ベイジアンフュージョンによるスパイキングニューラルネットワークの低エネルギ推論

赤木 勇統[1,a)]　粟野 皓光[2,b)]

概要：スパイキングニューラルネットワーク（SNN）はそのエネルギ効率の高さによる注目を集めている．しかし，一般的な SNN はスパイク発火頻度で情報を表現するため，一定期間のサンプリング時間が必要であり，依然として改善の余地がある．そこで，本研究では，ネットワークの中間層から早期予測結果を得て，これを最終層の予測結果とベイズ統合することで，サンプリング時間の更なる短縮と，省エネルギ化を実現する．Cifar-100 を対象とした画像分類タスクにおいて，ResNet18 をベースにしたネットワークに提案手法を適用することで，51.5% のエネルギ削減が達成可能であることを示した．

キーワード：深層学習，スパイキングニューラルネットワーク，低消費電力，ベイズ

Low Energy Inference of Spiking Neural Network via Bayseian Fusion

Yuto Akagi[1,a)]　Hiromitsu Awano[2,b)]

Abstract: Spiking neural networks (SNNs) have attracted much attention due to their high energy efficiency. However, since general SNNs represent information by spike firing frequency, they require a certain period of sampling time, and there is still a room for improvement. In this study, we further reduce the sampling time and save energy by obtaining the early prediction results from the middle layer of the network and Bayesian integration with the prediction results of the final layer. In the image classification task for Cifar-100, we have shown that by applying the proposed method to the ResNet18-based network, an energy reduction of 51.5% can be achieved.

1. はじめに

近年，人間の脳を模倣した人工ニューラルネットワークの研究が盛んである．特に，画像分類や物体認識の画像タスク等の限定された分野においては，最新のニューラルネットワークは人間の識別性能をも凌駕する程の圧倒的な性能を示しつつある．

現在広く用いられているニューラルネットワークは，形式ニューロンモデルというニューロンモデルを利用している．このような，形式ニューロンモデルを利用するニューラルネットワークを Artificial Neural Network (ANN) と呼ぶ．形式ニューロンモデルは，実数ないしは整数値で表現される入力と重みの積和を算出し，ReLU やシグモイド関数などの非線形な活性化関数を通して，アクティベーションと呼ばれる出力を得る．一般に，アクティベーションや重みは単精度の浮動小数点や 8-bit の整数値で表現される．例えば 8-bit 精度を仮定すると乗算 1 回あたりに必要なエネルギは 0.2pJ であり [1]，一見すると非常に小さいエネルギに思えるかも知れない．しかし，現代のニューラルネットワークは 100 万以上ものニューロンから構成されており，推論 1 回に占める乗算エネルギは非常に大きなウェイトを占めている．従ってニューラルネットワーク推論の低消費電力化に向けては，乗算に要するエネルギの削減が必須である．

近年，アクティベーションを数値として表現する ANN に代わって，スパイク発火で表現していると考えられる生体脳の働きを模倣した Spiking Neural Network（SNN）が

[1] 大阪大学 大学院情報科学研究科 情報システム工学専攻
[2] 京都大学 大学院情報学研究科 通信情報システム専攻
[a)] y-akagi@ist.osaka-u.ac.jp
[b)] awano@i.kyoto-u.ac.jp

注目されている．SNN を用いることで，アクティベーションを 1-bit（スパイク発火の有無）で表現でき，8bit 乗算器を必要とする ANN 向けハードウェアアクセラレータと比較して回路規模の削減が期待できる．また，SNN において，各ニューロンはスパイクを受け取った時にのみ膜電位を更新すれば良いため，非同期回路との相性が良く，更なる消費エネルギの削減が期待できる．現に，SNN 向けハードウェアアクセラレータの一種である IBM の TrueNorth [2] は，ニューロン間の通信を非同期とすることで大幅な消費エネルギ削減に成功している．

従来，画像分類や物体認識などの実用的なタスクにおいて十分な性能を発揮できるように SNN を学習させることは困難であったが，ANN の学習結果を SNN に転写する技術が開発されたことで，VGG19 や YoLo 等の複雑なネットワークの SNN 置換が進んでいる [3][4]．例えば，Seijoon Kim らは YoLo を SNN に変換することで，ANN 実装と比較して 280 倍ものエネルギ効率が達成できることを示した [5]．

