

# 深層学習における学習序盤の損失変化に注目した効率的な次元数選択手法 A Method for Efficiently Selecting the Number of Dimension in Deep Learning Focusing on Change in Loss in the Early Phase of Learning

内田 青澄<sup>†, (a)</sup> 今川 孝久<sup>†</sup> 榎田 修一<sup>†</sup>  
Uchida Aoto Imagawa Takahisa Enokida Shuichi

## 1. はじめに

映像に基づく人物の行動認識実現のため、深層学習による骨格抽出を用いた認識手法が注目されている[1][2]。エッジデバイスにおいて人物の行動認識を実現することで、自動運転[3]や安全・監視システム[4]、ホームヘルスケア[5]などへの活用が期待される。本研究では、高いリアルタイム性と小さな推論コストを実現している Action Transformer[6]に注目する。しかし、Action Transformer のような軽量の行動認識モデルを使用する際も、エッジデバイスでの計算リソースは、人物の行動認識以外にも割く必要がある。したがって、使用する行動認識モデルの推論コストが占める割合は小さい方がよく、モデル規模として、より小さなモデルが期待される。

深層学習におけるハイパーパラメータの選択に関して、ハイパーパラメータ最適化[7]や Neural Architecture Search[8]と呼ばれる手法が知られている。一般に、深層学習におけるハイパーパラメータとは、学習率、バッチサイズ、エポック数、層の数などが含まれる。ハイパーパラメータ最適化の基本的な手法として知られるグリッドサーチやランダムサーチでは、各ハイパーパラメータに対して最後まで学習し、最終的な結果から最適なハイパーパラメータを選択する。したがって、一度の学習に大きなコストが必要な深層学習を対象とするハイパーパラメータ最適化や、Neural Architecture Search には、膨大な学習コストが必要となる。

本研究では、モデル規模に影響を与えるハイパーパラメータである埋め込み次元数に着目し、学習にかかるコストを抑えつつ、精度とモデル規模に基づく埋め込み次元数の自動選択を可能とする手法を提案する(図 1)。提案手法の一つ目の特徴は、対象となるハイパーパラメータをモデル規模に影響する埋め込み次元数に限定することである。

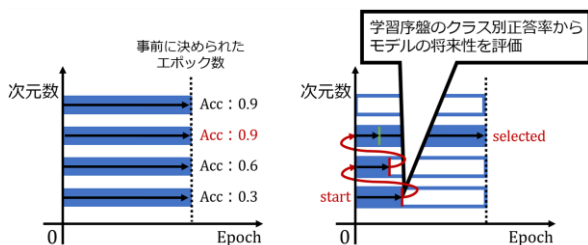


図 1 埋め込み次元数選択までに必要となるエポック数の例を示す。横軸がエポック数、縦軸が埋め込み次元数をあらわす。本図においては選択候補の次元数が四つある場合を示す。左図は全探索手法による例、右図は提案手法での例を示す。

さらに、低い次元数から順に処理することで、次元数選択にかかる学習コストと最終的に選択されるモデルの規模がそれぞれ抑制される。二つ目の特徴は、学習途中の損失変化に基づく評価タイミングにおいて、学習途中のモデルがもつ将来性を評価することである。将来性評価により、学習の収束を待たずに、必要な精度が得られないと評価された場合は、学習を打ち切る。結果、任意の埋め込み次元数候補における選択するか判断までの学習コストが抑制される。

本研究の貢献を以下に示す。

1. 深層学習において、認識精度、モデル規模に基づき、埋め込み次元数を効率よく選択する手法を提案する。
2. 提案手法について、Action Transformer を用い評価実験する。特に、次元数選択までにかかる学習コスト、埋め込み次元数および認識精度の三点に注目し、提案手法の有効性を確認する。

## 2. 埋め込み次元数の自動選択手法

本章では、深層学習における埋め込み、および埋め込み次元数の自動選択に関係する、ハイパーパラメータ最適化の手法について述べる。2.1 節では、深層学習における埋め込みについて詳述する。2.2 節では、深層学習におけるハイパーパラメータの選択を自動化するためのハイパーパラメータ最適化について概説する。

