人間計算可能なパスワード認証に対する深層学習による安全性評価

村田壱生[†1,a)]

概要：パスワード認証の 1 つに人間計算可能なパスワードが提案されている。これは、人間が秘密を記憶し、その記憶をもとにある質問（チャレンジ）へ応答することでユーザー認証を可能にしたものである。既存研究としては、組み合わせ論的解析が行われているが、これは関数の一部が既知の場合で行われている。本研究では、深層学習（多層パーセプトロン）を用いて、全ての関数が未知の状態でチャレンジからその応答を予測可能であるか否かについて検証を行った。加えて、関数を変更した場合の予測に関しても検証を行い、考察を深めた。

キーワード：パスワード、人間計算可能、深層学習

Security Assessment of Human Computable Password Authentication by Using Deep Learning

ISSEI MURATA[†1,a)]

Abstract: Human-computable passwords have been proposed as a form of password authentication. It allows a user to authenticate himself by remembering a secret and responding to a certain question (challenge) based on the memory. In existing research, combinatorial analysis has been done, but this has been done when some of the functions are known. In this study, we used deep learning (multilayer perceptron) to verify whether or not it is possible to predict the response from a challenge when all functions are unknown. In addition, the prediction when the functions are changed is also examined and discussed.

Keywords: password, human computable, deep learning

1. 研究背景

1.1 認証・ユーザー認証

近年のデジタル社会の普及において、インターネットを介してデジタル情報をやり取りする機会が増えている。こうした対面で人を確認できない環境においては、通信している相手が本人がどうか、通信内容が意図されたものかどうかを確認する手段である「認証」が極めて重要である。その認証の 1 つとして「ユーザー認証」というものがある。ユーザー認証とは、正規のユーザーであることを確認してアクセスを許可することを指す。ユーザー認証は一般的に機械による人の認証であり、その方式としてパスワード認証やバイオメトリクス認証などが上げられる。パスワード認証とは、ユーザーを識別する ID とそれを確認するパスワードを組み合わせることによって行われるユーザー認証である。後ほど改めて詳しく述べる。バイオメトリクス認証とは、生物学的特徴を用いた認証方法で、生体認証とも呼ばれている。

1.2 パスワード認証

サーバー B に対するユーザー A のパスワード認証は次

†1 現在，九州大学
Presently with Kyushu University
a) murata.issei.662@s.kyushu-u.ac.jp

で与えられる。

$$A \to B : id_A, p_A$$

ただし、A の ID である id_A、および A のパスワードである p_A は AB 間で共有されているものとする。これは、A が B に対して ID とパスワードを送付する単純なパスワード認証である。これを共通鍵暗号を用いてパスワードを暗号化して保護する方式が以下で与えられる。

1. $A \to B : id_A, X = E_{p_A}(p_A)$
2. $B : p_A = D_{p_A}(X)$

A が B に対して ID と暗号化されたパスワード X を送付し、B が X を復号して p_A を得て認証する。なお、パスワードを p_A としたとき、E_{p_A} は暗号化関数、D_{p_A} は復号化関数を表す。

次に、パスワードを利用した 3 つの方式について紹介する。

イ．チャレンジレスポンス方式

パスワードを直接やり取りすることなくリプレイ攻撃を防ぐ方式であり、ユーザー A がサーバー B からチャレンジ（問題）を受け取り、そのレスポンス（応答）をして認証するため、チャレンジレスポンス方式と呼ばれる。ユーザー A の秘密鍵 k_A が AB 間で共有されているとき、共通鍵暗号ベースのチャレンジレスポンス方式の手順は以下の通りである。

1. $A \to B : id_A$
2. $B \to A : r$
3. $A \to B : c = E_{k_A}(r)$
4. $B : r \stackrel{?}{=} D_{k_A}(c)$

A が B に対して ID を送付し、チャレンジを要求する。次に B は乱数 r を作成して A に送付する。A は k_A で r を暗号化しレスポンス $c = E_{k_A}(r)$ を計算して、c を B に送付する。最後に B が c を復号して r を得て認証する。レスポンス c を計算できるのは正規のユーザー A のみであることから、認証が可能となる。このとき、B がチャレンジ r を毎回新たに生成することでレスポンス c が毎回変わるため、リプライ攻撃を防止できる。

ロ．ワンタイムパスワード方式

認証の度に異なるパスワードが使用され、パスワードを共有しない効率的な認証方法である。チャレンジレスポンス方式では、サーバー側にユーザーのパスワードを保存する必要があるため、サーバーがサイバー攻撃に遭うとそのパスワードが漏洩するリスクがあった。しかし、本方式ではパスワードがサーバー側に保存されないため、たとえサーバーがサイバー攻撃を受けたとしてもユーザーのパスワードが漏洩することはない。

