スマートフォンのモーションセンサを利用した個人認証アプリケーションの開発に関する研究

総合情報学研究科

知識情報学専攻

マルチメディア情報システムの基礎と実際

15M7112

髙坂 賢佑

要旨

スマートフォンの普及により，従来型のフィーチャーフォンと呼ばれる端末と比較して多種多様なサービスを利用できるようになった．例えば，スマートフォンはパーソナルコンピュータ向けに設計されたWebサイトを閲覧できるフルブラウザが利用できる．これを用いることで，オンラインショッピングを含むサービスをどこでも手軽に利用できるようになった．またスマートフォンは，様々な企業や個人が開発した多種多様なアプリケーションを自由にインストールし利用できる．例えば国内銀行各社からスマートフォン向けアプリが提供されているが，これを用いることでアプリ内で口座残高の確認や入出金明細の確認はもちろん，振り込みや振り替えなども行える．このようなサービスを利用する際には，あらかじめ登録したユーザ名とパスワードを用いた個人認証を行うのが一般的である．だがこれら情報をアプリ内に記憶しておくことで，利用毎に再入力する手間を省けるような仕組みを持つ場合もある．

このようにスマートフォンによって日々の生活がより豊かになった一方で，端末がより多くの個人情報を内包するようになった．このことから，第三者によって不正に個人情報にアクセスされたり，インストールされたアプリを通じて様々なサービスをなりすまし利用された場合の危険性は高くなった．そのため，端末利用時にはパスコード認証や指紋認証などを用いて本来の端末所有者であるか認証するよう設定することが推奨されている．

現在スマートフォンにおいて個人の認証方式として広く使われている方法として，パスコード認証と指紋認証が挙げられる．しかしながらこれら認証方式にはいくつかの問題点が考えられる．まずパスコード認証では認証作業が煩雑であったり認証に用いる鍵情報の自由度が低いという点がある．また指紋認証では指紋を読み取るためのハードウェアが必要である点や，指紋情報は変更ができないため，何らかの原因でこの情報が第三者に漏洩した場合はその指紋を用いた個人認証ができなくなるという点がある．

そこで本研究ではスマートフォンに一般的に搭載されている加速度センサと角速度センサを利用し，人間の動きを用いて端末を振ることで個人を認証するシステムを開発した．本シス

テムでは，認証時に入力されたデータが本来の端末所有者本人によるものかを識別するために，人工ニューラルネットワークを利用した．端末所有者はあらかじめ認証時に利用したい動きをシステムに入力し，登録処理を行う．登録処理では，まず Denoising Autoencoder を用いて入力されたデータの特徴を学習し，その後 Denoising Autoencoder の後ろに識別を行うニューロンを繋いで学習を行う．認証時には，登録処理で得られた識別器を用いて入力されたデータが端末所有者本人のものであるかを識別する．

評価の結果，云々．本システムにより，パスコード認証が抱える認証の煩雑さと鍵情報の自由度といった問題点を軽減した．また指紋認証における鍵情報が変更できないという問題点を解消し，直感的に個人認証を行うことが可能になった．

目 次

図 目 次

表 目 次

第1章　序論

1.1　研究背景

近年，スマートフォンと呼ばれる携帯端末が急速に普及しつつある．スマートフォンとは，パーソナルコンピュータ向けに設計されたWebサイトを閲覧できる機能を持つフルブラウザを搭載し，様々な企業や個人が開発した多種多様なアプリケーションをインストールし利用できる携帯端末のことを指す[1]．平成28年版の情報通信白書によるとスマートフォンの世帯普及率は2015年末時点で72.0%とあり，また前年比で7.8ポイント増となっている[2]．スマートフォンの普及によりどこでも手軽にオンラインショッピングやネットバンキングをはじめとする多種多様なサービスを利用できるようになった．