### 2.1 深層学習における埋め込み

深層学習における埋め込み(Embedding)とは、深層学習によるシステムが、現実世界のオブジェクトである入力データを、より詳細に解釈可能にするための処理である。具体的には、単語や画像を簡易な全結合層などでベクトルデータに変換することである[9][10]。現実世界のオブジェクトを適切に埋め込むことで、深層学習アルゴリズムは現実世界の複雑な情報を適切に扱うことができ、より高精度となることが期待される。単語の埋め込みは、主に自然言語処理に用いられている[11]。単語をベクトルとして埋め込むことで、深層学習アルゴリズムは、人間が直感的に理解している単語同士の類似度や意味の理解を深めることができる。画像に対する埋め込みの代表的なものとして、Transformer[12]を画像処理に応用した Vision Transformer 内における、バッチ分割後の画像に対する埋め込みが知られている。適切に画像を埋め込むことにより、人間が直感的に理解可能である抽象的な情報が数値化され、深層学習アルゴリズムの画像に対する理解が深まる。

適切な埋め込みを実現するには、重要となる要素がいくつか存在する。その一つが埋め込み次元数である。埋め込み次元数を高くすることで、埋め込み後の情報における表現力が高くなる。しかし、不必要な情報や埋め込み前のデータに含まれる情報量によって、効果的な埋め込み次元

<sup>†</sup> 九州工業大学 Kyushu Institute of Technology  
(a) uchida.aoto959@mail.kyutech.jp

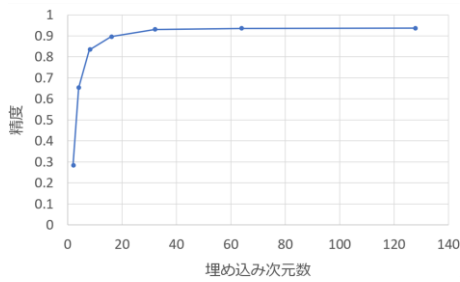


図 2 Action Transformer において埋め込み次元数を変更した際の、最後まで学習したモデルでの精度変化を示す。横軸が埋め込み次元数、縦軸が最終的な精度をあらわす。

数の上限が存在する(図 2)。加えて、埋め込み次元数が高いと、深層学習アルゴリズムの扱う次元数が高くなるため、計算量が増加してしまう問題もある。特に、Transformer ベースのモデルでは、埋め込み次元数の増加に対して、計算コストが指数的に増加する(図 3)。また、現実世界のオブジェクトごとに、有している情報量が異なり、適切な埋め込み次元数を一意に決定することは難しい。したがって、用いるデータや深層学習アルゴリズムごとに、適切な埋め込み次元数を選択する必要がある。

## 2.2 深層学習におけるハイパーパラメータ最適化

深層学習における埋め込み次元数の選択に関して、ハイパーパラメータ最適化[13]という手法が存在する。一般に、深層学習におけるハイパーパラメータとは、学習率、バッチサイズ、エポック数、層の数などが含まれる。ハイパーパラメータ最適化の手法に関して、基本的な手法として、グリッドサーチとランダムサーチが知られている。上記二つの手法では、あらかじめハイパーパラメータの取り得る範囲を設定する。グリッドサーチでは、設定されたハイパーパラメータの範囲において、事前に指定する幅をもつグリッドが設定される。各ハイパーパラメータに対して、網羅的にグリッド上の全ての組み合わせに対して学習することで、最適なハイパーパラメータを探索する。ランダムサーチでは、設定されたハイパーパラメータの範囲内においてランダムな値が設定される。ランダムに設定されるハイパーパラメータで、繰り返し学習することで最適なハイパーパラメータを探索する。上記二つの手法に対して、前回のハイパーパラメータでの学習結果を生かして、次のハイパーパラメータを決定する手法として、ベイズ最適化に基づく手法[14]が知られている。ベイズ最適化に基づく手法では、前回までの学習の結果を用いて次のハイパーパラメータを決定するため、ランダムサーチやグリッドサーチと比べて、効率的にハイパーパラメータを探索することが可能である。しかし、ベイズ最適化に基づく手法では、逐次的に処理するため、並列化が困難であるという問題が存在する。また、深層学習では一回の学習コストが大きいことから、深層学習におけるハイパーパラメータ最適化のコストは膨大であることが知られている。上記の問題に対して、高い性能の見込めないハイパーパラメータにおける深層学習モデルの学習を早期に打ち切ることで、最適化のための学習コストを減らす early-stopping に基づく手法が知ら

