

日常行動の Zero-shot Deep Learning に向けた 基底的动作の自動抽出の検討

竹下 昌志^{1,a)} 松木 萌¹ 前川 卓也² 井上 創造¹

概要：行動認識技術はユビキタスコンピューティングを実現する基盤技術の一つである。一般的な行動認識技術では教師あり機械学習が用いられているが、特徴設計を手動で行わなければならない点と教師データの収集が手間である点が、応用のボトルネックとなる。これらの問題に対し、深層学習を用いることで特徴量抽出の自動化をめざす取り組みや、データ収集効率化を目指し未知クラスを推定する取り組みが行われている。我々は、それらの問題を解決するために深層学習手法に基づいた未知クラス推定手法 (Zero-shot Deep Learning) を提案する。未知クラスの推定を実現するため、行動を構成する基底的な動作の自動抽出および、それらの生起を表す属性ベクトルと呼ばれる表現の自動獲得を目指す。本稿では属性ベクトル推定精度の向上を目指し、(検証 1) 深層学習過程での畳み込む時間長の調整、(検証 2) 活性化関数の検証、の 2 つの検証を行った。その結果、検証 1 では推定精度が向上し、検証 2 では活性化関数に温度付き Softmax 関数を用いることが基底的动作の抽出に有効であることがわかった。

キーワード：深層学習, Zero-shot Learning, 行動認識

1. はじめに

ユビキタスコンピューティングにおいて、センサを用いた行動認識技術が盛んに研究されている [4]。近年の行動認識技術では教師あり機械学習が多く用いられているが、特徴量設計が人の手で行われている問題 [5] の解決とデータ収集の効率化 [7] を行う必要がある。文献 [6] ではこれらの問題を解決するために、深層学習と Zero-shot 学習法を組み合わせたモデルが提案されている。Zero-shot 学習法とは、学習用データには存在しないクラス (未知クラス) を推定することを目的とした学習法である。文献 [6] で提案されているモデルは、未知クラスを推定するための属性推定器とよばれるものを未知クラスと学習用データに存在するクラス (既知クラス) について一般化するために、複雑な行動を構成する基底的な動作を抽出することを目的として設計されている。複雑な行動と基底的な動作は厳密に区別されるものではないが、例えば「コーヒーを飲む」という複雑な行動は「コップを掴む」「コップを持ち上げる」などの基底的な動作によって構成される、と考え区別される^{*1}。しかしこのモデルでは、一般化のための基底的な動

作の抽出がうまくできていないという問題がある。本稿では、文献 [6] で提案されているモデルがより基底的な動作を自動的に抽出できるようにするにはどうすべきかを検討するために、2 種類の検証を行う。1 つ目は時間窓の調整である。行動認識技術において、時系列データの連続した一部分を参照する際に、その参照する範囲の長さを時間窓 (time window) と呼ぶ。この時間窓を調整することで基底的な動作の自動抽出の改善が可能かどうかを検証する。2 つ目は正則化・活性化関数の検証である。正則化の導入、および活性化関数に温度付き Softmax 関数とよばれる関数を用いることで基底的な動作の自動抽出の改善が可能かどうかを検証する。

2 章で Zero-shot 学習法についての関連研究について述べる。3 章で文献 [6] で提案されているモデルを説明する。4 章で 2 種類の検証理由と手法、および評価方法を説明する。5 章で評価実験結果からモデルの検討・考察を行い、6 章でまとめる。

2. 関連研究

本章ではまず Zero-shot 学習法について述べる。次に

¹ 九州工業大学

² 大阪大学

^{a)} q108072m@mail.kyutech.jp

^{*1} 「コーヒーを飲む」という行動はさらに複雑な行動を構成する (例えば「カフェに行く」という行動を構成する) と考えること

ができるし、「コップを掴む」という動作は「指を曲げる」「指に力を加えてコップを固定する」というさらに基底的な動作から構成されるときも考えることもできる。これらはどのような行動を認識したいかに依存して適切に選択されるべきだと考える。

Zero-shot 学習法のセンサ行動認識における応用の先行研究を紹介し、その問題点を指摘する。

2.1 Zero-shot 学習法

Zero-shot 学習法では未知クラスを推定するために、全クラスに対して意味的な情報が表現されている意味ベクトルを用いる。意味ベクトルが集合する空間を意味空間と呼び、この空間には既知クラスと未知クラスがどちらも存在する。一方、センサデータの特徴量ベクトルが集合する空間を特徴量空間とした時、既存の Zero-shot 学習法は、学習時に特徴量空間から意味空間へ同一クラスを対応づけるための射影関数を学習する。テスト時に、未知クラスの特徴量ベクトルから意味空間上に学習モデルを用いて推定射影を行い、意味空間内に存在する未知クラスのどの意味ベクトルに類似するのかを探索する。つまり、両空間に存在する既知クラスを通じ、特徴量空間上と意味空間上の間で、ベクトルの射影方法を学習し、未知クラスが存在する意味空間上でクラス探索を行う。そうすることで、既知クラスと未知クラスとの情報を意味空間上で共有し、未知クラスの推定を行うことができる。