その一方で，これらサービスの利用にはユーザIDやパスワード等を含む個人情報を用いた個人認証を必要とする場合が多い．また，利用しているブラウザやアプリケーションによっては，サービスにログインすれば一定期間ログイン状態を保持し再ログインの手間を省くような機能を持つものもある．この機能により，ユーザはサービスを利用するたびに再ログインする手間が無くなることから利便性が向上する．しかしながら，悪意のある第三者がサービスへのログインに必要な情報を知らずとも，本来のユーザになりすましてサービスを利用できてしまうという危険性がある．

このように，スマートフォンは従来型のフィーチャーフォンと比較してより多くの個人情報を内包しており，第三者からのこれら情報への不正なアクセスを防ぐための仕組みが不可欠となっている．現在この仕組みを実現する方法として広く採用されているのが，端末利用時にあらかじめ登録したパスコード情報や指紋情報をもとに，現在の利用者が本来の端末所有者であるかを確認する個人認証システムである．パスコード認証方式では，あらかじめ端末所有者が特定の文字種からパスコードを構築し，これを端末に登録しておく．そして，端末利用時に入力されたパスコードと登録されたパスコードを比較して同一であれば端末所有者であるとみなして，その後の端末利用を許可する．指紋認証方式では，あらかじめ端末所有者が端末に搭載

された指紋スキャナを通じて自らの指紋をスキャンし，これを端末に登録しておく．そして，端末利用時に指紋をスキャンして登録された指紋との比較をし，同一であれば端末所有者であるとみなしてその後の端末利用を許可する．これらの個人認証システムを利用することにより，第三者によって不正に端末内の個人情報へアクセスされる危険性をある程度軽減できる．しかし，これらの認証方式にはそれぞれいくつかの問題点が挙げられる．

まずパスコード認証方式だが，これは個人認証を行う際にスマートフォン画面上に表示されたソフトウェアキーボードを目視し指でタッチして操作する必要があり，ユーザにとっては煩雑である可能性があるという点がある．またあらかじめ決められた文字種の中から一つずつ選んだ文字を並べてパスコードを構築することから，パスコードのパターン数が限られ，認証に用いる鍵の自由度が制限されてしまうという点がある．

指紋認証方式については指紋をスキャンするためのハードウェアをスマートフォンに搭載しなければならないという点がある．また指紋情報は変更ができないため，何らかの原因でこの情報が第三者に漏洩した場合は，今後その指紋を用いた個人認証ができなくなるという点がある．さらに，ドイツのハッカー集団である Chaos Computer Club の生体認証チームが，一般的なカメラで撮影された写真に写り込んだ指から指紋を複製することに成功している [3]．このことから，指紋情報が漏洩する可能性が十分にあり個人認証システムが担う機密性の確保が難しいといえる [4].

1.2　研究目的

本研究では，一般的なスマートフォンに搭載されている加速度センサと角速度センサを利用し，端末を振る動き（以下，モーション）で個人認証を行うシステムを開発する．これによりパスコード認証方式における認証作業の煩雑さと鍵情報の自由度が制限されるという課題点を軽減し，指紋認証方式における指紋情報が漏洩した場合に鍵情報の変更ができないという課題点を解消した生体認証システムの実現を目指す．

1.3　本論文の構成

第 2 章にて本研究に関連する先行研究について述べる．第 3 章では本研究で開発した個人認証システムを提案するにあたり必要となる知識について説明する．第 4 章では本研究で開発した個人認証システムの実装について，その詳細を述べる．第 5 章では本研究で開発した個人認証システムの評価実験とその結果を示し，第 6 章で結論と今後の課題を述べたあと，本論文を総括する．

第2章　関連研究

坂本の研究 [5] では，ユーザが入力したモーションの数値化に加速度センサを用いた．あらかじめ保存しておいた複数種類のジェスチャパターンと認証時にユーザが入力したモーションデータをパターンマッチング方式のアルゴリズムを用いて比較することで個人認証を行った．しかし，このプログラムは扱うジェスチャによって認証率が高いものと低いものに二分化する傾向が見られるという問題点があった．

