AIするディープラーニング

AI Love Deep learning

1016163 濱口　和希 Kazuki Hamaguchi

1. 背景

世界が情報化社会に移り変わる中，大量のデータを処理するための手法として機械学習は日々発展を続けてきた．その中で，企業が持つビッグデータの存在とその価値が周知されるようになると，機械学習は一気に注目を集め，様々な企業で導入されるようになった．ビッグデータを扱う上で，特に適しているとされたのがディープラーニングである．ディープラーニングは機械学習技術の一つで，ニューラルネットワークという人間の神経構造を模した機械学習モデルのうち，隠れ層という層をいくつか重ねたものを指す．層を重ねることで，データを分析するために必要な要素とその関係を，より多くかつ複雑に蓄積することができるようになっている．ディープラーニングの精度は用意する学習データ量に左右されるため，限られたデータ量でより精度を上げるための研究開発が進められている．本プロジェクトでは，このディープラーニングを用いて新たな問題解決に挑むことを全体の目標とする．

2. 基礎知識の習得とテーマ

4 月から 5 月は，主に基礎的な知識習得と先行研究調査に取り組んだ．

基礎的な知識として，Python によるプログラミングとディープラーニングの初歩を参考書[1]を使用し学んだ．Python は機械学習に関するライブラリが充実しているため，本プロジェクトで主に使用する言語として採用した．プログラミングの学習においては参考書[1]のサンプルコードを参考にし，動かすことによって行った．先行研究調査は論文や書籍，Web などの情報を参考にし，各自で情報をまとめた．その後，全体で一人ずつテーマを決めるため，プレゼンテーションをし，ブレインストーミングを行った．その結果，「声質変換」と「姿勢推定」の２つにテーマが絞られたため，各自の希望によりどちらかに配属され，課題解決を行うこととなった．

3. 課題解決のプロセスとその結果

3.1 グループ A：声質変換

3.1.1 先行事例

一般に利用されている声質変換の１つに，入力音声を一度テキストに変換させ，変換させたテキストから合成音声を出力する手法がある．ここで，入力音声を一度テキストに変換する段階を音声認識とする．また，テキストから合成音声を出力する段階をテキスト音声合成(Text-to-speech: TTS) とする．そして，この音声認識とテキスト音声合成を組み合わせた手法をテキストベース変換手法とする．

このテキストベース変換手法にはいくつかの問題がある．まず，テキストベース変換手法では，音声認識とテキスト音声合成の 2 つの段階を経る必要がある．そのため，音声の入力から出力までが遅くなるという問題がある．実際に，テキストベース変換手法を用いた「ゆかりねっと」というツールでは，音声の入力から出力までに約 5 秒から 8 秒の時間を必要とする．また，音声認識では，音の途切れで，文章と次文章の判断をしている．したがって，一定時間以上連続して話していると，音声認識精度が落ちてしまう．その結果，誤認識が発生しやすくなる．さらに，変換させたテキストから合成音声を出力する際，テキスト音声合成を用いる．合成音声が必要となるため特定の人物の声に変換する際は，その人の合成音声を作成する必要がある．そのため，変換先として，既に合成音声として作成されている，SofTalk や VOICEROID に限られる．これらが，我々が改善すべき問題である．

3.1.2 目標設定

グループ A の目的は，テキストベース変換手法よりも高性能な声質変換手法を開発することである．その後，テキストベース変換手法と開発した変換手法をグループ内で検証して，性能評価を行う．

3.1.3 音声を直接変換するメリット

我々は，入力音声を直接変換し，合成音声を出力する手法で声質変換に取り組む．入力音声を直接変換し，合成音声を出力する手法を直接変換手法とする．テキストベース変換手法ではなく，直接変換手法に取り組むことにより，現状における問題点の改善が図れると考えられる．

まず，直接変換手法では，入力から出力にかかる時間が短縮される．テキストベース変換手法を用いた「ゆかりねっと」というツールでは，音声の入力から出力までに約 5 秒から 8 秒の時間を必要とする．しかし，直接変換手法の先行事例であるディープラーニングを用いた声質変換システム[2][3]では，入力から出力までの遅延が約 3 秒から 4 秒である．また，テキストベース変換手法では，音声認識の段階で誤認識が発生していた．しかし，直接変換手法では音声をテキストに変換しないため，誤認識は発生しない．さらに，テキストベース変換手法では，音声の変換先が SofTalk や VOICEROID に限られていた．そのため，SofTalk や VOICEROID に収録されていない特定の個人の声や，アニメキャラクターなどの声に変換することが難しい．

