Chinese Handwriting Writer Identification Using Neural Networks

Hao Liu Advisor: Prof. Lawrence Carin
Department of Electrical and Computer Engineering, Duke University

hl259@duke.edu

Introduction

I applied latest recurrent neural network model to do the task of Chinese handwriting character writer identification. I implemented the softmax, MLP, LSTM, bidirection LSTM, Hierarchical RNN while using empirical techniques such as dropout and masking. I think using Hierarchical RNN is a new way for this task and it can improve the performance. It is still under experiment.

I have finished several steps in this project:
(*i*) Implement c++ and python program to process CASIA online chinese handwritting data
(*ii*) implement several latest deep neural network models for this task
(*iii*) propose a new way to use stroke information in this task by applying Hierarchical Recurrent Neural Network

Data and Task Description

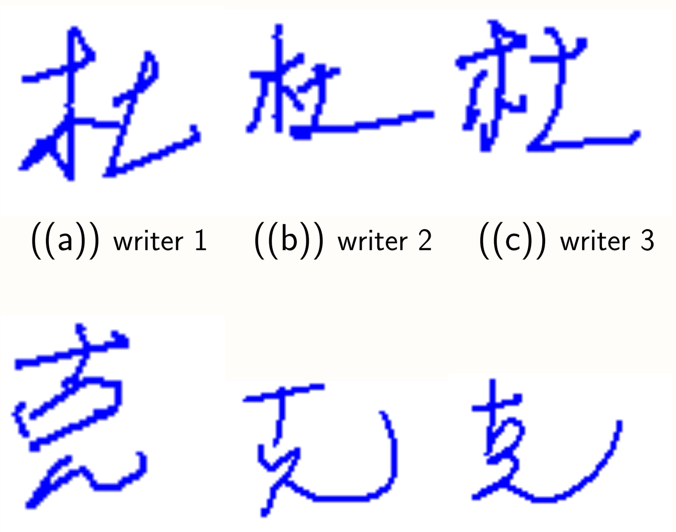
Handwriting Database

For online handwriting data collection using Anoto pen, all the template pages were printed on papers with dot pattern. During writing, the online data (stroke trajectory: sequences of (x,y) coordinates) were recorded by the Anoto pen and later transmitted to computers. This data can be tackled with recurrent neural network since it is in the form of sequence.

For offline data collection, the handwritten characters are stored in the form of images and it can be tackled with convolutional neural network.

Experiment Data

I use CASIA online Chinese handwriting Database. There are 45315 different Chinese handwriting characters from five writers in total. I split it into 40280 training data and 5035 test data. The image below shows some example from the dataset.



((a)) writer 1 ((b)) writer 2 ((c)) writer 3

((d)) writer 1 ((e)) writer 2 ((f)) writer 3

Figure 1: Chinese character of 'Duke' from the data

Task

Given the handwriting character, the model need to figure out who is the writer after training.

Model

Softmax

The Softmax works by converting a raw vector into class probabilities. It did it by exponentiating and dividing by a normalization constant.

Fully Connected Feedforward Neural Network

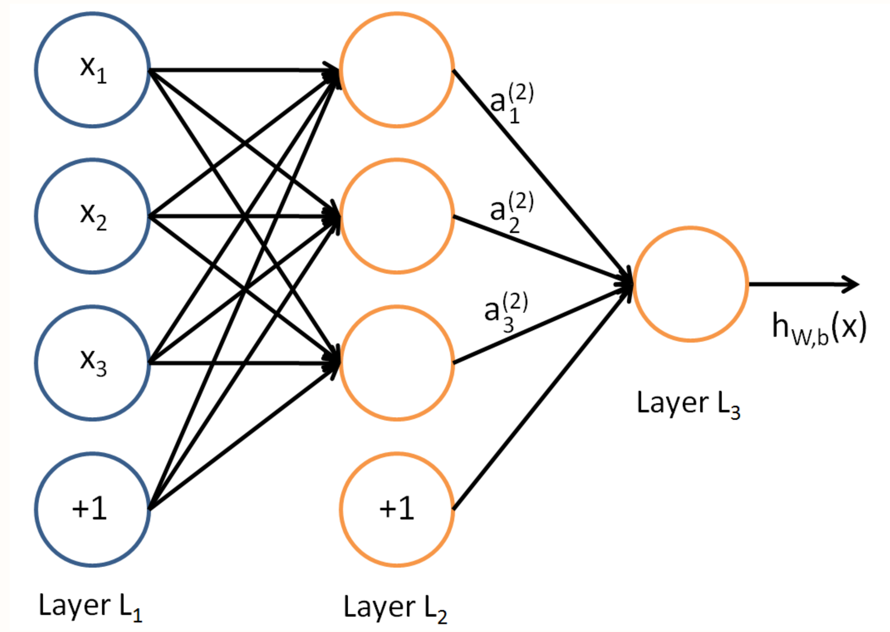


Figure 2: fully connected nerual network

Feedforward neural network, which is also called Multilayer Perceptions(MLP), is the first and simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes and to the output nodes. There are no cycles or loops in the network.

