

# BSP SoftHebb: A Learning Algorithm with Brain-Inspired Synaptic Pruning

日大生産工 ○濱田 圭惟 日大生産工 山内 ゆかり

## 1. まえがき

深層学習で用いられる誤差逆伝播法[1]は、ニューラルネットワークの出力と正解ラベルとの損失を最小化することで高い精度を実現している学習手法である。しかし、計算コストが非常に高く、モデル学習時に誤差を逆伝播する点において生物学的妥当性が欠如している。近年、これらの問題を軽減する誤差逆伝播法の近似手法が研究されてきたが、精度を大きく低下させることが多かった。そこで、精度を犠牲にすることなく、計算効率の高さと生物学的妥当性を兼ね備えた新たな学習手法が必要とされている。Timoleon Moraitisらは、Soft Winner-Take-All (Soft WTA)[2]と最新のヘップ学習則に基づいた研究[3]を重ね、SoftHebb[4]と呼ばれる多層ネットワークアルゴリズムを提案した。このアルゴリズムは、教師、フィードバック、ターゲットを一切使用せずに隠れ層の学習を可能にし、誤差逆伝播法と同等の画像分類精度を記録した。また、逆伝播信号を用いないため、重みの転置、非局所性、更新ロック問題、グローバル損失関数といった誤差逆伝播法の5つの制約からの回避にも成功した。しかし、ネットワークを深層化する際、各層のチャンネル数を制御する幅係数の設定によって特徴マップ数が急増し、結果としてモデルサイズが大きくなってしまう。そのため、単純な深層化によって性能向上を図ることが難しいという問題がある。

本研究では、ネットワークの軽量化と深層化による精度向上を目的として、一定期間にわたって小さな重みを動的に削除するシナプス刈り込みアルゴリズム Brain-Inspired Synaptic Pruning(BSP)[5]を導入したSoftHebbを提案する。実験では、画像分類タスクで提案手法とSoftHebbを比較し、精度、実行時間、有効な重みの数について報告する。

## 2. 従来研究

### 2.1 SoftHebb

SoftHebb[3][4]は、softmaxを用いたSoft Winner-Take-All(Soft WTA)[2]と独自のヘップ学習則（可塑性ルール）を組み合わせること

で、教師なしであってもクロスエントロピーを最小化し、画像分類において高い精度を実現している。SoftHebbのネットワークは、入力線形加重総和をSoft WTAにより競合した結果を出力とするブロックの積層から構成される。

#### 2.1.1 Soft Winner-Take-All(Soft WTA)

Soft WTAとは、Winner-Take-All(WTA) [6]と呼ばれる最も強く活性化されたニューロン活動を強調し、それ以外を抑制するというニューロンの活性を競う仕組みの1つである。このニューロンの競合は側抑制から生じ、哺乳類の大脳皮質全体に見られるため、生物学的に妥当なものであるとされている。WTAは大きく分けて2種類に分類され、勝者のニューロンのみが強調され他は完全に抑制されるHard WTAと、勝者以外のニューロンも完全には抑制されないSoft WTAがある。SoftHebbでは、後者のSoft WTAによるニューロンの競合と独自のヘップ学習則を組み合わせることによって教師ラベルがない状態でも有用な特徴を学習することを可能にしている。ここで、 $K$ 個のニューロンが存在する層においてSoft WTAによって競合を実現するとき、出力 $y_k$ は式(1)のように定義される。

$$y_k = \frac{b^{u_k}}{\sum_{l=1}^K b^{u_l}} = \frac{e^{\frac{u_k}{\tau}}}{\sum_{l=1}^K e^{\frac{u_l}{\tau}}} \quad (1)$$

$u_k$ は $k$ 番目のニューロンへの重み付き入力の合計、 $y_k$ は他のニューロン $l$ との競合した後の出力である。このとき、指数関数の底 $b$ または温度パラメータ $\tau$ を変化させることで出力 $y_k$ に柔軟性を持たすことができる。温度パラメータ $\tau$ が1より小さいときは確率分布が鋭くなり、1より大きいときはなめらかになる。