しかし，直接変換手法内の変換方法に，自分の音声をまるで特定の人物が話したかのように変換し，かつ変換先の対象を自由に変更する方法[3]がある．したがって，この手法を用いることで自由な対象への声質変換ができると考えられる．

現状，直接変換手法には，様々な種類の変換方法がある．例えば，隠れマルコフモデル（Hidden Markov Model：HMM）を用いる方法，混合ガウスモデル（Gaussian Mixture Model： GMM）を用いる方法，ディープニューラルネットワーク（Deep Neural Network）を用いる方法などがある．その中でも，DNN は HMM に比べ，より自然な合成音声を出力できる[4]．また，DNN は GMM と比べ，より高精度な変換が可能である[5]．以上より我々は，DNN を用いた手法での開発を前提とする．

3.1.4 開発した手法と結果

開発した手法では，先行事例[2][3]を参考にして低品質な声質変換モデル，高品質化モデルの実装を行った．モデルの全容は図 1 に示す．

低品質な声質変換モデルの学習では，入力話者と目標話者のパラレルデータからそれぞれの基本周波数・メルケプストラムを抽出し，入力話者が目標話者に近づくように学習を行った．そして，入力話者の基本周波数・メルケプストラムが，目標話者の基本周波数・メルケプストラムに近づいているのかを，変換された基本周波数・メルケプストラムと，目標話者の基本周波数・メルケプストラムで比較し，誤差を算出した．この誤差が小さくなれば小さくなるほど，変換の精度が向上する．高品質化モデルの学習では，目標話者の音声データから，意図的に低品質にしたスペクトル包絡と高品質なスペクトル包絡を作成し，低品質なスペクトル包絡を高品質なスペクトル包絡に変換できるように学習させた．

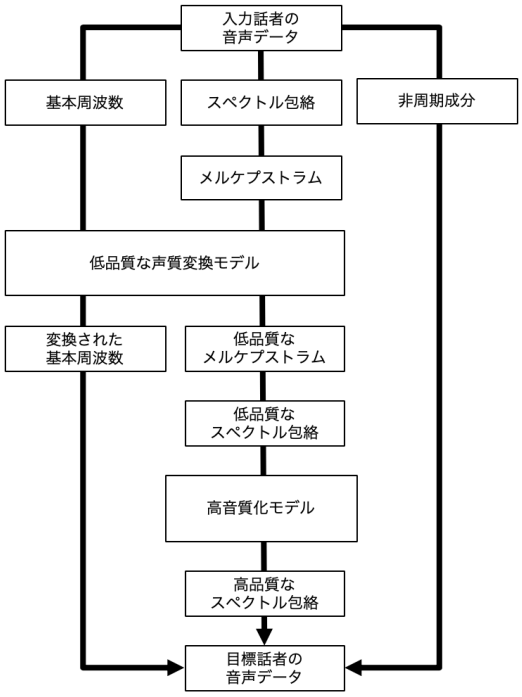

図 1 開発したモデル

最終的には，実際に構築したモデルを利用し，声質変換を行い，変換した音声を流すことまでを行うことができた．

3.2 グループ B：姿勢推定

3.2.1 先行研究

モーションキャプチャは，スポーツのフォームや日常生活の仕草のデータ収集などの研究目的にも多く利用されるなど，バーチャルリアリティに関わらず様々な分野を支えている．モーションキャプチャを機械学習によって行う試みが盛んに行われている．例として，Zhe Cao らの OpenPose[6]は映像に移った人間の各関節の二次元位置を推定することができる．また，Weipeng Xu らの Mo2Cap2[7]は，つば付き帽の先に下向きのカメラを付けることで全身を映し，三次元姿勢を推定する．

