

# ニューラルネットワークを用いた省電力 IP ルックアップ方式の検討

林 遼<sup>1,a)</sup> 高田 遼<sup>1</sup> 坂本 龍一<sup>1</sup> 近藤 正章<sup>1</sup> 中村 宏<sup>1</sup> 児玉 康弘<sup>2</sup> 新 善文<sup>2</sup>

概要：ネットワークルータにおいて、ルーティングやフィルタリングの処理を行うために、IP アドレスから該当する情報を検索する必要がある。その実装に現在広く用いられている TCAM(3 値連想メモリ) は高コストかつ高消費電力であり、今後ネットワークに接続されるデバイス数の増加が見込まれるなか、その消費電力削減は重要な課題である。ハッシュテーブルの利用などの代替手法もあるが、ハッシュ競合など解決すべき問題がある。本稿では、ニューラルネットワークを用いた IP ルックアップ方式を検討し、その性能と電力を評価する。

## 1. はじめに

近年、ネットワークに接続されるデバイスの数は爆発的な増加を続けている。特に、様々なセンサを含む、あらゆるモノがインターネットに接続され、相互に情報伝達が行われる Internet of Things (IoT) の時代が到来しつつあり、今後ネットワークに接続されるデバイス数は更に増えていくことが予想されている [1]。

インターネットでは、パケットを中継しつつ情報の伝達を行うが、その際にルータが重要な役割を担っている。ルータではフロー制御やパケットのフィルタリングを行うために、パケットを送出すべき宛先ポートや、フィルタリングのルール、QoS 情報などが記憶されたテーブル(アクセス制御テーブル)を保持している。送られてきたパケットの IP アドレス、あるいは宛先の IP アドレス等をキーとしてそのテーブルを検索し、対応する情報を読み出すことを行うべき処理を決定する。この IP ルックアップは、送られてきた全パケットに対して行う必要があるため、高速にキーに該当するエントリを参照することが重要である。

IP ルックアップによく用いられるのが TCAM (Ternary Content Addressable Memory) と呼ばれる 3 値連想メモリ [2] であり、1 アクセスでメモリ内の全エントリを並列に検索可能なため、高速なルックアップが行える。特に基幹ネットワークのルータでは、高スループットを安定して得られることが重要なため、TCAM は現在最もよく利用されている。

しかし、TCAM は製造コストと消費電力が高いという課題がある。並列検索を行うことから、保持するエントリ数にほぼ比例して消費電力が増大するため、インターネットに接続されるデバイスの数が飛躍的に増加し、処理するパケット量とテーブルに登録すべきエントリ数が増大すると予想される今後の IoT 時代では、解決すべき重要な問題になると考えられている。TCAM 以外に IP ルックアップを行う手法としてハッシュや  $n$  分木(トライ木)を用いたものも存在するが、TCAM に比べて省電力ではあるものの、スループットや性能安定性に課題がある。

本稿では、省電力で IP ルックアップが行えるハッシュ手法の代替として、ニューラルネットワークを利用した IP ルックアップ手法とそのルータアーキテクチャを検討する。ニューラルネットワークは脳の神経回路を模した数学モデルであり、様々な非線形関数を近似可能である。この性質を利用して、IP アドレスを入力、アクションを保持する外部メモリアドレスを出力とする非線形関数を機械学習により生成することで、高速な IP ルックアップが実現可能と考えられる。さらにニューラルネットワーク演算の高速・省電力化、またハードウェアの小規模化の目的で、本稿では計算の大部分を二値化した Binarized Neural Network (BNN)[3] による実装を検討する。小規模な TCAM と組み合わせることで、高速な IP ルックアップと省電力化を両立し、高い性能安定性も得られると期待される。

提案 IP ルックアップ手法の有用性を評価するために、10 万エントリ規模のテーブル検索を BNN で学習し、その精度調査と、回路構成を検討し、スループットおよび消費電力の予備評価を行った。本稿では既存の TCAM と比較しつつ、提案手法の有効性について議論する。

<sup>1</sup> 東京大学, The University of Tokyo.

