

Parallel residual adapterを用いた マルチタスク学習によるジェスチャー識別

Gesture Recognition by Multi-task Learning using Parallel Residual Adapters

鈴木 乃依瑠^{*1} 菊井 浩祐^{*2} 伊藤 勇太^{*3} 鹿島 久嗣^{*1} 山田 誠^{*1}
Noeru Suzuki Kosuke Kikui Yuta Itoh Hisashi Kashima Makoto Yamada

^{*1}京都大学 ^{*2}慶應義塾大学 ^{*3}東京工業大学
Kyoto university Keio University Tokyo Institute of Technology

The photo reflective sensor (PRS) is a tiny distant-measurement module which is widely used in wearable user-interface. A issue of such wearable PRS devices is the performance degradation when a user data is not included in the training set (we call the inter-user setup). Moreover, the recognition accuracy also degrades when the same user re-wears the device (we call the intra-user setup). In this paper, we introduce the multi-task learning algorithm using the Parallel residual adapters as a method to perform additional training at high speed using a small amount of additional data. In the experiments, we demonstrate the performance improvement particularly in intra-user setup with the data of an actual PRS device.

1. はじめに

近年、普及が広がり注目を集め始めているウェアラブルデバイスであるが、それを構成する電子部品の一つに Photo reflective sensor (PRS) というセンサーがある。小型で安価であるという利便性から、PRS を利用することで様々なウェアラブルデバイスのユーザインタフェースを設計することができる。

このようなウェアラブルデバイスで計測したデータに対して機械学習の手法を適用する場合、避けられない問題がいくつか存在する。まず、学習データに含まれない新規ユーザのデータに対する識別精度が低下してしまうという問題がある。これはデータのユーザ依存性が高いことに起因する。また、ユーザがデバイスを再装着してデータを計測した場合にも装着具合の差によってデータの分布が変化し、識別精度が低下することがある。したがって、ユーザが変わることやユーザがデバイスを再装着することによる識別モデルのパフォーマンス低下を防ぐことが実用において重要な課題となってくる。

この問題を解決するため、追加で収集したデータを用いたモデルの追加学習を行うことを考える。しかし、大量のラベル付きデータをユーザごとに収集するのは多くのコストがかかってしまう。また、ウェアラブルデバイスに搭載される CPU は処理速度が遅く、大量データでの追加学習を行うことは現実的ではない。

したがって本研究では、少量の追加データを用いて高速に追加学習を行うための手法として、Parallel residual adapter [Rebuffi 18] によるマルチタスク学習を導入する。具体的には、識別モデルをユーザ間で共通のパラメタを学習する部分（ベースの分類器）とユーザ特有のパラメタを学習する部分 (Parallel residual adapter) に分け、まず大量のデータを用意できるユーザに関して全体の学習を行った後、少量のデータを用意するユーザに対して Parallel residual adapter のみ追加学習を行う。Parallel residual adapter のパラメタ数は共通のパラメタ数よりはるかに少ないため、高速な追加学習が可能となる。

実験では PRS を使用した Head Mounted Display (HMD) で計測した実データを用いて、新規のユーザがデバイスを使用する場合 (Inter-user) と既存のユーザがデバイスを再装着して使用する場合 (Intra-user) の二つの問題設定についていくつか

の比較手法と提案手法の比較を行った。その結果、Inter-user においては提案手法が既存手法に精度の面で劣ったが、Intra-user においては既存手法を上回る精度を高速な学習で実現できることを示した。

本研究における貢献は、PRS を用いたウェアラブルデバイスのジェスチャー識別に対してマルチタスク学習を導入し、少量の追加データを用いて短時間で効率的に追加学習を行うための手法を提案したこと、そしてその有用性を実データを用いて実験的に示したことが挙げられる。

