Classification of book category using book titles

Yunkai Wang
Carleton University, School of Computer Science

Jules Kuehn
Carleton University, School of Computer Science

This is an example of trying to use convolutional neural network to classify book categories based on their titles.

I. Introduction

This is an example of trying to use convolutional neural network to classify book categories based on their titles.

II. The Dataset

The *Data Mining* is a dataset which can be found on *Github.com*. The set consists of detailed information of 207572 books from 32 categories, to help determining the potential relationship between different information related to a book, like the relationship between a book's cover image and its category. All books in the dataset have informations like the image url of the cover image of the book, the book's author, and the book's category. A sample book information in the database looks like:
"1588345297","1588345297.jpg","http://ecx.images-amazon.com/images/I/51l6XIoa3rL.jpg","With Schwarzkopf: Life Lessons of The Bear","Gus Lee","1","Biographies & Memoirs"
The dataset doesn't have the training dataset and testing set splitted by default, so we just use make a use of the *sklearn* library, which provides a nice *train_test_split* function to create the training set and test set.

Vectorize the data

All the titles are made up with words, which is hard to be feed to the neural network as input, to fix this problem, we make the following changes to the titles:
1. Create a vocabulary set containing all possible words that have appeared in one of title. The titles in the dataset used a total 71056 words.
2. We extend the titles so that all titles all have the same length. The longest title in the dataset has 96 words, but most of the titles are short, we extend those short titles by appending a special character to the end of those titles so that they all have the same length, these special characters all corresponds to 0 when we do the next step.
3. We convert the titles into array of integers, where each integer corresponds to the index of the word in the vocabulary list that we created, the index is between 1 and 71057, and we map all the special character that we added in step 2 to 0.
This vectorization process takes $\approx$ 2 hrs to run, therefore, we decide to store the vectorized data into a new .csv file, which we can use to train our data directly.