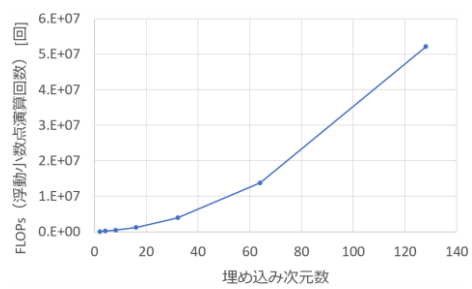


図 3 Action Transformer において埋め込み次元数を変更した際の、計算量(FLOPs)の変化を示す。横軸が埋め込み次元数、縦軸がモデルの計算量(FLOPs)をあらわす。FLOPs の計測には、keras-flops 0.1.2 ライブラリの get-flops 関数をバッチサイズ 1 として用いた。

れている。学習の早期打ち切り(early-stopping)を用いた主な手法として、Learning Curve Extrapolation、Bandit Based の二つが知られている。Learning Curve Extrapolation [15]では、深層学習における学習曲線をモデル化し、学習途中で外挿することで、最終的な性能を予測する。予測した性能をもとに、現在の最良のモデルと比較することで、学習を継続するか、打ち切るかを決定する手法である。Bandit Based の手法では、Successive Halving[16]と呼ばれる、多腕バンディット問題に基づく手法がベースとなっている。Successive Halving では、複数のハイパーパラメータ設定におけるモデルを並列学習し、学習途中で、事前に設定した予算に基づいたタイミングで、段階的に刈り込む(打ち切る)ことで、最適化のための学習コストを減らしつつ、ハイパーパラメータを最適化する。また、Successive Halving を非同期に拡張した手法として Asynchronous Successive Halving[17]、刈り込みのタイミングにバリエーションをもたせて、繰り返し Successive Halving を適応する手法として Hyper Band[18]が知られている。上記の early-stopping に基づく手法では、打ち切りのタイミングは、使用者が事前に設定、または事前に設定するコストに基づいて決定される。加えて、モデル規模に起因する学習コストの不均一性を考慮していないため、モデル規模に大きな影響を与えるハイパーパラメータを探索する場合、ハイパーパラメータ選択までの学習コストが不意に増大する可能性がある。提案手法では、損失の変化に基づいて将来性の評価タイミングが決定され、段階的な刈り込みをせず、一度の評価において各ハイパーパラメータによるモデルを評価する。また、探索するハイパーパラメータとして、モデル規模に影響を与える埋め込み次元数に限定し、低次元の埋め込み次元数から探索することで、精度に加えてモデル規模に基づく効率的な埋め込み次元数の選択を可能としている。

## 3. 学習序盤の損失変化に注目した次元数選択手法

本章では、提案手法について説明する。3.1 節で手法の概要について説明し、3.2 節および 3.3 節、3.4 節で、提案手法の処理について詳述する。

### 3.1 次元数選択手法の概要

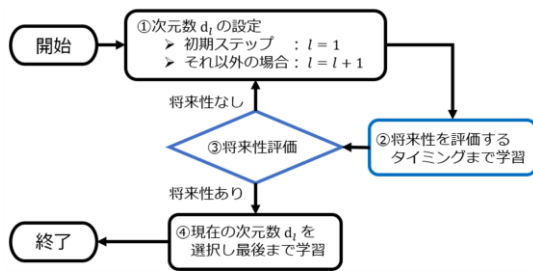


図 4 手法の概要

本研究では、学習序盤における損失変動から将来性評価タイミングを決定し、クラス別正答率に基づく将来性評価を用いて、埋め込み次元数を選択する手法を提案する(図 1)。まず、候補となる次元数のうち、最も低い次元数を初期次元数とする。次に、初期次元数の深層学習モデルで学習しながら、学習序盤において将来性を評価するタイミングを決定する。このとき決定した将来性評価タイミングにおけるモデルでのクラス別正答率を用いて、モデルの将来性を評価する。将来性があると評価された次元数のモデルでは、学習の終了条件に基づき最後まで学習する。一方、将来性がないと評価された次元数のモデルでは学習を中断し、次元数を更新することで、次の候補となる次元数で学習する。以上の処理を繰り返し実行することで、最終的な次元数を選択する(図 4)。