2. 関連研究

PRS を用いた研究はいくつか存在する。Yamashita らは複数の PRS を HMD のフレームに取り付け、フレームから頬までの距離を PRS で計測することによって頬の形の変化を検出し、頬の上で指を使って行われる複数のジェスチャーのデータに対して support vector machine (SVM) による識別を行った [Yamashita 17]。また、Kikui らは本研究と同様の問題設定において fine-tuning によるドメイン適応を導入し、CheekInput [Yamashita 17] で計測したデータを用いてジェスチャー識別を行い、新しいユーザのデータに対する識別精度が 63.26% から 80.06%、既存ユーザのデバイス再装着のデータに対する識別精度が 68.96% から 87.43% に向上することを実験的に示した [Kikui 18]。

3. 問題設定

本研究におけるジェスチャー識別問題とは、ジェスチャーを計測したセンサーの時系列データからそのジェスチャーのラベルを予測する問題である。

集合 $\mathcal{X} \subset \mathbb{R}^{S \times F}$ をセンサーデータの集合 (S はセンサー数、 F はフレーム数)、 $\mathcal{Y}$ をラベル集合とする。本研究では、元ドメインの大量のラベルつき学習データ $D_s = \{(X_i^{(s)}, y_i^{(s)}) | X_i^{(s)} \in \mathcal{X}, y_i^{(s)} \in \mathcal{Y}\}_{i=1}^{N_s}$ を用いて学習した分類器 $H: \mathcal{X} \rightarrow \mathcal{Y}$ に対して、目標ドメインの少量の追加のラベルつき学習データ $D_t = \{(X_i^{(t)}, y_i^{(t)}) | X_i^{(t)} \in \mathcal{X}, y_i^{(t)} \in \mathcal{Y}\}_{i=1}^{N_t}$ を用いて追加学習を行うことでテストデータに対する精度を上げることを目的とする。なお、 $N_t \ll N_s$ である。

連絡先: 鈴木 乃依瑠, suzuki.noeru.47c@st.kyoto-u.ac.jp

追加の少量学習データ D_t 及びテストデータに関して、以下の二つの設定を想定する。

- 大量の学習データ D_s には含まれない新しいユーザのデータ (以下この設定を Inter-user と呼ぶ)。
- 大量の学習データ D_s に含まれるユーザがデバイスを再装着して計測したデータ (以下この設定を Intra-user と呼ぶ)。

4. 提案手法

4.1 Convolutional Neural Network

本研究ではベースとなる分類器として Convolutional Neural Network (CNN) を採用した。CNN とは、画像認識などに使われる代表的なニューラルネットワークの一つで、いくつかの畳み込み層、pooling 層と全結合層から成る。畳み込み層は元の画像にフィルタを畳み込むことで特徴マップを作成する層で、これにより領域ベースでの特徴抽出が可能となる。pooling 層は畳み込み層で抽出した特徴マップを重要な情報を残しながら圧縮する層で、これらの処理により特徴の位置感度が低下し、画像の移動や変形に対する頑健性が高くなる。全結合層では畳み込み層とプーリング層で抽出した特徴から最終的な分類を行う。

本研究では畳み込み層と pooling 層がそれぞれ二層と全結合層から成る CNN を構築した。pooling 層には小領域の最大値を選択する手法である max-pooling を採用した (したがって以下 max-pooling 層と呼ぶ)。

4.2 Parallel residual adapters

本節では、まず Rebuffi らの提案したドメイン特有のパラメタを学習するためのモデルである Parallel residual adapter [Rebuffi 18] の概要について説明し、次に本研究における工夫点について述べる。最後に追加学習を行う際の Parallel residual adapter の初期値について述べる。