文献 [3] は動物の画像から動物の意味ベクトルを推定するようなモデルを提案している。この場合の意味ベクトルは、「色」や「形」などの属性を含むかどうかを 1 か 0 で表したベクトルを仮定している。以下ではこうした意味ベクトルを属性ベクトルと呼ぶことにする。例えばシロクマの属性ベクトルは「黒」が 0, 「白」が 1, と表現される。シロクマが既知クラス、シマウマが未知クラスになっている場合、シマウマの属性ベクトルを構成する各属性、つまり「白」の属性を、シロクマの「白」の属性を通して、既知クラスと未知クラスで情報を共有する。このように既知クラスデータを用いた学習のみから未知クラスの意味情報を共有することによって、未知クラスの推定を行う。

2.2 センサ行動認識における Zero-shot 学習法

Zero-shot 学習法のセンサ行動認識への応用の初期の研究として、例えば文献 [2] の研究があげられる。彼らのモデルでは、まずセンサデータから平均、標準偏差などの特徴量を抽出し、抽出された特徴量から属性ベクトルを推定し、属性ベクトルを通して行動クラスを推定する。彼らの研究では、推定したい行動は「昼食を取る」や「デスクで仕事をやる」などの複雑な行動で、これらの推定を目的としている。そうした複雑な行動は「座る」や「テーブルの上に手を置く」などの基底的な動作によって構成されていると仮定し、各属性を単純な行動とし、属性ベクトルを設計している。そして、センサデータから平均、標準偏差などの特徴量を抽出し、特徴量から属性ベクトルを学習するモデルを提案している。1 章で述べたように、こうしたモデルの問題点は、特徴量の抽出が人の手で行われる点、それ

によってほとんどの場合、統計的な情報しか抽出できない点があげられる。これらの研究は事前に統計的な特徴量を計測し、学習を行うという 2 段階の作業が独立しているモデルであるのに対し、本研究では Convolutional Neural Network (CNN) を取り入れ特徴量抽出の学習を自動化する Zero-shot 学習法の提案を目的とする。しかしこのようなモデルでは、学習時に既知クラス分類のための偏った特徴量が抽出され、未知クラス推定のための共有情報が得られないという懸念がある。本研究では、属性ベクトルを推定する部分で各属性を分類するモデル（基底的な動作を抽出するモデル）を組み込むことで、未知クラス推定可能なモデルを目指す。

3. SAPB - NN モデル

本稿では文献 [6] で提案されている「時系列ごとに 1 つの属性を推定する手法 (SAPB-NN: Single-Attribute Prediction Based Neural Network)」を検討する。以下では検討にあたっての問題設定、およびモデルを説明する。

3.1 問題設定

入力にはセンサデータで、これを $\mathbf{x}$ と表す。出力は行動クラス y と表す。行動クラス y にはそれぞれ属性ベクトル $\mathbf{z}$ が与えられているものとする。属性ベクトル $\mathbf{z}$ は、属性ラベル $\{z_1, z_2, \dots, z_n\}$ で表現され、各 z_i にはそれぞれ 1 か 0 が割り当てられている。

3.2 モデルの概要

図 1 にモデルの概要を示す。まずセンサデータ $\mathbf{x}$ を CNN 層に入力し、特徴量を抽出・圧縮する。CNN 層によって抽出・圧縮された特徴量を時間軸に沿って分割し、分割したそれぞれの特徴量を各ニューラルネットワークモデルに入力する。このとき、各ニューラルネットワークモデルは属性を 1 つだけ推定することを目的として、属性についての多クラス分類問題を解くよう構築される。この「属性を 1 つだけ持つ属性ベクトル」を単属性ベクトルと呼ぶことにする。こうして推定された単属性ベクトルを統合し、最終的にマルチラベルの属性ベクトル $\mathbf{z}$ を出力する。出力されたマルチラベルの属性ベクトルを用いて最近傍探索を行い、クラス y を出力する。

