事前学習を考慮したシングルショット非構造枝刈り手法に関する研究

TP22011 小濱大和　　指導教授：山下隆義

1. はじめに

十分な事前学習を行った大規模なニューラルネットワークモデルは，様々な下流タスクで高い性能を発揮する．その一方で，サイズが大きいモデルは，下流タスクにファインチューニングする際に大きな計算コストを必要とする．そのため，ファインチューニングする前に，ニューラルネットワークのモデルサイズを圧縮することが求められている．

本研究では，事前学習時済みモデルに含まれる重みの重要度を評価し，重要度の低い重みをファインチューニングする前に削除（枝刈り）することを考える．先行研究 [1, 2, 3] で提案されている重みの評価基準は，事前学習時の学習ダイナミクスを考慮して設計されている．そのため，事前学習によって獲得した重みの近傍に，下流タスクで高い性能を発揮するようなローカルミニマムが存在することを考慮していない．これは，事前学習済みモデルの下流タスクに対する収束を妨げる可能性がある．そこで，本研究では，事前学習済みモデルの特徴表現を，枝刈り前後で維持することを目的とした重みの評価基準を提案する．これにより，事前学習によって獲得した特徴表現が，下流タスクに対するファインチューニングへ活用され，枝刈りしたモデルが下流タスクに対してより高い性能を発揮することを期待する．

2. 先行研究

ニューラルネットワークの枝刈りは，重み行列 $\boldsymbol{W}$ を要素単位で削除する非構造的枝刈りと，行，または列単位で枝刈りを行う構造的枝刈りに大別される．一般的に，非構造的枝刈りは，構造的枝刈りと比較してモデルサイズを圧縮しやすい．非構造的枝刈りの先行研究として，ニューラルネットワークを事前学習する前に，冗長な重みを特定する評価基準が提案されている [1, 2, 3] ．また，これらの評価基準は学習と枝刈りを繰り返さないため，シングルショット枝刈り手法に分類される．本節では，それらの学習を必要としない枝刈り手法について紹介し，それらが事前学習済みモデルに対して十分に有効でない理由について述べる．

2.1. Single-Shot Network Pruning

Single-Shot Network Pruning (SNIP) [1] は，学習を必要としない枝刈りとして提案された手法である．SNIP は，目的関数 $\mathcal{L}$ に対する重み $w_i \in \boldsymbol{W}$ の感度を評価するために，評価基準 $S_{\mathrm{SNIP}}(w_i)$ を式 (1) のように導出する．

$$\begin{aligned} S_{\mathrm{SNIP}}(w_i) &= \Delta\mathcal{L} = |\mathcal{L}(w_i) - \mathcal{L}(w_i + \delta)| \\ &= \left| \frac{\partial \mathcal{L}}{\partial w_i} w_i \right| \end{aligned} \tag{1}$$

そして，評価値の低い重みを優先して枝刈りする．

2.2. Gradient Signal Preservation

Gradient Signal Preservation (GraSP) [2] は，枝刈りの前後で勾配フローの維持，または増加を目的として提案された手法である．ここで，勾配フローとはモデル全体の学習中の変化量を表しており，ニューラルネットワークの全体の重み $\boldsymbol{W}$ における勾配フローを式 (2) のように導出する．

$$\begin{aligned} \mathcal{L}_{\mathrm{Flow}} &= \Delta\mathcal{L}(\boldsymbol{W} + \boldsymbol{\delta}) - \Delta\mathcal{L}(\boldsymbol{W}) \\ &= 2\boldsymbol{\delta}^{\mathrm{T}} \nabla^2 \mathcal{L}(\boldsymbol{W}) \nabla \mathcal{L}(\boldsymbol{W}) + \mathcal{O}(||\boldsymbol{\delta}||_2^2) \end{aligned} \tag{2}$$

GraSP は枝刈り後の勾配フローの維持，または増加を期待することから，重み w_i に対する評価基準 $S_{\mathrm{GraSP}}(w_i)$ は式 (3) のように定義される．

$$S_{\mathrm{GraSP}}(w_i) = -[\nabla^2 \mathcal{L}(\boldsymbol{W}) \nabla \mathcal{L}(\boldsymbol{W})]_i w_i \tag{3}$$