<sup>2</sup> アラクサラネットワークス株式会社, ALAXALA Networks Corporation.

a) ryohayashi@hal.ipc.i.u-tokyo.ac.jp

## 2. IP ルックアップ

### 2.1 IP ルックアップの概要

ネットワークルータでは、送られてきたパケットに対し、ルールに従って他のルータへそのパケットを転送する。また、セキュリティ対策などの目的で、当該パケットを実際に転送すべきか、あるいはブロックすべきかなどのフィルタリング処理も予め登録されたルールにしたがい行う。各送信先・送信元 IP アドレス（さらにポート番号の組）に対してどのような操作を行うかは、あらかじめルータ上のメモリに記録されており、ルータはパケットが送られてくるたびに当該メモリに構築されたアクセス制御テーブルを参照し、対応する操作を行う。この、送られてきたパケットに対し、送信先 IP アドレスなどからテーブルを参照することを IP ルックアップと呼ぶ。IP ルックアップでは、対応する IP アドレスなどからテーブル上のどのエントリを参照すれば良いかの検索を行う必要がある。これを高速に行うことが可能であれば、単位時間により多くのパケットを処理できることにつながるため、ルータのスループットが向上する。したがって、IP ルックアップはルータにおいて最も重要な機能の一つである。

アクセス制御テーブルは IP アドレスやポート番号全体をキーとしてアクセスされるとは限らず、プリフィックス（接頭辞）部のみによって、該当するエントリが決定されることも多い。プリフィックスを除いた部分はワイルドカードとして扱われる。プリフィックス長は一意ではなく、Longest Prefix Matching (LPM) と呼ばれる、検索する IP アドレスと接頭辞が最も長く一致するエントリをアクセスするという規則が用いられる。

### 2.2 従来の IP ルックアップ手法

現在 IP ルックアップとしてもっとも一般的に用いられている TCAM は、高速なルックアップが可能である反面、消費電力やハードウェアコストが大きいという課題がある。今後ネットワークに接続されるデバイスが増えるに従いさらに増加すると考えられている。代替手法としてハッシュやトライ木などの手法が研究されているが、それぞれのデータ構造上の問題から高速なルックアップを実現するには至っていない。以下、それら従来の IP ルックアップ手法を紹介する。

#### 2.2.1 TCAM

TCAM の各エントリは、検索するキーが格納されたエントリからなり、各セルは“0”、“1”、または“don't care”の3状態をとる。プリフィックスが長いエントリから順にテーブルに記録し、IP アドレスが入力されると全エントリが並列に検索され、“don't care”以外のビットが完全一致する全てのエントリがマッチングし、Priority Encoder を通して最上位のエントリの番号が出力される。IP ルックアップ時には、この TCAM の出力をアドレス（の一部）として、実際に制御情報が格納されているメモリをアクセスし、制御を行う。TCAM は全エントリを並列に検索することで、高速なルックアップを実現している反面、消費電力が大きくコストも高い [4]。

クアップ時には、この TCAM の出力をアドレス（の一部）として、実際に制御情報が格納されているメモリをアクセスし、制御を行う。TCAM は全エントリを並列に検索することで、高速なルックアップを実現している反面、消費電力が大きくコストも高い [4]。

#### 2.2.2 ハッシュ

ハッシュはあるデータに対して別の数値を割り当て、データ検索の高速化やメモリの小容量化に用いる良く知られた手法である。IP ルックアップでは、検索する IP アドレスに対してハッシュを適用し、テーブルの該当エントリが存在するメモリアドレスを得る。ハッシュを用いる場合、2つの異なる IP アドレスが同じハッシュ値をとる衝突が起こる可能性がある。衝突への対処として、1つのハッシュ値に複数のキー登録を許し、同じハッシュ値のグループの中でさらに検索をするという方法や、衝突した場合にはある規則に従い別の値へと変換を繰り返すなどの方法がとられるが、検索時間が一定に定まらず、最悪時間の保証ができなくなるという欠点を持つため、特に基幹ルータなど、安定的な IP ルックアップのスループットが求められるルータには不向きである。さらに、LPM の実現には複雑な処理が必要となるという課題もある。