4. 検証・評価方法

本章では検証方法及び評価方法を説明する。4.1 節で評価用データについて説明する。4.2 節で 2 つの検証の概要を説明する。

4.1 評価用データ

本稿でのモデルを評価するために文献 [6] で用いられている人工データを用いる。この人工データは以下の手順で

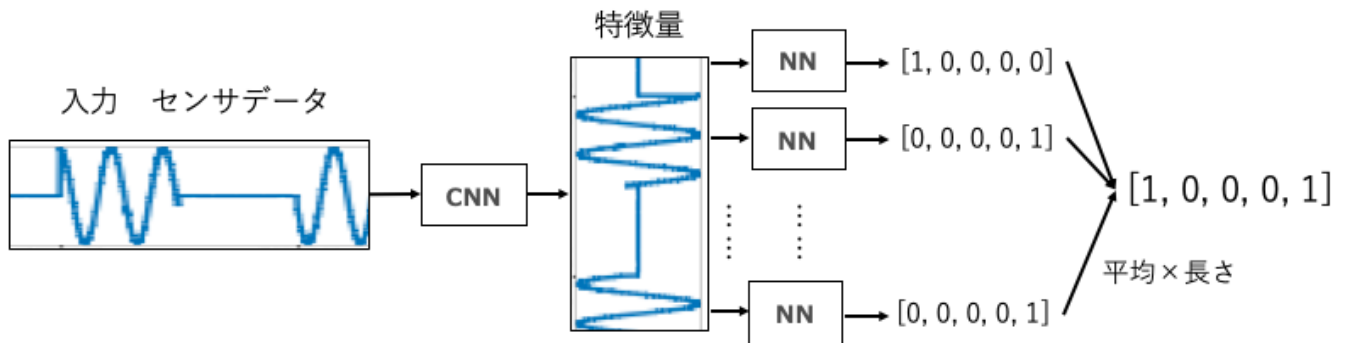


図 1: SAPB-NN の概要 [cf.[6]]. まずセンサーデータが CNN に入力され、特徴量が抽出・圧縮される。この特徴量を時間軸に沿って分割し、分割したそれぞれの特徴量を各ニューラルネットワークモデルに入力し、属性を 1 つしか持たないベクトル（単属性ベクトル）を推定する。そして、推定された単属性ベクトルを統合して、マルチラベルの属性ベクトルを出力する。

表 1: 人工データのクラスと属性の対応表 [6]

act	wave_walk	wave_run	wave_stay	wave_sitting	wave_lie
Wave_1	1	1	0	0	0
Wave_2	1	0	1	0	0
Wave_3	1	0	0	1	0
Wave_4	1	0	0	0	1
Wave_5	0	1	1	0	0
Wave_6	0	1	0	1	0
Wave_7	0	1	0	0	1
Wave_8	0	0	1	1	0
Wave_9	0	0	1	0	1
Wave_10	0	0	0	1	1

作成される [6].

- (1) 振幅や振動数が異なる 5 つの種類の sin 波を用意する
- (2) 2 つの sin 波を、時系列上に交互に組み合わせ、複合した波データを作成する。これを 5 つの sin 波全ての組み合わせで、計 10 個の複合派を作成する
- (3) 複合波からランダムに一部サンプルを 200 個抜き出し、形の異なる複合波を複製する。

異なる 5 つの種類の sin 波はそれぞれ単純な行動（属性）を表現しており、それぞれ “wave_walk”, “wave_run”, “wave_stay”, “wave_sitting”, “wave_lie” とラベル付けされる。複合波は異なる 2 つの単純な行動（属性）を持つ複雑な行動クラスを表現しており、“Wave_1”, “Wave_2”, ..., “Wave_10” とラベル付けされる。表 1 に複合波と属性の対応関係を示す。

このような人工データを用いる主な理由は (1) 複雑な行動と基底的な動作を明確に区別している、またそのため (2) モデルの評価を行いやすいという点である*2。

4.2 検証方法

3 章で説明した SAPB-NN を用いて 2 種類の検証を行う。

*2 今後の課題として、現実の環境で得られたセンサーデータを用いて実験を行う。7 章を参照。

まず最初に実験設定を説明し、その後、2 種類の検証について説明する。

4.2.1 実験設定

2 つの設定で実験を行う。1 つ目の設定は既知の行動認識、つまり未知クラスのない設定である。全データを学習用データとテスト用データに分け、学習用データを用いてモデルを学習し、テスト用データを用いて評価を行う。2 つ目の設定は Zero-shot 設定、つまりいくつかのクラスを未知クラスとして設定し、他のクラス（既知クラス）を学習用データとして用いる。テスト時には未知クラスを用いて評価を行う。Zero-shot 設定において未知クラスに用いるのは次の 5 つである。

- (1) Wave_1
- (2) Wave_8
- (3) Wave_10
- (4) Wave_2
- (5) Wave_7

上から順に未知クラスに設定する。例えば未知クラス数が 2 つのときは Wave_1 と Wave_8 を、3 つのときはさらに Wave_10 を未知クラスに設定する。

4.2.2 検証 1: 時間窓の調整

SAPB-NN モデルにおいて、CNN 層の役割は特徴量抽出とセンサーデータの圧縮である。圧縮されたデータは時間軸に沿って分割され各ニューラルネットワークモデルに入力される（図 1）。このとき、CNN 層でのプーリング幅（pooling size）を変えることで、各ニューラルネットワークモデルに入力されるデータに情報として含まれる元々のデータの長さを変えることができる。したがってプーリング幅を変えることで実質的に時間窓を変えることになる。検証 1 ではプーリング幅を調整して、基底的な動作の自動抽出が改善されるかどうかを検証する。

