

## 画像認識 AI の理解とそのプログラムの製作

石川県立金沢泉丘高等学校理数科

村山 愛理子 砂本 祐輔 近岡 倫太郎 長屋 龍司 谷内 拓哉

### 1. 1 研究背景

「人工知能」が一つの研究分野を指し示す様になったのは、「人工知能に関する研究会議」が 1956 年の夏に米・ダートマス大学で行われて以来である。それ以来約 60 年に渡り、

- ・機械学習
- ・感性処理
- ・遺伝アルゴリズム
- ・知識表現
- ・医学、芸術、数学への応用
- ・感性処理
- ・ヒューマンインターフェース

等、多岐にわたって人工知能に関連する研究が行われている。

今となつては、

- ・ゲーム AI がボードゲームの世界王者に勝利する
- ・人間の特定の動きに合わせて反応を取る
- ・機器周辺の情報を集めてその情報を計算、予測し、機器を動かす

といった事も可能になっており、これからも進化が期待される分野であるため、私達はこれに興味をもち、AI を作成することを通じてその手法を学び研究する事となった。中でも、顔認識等で特に私たちに身近な画像認識についての研究を行う事にした。

### 1. 2 研究目的

画像認識、とくに文字を認識する AI プログラムの製作を最終的な目標とする。認識の際は書き順や画数などの情報を使わず、画像データのみを使って文字を認識させる。

現時点で Google の OCR 機能が存在し、印刷物などの活字は高精度で読み取ることが可能であるが、手書き文字に対応できておらず精度が低い。そのため、本研究では特に手書き文字（点画の省略や個人の癖がみられる文字）へ対応したプログラムの製作を目指す。

### 2. 研究方法

AI のシステムの理解とともに、プログラミング言語の学習を行う。プログラミング言語は Python を用いることとし、その中で TensorFlow というライブラリを使用して実験を進めていく。Python を選んだ理由としては、現在プログラミング、とりわけ AI の分野に於いてこの言語が多用されており、情報をインターネットで収集しやすいことが挙げられる。又、TensorFlow は AI を作成するにあたって文章量を大幅に短縮できる。Google が提供しているライブラリであり、これを使用することによって内容の簡潔化を図れるのでこのライブラリを使用した。

### 3.1 実験の準備

まず、Python の総合開発環境（プログラミングをする上で必要なソフト）として、Anaconda をダウンロードする。Anaconda は、科学計算技術を中心とした、多くのツールが始めから入っており、AI を作成するうえで利便性が高い。その後、Anaconda 上でコマンドプロンプトを起動して、TensorFlow をダウンロードする。Anaconda にはデフォルトで TensorFlow が入っていないので、別途 DL する必要がある。

### 3.2 AI の学習・収集

AI の作成及び理解を行うために、参考文献を元に本研究を進めていく。

まず、参考文献を元に、最も簡単（基礎）である数字の認識 AI の作成を行う。画像認識において必要となるのはデータセットと呼ばれる画像の集まりである。本来はこのデータセットの収集にはかなりの手間と労力が必要となる。そこで、Mnist と呼ばれるフリーのデータセットを用いて試行を行った。Mnist はこの手の画像認識で多用されている。

また、今後の平仮名や漢字の認識においてもデータセットが必要となると考え、それらのデータの収集も行う。当初は人力で回収しようとしていたが、インターネット上でデータを配布しているサイトからデータを引用した。そのデータには、1文字につき 200 枚の画像データが入っており、文字は常用漢字の JISCode に含まれる全日本語の計 3000 文字以上が入っている。

### 3.3 AI の仕組み

情報分野というのはなじみのない分野だと思われるため、最も基本的な AI の仕組みを画像認識の AI を例として説明していく。AI は基本的に 3 つの層で構成されており、一つ目の初めの層は入力層と呼ばれる層で、ここに画像のデータを入力する。画像データは 1 列の行列で表すことができるので、これを入力データとして入力する。次の中間にある層は、隠れ層と呼ばれ、ここで計算を行う。計算の際に、バイアス（重み値）と呼ばれるものを掛け、この値を変更することによって AI の精度を上げることが出来る。隠れ層は何層も重ねる事ができ、重ねれば重ねるほど精度は上がるが、その分プログラムも重くなり、また、重ねていっても最終的に結果が収束してしまうので、隠れ層の層数も考慮する必要がある。最後の 3 層目は、出力層と呼ばれるもので、ここで計算結果、つまりは画像データの答えを出力する。この答えは、確率で出され、その中で一番確率の高いものを答えとして出力する。その際に出される確率によってバイアスの値の変更を行う。その確率群から、その AI がどれだけ間違えたかの指標である損失関数を求められる。バイアスの値を変えると当然結果も変わるので、それをもとに AI の精度を上げていく。画像 1 枚 1 枚にその処理を行い、その試行を一定回数行うことで最終的な AI の出力となる。そして、そのバイアスの値を保持することで、AI を手軽に実用できる。

### 3.4 CNNについて

畳み込みニューラルネットワーク、通称CNNは、画像認識のニューラルネットワークの仕組みを複雑にしたものである。前記のニューラルネットワークは、ただの画像のデータの羅列のため、線のつながりやその文字の特徴を認識出来ない。CNNは、その点をクリアしたもので、画像に対し、幾枚のフィルタをかけて、画像の特徴を抽出する。このフィルタがCNNにおけるバイアスになり、フィルタの値を学習によって変化させていくことによって、AIの学習を図る。前記のニューラルネットワークよりも高精度を実現できる反面、学習により多くの時間がかかる。