#### 2.2.3 トライ木

トライ木は、IP アドレスのプリフィックスの各ビットの 0,1 にしたがって分岐していくデータ構造を表す。分木を重ねリーフに位置するエントリより、テーブルの該当エントリが存在するメモリアドレスを得る。トライ木では構造上、検索遅延が大きくなりがちであり、親ノードが子ノードへのポインタを保持しなければならないため必要なメモリ容量が大きくなる。また高スループットを得るためには、メモリ多数用いてパイプライン的にアクセスする必要があり、コストも高くなる可能性がある。

### 2.3 関連研究

ルータの消費電力は、2010 年時点で既に国内総発電量の 0.4-1.7% に達しているという報告もあり [5][6]、その低消費電力化は社会的に見ても大きな課題であると考えられている。そこで、高速な IP ルックアップと低消費電力の両者を実現可能なルータアーキテクチャに関して様々な研究が行われている。

文献 [7] では、ハッシュの衝突が生じないようなハッシュを用いた IP ルックアップ手法が提案されている。本稿と同様に、小規模の TCAM を併用して最終的なルータアーキテクチャを構成している。文献 [8] では、トライ木を保存するデータ構造を圧縮してメモリ上に保存することで、容量効率に優れたトライ木をベースとした IP ルックアップについて述べられている。文献 [9] では、ニューラルネットワークの一種である Sparse Clustered Networks (SCN) を用いた IP ルックアップ手法が提案されている。さらに、キャッシュを用いて TCAM へのアクセスを減らすことで

高速・省電力化を図る研究も行われている [10], [11].

### 3. ニューラルネットワークを用いた IP ルックアップ手法

#### 3.1 ニューラルネットワークの利用

本稿では、検索 IP アドレスのキーに対して対応する制御情報が記録されたテーブルのエントリの番号 (メモリアドレス) を得るために、ニューラルネットワークを応用する。ニューラルネットワークは様々な非線形関数を近似可能であり、IP アドレスを入力、該当するアクションを保持するテーブルの番号を出力とする非線形関数モデルを機械学習により生成することで、IP ルックアップを実現する。

非線形関数モデルの生成には、検索する可能性のある IP アドレスを入力、その情報が格納されたエントリ番号を教師値として学習を行う。ニューラルネットワークは統計的学習手法であり、学習の結果として常に検索キーの入力に対し、対応するエントリ番号が正しく出力されるモデルが生成されるとは限らない。つまり、精度 100% を保証することはできず、正しいルックアップを行うことのできない入力が存在する可能性がある。本稿では、それへの対処として、ニューラルネットワークと、補助のルックアップを行う小規模な TCAM を組み合わせる手法を検討する。

学習したモデルに対して、予め正しい出力が得られるかを調査する。正しい出力が得られないキーに関しては、正しいキーとエントリ番号出力の対応を小規模 TCAM とそれに対応する小規模メモリに保存し、もし TCAM に一致するエントリがあった場合は、その結果を優先して利用することにする。これで、精度が 100% でない場合も正しいルックアップが行えると考えられる。また、ルータではルーティング情報の更新や新しい IP アドレスの登録など、運用中にもアクセス制御テーブルの更新が必要となることがある。ニューラルネットワークの学習にはある程度の時間を要するため、その間の一時的な新規キーのルックアップ用にも小規模 TCAM を利用する。

#### 3.2 Binarized Neural Network を用いた実装

一般的なニューラルネットワークの計算には、多数の積和演算が必要であり、逐次的に計算処理を行うと性能が大きく低下する。しかし、ハードウェアで積和演算を並列に実行すると、その回路コストや消費電力が大きくなるという問題がある。そこで、本稿では低ハードウェアコストで、ニューラルネットワークの計算処理が行える Binarized Neural Network (BNN)[3] の適用を考える。