以下では CNN 層の圧縮に関わるプーリング層の働きと、それが実質的に時間窓を表現することを説明する。

CNN は一般的に、畳み込み層とプーリング層によって

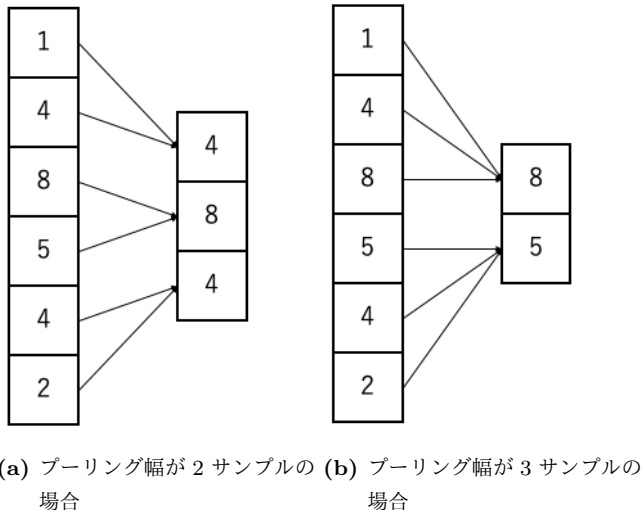


図 2: プーリング層の役割を説明するための例。右図はプーリング幅が 2 の場合を、左図はプーリング幅が 3 の場合を示しており、プーリング幅の違いによってデータの圧縮率が異なることを示している。

構成される。CNN 層におけるプーリング層の働きの概要を図 2 に示す。図 2 の各マスの列は一次元データを示している。

プーリング層の働きを説明する。図 2 のように一次元データが入力されたとき、プーリング層では一次元データからある範囲（プーリング幅）を選択し、その中の最大値を抽出する。この範囲をずらしながら繰り返すことで次の一次元データを作り、これを出力する。こうすることでデータを圧縮して表現することができる。図 2a はプーリング幅が 2 サンプルの場合、図 2b はプーリング幅が 3 サンプルの場合である。2 つの図を比較すると、プーリング幅が 3 サンプルの場合のほうが圧縮率が高い（6→2 サンプルに圧縮されている）。したがってプーリング幅によって圧縮率が異なる。

CNN 層によって圧縮されたデータは各ニューラルネットワークモデルに入力され、各ニューラルネットワークモデルは圧縮されたデータから属性を推定する（図 1）。ここで、各ニューラルネットワークモデルにどのようにデータが入力されるかを説明する。図 3 にプーリング幅が 2 サンプル、入力される時系列データの長さが 36 サンプル、プーリング層が 2 つの場合を示す。この例では、入力されるデータはプーリング層を通過するたびにデータの長さが半分に圧縮される。したがって最終的にデータは 4 分の 1 の長さに圧縮される。圧縮されたデータのサンプルそれぞれが各ニューラルネットワークモデルに入力される。ここで、元々のデータ長が 4 分の 1 に圧縮されたことで、圧縮された長さ 9 サンプルのデータそれぞれには圧縮前のデータの 4 サンプル分の情報が圧縮されて表現されている。したがって各ニューラルネットワークモデルには 4 サンプル

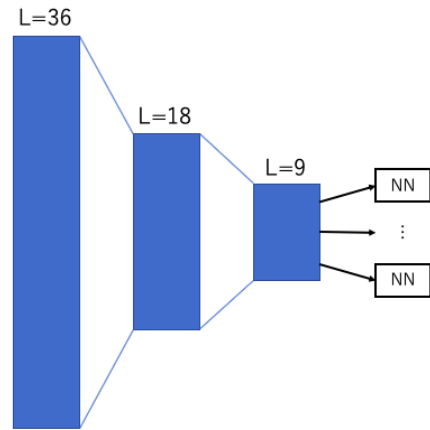


図 3: SAPB-NN でのデータ圧縮の流れ。L はデータの長さ。プーリング層を通過する度にデータがプーリングされ、データの長さが半になる。SAPB-NN では圧縮したデータを時間軸方向に分割して各ニューラルネットワークモデルに入力する。

分の情報が入力されていると考えることができ、実質的に時間窓が 4 サンプルであるようなニューラルネットワークモデルとなる。

以上のように考えると、プーリング層によってデータを圧縮することで時間窓を表現することができる。プーリング幅を大きくすればそれだけデータを圧縮することになり、時間窓を大きくすることになる。

本稿ではプーリング幅を変更し時間窓を調整することによって基底的な動作の自動抽出の改善につながるかどうかを検証する。したがって、本稿ではプーリング層におけるプーリング幅を変更することで時間窓の大きさを変更して学習を行う。また実験設定には既知の行動認識設定を用いる。

4.2.3 検証 2: 正則化・活性化関数の検証

SAPB-NN モデルにおける各ニューラルネットワークモデルは属性を 1 つだけ持つ単属性ベクトルを推定することを目的としている（図 1）。その理由は、それによって基底的な動作の抽出が可能になると考えるからである。