#### 2.1.2 ヘップ学習則

SoftHebbのSoft WTAは、本質的には生成モデルであり識別モデルではないが、Moraitisらが導出した新たなヘップ学習則によって教師ラベルを用いずにモデルの確率分布と入力の分布のKLダイバージェンス、および教師ラベルからのクロスエントロピーを最小化する。すなわち、

生成モデルと識別モデルの両面を最適化する。ここで、 $k$ 番目のニューロンにおける入力 $i$ の線形加重和を $u_k$ 、Soft WTA出力の非線形成分 $y_k$ とすると、ヘップ学習則は式(2)のように定義される。

$$\Delta w_{ik}^{(SoftHebb)} = \eta \cdot y_k \cdot (x_i - u_k \cdot w_{ik}) \quad (2)$$

このとき、 $w_{ik}$ は入力 $i$ からニューロン $k$ への重み、 $\eta$ は学習率である。式(2)からわかるように、導出された学習則に使用されるすべての変数は入力側 $i$ と出力側 $k$ のみに依存しており、シナプスに局所的なものである。すなわち、誤差逆伝播法[1]のような遠方の層や他のシナプスからの信号を一切必要とせず、順伝播時に重みやバイアスの更新を行うことができる。

### 2.1.3 SoftHebbのアーキテクチャ

前述の通りSoftHebbはSoft WTAブロックの積層から構成される。ここで、ネットモデルがMultilayer Perceptron (MLP)[1]もしくはConvolutional Neural Network (CNN)[7]であるときの各BlockのアーキテクチャをFigure 1に示す。

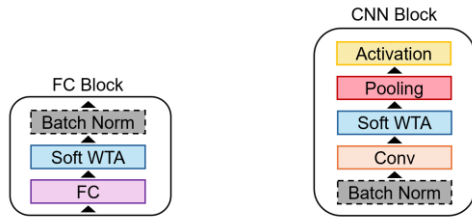


Figure 1 各 Block のアーキテクチャ

FC Block は全結合層と Soft WTA から構成され、必要に応じてバッチ正規化を追加する。一方、CNN Block は畳み込み層、Soft WTA、プーリング層、活性化関数から構成され、同様にバッチ正規化をオプションとして導入できる。

SoftHebb は、これらの Block を積層した後、その出力特徴に対して誤差逆伝播法で学習される線形分類器を付加する構造を取る。この線形分類器は、教師なし学習で得られた特徴表現の有用性を評価するために用いられるものであり、教師ラベルはこの分類器の訓練時のみに使用される。そのため、各 Block 内の重みおよびバイアスは分類器の学習過程では更新されない。

## 2.2 Brain-Inspired Synaptic Pruning(BSP)

BSP[5]は、脳の発達過程で使用されるシナプス規則に基づいて冗長な接続を削除するシナプス剪定アルゴリズムである。MLPネットワークの各重みの絶対値をニューロンの結合度とみな

し、継続的に結合度が弱い重みを各エポックで評価して剪定する。

### 2.2.1 剪定候補重みの選定

接続の重要度は、各重みの絶対値によって決定され、小さな重みは弱いシナプスとみなして剪定候補とする。剪定候補は、現在の反復での残存重みの数と剪定率によって動的に決定される。

ここで、隠れ層の初期重みの数を $N_{neu}$ 、訓練サンプル数を $N_{sam}$ とすると、剪定率 $p_r$ は式(3)のように定義される。

$$p_r = \frac{0.048 \times \log_e(N_{neu})}{\log_e(50 \times (N_{sam}/60000))^2} \quad (3)$$