2.3. Synaptic Flow

Synaptic Flow (SynFlow) [3] は，SNIP や GraSP で引き起こされるレイヤー崩壊と呼ばれる現象を回避するために提案された枝刈り手法である．ここで，レイヤー崩壊とは，高い枝刈り率を設定した場合に一部の層の重みが全て削除されてしまう現象である．式 (4) に示すように，シナプスの強度（出力特徴量）を増加させるような評価基準 $S_{\mathrm{SynFlow}}(w_i)$ を導入して，レイヤー崩壊を回避する．

$$S_{\mathrm{SynFlow}}(w_i) = \left| \frac{\partial \sum_k f(\mathbf{1}; |\boldsymbol{W}|)}{\partial |w_i|} |w_i| \right| \tag{4}$$

ここで，f はニューラルネットワーク，k は f の出力するロジット数を表す．

2.4. 従来手法の限界

SNIP は，テイラー展開した目的関数の 1 次の項までを考慮して評価基準を導出している．事前学習済みモデルを対象とする場合，一部の重みの学習は既に収束している可能性があるため，式 (5) のような状態が起こる．

$$\mathcal{L}(w_i + \delta) \approx \underbrace{\mathcal{L}(w_i)}_{\text{const}} + \underbrace{\frac{\partial \mathcal{L}}{\partial w_i} \delta w_i}_{0} + \frac{1}{2} \frac{\partial^2 \mathcal{L}}{\partial w_i^2} \delta w_i^2 + \mathcal{O}(||\delta w||^3) \tag{5}$$

$|\mathcal{L}(w_i) - \mathcal{L}(w_i + \delta)|$ より，初項が打ち消し合い，1 次の項が 0 となるため，収束している重みについての評価値が 0 となり，事前学習によって獲得した有用な知識が失われる．

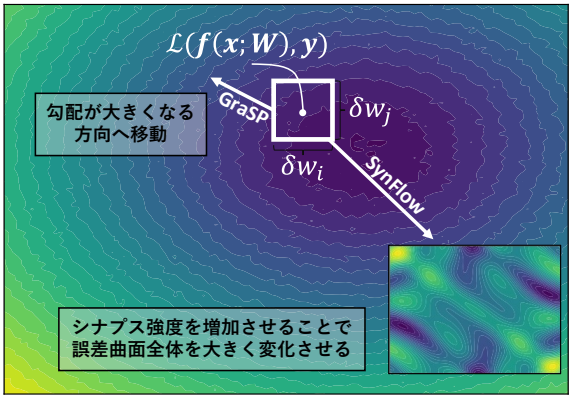

図 1：事前学習済みモデルの誤差曲面と枝刈りによる影響

次に，GraSP と SynFlow が，事前学習済みモデルの誤差曲面に与える影響を図 1 に示す．白い矢印は，枝刈りしたモデルの出力誤差の変動を表す．GraSP の評価基準は，枝刈りしたモデルの勾配フローを増加させる．そして，枝刈り後のニューラルネットワークに対して積極的な学習を期待するため，近傍に存在するローカルミニマムに逆行するような枝刈りが行われる．また，SynFlow の評価基準は，ニューラルネットワークのシナプス強度を増加させることを目的としている．そのため，枝刈り後に出力特徴量が活性化しやすい状態を期待する．しかし，同様の枝刈りを事前学習済みモデルに適用すると，事前学習によって獲得した特徴表現を大きく変化させることになる．つまり，同心円状に広がる誤差曲面の形状は大きく歪む．これらの理由から，GraSP と SynFlow による枝刈りは事前学習済みモデルの学習を妨げる可能性がある．

3. 提案手法

本研究では，枝刈りの前後でニューラルネットワークの特徴表現を変化させないような重みの評価基準の提案を目指す．これにより，枝刈りした事前学習済みモデルが，より高い性能を発揮するようなローカルミニマムへ収束することを期待する．

