

卒業論文

LSTM を用いた
ソフトテニスのサービス予測モデル

指導教官 村上 陽平 准教授

立命館大学 情報理工学部
先端社会デザインコース 4 回生
2600180039-5

上野 颯人

2022 年度（秋学期）卒業研究 3（CH）
令和 5 年 1 月 31 日

LSTM を用いたソフトテニスのサービス予測モデル

上野 颯人

内容梗概

数年前までは日本に一人もソフトテニスのプロ選手が存在しなかったが、現在は多くなりつつある。賞金のある大会や個人にスポンサーが参画することが増え、ソフトテニスの強化が加速している。特に、プロ選手や日本トップの選手の練習プログラムではサービスの強化が組み込まれている。サービスは試合中で唯一ボールが止まっている状態から開始でき、確実に先攻することが可能なため、重視されている。ただし、練習で良いフォームを見つけたとしても、次の日に調子が合わないことがあり、サービスを課題にしているプロの選手も多い。そのために、プロ選手は専有のテニスコートでサービスの練習を繰り返すことで、インかフォルト、レットなどのフィードバックを受けてサービスのフォームを固めている。しかしながら、多くの選手には専有のテニスコートがないため、限られた練習時間にサービスに多くの時間を割くことができないのが現状である。さらに、サービスを打つ場所やコースによってフォームの形も変化するため、それぞれごとに多くのサービス練習を課すことができない。さらに、フォームは人によって異なるため、複数人同時に画一的な指導ができない。

そこで、本研究では個々の選手のサービスの動画から LSTM を用いてサービスの予測モデルを生成し、限られた練習場所でもサービスが入ったかどうかを予測可能とすることを目的とする。具体的には、サービスの動画から OpenPose を用いて姿勢データを取得し、姿勢データとサービスの位置、狙ったコースのデータを入力データ、サービスの結果のイン、フォルト、レットを教師信号として学習を行う。本手法の実現にあたり、取り組むべき課題は以下の2点である。

サービスの姿勢データの拡充

サービスの位置と狙ったコースにおいて、イン、フォルト、レットというサービスの結果を予測するには、大量の姿勢データが必要である。特に、サービスの位置と狙うコースの種類によってフォームが異なるため、多様な姿勢データを収集する必要がある。

予測モデルの妥当性検証

サービスの結果がイン、フォルト、レットの3種類あるため、予測モデルはランダムに予測した場合の 0.33 よりも精度を向上できている必要がある。

る。

前者の課題に対しては、ソフトテニスにおける基本のサービスを打つ位置をデュースサイドとアドバンテージサイドの2箇所とし、クロスとセンターの各コースに20本入るまで姿勢データを183回分の動画で収集した。その動画からOpenPoseによって得られた姿勢データを、ランダムに得られた値だけ、水平、垂直方向に移動させることで、データを拡充した。これにより、サービスの結果が変わらないが姿勢データを増やすことが可能になり、学習データの拡充に繋がる。

後者の課題に対しては、学習に用いていないサービスの姿勢データ183件により、獲得した予測モデルの正確性を評価した。さらに、データの拡充の効果を検証するために、データの拡充サイズを少しずつ大きくしながら、予測モデルを評価した。本研究の貢献は以下の通りである。

サービスの姿勢データの拡充

動画から獲られた姿勢データをシーズデータとし、水平、垂直方向にランダムに移動させることで、2000回分のサービスデータを生成し、訓練データを拡充した。

予測モデルの妥当性検証

データ拡充前の訓練データのみの場合、正確性が0.29であったが、データ数を増やすにつれて正確性が向上し、データ数2000件で正確性が0.42となった。ランダム予測の0.33よりも正確性が向上したため、拡充したデータによって構築したモデルの妥当性を確認した。

LSTM-based Service Prediction Model for Soft Tennis

Hayato UENO

Abstract

A few years ago, there was not a single professional soft tennis player in the Japan, but now the number is increasing. Sponsors are increasingly participating in tournaments and individuals with prize money, and the strengthening of soft tennis is accelerating. In particular, the training programs of professional players and top Japan athletes incorporate enhanced services. Service is emphasized because it is the only time in the game that the ball can be stopped and it is possible to attack first. However, even if you find a good form in practice, you may not be in good shape the next day, and many professional players are tasked with service. For this reason, professional players repeatedly practice their services on their own tennis courts to receive feedback from in, faults, nets, etc. To solidify their form of service. However, many players do not have their own tennis courts, so they cannot devote much time to services during their limited practice time. In addition, since the shape of the form changes depending on the place and course of the service, it is not possible to impose many service exercises for each one. Furthermore, since the form differs from person to person, it is not possible to teach multiple people uniformly at the same time. Therefore, the purpose of this study is to generate a service prediction model using LSTM from the videos of individual players' services, and to predict whether or not the service has been entered even in a limited number of practice areas. Specifically, posture data is acquired from the video of the service using OpenPose, and learning is performed using the posture data and the position of the service, the target course data as input data, and the in, fault, and net of the service result as teacher signals. In order to realize this method, there are two issues to be addressed.

Expansion of service attitude data

In order to predict the results of services such as in, fault, and netlet in the position of the service and the target course, a large amount of posture data is required.

Validation of predictive models

Since there are three types of service results: in, fault, and net, the prediction model must be able to improve the accuracy of 0.33 when predicting randomly. For the former task, the basic service in soft tennis was set to two positions, the deuce side and the advantage side, and posture data were collected from 183 videos until 20 shots were entered on each course of cross and center. The posture data obtained by OpenPose from the video was expanded by moving the posture data horizontally and vertically only by the randomly obtained value. In response to the latter problem, we evaluated the accuracy of the obtained prediction model based on 183 attitude data of services not used for learning. Furthermore, in order to verify the effect of data expansion, we evaluated the prediction model while gradually increasing the size of the data expansion. The contributions of this study are as follows.

Expansion of service attitude data

By using the posture data captured from the video as seed data and moving it randomly in the horizontal and vertical directions, service data for 2000 sessions was generated and the training data was expanded.

Validation of predictive models

In the case of only the training data before the data expansion, the accuracy was 0.29, but the accuracy improved as the number of data increased, and the accuracy was 0.42 for 2000 data. Since the accuracy was improved compared to the random prediction of 0.33, the validity of the model constructed using the expanded data was confirmed.

