Development a Program to Recognize Handwritten Characters

MURAYAMA Eriko SUNAMOTO Yusuke CHIKAOKA Rintaro NAGAYA Ryuji YACHI Takuya

Abstract

In our study, our goal is to understand the mechanism of image recognition AI and develop it. We made a Deep Learning program that can recognize numbers.

1. Introduction

We made a Deep Learning program that can recognize numbers. These days, AI is becoming more and more interacted in everyday life. It is used in several ways, such as, automatic driving. We were interested in AI and decided to learn how to develop AI programs. AI can do a lot of things such as defeat board campion game and communicate with human beings. We decided to research image recognition programs that we often use in face authentication. Our purpose is to make AI programs that can recognize characters. Stroke numbers and stroke order are not used in recognition. We use only the data of image. Today, some optical character readers (OCR) functions already exist. However, they can read only fonts not handwriting. Therefore, we make a program which can read handwritten characters.

2.1. Method

We learned programming languages to understand AI systems. We then used Python as the programming language, in which we used a library called TensorFlow to conduct experiments.

2.2. Structure of AI

I will describe the most basic AI structure with using an example of image recognition AI. AI is composed of three layers, and the first layer is called "Input layers". Image data is input there. The next middle layer, called "Hidden layers", performs calculations. In the calculation, it is possible to increase the accuracy of the AI by multiplying what is called a bias and changing this value. The final third layer is called "Output layers", which outputs calculation results. This answer is given by probability, and the one with the highest probability is output as the answer. From the output probability group, “Loss function” that is an index of how many mistakes the AI made is required. Based on Loss function, AI is made more accurate based on that. The processing is performed on each, and the trial is performed a certain number of times to obtain the final AI output.

2.3. About CNN

Convolutional neural network (CNN) complicates the mechanism of image recognition neural networks. The above-mentioned neural network can not recognize the line connection or character features because it is simply a list of image data. CNN clears the point and applies several filters to the image to extract the features of the image. This filter becomes a bias in CNN, and learns AI by changing the value of the filter through learning. While higher accuracy can be realized than the simple neural network, it takes more time for learning.

3. Result

This graph (No.1) shows a loss function. The Y axis of this graph represents how incorrectly this program recognized images and the X axis represents the number of trials. This graph (No.2) shows a function whose abscissa represents the accuracy and the ordinate represents the number of trials. As these graphs show, the more it tries, the more correctly it recognizes images and the higher it becomes. Therefore, it studied correctly through trials. This graph (No.3) shows the accuracy of this program. As it shows, its accuracy is only 20%.

4. Consideration & Prospect

There are some images which even human beings cannot recognize what number it is. For example, number “1” is similar to “7” so that it is difficult for AI to distinguish them. Therefore, the accuracy cannot reach 100%. In this study, we make the program with 2 hidden-layers. It is conceivable that the more hidden-layers we build, the more accurately the program can recognize numbers.

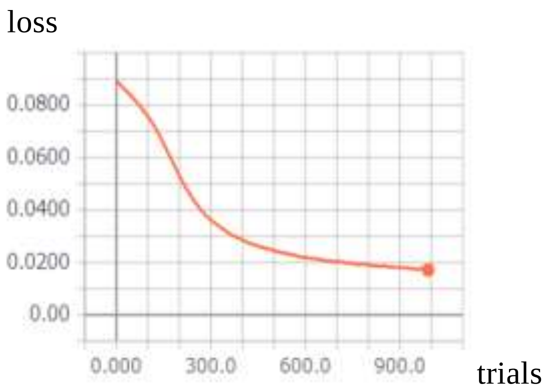
After this, we would like to develop a program that can recognize *hiragana, katakana,* and *kanji* (Chinese character). We are going to use the program with the same structure as what we made. Although Chinese characters are more complex than numbers, the program can obtain more information of the letters. Also, characters can be disassembled to some parts and the program can recognize their feature. Therefore, the program would be able to read them by using CNN.

5. References

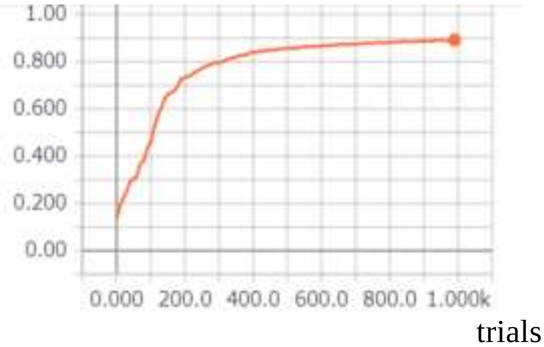
TensorFlow ではじめる Deep Learning 実装入門,新村拓也
ゼロか
ら作る Deep Learning,斎藤康毅

6. Key words

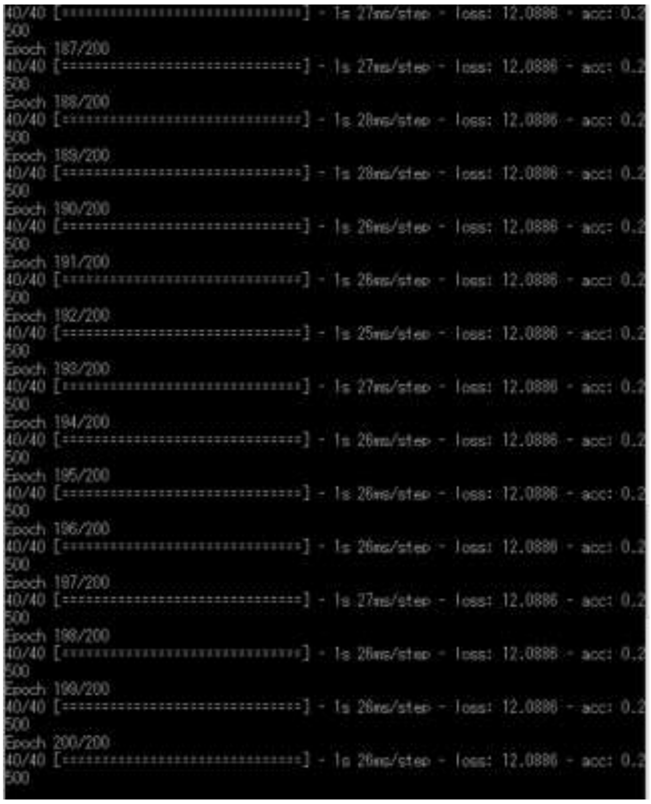
Artificial Intelligence, Deep Learning, Neural Network, Programming



No.1



No.2



```
40/40 [==============================] - 1s 27ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 187/200
40/40 [==============================] - 1s 27ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 188/200
40/40 [==============================] - 1s 28ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 189/200
40/40 [==============================] - 1s 28ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 190/200
40/40 [==============================] - 1s 26ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 191/200
40/40 [==============================] - 1s 26ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 192/200
40/40 [==============================] - 1s 25ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 193/200
40/40 [==============================] - 1s 27ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 194/200
40/40 [==============================] - 1s 26ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 195/200
40/40 [==============================] - 1s 26ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 196/200
40/40 [==============================] - 1s 26ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 197/200
40/40 [==============================] - 1s 27ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 198/200
40/40 [==============================] - 1s 26ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 199/200
40/40 [==============================] - 1s 26ms/step - loss: 12.0886 - acc: 0.2
500
Epoch 200/200
40/40 [==============================] - 1s 26ms/step - loss: 12.0886 - acc: 0.2
500
```

No.3