このとき、 $N_{neu}$ の増加に伴って $p_r$ は増加し、 $N_{sam}$ の増加に伴って $p_r$ は減少する。

また、残存重みの数を $N_s$ とすると剪定候補重みの数 $N_c$ は式(4)のように定義される。

$$N_c = N_s \times p_r \quad (4)$$

### 2.2.2 剪定候補連続回数の計算

重みが剪定候補に含まれた連続回数を計算する。もし、重みが常に剪定候補に含まれ連続回数が大きい場合、その重みは重要ではないとされる。まれに使用される重みについては、監視を続ける。

### 2.2.3 重みの剪定

連続回数が閾値を超えた重みを剪定する。長期間小さな重みのみを剪定することで、削除される重みが冗長であることを保証でき、急激な精度低下を回避できる。剪定された重みは恒久的に除去され、その後の学習およびテストには影響を与えない。

ここで、初期ネットワークの重みの数 $N$ 、剪定された重みの数を $N_p$ とすると、連続回数の閾値 $p_c$ は式(5)のように定義される。

$$p_c = 2^{N/(N-N_p)} \times \frac{1244}{N_{neu} + 146} \times \left[ \left[ \log_e \left( 50 \times \frac{N_{sam}}{60000} \right) \right]^2 + 1 \right] \quad (5)$$

このとき、 $N_{neu}$ の増加に伴って $p_c$ は減少し、 $N_{sam}$ の増加に伴って $p_r$ は増加する。 $p_c$ は剪定された重み数に応じて指数関数的に変化し、過剰剪定を防ぐ。また、残存重みの数が少ない場合は、 $p_c$ は自動的に増加して剪定を制限する。

### 3. 提案手法

SoftHebbのCNNモデルを深層化する際には、各層のチャンネル数を制御する幅係数の設定によって特徴マップ数が増加し、モデルサイズが大きくなるため、深層化が困難になるという問題がある。MLPモデルの場合も、適度に冗長な表現を得るために隠れ層のノード数を多く設定することが多く、ノード数の増加に伴って重みも増加し、ネットワーク全体が肥大化して深層化が難しくなる。さらに、SoftHebbでは学習中の競合により、一部のノードが長期間出力にほとんど寄与しなくなり、不要なノードや重みが生じてしまうという問題も存在する。そこで本研究では、これらの問題の解決を図るため、一定期間にわたって小さな重みを不要なものとし、動的に剪定を行うSoftHebbを提案する。

#### 3.1 SoftHebb 向けの剪定率

SoftHebbは、Soft WTAを導入しており、隠れ層のノード数と重みの数が比較的多い。そのため、BSPアルゴリズムをそのままSoftHebbに適用すると、剪定率 $p_r$ と剪定候補重みの数 $N_c$ が増加し、さらに連続回数の閾値 $p_c$ が減少するため、過剰な剪定が生じ、正常な競合が阻害されて精度が低下してしまうと考えられる。そこで、提案手法では式(2)にスケーリング係数 $\gamma$ をかけ、剪定率を抑えたものを用いる。本研究では、 $\gamma = 0.1$ と仮設定をした。この値が最適であるかは不明なため、今後グリッドサーチを行う必要がある。

#### 3.2 CNN における重みの剪定

BSP アルゴリズムはMLP 向けのフレームワークあり、CNN において有効性は示されていない。そこで、CNN における重みの剪定手法を新たに設計した。

CNN では、重みが空間的に共有されるため、単純に重み単位で剪定を行うと特徴マップの一貫性が失われ、学習が不安定になる恐れがある。そこで、CNN ではカーネル単位での重み剪定を行う。まず、重要度をカーネル内の全重みの絶対値平均として計算をする。この重要度が一定期間にわたり小さいものを冗長なものとし、剪定する。この際に必要になる剪定率や剪定候補重みの数、連続回数の閾値は、それぞれの重みの数をカーネルの数に置き換えることで算出する。