LSTM を用いたソフトテニスのサービス予測モデル

目次

第 1 章 はじめに	1
第 2 章 姿勢推定を用いたスポーツ支援	3
2.1 姿勢推定手法	3
2.1.1 OpenPose	3
2.2 関連研究	4
2.3 ソフトテニスのサービス	5
2.3.1 サービスのルール	5
2.3.2 テニスコート	6
第 3 章 サービス予測モデル	8
3.1 サービスの姿勢データ取得	8
3.1.1 姿勢データの抽出	8
3.1.2 姿勢データの拡充	10
3.2 LSTM による予測モデルの学習	10
3.2.1 LSTM	10
第 4 章 実験	13
4.1 実験データ	13
4.2 評価	13
4.2.1 デュースサイドの正確性	13
4.2.2 アドバンテージサイドの正確性	14
4.2.3 イン, フォルト, ネットの正確性	13
第 5 章 考察	18
第 6 章 おわりに	22
謝辞	23
参考文献	24
付録：サービスの予測モデル	25
A.1 姿勢データのベクトル生成	25

A.2 テストデータの分類.....	26
A.3 テストデータの前処理.....	26
A.4 訓練データの拡充.....	27
A.5 サービスの予測モデルの学習と評価.....	28

第1章 はじめに

数年前までは日本に一人もソフトテニスのプロ選手が存在しなかったが、現在は多くなりつつある。賞金のある大会や個人にスポンサーが参画することが増え、ソフトテニスの強化が加速している。プロの大会では試合の動画をもとに選手のコート内位置を入力データ、打球方向を正解データとして学習を行い、ある選手がどの方向に打つのかを検出することでデータを分析して試合をすることが増えている。一方で、唯一ボールが止まった状態から開始するサービスでは、選手個人やポジションによってフォームが異なるため、指導が困難である。例えば、後衛と呼ばれるコートの後方のポジションのサービスは、サービスを打った後にそのままの位置にいることが多いが、前衛と呼ばれるコート前方のポジションのサービスは、サービスを打った後に前方につかなければならないため、後衛と前衛というポジションの違いだけでも打ち方が変化する。

試合でサービスの確率が高い選手は、サービスゲームをキープできている確率も高い。特に、プロ選手や日本トップの選手の練習プログラムではサービスの強化が組み込まれている。高校や大学でもサービス専門のコーチに指導してもらったところもある。サービスは試合中で唯一ボールが止まっている状態から開始でき、確実に先攻することが可能なため、重視されている。ただし、練習で良いフォームを見つけたとしても、次の日に調子が合わないことがあり、サービスを課題にしているプロの選手も多い。そのために、専有のテニスコートでサービスの練習を繰り返すことで、インかフォルト、レットなどのフィードバックを受けてサービスのフォームを固めている。しかしながら、多くの選手には専有のテニスコートがないため、限られた練習時間にサービスだけに多くの時間を割くことができないのが現状である。さらに、サービスを打つ場所やコースによってフォームの形も変化するため、それぞれごとに多くのサービス練習を課すことができない。さらに、フォームは人によって異なるため、複数人同時に画一的な指導ができない。

そこで、本研究では個々の選手のサービスの動画から LSTM を用いてサービスの予測モデルを生成し、コート以外の練習場所でもサービスが入ったかどうかを予測可能とすることを目的とする。具体的には、サービスの動画から OpenPose を用いて姿勢データを取得し、姿勢データとサービスの位置、狙っ

たコースのデータを入力データ，サービスの結果であるイン，フォルト，レットを教師信号として学習を行う．本手法の実現にあたり，取り組むべき課題は以下の2点である．

サービスの姿勢データの拡充

サービスの位置と狙ったコースにおいて，イン，フォルト，レットというサービスの結果を予測するには，大量の姿勢データが必要である．特に，サービスの位置と狙うコースの種類によってフォームが異なるため，多様な姿勢データを収集する必要がある．

予測モデルの妥当性検証

サービスの結果がイン，フォルト，レットの3種類あるため，予測モデルはランダムに予測した場合の 0.33 よりも精度を向上できている必要がある．以下，本論文では，初めに姿勢推定手法の一つである．

OpenPose に関する説明と OpenPose を用いた関連研究，そしてソフトテニスのサービスについて述べる．次にサービスデータの拡充手法とサービスの予測モデルの構築手法について説明する．その後，構築した予測モデルの検証実験について説明し，実験結果について分析を行う．最後に分析結果に基づき，精度の改善手法などについて考察し，本研究のまとめを述べる．

第2章 姿勢推定を用いたスポーツ支援

2.1 姿勢推定手法

姿勢推定手法とは、動画、静止画から人物の関節点を推定する手法である。推定された関節点を結ぶことで、人物の姿勢を検出することが可能である。近年、ディープラーニングを用いることで、衣服の上からでも特徴点を検出することができるようになり、これまで視覚など人間に頼らざる得なかった領域で、定量的なデータから自動でコーチングやレポートが可能になっている。ディープラーニングを使用することで、モーションキャプチャ用のマーカーや慣性センサーといった動画から特徴量を計測し数値化するために必要であった専用機材や専門知識は不要となり、動画から直接姿勢推定が可能である。ただし、姿勢推定モデルを含めた画像認識では、モデルの学習のために非常に多くの画像データが必要となり、認識精度を上げることが困難である。そこで本研究では、既に大量の画像データによって学習済みの **OpenPose** と呼ばれる姿勢推定技術を用いた。

2.1.1 OpenPose

OpenPose はカーネギメロン大学 (CMU) の Zhe Cao らが考案した、深層学習を用いた人物の姿勢を可視化する手法である [1]。OpenPose は、モーションキャプチャなどの画像の特徴量を検出する機器は不要で、画像データを入

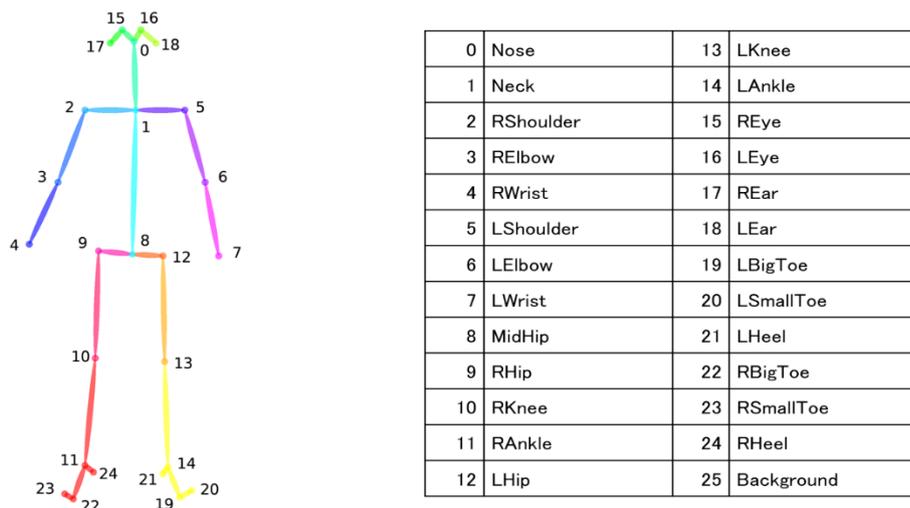


図 1 25 点の関節 (左) と関節の名称 (右)

力するだけで人間の関節点を検出することが可能である。さらに、GPU などの高性能プロセッサを使うことで動画内に同時に複数の人物がいても、リアルタイムに検出することも可能である。従来の技術では画像内の検出に失敗すると関節の抽出が困難であった。外乱の影響を阻止するロバスト性を強化することで改善された OpenPose を用いることで、25 点の関節が取得でき、関節同士を繋いだ姿勢が 2 次元として表すことができる (図 1)。また、0 から 24 の関節点には名称がつけられどの関節が身体のどの部分を示しているのかも理解できる (図 1)。

