Research on the Performance of Character Convolutional Neural Network in Different Text Encoding Formats

Zhiyao Bao
The University of Sydney
zbao0587@uni.sydney.edu.au

ABSTRACT: With the continuous progress and development of society, text classification has become an important task in the field of text data mining and text value exploration. Compared with existing text classification technology, deep learning technology has many advantages, such as high accuracy and effective feature extraction. This paper is mainly based on the character convolutional neural network to study the classification performance of different texts, hoping to provide experience for related research work. Through research, it is found that character-level convolutional neural network deep learning technology can achieve Chinese text classification more effectively.

1. INTRODUCTION
With the advent of the increasingly open internet era, people have obtained a large amount of text information and data on the internet. For example, users read and create conversations when they chat online; they will look for what they are more concerned about when browsing news; they go for categories that they are interested in among text messages; and they will make decisions on traveling, accommodation, dining and other activities based on the internet ratings. However, the explosive growth of text data makes it impossible for users to process it personally, and text processing technology is particularly important in this era. Text processing technology can identify and classify keywords and display them to users after statistical processing. It not only greatly shorten people's reading time, but also mine out and generate other valuable information from these unstructured data.

2. Overview of text classification
Text classification refers to classifying text into a specific category or multiple categories based on the title and content of the text and predefined tags. Text classification has a wide range of applications, including news topic classification, movie review sentiment analysis and spam recognition. Before the 1990s, text classification was mainly based on knowledge engineering[1]. In other words, experts formulate knowledge rules for each category based on predefined category tags, and then the text function determines the appropriate category for the text. This knowledge engineering-based classification has the following disadvantages: the text classification process is cumbersome and inefficient; the quality of classification depends on the quality of knowledge rules; text classification in different fields is very costly because different experts are required to formulate the rules; and the required manpower and material sources are huge. Therefore, it is necessary to explore better classification algorithms.
3. Research on Deep Learning Text Classification Technology Based on Convolutional Neural Network

3.1 Character-level Convolutional Neural Network Model

The character-level convolutional network designed in this paper treats texts as character-level raw unprocessed signals and applies deep convolution to it. Character-level convolutional neural networks have many advantages. As the input is character-level features, the convolutional neural network does not require knowledge of words and knowledge of language grammar or semantic structure; the model only runs on the character-level, which makes it easy to learn uncommon character combinations such as emoji and spelling errors [2]. Character-level convolutional neural network structure is shown in Figure 1. (The numbers "3, 128" in the convolutional layer represent the size of the convolution kernel and the number of convolution kernels, the "100" in the fully connected layer represents the output dimension, and the "5" in the output layer represents the number of categories.)

![Figure 1: The Structure of Character-level Convolutional Neural Network](image)

3.1.1 Embedded Layer

The input of the embedded layer is a matrix composed of L×S One-Hot encoding vectors. Since the One-Hot encoding vector is a discrete and sparse representation, and for the text represented by the original Chinese characters, the dimension of the One-Hot encoding vector will reach 6757 dimensions. Having too high dimensionality will cause the calculation cost to skyrocket and affect the classification performance of the model. Therefore, whether it is the original Chinese character or the text expressed in pinyin format, the embedded layer maps each One-Hot encoding vector onto the d-dimensional continuous vector space \( \mathbb{R}^d \). When One-Hot encoding vector is multiplied by the weighing matrix, where \( W \in \mathbb{R}^{s \times d} \), and the initialization of \( W \) follows Gaussian distribution.

Expressing the matrix \( L \times S \) as \( Z = (z_1, z_2, \cdots, z_L)^T \), where \( z_1, z_2, \cdots, z_L \) are the One-Hot encoding vectors of the characters accordingly, the transformation of original matrix \( Z \) is \( \tilde{Z} \in \mathbb{R}^{L \times d} \).

3.1.2 Convolutional Layer

Since text is a character-level sequence, it is not recommended to treat the feature matrix as a two-dimensional matrix in image processing for sequence problems. On the contrary, the element matrix complex product layer should be regarded as a one-dimensional object (where length is 1 unit, height is \( L \) units). Therefore, three temporal convolution modules are constructed in the convolution layer, that is, only one-dimensional convolution was applied.

The principle of time convolution is to assume that with a discrete function \( g(x) \in [1, L] \rightarrow \mathbb{R} \) as input, where \( L \) is the sequence length, and the convolution kernel is represented by \( f(x) \in [1, k] \rightarrow \mathbb{R} \), where \( k \) is the size of the convolution kernel; the convolution result of \( f(x) \) and \( g(x) \) is \( h(y) \in \{1, \lfloor(L-k+1)/\Delta\}\} \rightarrow \mathbb{R} \). This paper selects 128 convolution kernels with the size of 3. The calculation process is as follow:
where $\Delta$ is the sliding step length of the convolution kernel, and $c=k-\Delta+1$ is the offset constant.