BNN は図 1 に示すように、ニューラルネットワークの中間層の出力結果、およびパラメータ (重み, バイアス) の値を +1 と -1 に二値化することで、全体の計算コストとメモリ容量を削減する手法である。二値化による精度低下を、Batch Normalization という正規化処理により抑える手法が提案されている [12] の。BNN を利用すると、数値が二

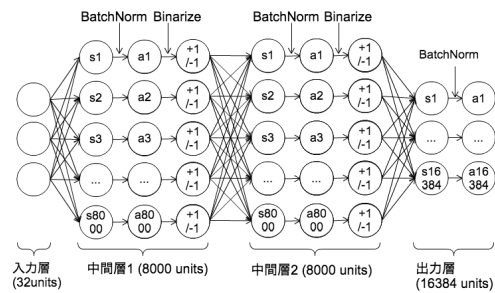


図 1 Binarized Neural Network の例

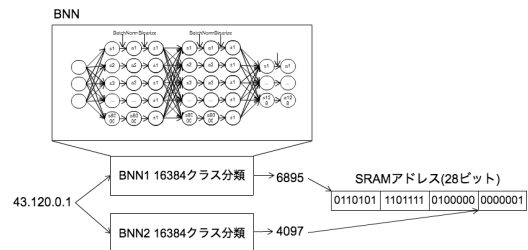


図 2 BNN を用いた IP ルックアップアーキテクチャ

値化されていることによって、必要な演算処理を XNOR 演算とその結果で 1 が出力されたビット数をカウントする POPCOUNT 処理に置き換えることができ、回路の小規模化と消費電力の削減および、ルックアップ処理の高速化を図ることができる。

BNN は Batch Normalization を行うために、回帰問題には不向きである。つまりエンコードされたエントリ番号を出力結果とすると高い精度は得られない。そこで、入力された IP アドレスをエントリ番号に対応する個数のクラスに分類する分類問題に帰着させることでルックアップを行うことを考える。例えば、エントリ数が  $2^{28}$  個のテーブルであれば、 $2^{28}$  クラスに分類する分類問題となる。しかし、 $2^{28}$  クラスの分類問題は、一般的な学習に利用するフレームワークではメモリサイズ不足により学習の計算ができない。加えて、出力として  $2^{28}$  ビットを出力するようなハードウェアを実装することも現実的ではない。そこで、図 2 に示すように、今回は出力エントリ番号のビット列を  $N$  個に分割し、それぞれを  $2^N$  クラスの分類問題に帰着させることで実装する。先と同様に  $2^{28}$  エントリのテーブルであれば、 $\frac{28}{N}$  ビットずつの  $N$  個に分割することになる。

この分割が精度に与える予備評価として、32 ビットの IP アドレスから 28 ビットのアドレス空間を持つメモリのアドレスへと変換する際の精度を評価した。本予備評価では、実パケットトレースデータ [13] から送信先 IP アドレスを抽出したものを入力、各アドレスに対して 28 ビットのテーブルのエントリ番号をランダムに割り当てたものを出力とする教師データセットを作成して学習データとした。BNN の学習には、ディープラーニングのフレームワークである Chainer を用いた。