ANN と比較して優れた特性を有する SNN ではあるが，依然としてエネルギ・レイテンシ削減の余地が残されている．これは，SNN が情報表現にスパイク発火確率を利用しているため，発火確率を十分な精度で測定するためには十分に長いスパイク発火系列を観測する必要があるためである．

この様な背景のもと，本研究では，ベイジアンフュージョンを利用した SNN の推論エネルギ・レイテンシ削減手法を提案する．スパイクは発火状態とそれ以外を取る 2 値変数として表現でき，特定の時間内に観測されたスパイク発火数は 2 項分布に従うとしてモデル化できる．つまり，スパイク発火系列で表現された情報をデコードすることは，スパイク発火系列を生成する尤もらしい 2 項分布のパラメータを求めることと等価である．一般的に推論精度とレイテンシ・エネルギはトレードオフの関係にあり，スパイク発火系列を長時間観測できれば，その分だけパラメータも高精度に推定できるが，反対にレイテンシは増大し，エネルギ効率は低下してしまう．本研究では，スパイク発火系列の観測長を減らすことによる情報劣化を，スパイク発火のしやすさに関する事前知識で補うことで，推論精度を維持しつつエネルギ・レイテンシの削減を実現する．具体的には，ネットワークの浅い階層に存在するニューロンの発火系列を用いて，最終出力層のニューロンが発火するであろう確率を予測し，実際に最終出力層で観測された発火系列とベイズ統合することで，発火確率推定精度を損なうことなく，スパイク発火系列の観測長削減を実現する．ResNet18 を用いた数値実験の結果，CIFAR100 画像分類タスク [6] において，推論精度を同等に保ちつつ，エネルギを 51.5%削減できることが明らかになった．

2. 事前知識

2.1 Artificial Neural Network (ANN)

ANN の各ニューロンでは，式 (1) で示すように，入力アクティベーション $\boldsymbol{x}$ とシナプス結合重み $\boldsymbol{w}$ の積をとり，バイアス b を加算したものを非線形関数 f に通して，出力アクティベーション y を得る．

$$y = f\left(\sum_{i=1}^{n} \boldsymbol{w}_i \boldsymbol{x}_i + b\right) \tag{1}$$

アクティベーション値の表現には，単精度浮動小数や半精度浮動小数点数，それから 8bit 整数などが多く用いられる．

ANN において，訓練中にある層のパラメータが変化した際に，それ以降の層に対する入力分布が変化する．この変化分は層が深くなるに連れて増大するため，学習の発散を抑えるために学習率を大きく取れないという問題があった（共変量シフト）．これを解消するためにバッチ正規化 (BatchNormalization) [7] が提案された．バッチ正規化では，ミニバッチ毎に層の入力が平均 0，分散 1 となるように正規化し，正規化した後で学習可能なスケーリング係数とバイアスパラメータによってスケーリング処理とバイアス処理を行う．

$$y' = \gamma \frac{y - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{2}$$

ここで μ，σ^2 はそれぞれミニバッチ毎の平均，分散であり，γ，β は学習可能なスケーリング係数とバイアスパラメータである．推論時には μ，σ^2，γ，β は固定され定数として扱われるため，推論時にはバッチ正規化層を直前の線

