JAPANESE CHARACTERS (KANJI) STROKE PREDICTION USING LSTM

Aftab Dange*1

*1 Indian Institute Of Technology Jammu, Jammu, Jammu and Kashmir, India.

DOI : https://www.doi.org/10.56726/IRJMETS44512

ABSTRACT

Japanese characters (Kanjis) are composed of strokes and each stroke is drawn sequentially. Sequence of strokes is specific and crucial to decide next stroke. Stroke count can go up to 85 strokes and which makes it one of the complex characters to predict next stroke. In this paper, LSTM is used to predict next stroke based on strokes drawn till current instance. Training of LSTM model is carried out based on tokenizing each stroke drawn in PNG image using Keras Tokenizer. While applying LSTM model for stroke prediction, Arithmetic operations performed on drawn stroke and strokes image stored in database for Image correlations and comparison and hence to identify the stroke drawn. Stokes drawn is fed to the trained LSTM model. The stroke predicted by LSTM model is fetched from database. A composite video is prepared interlacing drawn stroke with model input and predicted strokes to visualize the LSTM model performance. This paper also demonstrates comparative study of LSTM, Simple RNN and GRU along with hyperparameter tuning of LSTM.

Keywords: LSTM, Naive Approach, Keras, Tokenizer, RNN, GRU, Kanji

I. INTRODUCTION

The Japanese language has three types of characters: Hiragana, Katakana, and Kanji. Hiragana and Katakana are phonetic symbols, each representing one syllable while Kanji is ideogram, each stand for certain meaning. The modern Japanese writing system uses a combination of logographic kanji, which are adopted Chinese characters, and syllabic kana (Hiragana and Katakana). Japanese kanji writing is a complex and fascinating aspect of the Japanese writing system. Stroke Order is one of the key aspects of Japanese kanji writing. Kanji characters are composed of various strokes, each with a specific order. Proper stroke order is crucial not only for aesthetic reasons but also for legibility and consistency. Figure 1 shows an example of stroke order or Japanese character read as – Utsukushi. Research [1] shows the importance of stroke order in writing Chinese character which are very similar to Japanese characters and hence methods developed to aid teachers diagnose their students’ writing and know their learning condition in order to help teachers’ teaching for writing Chinese characters.

![Figure 1: Stroke order example of Japanese Character “Utsukushi”](image)

LSTM (Long Short-Term Memory) – a type of Simple RNN (Recurrent Neural Network) can be trained to remember the sequence [3] in which the character is drawn and predict the next stroke given current drawn stroke. LSTM has proven to be highly effective in dealing with sequential data and time series problems. LSTM was designed to address the limitations of traditional RNNs, such as the vanishing gradient problem, which can make it difficult for RNNs to capture long-range dependencies in sequences. LSTM overcomes these limitations by introducing memory cells and gating mechanisms. LSTM introduces three gating mechanisms that regulate the flow of information within the network:

- Forget Gate: Determines what information to discard from the previous memory cell state.
- Input Gate: Determines what new information to store in the memory cell.
- Output Gate: Controls what information to output from the current memory cell state.

Figure 2 shows Architecture of the LSTM Cell which is used in this paper and which is mathematically discussed and compared with simple RNN along with hybrid learning approaches in research [3] [4][5]. [4] also proves that deep LSTM RNN architectures achieve state of-the-art performance for large scale acoustic modeling.
In this paper, GRU performance is also compared with LSTM. Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture that is designed to address some of the limitations of traditional RNNs in capturing long-term dependencies in sequential data. GRU is similar in concept to Long Short-Term Memory (LSTM) networks but has a simplified architecture with fewer parameters, making it computationally more efficient and easier to train in some cases. Figure 3 shows architecture of GRU. Figure 3.0 shows are as compared to LSTM, GRU has fewer gates (Reset Gate and Update Gate) which makes it computationally efficient.

Figure 2: LSTM Cell Architecture

Figure 3: GRU Cell Architecture

The research [6] has shown the evaluation of each of the classical gated architectures for language modeling for large vocabulary speech using LSTM and GRU architectures along with its comparative studies. There are several methods researched to identify Kanji strokes order like HMM (Hidden Markove model) explained in [2] which uses pen location to identify stroke order.

II. METHODOLOGY

In this paper, methodology used to train and evaluation LSTM model to predict Japanese Kanji stokes is illustrated in Figure 4.

Figure 4: Work flow and methodology

Congrous and application specific data gathering followed by selecting most suitable data pre-processing methods followed by model training phase is conducted. After training, appropriate model performance visualization techniques study, analysis and selection is performed.
III. MODELING AND ANALYSIS

1. Gather and Preprocess Data:
Kanji-Alive dataset is used to gather stroke order images of Japanese characters and names. Kanji alive is a free resource for learning to read and write kanji. With few exceptions, this repository includes all of the language data and media files created by the Kanji alive team for web application. Data preprocessing includes,

Converting SVG format data to PNG format so that it can be accessed and visualized.