Parallel residual adapter は、ベースとなる分類器の畳み込み層に対し並列に接続されるフィルタサイズ 1×1 の畳み込み層で、以下のように定義される。

$$q = f * p + \text{diag}(A) * p$$

ここで p は元の畳み込み層への入力、 q は次の max-pooling 層への出力で、 f は元の畳み込みフィルタ、 $*$ は畳み込み演算子である。畳み込み層への入力チャンネルを C 、出力チャンネルを D とすると、Parallel residual adapter のフィルタセットは $A \in \mathbb{R}^{C \times D}$ となる。 diag は上記の式の第一項と第二項の行列のサイズを合わせて加算ができるようにするための関数であり、 $f \in \mathbb{R}^{L \times L \times C \times D}$ (f のフィルタサイズが $L \times L$) とすると $\text{diag} : \mathbb{R}^{C \times D} \rightarrow \mathbb{R}^{L \times L \times C \times D}$ で、 $\text{diag}(A)$ の (u, v, i, j) 成分を $[\text{diag}(A)]_{uvij}$ 、 A の (i, j) 成分を A_{ij} とすると、以下のよう定義される。

$$[\text{diag}(A)]_{uvij} = \begin{cases} A_{ij} & (u = v = (L + 1)/2) \\ 0 & (\text{otherwise}) \end{cases}$$

f をドメイン共通のパラメタを学習するパラメタ、 A をドメイン特有のパラメタを学習するパラメタと見なし、ドメインごとに Parallel residual adapter を用意し、切り替えて学習、識別を行う。具体的にはまず、大量の学習データ D_s を用いて、 D_s に含まれるドメインの A を f と共に学習する。次に少量

の追加学習データ D_t を用いて、 f を固定したまま D_t に含まれるドメインの A を学習する。これにより少ない追加データによる短時間での学習が可能となる。識別の際は入力データのドメインに対応する Parallel residual adapter の A を用いて行う。本研究でベースとなる CNN は畳み込み層が二層であるので、両方に Parallel residual adapter を接続する。したがって、ユーザー一人あたり二箇所の Parallel residual adapter を学習することとなる。バッチ学習を行う場合、一つのバッチに含まれるデータは全て同一ユーザのデータである必要があることに注意されたい。本研究においては D_s での学習時、ユーザ共通のパラメタの学習が特定のユーザに対して偏ることがないようランダムな順番でユーザを選んで学習した。

次に、本研究における Parallel residual adapter の工夫点について述べる。本研究において実験に使用するデータはセンサー数 10、フレーム数 180 となっており、フレーム数の方向にデータが偏っている。したがって、データの端を切り落とすこととなる上記の $\text{diag}(A)$ をそのまま用いるとセンサー数の方向で失われるデータ量が多いと考えた。したがって代わりに p を A で畳み込んだ後、 f と同じフィルタサイズの max-pooling を適用することによって失われるデータ量を抑えつつ第一項の行列と同じサイズとなるようにした。max-pooling を関数 mp として、以上の操作を式で表すと以下ようになる。

$$q = f * p + mp(A * p)$$

最後に、追加学習時の A の初期値について述べる。Inter-user においては大量の学習データ D_s に含まれない新しいユーザについて学習するので、ランダムな初期値から学習を行う。Intra-user においては D_s と D_t に含まれるユーザが共通であるので、 D_s で学習した A の値が存在する。したがって、ランダムな初期値から学習を行う場合と D_s で学習済みの A の値から学習を行う場合の二通りが考えられる。実験において両者の学習結果について比較を行う。

5. 実験

5.1 データセット

本研究では実験のデータセットとして、Kikui ら [Kikui 18] が CheekInput [Yamashita 17] を用いて計測した実データを利用した。具体的には PRS を取り付けた HMD を用いて計測したジェスチャーの時系列データである。ユーザは予め決められた 4 つのジェスチャーを左頬の上で指を動かして行い、10 個の PRS がその頬の形を計測する。4 つのジェスチャーは以下の通りである。

1. Line 一直線に動かす。
2. Caret V を横に倒した形に動かす。
3. Circle 円を描くように動かす。
4. Stairs 階段状 (2 段) に動かす。

各ジェスチャーは 6 秒間 (30Hz で 180 フレーム) の間に行われる。被験者となるユーザは 6 人で、各ユーザはランダムに指示される順にそれぞれのジェスチャーを 30 回ずつ行う試行 (1 回目) を行った後、3 分のインターバルを挟み HMD を再装着して同じ試行 (2 回目) を行う。したがってユーザ 1 人あたりのデータ数は 240 となる。データの値は 0 から 1023 で、最大値で割ることで正規化を行っている。