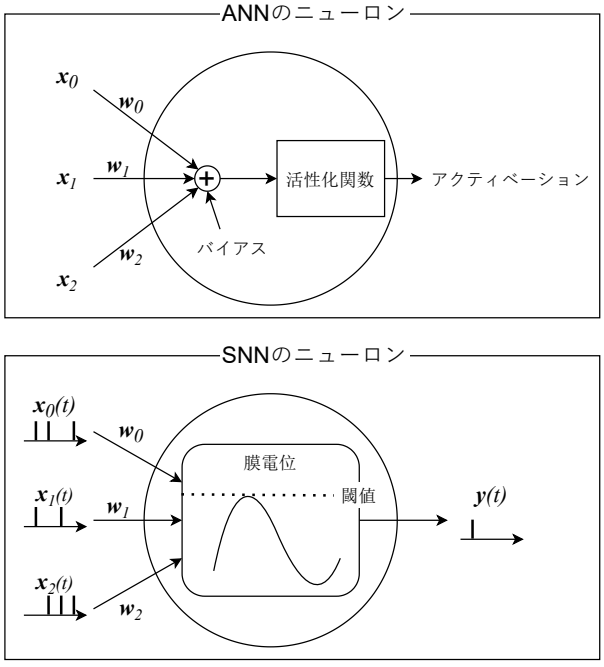

図 1 ANN と SNN のニューロン

形レイヤに統合し，演算を減らすことができる．具体的には，式（3）及び（4）で示すように，直前の層の重みに統合することができる．

$$\hat{\boldsymbol{w}}_i = \frac{\gamma}{\sigma}\boldsymbol{w}_i \tag{3}$$

$$\hat{b_i} = \frac{\gamma}{\sigma}(b_i - \mu) + \beta \tag{4}$$

2.2 Spiking Neural Network (SNN)

生体脳はスパイク発火と呼ばれる一過的電圧信号で情報を表現していると考えられており，これを模した計算モデルが SNN (Spiking Neural Network) である．SNN の計算モデルには様々な種類が提案されているが，本研究では最も普及しているとされ，ハードウェア実装も多数提案されている漏れ積分発火（Leaky integrate-and-fire; LIF）モデルを利用する．LIF モデルにおいて，ニューロンは内部状態として膜電位を持ったノードとして表現さる．各ニューロンは，他ニューロンからのスパイクが入力された時に，そのニューロンとの間にあるシナプス結合重みに応じて膜電位を更新する．この動作は以下の様に表現される．

$$V_i^{(t)} = V_i^{(t-1)} + \sum_j w_{ij}\Theta_j^{(t-1)} + b_i \tag{5}$$

ここで，$V_i^{(t)}$ は時刻 t でのニューロン i の膜電位，w_{ij} はニューロン i から j へのシナプス結合重み，b_i はニューロン i のバイアス値である．$\Theta_j^{(t-1)}$ はニューロン j の時刻 $t-1$ におけるスパイク発火の有無を表現する 2 値変数であり，以下の様にニューロン j の膜電位から計算される．

$$\Theta_j^{(t-1)} = \begin{cases} 1 & V_j^{(t-1)} > V_{th} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

ここで V_{th} は閾値電圧である．各ニューロンはスパイクを発火した後に膜電位をリセットする． 膜電位のリセットの方法には膜電位を 0 とする方法と，閾値分を減算する方法の 2 種類あり，後者の方が情報劣化が少ないことが知られているため [4]，本研究でも後者のリセット方式を採用する．従ってスパイク発火直後の膜電位は以下の様に更新される．

$$V_i^{(t)} = V_i^{(t)} - V_{th} \tag{7}$$

SNN はハードウェア実装の観点からも優れた特性を有している．まず，スパイクが 2 値であるため，乗算器が不要になることが挙げられる．また，SNN はスパイクの到着により演算を開始させる「イベントドリブン型」アルゴリズムであり，低消費電力特性に優れた非同期回路実装との親和性が高い．例えば SNN アクセラレータの 1 つである IBM の TrueNorth [2] は，54 億個ものトランジスタを用いていながら，非同期回路を採用することで，消費電力を 70mW 程度に抑えている．

	ANN	SNN
ニューロンモデル	形式ニューロン	LIF
時間の概念	なし	あり
ニューロン間で伝達されるデータ	単精度浮動小数点数 8bit 整数 等	二値のスパイク
データの表現	アクティベーションの値で表現される	スパイクの発火確率で表現される
演算コスト	ニューロン間の接続数に依存	イベントドリブン (発火スパイク数に依存)
実行ハードウェア	CPU GPU TPU	ニューロモフィック ハードウェア (IBM TrueNorth 等)

表 1 ANN と SNN の比較

2 値のスパイクを用いた情報伝達は SNN のエネルギ効率を大幅に向上させる一方で，勾配計算が必要なバックプロパゲーションによる学習を困難にしている．生体脳の基本的な学習アルゴリズムの 1 つであると考えられている，スパイク間の位相差に応じてシナプス結合重みを変化させる STDP 則を利用した研究も存在するが，その応用は MNIST 等の簡易タスクに限定されており，現代の DNN が実現しているような非常に複雑なタスクをこなすことは依然として困難である [8]．そこで，ANN で学習した重みを SNN 用に変換し，推論のみを SNN で行う方法が提案された [4]．この方法では，ANN の各レイヤで ReLU の出力アクティベーションがスパイク発火確率と比例する様に重みを正規化することで，ANN で学習した重みを SNN 用に変換し，推論のみを SNN で実行する．具体的には ANN における各レイヤの重みとバイアスを以下のように正規化する．

$$\boldsymbol{w}_l' = \frac{\lambda_{l-1}}{\lambda_l}\boldsymbol{w}_l, \quad \boldsymbol{b}_l' = \frac{1}{\lambda_l}\boldsymbol{b}_l \tag{8}$$