### 3.2 将来性評価タイミングの制御手法

提案手法におけるモデルの将来性の評価では、学習の序盤において十分な精度でモデルの将来性を評価することを目的として、損失の変化量の大きさに着目した将来性の評価タイミングを設定する。その結果として、得られる評価タイミングでのモデルは、クラス別正答率による違いがみられることが期待される。 $k$  エポック分の損失(Loss)の変化量の大きさが学習の進行度合いと関係していると考え、 $k$  エポック分の損失の変化量の大きさが、最初と比べてある一定割合( $0 < \alpha < 1$ )以下になる場合、そのエポックを将来性評価するエポック  $e$  とする。具体的には、単位時刻前の損失  $L_{i-1}$  と現時刻の損失  $L_i$  との差に注目し、 $k$  エポックごとに式(1)で定義する損失変動  $m_p$  を算出する。 $p$  は一つの埋め込み次元数における学習に必要な最大エポック数を  $s$  とし、 $0 \leq p \leq s/k$  を満たす自然数とする。

$$m_p = \frac{1}{k} \sum_{i=k(p-1)+1}^{kp} |L_i - L_{i-1}| \quad (1)$$

損失変動  $m_p$  は、学習が進むにつれて小さくなると期待される。式(2)(3)に示すように、 $m_p$  が最初の損失変動  $m_1$  の  $\alpha$  ( $0 < \alpha < 1$ ) 倍よりも、はじめて小さくなるエポックを将来性評価するエポック  $e$  とする。

$$e = k \min_{p=2,3,\dots} f(p) \quad (2)$$

$$f(p) = \begin{cases} p & \text{if } \alpha m_1 > m_p \\ \infty & \text{otherwise} \end{cases} \quad (3)$$

ここで、式(3)における  $\alpha$  はエポック  $e$  を制御するパラメータであり、 $\alpha$  の値を変更することによって、将来性評価タイミングを制御可能である。具体的には、 $\alpha$  を大きくする

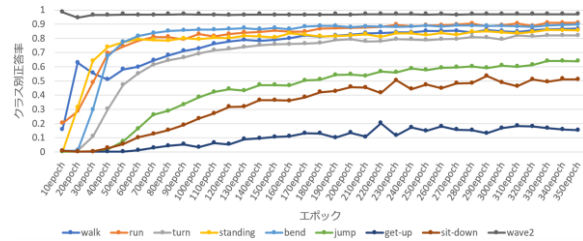
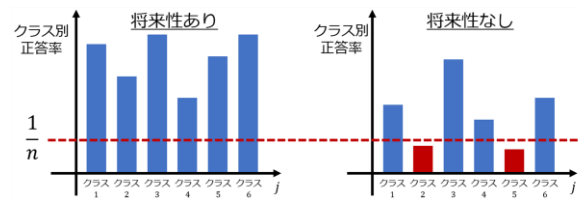


図 5 Action Transformer 埋め込み次元数 8 においてエポックごとのクラス別正答率を示す。横軸がエポック数、縦軸が対応するエポックでのクラス別正答率をあらわす。

図 6 将来性評価の例を示す。横軸が分類クラス、縦軸がクラス別正答率をあらわす。左図は将来性ありの例、右図は将来性なしの例を示す。横破線は、分類クラスを  $n$  として、クラス別正答率  $1/n$  の境界をあらわす。

ことで、評価タイミングが早まり、より学習序盤でのエポックにおいて将来性を評価する。

### 3.3 学習序盤のクラス別正答率に基づく将来性評価

提案手法では、学習序盤のクラス別正答率に着目し、将来性を評価する。予備実験をつうじて、学習序盤のクラス別正答率に着目する妥当性を検証するために、Action Transformer において、埋め込み次元数が 8 の場合における学習序盤のクラス別正答率の変化を調査した(図 5)。クラス別正答率が将来的に高くなるようなクラスでは、学習序盤の変動率が高い傾向が確認された。したがって、将来性の評価において、学習序盤のクラス別正答率に着目することは妥当であると考えられる。学習序盤のクラス別正答率を用いた将来性評価については、全クラス数を  $n$ 、クラス  $j$  ( $j = 1, 2, \dots, n$ ) のクラス別正答率を  $C_j$  とし、将来性評価するエポック  $e$  において式(4)を満たすクラスが存在しない場合、将来性があると評価する。

$$C_j < \frac{1}{n} \quad (4)$$

将来性を評価するエポック  $e$  においてクラスの正答率が無作為な場合と比較し、正答率の低いクラスが存在する場合には、クラス別正答率の変化が低いクラスが存在するとして、将来性がないと評価する(図 6)。