2.2 関連研究

本節では OpenPose を用いた関連研究を紹介する。ここで挙げる若林は、OpenPose を用いたビーチフラッグスの動作分析を研究している [2]。OpenPose を利用してビーチフラッグスにおける効率の良い起き上がり方法を明らかにすることでパフォーマンス向上だけでなく、ビーチフラッグス界全体としてのレベルアップを図ることが出来るのではないかと考えている。OpenPose で姿勢検出アルゴリズムをライブラリ化したものを動画に適用することで、人体が検出され、複数の特徴点の座標と検出率が得られる。起き上がり動作の流れを 4 分割し、第 1~4 局面とし、特徴を分析し、各特徴点のブレを比較することで、ビーチフラッグスにおける効率の良い起き上がり方法を明らかにしている。起き上がり方法は「ジャパニーズ」と「ポール」の 2 種類があり、どちらに近い起き上がりが向いているのかを最終結果にしている。リアルタイムの複数人 2D 姿勢推定は、それを可能にするための重要な要素であり、画像やビデオの人々を理解するための機械ある。この論文では、複数の人の 2D ポーズを検出するためのリアルタイムアプローチを提示している画像である。提案手法はノンパラメトリック表現を用い、参照しているパートアフィニティフィールド(PAF)として、身体の部分を関連付ける方法を学ぶ画像内の個人である。このボトムアップシステムは、高精度と画像内の人数に関係なく、リアルタイムパフォーマンスであり、PAF と身体部位の位置推定がトレーニングステージ間で同時に改良されている。PAF のみであることを示してお、PAF とボディ部分の位置の両方ではなく、リファインすると、実行時のパフォーマンスと精度の両方が大幅に向上することがわかる。内部に基づく最初のボディとフットを組み合わせたキーポイント検出器を公開した注釈付き

フットデータセットのことであり、組み合わせた検出器は、それらを実行する場合と比較して推論時間を短縮するだけではない。しかしまた個々の各コンポーネントの精度を維持している。この作業は、最初のオープンソースである **OpenPose** のリリースで最高潮に達している。体、足、手を含む複数の人物の **2D** ポーズ検出のためのリアルタイムシステム、と顔のキーポイントを抽出している研究である。

2.3 ソフトテニスのサービス

ソフトテニスのサービスとは、ポイントを始める最初のショットで、スマッシュのように高い打点で打つのが基本であり、空中にボールを投げてからラケットで打たなければならない。なお、サーブを打つ人のことを「サーバー」、受ける人のことを「レシーバー」と呼ぶ。

2.3.1 サービスのルール

ネットを超え、対角線上にあるサービスエリア（サービスコート）に入れる必要がある。サービスエリアには、アドバンテージサイドとデュースサイドがあり、センターラインと右のサイドラインの間の位置から左のサービスボックスに入れることをデュースサイドといい、センターラインと左のサイドラインの間から右のサービスボックスに入れることをアドバンテージサイドという。サービスエリアとはサイドライン、サービスサイドライン、センターライン、ネットで囲われたエリアのことである。サービスエリアに入れることができないと、サーブ失敗となり、フォルトという。サーブは各ポイント 1 回ずつフォルトが許されており、初めのサービスをファーストサービス、二度目のサービスをセカンドサービスという。セカンドサービスをフォルトしてしまうとダブルフォルトといい、失点となる。サーブがネットに触れた後、サービスエリアに入った場合は、サーブの打ち直しを行う。これをレットという。サーブを打つ人はベースラインより後ろでサイドラインとセンターラインの間で打つ。規定のエリアの外に足がついてしまった状態でサービスを打つことをフットフォルトといい、フォルトと同じ扱いになる。硬式と違い、トスを 1 回上げてしまうと、打たなければならない、やり直すことはできない。サーブを打つ位置と方向としてデュースサイドとアドバンテージサイドの 2 カ所がある。デュースサイドはセンターラインと右サイドラインの間から左のサービスエリアに打つ。アドバンテージサイドはセンターラインと左サイドラインの間から右のサービ

スエリアに打つ.

2.3.2 テニスコート

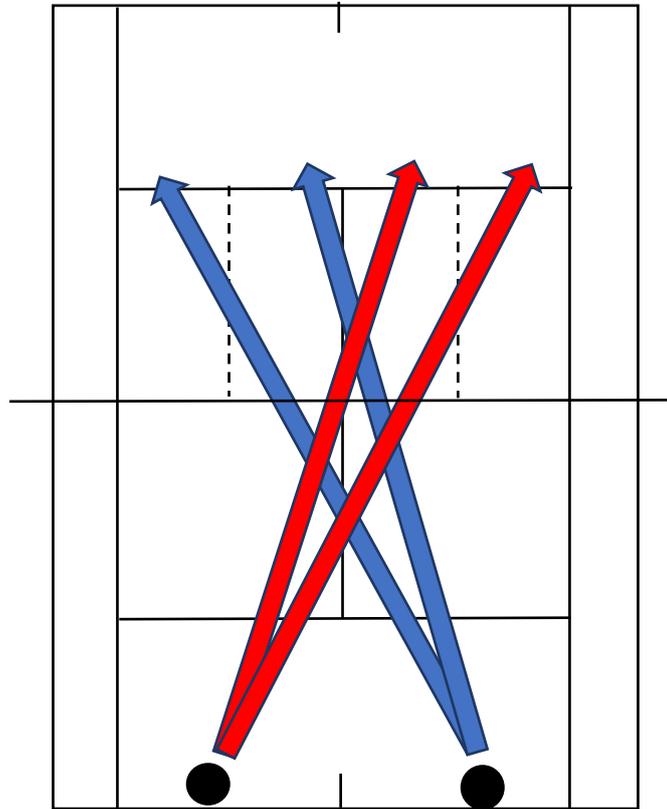


図 2 テニスコート

ソフトテニスのコートについて説明していく (図 2). 黒円はサーバーを表している. 1のラインはサイドライン, 2のラインはサービスサイドライン, 3のラインはサービスライン, 4のラインはセンターライン, 5のラインはベースライン, 6のボックスがサービスボックスである. ソフトテニスは主にセンターとサイドに打ち分けるため, サービスの姿勢も異なる. 本研究では 6のサービスボックスを中心で割り, 全 4 コースのサービスを打つ. デュースサイドはセンターラインと右サイドラインの間から左のサービスエリアに青の軌道で, アドバンテージサイドはセンターラインと左サイドラインの間から右のサービスエリアに赤の軌道で打つ. 図 3 はデュースサイドからサイドラインにサービスを打ったときのフォームである.



図3 デュースサイドからサイドラインへのサーブ

第3章 サービス予測モデル

本章では，ソフトテニスのサービス予測モデルの構築について説明する．数年前までは日本で一人もソフトテニスのプロ選手が存在しなかったが，現在は多くなりつつある．賞金のある大会や個人にスポンサーが参画することが増え，ソフトテニスの強化が加速している．しかしながら，プロ選手になりたいという目標がありながらも多くの選手には専有のテニスコートがないため，限られた練習時間しか確保できないこと現状である．そこで本研究では，コート以外の場所でもサービスの練習ができるように，サービスの動画からサービスのイン，フォルト，ネットを予測する予測モデルを構築する．以下ではソフトテニスのプレーの中でも重要であるサービスに重きを置き，データの取得法とモデルの構築法について述べる．

3.1 サービスの姿勢データ取得

サービスの予測モデルを構築するにあたり，サービスの学習データとして姿勢データを収集する必要がある．そのために立命館大学ソフトテニス部の部員1名に，デュースサイドとアドバンテージサイドの2カ所の位置でサーブを打ってもらい，データを収集した．デュースサイドはセンターラインと右のサイドラインの間から左のサービスエリアに打つことで，アドバンテージサイドはセンターラインと左のサイドラインの間から右のサービスエリアに打つことである．各サービスエリアを中心にサイドとセンターに区切り，サイドを狙うクロスとセンターを狙うセンターの2箇所のサービスコースを設定した．各位置から2コースのサーブを打ち，イン，フォルト，レットを記録した．1つのコースにつきインが20本になるまで打ち続け，その間のフォルトとネットも記録した結果，計183回のサービスのデータを取得した．各コースの結果をまとめた表が表1である．インを0，フォルトを1，ネットを2と示した．

3.1.1 姿勢データの抽出

上記で記述した183回のサービスの動画データから，予測モデルの学習で利用できる姿勢データに変換する必要がある．まず，183件のサービスデータからOpenPoseを用いて25カ所の関節データを抽出し，jsonファイルにデータを生成した(図4)．そのjsonデータには関節ではない情報が含まれているため，jsonデータから25カ所の関節(X,Y)座標のみを抽出しなければなら

ない. この抽出される関節はフレームごとの関節である. 動画はコマおくり
している画像のことであり, このフレーム数を 100 になるように揃えた. 1つ
のフレームからそれぞれの 25 カ所の関節点 (X,Y) 座標により 51 次元の三次
元配列のデータを生成した.