濵野らの研究 [6] では，加速度センサに加えて角速度センサを用いたジェスチャ動作による認証手法を提案した．これにより回転動作の取得によるモーションの自由度向上となりすまし認証に対する強度の向上を可能にした．認証手法として単一動作を組み合わせて認証する単一動作組み合わせ認証と，ユーザが自由に考えたモーションを用いて DP マッチングによって認証する一筆書き認証の二つを提案した．このシステムの実証実験は複数日かけて実施されており，一筆書き認証において日を経ることによる習熟度の向上から，本人拒否率が改善したことが確認された．しかし，初日の認証での本人拒否率が高く，さらなる本人拒否率の改善が課題として挙げられていた．

第3章　予備知識

本章では，本研究で開発した個人認証システムで用いた技術について説明する．

3.1　人工ニューラルネットワーク

人工ニューラルネットワーク（以下，ニューラルネットワーク）とは，脳内に存在する多数のニューロンによる，シナプスを介した信号のやりとりからなる情報処理の機能を計算機上に再現することを目指したものである．ニューラルネットワークにおけるニューロンは図 3.1 のように表され，図中の x_1 から x_i からなる入力から，式 3.1 によって y_j を出力する．図 3.1 および式 3.1 においてそれぞれ w_{j1} から w_{ji}，w_{ji} と表されているものはシナプスの結合荷重（以下，結合荷重）で，対応する入力にどれだけの重みを持たせるかを示している．また，式 3.1 において b_j と表されているものはバイアスと呼ばれ，ニューロンが発火する傾向の高さを示している．式 3.1 における f は活性化関数と呼ばれ，入力とそれに対応する結合荷重の積を総和したものにバイアスを足した出力を正規化するために用いる．活性化関数には恒等写像や ReLU（ランプ関数）など様々なものがある．このようなニューロンを複数個・複数層に重ねることにより，より複雑な問題に対応できる．

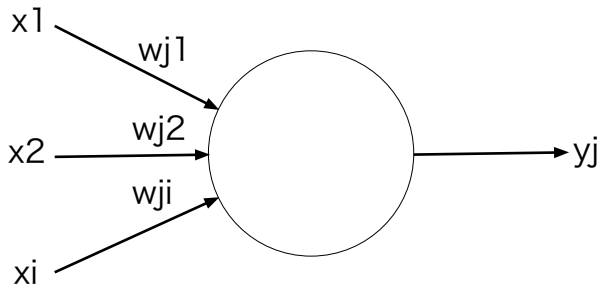

図 3.1: ニューラルネットワークにおけるニューロン

$$y_j = f(\sum_i w_{ji} x_i + b_j). \tag{3.1}$$

ただし，ただニューラルネットワークを構築して入力を与えただけでは，期待した出力が得られることは稀である．そのため，期待した出力が得られるように誤差逆伝播法を用いて結合荷重とバイアスを更新していく，ニューロンの学習を行わなければならない．学習を行うためには，何を目標に結合荷重やバイアスを更新していくのかという基準を設定する必要がある．これは損失関数と呼ばれ，ニューラルネットワークにより得られた出力が，期待する出力（以下，教師信号）とどれだけの誤差があるかを定量的に測るために用いられる．

ニューロンの学習を行う際は，図 3.2 のような縦軸に損失関数（エネルギー関数）E，横軸に結合荷重 w を置いたグラフで，E を最小化するような opt_w に近づくように結合荷重を更新していく（勾配法）．