単属性ベクトルは属性を 1 つだけ持つ（スパースなベクトルである）ので、CNN 層から出力された特徴量から、その部分のデータがどの基底的な動作（属性）であるかを推定していることになる。スパースとは「多くの変数のうちほとんどがゼロでごく一部だけが非ゼロの値を取ることを意味」する [8]。以上のことから、我々は単属性ベクトル（スパースなベクトル）を推定するように設計することで、基底的な動作の抽出を目指す。

単属性ベクトルを推定させ基底的な動作を抽出できるようにするために (a) 各ニューラルネットワークモデルに正則化を用いる、また (b) 活性化関数に温度付き Softmax 関数を用いることを検証する。

まず正則化について、文献 [8] を参考にして説明する。

入力 $\mathbf{x}$ に対してクラスラベル y を出力する関数を $f(x)$, 損失関数を $L(f)$ とする. 機械学習では損失関数を最小にするような関数 f の重み $\mathbf{w}$ を学習することになる. したがって機械学習は次式を解くように学習する.

$$\min L(f) \quad (1)$$

このままでは関数 f が学習用データに適合しすぎる場合があること (過剰適合) が経験的に知られている [8]. そこで重み $\mathbf{w}$ について, 関数 f の学習用データへの過剰適合を防ぐことを目的として, 式 1 を以下のように変更する.

$$\min L(f) + \lambda |\mathbf{w}| \quad (2)$$

$$\min L(f) + \frac{\lambda}{2} |\mathbf{w}|^2 \quad (3)$$

ここで $\lambda > 0$ は正則化パラメータである. また式 2,3 で $\min L(f)$ に加算されている項を正則化項と呼ぶ. 正則化項を導入することで重み $\mathbf{w}$ がスパースになることが知られており, また重み $\mathbf{w}$ がスパースになることによって関数 f の学習用データへの過剰適合を防ぐことができる. このように, 正則化項を導入して重みをスパースにし, モデルの過剰適合を防ぐことを正則化とよぶ^{*3}. 特に, 式 2 の正則化を L1 正則化, 式 3 の正則化を L2 正則化と呼ぶ. 本稿では L1 正則化と L2 正則化を用いてモデルが改善されるかどうかを検証する.

次に温度付き Softmax 関数について説明する. Softmax 関数と温度付き Softmax 関数はそれぞれ次式で表される [1].

$$S(y_i) = \frac{\exp(y_i)}{\sum \exp(y_k)} \quad (4)$$

$$S(y_i) = \frac{\exp(\frac{y_i}{T})}{\sum \exp(\frac{y_k}{T})} \quad (5)$$

ここで $S(y_i)$ はクラス i 番目の出力であり, T は温度と呼ばれるパラメータである. Softmax 関数は入力から各クラスの推定確率を出力する.

温度付き Softmax 関数において, T を 1 より小さくすると, 高い確率で推定されているものをさらに高く, 低い確率で推定されているクラスをさらに低く出力する. 逆に T を 1 より大きくすると, 確率の高いクラスと低いクラスとで出力される推定値の差が小さくなる.

我々は温度 T を 1 より小さくし高い確率で推定される属性を強調させることが単属性ベクトル (スパースなベクトル) の推定に貢献すると考える. したがって本稿では温度付き Softmax 関数を用いることで単属性ベクトルの推定がどのように変化するかを検証する.

正則化, 温度付き Softmax 関数の検証には Zero-shot 設定を用いる.

4.3 評価指標

評価指標は 3 種類の正解率とスパースさの度合い (スパース性) を表す指標を用いる.

4.3.1 正解率

3 種類の正解率はそれぞれ, 各クラスに与えられている属性ベクトルとの完全正解率 (total accuracy) と部分正解率 (binary accuracy), およびクラス分類の正解率 (accuracy) である. 完全正解率とは, 推定された属性ベクトルが, 正解のクラスの属性ベクトルと完全に一致していた場合の正解率を 1, そうでない場合の正解率を 0 とする評価指標である. 部分正解率は次式で表現される評価指標である.

$$\text{binary accuracy} = \frac{a}{l} \quad (6)$$

ここで a は正解の属性ベクトルと推定された属性ベクトルとの属性値の一致数であり, l は属性ベクトルの長さを示す. したがって部分正解率は正解の属性ベクトルと推定された属性ベクトルとの一致率を表す評価指標である. 3 つ目の正解率はクラス分類の正解率であり, 属性を用いたクラス分類の正解率を表す. ここで正解率とは, 推定したクラスと正解のクラスの一致数を全データ数で割ったものである.

4.3.2 スパース性

すでに述べたように, スパースとは「多くの変数のうちほとんどがゼロでごく一部だけが非ゼロの値を取ることを意味」する [8]. 我々はモデルの各ニューラルネットワークモデルが出力する属性ベクトルが単属性ベクトル, すなわちスパースなベクトルであるように設計したいから, 出力される属性ベクトルがどれほどスパースであるかを計算し, それを用いて評価する.