3.2.2 従来の手法における問題

現在のモーションキャプチャの手法では，精度を向上させようとするほどセンサを増やす必要があり，費用や管理システムのコストが問題となる．正面からのセンサを用いる場合，少ないセンサで全身の動きをとることができるが，移動範囲がセンサの範囲内に制限されてしまう．また，正面からのセンサが全身を認識することができ，かつ対象が十分に動き回れるような比較的広いスペースも必要になる．以上のモーションキャプチャの問題は，一般にバーチャルリアリティが普及する上での課題となっている．

3.2.3 検討した機械学習手法

グループ B では，前方 360 度撮影できる全方位カメラを用いて姿勢推定することによって大掛かりな機材や撮影する上で妨げとなる第三者からの計測を必要としないモーションキャプチャができると考え，我々はディープラーニングと全方位カメラを用いた姿勢推定を提案する．

3.2.4 提案するアイデア

我々は，モーションキャプチャや姿勢推定の調査，画像処理の手法である畳み込みニューラルネットワークの構築及びデータの収集のために参考文献[1]と先行事例[7][8][9]を用いた．

畳み込みニューラルネットワークは画像認識の分野で精度が高く，深い層での学習が比較的安定する DenseNet と呼ばれるモデルを用いる．DenseNet の構造を図 2 に示す．

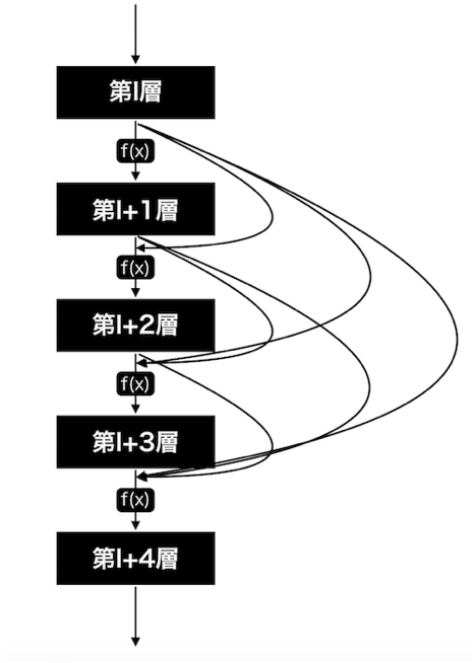

図 2 DenseNet の第 l 層から l+4 層の構造

データ収集には，全方位カメラと KinectV2 を用いる．全方位カメラで動画を撮影し，KinectV2 で動画における各フレームに対する関節の角度の収集を行う．

これらのデータを用いて畳み込みニューラルネットワークの訓練と調整を行う．その後，今回は歩行のみの姿勢推定に限定し，Unity の 3D モデルを用いて，全方位カメラの映像から畳み込みニューラルネットワークを用いた推定を行い 3D モデルに反映させる．

3.2.5 実験

実験環境は．まずコンピュータのスペックはメインメモリ 64GB，プロセッサは Intel Core i7-7700k，グラフィックボードは GeForce GTX1080Ti である．OS は Ubuntu17.10 を使用した．また，TensorFlow 上で実行される高水準なニューラルネットワーク用のライブラリである Keras と，TensorFlow や Keras で実装したモデルの様々なデータを可視化するライブラリである tensorboard を追加した．

モデルの入力は映像とし，出力は quaternion の形式の関節角度とした．入力の一次元目は 15 フレーム分，二次元目および三次元目はそれぞれ映像の縦と横のピクセル数である．出力の一次元目は 15 フレーム分，二次元目は 11 の関節(腰，肩，ひじ，手首，足の付け根，膝)の角度を quaternion で表現する．

全てにおいて epoch 数は 300 とし，ResNet およびいくつかの DenseNet を学習させた．ResNet の学習時間は約 14 時間，DenseNet の学習時間は約 41 時間となった．何度か仮実験を行い，正則化しなければ過学習を起こしてしまうことがわかった為，L1 正則化および Dropout をモデルに追加して本番の実験を行った．

我々は二種類の検証を行った．一つ目は tensorboard を利用して誤差を定量的に測る方法，二つ目は学習済みのモデルが出力した関節のデータを Unity で読み込み，Unity 上のモデルをその関節データに従って動かしてどれだけ自然かを主観的に判断する方法である．tensorboard ではトレーニングデータと検証用データそれぞれの平均絶対値誤差と Loss を 1epoch 毎にプロットして監視を行った．