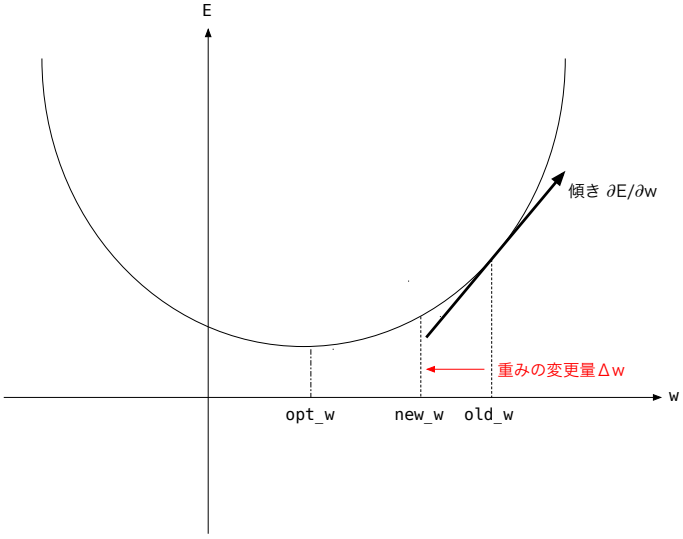

図 3.2: ニューロンの結合荷重の更新

更新した new_w は，式 3.2 で得られる．

$$new\_w = old\_w + \Delta w. \tag{3.2}$$

式 3.2 における Δw が結合荷重の変更量となるのだが，これは式 3.3 で得られる．

$$\Delta w = -\eta \frac{\partial E}{\partial w}. \tag{3.3}$$

η は学習率を表し，どれだけの割合で結合荷重を更新するかを示している．この値を小さくすることで，結合荷重の更新幅が小さくなる．その後ろの $\frac{\partial E}{\partial w}$ は傾きを示す． 結合荷重 w が opt_w に近づくためには，傾きが正の場合は Δw が負に，傾きが負の場合は Δw が正になる必要がある． そのため，$\eta\frac{\partial E}{\partial w}$ の結果得られた値の符号を逆にしている．

結合荷重の変更量 Δw を得るために必要な傾き $\frac{\partial E}{\partial w}$ は，式 3.4 で得られる．

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j}x_i. \tag{3.4}$$

式 3.4 は，合成関数の微分の公式を用いて式変形を行っている． x_i は学習中のニューロンに入力された値を示している． ここで，$\frac{\partial E_n}{\partial a_j}$ を δ_j と置く． この δ_j と学習中のニューロンの入力値 x_i を掛けることで傾きが得られ，これに $-\eta$ を掛けることで結合荷重の変更量 Δw が得られる．

前述したように，損失関数を用いて教師信号との誤差を測るために用いるのは出力層の出力値である．ここで，出力層ニューロンの活性化関数と損失関数が特定の組み合わせであれば，δ_j は式 3.5 で得られる．

$$\delta_j = y_j - t_j. \tag{3.5}$$

式 3.5 における y_j は出力層より得られた出力値，t_j は教師信号である．このように極めて単純な計算式で δ_j を得られるため，基本的に出力層では損失関数と活性化関数を合わせて考えるのが一般的である．

まず，ニューラルネットワークで回帰を行う場合は，活性化関数と損失関数にそれぞれ恒等写像（式 3.6）と最小二乗誤差（式 3.7）を用いる．

$$f(a_j) = a_j. \tag{3.6}$$

$$E_n = \frac{1}{2}\sum_{k=1}^{K}(y_{nk} - t_{nk})^2. \tag{3.7}$$

ニューラルネットワークで二値分類を行う場合は，活性化関数と損失関数にそれぞれシグモイド関数（式 3.8）と交差エントロピー誤差（式 3.9）を用いる．

$$f(a_j) = \frac{1}{1 + e^{-a_j}}. \tag{3.8}$$

$$E_n = -\sum_{k=1}^{K}\{t_{nk}\ln y_{nk} + (1-t_{nk})\ln(1-y_{nk})\}. \tag{3.9}$$