ここで，λ_l は学習用データで推論を行った際の第 l 層のアクティベーションの値から選ぶ．λ_l の選び方の一つとして，アクティベーションの最大値とする方法もあるが，外れ値を選んでしまう可能性があり，その場合，変換後の SNN の推論収束までに必要な時間が長くなってしまう．そこで，ロバスト性を高めるため，最大値ではなく，全体の 99% の値や 99.9% の値を選ぶほうが良いことが知られている [4].

前述のように ANN から変換された SNN は，スパイクの発火確率によって値を表現しているため，スパイク発火を観測する期間と精度の間にはトレードオフの関係が存在する．つまり，十分な期間，スパイク発火を観測できれば，発火確率は高精度に推定できるが，エネルギ・レイテンシが増大してしまい，逆に，エネルギ・レイテンシを削減するためにスパイク発火の観測時間を短くしてしまうと，今度は推論精度が低下してしまう．そこで，本研究では，スパイク発火確率の推定にベイズ法を取り入れることで，このトレードオフを改善させる．

3. 提案手法

3.1 ベイジアンフュージョン

SNN におけるスパイク $\Theta^{(t)}$ はベルヌーイ分布に従うと仮定でき，その確率密度関数は次のベルヌーイ分布でモデル化出来る．

$$P(\Theta_j^{(t)}|\mu_j) = \mu_j^{\Theta_j^{(t)}}(1-\mu_j)^{1-\Theta_j^{(t)}} \tag{9}$$

ここで，μ_j はニューロン j の発火確率である． スパイク発火確率を求めることは，N ステップのスパイク発火系列 $\boldsymbol{\Theta}_j = (\Theta_j^{(1)}, \Theta_j^{(2)}, \cdots, \Theta_j^{(N)})$ が観測されたときに，それを生成する尤もらしいベルヌーイ分布のパラメータを求めることと等価であり，μ_j の最尤推定値は以下の式で与えられる．

$$\mu_j = \frac{M_j}{N} \tag{10}$$

ここで，$M_j = \sum_{t=1}^{N} \Theta_j^{(t)}$ である．

しかし，観測されるスパイクの発火確率には，何かしらの偏りがあるはずで，それをあらかじめ予測し，観測されたスパイク系列と統合することで，推定精度を高めることができることが予想される．本研究では，発火確率の推定に発火確率の偏りの情報を事前情報として取り込めるよう，ベイズ推定を導入する．ベイズ推定では，求めたいパラメータ μ_j が何らかの確率分布（事前分布）に従う確率変数であるとみなして，スパイク発火系列を観測した後に，パラメータが取るであろう値の確率分布（事後分布）を推定する方法である．発火確率の偏りを事前分布として表現することで，スパイク発火系列の観測長を減らすことに伴う情報劣化を補うことができる．

本研究では， 事前分布としてベータ分布を採用する．ベータ分布は α，β の二つのパラメータを持つ確率分布であり，その確率密度関数は次式で与えられる．

$$P(\mu_j|\alpha_j, \beta_j) = \frac{\mu_j^{\alpha_j-1}(1-\mu_j)^{\beta_j-1}}{B(\alpha_j, \beta_j)} \tag{11}$$

ここで，$B(\cdot,\cdot)$ は次式で定義されるベータ関数である．

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \tag{12}$$

N ステップのスパイク発火系列を観測した時の，スパイク発火確率 μ_j の事後分布は，式 (9) 及び (11) から以下のベータ分布に従うことが知られている．

$$P(\mu_j|\boldsymbol{\Theta}_j) = \frac{\mu_j^{\alpha_j+M_j-1}(1-\mu_j)^{\beta_j+N-M_j-1}}{B(\alpha_j+M_j, \beta_j+N-M_j)} \tag{13}$$

発火確率 μ_j は式 (13) の期待値である次式で推定する．

$$\mu_j^{\mathrm{EAP}} = \frac{\alpha_j + M_j}{\alpha_j + \beta_j + N} \tag{14}$$