表 1 サービスの結果の記録

図 4 OpenPose により抽出した姿勢データ

1	2	3	4	5	6	7	8	9	10
1	0	1	2	0	0	1	0	1	0
11	12	13	14	15	16	17	18	19	20
2	1	1	1	2	2	2	0	0	0
21	22	23	24	25	26	27	28	29	30
1	0	0	2	0	2	0	0	1	2
31	32	33	34	35	36	37	38	39	40
2	1	1	0	1	0	0	0	1	2
41	42	43	44	45	46				
0	0	2	2	2	0				

3.1.2 姿勢データの拡充

動画を撮る際に手元を固定していないことや、そもそものサーブの位置がずれていることなどがあるため正確な学習が困難である。そのため、ソフトテニスのサービスの予測モデルを構築するには、大量の姿勢データが必要である。サービスを打つ場所や、ボールの位置によって関節の位置も異なるため、大量のデータを収集する事が困難である。さらに、データを多く収集しようとして、大量にサーブを打ったとしても疲労が関わってくるため、正確なデータも得ることが難しい。そこで、183 件のシーズデータから姿勢データの拡充が必要である。そこで、姿勢データからランダムにデータを選び、ランダムに関節位置 (X,Y) の移動方向と移動距離を設定し、全関節位置を設定値に基づいて平行移動させて姿勢データを拡充した。それぞれ拡充するデータ数は 200, 400, 800, 1500, 2000 である。

3.2 LSTM による予測モデルの学習

LSTM とは、「Long Short Term Memory」の略称であり、ニューラルネットワークに使用される層の一つである。上記で構築した、サービスの各関節を LSTM を用いて学習させる。

3.2.1 LSTM

ここでは、LSTM がどのようなものかを説明する。LSTM とは、「Long Shot

Team Memory」を略した言葉で、ディープラーニングが流行する前から存在しており、1997年に原著論文が発表された。まず、ディープラーニングとは人間が脳などを使って自然に行うことをコンピュータに学習させる機会学習の1つである。LSTMは長期的な依存関係を学習することができるRNNの一種である。ニューラルネットワークは、あるデータ入力し、それに基づいた出力が検出される。しかし、従来のニューラルネットワークは各時点での時系列がわからない。これを解決しているのがRNNである。ここ数年ではRNNが活用されており、音声認識、言語モデリングなど、配列やリストに関連する操作に重要になっている。LSTMとRNNは時系列データを入力する事によってそれに対する結果を予測できるようになっている。ここで、LSTMとRNNには違いがあり、RNNは時系列が長くなることで、勾配消失問題が発生する。一方、LSTMは時系列が長くなったとしても、考慮した上で、この問題を解決できる。それは、RNNの構造を維持しながらも、古いアウトプットを次のインプットとして、長期記憶を少しずつかえることができているからである。LSTMには忘却ゲート、入力ゲート、出力ゲートの3つのゲートがある。忘却ゲートは長期記憶を行う上で情報を忘却するときに必要な制御ゲートのシグモイド層である。入力ゲートでは2つの操作を行う。初めに、セル状態で新しい情報を保存する判定をする。次にそのセル状態に加えることができる新しい候補値を作成し、これらを組み合わせる。出力ゲートは、出力するものを判定するゲートであり、入力に基づいて実行する。LSTMは勾配消失問題に対応するために発明され、潤沢な計算リソースと合わせて音声認識・機械翻訳などで目覚ましい性能を見出してきた。しかし、LSTMの重大な欠点としてLSTMの構造自体からは、どのような問題が解けるのか、なぜ解けない問題があるのかを判断できないことが挙げられる。例として、Seq-to-Seq Learningによる機械翻訳モデルでは、LSTMが文章の内部表現を獲得していることは示唆されているが、木構造などによる人間が理解可能な分析は非常に難しくなっている。LSTMを使用する場合は時系列データに対する強力な予測モデルであることや勾配消失問題を解決し、長期の時系列を学習可能になったこと、LSTM単体での使用に留まらず、双方向の複数の組み合わせによって表現力を拡大していることを理解しておくことが重要である。

本研究では、LSTMを用いて、サービスを打つ位置の一データとその時の関節の姿勢データを入力データとし、サービスエリアへのイン、フォルト、ネット

を教師信号として学習を行う。

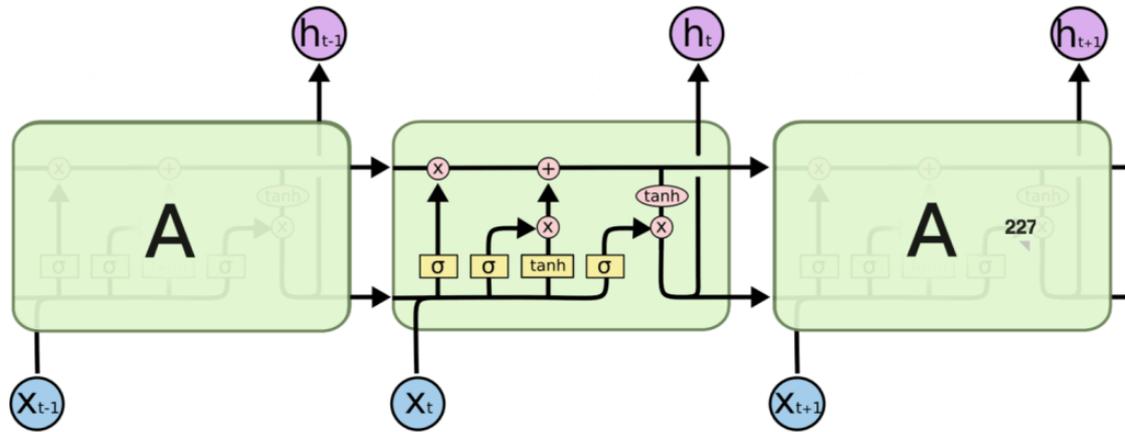


図5 LSTMの構造

第4章 実験

第 3 章では、186 回のサービスデータを抽出し、訓練データのサイズを拡充した。その拡充したデータ数に応じて学習させ、精度を出力する。

4.1 実験データ

全 183 件の姿勢データはアドバンテージサイドのクロス、アドバンテージサイドのセンター、デュースサイドのサイド、デュースサイドのセンターに打ち分けているデータであり、それぞれの位置を 0, 1, 2, 3 とする。183 件の内、89 未満を `test_idx1` (アドバンテージサイド)、89 以上を `test_idx2` (デュースサイド)、お互いを合わせたデータを `pre_test_idx` とし、評価する際のテストデータを構築する。183 件の姿勢データからランダムに 20 件を抽出するが、ランダムなため、姿勢データが偏る場合がある。