また，ニューラルネットワークで多クラス分類を行う場合は，活性化関数と損失関数にそれぞれソフトマックス関数（式 3.10）と交差エントロピー誤差（式 3.11）を用いる．

$$f(a_j) = \frac{e^{a_j}}{\sum_{k=1}^{K} e^{a_k}}. \tag{3.10}$$

$$E_n = -\sum_{k=1}^{K} t_{nk}\ln y_{nk}. \tag{3.11}$$

このような組み合わせで出力層を構築することで，前述した式 3.5 で δ_j を得られる．

ニューラルネットワークが入力層を含めて 3 層以上あるような複雑なもので中間層を学習したい場合，この δ_j は式 3.12 で得られる．

$$\delta_j = \frac{\partial f}{\partial a_j}\sum_{k=1}^{K} w_{kj}\delta_k. \tag{3.12}$$

つまり，学習しているニューロンの活性化関数を微分したものと，出力層に一つ近い層の各結合荷重と δ を掛けたものの総和を掛けることで，中間層の δ_j が得られる．

以上で学習に必要な傾きが得られるが，これを用いた結合荷重の更新方法として式（3.2）を改良したアルゴリズムが複数考案されている．

まず，確率的勾配降下（SGD）というものがある（式 3.13）．

$$new\_w_{ji} = old\_w_{ji} - \eta\delta_j x_i - \eta\lambda old\_w_{ji}. \tag{3.13}$$

$-\eta\lambda old\_w_{ji}$ は荷重減衰と呼ばれるもので，λ を正の小さな定数に設定することで傾きがゼロの場合でも結合荷重を減らすことができるというものである． η の値が結合荷重の更新に強く影響するため，通常は学習の初期段階では大きめの値にし，学習が進むにつれて小さくしていく．

次に，AdaGrad[7] というものがある（式 3.14）．

$$\begin{aligned} new\_g_{ji} &= old\_g_{ji} + (\delta_j x_i)^2. \\ new\_w_{ji} &= old\_w_{ji} - \frac{\eta}{\sqrt{new\_g_{ji} + \epsilon}}\delta_j x_i. \end{aligned} \tag{3.14}$$

これは，過去の傾きの二乗和を結合荷重ごとに覚えておき（new_g_{ji}），その平方根で η を割ったものを学習率とする． これにより，総更新量が少ない結合荷重はより大きな幅で，総更新量が多い結合荷重はより小さな幅で更新がなされる． この方式では，学習率が自動的に調整される [8].

また，Adam[9] というものがある（式 3.15）.

$$
\begin{aligned}
new\_m &= \beta_1 old\_m + (1-\beta_1)\delta_j x_i. \\
new\_\nu &= \beta_2 old\_\nu + (1-\beta_2)(\delta_j x_i)^2. \\
new\_\hat{m} &= \frac{new\_m}{1-\beta_1^t}. \\
new\_\hat{\nu} &= \frac{new\_\nu}{1-\beta_2^t}. \\
new\_w_{ji} &= old\_w_{ji} - \frac{\alpha}{\sqrt{new\_\hat{\nu}}+\epsilon} new\_\hat{m}.
\end{aligned}
\tag{3.15}
$$

まず，傾きの一次モーメント（平均値）と二次モーメント（分散した平方偏差）の概算値を求める（new_m 及び new_ν）．そしてこれらの偏りをバイアス補正した推定値を計算することで小さくし（$new\_\hat{m}$ 及び $new\_\hat{\nu}$），これらを用いて結合荷重の更新を行う [10].