### 3.4 将来性評価タイミングに基づく次元数の選択

提案手法では、学習コストを減らすこと、および選択した次元数での深層学習モデルは、エッジデバイスでの動作を目指している。したがって、候補となる次元数において、低次元な埋め込み次元数から学習を開始する。低い次元数から開始することで、高い次元数での学習をせずに済み、

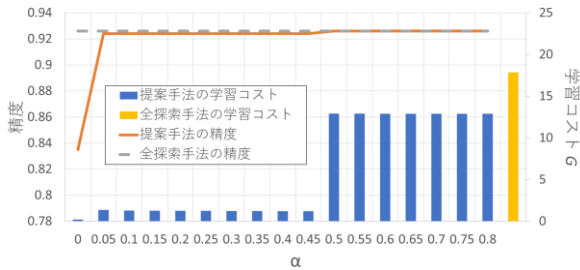


図 7 式(3)の  $\alpha$  を変更した際の精度、および次元数選択までに必要な学習コストに注目し比較した結果を示す。横軸が  $\alpha$ 、縦軸(右)が学習コスト  $G$ 、縦軸(左)が選択された埋め込み次元数でのモデルの最終的な精度をあらわす。

学習コストが少なくなると期待される。さらに、候補のうち、同等の精度をもちながら、可能な限り低い次元数が選択されることも期待される。

#### 4. 学習序盤の損失変化に注目した次元数選択手法の評価実験

##### 4.1.1 評価方法

本実験では、将来性評価タイミングを制御するために導入したパラメータ  $\alpha$  の変更が学習に与える影響に注目した。具体的には、一般的に行動認識に利用される Action Transformer を対象に、提案手法による埋め込み次元数の選択へ  $\alpha$  が与える影響と、行動認識の精度および次元数が選択されるまでにかかる学習コストに基づき評価する。本実験では、提案手法と全探索による手法を比較評価した。ここで、全探索による埋め込み次元数選択手法について説明する。まず、候補となる全ての埋め込み次元数に対して、指定された学習の終了条件を満たすまで学習する。その後、それぞれの学習で得られたモデルのうち、最も高い精度を評価に用いる。このとき得られたモデルに埋め込まれた次元数を全探索手法により選択された次元数として評価に用いる。本実験では、精度と次元数に加え、埋め込み次元数を選択するまでにかかる学習コストの比較により、パラメータ  $\alpha$  の変更による影響を多角的に評価した。本実験では、任意の埋め込み次元数におけるモデルの計算量(推論コスト)  $F$  [GFLOPs]と、学習に要したエポック数  $E$  との乗算結果を学習コスト  $G$  として導入した(式 5)。

$$G = F \cdot E \quad (5)$$

$G$  を導入した理由として、高次元な埋め込み次元数での 1 エポックと低次元な埋め込み次元数の 1 エポックでは、総計算量に大きな差が存在するからである。

##### 4.1.2 実験に用いた Action Transformer について

本実験では Action Transformer のうち、最も小さいネットワークである AcT- $\mu$  を採用した。候補とする埋め込み次元数は 2,4,8,16,32,64,128 とし、損失変動として用いる検証損失(Validation Loss)およびクラス別正答率は、3 スプリットそれぞれ 10 回分の平均を使用した。また、提案手法における評価タイミングを制御するパラメータは  $\alpha = \{0, 0.05, 0.1, \dots, 0.75, 0.8\}$  とした。ここで、 $\alpha = 0$  とは、早期における評価をせず、指定された学習の終了条件をみたくエポックにおいて将来性の評価をする場合であり、実験的な値として追加した。加えて、提案手法における損失変動

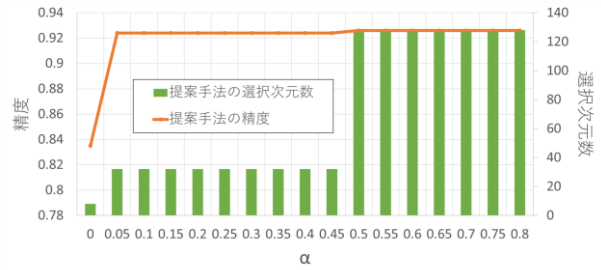


図 8 式(3)の  $\alpha$  を変更した際の精度、および選択次元数の関係について示す。横軸が  $\alpha$ 、縦軸(右)が最終的に選択された埋め込み次元数、縦軸(左)が選択された埋め込み次元数でのモデルの最終的な精度をあらわす。