For a convolutional network where the input and output features are $u$ and $v$ respectively, each input and output feature should be expressed as $g_i(x)$ and $h_j(y)$, and the weight of the kernel function as $f_{ij}(x)$, where $i=1,2,...,u; j=1,2,...,v$. The output feature $h_j(y)$ is the sum of the convolution between $g_i(x)$ and $f_j(x)$ at $i$.

The calculation process can be imagined as the convolution kernel moving from the beginning to the end in a one-dimensional object to extract high-level features of the text. The weight initialization of the convolution kernel follows the Gaussian distribution, and in order to keep the length of the output vector of the convolution layer at $L$, the boundary of the one-dimensional object input must be filled with zeros. Based on the discovery of Conneau [3], this paper selects 128 convolution kernels with the size of 3. This means that each layer can automatically combine these ternary character features. In this paper, ReLU, which is widely used in recent researches, is selected as the nonlinear activation function of the convolutional layer. Unlike the "Sigmoid" function, the ReLu activation function can better handle the gradient loss problem, and the threshold can better simulate the functions of the human brain. To avoid overfitting, L2 regularization is used for these convolutional layers. The final output of the convolutional layer is the tensor $L\times128$, which is the hierarchical expression of the text function. The temporal convolutional network can automatically extract three-element character features from the filled text, so these features can represent the long-term and short-term relationships of the text.

### 3.1.3 Pooling Layer

It is necessary to place the pooling layer behind the convolution template as the pooling layer can select the most important features from the feature map output from the convolutional layer to reduce the number of parameters and speed up the training. Each feature mapping vector can be expressed as $E \in \mathbb{R}^{1000 \times 1}$. This study chooses the 1-max pooling method to make the model only focus on the most important feature $E \in \mathbb{R}^{1000 \times 1}$, and $E$ can be calculated as

$$E = \max(v)$$

The 128 feature maps output from the convolutional layer will produce 128 feature values, namely $v_1, v_2, ..., v_{128}$ respectively.

### 3.1.4 Fully-Connected Layers

The 128 feature values obtained from the pooling operation are the most influential local features in text classification. The task of full-connection is to concatenate these 128 attribute values to form a 128-dimensional fusion attributing vectors representing the original text. The final output layer has 5 target categories in the data set, so this layer has 5 neurons, and we use "softmax" as a nonlinear activation function. The currently used methods, such as Conneau, do not use the dropout strategy in the fully-connected layers, but for the deep modeling structure, the batch normalization method is a better choice to speed up the learning speed. The model built in this article is not very in-depth, so the dropout strategy is still used to prevent the model from overfitting.

### 3.2 Experimental Design and Analysis

Comparing the model performance with datasets in different encoding formats.

#### 3.2.1 Experimental Design

#### 3.2.1.1 Task Description

The task is mainly to verify the model of the Chinese character dataset and the corresponding Pinyin
format dataset in this article. By comparing the performance of the model on datasets with different encoding format, we can prove that the character-level convolutional neural network performs better on the Chinese character datasets.

3.2.1.2. Datasets Description
There are very few large Chinese datasets in the field of Chinese text classification. Therefore, in order to carry out the research in this experiment, it is necessary to generate a Chinese dataset, and then use the Python library to convert the original Chinese character data into a pinyin dataset encoded in format I and format II respectively. In this experiment, we used Sogou Lab's entire network news data to construct a new Chinese dataset, and used the domain part of the URL link to mark the category of each text. Since there is not enough data for each category in the original news dataset, we had only selected the top five categories with the highest amount of data in this experiment: sports, finance, entertainment, automotive, and technology. The data range for each category is between 50,000 and 200,000. When filtering data, texts with less than 20 Chinese characters were omitted. After data preprocessing is completed, 80% of the data is used for training and the remaining 20% is used for testing.

Table 1: Data sets in different encoding formats

| Data Set          | Categories | Training Mode | Test Mode | Overall Scale |
|-------------------|------------|---------------|-----------|---------------|
| Chinese Characters| 5          | 460000        | 115000    | 575000        |
| Pinyin Format I   | 5          | 460000        | 115000    | 575000        |
| Pinyin Format II  | 5          | 460000        | 115000    | 575000        |
| Data Enhanced     | 5          | 920000        | 230000    | 1150000       |

Table 1 shows the data volume of the datasets using different encoding formats. Using data enhancement methods in deep learning models can improve their performance. Data enhancement technology is widely used in computer visual and speech recognition mainly by converting signals and rotating pictures to scale datasets. In this experiment, the datasets encoded with pinyin format I and II are combined to expand the size of the pinyin format datasets through the data enhancement method, and the data enhanced pinyin format datasets are obtained and reaches 1.15 million in volume.