実験的に、分割数  $N$  を 2, 4, 7, 14 とした場合の精度をプロットしたグラフを図 3 に示す。分割数が少ないほど、良

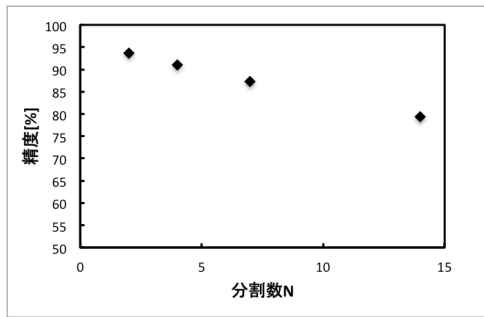


図 3 分割数 N=2, 4, 7, 14 の場合のルックアップの精度

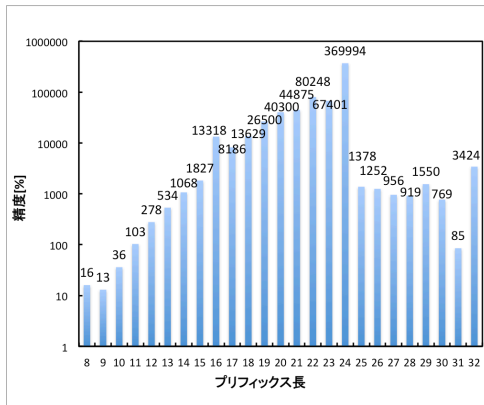


図 4 Route Views Project[14] のプリフィックス長の分布

い精度が得られていることがわかる。分割数が少ない場合 1 つの BNN で分類するクラス数は大きくなるが、全体の精度は N 個の BNN の精度の積となるため、結果的に分割数が少ないほど高い精度が得られるためである。そこで、現実的に実装可能な BNN 構成の一つとしてアドレスビットを 2 分割し、またその他の予備評価より、ネットワークの構成は中間層が 2 個でユニット数 8000、出力ノード数 16384 の 4 層 BNN (図 1 参照) をベースとして以降議論を行う。

### 3.3 ワイルドカードサポート

2.1 節で述べたように、実際のルータでは長さの違うプリフィックスによってアクセス制御テーブルが構成され、LPM にしたがってルックアップが行われる。一方で、前節までに述べた手法では基本的に入力サイズは固定であり、入力に対して 1 対 1 で出力が決定される。そのため、プリフィックス長の違う入力をそのまま学習させることはできない。そこで、予め学習時にワイルドカードとしてマッチする入力 IP アドレスを展開し、それらには同じエントリ番号が出力されるように教師データを作成することで LPM を実現する。ただし、入力パターン数増加により精度が低下する恐れがあるため、本手法を適用するプリフィックス長の範囲を限定する。

オレゴン大学の Route Views Project[14] が公開している実際のルーティングテーブルのデータを用いて、プリフィックス長の分布を調べると、図 4 に示すように、8 ビッ

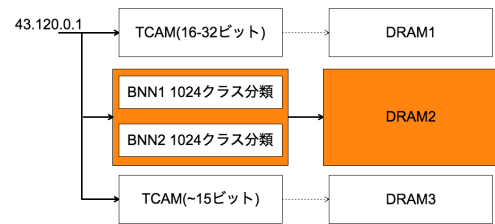


図 5 BNN と TCAM による全体アーキテクチャ

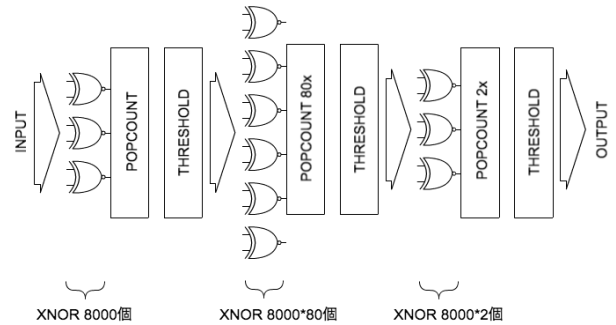


図 6 BNN のハードウェア構成

ト未満のエントリは存在せず、また 16-24 ビットのエントリが全体の約 97.9% を占めることがわかる。そこで、図 5 に示すように、BNN では 16 ビットから 24 ビットまでのプリフィックス長の IP アドレスに対するルックアップを行い、それ以外の 16 ビット未満または 25 ビット以上のプリフィックス長の IP アドレスは TCAM に保存し、BNN と TCAM で並列に IP ルックアップを行うことで全体のアーキテクチャを構成する。先に述べた BNN によって正しく出力を得られない場合と同様、ルックアップ時に TCAM に一致するエントリが存在する場合で、かつ TCAM が 16 ビット以上のプリフィックスのものであれば TCAM の出力を優先して利用する。一方、16 ビット未満のプリフィックスであった場合は BNN の結果を利用する。