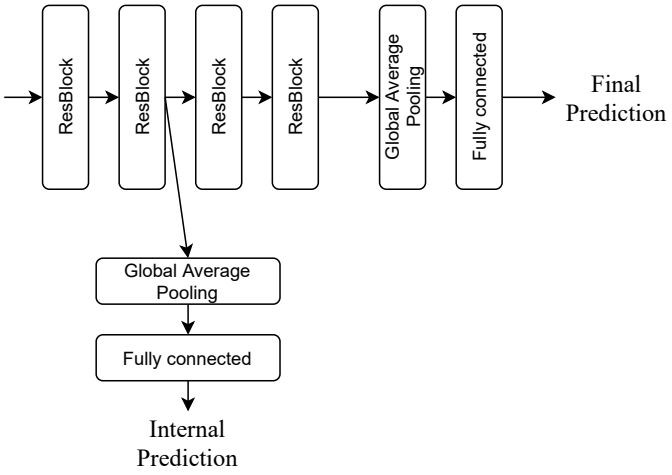

図 2 中間出力層のあるネットワーク

3.2 事前分布の構築方法

ベイジアンフュージョンの効果は事前分布の設計に依存しており，適切な事前分布を設定できれば，収束速度の改善に非常に効果的であるが，不適切な事前分布を選択してしまうと，収束を妨げてしまうことになりかねない．そこで，本研究では，ニューラルネットワークの浅い階層に位置するレイヤのアクティベーションから最終層のアクティベーションを予測させ，これを用いて，最終層のスパイク発火確率に対する事前分布を構築する [9]．具体的には，図 2 に示すように，ネットワークに中間出力層（Internal Prediction; IP）を加え，最終出力層（Final Prediction; FP）と同様に入力画像の属するクラス分類確率を予測できるよう学習させる．事前分布は，中間出力層のスパイク発火確率を平均に持つように設定する．中間出力層のニューロン j の発火確率を $\hat{\mu}_j$ とすれば，α_j，β_j は以下の関係を満たすように選択する．

$$\frac{\alpha_j}{\alpha_j + \beta_j} = \hat{\mu}_j \tag{15}$$

さらに，事前分布の影響の大きさを調整するために重みづけを行う．ここで，重みづけパラメータ P を次のように定義する．

$$\alpha_j + \beta_j = P \tag{16}$$

つまり，事前分布のベータ分布のパラメータ α_j, β_j は式 (15)，(16) より，次のようになる．

$$\alpha_j = P\hat{\mu}_j \tag{17}$$

$$\beta_j = P(1-\hat{\mu}_j) \tag{18}$$

この手法では，単一のネットワークに中間出力層を追加するだけで良く，オーバーヘッドはその中間出力層の

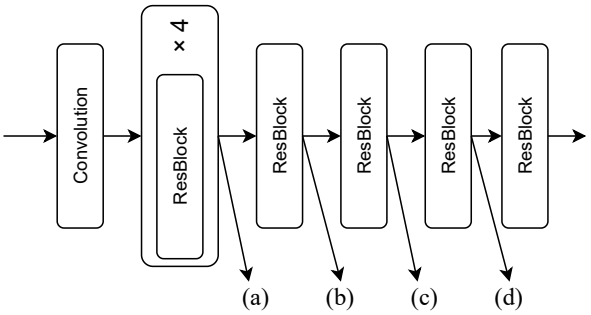

図 3 実験に用いるモデル

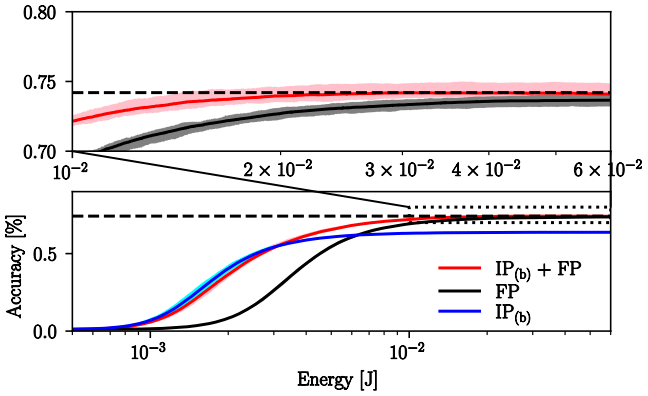

図 4 エネルギと精度の関係

Global Average Pooling と全結合層分だけとなり，それほど大きくはない．

4. 実験

4.1 実験設定

本手法の有効性の検証のため，PyTorch を用いたシミュレーション実験を行う．

対象データセットは CIFAR100[6] とする．CIFAR100 は，動植物や乗り物など 100 クラスの画像クラス分類データセットであり，50,000 枚の学習用画像と 10,000 枚の検証用画像から構成される．