スパース性は次のように計算される. モデルにセンサデータを入力して各ニューラルネットワークモデルの 1 つが出力した属性ベクトルが $\{0.1, 0.1, 0, 0.9, 0.2\}$ だとする. この属性ベクトルの最大値の部分 (0.9) の値を 1 とし, 他の部分が 0 であるような one-hot ベクトルを作る ($\{0, 0, 0, 1, 0\}$). この one-hot ベクトルと属性ベクトルとの L1 ノルムを計算する (0.1). この手順を出力された全ての属性ベクトルについて行い, その平均を計算する. このような計算によって得られた値は, モデル内部の各ニューラルネットワークモデルが出力した属性ベクトルがどれだけ one-hot ベクトルに近いかを表現している. one-hot ベクトルは単属性ベクトルと同じであるため, これによって各ニューラルネットワークモデルが単属性ベクトルを推定できているかどうか, つまりスパースな属性ベクトルを推定できているかどうかを評価できる. このとき, このスパース性の値が小さいほど推定された属性ベクトルはスパースである.

^{*3} 正則化が重み $\mathbf{w}$ をスパースにし, モデルの学習用データへの過剰適合を防ぐことができるという根拠は文献 [8] を参照.

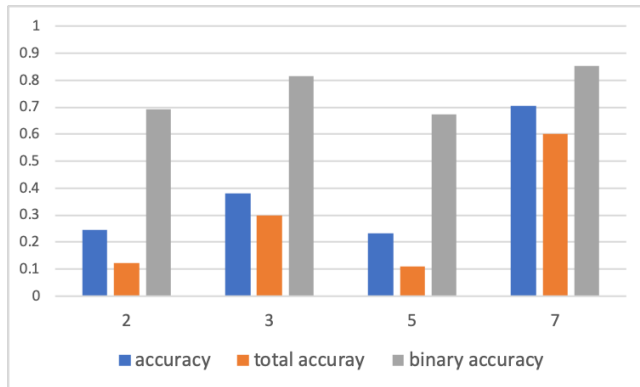


図 4: 時間窓を調整した結果. 縦軸は正解率であり, 横軸はプーリング幅を表している.

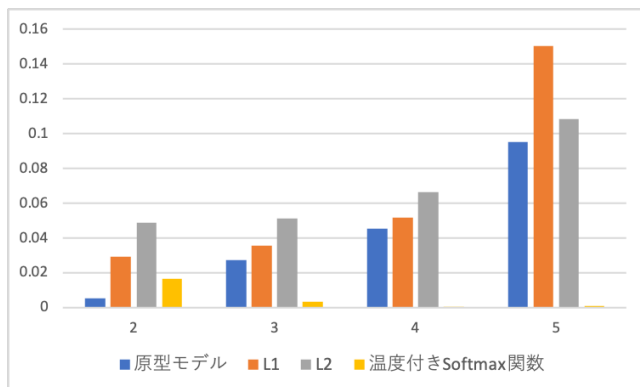


図 5: 各ニューラルネットワークモデルの出力する単属性ベクトルのスパース性の評価結果. 縦軸は 4.3.2 節で述べた手順で計算されたスパース性を示す指標の大きさを, 横軸は未知クラスの数を表している. また原型モデルは正則化や温度付き Softmax 関数を用いていないモデルである.

5. 結果

4 章で述べた実験の結果および評価を以下で行う.

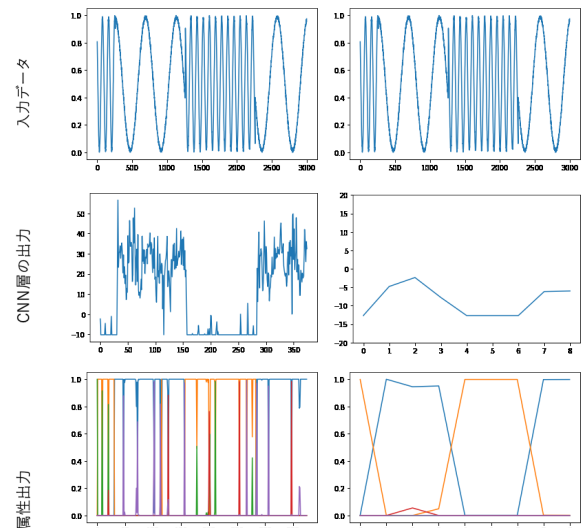
5.1 検証 1 の結果

4.2.2 節で述べた「検証 1: 時間窓の調整」の結果を図 4 に示す. 図 4 を見ると, プーリング幅が 7 のとき, すなわち時間窓が最も大きいときに精度が最もよいことがわかる. 一方でプーリング幅が大きくなるにつれて精度が高くなっていないことも確認できる.

5.2 検証 2 の結果

4.2.3 節で述べた「検証 2: 正則化・活性化関数の検証」の結果を表 2 に示す. 表 2 を見ると, 正則化や温度付き Softmax 関数を用いることが一貫して精度向上に貢献しているわけではないことがわかる.