他にも様々なアルゴリズムが提案されている．

3.1.1　Autoencoder

Autoencoder とは，図 3.3 のような入力層と出力層のニューロン数を入力データの次元数と同数にし，中間層のニューロン数を一定の割合で削減した 3 層構造のニューラルネットワークにおいて，入力データと教師信号に同じデータを用いて教師あり学習させたものである．入力層から中間層への処理をエンコーダ，中間層から出力層への処理をデコーダと呼ぶ．中間層と出力層の活性化関数は自由に選ぶことができるが，出力層の出力値が教師信号を表現できるような活性化関数を選ばなければならない．

Autoencoder の学習を進めた結果出力層の出力と教師信号との誤差が無くなった場合，中間層においてより少ないニューロン数で入力データの情報を表現できていると見ることができる．このことから，学習済みの Autoencoder からデコーダの部分を取り除き，エンコーダの出力を別のニューラルネットワークへの入力値として渡すことで，入力データの特徴抽出器として用いることができる．

3.1.2　Denoising Autoencoder

Denoising Autoencoder とは

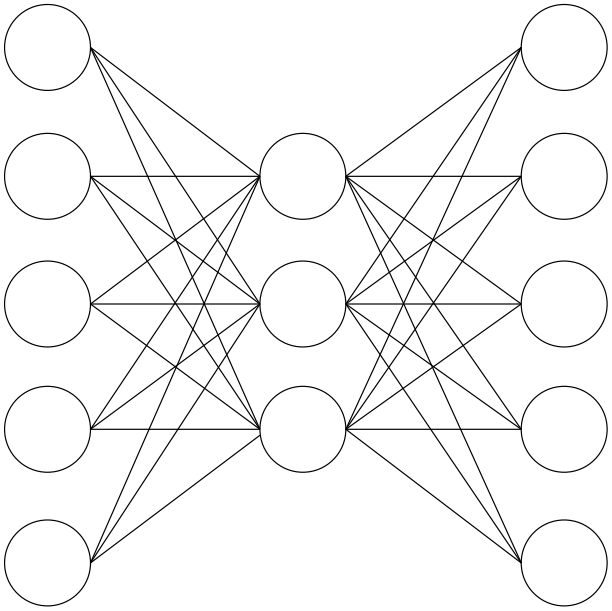

図 3.3: Autoencoder

3.1.3 Dropout

Dropout とは

参考文献，参考URL等

[1] 総務省｜電気通信サービスＦＡＱ（よくある質問）｜スマートフォンとはなんですか？，http://www.soumu.go.jp/main_sosiki/joho_tsusin/d_faq/faq01.html，2017 年 1 月 27 日確認.

[2] 総務省，“平成 28 年版情報通信白書”，2016 年.

[3] CCC｜Fingerprint Biometrics hacked again, https://www.ccc.de/en/updates/2014/ursel, 2017 年 1 月 27 日確認.

[4] Chaos Computer Club claims to have “cracked” the iPhone 5s fingerprint sensor Naked Security，https://nakedsecurity.sophos.com/2013/09/22/chaos-computer-club-claims-to-have-cracked-the-iphone-5s-fingerprint-sensor/, 2017 年 1 月 27 日確認.

[5] 坂本翔，“ユーザの直感的な入力をとらえるための 3 軸加速度センサによるジェスチャ認識の研究”，2009 年度公立はこだて未来大学卒業論文.

[6] 濵野雅史，新井イスマイル，“加速度センサ・ジャイロセンサを併用したスマートフォンの利用認証手法の提案”，情報処理学会研究報告，Vol.2014-MBL-70，No.17，Vol2014-UBI-41，No.17，2014.

[7] John Duchi, Elad Hazan, Yoram Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”, Journal of Machine Learning Research, 12, 2121-2159, 2011.

[8] 実装ノート · tiny-dnn/tiny-dnn Wiki，https://github.com/tiny-dnn/tiny-dnn/wiki/実装ノート，2017 年 1 月 29 日確認.

[9] Diederik P. Kingma, Jimmy Lei Ba, “Adam: A Method for Stochastic Optimization”, The International Conference on Learning Representations, San Diego, 2015.

[10] 勾配降下法の最適化アルゴリズムを概観する｜コンピュータサイエンス｜ POSTD, http://postd.cc/optimizing-gradient-descent/，2017 年 1 月 29 日確認.