アルゴリズム 1 事前学習を考慮した枝刈り

Require: 枝刈り率 k，学習データ $\mathcal{D}$，重み $\boldsymbol{W}$

$\mathcal{D}_b = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{b} \sim \mathcal{D}$ ▷ 学習データのミニバッチ
for w_i in $\boldsymbol{W}$ **do**
  $f(\mathcal{D}_b)$ を計算して $\boldsymbol{F}$ を取得
  $s_i \leftarrow \left| \frac{\partial \sum_{l=1}^{L} |\boldsymbol{F}^l|_1}{\partial w_i} w_i \right|$ ▷ 各パラメータを評価
end for
全ての s 集合 $\boldsymbol{S}$ の k%目のパーセンタイルを κ として算出
$\boldsymbol{M} \leftarrow \boldsymbol{S} < \kappa$ ▷ 評価値の低いパラメータを削除
$\boldsymbol{M} \odot \boldsymbol{W}$ を $\mathcal{D}$ を用いて収束するまで学習

3.1. 事前学習を考慮した評価基準

枝刈りした際に，N 次元の特徴量 $\boldsymbol{F}$ の変化を抑制するような枝刈りを考える．ここで，$\boldsymbol{F}$ に対する重み w_i の感度を評価するために，特徴量の各要素のノルム $|\boldsymbol{F}|_1$ を式 (6) のように定義する．

$$|\boldsymbol{F}|_1 = \sum_{n=1}^{N} |\boldsymbol{F}_n| \tag{6}$$

ここで，$|\boldsymbol{F}|_1$ の維持を目的とする場合，枝刈りの前後で $|\boldsymbol{F}|_1$ の値に影響しない重みを特定する必要がある．ニューラルネットワークの l 層目の特徴量を $\boldsymbol{F}^l = f_l(\boldsymbol{F}^{l-1}; \boldsymbol{W}^l)$ とすると，w_i^l に摂動 δ を与えた際の $|\boldsymbol{F}^l|_1$ の変化量 $\Delta|\boldsymbol{F}^l|_1$ は式 (7) のように表される．

$$\begin{aligned}
\Delta|\boldsymbol{F}^l|_1 &= |\boldsymbol{F}^l|_1(w_i^l) - |\boldsymbol{F}^l|_1(\delta w_i^l) \\
&= |\boldsymbol{F}^l(w_i^l)|_1 - |\boldsymbol{F}^l|_1(w_i^l) - \frac{\partial |\boldsymbol{F}^l|_1}{\partial w_i^l}(\delta w_i^l - w_i^l) \\
&\approx -\frac{\partial |\boldsymbol{F}^l|_1}{\partial w_i^l} w_i^l
\end{aligned} \tag{7}$$

このとき，$\boldsymbol{F}^l$ のノルム $|\boldsymbol{F}^l|_1$ に与える w_i^l の影響度は，

$$|\Delta \boldsymbol{F}^l| = \left| \frac{\partial |\boldsymbol{F}^l|_1}{\partial w_i^l} w_i^l \right| \tag{8}$$

と表される．そして，全ての層について式 (8) を考慮することで，ある重み w_i に対する評価 $S(w_i)$ は式 (9) となる．

$$\begin{aligned}
S(w_i) &= \sum_{l=1}^{L} \left| \frac{\partial |\boldsymbol{F}^l|_1}{\partial w_i^l} w_i^l \right| \\
&= \left| \frac{\partial \sum_{l=1}^{L} |\boldsymbol{F}^l|_1}{\partial w_i} w_i \right|
\end{aligned} \tag{9}$$

ここで，式 (9) の 1 行目から 2 行目への式変形は，$|\boldsymbol{F}^l|_1$ は重み w_i^l に対して連続であり，$S(w_i)$ に対してもまた連続であることから，$\frac{\partial}{\partial w_i} \sum_{l=1}^{L} |\boldsymbol{F}^l|_1 = \sum_{l=1}^{L} \frac{\partial}{\partial w_i} |\boldsymbol{F}^l|_1$ が成り立つことを利用した．

3.2. 枝刈りアルゴリズム

アルゴリズム 1 に，式 (9) を用いた枝刈りのアルゴリズムを示す．提案手法は，事前学習済み重み $\boldsymbol{W}$ の全ての要素に対して，重みの評価を行い，各重みの重要度を算出する．そして，指定した枝刈り率を用いて閾値を算出し，閾値を下回る評価値を持つ重みを削除する．