ワンタイムパスワードを実現するためにはハッシュチェーンが利用される。ワンタイムパスワード方式の手順は以下の通りである（サーバー b が x_n のハッシュ回数 n を管理する）。

(1) 登録フェーズ

1. $A : x_1 = H(k_A), x_2 = H(x_1), \cdots, x_n = H(x_{n-1})$
2. $A \to B : id_A, x_n$

(2) 認証フェーズ（i 回目の認証の場合）

1. $A \to B : id_A, x_{n-1}$
2. $B : x_n \stackrel{?}{=} H^i(x_{n-i})$

登録フェーズにおいては、ユーザー A はマスターシークレット k_A を選び、k_A に対してハッシュ関数を n 回繰り返して長さ n のハッシュチェーンを計算する。そこからこのハッシュチェーンの最後の値 x_n をサーバー B に認証チャネルを通じて送信する。このとき、x_n はハッシュチェーンのコミットメントとなる。認証フェーズにおいて（i 回目の認証の場合）は、A は B に対して ID と現在のパスワード x_{n-i} を送信する。x_n はコミットメントされているため、x_n を用いて x_{n-i} を検証できる。この方式により、A はマスターシークレット k_A を漏らすことなく、B に対して認証できる。

ハ．シングルサインオン（SSO）

シングルサインオン（Single Sign-On）は、「シングル」と「サインオン」を組み合わせた造語で、「1 度システムを利用開始のユーザー認証（ログイン）を行うと、複数のシステムを利用開始する際に、その都度認証を行う必要がない仕組み」や「1 度の認証で、以後その認証に紐づけらている複数のシステムやアプリ・サービスにも、追加の認証なしで利用できる製品・システム・ツール」を意味する。

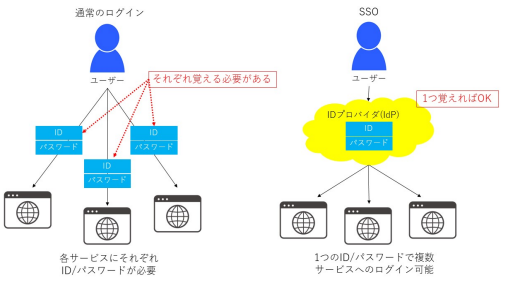

図 1 通常のログインと SSO の比較

認証方式の仕組みとして、ここでは 2 つ紹介する。

(a) SAML 認証

「SAML(Security Assertion Markup Language) 認証」とは、ID 管理と認証を行う ID プロバイダ（IdP）で保障されたユーザー認証情報を利用することで、連携している各種サービスのシングルサインオンが可能になる仕組みである。図 2 はクラウドサービス（SP）を起点とした場合の認証フローを示している。SP 起点の他に、IdP を起点と

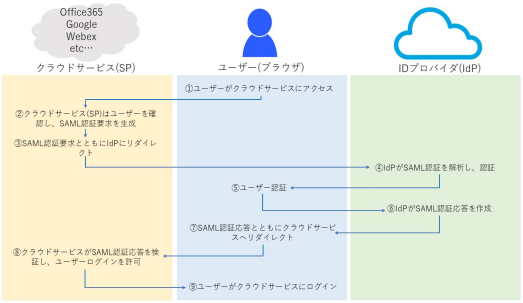

図 2　SAML 認証（SP 起点の場合）

した場合も存在する。

（b）フォームベース認証（代理認証方式）
シングルサインオンのシステムに登録した ID とパスワードを他システムのログインフォームに代理入力することでログインする仕組みである。

1.3　深層学習

深層学習は簡潔にまとめると、ニューラルネットワーク（neural-network）と呼ばれる概念を用いて行われるデータの処理技術を指す。ニューラルネットワークは、脳の構造の模倣から始まった数学的な概念で、入力されたデータや信号を変換するシステムを実現する。実際にデータを解析する際には、我々はコンピュータ上にニューラルネットワークを構成し、それを用いて入力されたデータを変換する。

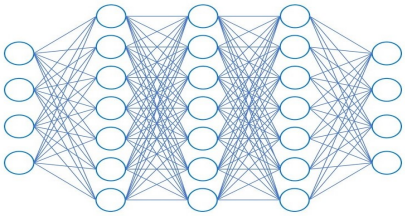

図 3　5 層を持つニューラルネットワーク

図 3 はニューラルネットワークの構成を図で表したもので、丸のそれぞれをノードもしくはニューロンと呼び、それらを繋ぐ線をエッジと呼ぶ。このニューロンの列そのものを層と呼び、各層のデータを順番に変換する（挟まれたエッジの集合を層と呼ぶ場合もあり、これで言えば図 1.2 は 4 層となる）。この層が多い（特に 3 層以上）状態を層が深いと呼び、層が深いニューラルネットワークを用いたデータ分析技術を深層学習と呼ぶ。