$m_p$  の算出ステップ幅  $k$  について、5,10,15 とした場合、分類クラス数  $n$  について 5,9,20 とした場合、について提案手法への影響を評価した。クラス数変更の際のみ、候補とする埋め込み次元数に 256 を追加した。

##### 4.1.3 評価実験における深層学習モデルの学習終了条件

評価実験にあたり、学習の終了条件を設けた。一般に学習では、事前に決められたエポック数を学習するが、埋め込み次元数を変化させることにより、過学習や学習不足などの問題が発生する可能性がある。本実験では、過学習を防ぐため、検証損失に注目し、学習の収束を見ることで学習の終了条件を設定した。具体的な学習の終了条件は、 $k$  エポック分の検証損失の変化量が 3 回 0.001 未満となる場合とし、学習するエポック数、および将来性評価するエポック  $e$  の上限は 340 とした。

##### 4.1.4 歩行者の行動に注目したデータセット

データセットとして、短時間行動認識のためのデータセットである MPOSE2021[6]を使用した。MPOSE2021 は、スケルトンベースの短時間行動認識を目的とし、複数の行動認識のためのデータセットを用いて作成された。特徴として、一つの動画シーケンスが最大 30 フレームであることや、データがスケルトンデータであることが挙げられる。本研究では、歩行者の行動に注目し、用いるデータセットに変更を加えた。使用クラスを歩行者の行動として多くみられる 9 クラス(Walk, Run, Turn, Standing, Wave2, Bend, Jump, Get-up, Sit-down)に限定した。加えて、IXMAS[19]データセットの Walk クラスを Turn クラスへ追加、全ての Check-Watch クラスを Standing クラスへ追加した。分類クラス数の変化に対する提案手法の影響を評価実験する際は、前述した処理を加えた Class-9 ( $n = 9$ )、MPOSE2021 データセットにおける全てのクラスを含む Class-20 ( $n = 20$ )、より基本的な歩行者の行動に注目した Class-5 ( $n = 5$ ) の 3 種類のクラス数において評価した。Class-5 の内訳は、(Run, Walk, Standing, Bend, Sit-down)とした。

#### 4.2 全探索による次元数選択手法との比較評価

全探索手法と提案手法の評価のために、まず、精度、および学習コスト  $G$  に注目し比較した結果を図 7 に示す。結果、いずれの  $\alpha$  を選択した場合でも、提案手法による学習コスト削減を確認した。また、精度に関しては、 $\alpha = 0$  の場合のみ大きく低下していることを確認した。したがって、提案手法は、精度、および学習コストに注目した場合、全探索手法よりも有効である。

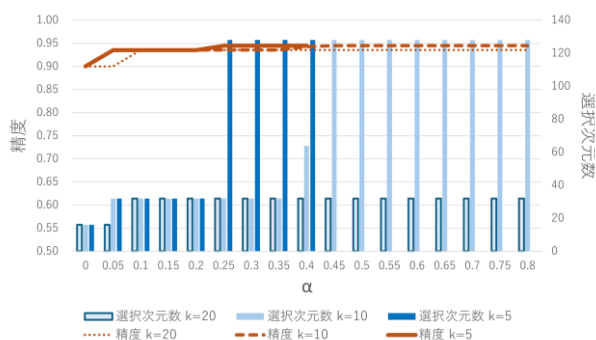


図 9 損失変動  $m_p$  の算出ステップ幅  $k$  を変更した場合の精度、および選択次元数に注目し比較した結果を示す。横軸が  $\alpha$ 、縦軸(右)が最終的に選択された埋め込み次元数、縦軸(左)が選択された埋め込み次元数でのモデルの最終的な精度をあらわす。 $k = 5$  では、 $\alpha \geq 0.45$  において、将来性のある次元数が存在せず、どの次元数も選択されなかった。

次に、精度、および選択次元数に注目し比較した結果を図 8 に示す。提案手法では  $\alpha$  の増加にともない、次元数が段階的に単調増加することから、選択次元数の推移がある程度予測可能である。選択される埋め込み次元数の変化点である  $\alpha = 0.45, \alpha = 0.5$  の間では、学習コストに大きな差が生じた。しかし、 $0.05 \leq \alpha \leq 0.45$  では、低次元でありながら、高次元の埋め込み次元数が得られた  $0.50 \leq \alpha \leq 0.80$  における精度と同等な精度を示した。以上より、将来性評価タイミングを制御するパラメータ  $\alpha$  は、ある程度小さく設定する必要はあるが、厳密な値や条件は不必要であると考える。