### 4. 実験および検討

本研究では、画像分類タスクにおける分類精度、実行時間および有効な重みの数を従来手法

(SoftHebb)と提案手法(BSP SoftHebb)で比較した。実験にはMNISTデータセットを用い、訓練データを60000、テストデータを10000とした。また、ネットモデルをノード数2000の隠れ層1層を持つMLPとし、学習率 $\eta$ を0.02、温度パラメータ $\tau$ を0.05に設定して実験を行った。

SoftHebbとBSP SoftHebbの分類精度と実行時間の実験結果をTable 1に示す。

Table 1 分類精度と実行時間の比較

Model	Test Accuracy[%]		Run Time[s]	
SoftHebb	95.67	$\pm(0.0363)$	<u>6775.4</u>	$\pm(81.997)$
<b>BSP SoftHebb</b>	<u>95.69</u>	$\pm(0.1480)$	7680.0	$\pm(60.000)$

BSP SoftHebbは、テスト精度が0.02%向上した一方で、実行時間が904.6秒増加した。BSP SoftHebbでは重みの剪定を行ったにもかかわらずテスト精度が向上したこと、冗長な重みを効果的に剪定できたと考えられる。今回は剪定した重みをマスクするのみであったため、実行時間の短縮にはつながらなかった。実行時間が増加した要因としては、BSP処理に要する計算時間が増えたことが挙げられる。

次にSoftHebbとBSP SoftHebbの有効な重みの数についての実験結果をTable 2に示す。

Table 2 有効な重みの数の比較

Model	Weights	Network Compression[%]
SoftHebb	1588000	—
<b>BSP SoftHebb</b>	<u>1354673</u>	<u>14.69</u>

最終エポック時にBSP SoftHebbは、233327個の重み削減に成功し、14.69%のモデル圧縮を達成した。

SoftHebbとBSP SoftHebbの分類精度の推移をFigure 2に示す。

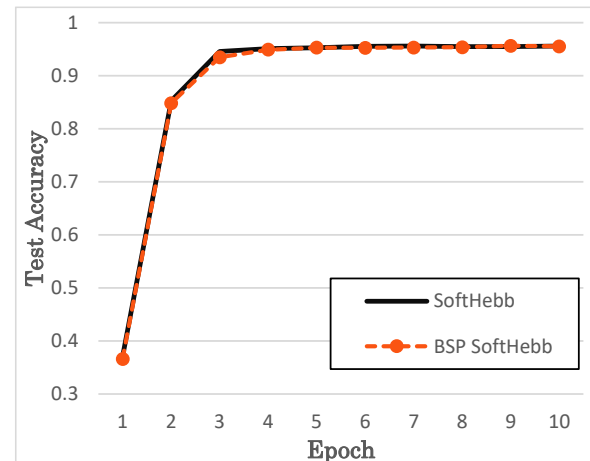


Figure 2 分類精度の推移



BSP SoftHebbは、すべてのエポックにおいてSoftHebbと同程度の分類精度で推移しており、安定した学習が行えていると考えられる。

次にSoftHebbとBSP SoftHebbの実行時間の推移をFigure 3に示す。

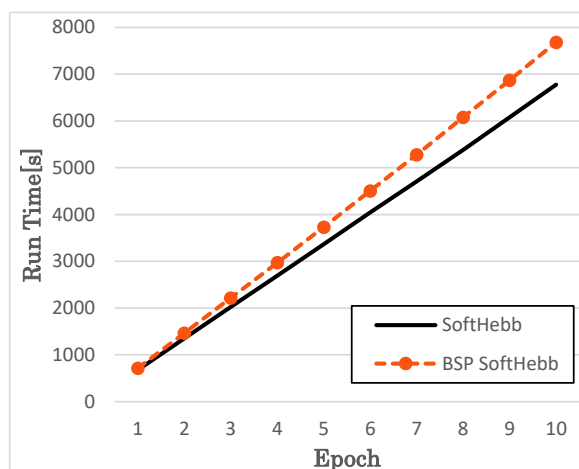


Figure 3 実行時間の推移

SoftHebb とBSP SoftHebbの実行時間はいずれもエポック数に対して線形的に増加した。しかし、BSP SoftHebbの傾きの方が大きく、エポック数の増加に伴い両者の実行時間の差が拡大した。