（1）深層学習とは何をするのか
深層学習は何をしているのか？という反駁な問いに一言で答えるのであれば「コンピュータ上で“関数”を構成しいている」である。関数とは、ある値が入力されたときに、それに対応する値を出力する数学的な概念である。深層学習では、複数の数値が同時に入出力されるので、深層学習で作る関数はベクトル（複数の数値の列）を出力する。関数などの数学的概念を作る方法をモデルと呼び、ニューラルネットワークはモデルの一種とみなせる。

（2）ニューラルネットワークについて
ニューラルネットワークは複数の層の重ね合わせで構成され、各層では①式で表されるベクトル変換が行われる。

$$f(x) = \eta(Ax+b) \quad \cdots①$$

ここで A と b はパラメータ（もしくは重み）と呼ばれ、b は数値を列に並べたベクトル、A は数値を縦横に並べた行列という構造をもつ。$\eta(\cdot)$ は活性化関数と呼ばれる関数である。また、x は入力されたベクトルとする。すなわち、各層においては、入力されたベクトル x にパラメータ A をかけてパラメータ b を足し、最後に活性化関数 $\eta(\cdot)$ で変換するという、変換セットで行っている。以下、ベクトル b、x と行列 A と、変換されたベクトルの式を示す。

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_d \end{pmatrix}, A = \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{d1} & \cdots & a_{dd} \end{pmatrix}$$

$$\eta(Ax+b) = \begin{pmatrix} \eta(\sum_{j=1}^{d} a_{1j}x_j + b_1) \\ \eta(\sum_{j=1}^{d} a_{dj}x_j + b_d) \end{pmatrix}$$

ニューラルネットワークの内部では、この変換が層の数だけ繰り返される。用いるパラメータ A と b は層ごとに異なる。図 4 のニューラルネットワークは 5 層を

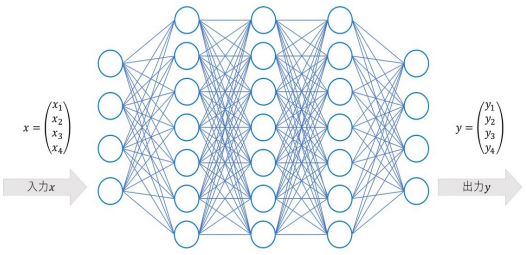

図 4　5 層を持つニューラルネットワークにおける入出力

持っているため、入力されたベクトルに対して変換が 4 回繰り返されている。i 層目の変換を $f_i(x)$ と書くと、L 層を持つニューラルネットワークによる関数は④式のように書ける。

$$f_{L-1}(f_{L-2}((\cdots f_2(f_1(x))))\cdots② \quad \cdots②$$

（3）原理の分からない深層学習
深層学習が素晴らしい性能を発揮する実例はいくつも存在する。しかしながら、深層学習の原理は未だに完

全には理解されていない。すなわち、深層学習が既存のデータ分析手法より良い性能を発揮できる原因は、十分には解明されていない。原理を理解するということを数学に基づいた理論による理解と考えると、深層学習の挙動は数学の既存理論（深層学習以前のデータ分析手法の理解に貢献してきた）と矛盾していることが明らかになっている。しかし、深層学習は理論的な主張と正反対の方法も用いて高い性能を実現している。この実際と理論のギャップが深層学習の理解を阻んでいるのが現状である。

2. 先行研究

2.1 人間計算可能なパスワードの定義

2.1.1 パスワードについて

本パスワードはチャレンジへの応答である。特に、このパスワードはパスワードスキーマによって生成されるが、そのスキーマは（チャレンジ、鍵）のペアからパスワードにマッピングするための人間が計算可能なアルゴリズムである。パスワードは必要なときに簡単に生成でき、たとえ攻撃者がユーザーのスキーマを知っていて、少数の Web サイトのパスワードを見たとしても、攻撃者が偽造困難でなければならない。より一般的には、分析可能、公開可能、人間的に使用可能、安全、自己リハーサル可能なスキーマを求める。

(1) 分析可能…スキーマが非常に正確に定義されているため、チューリングマシンで実行できること
(2) 公開可能…スキーマ自体（ユーザーの秘密鍵）が公開できること
(3) 人間的に使用可能…次の 3 つのことを意味する 1) スキーマ自体（秘密鍵ではなく）が数分で学習可能であること、2) 秘密鍵の生成と記憶が、多くても 1 時間、できれば 30 分以内でできること、3) パスワードの生成または再生成が 1 分以内、できれば 20 秒以内で可能であること。
(4) 自己リハーサル可能…自己リハーサルとは、時折発生するランダムな課題に対応する過程で、ユーザーがスキーマのあらゆる面をリハーサルすることで、それが可能であること