After converting data to PNG format as database is maintained to map PNG images to PNG file location. This database plays a crucial part during all stages of experiment including Model training, Image Comparison and Visualization. The dataset of about 1000 different Japanese characters with about 11000 images is included in the dataset.

2. Tokenize and Create Sequence:
Using Keras Tokenizer, a sequence of characters stored in database is converted to unique tokens followed by sequences. Keras Tokenizer is a utility in the Keras deep learning library that facilitates text preprocessing and tokenization, making it easier to prepare text data for training neural networks. Tokenization involves breaking down a text into individual words or sub word units, which are then typically converted into numerical representations that can be fed into a neural network.

3. Training LSTM Model:
The LSTM model trained using sequence of tokens generated. A typical LSTM architecture is used here. Embedding followed by LSTM described in Figure 2 and Dense layer. Figure 6 shows training accuracy and loss achieved. Table 1 shows hyperparameters of LSTM model during training phase.

<table>
<thead>
<tr>
<th>#</th>
<th>Hyperparameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of LSTM Units (Neurons)</td>
<td>512</td>
</tr>
<tr>
<td>2</td>
<td>Return Sequences and Return State</td>
<td>false</td>
</tr>
<tr>
<td>3</td>
<td>Activation Function</td>
<td>Selu</td>
</tr>
<tr>
<td>4</td>
<td>Recurrent Activation</td>
<td>Sigmoid</td>
</tr>
<tr>
<td>5</td>
<td>Dropout and Recurrent Dropout</td>
<td>0.0, 0.0</td>
</tr>
<tr>
<td>6</td>
<td>Kernel Initializer</td>
<td>glorot_uniform</td>
</tr>
<tr>
<td>7</td>
<td>Recurrent Initializer</td>
<td>orthogonal</td>
</tr>
</tbody>
</table>
4. Video Frame Extraction and Feeding to Model:
The trained model is used to predict next stroke. Current stroke is a frame from a recorded video. The complete process is elaborated in Figure 7. Since, position of the stroke is crucial to decide to stroke type, peculiar image processing is required before processing for image comparison and database handling. Amongst various image comparison methods discussed during research [7] [8], arithmetic operations proven to be useful amongst other. Table 2 shows comparative analysis of Image correlation methods.

Table 2: Image Correlation Methods Comparison

<table>
<thead>
<tr>
<th>#</th>
<th>Image Correlation Method</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sentence Transformer Encoder</td>
<td>Highly Depend on Image quality. Good quality image will be closer than actual image</td>
</tr>
<tr>
<td>2</td>
<td>Mean Squared Error</td>
<td>Position of pixel has less effect on MSE value</td>
</tr>
<tr>
<td>3</td>
<td>Image Airthmatic Subtraction</td>
<td>Position of pixel has more weightage</td>
</tr>
</tbody>
</table>

5. Create Video Including Model Prediction:
Using OpenCV, video processing is performed to visualize model performance along with inputs provided to the LSTM model. Figure 8 shows model predictions in blue color.

IV. RESULTS AND DISCUSSION

1. Japanese character stroke prediction:
It is observed that LSTM trained model is able to next strokes correctly as shown in Figure 8 and 9.

2. LSTM Hyperparameter Tuning and Comparison:
LSTM Hyperparameters discussed in Table 1 are compared with various values. Selu is observed to be most appropriate activation function. (Figure 9). Increase number of LSTM units were beneficial to train model with lesser number of epochs. (Figure 10). Also, Kernel initializer type is one of the crucial hyperparameter to achieve training accuracy with lesser epochs and hence faster and consuming less resources. (Figure 11)
3. **Comparative Analysis of LSTM, GRU and Simple RNN:**

Considering architecture discussed in Figure 2 and Figure 3, Due to difference in number of gates, there is significant difference in number of training parameters. Simple RNN being least number of training parameters and LSTM the greatest number of training parameters. Figure 12 shows the comparison of number of training parameters and Hence, memory requirement differs significantly (Figure 13).

It has been also observed that even though memory consumption of GRU is comparatively less than LSTM, execution time is considerably higher.
The study and experiments done during this paper shows that the crucial part of Japanese language study like stroke order can be learned through LSTM – Deep learning models which can be applied to various applications like student assessment or kids’ education. Moreover, hyperparameter tuning study and comparative analysis of LSTM, RNN and GRU shows that appropriate hyperparameters can be selected to achieve most suitable model performance and architectures of individual plays very important role in memory consumption and training duration.

ACKNOWLEDGEMENTS

This paper has been published as a result of Executive M.Tech – Deep Learning Course study at Indian Institute of Technology, Jammu.

VI. REFERENCES