4.2 評価

それぞれの位置データとテストデータを学習させることでトレーニングデータを構築することができる。3.1.2 で説明したように、動画を撮る際に手元を固定していないことや、そもそも、位置がずれていることなどがあるため正確な学習が困難であるため、サービスの予測モデルを構築するには、大量の姿勢データが必要である。そこで 183 件のシーズデータからランダムに関節位置 (X,Y) の移動方向と移動距離を設定し、全関節位置を設定値に基づいて平行移動させて姿勢データを拡充した。これをラーニングした結果、各ポジションの評価が得られる。また、全姿勢データから LSTM を用いて予測モデルの精度を 200, 400, 800, 1500, 2000 件のそれぞれのデータ数で精度を比較した。

4.2.1 デュースサイドの正確性

全姿勢データから LSTM を用いて予測モデルの精度を 200, 400, 800, 1500, 2000 件のそれぞれのデータ数で予測精度を抽出した。183 件の姿勢データからランダムに 20 件を抽出し、姿勢データが偏る場合があるため、それぞれのデータ数を 5 回ずつ行った結果、表 2 のようなアベレージが確認できた。これを折れ線グラフとして図 6 で示した。データを拡充しても、精度が上がらず、サービスのイン、フォルト、ネットかに姿勢の特徴が無かった事が分かる。

表 2 デュースサイドのサービスの予測精度

訓練データサイズ	200	400	800	1500	2000
精度	40.78	42.23	43.71	40	41.22

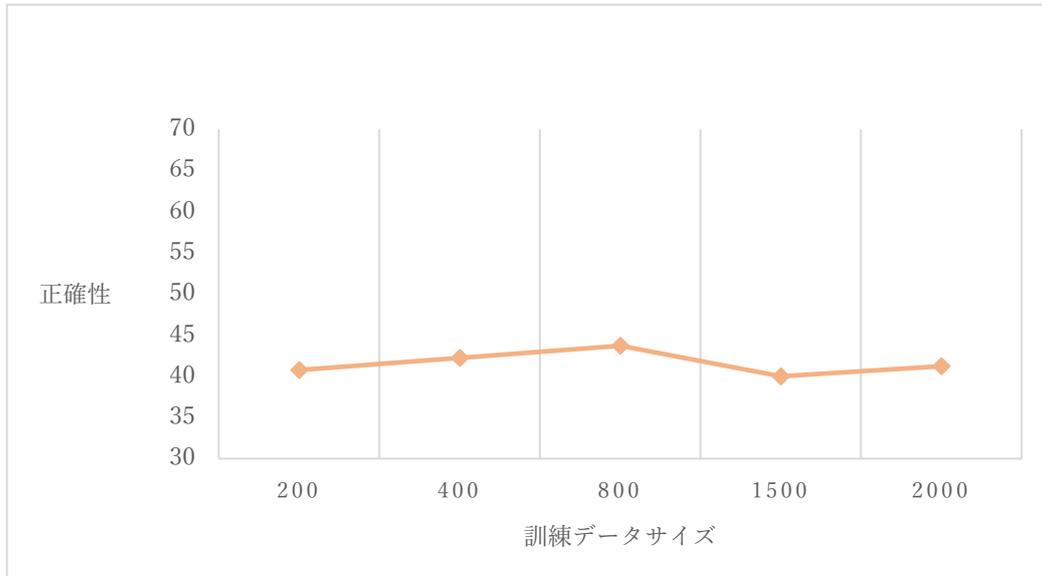


図 6 デュースサイドのサービスの予測精度の推移

4.2.2 アドバンテージサイドの正確性

デュースサイドと同じ手法で行う。全姿勢データから LSTM を用いて予測モデルの精度を 200, 400, 800, 1500, 2000 件のそれぞれのデータ数で予測精度を抽出した。183 件の姿勢データからランダムに 20 件を抽出し、それぞれのデータ数を 5 回ずつ行った結果、表 3 のようなアベレージが確認できた。アドバンテージサイドの評価は、全関節位置を設定値に基づいて平行移動させて姿勢データ数を拡充するにつれ、精度が高くなっている。これを折れ線グラフとして図 7 で示した。姿勢データをランダムに拡充しているため拡充数が 400 の時点では、精度が高くなっているが、その他では訓練データサイズに応じて精度が向上している。

表3 アドバンテージサイドのサービスの予測精度

訓練データ サイズ	200	400	800	1500	2000
精度	38.5	58.56	44.72	45.77	46.91

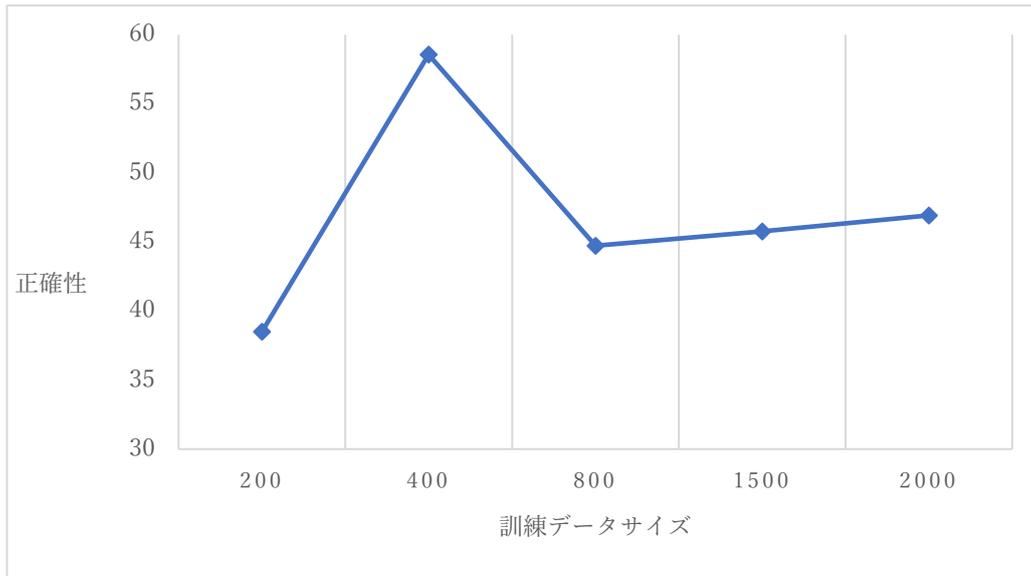


図7 アドバンテージサイドのサービスの予測精度の推移

・4.2.3 イン, フォルト, ネットの正確性

前節では、アドバンテージサイドとデュースサイドの訓練データサイズに基づいて正確性を導いた。位置データのみでの予測モデルは正確性に確信できないと考える。ここではサービスを行った際に、イン、フォルト、ネットの結果からサービス予測モデルの正確性を抽出した。全姿勢データから LSTM を用いて予測モデルの精度を 200, 400, 800, 1500, 2000 件のそれぞれのデータ数で予測精度を抽出した。183 件の姿勢データには、イン、フォルト、ネットの 3 つに分かれる。インは 80 件、フォルトは 38 件、ネットは 65 件あり、姿勢データが少ないため、それぞれのデータ数で拡充する。ランダムに 20 件を抽出し、姿勢データが偏る場合があるため、それぞれのデータ数を 5 回ずつ行った結果、表 4 のようなアベレージが確認できた。これを折れ線グラフとして図 8 で示した。訓練データサイズが 200 を除いて、精度は 50~70 を維持しており精度が高い。インのフォームから関節のずれが少ないため、学習の際にイ