2.1.2 人間計算について

多くの暗号問題やゲーム（スピードチェス、数独）では、人間の計算は前処理フェーズの PRPE（公開スキーマの記憶に加え、個人情報や鍵の生成と記憶）に加え、処理のフェーズの PROC（入力に関連付けらた鍵をもつスキーマの実行）で構成される。ここではまず、人間を以下のようにモデル化する。

1. 人間はチューリングマシンの一種であり、通常のテープを 2 つのランダムアクセスメモリ（１つは長期用、もう 1 つは短期用）に置き換えたもの。
2. 長期記憶は無限に広がる可能性があり、その上限は人間の使用可能な寿命と、その記憶に情報を格納するのにかかる時間だけである。スキーマやキーなどの情報を長期記憶に保存するのは時間がかかるが、長期記憶から情報を読み出すのは、記憶上の位置へのポインタがあれば比較的速い。長期記憶は、前処理段階において、書き込まれ、読み出される。処理段階では、長期メモリは読み出しにのみ使用される。
3. 短期読み書きメモリは高速だが小さく、通常 2 つか 3 つのチャンクを格納し、各チャンクは数字や数値、文字や単語、画像や音楽クリップなど、何らかのアイテムへのポインタである。
4. パスワードにおいては、入力（チャレンジと呼ばれる）は単一連結リストとして提示され、その左端の開始位置へのポインタが短期メモリに格納される。チャレンジのどの位置にあるにせよ、ポインタは 1 ステップで 1 リンク右に移動するか（左には移動しない）、読むか（過去形）、開始位置にリセットすることが可能である。

一般的にスキーマは、パスワードスキーマだけでなく、人間を対象としたアルゴリズムである。これらは、パスワード用のキーと呼ばれるパラメータを含む永久メモリに格納された情報と関連付けて使用される。人間はサイコロ、紙、鉛筆、及びその他のツールを使用して、秘密鍵を生成・記憶し、公開スキーマを記憶する（PRPE）ことができる。その後、頭の中でスキーマを実行（PROC）しなければならない。

2.1.3 人間計算可能な条件

まず、PRPE（公開スキーマの記憶と秘密鍵の生成と記憶に関する前処理）を MEM（秘密鍵の生成と記憶、加えて記憶保持のために必要なリハーサル）と COMM（スキーマの記憶・学習）に分ける。これより、スキーマと鍵の組み合わせは、以下の要件を満たす場合に限り、人間が使用できるとみなす。

1. COMM-TIME はスキーマを学習する上限である。この時間は、いくつかのサンプルチャレンジをパスワードに変換する時間とすべてのリハーサル時間が含まれている。これは最大 10 分程度でなければならない。
2. MEM-TIME は、スキーマに関連する秘密鍵を生成し、記憶する時間、およびその記憶を維持するために必要なすべのリハーサル時間の上限である。パスワードの場合、これは最大で 2 時間であることが要求される。
3. PROC-TIME は、1 つの入力に対してスキーマを実行する時間の上限である。パスワードの場合は、これは最大で 1 分であることが要求される。
4. スキーマは最大 3 つ（できれば最大 2 つ）の長期・短期記憶へのポインタ（チャンク）を使用する。
5. パスワードの場合、スキーマとそれに関連する鍵（両方とも長期記憶に保存される）は、完全に自己リハー

サルされる。つまり、一つ一つの命令が実行され、鍵のすべての要素がチャレンジ・レスポンス計算の「かなりの割合」でリハーサルされる。

改めて、時間的条件を以下に示すと、

$$\text{COMM-TIME} \leq 10min$$
$$\text{MEM-TIME} \leq 2h$$
$$\text{PROC-TIME} \leq 1min$$
$$\text{ポインタ（チャンク数）} \leq 3$$

となる。

2.2 人間計算可能なパスワード

2.2.1 提案

ユーザーが複数のパスワードを作成、記憶できるような体系的な戦略、使いやすく安全なパスワード管理スキームを開発することを目標として研究が行われた。多くの侵害があった後でもセキュリティが強く維持される（例えば、ユーザーのパスワードを 100 個見た敵は、ユーザーの残りのパスワードについて依然として高い不確実性を持つ）人間計算可能なパスワード管理方式の開発を焦点にした。人間計算可能なパスワード管理方式では、ユーザーは公開チャレンジに対する応答を計算することで各自のパスワードを再構築する。

ここで提案される人間計算可能なパスワード方式はユーザーが頭の中で単純な計算（例えば、2 つの 1 桁の足し算）をすることを要求する。ここで注意すべきは、この人間計算可能なパスワード方式は、やる気のあセキュリティ意識の高いユーザーが数時間で方式を使用することを学習し、関連するすべての秘密を記憶できるという意味で、人間が使用可能である。特に、本方式における人間の計算は、ユーザーが記憶した秘密値に対するいくつかの非常に簡単な演算（例えば、mod 10 の加算）のみを含む。