LSTM

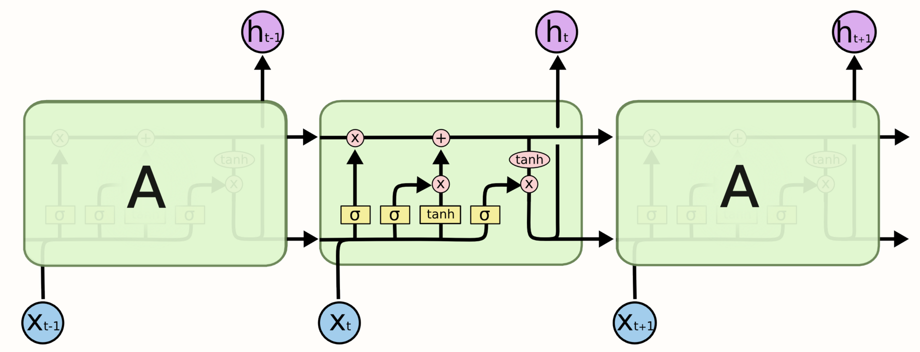


Figure 3: Long-Short term memory

Each box in the upper figure is a LSTM, which can be written as

$$
\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}^i\mathbf{x}_t + \mathbf{U}^i\mathbf{h}_{t-1} + \mathbf{b}_i\right)\\
\mathbf{f}_t &= \sigma\left(\mathbf{W}^f\mathbf{x}_t + \mathbf{U}^f\mathbf{h}_{t-1} + \mathbf{b}_f\right)\\
\mathbf{o}_t &= \sigma\left(\mathbf{W}^o\mathbf{x}_t + \mathbf{U}^o\mathbf{h}_{t-1} + \mathbf{b}_o\right)\\
\hat{\mathbf{c}}_t &= \tanh\left(\mathbf{W}^c\mathbf{x}_t + \mathbf{U}^c\mathbf{h}_{t-1} + \mathbf{b}_c\right)\\
\mathbf{c}_t &= \mathbf{i}_t \odot \hat{\mathbf{c}}_t + \mathbf{f}_t \circ \mathbf{c}_{t-1}\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}
$$

Here $\mathbf{x}_t$ is word at step t; $\mathbf{h}_t$ is the hidden state in step t, which represents the short-term memory; $\mathbf{c}_t$ is the cell state at step t, which represents the long-term memory. LSTM is a special kind of RNN, capable of learning long-term dependencies.

Dropout

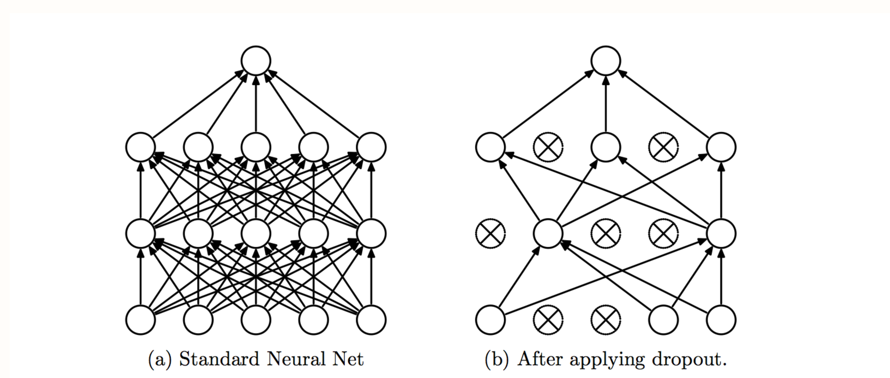


Figure 4: Dropout

Dropout is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. It is a very efficient way of performing model averaging with neural networks. The term "dropout" refers to dropping out units (both hidden and visible) during every training time.

