4L-05

Early-Exit 機構を有したニューラルネットワーク推論器の部分ハードウェア化

八木　颯太†　　松谷　宏紀†

慶應義塾大学大学院　理工学研究科†

1. 概要

深層学習は画像認識を始めとした多くの領域で活用されているが、近年需要の高まる携帯端末への活用においては、計算時間及び消費電力の削減が重要な課題となっている。

この課題の解決策の一つとして、推論ネットワークに分岐構造を導入することにより精度を担保しつつ計算コストを削減する Early-Exit が提案されており、本論文ではこれの一部分を専用ハードウェアとして FPGA に実装し更なる計算時間・消費電力の削減に取り組んだ。具体的には、ネットワーク内の分岐のうち層数が少なく計算コストの小さい部分をハードウェア化し、残りをソフトウェアで実装し評価を行った。

2. 関連研究

2.1. 組み込み機械学習

近年の携帯端末・IoT デバイスなどの普及にあたり、組み込み機械学習の需要が高まっており、それに合わせ様々な提案がなされている。文献[1]では重み値の 2 値・3 値化により、浮動小数点演算における電力消費やハードウェア領域の削減を図っている。またチップ内メモリを可能な限り活用する試みもなされている[2]。

2.2. 推論器への階層構造の導入

文献[3]は、学習により入力データをその推論の難易度に応じて適切なモデルを持った推論器に振り分ける手法を提案している。文献[4]では 2 種の規模の異なる NN 推論器をネットワークの出力値から推論結果の信頼度を求め、データの振り分けに活用する方法を提案している。

3. Early-Exit 機構

3.1. ネットワークの構成

2.2.節で述べた入力データの適切な配分の手法の 1 つとして Early-Exit が提案されている[5]。図 1 にネットワーク全体の構造及び処理の流れを示す。図中の C と書かれた四角形は畳み込み層、F と書かれた四角形は全結合層を表している。また図の簡略化のため、活性化層やプーリング層などは省略している。

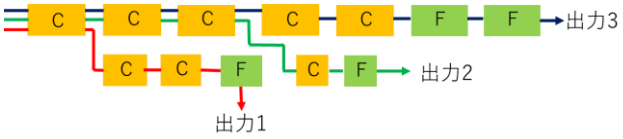

図 1:Early-Exit 構造を有したニューラルネットワークの例

初めに用意されたネットワーク(図中の出力 3 に向かう青矢印)について、その処理の途中からより単純なネットワークを枝分かれさせる形で配置する(図中の出力 1 に向かう赤矢印および出力 2 に向かう緑矢印)。入力に対してはまず出力 1 の推論結果を求めそのエントロピー値を求め、それが閾値を下回っている場合は推論結果が十分信頼できるとみなし処理を終了する。逆にエントロピーが閾値を超えている場合は次の出力の推論結果を求め(図 1 でいう出力 2)、同様にエントロピー値と閾値の比較を行う。これを最後の出力(図 1 の場合は出力 3)になるまで行う。

この手法によりデータセット内の推論の容易なデータについて計算コストを削減し、結果スループットの向上や消費電力の削減などの効果を得ることができると考えられる。

本提案ではこの機構を有したネットワークについて、構成層の少ない分岐にあたる部分に対しハードウェア化を施すことで、さらなる消費電力・計算時間の削減を目指した。

3.2. 処理の流れ

ネットワーク全体における学習は、各分岐の損失関数を重み付けし、それらの和として求まる関数の最適化問題を解くことによって行われる。以下に式を示す。

$$L_b(\hat{\boldsymbol{y}}, \boldsymbol{y}; \theta) = \sum_{n=1}^{N} w_n L(\hat{\boldsymbol{y}}_{exit_n}, \boldsymbol{y}; \theta)$$

Nは設計する Early-Exit ネットワークにおける出力の数を示している。$L(\hat{\boldsymbol{y}}_{exit_n}, \boldsymbol{y}; \theta)$は各分岐ネットワークにおける損失関数であり、$w_n$は分岐ごとに与える重み値である。

次にエントロピーを用いた推論の流れを述べる。まず推論結果に対するエントロピーを以下

Partial Hardware Implementation of Neural Network Inference using Early-Exiting

†Sota Yagi, Department of Information and Computer Science, Keio University.

†Hiroki Matsutani, Department of Information and Computer Science, Keio University.