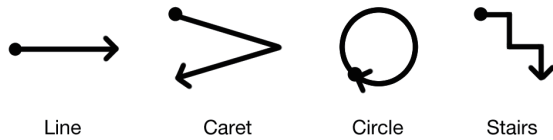


図 1: 4 つのジェスチャー [Kikui 18]

表 1: データの分割

設定	D_s	D_t 及びテストデータ
Inter-user	4 人のユーザのデータ	2 人のユーザのデータ
Intra-user	6 人の 1 回目のデータ	6 人の 2 回目のデータ

次に、3. で述べた二つの設定それぞれに関してデータの分け方を説明する。Inter-user においては、ユーザ 6 人中 4 人（以下学習ユーザと呼ぶ）のデータを大量の学習データ D_s 、残り 2 人（以下テストユーザと呼ぶ）のデータを少量の学習データ D_t 及びテストデータとして用いる。 D_t のサイズは各テストユーザについて 5、10、20、30、40、50、60 とし、残りをテストデータとする。Inter-user の場合は 1 回目と 2 回目のデータの区別をしないので、5 以外の場合は 1 回目と 2 回目のデータから半数ずつを取る。5 の場合はデータサイズが小さいので、ジェスチャーの種類数が 4 であることを考慮して 1 回目のデータから全種類のジェスチャーを含むようにデータを取る。

Intra-user においては、ユーザ 6 人の 1 回目の試行のデータを D_s 、2 回目の試行のデータを D_t 及びテストデータとして用いる。 D_t のサイズは各テストユーザについて 5、10、20、30、40、50、60 とし、残りをテストデータとする。データの分け方をまとめたものを表 1 に示す。

5.2 ベースライン

本節では、提案手法と比較するベースラインとして用いた手法について説明する。

5.2.1 CNN(D_s)

Parallel residual adapter を接続しないベースの CNN を用いる。 D_s を用いて学習した後、追加学習を行わずにテストデータの精度を調べる。したがってこの手法においては D_t は存在せず、 D_s 以外のデータが全てテストデータとなる。

5.2.2 CNN(D_t)

Parallel residual adapter を接続しないベースの CNN を用いる。この手法においては D_s での学習は行わず、 D_t のみで最初から学習を行った場合のテストデータの精度を調べる。ユーザごとにモデルを用意し、別々に学習とテストを行う。

5.2.3 CNN($D_s + D_t$)

Parallel residual adapter を接続しないベースの CNN を用いる。 D_s と D_t を混ぜ合わせたものを学習データとして最初からモデルの学習を行い、追加学習は行わずにテストデータの精度を調べる。ユーザごとにモデルを用意し、別々に学習とテストを行う。

5.2.4 CNN(fine-tuning)

Kikui ら [Kikui 18] によって提案された手法である。Parallel residual adapter を接続しないベースの CNN を用いる。 D_s

を用いて学習した後、追加学習として D_t を用いて CNN の全結合層の fine-tuning を行う。 D_s での学習は CNN(D_s) で学習したモデルを利用し、追加学習とテストはユーザごとに別々に行う。

5.2.5 Parallel residual adapters (追加学習なし)

この手法は Intra-user のみに適用する。提案手法と同様の Parallel residual adapters を接続した CNN を D_s を用いて学習した後、追加学習を行わないでテストデータの精度を調べる。したがってこの手法も CNN(D_s) と同様に D_t は存在しない。

5.3 実験結果

二つの問題設定それぞれについて実験結果を示す。本研究では評価指標としてテストデータの正解率 (Accuracy) を用いる。各手法に対して 1 回の学習につき 200 エポックを回し、10 回学習を実行したものの平均を取る。ユーザごとに別々に学習、テストをしている手法についてはさらに全ユーザの平均を取る。なお、表における PRA とは Parallel residual adapter の略である。