Bidirectional LSTM

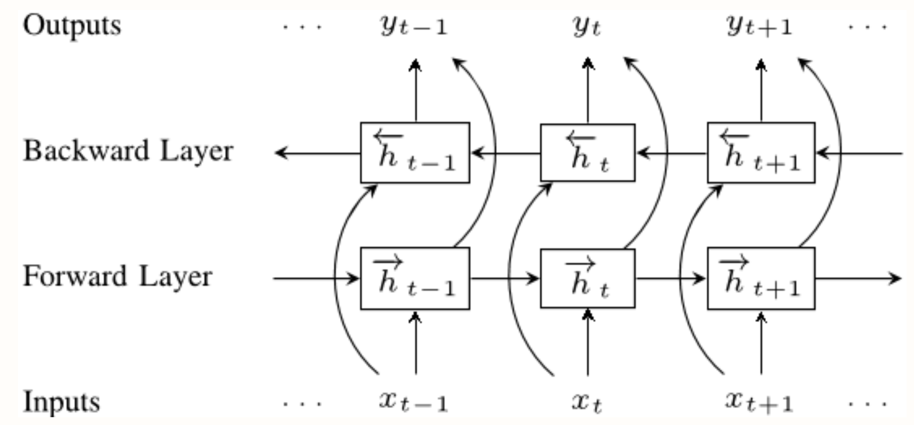


Figure 5: Bidirectional LSTM

A Bidirectional LSTM is a type of Neural Network that contains two LSTMs going into different directions. The forward LSTM reads the input sequence from start to end, while the backward LSTM reads it from end to start. The two LSTMs are stacked on top of each others and their states are typically combined by appending the two vectors. Bidirectional LSTMs are often used where we want to take the information from both before and after into account before making a prediction.

Hierarchical Recurrent Neural Networks

HRNNs can learn across multiple levels of temporal hierarchy over a complex sequence.In this situation, the first recurrent layer of an HRNN encodes a stroke (e.g. of (x,y) coordinate vectors)into a stroke vector. The second recurrent layer then encodes a sequence of such vectors (encoded by the first layer) into a character vector. This character vector is considered to preserve both the stroke-level and character-level structure of the context. Currently, I'm still tuning parameter and doing experiments on this model.

Training

These model are all trained via stochastic gradient descent. I use Adam as the optimizer and 'categorical cross-entropy' as the loss function.

Experimental Results

Figure 6 shows the average performance from 10 experiments of these five different models. The best result is achieved by Bidirectional LSTM. I'm still tuning parameters to get the best performance of Hierarchical RNN, so the result is not shown.

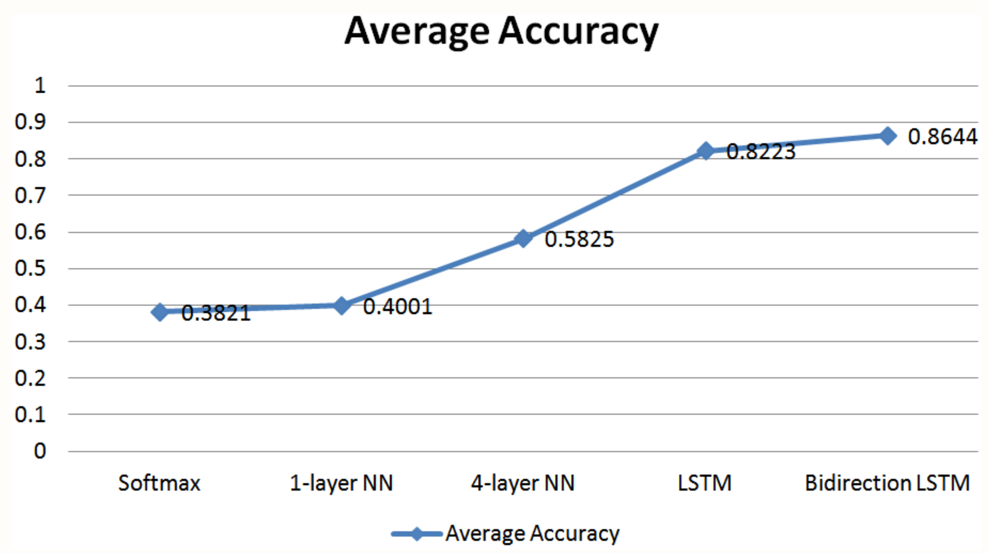


Figure 6: result of the experiment

Conclusion and future work

I show that deep learning model, more specifically, recurrent neural network, outperforms traditional machine learning techniques in the task of modeling Chinese handwriting data. And Hierarchical RNN has the potential to have SOTA performance on this task. And this framework can be directly applied to other related tasks such as Chinese character recognition. Besides, there can be some future work on applying attention and memory mechanism to achieve better performance. Moreover, tasks of how to generate human-readable handwriting, such as text synthesis, can be done under this framework.