### 3.5 結果の測定

結果の測定には、プログラム上での測定と、TensorFlowの機能の一つであるTensorBoardを使用して行う。TensorBoardにはAIの構造や、指定したデータのグラフ化など、様々な便利機能が含まれている。それを用いてAIの精度の測定を行う。また、AIは訓練段階で多くの施行を行って精度を高めていくという性質上、複数回の試行はそれほど効果がないので、試行は最低回数のみ行う。

#### 4.1 研究結果

右上のグラフ(図1)は縦軸に損失・誤差、横軸に試行回数をとった損失関数、グラフ(図2)は縦軸に精度、横軸に試行回数をとったものである。

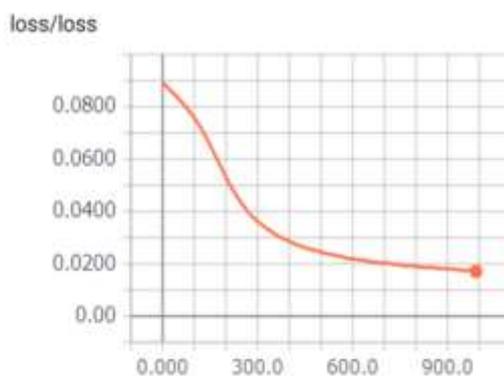


図1 縦軸:損失, 横軸:試行回数

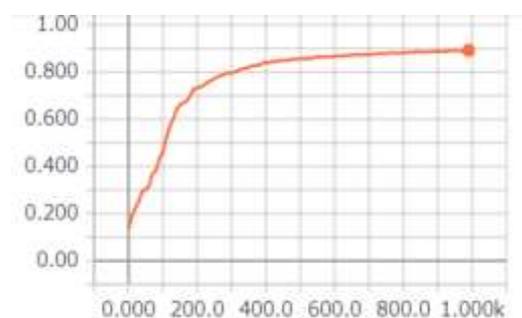


図2 縦軸:精度, 横軸:試行回数

このグラフからわかるように、試行回数が増えるにつれて損失・誤差が減っていき精度が上がっている。したがって、試行を通してプログラムが正しく学習していることがわかる。

#### 4.2 考察

本研究で用いたデータセットの中には、人間が見ても正解とされている数字を判断できない文字が含まれている。そのため、100%の精度には及ばなかったと考えられる。また、今回は隠れ層を2層で学習を行ったが、隠れ層をより多く設定することでより高い精度での認識が可能になると考えられる。

## 5.1 実験2

実験1で作成したプログラムを用いて漢字を認識するプログラムの製作を試みた。実験1では画像の収集に Mnist の画像データを用いていたが、この実験では独自性を出すために自分たちで漢字を書き、それを画像データとすることにした。しかし、自分たちで書くことのできる漢字の種類、数には限度があると考え、何かしら工夫が施された漢字を認識するプログラムを製作すると目的を変更して実験を行った。

## 5.2 方法

何かしらの工夫が施された漢字として一部分だけ欠けている漢字を認識させることにした。そこで「龍」、「繕」、「議」、「護」、「機」の5文字をそれぞれ10枚ずつ書いて画像データとした。また画像データの数が少ないことを改善するためにpythonのライブラリであるkerasを用いて、画像を回転させたり引き延ばしたりして画像データの数を50枚から5000枚に増やした。ここで画像データの数を増やしたのは、トレーニングデータのみを正確に認識してしまうこと(過学習)を防ぐためである。プログラムの内容は実験1とほとんど同じだが隠れ層を2層から10層に増やした。

## 5.3 結果

右の図3は製作したプログラムの最終的な精度を示したものである。この図から精度は20%とかなり悪い結果になってしまったことがわかる。

## 5.4 考察

このように精度が悪くなってしまった原因として2つのことが考えられる。1つは画像を回転させる際に左右に90度まで回転させてしまったことで、もう一つはシャープペンシルで書いたことで文字が薄くなってしまったことだ。これらを踏まえて、次の実験ではマーカーペンを使って文字を書き、画像を回転させる際には左右に30度程度まで回転させたい。

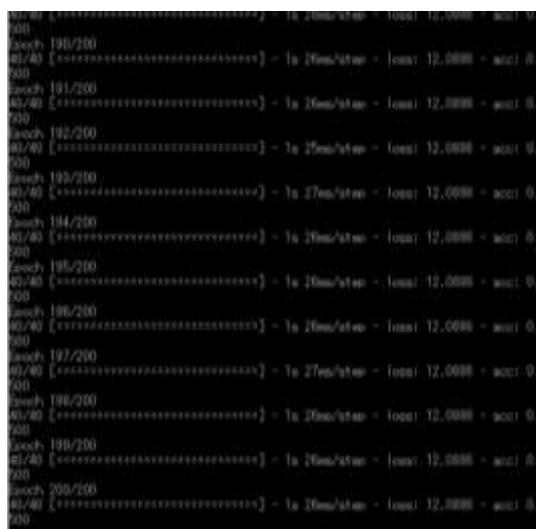


図3

## 6. 展望

今後は、本研究で製作したプログラムを基盤にひらがな、カタカナ、漢字の認識に対応したプログラムの製作を目指す。漢字の場合、数字に比べて形が複雑になるが、その分プログラムによる判断材料が増えることでもあり、また、辺やつくりなどから文字の特徴を捉えることができる。画像をパーツに分割して特徴を捉えることのできる畳み込みニューラルネットワーク(CNN)を用いれば認識できだろうと予想できる。

## 7. 参考文献

TensorFlow で始める DeepLearning

実装入門                      新村拓也 著

ゼロから作る Deep Learning

                                    斎藤康毅 著