### 3.4 BNN を実現する回路構成

本節では、3.2 節で述べた BNN を用いた IP ルックアップをハードウェアで実現する上での回路構成を示す。BNN で主となる XNOR、および POPCOUNT を用いたハードウェア構成を図 6 に示す。図中の“THRESHOLD”については後に述べる。なお、BNN の出力ビットの分割数に応じて、本ハードウェアを並列に実装する必要がある。また、重みのデータはチップ内の SRAM に保存する必要があるが、それぞれが 1-bit のデータであり、たかだか合計数 100KB 程度の容量となる。

BNN の推論時には Batch Normalization の演算が必要である。当該演算はやや複雑なものであるが、BNN では各層の出力は最終的に Sign 関数によって二値化されるため、この演算はある閾値に対する二値化演算と見なすことができる [15]。そこで、予め閾値を計算しておき、レジスタに保存しておくことで、単純な比較演算となる。なお、比較

表 1 BNN1 つあたりの各層の回路数

回路	個数
XNOR1	8000
POPCOUNT1	1
XNOR2	8000 * 80
POPCOUNT2	80
XNOR3	8000 * 2
POPCOUNT	2

演算は各ノードの積和の出力に対して行えばよく、回路の個数は比較的少数で済むため、以降の評価では含めない。

なお、必要な全演算を個別の回路として実装すると、回路規模が大きく成りすぎてしまう。そこで、積和演算に相当する XNOR と POPCOUNT 回路は IP ルックアップ 1 回の中で複数回使い回すことを想定する。およそ 1 秒間に  $10^8$  以上の IP ルックアップのスループットをターゲットとした場合に十分な性能が得られる範囲で XNOR と POPCOUNT 回路を縮小すると、最終手的に本稿で検討する BNN 構成では、各層の各回路数は表 1 のようになる。

## 4. 評価

### 4.1 評価方法

本稿では、提案する BNN を用いた IP ルックアップ機構に関して、スループットと消費電力に関して初期評価を行う。また、消費電力に関しては、TCAM を用いる場合との比較評価を行う。評価にあたり、あるプロセスを用いた際の XNOR と POPCOUNT の遅延時間および消費電力を評価した。POPCOUNT は 64bit の POPCOUNT を複数個用い、その結果を 10bit 加算器をツリー上に接続して集計する構成とした。なお配置配線までは行わず、合成後のレポートより回路遅延と消費電力を算出した。そのため、遅延と電力の値は参考値であり、また仮定するプロセスによっても大きくその値が異なるため、本評価では XNOR と POPCOUNT の遅延と電力の比を合成結果より算出し、XNOR ゲートの基本遅延と電力をパラメータとして変更しつつ評価を行うこととする。

### 4.2 評価の仮定

本稿では、初期評価としてアクセス制御テーブルに登録するアドレスは全て 32 ビット IPv4 アドレスとし、プリフィックス部分として、そのうちの 24 ビットを BNN の入力とする。制御情報を記録するアクセス制御テーブルを保存するメモリは 20 ビットのアドレス空間を持つと仮定し、そのため BNN の出力は合計で 20 ビットとなる。また、分割数は 2 とし各 BNN は 10 ビットを出力する。各 BNN は 8000 ノードを持つ中間層が 2 層であり、入力層と出力層を合わせて合計 4 層とする。

なお、アクセス制御テーブルの更新頻度は高くないと考えられるため、その影響は無視する。BNN の関数モデルの作成は、別途他の CPU あるいは GPU で行うものとし、

表 2 入力パターン数に対する BNN の推定精度

入力パターン数	精度 [%]
20000	99.9
40000	99.2
60000	99.0
100000	92.1

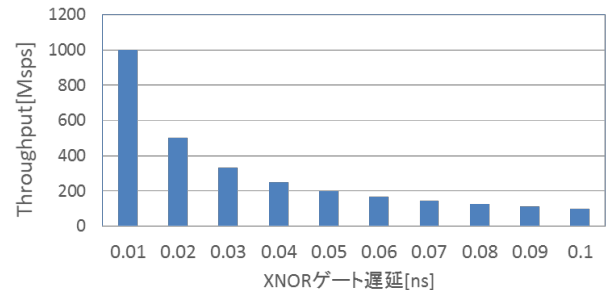


図 7 スループットの評価結果

今回の評価では学習後のモデルが既にチップ内のメモリに保存されていることを仮定する。このような条件のもと、テーブルに登録する IP アドレス数は約 10 万アドレスとして評価を行う。