2.2.2 具体的な提案

ユーザーは秘密マッピング $\sigma : [n] \to Z_d$ を記憶し、単純な関数 $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ を計算することを学習する、。ユーザーは一連の 1 桁チャレンジに応答することによって認証される。ここで、チャレンジ $C \in X_k \subseteq [n]_k$ に対し、それに対応するレスポンスを $f(\sigma(C))$ とすると、チャレンジ・レスポンスペア $(C, f(\sigma(C)))$ と表される、

これより、人間計算可能なパスワードに用いられる、人間計算可能な関数に注目する。人間計算可能な関数（人間が計算可能な関数の候補 $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ について、ほとんどの人間は数字の算術に慣れているので、$d = 10$ を固定する）の候補として次の式が挙げられる。与えられた整数 $k_1 > 0$ と $k_2 > 0$ に対して、関数 $f_{k_1,k_2} : \mathbb{Z}_{10}^{10+k_1+k_2}$ を以下に定義する。

$$f_{k_1,k_2}(x_0, \cdots, x_{9+k_1+k_2}) = x_j + \sum_{i=10+k_1}^{9+k_1+k_2} x_j \ mod\ 10 \cdots ③$$

$$j = (\sum_{i=10}^{9+k_1}) \ mod\ 10 \cdots ④$$

また、ユーザーが n 枚の画像から数字への秘密マッピング σ を記憶する場合、画像を $I_i (1 \leq i \leq n)$ とすると、秘密マッピング σ は、

$$\sigma(I_i) = \{x_i \mid 0 \leq x_i \leq 9\} \cdots ⑤$$

で表される。ここで関数 $f_{k_1,k_2}(x_0, \cdots, x_{9+k_1+k_2})$ の x の添字と、マッピング関数 σ の x の添字は異なることに注意する。

2.2.3 1 桁チャレンジ

今回扱う人間計算可能なパスワードにおいて、入力するパスワードに相当するものが 1 桁チャレンジの答え（応答）になる。以下、1 桁チャレンジについて具体的に述べる。ここでは、関数 $f_{2,2}(k_1 = 2, k_2 = 2)$、$n = 100$（画像の枚数 100 枚）の条件下で行うものとする。

(1) 事前知識

事前知識については、画像 n 枚に対するマッピング関数 $\sigma(I_i)$ を記憶することである。ユーザーが記憶するマッピングは図 5 の表で表される。

図 5 マッピング関数 σ の表

(2) 1 桁チャレンジへの応答

ユーザーはパスワード入力画面に提示された 14 枚（$k_1, k_2 = 2$ より）の画像をもとに、関数 $f_{2,2}$ に従って頭の中で計算（暗算）を行う。図 6 では、14 枚の画像の表示形式の一例と頭の中で行う処理（暗算の内容）を示している。図 6 では 1 桁チャレンジの応答は「0」

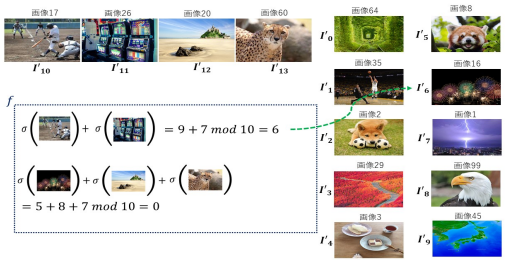

図 6 1 桁チャレンジを解く

になることが分かる。パスワードの入力の際は、この 14 枚の画像が記憶した画像の中からランダムに選択され、1 桁チャレンジに応答する度に、14 枚の画像が切り替わる。これを λ 回繰り返すことで、λ 桁のパスワードが生成されることになる。図 6 において、14 枚の画像 $I'_i (0 \leq i \leq 13)$ における画像番号 i はこの 14 枚

の画像をナンバリングしたにすぎず、図 5「マッピング関数 σ の表」における画像番号とは異なることに留意すること。

2.2.4 人間計算可能なパスワードの要件を満たすか

(1) 関数 $f \circ \sigma$ について

2.2.2「具体的な提案」で提案した関数 f_{k_1,k_2}（以後、単に f と表記する場合あり）を $\sigma(I_i)$（以後、単に σ と表記する）の合成関数 $f \circ \sigma : X_k \to \mathbb{Z}_d$ が人間計算可能であることを示す。

(a) 要件 1

非常に単純なプリミティブ演算のみを用いて f を計算する高速ストリーミングアルゴリズムがあるときは、いつでも関数 f は人間が計算できる。人間のストリーミングアルゴリズムは、人間が頭の中で素早く実行できるプリミティブの操作だけを含むことができる