3.2.6 結果と考察

ResNet を使っての学習の結果，Loss がほぼ一定の値に収束した．しかし各クォータニオンの誤差の値が大きいことから，上手く学習ができなかった．また Loss が収束し，現

状これ以上の学習を行うことができないため，データまたはモデルに対しての改善が必要であることが分かった．

次に，層を深くすることのできる DenseNet を用いて学習した結果，ResNet と同様に Loss が収束した．だが，ResNet と比較して少しは改善されたが，良い精度ではなかった．出力を分析した結果，手首の精度が著しく低いことがわかった．原因として考えられることは，Kinect から取得されたデータの精度が低いということである．期待する出力として与えているデータの精度が低いため，ネットワークから出力されるデータの精度も低くなる．また，歩行というタスクにおいて，手首の動きは重要なパラメータでないと判断したため，予測するパラメータから手首を除いて学習することとした．

手首を除いて学習した場合においても，同様に Loss は収束した．手首を除いて学習した DenseNet は，手首を含んで学習したモデルと比較して，少し精度が改善された．手首を除いて学習することで多少の改善は見られたが，全体的な精度としては依然低いままであり，著しく精度の低いパラメータを除いても，精度が改善されていないため，モデルの表現力を増す必要があると考えた．そこで，各関節あたりに使うパラメータを増やすために，関節を上半身と下半身に分割することとした．

関節を分割して学習した結果においても Loss は収束した．関節を分割して学習した DenseNet は手首を除いて学習したモデルと比較して，精度が悪化した．

このような様々な方法を用いて実験を行ったものの，いずれも良い精度を出すことができず，今回の目標に達する結果を出すことはできなかった．

4. 今後の課題

今後の課題として，グループ A では声質変換のバリエーションを増やして行く必要があると考える．例えば，ささやき声や叫び声，歌うなどの声を収録し変換してみる必要がある．また，リアルタイム性を持たせることができなかった．そのため，リアルタイム性を持たせた上での変換速度も計測する必要がある．

グループ B では，今回の目標を達成することができなかったが，今後の課題として目標に達成する十分な結果を得るには，入力部分の学習の妨げとなる情報を取り除き，更なるモデルの表現力の向上や，タスクに必要な情報を取得し簡略化を行うことによりプロジェクトの目標を達成できると考えられる．

参考文献

[1] 斎藤 康毅，ゼロから作る Deep Learning -Python で学ぶディープラーニングの理論と実装，オライリー・ジャパン，2016.

[2] Hiho「ディープラーニングの力で結月ゆかりの声になってみた」，2018.
[online]https://blog.hiroshiba.jp/became-yuduki-yukari-with-deep-learning-power/
2018 年 7 月 20 日アクセス

[3] Yoshikazu Oota「「ディープラーニングの力で結月ゆかりの声になる」ための基礎知識とコマンド操作」，2018
[online]https://qiita.com/atticatticattic/items/575d71dab4ee716e4969
2018 年 7 月 20 日アクセス

[4] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng, “Phonetic posteriorgrams for many-to-one voice conversion without parallel data training,” Multimedia and Expo (ICME), 2016 IEEE International Conference on, pp. 1-6. 2016.

[5] 廣芝 和之，能勢 隆，宮本 颯，伊藤 彰則，小田桐 優理 「畳込みニューラルネットワークを用いた音響特徴量変換とスペクトログラム高精細化による声質変換」，研究報告音声言語情報処理，Vol. 2018-SLP-122, No. 27, pp.1-4. June 2018.

[6] Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. arXiv:1611.08050. 2016.

[7] Xu,Weipeng,Chatterjee,Avishek,Zollhoefer,Michael,Rhodin,Helge,Fua,Pascal,Seidel,Hans-Peter,Theobalt, and Christian. Mo2Cap2: Real-time Mobile 3D Motion Capture with a Cap-mounted Fisheye Camera. arXiv:1803.05959. 2018.

[8] Kaiming He,Xiangyu Zhang,Shaoqing Ren,Jian Sun Deep Residual Learning for Image Recognition. arXiv:1512.03385. 2015.

[9] Gao Huang,Zhuang Liu,Laurens van der Maaten,Kilian Q. Weinberger Densely Connected Convolutional Networks. arXiv:1608.06993. 2016.