## 5. 評価結果

### 5.1 BNN の精度

まず、入力の IP アドレスのパターン数を変化させ、対象の構成を用いた場合の BNN で正しくルックアップが行える精度を評価した。評価結果を表 2 に示す。表より、パターン数が少ない場合は 99% 以上の高い精度で BNN により IP ルックアップが行えることがわかる。一方で、10 万パターン程度になると、精度は 92% 程度まで悪化してしまう。ただし、出力のエントリ番号の与え方を工夫することで精度は改善可能であり、この精度改善の検討は今後の課題である。

### 5.2 スループット評価

XNOR ゲートの遅延時間の想定をパラメータとして変化させた場合の、BNN による IP ルックアップのスループットの評価結果を図 7 に示す。スループットは 1 秒間に IP ルックアップを行える回数 (sps: Search Per Second) で示されている。層毎に別ステージとしたパイプライン化を想定しているため、最もクリティカルパスの長いステージでスループットは律速される。本稿の評価の仮定では、中間層 2 がクリティカルパスである。当然であるが、ゲート遅延が短いほどスループットは高くなる。例えば XNOR ゲート遅延が 0.05ns の場合は 200MSPS を達成可能であり、将来的に基幹ルータに要求されるであろう高いスループットを達成できる可能性を示唆している。

### 5.3 消費電力評価

XNOR ゲートの消費電力の想定をパラメータとして変化

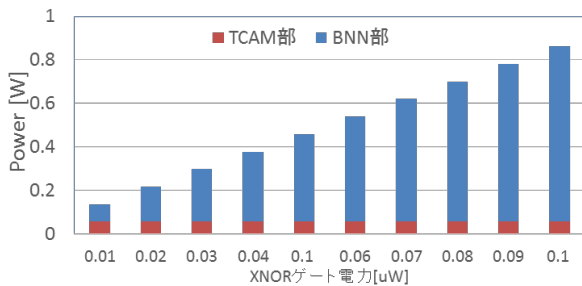


図 8 消費電力の評価結果

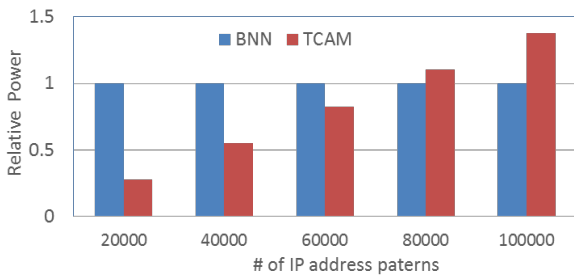


図 9 TCAM と BNN の消費電力比較

させた場合の、消費電力の評価結果を図 8 に示す。回路の 1 と 0 の遷移率は 0.3 を仮定した。本評価では、並列に検索する TCAM 部は 10000 エントリの TCAM を用いることを仮定した。10 万パターンの IP アドレスのルックアップであれば、BNN 部の精度が 92% で、またプリフィックスが 16-24 ビット以外のエントリを TCAM にアクセスするとしても、10000 エントリの TCAM で十分である。なお、TCAM の消費電力は文献 [16][17] の TCAM 電力モデルにしたがって見積ったが、文献では 0.18 $\mu$ m 半導体プロセスを仮定しているため、65nm の半導体プロセスを想定して容量の値はそれに合わせてスケールさせた。

本結果をもとに、XNOR ゲートの消費電力が 0.05[uW] の場合について、TCAM のみを利用して IP ルックアップを行う場合と比較する。入力パターンを変化させた場合の BNN を用いた場合に対する TCAM の相対消費電力を図 9 に示す。TCAM は入力パターン数が増えるに従って消費電力が増大するのに対して、提案手法は補助で用いる TCAM の容量が足りていれば、入力パターンに対する消費電力の増加はない。入力パターン数が増えると提案手法は TCAM よりも省電力性に優れていることがわかる。

