いて、以下の式で求められる。

$$u_i = \Sigma N_j w_{ij} - \theta_i \tag{1}$$

(1) を活性化関数 f に適用してニューロンの出力（z_i）を得る。

$$z_i = f(u_i) \tag{2}$$

出力値に変換する活性化関数としてシグモイド関数 (3) がよく使われている。

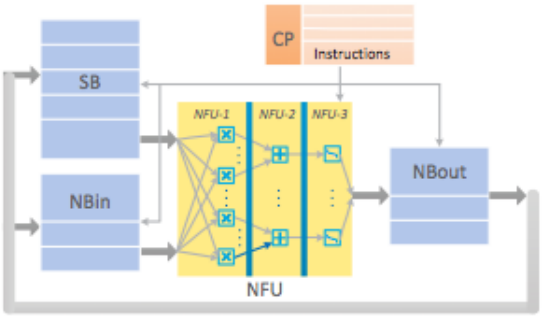
$$f\,(u) = \frac{1}{1 + exp\,(-x)} \tag{3}$$

これらの計算が NN の大規模化に伴って増加するのでアクセラレータを使用する。

3　先行研究

既存アクセラレータの代表例として DaDianNao のアーキテクチャを図 1 に示す。DaDianNao は複数のコアによって構成されており、各コアが複数ニューロンの出力計算を並列に行う。ニューロンの出力計算を行うユニット (NFU) はカスタムロジックによって構成されている。重みは各コアが備えるメモリ（SB）に格納されている。一方、図には記載されていないが、ニューロンの入力値はコア外部の共有メモリに格納されており、必要なデータが制御プロセッサ（CP）によってバッファ（NBin）にロードされる。NFU は SB と NBin の値を用いてニューロンの出力計算を行い、計算結果をバッファ（NBout）へ出力する。DaDianNao は、このように計算に必要なデータを格納するメモリを NFU の近くに配置することで、これらのデータのアクセスレイテンシとアクセスエネルギーを抑制している。

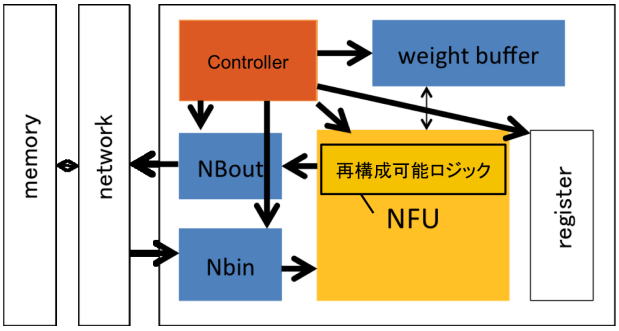
図 1: DaDianNao のアーキテクチャ



4　提案アーキテクチャ

本研究では 1 つのコアに NFU（再構成可能ロジック、積和演算器)、制御部、重みとニューロンのメモリ、レジスタを構成するアクセラレータアーキテクチャ（図 2）を開発する。再構成可能ロジックは、活性化関数などの計算をハードウェアで実現する。積和演算器は多くの NN で必要とされる計算をロジック化し、レジスタはニューロンの状態や膜電位を保持する。DaDianNao と同じような構成にすることでメモリと NFU 間の重みを転送する際にレ

図 2: 提案アーキテクチャ



イテンシとエネルギーを抑制する。

この提案アーキテクチャを NNA-Sim（ニューラルネットワークアクセラレータシミュレーション）でシミュレートする。シミュレーションは、重み・命令・共有メモリに初期データがセットされた状態から始まるのでニューロン値をセットする処理は実行サイクル数としてカウントしない。現在は推論のみ実行可能で学習はできない。

5 予備実験

提案アクセラレータと DaDianNao は、NFU 以外の構成はほとんど同じである。そのため、パラメータを適切に設定すれば、NNA-Sim で DaDianNao の性能シミュレーションを行うことも可能である。そこで、予備実験として、DaDianNao の論文で行われている性能評価実験の再現実験を NNA-Sim を用いて行った。DaDianNao の論文で使用していた評価用 NN を使用し、CPU、GPU、アクセラレータの 3 つの性能（推論時間）をシミュレートし比較した。

layer	N_x	N_y	K_x	K_y	N_i	N_o
CLASS1	-	-	-	-	2560	2560
CLASS2	-	-	-	-	4096	4096

表 1: 評価対象の NN

5.1 評価方法

CPU（E5-2630v2）、GPU（QuadroK2000）上での NN シミュレーションは、DeepLearning のフレームワークである Caffe を用いて行い、推論処理に要した時間を計測する。アクセラレータによる推論時間は NNA-Sim を用いて求める。NNA-Sim ではコア数、共有メモリ・サイズ等のアーキテクチャ・パラメータを設定することができるが、これらの値は DaDianNao の論文を参考に表 5.1 の値とした。

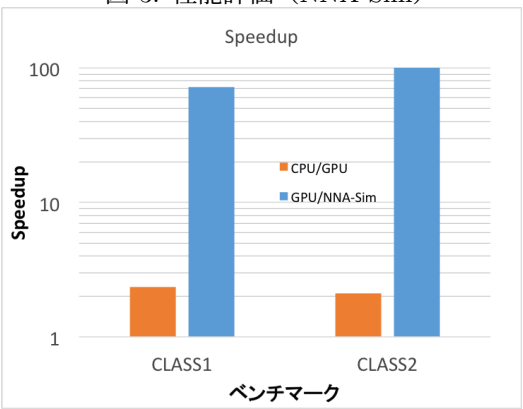
パラメータ	設定	パラメータ	設定
周波数	606MHz	入力ニューロン数	16
共有メモリ	4MB	出力ニューロン数	16
レイテンシ	10cycles	コア数	16

表 2: DaDianNao のパラメータ

5.2 評価結果

性能を比較すると（図 3）のようなグラフが得られた。論文では計 10 種類の NN の評価が行われていたが，時間

図 3: 性能評価（NNA-Sim）



の都合により、今回は CLASS と CONV のみ評価した。また、GPU の推論時間は、測定プログラムの問題で活性化関数の計算時間を計測できなかったため、積和演算のみの計算時間となっている。積和演算の計算時間は活性化関数のそれに比べて大きいため、活性化関数の計算時間を含めても結果にはほとんど影響がないと考えられる。DaDianNao の評価と比べるとほぼ同じくらいの性能であることが確認できた。

6 今後の予定

評価用 NN の LRN と POOL の性能評価を行う。CPU、GPU、アクセラレータの面積と電力の評価を行う。また、任意のニューロンモデル、学習アルゴリズムをシミュレートするために再構成可能ロジックをどのように使用すればよいかを検討する。

参考文献

[1] Y.Chen,el al. DaDianNao: A Machine-Learning Supercomputer MICRO 2014 pp.609-622

[2] P.A.Merolla,el al.A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface Science 2014 vol.345 no.6197 pp.668-673 2014