(b) 要件 2

P をプリミティブ演算の集合とする。ここでは、Add、Recall、TableLookup というプリミティブ演算 P について考える。

- Add：$\mathbb{Z}_{10} \times \mathbb{Z}_{10} \to \mathbb{Z}_{10}$ は x_1 と x_2 の 2 桁をとり、$x_1 + x_2 \ mod\ 10$ を返す
- Recall：$[n] \to \mathbb{Z}_{10}$ は、インデックス i をとり、ユーザーが記憶している秘密値 $\sigma(i)$ を返す
- TableLookup：$\mathbb{Z}_{10} \times [n]_{10} \to [n]$ は、数字 x_1 を受け取り、10 個のインデックスからなるテーブルから x_1 番目の値を検索する

これより、関数 $f \circ \sigma$ の評価を行う。

まず、j（④式）の計算は $j = x_{10} + \cdots + x_{9+k_1}$ となるので、k_1 個の足し算であるから、Add：$k_1 - 1$ 回、Recall：k_1 回である。次に、画像 I'_j を探すので、TableLookup：1 回。最後に関数 f（③式）は $f = x_j + x_{10+k_1} + \cdots + x_{9+k_1+k_2}$ となるので、$k_2 + 1$ 個の足し算であるから、Add：k_2 回、Recall：$k_2 + 1$ 回となる。以上より、プリミティブ演算 $P\{Add, Recall, TableLookup\}$ の合計回数は $2k_1 + 2k_2 + 1$ 回である。また、必要な空間 $\hat{m} = 3$ である。そして、1 つのプリミティブ演算にかかる時間を 1 秒とする。

ここで、特に $k_1 = 2, k_2 = 2$ の時 $f \circ \sigma$ について考えると、計算にかかる時間は理論上 9 秒である。これに対して、論文の筆者は適度な数学的背景をもつ人間であれば、7.5 秒で計算可能であるとした。

(2) 人間計算可能なパスワードの要件との比較

関数 $f \circ \sigma$ を 2.1.3「人間計算可能な条件」における条件に照らし合わせる。まず、COMM-TIME に関しては、記述がなかった。次に、MEM-TIME に関して、$n = 100$（画像が 100 枚（数字の対応を 100 個記憶するということ））のとき、約 2 時間。PROC-TIME に関して、$k_1 = 2, k_2 = 2$ のとき、理論的にはプリミティブ演算が 9 つであるから 9 秒、筆者は 7.5 秒。ポインタ（チャンク数）に関して、空間 $\hat{m}$ をチャンク数とみなし、3 である。以上より、関数 $f \circ \sigma$ はおおかた条件を満たすと言えるので、人間計算可能なパスワードといえる。

2.3 CSPsolver による人間計算可能なパスワードの強度評価

先行研究では、この 1 桁チャレンジの問題（14 枚の画像（正確には画像番号））と応答から、マッピング関数 σ を予測することが可能であるか否かについて CSPsolver を用いて実験が行われた。CSPsolver による計算は、2.83GHz Intel Core2 Quad CPU と 4GB の RAM を搭載した PC で行われた。その結果を図 7 に示す。

	$m = 50$	$m = 100$	$m = 300$	$m = 500$	$m = 1000$	$m = 10000$
$n = 26$	23.5h	40min	4.5h	29min	10min	2min
$n = 30$	HARD	UNSOLVED	2.33h	35.5min	10min	20s
$n = 50$	HARD	HARD	HARD	HARD	UNSOLVED	7h
$n = 100$	HARD	HARD	HARD	HARD	HARD	UNSOLVED

図 7 CSPsolver による解読 [Towards Human Computable Psswords より]

ソルバーが 2.5 日以内にマッピング関数 σ を答えられなかった場合 (n, m) はＨＡＲＤとし、(n', m') で $n' \geq n, m' \leq m$ に対してはソルバーを実行していないので UNSOLVED と表記している。

CSPsolver による実装に関して説明する。まず、入力としてデータセットと、関数 f を受け取る。データセットは、14 枚の画像番号と 1 桁チャレンジの応答のデータ（図 7 の m とはそのデータの数である）である。m 個のデータと関数 f を制約条件として、それらの満たすようなマッピング関数 σ を見つける処理を行っている。つまり、チャレンジ C、マッピング関数 σ、関数 f において、$(C, f(\sigma(C)))$ をデータセットとして、データセット $(C, f(\sigma(C)))$ と関数 f を制約条件としてマッピング関数 σ を求めるという、制約充足性問題 (CSP=Constraint Satisfaction Problem) を解くということである。

3. 研究手法の提案

3.1 課題と貢献

3.1.1 本研究における課題

主に 2 点を考えている。人間計算可能なパスワードについて、1）深層学習を用いて評価（予測）可能であるのか、さらには、評価可能であれば、2）深層学習による安全性の評価かが可能であるか、である。この 2 点を課題におき、研究を進めている。