また図 5 に各ニューラルネットワークモデルの出力する単属性ベクトルのスパース性の評価結果を示す. この図を



(a) プーリング幅が 2 サンプルの場合 (b) プーリング幅が 7 サンプルの場合

図 6: センサデータと CNN 層の出力と属性の中間表現. すべての図について, 横軸は時間軸を表している. また上段の図の縦軸はセンサデータ (自作データ) の値を, 中段の縦軸は CNN 層から出力された値を, 下段の縦軸は各ニューラルネットワークモデルが出力した属性値の大きさを表している.

見ると, 未知クラス数が大きくなるに連れスパース性も大きくなっている (スパースでなくなっている) ことがわかる. また温度付き Softmax 関数を用いたモデルが, 未知クラス数が 2 つの場合を除きスパース性が最も小さく (最もスパースであり), また未知クラス数が 2 つのときでも 2 番目に小さいことがわかる.

6. 考察

本章では 5 章の結果を元に, それぞれの検証について考察を行う.

6.1 検証 1 についての考察

検証 1 から, プーリング層の調整が推定精度に貢献し, 最も圧縮率が大きい場合に推定精度がいいことがわかった. ここでは, 推定精度だけでなく, 中間層 (単属性ベクトル推定層) でのパフォーマンスも向上しているかどうかを考察する.

図 6 は, 各層の波データを可視化している. 上から入力層の波データ, CNN 層の出力時の波データ, そして単属性ベクトル推定の結果を示している. また, 左側がプーリング幅が 2 サンプルの場合, 右側がプーリング幅が 7 サンプルの場合を示している. 下段の波の色は, 属性ベクトルの 5 次元を表している.

図 6a に着目すると, CNN 層の出力は入力データの波に対して細かく波の変化に敏感に対応している様子がわか

表 2: 検証 2 の結果. 原型モデルは正則化や温度付き Softmax 関数を用いていないモデルである. 赤太字は各未知クラス数のときの最高精度である.

未知 クラス数	accuracy				total accuracy				binary accuracy			
	2	3	4	5	2	3	4	5	2	3	4	5
原型モデル	0.974	0.648	0.596	0.478	0.398	0.138	0.049	0.000	0.859	0.610	0.668	0.617
L1 正則化	1.000	0.607	0.845	0.200	0.428	0.012	0.288	0.000	0.855	0.647	0.790	0.456
L2 正則化	0.998	0.628	0.279	0.472	0.288	0.172	0.080	0.003	0.792	0.624	0.621	0.697
温度付き Softmax 関数	0.979	0.373	0.357	0.600	0.089	0.235	0.000	0.285	0.778	0.567	0.544	0.717

る. 下段の属性ベクトルを可視化した図は, 様々な属性が交互に推定されている様子が見える. 一方, 図 6b のとき波の CNN 層の出力の波は穏やかで, 下段の属性推定も 2 種類の属性が対応していることがわかる.

図 2 で説明しているように, プーリング幅が小さいほど時間窓は短いので, プーリング幅が 2 サンプルの場合はプーリング幅が 7 サンプルの時に比べ入力データからの時間窓が短い. そのため, 図 4 からみてわかるように, CNN 層によって, 細かい特徴量が抽出されているかどうかが顕著に現れている. さらに, この特徴量抽出の結果が, 単属性ベクトルの推定にも大きな影響を与えていることがわかった.

今後は, 実際の加速度センサデータを用いた行動認識への応用を検討する. その場合, 特徴量抽出のためのプーリング幅の検討を優先的にすべきであると考えている.

6.2 検証 2 についての考察

検証 2 では, 中間層のスパース化について検討を行い, 温度付き Softmax 関数を用いた場合が最もスパース化されていることがわかった. しかし, 表 2 を見ると, 温度付き Softmax 関数を用いた手法は原型モデル (正則化や温度付き Softmax 関数を用いてないモデル) より正解率が向上しているとは言えない. 仮説として, スパース化ができていなくても関わらず推定精度が原-モデルに比べて向上しないのは, 単属性ベクトル推定の誤推定が起きていると考えられる. ここでは, その誤推定の改善のための考察をする.

図 7 は, 原型モデルと温度付き Softmax 関数を用いた手法の推定精度の差をグラフにしたものである. 横軸が未知クラスの数, 縦軸は温度付き Softmax 関数を用いた手法の推定精度を基準に原型モデルの推定精度を引き算した結果を示している. ここから, 常に推定精度が劣っているわけではなく, 未知クラス数が 5 つの場合のみ原型モデルより推定精度が優れていることがわかる. この理由として, 学習データの属性サンプル数に注目した.

