Using Neural Speech Recognition Models For Optical Character Recognition of Cursive Script

By Jon Braatz

Problem Definition

Input: Text lines pictures of Arabic text

Output: A transcription of the text line.

Optical Character Recognition (OCR) systems are typically designed with Latin character-based languages in mind. Existing system often struggle with cursive input (for example, Arabic script) due to inability to segment into characters and context dependence of appearances.

Automatic Speech Recognition systems like Baidu's DeepSpeech 2 excel at transcribing voice, which is more context-dependent than cursive is. We adapt this model to work on images instead of audio.

Dataset

Arabic Printed Text Database (APTI)

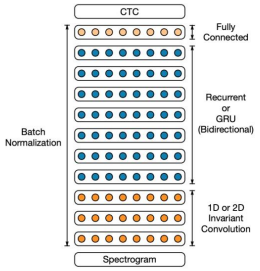
- Generated synthetically generated using a lexicon of 113,284 words, 10 Arabic fonts, 10 font sizes, and 4 fonts styles.
- Images also feature ligatures and flourishes that are common in Arabic text and have no parallel in Latin-script based

نقدم في هذا البحث قاعدة بيانات لكلمات عربية

نقدّم في هذا البحث قاعدة بيانات لكلمات عربية

Methods

- 5 layers of 2D convolutionst (used on spectrogram in original DeepSpeech application)
- Use bidirectional RNN with GRU units on CNN outputs
- Use sequential BatchNorm to speed up training
- 2 Fully connected layers at the end with softmax layer, one output per character.
- Connectionist Temporal Classification loss for segmentation-free learning

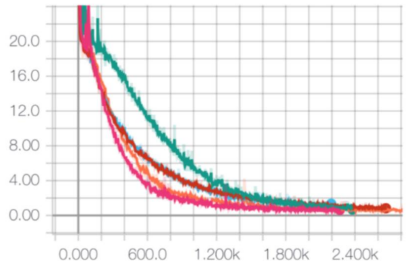


Conclusions

We were able to attain character errors rates of < 4% using a model that was built for an entirely separate purpose.

Found that number of hidden units mattered more than number of RNN layers. BatchNorm sped up training drastically. Every configuration was able to get <5% character error rate.

Results



Experiment	Character Error Rate (%)	Word Accuracy (%)
1 Layer RNN, 50 hidden units, w/ BatchNorm	4.50	78.90
4 Layer RNN, 50 hidden units, w/ BatchNorm	5.50	76.08
2 Layer RNN, 200 hidden units, w/ BatchNorm	3.36	82.64
2 Layer RNN, 200 hidden units, no BatchNorm	4.43	79.29
8 Layer RNN, 512 hidden units, w/ BatchNorm, doubled kernel sizes	3.72	80.09

Future Work

- Common errors were missing or incorrect ligatures
- Small marks should be emphasized. Max pooling layers in CNN could help with this
- Integrate into existing OCR pipeline that does classical preprocessing

تقطر <- فطر
يفتن <- يفنن
أدوا <- أدوا
فيخشى <- فيخشي
مستقتلتين <- مستقتلتين
الشأن <- الشأان