3.2.1.3. Model Settings
A dictionary is created based on the Chinese characters appearing in the training set and the test set of the Chinese character dataset, with the addition of some commonly used punctuation marks and spacing characters. The size of the dictionary is 6757. In addition, several alphabets are needed to express the pinyin when converting Chinese characters into pinyin format. In other words, in order to represent a single Chinese character, several alphabets in pinyin form need to be used. According to the statistics of the Chinese character data set, most texts do not exceed 250 Chinese characters, so the L value is set to 250 in the data preprocessing step.

For the three datasets in pinyin format, the dictionary size of the data set encoded in pinyin format I is 54, the dictionary size of the data set encoded in pinyin format II is 35, and the dictionary size of only the enhanced dataset in pinyin format is 59 . The five letters "v", "l", "2", "3" and "4" are added to the dictionary of the format I data set. In these three pinyin format datasets, the L value is set to 1000.
3.3 Experimental Environment
The experiment in this article is carried out on a desktop computer with an Intel Core i9 processor. All codes are implemented in python language under the Ubuntu system environment with keras as the deep learning framework.

3.4 Experimental Results
This model conducts comparative experiments on the Chinese character data set and the three derived Pinyin encoded format data sets to explore the Chinese character format suitable for Chinese character classification. Table 2 shows the experimental results of each dataset model.

Table 2: Performance comparison of the model on datasets in different encoding formats

| Model                             | Error Rate | Network Layout | Dataset Encoding Format |
|-----------------------------------|------------|----------------|------------------------|
|                                   |            | No. of          |                        |
|                                   |            | Convolutional   |                        |
|                                   |            | No. of Fully    |                        |
|                                   |            | Connected       |                        |
|                                   |            | Layers          |                        |
| Character-level Convolutional     | 5.53%      | 3              | Chinese Character      |
| Neural Network                    |            | 4              |                        |
|                                   | 6.21%      | 5              | Chinese Character      |
|                                   | 6.93%      | 4              | Chinese Character      |
|                                   | 11.12%     | 3              | Pinyin Format I        |
|                                   | 8.63%      | 4              | Pinyin Format I        |
|                                   | 11.35%     | 5              | Pinyin Format II       |
|                                   | 8.75%      | 4              | Pinyin Format II       |
|                                   | 9.68%      | 3              | Enhanced Data          |
|                                   | 7.37%      | 5              | Enhanced Data          |

From the experimental results it can be seen that the character-level convolutional neural network has achieved satisfactory results on both the Chinese character data set and the Pinyin format data set. In the Pinyin format I data set, the Pinyin format II data set and the enhanced data set, the lowest error rate of the model is 8.63%, 8.75% and 7.37%, respectively, while the error rate of the Chinese character data set is only 5.53%. These results show that the character-level convolutional neural network can effectively extract Chinese text features. This model is most suitable for Chinese data sets and shows that the character-level convolutional neural network can solve the Chinese text segmentation problem. In the case of character-level convolutional neural networks, text is processed based on character sets rather than on word-level. In other words, the convolutional layer is characterized by character combinations and ignores the segmentation between words. This is also the reason why the performance of the model on the Chinese character dataset is better than that of the Pinyin format dataset, because the conversion of Chinese characters to Pinyin results in information compression.

4. Conclusions
The experimental results on the Chinese character model and the corresponding Pinyin format data set show that the character-level convolutional neural network performs better on the Chinese character data set. The main reason is that the character convolutional neural network successfully solved the problem of word segmentation and information storage of Chinese corpus. In addition, through the above experiments and related theoretical knowledge, we also understand that proper character convolutional neural network hyperparameters play an important role in the task of classifying text encoded in Pinyin format. Therefore, we must continue to develop the text classification technology of
the character convolutional neural network so that people can carry out text data mining and text value exploration more effectively.

References
[1] Xin Miao Meng. Research on Text Classification Based on Big Data Mining Technology [J]. Modern Electronics Technique, 2020, 43(17):126-129.
[2] Xiao Ming Wang. Research on Text Classification of Chinese News Based on Deep Learning [D]. University of Electronic Science and Technology of China, 2020.
[3] Conneau A, Schwenk H, Barrault L, et al. Very deep convolutional networks for natural language processing [J]. arXiv Preprint arXiv:1606.01781, 2016, 2