ネットワークモデルは ResNet-18 をベースとした図 3 に示すモデルである．図中の (a) から (d) のそれぞれに中間出力層（$IP_{(a)}$，$IP_{(b)}$，$IP_{(c)}$，$IP_{(d)}$）として，GlobalAveragePooling と全結合層を入れることで中間出力を取得できるようにし，それぞれの中間出力と最終出力をフュージョンした場合の消費エネルギと精度の関係を評価する．

ネットワークは ANN で学習を行い，SNN 用に重みを変換して，SNN のシミュレーションによる推論を行う．

4.2 ANN の学習

ANN を学習するとき，重みの初期値には，He の初期値を利用している [10]．入力画像はチャンネル毎に平均 $(0.5071, 0.4865, 0.4409)$，標準偏差 $(0.2673, 0.2564, 0.2762)$ で正規化を行った．また，パディングを周囲 4 ピクセルずつ挿入した後 32×32 をランダムな切り抜き，確率 0.5 で左右反転，最大 15 度までのランダムな回転のオーグメンテーションをしている．学習エポック数は，120 エポックとした．

ANN で学習を行うとき，ロスの計算は式 (19) で示されるように行う．式 (19) 中の $\tau(e)$ はロスの重みづけ係数であり，0.01 から C まで線形に増加するものとする．第 e エポックにおける $\tau(e)$ は式 (20) で示される．ここで，C はその中間出力までを小さなネットワークとして考えたとき，その小さなネットワークの FLOPS が，最終出力までのネットワーク全体の FLOPS に対する割合である．

$$loss = loss_{final} + \tau(e) loss_{internal} \tag{19}$$

$$\tau(e) = 0.01 + \frac{e}{120}(C - 0.01) \tag{20}$$

$$C = \frac{\text{中間出力までの FLOPS}}{\text{最終出力までの FLOPS}} \tag{21}$$

4.3 SNN への変換と SNN による推論

SNN 用に重みを変換するとき，重みの正規化に用いるパラメータ λ は，学習用データからランダムに選んだ 5000 枚の画像を入力としたときのアクティベーションの値のうち，99.9% の値とする．

SNN のシミュレーションは，シミュレーション時間を離散時間とし，各タイムステップ毎に膜電位の更新とリセット，スパイクの発火が行われるものとする．本実験では，膜電位はすべて 0 で初期化し，スパイク発火の閾値は 1，シミュレーション時間は 3000 タイムステップとし，膜電位のリセットは減算によるリセットを用いる．また，エネルギの計算には，[11] で示されている，1 スパイクあたり 26 pJ を用いる．

4.4 実験結果

図 4 の緑の実線は，中間出力層 $IP_{(b)}$ のスパイク発火確率をもとに式 (15) に従って事前分布のパラメータである α_j，β_j を設定し，最終出力層 FP の発火確率を式 (13) に従って EAP 推定したときの推論エネルギと推論精度の関係を示している．比較として，中間出力層を取り除き最終出力層のみを用いた場合のエネルギ・推論精度の関係を青の実線で示す．また，赤の実線は ANN で達成された分類精度（74.2%）を示している．前述のように，ANN では連続値で表現されているアクティベーションを，SNN ではスパイク発火確率として表現している．従って，適切な正規化手順を経て SNN に変換されたネットワークは，シミュレーションに十分な時間を掛けられれば，ANN と遜色ない性能を発揮するはずであり，図 4 を見てもシミュレーション時間を十分長くとれば提案手法（緑）及び既存手法（青）は両者とも ANN の推論精度である 74.2%に漸近している．一方で，ベイジアンフュージョンを用いた提案手法