5.3.1 Inter-user

Inter-user における実験結果を表 2 に示す。 D_t のサイズが 20 以上になると追加学習を行わない CNN(D_s) に対し提案手法が勝るが、CNN(fine-tuning) には及ばない結果となった。CNN(D_t) が D_t のサイズ 60 において CNN(D_s) を上回るのに対し提案手法は 20 で上回るの、追加のデータで最初から学習し直すよりも少ないデータ量で高い精度を出せることがわかる。また、CNN($D_s + D_t$) が一番高い精度を出しているが、この手法は学習のデータ量が多く他の手法に比べて大幅な時間がかかるため実用性という点においては短時間で追加学習が行える提案手法の方が優れているといえる。

5.3.2 Intra-user

Intra-user における実験結果を表 3 に示す。PRA (ランダム初期値) がランダムな値を初期値として A を学習した場合、PRA (学習済み初期値) が D_s で学習した値を初期値として A を学習した場合である。

PRA (ランダム初期値) は D_t のサイズが 40 を超えると CNN(D_s) を上回るが、CNN(fine-tuning) とはあまり大差ない結果となった。一方で PRA (学習済み初期値) は D_t の全てのサイズにおいて CNN(D_s) を上回っており、PRA (ランダム初期値) 以上の追加学習の効果が見られる。また CNN(fine-tuning) に対しても全てのサイズにおいて上回っていて、学習済みの値から学習する場合は D_t が小さくても高い精度を出すことができるということがわかる。Inter-user 同様、CNN($D_s + D_t$) が高い精度を出しているが、これも学習に時間がかかるため実用的ではないと思われる。

5.4 考察

以上の実験結果について、Inter-user と Intra-user の結果を比較しながら考察を行う。

Intra-user では既存手法を上回った提案手法が Inter-user では既存手法を下回ってしまった原因の一つとして、まず二つの問題設定における D_s と D_t の性質が異なることが挙げられる。Inter-user では D_s と D_t に含まれるユーザが異なるのに対して Intra-user では共通であるため、 D_s と D_t の類似度が高い。このことは、CNN(D_s) の実験結果において Inter-user では 80.5% まで下がってしまう精度が Intra-user では 95.5% に留まることからわかる。

また、Intra-user において PRA (ランダム初期値) が PRA (学習済み初期値) に劣る結果となったことから、Parallel resid-

表 2: Inter-user 実験結果

手法	D_t のサイズ								実行時間 (s/user)
	なし	5	10	20	30	40	50	60	
CNN(D_s)	0.805	-	-	-	-	-	-	-	-
CNN(D_t)	-	0.517	0.630	0.710	0.764	0.840	0.762	0.843	3.928
CNN($D_s + D_t$)	-	0.811	0.855	0.897	0.913	0.926	0.932	0.952	125.370
CNN(fine-tuning)	-	0.731	0.809	0.843	0.907	0.920	0.921	0.906	3.881
PRA	-	0.761	0.777	0.856	0.841	0.874	0.881	0.885	5.737

表 3: Intra-user 実験結果

手法	D_t のサイズ								実行時間 (s/user)
	なし	5	10	20	30	40	50	60	
CNN(D_s)	0.955	-	-	-	-	-	-	-	-
CNN(D_t)	-	0.629	0.755	0.844	0.890	0.873	0.906	0.906	3.984
CNN($D_s + D_t$)	-	0.969	0.962	0.981	0.986	0.984	0.972	0.991	98.641
CNN(fine-tuning)	-	0.878	0.914	0.943	0.933	0.948	0.963	0.969	3.792
PRA(追加学習なし)	0.938	-	-	-	-	-	-	-	-
PRA(ランダム初期値)	-	0.916	0.917	0.928	0.936	0.956	0.955	0.964	6.209
PRA(学習済み初期値)	-	0.960	0.972	0.976	0.974	0.976	0.976	0.979	5.309