## 6. おわりに

本稿では、インターネットルータの省電力化に向け、ニューラルネットワークの一種である Binarized Neural Network を用いた IP ルックアップ手法を提案した。実パケットトレースを用いて、IP アドレスを入力、アクセス制御テーブルのエントリ番号を出力として学習を行い、精度的にも IP ルックアップに応用可能であるとの知見を得た。スループットと消費電力の評価を行った結果、実用的に十分なスループットを得ることができ、また、テーブルのエ

ントリ数が多い場合には TCAM ベースの手法に対して消費電力も低く抑えることができることがわかった。BNN の出力のエントリ番号の与え方を工夫して BNN の精度を改善することや、回路構成の最適化、より現実的なパラメータにより評価を行うことなどが今後の課題である。

謝辞 本研究の一部は NEDO のプロジェクト「IoT 社会の実現に向けた電子・情報分野事業の周辺技術・関連課題における小規模研究開発 / IoT 情報基盤のための省電力ネットワークセキュリティ技術の研究開発」の助成により行われたものである。

## 参考文献

- [1] A. Zanella, et al., "Internet of Things for Smart Cities, IEEE INTERNET OF THINGS JOURNAL, Vol.1, No.1, (2014).
- [2] S.K. Maurya, L.T. Clark: A Dynamic Longest Prefix Matching Content Addressable Memory for IP Routing, IEEE Tr. on VLSI Systems, Vol.19, No.6, (2011).
- [3] M. Courbariaux, et al.: Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1", arXiv:1602.02830 [cs.LG], (2016).
- [4] 奥田ほか: TCAM の並列パイプライン化により検索エンジンの省電力化を実現する多次元 TCAM アーキテクチャの提案, 信学技報 IN2014-7, (2014).
- [5] 小笠: 情報通信のエネルギー問題 - 求められる通信インフラの省電力化 -, 文部科学省 科学技術動向, (2006).
- [6] 阿多ほか: 低コスト・低消費電力 TCAM における効率的なルーティングテーブル管理法, 信学技報, Vol.107, No.443, NS2007-120, pp.7-12, (2008).
- [7] J. Hasan, et al.: Chisel: A Storage-efficient, Collision-free Hash-based Network Processing Architecture, Proc. of the 33rd ISCA (ISCA'06), pp.203-215 (2006).
- [8] W. Eatherton, G. Varghese, and Z. Dittia: Tree bitmap: Hardware/Software IP Lookups with Incremental Updates, ACM SIGCOMM Computer Communication Review, Vol.34, Issue 2, pp.97-122 (2004).
- [9] 鬼沢ほか: Sparse Clustered networks に基づく IP lookup 処理用 LSI 実現に関する研究, 信学技報 CAS2012-123, SIP2013-168, CS2013-136 (2014).
- [10] G. Liao, et al.: A New IP Lookup Cache for High Performance IP Routers, Proc. 47th DAC (DAC2010), 338-343 (2010).
- [11] 八巻, 西: L1 規模のパケット処理キャッシュにおけるライン入れ替え方式の検討, 情報処理学会研究報告, Vol.2016-ARC-221, No.16, (2016).
- [12] S. Ioffe and C. Szegedy: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, Proc. ICML, pp.448-456 (2015).
- [13] WIDE MAWI Working Group: MAWI Working Group Traffic Archive, (<http://mawi.wide.ad.jp/mawi/>).
- [14] <http://www.routeviews.org/>
- [15] Y. Umuroglu, et al.: FINN: A Framework for Fast, Scalable Binarized Neural Network Inference, arXiv:1612.07119 [cs.CV] (2016).
- [16] B. Agrawal and T. Sherwood: Ternary CAM Power and Delay Model: Extensions and Uses, IEEE Tr. VLSI, Vol.16, No.5, pp.554-564 (2008).
- [17] B. Agrawal and T. Sherwood: Modeling TCAM Power for Next Generation Network Devices, Proc. of IEEE ISPASS-2006, pp.120-129 (2006).