表 3 は, 学習データに含まれる基底動作のサンプル数を示す. この表から, 未知クラス数が 5 つのときのみ,

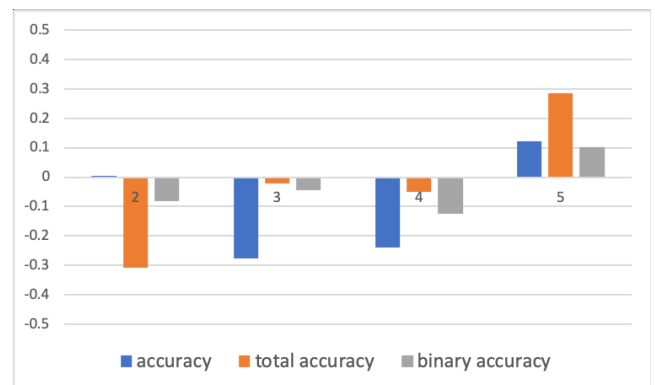


図 7: 温度付き Softmax 関数と原型モデルの正解率の差. 縦軸は各正解率の差の大きさであり, 横軸は未知クラスの数である. 各値は温度付き Softmax 関数を用いた場合の値から何も用いなかった場合 (原-モデル) の値を引いたものである.

表 3: 未知クラス数と学習に利用可能な属性数の対応表. 例えば未知クラス数が 2 つのときは, wave_walk の属性を含むクラスが 3 つ, wave_run の属性を含むクラスが 3 つ, というように見る.

未知 クラス数	wave_ walk	wave_ run	wave_ stay	wave_ sitting	wave_ lie
2	3	3	3	3	4
3	3	3	3	2	3
4	2	3	2	2	3
5	2	2	2	2	2

基底動作のサンプル数が均一であることがわかる. つまり, 単属性ベクトルの推定において, サンプル数の偏りが誤推定に影響を及ぼしている可能性がある. この解決策として, 一般的に (解決策 1) データ数の少ないクラスを誤答した時のペナルティを重くする方法 (cost sensitive learning), (解決策 2) 学習前にサンプル数が均一になるように調整するの 2 点が考えられる. しかし, 中間層に属性の正解ラベルは与えられていないので解決策 1 はできない. 今後解決策 2 の検討を行っていく.

また, 本論文で結果は示していないが, 温度付き Softmax 関数を用いた手法の初期値を変更すると, 推定精度が大き

く変わってしまうという問題がある。つまり、安定した結果が出ないため、その問題についても今後検討していく。

7. まとめ

本稿では、特徴量設計が人の手で行われること、およびセンサ行動認識におけるデータ収集の問題を解決するために提案されている深層学習と Zero-shot 学習法を組み合わせたモデルを検証した。検証方法は（検証 1）時間窓の調整、（検証 2）正則化・活性化関数の検証、の 2 つである。検証の結果、検証 1 では時間窓を調整することで各正解率を向上させることがわかった。また検証 2 では温度付き Softmax 関数を活性化関数に用いることで、モデル内部の各ニューラルネットワークモデルが推定する属性ベクトルのスパース化が可能であることがわかった。また、検証 1 について、CNN 層の出力と各ニューラルネットワークモデルの出力を可視化することで、時間窓を調整することで単純波（基底的な動作）の認識を改善することができることがわかった。また検証 2 について、学習に利用可能な属性数と単属性ベクトルの推定を考察し、利用可能な属性数の偏りが影響していると考えた。今後は、学習に利用可能な属性数を調整して、利用可能な属性数の偏りと単属性ベクトルの推定との関連性を調べる。また本稿では人工データを用いたが、今後は SAPB-NN モデルが人工データだけでなく、実際に現実の環境で得られたセンサデータにも有効であるかどうかを調べる。

参考文献

- [1] : 温度付き softmax 覚書, , 入手先 (<https://qiita.com/nkriskeeic/items/db3b4b5e835e63a7f243>) (参照 2019-1-23).
- [2] Cheng, H. T., Griss, M., Davis, P., Li, J. and You, D.: Towards zero-shot learning for human activity recognition using semantic attribute sequence model, UbiComp 2013 - Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, No. 2, pp. 355–358 (2013).
- [3] Lampert, C. H., Nickisch, H. and Harmeling, S.: Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer, 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009, Vol. 2009 IEEE, pp. 951–958 (2009).
- [4] Lara, Ó. D. and Labrador, M. A.: A survey on human activity recognition using wearable sensors, IEEE Communications Surveys and Tutorials, Vol. 15, No. 3, pp. 1192–1209 (2013).
- [5] Wang, J., Chen, Y., Hao, S., Peng, X. and Hu, L.: Deep learning for sensor-based activity recognition: A survey, Pattern Recognition Letters, Vol. 119, pp. 3–11 (2018).
- [6] 竹下昌志, 松木 萌, 井上創造: ゼロショット行動認識のための中間表現の探索, 日本知能情報ファジィ学会九州支部学術講演会予稿集, pp. 6–9 (2019).
- [7] 松木 萌, 井上創造: 未学習行動推定のための Zero-shot 学習法における精度向上の試み, 人工知能学会全国大会論文集, Vol. JSAI2017, pp. 2C21–2C21 (2017).

- [8] 富岡亮太: スパース性に基づく機械学習, 講談社 (2015).