ンのフォームを判別しやすいといえる。

表 4 インのサービス予測精度

訓練データ サイズ	200	400	800	1500	2000
精度	46	64	64	56	62

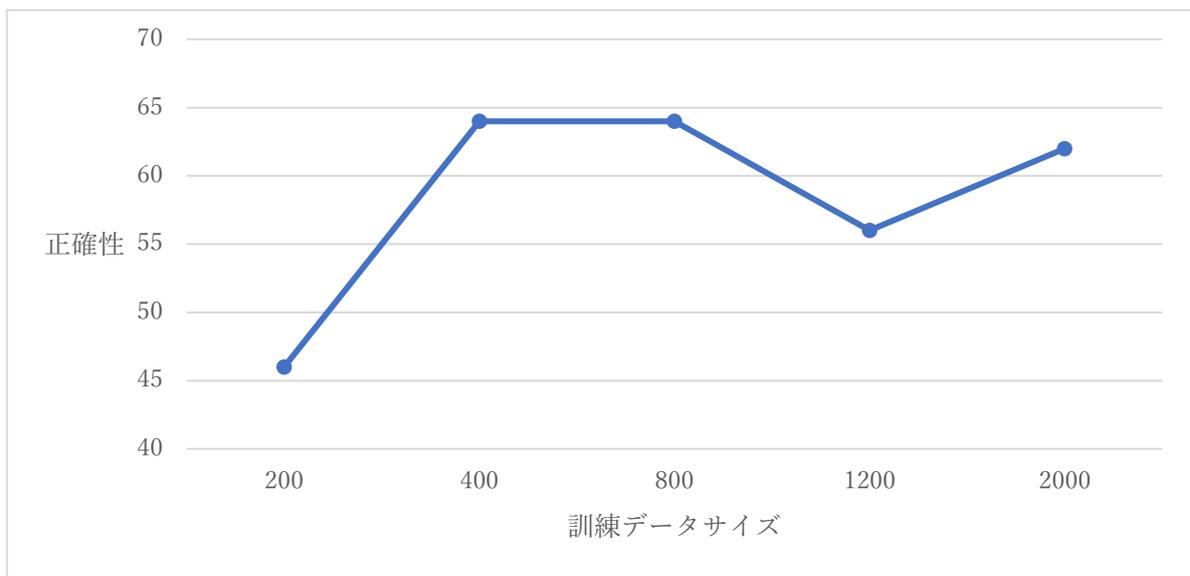


図 8 インのサービス予測精度

フォルトとネットもインと同じ手法で行う。全姿勢データから LSTM を用いて予測モデルの精度を 200, 400, 800, 1500, 2000 件のそれぞれのデータ数で予測精度を抽出した。フォルトは 38 件、ネットは 65 件あり、姿勢データが少ないため、それぞれのデータ数で拡充する。ランダムに 20 件を抽出し、姿勢データが偏る場合があるため、それぞれのデータ数を 5 回ずつ行った結果、表 5 のようなアベレージが確認できた。これを折れ線グラフとして図 9 で示した。

表4 フォルトとネットのサービス予測精度

訓練データ サイズ	200	400	800	1500	2000
フォルト (精度)	24	38	48	30	34
ネット (精度)	36	38	38	20	28

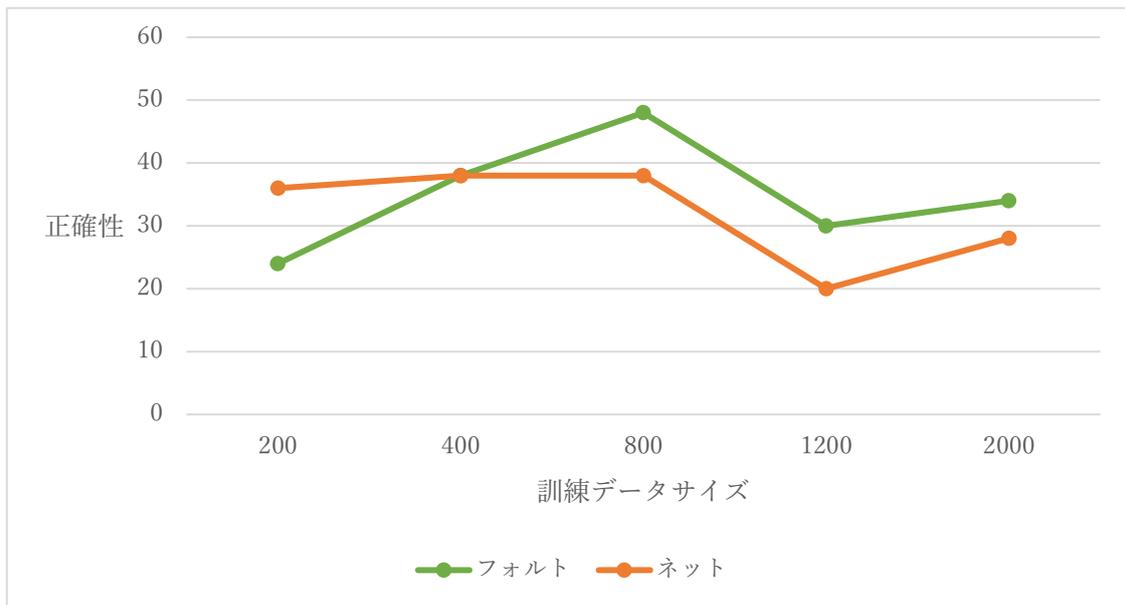


図9 フォルトとネットのサービス予測精度

第5章 考察

全姿勢データから LSTM を用いて予測モデルの正確性を 200, 400, 800, 1500, 2000 件のそれぞれの訓練データサイズでデュースサイドとアドバンテージサイドの精度を算出し、比較したものが図 10 と表 6 である。デュースサイドとアドバンテージサイドの正確性を比較することでサービスを打つ姿勢に活用できると考える。デュースサイドは訓練データをいくら増やしても、サービスの予測精度が高くなることから、サービスのフォームの違いがサービスの結果に影響を与えないと考えられる。一方、アドバンテージサイドは訓練データサイズが増えるにつれ、正確性も向上している。この違いを得て、デュースサイドはサービスと打つ際、身体を捻り、手首を返すように打つため関節の座標位置のずれが大きいため、イン、フォルト、ネットの判別が困難だと考える。アドバンテージサイドは、サービスボックスの正面に身体を置き、身体をあまり捻らず、手首もあまり返さない。よって、関節の座標位置のずれが少ないため、イン、フォルト、ネットの判別ができやすく拡充したデータ数に応じて精度も向上するといえる。