4. 評価実験

提案手法による有効性を確認するために，事前学習済みモデルを枝刈りした際の分類精度を比較する．

4.1. 実験条件

ViT-B/16 を用いて分類精度の比較を行う．ImageNet-1k，ImageNet-21k，自己教師あり学習である self-Distillation with NO labels (DINO) （ImageNet-1k を使用）による事前学習を行い，下流タスクデータセットを用いてファインチューニングを行う．下流タスクデータセットとして，CIFAR-10，CIFAR-100，及び Tiny-ImageNet を使用する．

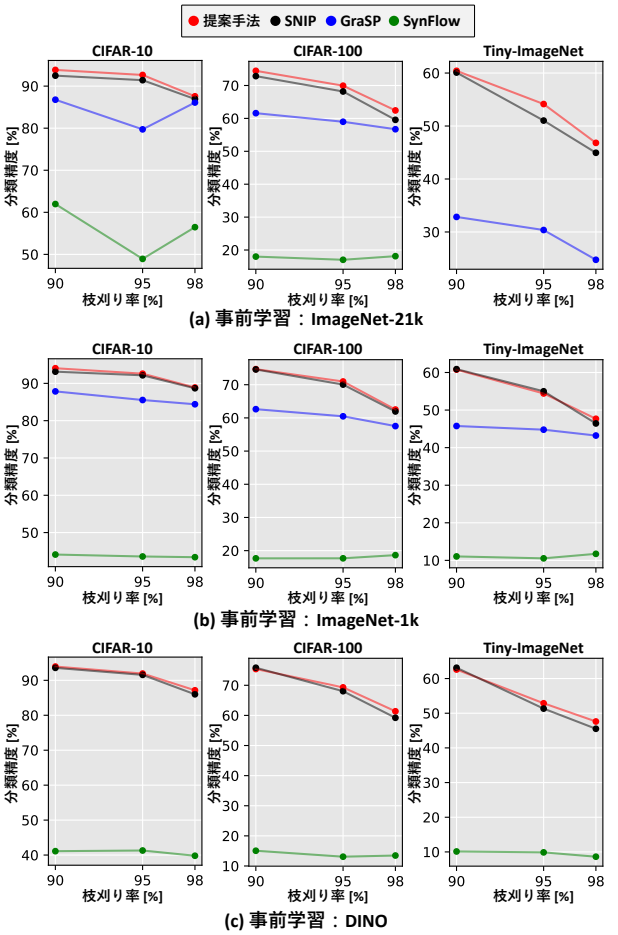

図 2：枝刈りした ViT-B/16 における分類精度の比較

4.2. 実験結果

図 2 に，枝刈りした各事前学習済みモデルの分類精度を示す．縦軸は分類精度，横軸は枝刈り率を表す．結果より，ほぼ全ての比較において，提案手法による枝刈りが最も高い分類精度であることを確認した．具体的には，ImageNet-21k で事前学習した場合に最も分類精度が改善している (図 2-a)．また，2 行目の ImageNet-1k を用いて事前学習した場合 (図 2-b) と，3 行目の ImageNet-1k を使用した DINO で事前学習した場合 (図 2-c) を比較すると，DINO による事前学習時において，より顕著に分類精度の向上を確認することができた．事前学習の規模に比例して精度が大きく改善していることから，提案手法は事前学習によって獲得した知識を下流タスクに対して活用できたと考えられる．

5. おわりに

本研究では，事前学習によって獲得した特徴表現を維持するような重みの評価基準を提案した．また，評価実験により，提案した重みの評価基準に基づいて重みを枝刈りすることで，枝刈り後の分類精度が向上することを確認した．今後は，物体検出タスクやセグメンテーションタスクなどにおいても同様の結果が得られるかの調査を行う．

参考文献

[1] N. Lee *et al.*, “SNIP: Single-shot Network Pruning based on Connection Sensitivity”, ICLR, 2019.

[2] C. Wang *et al.*, “Picking Winning Tickets Before Training by Preserving Gradient Flow”, ICLR, 2020.

[3] H. Tanaka *et al.*, “Pruning neural networks without any data by iteratively conserving synaptic flow”, NIPS, 2020.

研究業績

[1] 小濱大和 等, “Recurrent Neural Network における移動エントロピーを用いた枝刈り”, 画像の認識・理解シンポジウム, 2022.
(他 3 件)