Pedagogy Project - Introduction to CTC-based Automatic Speech Recognition

Peter Sullivan
CPSC 503

April 24th 2020

Motivation

I was initially drawn to ASR as a topic due to personal interest, and the personal belief that there are major research opportunities available in the field as well as potentially interesting careers in industry that build upon these skills. My initial choice of project, a state detection project, ended up being delayed by issues with transcription and processing. This inspired me to look at other ways that speech could be processed without having to manually annotate datasets. End-to-End Speech Recognition, seems to address similar problems, having grown out of a demand to replace human-tuned models with deep neural networks (Graves and Jaitly 2014). For the scope of the project, I thus hope to build my own understanding of the topic so that I could be comfortable to get started in speech research on my own, and what better way than by teaching the topic to others (or at least creating a plan to teach it).

Similarly, ASR as a topic dovetails nicely with many of the other topics in 503. Many traditional NLP tasks, such as sentiment analysis or summarization, can be done in speech as a domain. ASR also fits in an interesting niche, where many traditional models remain fairly competitive even despite fast growth by deep neural network approaches (Kamath, Liu, & Whitaker, 2019, pg 571), echoing similar themes to 503 as a whole. Thus the topic choice acts as a complement to material already covered over the course of the term. In choosing a scope of the topic, I decided to focus on Connectionist Temporal Classification (CTC), not because it is particularly state of the art, but rather because we have already covered attention based methods in class, and thus it complements other methods by filling a gap in understanding of the recent history of Neural approaches to ASR.

Overview of Topic

CTC ASR represents one of the first forays into purely neural approaches to ASR. The motivation for this movement to neural approaches was built on the idea that larger amounts of data, and less human-tailoring (either in model or dataset annotation), would lead to better performance (Graves and Jaitly 2014). While ultimately, the push for End-to-End solutions seems to have been successful, with inclusion in production speech recognition at Baidu and Google (Jaitly 2017), there are challenges to getting solid performance with off-the-shelf models, and early models such as Deep Speech 2 perform poorly compared to traditional approaches without the aid of a language model and decoding (Kamath, Liu, & Whitaker, 2019, pg 571). That said the major advantage of not having to laboriously annotate data is one that cannot be understated.

CTC as an algorithm aims to deal with this alignment automatically. By generating tokens at every timestep, you can then remove overlapping letters and "blank" symbols to arrive at generated text that closely mirrors the target text in alignment (Jaitly 2017). The problem arises that simply picking the most probable token each timestep often leads to poor results, and thus attempts to improve decoding, such as Beam Search, are essential (Graves and Jaitly 2014).

Resources Consulted

A mixture of resources was consulted during the course of the project. While few books seem to have been published which give an overview of both E2E ASR and traditional methods, I found *Deep Learning for NLP and Speech Recognition* (Kamath, Liu, and Whitaker, 2019) to be

a reasonable at assembling context for the topic, however, at times being sloppy with specifics. Blogs proved to be somewhat decent, with Johnathan Hui's Speech Recognition Series being a reasonable overview of the topics, covering both phonetic topics, traditional ASR, as well as E2E ASR. I found several good lectures covering E2E ASR as well, with talks by Jaitly (2017) and Coates and Rao (2016) both being particularly effective. When it came time to actually implement Beam search, I found that the most effective resource was the original Graves and Jaitly (2014) paper, with the description of it in *Deep Learning for NLP and Speech Recognition* to be wholly inadequate (lacking any explanation of variables in the algorithm, nor including a precise explanation of the algorithm).

Learning Outcomes

The learning outcomes I selected aim to scaffold student understanding of the topic, moving from simple comprehension questions, to checks of understanding the coding aspect of PyTorch with respect to implementing neural networks, and to a final task requiring decent algorithmic understanding by implementing one of the decoding algorithms from scratch. The ultimate goal will be to build both comfortability in the coding of Neural Networks, but algorithmic and theoretic literacy through the final task.

Students will be able to:	These will be assessed through:
1. Explain how components of the E2E model function 2. Correctly write PyTorch code to implement common features of the E2E model	1. Q1 Code Commenting 2. Q2 'Flesh-out' E2E model 3. Q3 Beam search implementation

3. Implement decoding algorithms for E2E models using PyTorch	

Lesson Overview

The lecture aims to cover the basic ground of CTC based E2E ASR. Because the assignment focuses on decoding, emphasis has been placed on understanding of the model (both CTC as an algorithm in general as well as the Neural Network structure) and how decoding can improve it. I start with background information about ASR in general, sound as input into neural networks, and briefly cover traditional methods (HMM-GMM). With the background established I go on to cover a brief overview of a standard E2E model, Connectionist Temporal Classification, and decoding approaches to E2E models. Finally, I discuss areas for improvement on the model, with the context that if this was offered as a lecture in a course (like 503) these would be potential topics for course projects, for instance experimenting with wav2vec (Schneider, Baevski, Collobert, & Auli, 2019), attention based models such as Listen, Attend and Spell (Chan, Jaitly, Le, & Vinyals,2016) or applying Transformers to E2E speech (Zhou, Dong, Xu, S., & Xu, B. 2018).

Assignment Overview

As mentioned earlier, the assignment has been designed to align with learning objectives through a backwards mapping planning approach. Basic understanding of the code will be developed by the first commenting exercise, this will then be built upon as students extend aspects of the model through the second exercise, and finally student theory and implementation practices will be tested through implementing a decoding algorithm and a basic language model to accompany it.

The assignment itself will take the form of a Jupyter Notebook, to allow students to easily play and tinker with aspects of the code. Being able to quickly print out dimensions of tensors, or similar diagnostics, is quite handy in this format.

Challenges

A major challenge for this project was both in finding accessible material as well as computing power to run and debug the models.

Many of the lectures, books, and blogs presented extremely high level overviews of the topic, without spending much time on the details of actual implementation. This posed issues with actually getting a start on the project, with respect to minor implementation issues that seemed to be hand-waved away in these overview materials. Conversely, looking at existing open source projects proved frustrating for someone new to ASR, due to most of them aiming at being production models rather than pedagogic models.

In terms of computing issues, many of the useful speech recognition tools seem to have been only maintained for Linux-based operating systems. Kaldi the powerful ASR toolkit, for instance prioritizes Unix-like systems (“The Build Process (how Kaldi is Compiled)” n.d.).

Similarly, the models themselves take a tremendous amount of memory to run, making them difficult to debug without GPU support.

References

Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016, March). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4960-4964). IEEE.

Coates, A. Rao, V. (2016). Speech Recognition and Deep Learning. Retrieved from: https://cs.stanford.edu/~acoates/ba_dls_speech2016.pdf

Graves, A., & Jaitly, N. (2014, January). Towards end-to-end speech recognition with recurrent neural networks. In International conference on machine learning (pp. 1764-1772).

Hui, J. (2019, December 26). Speech Recognition Series. Retrieved from https://medium.com/@jonathan_hui/speech-recognition-series-71fd6784551a

Kamath, U., Liu, J., & Whitaker, J. (2019). Deep learning for nlp and speech recognition. Springer International Publishing.

Jaitly, N.. (2017). Natural Language Processing with Deep Learning -Lecture 12: End-to-End Models for Speech Processing Retrieved from https://www.youtube.com/watch?v=3MjIkWxXigM

Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. arXiv preprint arXiv:1904.05862.

Zhou, S., Dong, L., Xu, S., & Xu, B. (2018). Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese. arXiv preprint arXiv:1804.10752.