の式で定める。

$$\mathrm{entropy}(\boldsymbol{y}) = \sum_{c \in C} y_c \log y_c$$

式中$\boldsymbol{y}$は全てのクラスに対する確率を含んだベクトルとなっており、Cは全てのクラスラベルを指す。これを各分岐内の推論結果に対して計算を行い、その値があらかじめ定めた閾値より小さければそこで処理を終了する。これを各分岐に対し行っていく。ここで、入力データが推論を終えるかを決めるための閾値ベクトルとして$\boldsymbol{T} = (T_1, \cdots, T_n)$を定義する。この値を大きく設定した場合、より多くのデータが早い分岐で推論を終えるが、精度が低下すると考えられる。小さく設定した場合は精度が向上するものの、エネルギー低減の効果が小さくなってしまう。

4. 評価

4.1. 評価環境

今回評価に使用した CPU は Intel Core i5-4460 で動作周波数は 3.2GHz である。また、推論器のハードウェア化には Xilinx 社 Vivado HLS 2016.4 を用い、実装対象の FPGA としては Virtex-7 XC7VX690TFFG1761-3 を想定して同社 Vivado により実行時間などのシミュレーションを行った。

今回評価に用いたネットワークは LeNet-5 で、分岐を設けた箇所は 1 つ目の畳み込み層の処理が終わった後の部分である。分岐先のネットワークには畳み込み層と全結合層を 1 つずつ設けた。評価に用いたデータセットは MNIST で、画像分類問題を今回のベンチマークとした。入力層のノード数は 28×28=784 とし、出力層のノード数は 10 に設定した。

表 1 に LeNet-5 に Early-Exit 機構を導入したネットワークをソフトウェア上で実装した時の元々の LeNet-5 との精度・実行時間での比較を示す。表 1 から、Early-Exit 機構の導入が実行時間の削減を達成していることが確認できる。

表 1:Early-Exit 機構を有した NN 推論器の性能比較

ネットワーク	精度(%)	実行時間(ms/1data)
LeNet	98.83	8.40
LeNet+Early-Exit	98.89	4.34

評価では 2 つあるうちの 1 つ目の出力に向かうネットワークをハードウェア化した際の性能について、全てのネットワークをソフトウェア上で実装した場合との比較を行った。

ハードウェアにおける実装には C++を用い、高位合成を施すことにより回路の構築を行った。表 2 に作成したハードウェアの目標動作周波数及び、作成したハードウェア内のテスト部分におけるクロック数を示す。

表 2:作成ハードウェアの情報

項目	
目標動作周波数	100MHz(1clk=10ns)
クロック数	144268clk

4.2. 評価結果

表 3 に、提案の推論器とソフトウェア上に全てのネットワークを実装した推論器との比較結果を示す。提案手法の実行時間はクロック数と動作周波数から見積り値を算出した。

表 3:提案手法の性能比較

	実行時間 (ms/1data)	精度 (%)
全ソフト実装	4.34	98.89
部分ハード実装	2.22	98.06

表 3 より、提案した推論器の方がスループットにおいて約 1.9 倍改善されていることが分かった。また推論の精度についても 0.8%程度の低下にとどまっており、精度を維持しつつの計算コストの削減ができていることが示せた。

5. 結論

本論文では Early-Exit 機構を含んだニューラルネットワーク推論器のうち、計算の簡単なネットワークについてハードウェア化を行った。その結果、ソフトウェアのみでの実行よりもエネルギー効率・スループットを向上させることができた。今後の課題として精度やスループットに一定の制約を設け、その下でデータの振り分けを行うなどの手法によりさらなる計算コストの削減を行うことなどが挙げられる。

参考文献

[1] G. Venkatesh, et al., “Accelerating deep convolutional networks using low-precision and sparsity,” ICASSP 2017.
[2] G. Desoli, et al., “Deep Convolutional Neural Network SoC in FD-SOI 28nm for Intelligent Embedded Systems,” ISSCC 2017.
[3] Z. Takhirov, et al., “Energy-efficient Adaptive Classifier Design for Mobile Systems.” ISLPED 2016.
[4] E. Park, et al., “Big/little Deep Neural Network for Ultra-low Power Inference,” CODES+ISSS 2015.
[5] S. Teerapittayanon, et al., “Branchynet: Fast Inference via Early Exiting from Deep Neural Networks,” ICPR 2016.