#### 4.3 損失変動 $m_p$ の算出ステップ幅 $k$ とクラス数 $n$ 変更による影響の評価

まず、提案手法における損失変動  $m_p$  の算出ステップ幅  $k$  を変更した場合の精度、および選択次元数に注目し比較した結果を図 9 に示す。 $k$  の値が大きくなるほど、より広範な  $\alpha$  に対して、低次元ながら高次元と同等の精度をもつ埋め込み次元数が自動選択可能であることを確認した。一方、 $k$  の値を増加させると、将来性評価タイミングであるかを判断する間隔が大きくなり、提案手法の早期評価という利点を失う。極端な場合として、ステップ幅を最大である学習エポック数に設定した場合、全探索による手法と同様に、与えられたエポック数を学習したのちに評価することになる。しかし、本実験では、図 7 と図 8 より、選択次元数の変化が学習コストの変化に大きく影響していることが分かる。ここで、学習コスト  $G$  は、式(5)より、モデルの計算量と学習に要したエポック数の乗算によって算出される。本実験結果では、選択次元数の変化がモデルの計算コストに関連し、学習コストへ大きな影響を与えた。そのため、選択次元数による計算コストの変化と比較して、学習に要するエポック数が学習コストへ与える影響は、小さいと考える。したがって、 $k$  の値は、ある程度大きく設定することが望ましい。

次に、分類するクラス数を変更した場合の、精度、および選択次元数に注目し比較した結果を図 10 に示す。クラス数変更に対して、 $\alpha$  による精度や選択次元数への影響の違いは、あまり見受けられなかった。ここで、それぞれの

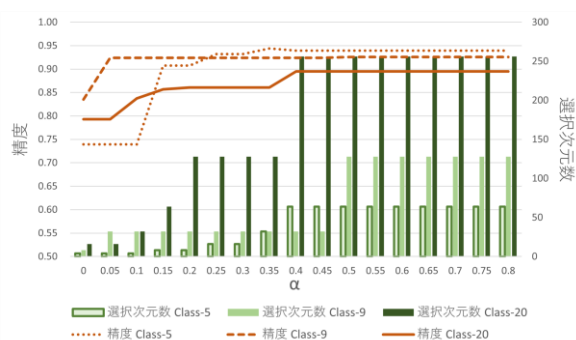


図 10 クラス数  $n$  を変更した場合の精度、および選択次元数に注目し比較した結果を示す。横軸が  $\alpha$ 、縦軸(右)が最終的に選択された埋め込み次元数、縦軸(左)が選択された埋め込み次元数でのモデルの最終的な精度をあらわす。

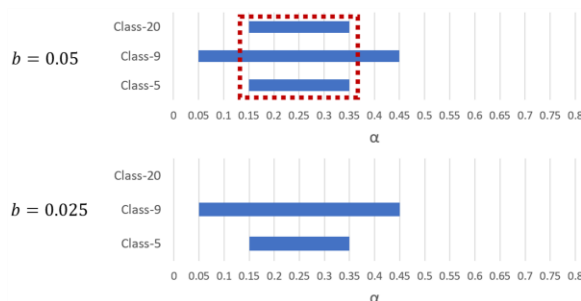


図 11 各クラス数において、被選択最高次元数 (Class-20 : 256, Class-9 : 128, Class-5 : 64) と同等 ( $\pm b$ ) の精度をもつ次元数のうち、被選択最高次元数よりも低次元な埋め込み次元数を自動選択可能な  $\alpha$  の範囲を示す。横軸が  $\alpha$ 、縦軸がクラス数をあらわす。

クラス数において、 $\alpha$  の変更により選択された最高の埋め込み次元数 (Class-20 : 256, Class-9 : 128, Class-5 : 64) を被選択最高次元数とする。また、被選択最高次元数と同等 ( $\pm b$ ) の精度をもつ次元数のうち、被選択最高次元数よりも低次元な埋め込み次元数を、自動選択可能な  $\alpha$  の範囲を図 11 に示す。 $b = 0.05$  では、点線の枠で示す部分において、クラス数に寄らず、同程度の  $\alpha$  の範囲において上記のような埋め込み次元数を自動選択可能であった。また、 $b = 0.025$  では Class-20 において、被選択最高次元数と同等の精度をもち、被選択最高次元数よりも低い次元数が存在していない。これは、上記を満たす次元数が、今回候補とした次元数 128 と 256 の間に存在しており、候補となる次元数に存在しなかったことが原因であると考えられる。したがって、提案手法は、適切な次元数を候補として設定した際、分類クラス数に対して頑健であると考えられる。