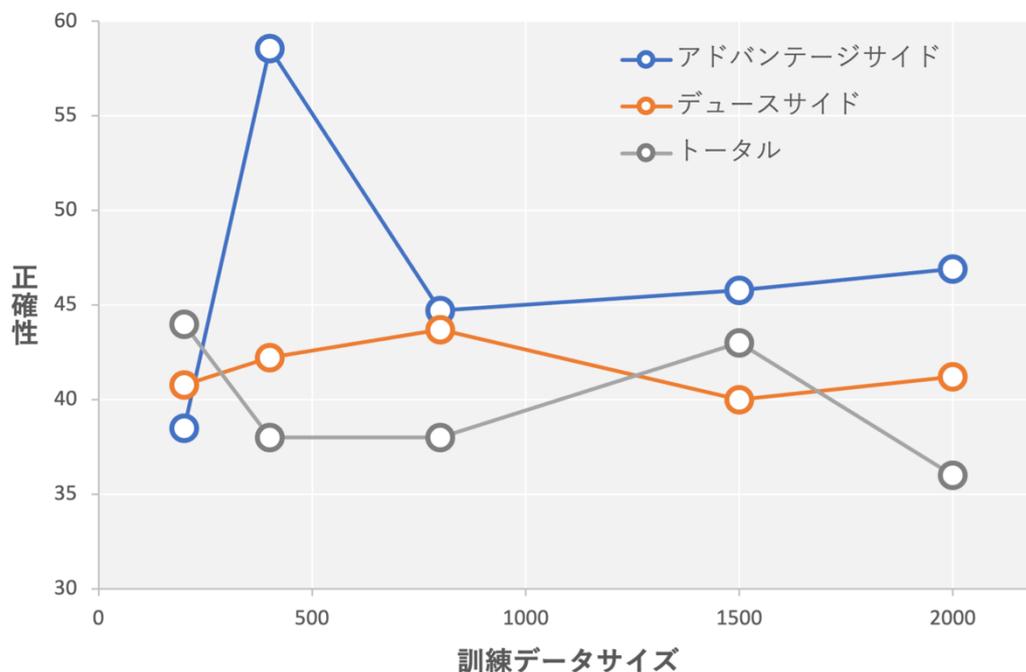


図 10 精度の比較

表 6 精度の比較

拡充数	アドバンテージサイド	デュースサイド	トータル
200	38.5	40.78	44
400	58.56	42.23	38
800	44.72	43.71	38
1500	45.77	40	43
2000	46.91	41.22	36

イン、フォルト、ネットの結果からサービス予測モデルの正確性を抽出した。全姿勢データから LSTM を用いて予測モデルの精度を 200, 400, 800, 1500, 2000 件のそれぞれのデータ数で予測精度を抽出した。183 件の姿勢データには、イン、フォルト、ネットの 3 つに分かれ、インは 80 件、フォルトは 38 件、ネットは 65 件ある。姿勢データが少ないため、それぞれのデータ数で拡充した。また、イン、フォルト、ネットの精度をテストするために、それぞれランダムに 10 件、抽出し、5 回ずつ学習した。この精度が表 7 であり、それを折れ線グラフに表したものが図 11 である。訓練データサイズに応じて正確性が向上することはなかったが、それぞれの訓練データサイズにばらつきが少なく、一定の正確性が得られることが分かった。インは 40%~70%と高く、フォルトは 20%~50%、ネットは 20%~40%と低い結果となった。正確性からフォルトとネットと比較し、インは関節に特徴があるといえる。実際、サービスを打つフォームが少しでも崩れてしまうことで、インにならない。学習した際に、それぞれの関節にずれが少なく、自分に適しているフォームで打っていると考えられる。一方、フォルトとネットは、正確性が低かったため、関節に特徴が無いと言える。今回の動画を参考にし、どの関節に問題があるのかをインとフォルトとネットを使って比較した。フォルトになる場合は、身体よりもボールが後ろになってしまうことで手首の関節が前に突き出るような形が多い。そのため、リリース時も身体全体、体重が前に乗らず反ったフォームになっている。ネットの場合は、インのフォームの身体よりもボールが前になってしまう、全体的に前に倒れるような形になることが多い。ここから共通する事とし

て、互いにボールを上げるトスを改善することで良いフォームが見つかると思う。インした場合は、フォルトとネットの間にトスが上がっているため、良いフォームで打っている事が分かった。アドバンテージサイドとデュースサイドの正確性とイン、フォルト、ネットの正確性を組み合わせてフォームを確立させる。アドバンテージサイドでは、サービスボックスの正面に身体を置き、身体をあまり捻らず、手首もあまり返さないようにし、身体の少し前でトスを上げて打つことが良いフォームである。デュースサイドでは、身体の少し前でトスを上げ、身体を捻り、手首を返すように打つことで良いフォームで打つことができ、サービスの確率が向上すると考える。

表7

	イン	フォルト	ネット
200	46	24	36
400	64	38	38
800	64	48	38
1200	56	30	20
2000	62	34	28

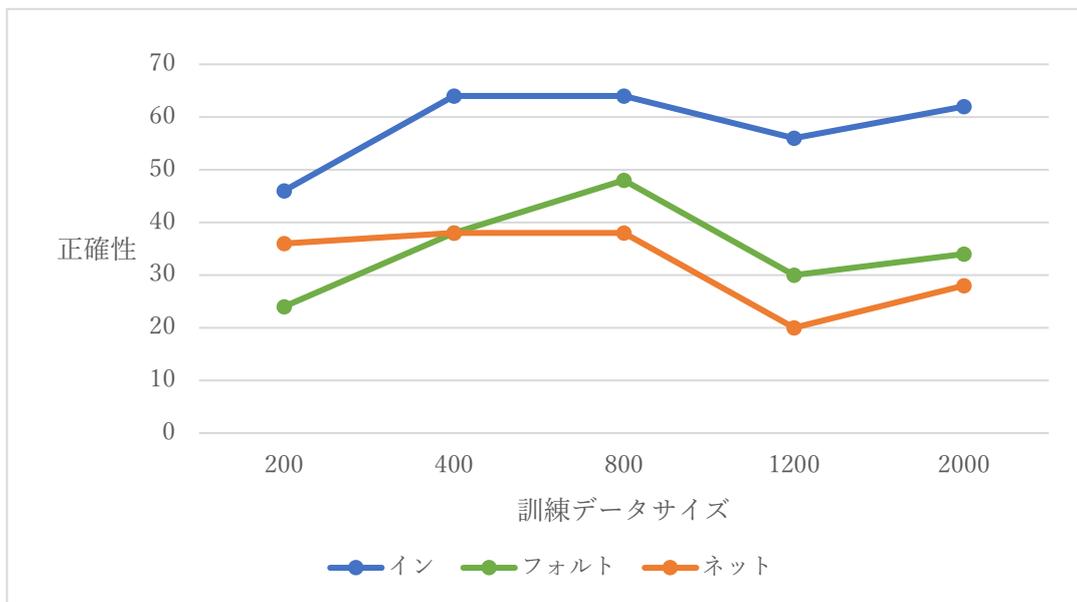


図 11 精度の比較

第6章 おわりに

本研究では、サービスの動画から姿勢推定法である OpenPose を用いてサービスデータから関節座標データを抽出し、LSTM を用いてサービスの予測モデルを構築した。2カ所の位置から、各4コースにサーブを打ち分けた姿勢データの全183件を収集した。また姿勢データから各コースに打ち分けるサービスがイン、フォルト、ネットの result データも収集した。しかし、183件の姿勢データは、動画を撮る際に手元を固定していないことや、そもそも、位置がずれていることなどがあるため正確な学習が困難である。そこで、姿勢データからランダムにデータを選び、ランダムに関節位置 (X,Y) の移動方向と移動距離を設定し、全関節位置を設定値に基づいて平行移動させて姿勢データを拡充した。データ拡充前の訓練データのみの場合、正確性が29%であったが、データ数を増やすにつれて正確性が向上し、データ数2000件で正確性が42%となり、より正確なサービス予測モデルを構築できた。このサービス予測モデルを用いることで時間に限られることなく、コートがなくても練習することが可能である。アドバンテージサイドの正確性とデュースサイドの正確性を比較し、イン、フォルト、ネットに関わる関節が分かった。しかし、イン、フォルト、ネットに関わる関節をどう修正することでフォームが改善できるのかを考慮していない。ソフトテニス屋外と屋内の両方で行われるため、その時の天候もデータとして研究することで、より優れたサービス予測モデルが構築できる。