3.1.2 本研究の情報学的な貢献

主に 2 点考えている。1）深層学習による人間計算可能なパスワードの評価（可能であると否かも含めて）、2）深層学習に対して安全性が高い人間計算可能なパスワードの提案・考察、である。
加えて、深層学習は第 3 次 AI ブームの中枢であり、その認知度は上がっているといえるだろう。このような現状を踏まえても、深層学習を使って研究をすること自体に意味（貢献）があると考えている。

3.2 深層学習による研究方法

深層学習といっても、そのモデルは複数存在する。例えば、多層パーセプトロンや RNN、CNN、GAN などである。今回は、その中でも基本的なモデルである多層パーセプトロンを用いて検証を行った。

3.2.1 多層パーセプトロンについて

多層パーセプトロン（MLP:MUltilayer perceptron）は、（単純）パーセプトロンを複数繋いで、多構造にしたニューラルネットワークである。単純パーセプトロンは、入力層を出力層のみであるのに対し、MLP は中間層と呼ばれる層を複数もつネットワーク構造である。

3.2.2 プログラムの内容について

まず、本研究では、「深層学習を用いて、1 桁チャレンジの答えを予測できるのか？」という点を実験する。
※ 1 桁チャレンジの答えを予測するとは、つまりは、合成関数 $f \circ \sigma$ の出力を予測していることと同義である。それを踏まえ、先行研究における CSPsolver での実験では、マッピング関数 σ を解読できるか否かを評価しているが、深層学習では、1 桁チャレンジの応答をどれぐらいの精度で予測できるか、という点で実験を行っていることに注意する必要がある。CSPsolver によるマッピング関数 σ の解読ができた場合は、1 桁チャレンジの応答を 100 %で予測できることを意味している。

（1）データセットに関して

データセットは、1 桁チャレンジで提示される 14 枚の画像の画像番号 (マッピング関数 σ に置ける 1 ～ n 枚の画像にナンバリングがしてある)(図 3.2 では「X0～X13」) と、1 桁チャレンジの答え（0 ～ 9 の整数）（図 8 では「Z」）をデータセットとして受け取る。図 3.2

(1桁チャレンジで提示される14枚の画像の番号)														(答え)
X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	Z
19	16	25	3	9	47	90	62	48	51	69	4	77	46	4
1	45	28	99	63	46	24	33	70	75	65	54	58	13	6
							⋮							

図 8 データセット

のとき、画像の枚数 n は 100 のときのデータセットである。

（2）学習に関して

データセットを訓練データとテストデータに分けて、学習データを用いて、ＭＬＰに学習させる。MLP における入力データは画像番号である、出力データは 1 桁チャレンジの答えである。

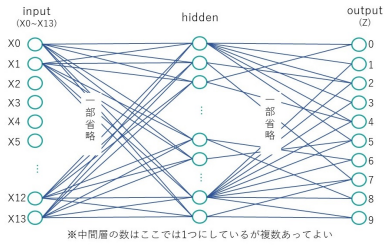

図 9 MLP モデル

ニューラルネットワークの特性上、出力値の 0 ～ 9 は各ニューロンで分類されるので、出力層は 10 個のニューロンが存在する。

4. 結果・考察

4.1 実験 1：関数 $f \circ \sigma$ での検証

CSPsolver での実行結果と比較できるように、$n = 26, 30, 50, 1000 \cdots$ でそれぞれ実験を行った。MLP の中間層やノード数に関しても、複数パターンで実行した。その結果を図 10 に示す。

データ数	中間層の数	中間層のノード数	予測精度 $n = 26$	予測精度 $n = 30$	予測精度 $n = 50$	予測精度 $n = 100$
1000	3	128,64,32	0.0800	0.0975	0.1033	0.0800
1000	4	128,64,32,16	0.0900	0.0850	0.1133	0.0950
1000	4	256,128,64,32	0.1067	0.0850	0.1033	0.0800
10000	3	128,64,32	0.1017	0.1050	0.0980	0.1060
10000	4	128,64,32,16	0.0987	0.0860	0.0890	0.0905
10000	5	256,128,64,32,16	0.0980	0.0915	0.1017	0.1010
100000	4	128,64,32,16	0.1174	0.1013	0.1012	0.0991
100000	5	256,128,64,32,16	0.1045	0.1041	0.0996	0.0976
100000	6	512,256,128,64,32,16	0.1022	0.1032	0.0990	0.1007