## 5. まとめ

本研究では、損失変化に着目した評価タイミングにおいて、クラス別正答率に注目した将来性評価に基づく、次元数の選択手法を提案した。また、提案手法は、将来性評価タイミングを制御するためのパラメータ  $\alpha$ 、および損失変動  $m_p$  の算出ステップ幅  $k$  を調節することで、学習コストを抑えつつ、低次元の埋め込み次元数でありながら、精度は高次元の埋め込み次元数と同等な次元数を、自動選択可

能であることを示した。加えて、提案手法がクラス数に対して頑健であることも確認した。

今後は、他のネットワークを用いた実験や、設計者が将来性評価タイミングを制御するパラメータ  $\alpha$ 、および損失変動  $m_p$  の算出ステップ幅  $k$  の関係から、 $\alpha$  と  $k$  における望ましい範囲を推定する手法について検討する。

#### 参考文献

- [1] Duan, Haodong, *et al.*, “Revisiting skeleton-based action recognition”, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.
- [2] Xin, Wentian, *et al.*, “Transformer for skeleton-based action recognition: A review of recent advances”, Neurocomputing (2023).
- [3] Yang, Lichao, *et al.*, “A lightweight temporal attention-based convolution neural network for driver’s activity recognition in edge”, Computers and Electrical Engineering, Vol. 110, pp. 108861(2023).
- [4] Dai, Cheng, *et al.*, “A light-weight skeleton human action recognition model with knowledge distillation for edge intelligent surveillance applications”, Applied Soft Computing, Vol. 151, pp. 111166(2024).
- [5] Sufian, Abu, *et al.*, “A deep transfer learning-based edge computing method for home health monitoring”, 2021 55th annual conference on information sciences and systems (CISS). IEEE, 2021.
- [6] Vittorio Mazzia, *et al.*, “Action transformer: A self-attention model for short-time pose-based human action recognition”, Pattern Recognition, Vol. 124, pp. 108487(2022).
- [7] Yoshihiko OZAKI, *et al.*, “Hyperparameter Optimization Methods: Overview and Characteristics”, 電子情報通信学会論文誌, Vol. J103-D, No. 9, pp. 615-631(2020).
- [8] Petro Liashchynskyi, *et al.*, “Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS”, arXiv:1912.06059(2019).
- [9] Wang, Shirui, *et al.*, “A survey of word embeddings based on deep learning”, Computing, Vol. 102, no. 3, pp. 717-740(2020).
- [10] Feng, Yang, *et al.* “Unifying Specialist Image Embedding into Universal Image Embedding”, arXiv preprint arXiv:2003.03701 (2020).
- [11] Koroteev, M. V. “BERT: a review of applications in natural language processing and understanding”, arXiv preprint arXiv:2103.11943 (2021).
- [12] Ashish Vaswani, *et al.*, “Attention is all you need”, Advances in Neural Information Processing Systems, Vol. 30(2017).
- [13] Yu, Tong, *et al.*, “Hyper-parameter optimization: A review of algorithms and applications.” arXiv preprint arXiv:2003.05689 (2020).
- [14] Hutter, Frank, *et al.*, “Sequential model-based optimization for general algorithm configuration”, Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5. Springer Berlin Heidelberg, 2011.
- [15] Domhan, Tobias, *et al.*, “Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves.” Twenty-fourth international joint conference on artificial intelligence(2015).
- [16] K. Jamieson and A. Talwalkar, “Non-stochastic best arm identification and hyperparameter optimization”, Artificial intelligence and statistics, pp. 240-248(2016).
- [17] L. Li, *et al.*, “Massively parallel hyperparameter tuning”, arXiv preprint arXiv:1810.05934, pp. 1-16(2018).
- [18] L. Li, *et al.*, “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization”, J. Machine Learning Research, Vol. 18, no. 185, pp. 1-52(2018).
- [19] Daniel Weinland, *et al.*, “Free viewpoint action recognition using motion history volumes”, Computer Vision and Image Understanding, Vol. 104, no. 2, pp. 249-257(2006).