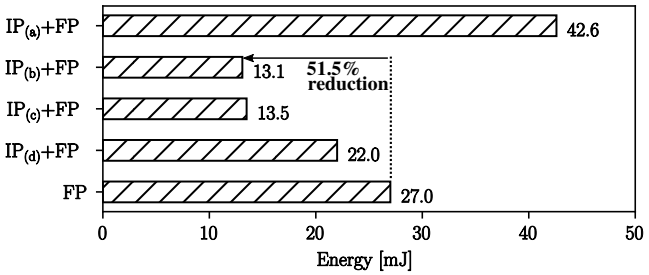

図 5 73.2%を達成するのに必要なエネルギ比較

の推論精度はエネルギが 1mJ を超えたあたりから急峻に立ち上がり始めているのに対して，既存手法では 2mJ を超えないと精度の立上りが見られていない．提案手法では中間分類層を付加した分だけネットワーク規模が増加しているが，それにも関わらず同等の推論精度を達成するために必要なエネルギを削減で来ていることが確認できる．また，後段の ResBlock3 段が冗長であるか確認するために，図 3 における $IP_{(b)}$ までを含むネットワークを用いて画像分類したときのエネルギ・精度の関係を橙色の実線で示す．この小規模ネットワークでは，提案手法よりも推論精度の立上りが早いが，およそ 60%程度で頭打ちとなっており，ANN の推論精度には到達できていない．このことから複雑な画像の推論には深いレイヤの高度な抽象化能力が必要不可欠であること，提案手法により浅いレイヤで早期に得られる特徴量と，深いレイヤの高精度な特徴量を統合出来ていることが確認できる．

次に，中間出力層を挿入すべき最適なレイヤを探索するべく，図 3 で示される (a) から (d) までのそれぞれに中間出力層を挿入し，同様の手順でエネルギと推論精度の関係をシミュレーションにより求めた．ANN の推論精度である 74.2%から 1%低い 73.2%の推論精度を達成するために必要なエネルギを図 5 に示す．図からわかる通り，(b) の位置から得られる中間出力を元にベイジアンフュージョンを行った場合が，73.2% を達成するエネルギが最も小さく 13.1mJ となっている．これは，最終出力だけの SNN の結果 27.0mJ と比較して，51.5% のエネルギが削減されている．一方で，(a) の位置から得られる中間出力を元にベイジアンフュージョンを行った場合，同等の精度を達成するために必要なエネルギは 42.6mJ と増加してしまっている．これは，(a) の位置から得られる中間出力の推論精度が低く，発火確率の推定に悪影響を与えたため，推論精度の収束が遅くなったためである．

5. おわりに

ネットワークの浅い層におけるアクティベーションから最終層の発火確率を予測し，これを実際に観測された発火事象とベイジアンフュージョンすることで SNN の推論に必要なエネルギの削減の手法を提案した．

提案手法の有効性を評価すべく，ResNet-18 を PyTorch で実装し，Cifar-100 を対象に画像分類タスクを学習させた．ANN の推論精度から 1%劣化した 73.2%の分類精度を達成するために必要なエネルギを比較したところ，単純な SNN では 27.0 mJ であった推論エネルギを 13.1 mJ にまで削減でき，比率にして 51.5%のエネルギ削減を達成した．

謝辞

本研究は，JST，さきがけ，JP-MJPR18M1，JSPS 科研費 21H03409 の支援を受けたものである．

参考文献

[1] M. Courbariaux and Y. Bengio, “BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1,” *CoRR*, vol. abs/1602.02830, 2016.

[2] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam, B. Taba, M. Beakes, B. Brezzo, J. B. Kuang, R. Manohar, W. P. Risk, B. Jackson, and D. S. Modha, “TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, 2015.

[3] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, “Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing,” in *IJCNN2015*, pp. 1–8.

[4] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, “Conversion of continuous-valued deep networks to efficient event-driven networks for image classification,” *Frontiers in Neuroscience*, vol. 11, p. 682, 2017.

[5] S. Kim, S. Park, B. Na, and S. Yoon, “Spiking-YOLO: Spiking Neural Network for Energy-Efficient Object Detection,” in *AAAI2020*, pp. 11 270–11 277.

[6] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-100 (canadian institute for advanced research).” [Online]. Available: http://www.cs.toronto.edu/ kriz/cifar.html

[7] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *ICML2015*, pp. 448–456.

[8] P. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Frontiers in Computational Neuroscience*, vol. 9, p. 99, 2015.

[9] Y. Kaya, S. Hong, and T. Dumitras, “Shallow-Deep Networks: Understanding and Mitigating Network Overthinking,” in *ICML2019*, pp. 3301–3310.

[10] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *ICML2010*, pp. 249–256.

[11] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science*, vol. 345, no. 6197, pp. 668–673, 2014.