図 10 実行結果 1

n の値、中関層、ノード数によらず、予測精度は 10 %前後であった。

4.2 実験 2：関数 f での検証

実験 1 において、関数 $f \circ \sigma$ が予測精度 10%で、マッピング関数 σ が関数を複雑にしている原因であるを考え、関数 f の予測が MLP で予測できるか否かを実験した。この時学習データは 14 個の 0 ～ 9 と、関数 f の出力を組み合わせたデータとなる。結果を図 11 に示す。
関数 f の予測に関しても、MLP では予測精度は 10%前後であった。ここで、関数 f は③で与えられるが、j の値が④式によってその都度変わることが MLP による予測を難しくしている要因と考えた。そのため実験 3 を行った。

f			
データ数	中間層の数	中間層のノード数	予測精度
1000	3	128,64,32	0.0900
1000	4	128,64,32,16	0.1100
1000	4	256,128,64,32	0.0850
10000	3	128,64,32	0.0965
10000	4	128,64,32,16	0.0905
10000	5	256,128,64,32,16	0.1070
100000	4	126,64,32,16	0.1026
100000	6	512,256,128,64,32,16	0.1297

図 11 実行結果 2

4.3 実験 3：関数 g での検証

関数 f の入力に対して、3 つの和を答えとするが、その 3 つのうちの 1 つが④の j によって変わるので、この値を固定した場合に関して実験を行った。そのため新たに関数 g を次のように定義した。

$$g(x_0,x_1.\cdots,x_{13}) = x_i + x_{12} + x_{13}\ i \text{ は固定}$$

以下結果を示す。

関数g	データ数	中間層の数	中間層のノード数	予測精度
$x_2+x_{12}+x_{13}$	10000	3	128,64,32	0.7896
$x_2+x_{12}+x_{13}$	10000	4	128,64,32,16	0.742
$x_5+x_{12}+x_{13}$	10000	3	128,64,32	0.863
$x_5+x_{12}+x_{13}$	10000	4	128,64,32,16	0.6155

図 12 実行結果 3

この場合の予測精度は 80%前後で、関数 f の予測制度に比べて 70 %近く上昇した。

4.3.1 全体的な結果・考察

- 関数 $f\circ\sigma$ と f は予測できない結果となった。
  - 10%が予測できないと結論付ける理由は、出力として考えられる値は 0～9 の 10 個の値である。故に、仮にランダムで予測したとしても確率論的に 1/10（10%）で正解するからである。
- 入力に対して、使う値が定まっている線形関数 (関数 f は出力に関わる③式は線形であるが、j が変わるが、関数 g は定まっているという意味) は精度 80%で予測可能である。

5. おわりに

論文で提案された関数 $f\circ\sigma$ を深層学習を使って予測を試みたが、今回用いた MLP モデルでの予測はできない結果となった。加えて、関数 f も予測できなかった。一方で、関数 g のように入力に対して使用する値が定まっている線形関数は深層学習を用いて 80%の精度で予測可能であることが分かった。

深層学習を用いた先行研究がないため、方向性を立てることが難しい側面もある。また、現在、深層学習が用いられているデータ分析を見ても、入力データ同士に相関がある場合が多く、今回の場合のように入力データ同士が独立的な場合に深層学習がその性能を発揮できるか否かが分からない部分がある。

しかし、今回は基本的なモデルである MLP を用いたが、深層学習には他にもモデルが存在する。そのため、本研究のテーマである「人間計算可能なパスワードの深層学習の強度評価」に対して、評価が可能であるか否かの部分も含めて、結論を下すのは、より多くの深層学習の手法を試してからになるのではないだろうか。

謝辞 本研究を進めるにあたり，ご多忙にもかかわらず熱心にご指導頂きました，櫻井幸一教授に心から感謝致します．また, 研究に協力頂きました櫻井研究室の皆様に心から感謝致します.

参考文献

[GZB19] Yushuo Guan, Yuanxing Zhang, Lin Chen, Kaigui Bian. A Neural Attack Model for Cracking Passwords in Adversarial Environments. IEEE International Conference on Communications in China (ICCC). 2019. 183-188

[BBDV17] Jeremiah Blocki, Manuel Blum, Anupam Datta, Santosh Vempala. Towards Human Computable Psswords. Innovations in Theoretical Computer Science (ITCS). 2017. 10:1-10:47

[BV20] Manuel Blum, Santosh Vempala. The complexity of human computation via a concrete model with an application to passwords. Proceedings of the National Academy of Sciences of the USA, Volume117 (PNAS). 2020. 9208-9215

[RVB19] Elan Rosenfeld, Santosh Vempala, Manuel Blum. Human-Usable Password Schemas: Beyond Information-Theoretic Security. Computing Research Repository, June 2019.

[Ima21] 今泉允聡. 深層学習の原理に迫る数学の挑戦. 岩波書店. 2021

[CYB] CYBERNET. シングルサインオン. https://www.cybernet.co.jp/onelogin/function/saml.html. 閲覧日 2021/12/20

[SSO] トラスト・ログイン. シングルサインオン（SSO）とは. https://trustlogin.com/sso/. 閲覧日 2021/12/20