ual adapter の初期値が少なからず役割を果たしていることがわかる。したがって、Intra-user においては学習の際にユーザの区別をしない CNN(fine-tuning) よりもユーザ特有のパラメタを学習する提案手法の方がより各ユーザの特徴を学習し、PRA (学習済み初期値) において追加学習の際にその情報を Parallel residual adapter の初期値として得ることができるため精度の向上に貢献しているのだろうと考えられる。逆に、Intra-user の PRA (ランダム初期値) や Inter-user では Parallel residual adapter の初期値がランダムであるために情報を得られず、追加学習でテストユーザの特徴を学習しきれなかったのだろうと思われる。

次に実行時間に関する考察を行う。5.3 節でも述べた通り、一番高い精度を出している CNN($D_s + D_t$) は学習時間が多くかかるため実用性に欠ける。また、Inter-user、Intra-user 共に CNN(fine-tuning) の方が提案手法より学習時間が短い。追加学習時の学習パラメタ数は CNN(fine-tuning) に比べ提案手法の方が少ないが、重みの更新を行う際に提案手法の方がモデルの構造が複雑であるため時間がかかってしまうのではないかとと思われる。この差は D_t のサイズが大きくなるにしたがって大きくなると考えられるが、本研究の目的は小さいサイズの D_t で追加学習することであるので、実用において支障が出る程の差ではないと考えられる。

6. 結論

本論文では、PRS を用いたウェアラブルデバイスによるジェスチャー識別問題に取り組み、Parallel residual adapter によるマルチタスク学習を導入したモデルを提案した。さらに新規のユーザがデバイスを使用する場合 (Inter-user) と既存のユーザがデバイスを再装着して使用する場合 (Intra-user) の二つの問題設定において実データを用いた実験を行い、少量データによる短時間の追加学習で精度の向上を実現できることを示した。また、特に Intra-user において提案手法が既存手法を上回る精度を出せることを示した。

今後の展望としてはまず、提案手法が既存手法を上回ることでできなかった Inter-user において、より効果的に追加学習

を行うことができるような Parallel residual adapter の A の初期値を求めることが挙げられる。例えば学習ユーザとテストユーザのデータの分布などを解析し、識別するテストユーザに似たデータを持つ学習ユーザの A の値を初期値とすることが考えられる。あるいは、ランダムな初期値からの追加学習であっても少ないデータ量でテストユーザ特有の特徴を十分学習できるように Parallel residual adapter の構造の改良を行うことが挙げられる。また Parallel residual adapter について、今回関数 $diag$ の代わりに max-pooling を用いているが、本研究においてはこれらの違いによる性能比較を行っていない。したがってどちらがより有用か、あるいはほかに有用な方法はないかなどの検証を行う必要がある。

7. 謝辞

本研究では慶應義塾大学の杉本麻樹准教授と杉浦裕太講師にデータ提供を頂きました。厚くお礼申し上げます。

参考文献

- [Yamashita 17] Yamashita, K., Kikuchi, T., Masai, K., Sugimoto, M., Thomas, B. and Sugiura, Y.: Cheek-Input: Turning your cheek into an input surface by embedded optical sensors on a head-mounted display, *Proceedings - VRST 2017*, Vol. Part F131944, Association for Computing Machinery (2017).
- [Rebuffi 18] Rebuffi, S., Bilen, H. and Vedaldi, A.: Efficient parametrization of multi-domain deep neural networks, *CoRR*, Vol. abs/1803.10082 (2018).
- [Kikui 18] Kikui, K., Itoh, Y., Yamada, M., Sugiura, Y. and Sugimoto, M.: Intra- /Inter-user Adaptation Framework for Wearable Gesture Sensing Device, *Proceedings of the 2018 ACM International Symposium on Wearable Computers, ISWC '18*, ACM, pp. 21-24 (2018).