謝辞

本研究を行うにあたり，熱心なご指導，ご助言を賜りました村上陽平准教授に深謝申し上げます。また，ご協力いただいた被験者の MondheeraPituxcoosuvarn 助教，普段からお世話になっている社会知能研究室の皆様にも心より感謝申し上げます。

参考文献

- [1] Cao, Z., Simon, T., Wei, S. E., and Sheikh, Y.: Realtime Multi-person 2d Pose Estimation Using Part Affinity Fields, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291-7299 (2017).
- [2] 若林明穂: OpenPoseを用いたビーチフラッグスの動作分析, 順天堂大学卒業論文, (2019).
- [3] 金子和樹, 中村拓馬, 矢入郁子, 平田均: OpenPoseを用いたサッカー熟練度の分類, 人工知能学会全国大会論文集, 2020年JSAI2020 巻 3M5GS1205 (2020).
- [4] 木村延明, 中田達, 桐博英, 関島建志, 安瀬地一作, 吉永育生, 馬場大地: LSTMモデルを用いた低平地排水機現場の水位予測, 土木学会論文州B1 75 巻 2 号 p. 139-144 (2019).


```

    Masks.append(mask)
    datas.append(data)
datas
a=np.array(datas)
print(a.shape)
print(len(Masks))
print(a[170][0][50])

print("Result:{}".format(len(result)))

```

A.2 テストデータの分類

```

NUM_TEST = 20

test_idx1 = []
test_idx2 = []
list_0 = [i for i, x in enumerate(result) if x == 0]
print('# of result=0: {}'.format(len(list_0)))
list_1 = [i for i, x in enumerate(result) if x == 1]

print('# of result=1: {}'.format(len(list_1)))
list_2 = [i for i, x in enumerate(result) if x == 2]
print('# of result=2: {}'.format(len(list_2)))

pre_test_idx = sorted(random.sample(range(len(dirs)), k=NUM_TEST))
for x in pre_test_idx:
    if x < 89:
        test_idx1.append(x)

    else:
        test_idx2.append(x)
print(test_idx1)
print(test_idx2)
print("Result:{}".format(len(result)))

```

A.3 テストデータの前処理

```

test_idx = pre_test_idx
print("Result:{}".format(len(result)))
test_result_master = []
test_data_master = []
for idx in test_idx:
    print("Idx:{},Result:{}".format(idx,result[idx]))
    test_result_master.append(result[idx])
    test_data_master.append(a[idx][0])

print('=====')

test_mask = []
for idx in reversed(test_idx):
    test_mask.append(Masks.pop(idx))

train_mask = Masks

```

```

test_mask = np.array(list(reversed(test_mask)))

test_result = []
for idx in reversed(test_idx):
    test_result.append(result.pop(idx))

train_result = result
test_result = np.array(list(reversed(test_result)))
#new_test_goal = np.expand_dims(new_test_goal, axis=1)

train_data = np.delete(a, pre_test_idx, axis=0)
test_data = a[test_idx]

print('len_train_mask:{}'.format(len(train_mask)))
print('len_test_mask:{}'.format(len(test_mask)))
print('len_train_result:{}'.format(len(train_result)))
print('len_test_result:{}'.format(len(test_result)))
print('len_train_data:{}'.format(train_data.shape[0]))
print('len_test_data:{}'.format(len(test_data)))
print(test_result)

```

A.4 訓練データの拡充

```

def expand_data(num):
    c = np.zeros((num,100,51))
    ext_result = []
    ext_maskdata = []
    for n in range(num):
        #i = random.randint(0,train_data.shape[0]-1)
        i = n%train_data.shape[0]
        #print(i)
        #print(i)
        #i=random.randint(11,a.shape[0]-1)
        ext_result.append(train_result[i])
        ext_maskdata.append(train_mask[i])
        x = random.randint(1,100)
        y = random.randint(1,100)
        for j in range(train_data.shape[1]):
            for k in range(train_data.shape[2] - 1):
                if k % 2 == 0:
                    c[n, j, k] = x + train_data[i, j, k]
                else:
                    c[n, j, k] = y + train_data[i, j, k]
            c[n, j, 50] = train_data[i,j,50]
    return c, np.array(ext_result), np.array(ext_maskdata)

i=4000
expanded_data, expanded_result, expanded_mask = expand_data(i)

train_json = (expanded_data, expanded_mask)
test_json = (test_data, test_mask)
train_goal = expanded_result
test_goal = test_result
print(expanded_data.shape)
print(expanded_result.shape)
print(expanded_mask.shape)

```

A.5 サービスの予測モデルの学習と評価

```
from tensorflow import keras
import matplotlib.pyplot as plt
import tensorflow as tf

EPOCHS = 40
MAX_SEQ_LENGTH = 100
NUM_FEATURES = 51

def get_sequence_model():
    #class_vocab = label_processor.get_vocabulary()

    frame_features_input = keras.Input((MAX_SEQ_LENGTH, NUM_FEATURES))
    mask_input = keras.Input((MAX_SEQ_LENGTH,), dtype="bool")

    x = keras.layers.GRU(16, return_sequences=True)(
        frame_features_input, mask=mask_input
    )
    x = keras.layers.GRU(8)(x)
    x = keras.layers.Dropout(0.4)(x)
    x = keras.layers.Dense(8, activation="relu")(x)
    output = keras.layers.Dense(3, activation="softmax")(x)

    rnn_model = keras.Model([frame_features_input, mask_input], output)

    rnn_model.compile(
        loss="sparse_categorical_crossentropy", optimizer="adam",
metrics=["accuracy"]
    )
    return rnn_model

# Utility for running experiments.
def run_experiment():
    filepath = "/home/ueno2/notebook/classifier"
    checkpoint = keras.callbacks.ModelCheckpoint(
        filepath, save_weights_only=True, save_best_only=True, verbose=1
    )

    seq_model = get_sequence_model()
    history = seq_model.fit(
        [train_json[0], train_json[1]],
        train_goal,
        batch_size=128,
        validation_split=0.3,
        epochs=EPOCHS,
        callbacks=[checkpoint],
    )

    seq_model.load_weights(filepath)
    _, accuracy = seq_model.evaluate([test_json[0], test_json[1]],
test_goal)
    print(f"Test accuracy: {round(accuracy * 100, 2)}%")
    predictions = seq_model.predict([test_json[0], test_json[1]])
```

```

    print(predictions)
    print(test_goal)
    print(test_idx)

    return history, seq_model

history, sequence_model = run_experiment()
# Plot training & validation accuracy values
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

```