さらにSoftHebbとBSP SoftHebbの有効な重みの数の推移をFigure 4に示す。

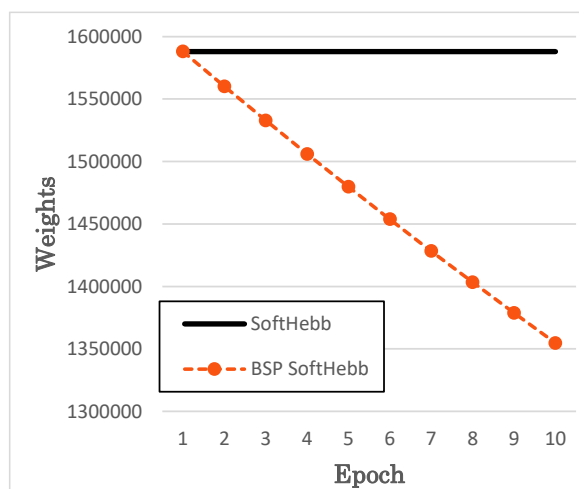


Figure 4 有効な重みの数の推移

BSP SoftHebbの有効な重みの数は、エポック数ごとに一定の割合で減少した。この傾向はエポック数を増やした場合でも継続する可能性があり、依然として剪定可能な重みが存在することが示唆される。したがって、より適切なスケール係数の設定が必要であると考えられる。

## 5. まとめ

本研究では、ネットワークの軽量化によって深層化を容易にし、その結果として精度向上を

実現できるという仮説のもと、動的な重み剪定を適用したSoftHebb手法を提案した。実験結果より、冗長な重みを効果的に剪定することで精度を向上できることを隠れ層1層のMLPで確認した。一方で、剪定した重みを単純にマスクするだけでは実行時間の短縮にはつながらないことも確認した。今後は実行時間の短縮を実現する手法の検討を進めつつ、CNNや多層モデルにおいても冗長な重み剪定が精度の向上につながることを確認し、さらなる深層化と精度向上を目指す。

また、本研究では仮設定をしたスケール係数を剪定率に用いたが、最適な値を明確にすることができなかった。そのため今後は、スケール係数による精度・実行時間・有効な重みの数の変化を調査し、最適な値を明確にすることを目指す。

## 参考文献

- 1) David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams., “Learning representations by back-propagating errors”, Nature Vol.323 (1986).
- 2) Shih-Chii Liu., “A Winner-Take-All Circuit with Controllable Soft Max Property”, Advances in Neural Information Processing Systems 12 (NIPS 1999).
- 3) Timoleon Moraitis, Dmitry Toichkin, Adrien Journé, Yansong Chua and Qinghai Guo., “SoftHebb: Bayesian inference in unsupervised Hebbian soft winner-take-all networks”, Neuromorph. Comput. Eng. 2 (2022) p.044017.
- 4) Adrien Journé, Hector Garcia Rodriguez, Qinghai Guo and Timoleon Moraitis., “Hebbian Deep Learning Without Feedback”, The Eleventh International Conference on Learning Representations (2023).
- 5) Feifei Zhao and Yi Zeng., “Dynamically Optimizing Network Structure Based on Synaptic Pruning in the Brain”, Frontiers in Systems Neuroscience. Vol.15(2021).
- 6) Y Fang, MA Cohen and TG Kincaid., “Dynamics of a Winner-Take-All Neural Network”, Neural Networks. Vol.9. No.7 pp.1141-1154(1996).
- 7) Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton., “ImageNet Classification with Deep Convolutional Neural Networks”, Advances in Neural Information Processing Systems 25 (NIPS 2012).