Handwritten Text Recognition

Dushyant Chetiwal, Pallav Mathur, Tejas A. Mayekar, Ajay Meena

Indian Institute of Technology Goa

Objectives

Develop neural network based model to read handwritten text from an image

- Train the model over IAM dataset
- Find the accuracy over a test set
- Check its result for a new input image

Introduction

Handwritten Text Recognition (HTR) systems are used to transcribe text contained in scanned images into digital text, an example is shown in Fig. 1. We will build a Neural Network (NN) which is trained on word-images from the IAM dataset.

Figure 1:Image of word (taken from IAM) and its transcription into digital text.

- Assumes text is color on white background
- Assumes that the image contains only the text
- Assumes that the text only contains characters as specified in the "charList.txt"
- Assumes that the input text is at most 32 characters long

The approach

Our model consists of convolutional NN (CNN) layers, recurrent NN (RNN) layers and a final Connectionist Temporal Classification (CTC) layer.

- **CNN**: each CNN layer consists of three operation. The convolution operation, which applies a filter kernel to the input followed by the RELU function and then a pooling layer outputs a downsized version of the input.
- **RNN**: the popular LSTM implementation of RNNs is used, as it is able to propagate information through longer distances and provides more robust training-characteristics than vanilla RNN.
- **CTC**: while training the NN, the CTC is given the RNN output matrix and the ground truth text and it computes the loss value. While inferring, the CTC is only given the matrix and it decodes it into the final text.

Contributions

- Developed a model that (a) can capture properties of characters in pixels (b) use them to predict the characters from any new image
- Performed evaluation using test dataset.

Graphical work flow

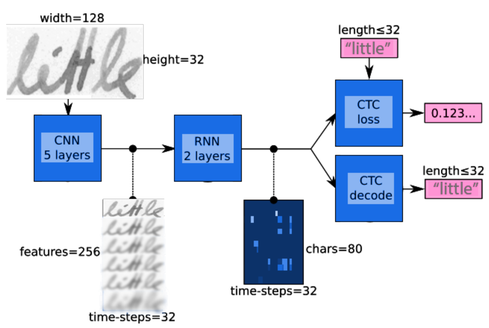

Figure 2:An overview of the model.

Inference

The accuracy obtained on the test data set is around 89.38% for characters.
This method for HTR is good enough for images that are not too far from the format of our IAM dataset.
We can use the model to scan simple pages of text with proper line segmentation and word segmentation algorithms.
The deslanting algorithm can be applied to increase the accuracy for cursive texts.
If the text only contains words that are in the dictionary then implementing word beam search on the CTC layer yields greater accuracy

Experiments and analysis

After each epoch of training, validation is done on a validation set (the dataset is split into 95% of the samples used for training and 5% for validation.
Character error rate of model: 10.62%
Word accuracy: 67.721739%

```
> python main.py --train
Init with new values
Epoch: 1
Train NN
Batch: 1 / 500 Loss: 130.354
Batch: 2 / 500 Loss: 66.6619
Batch: 3 / 500 Loss: 36.0154
Batch: 4 / 500 Loss: 24.5898
Batch: 5 / 500 Loss: 20.1845
Batch: 6 / 500 Loss: 19.2857
Batch: 7 / 500 Loss: 18.3493
...

Validate NN
Batch: 1 / 115
Ground truth -> Recognized
[OK] "," -> ","
[ERR:1] "Di" -> "D"
[OK] "," -> ","
[OK] """ -> """
[OK] "he" -> "he"
[OK] "told" -> "told"
[ERR:2] "her" -> "nor"
```

Figure 3:A snippet from the console which shows training and validation.

References

- http://www.fki.inf.unibe.ch/databases/iam-handwriting-database
- https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5
- https://medium.com/@arthurflor23/text-segmentation-b32503ef2613

Acknowledgements

A term project completed under the requirements of course CS 386: Artificial Intelligence (Instructor: Clint P. George)