Improving Automatic Speech Recognition

Mask-CTC

Vaibhav Raj & P.Balasubramanian

May 2022

1 Introduction

Automatic Speech Recognition, or ASR for short, is the technology that allows us to use our voices to speak with a computer interface in a way that resembles normal human conversation. The most advanced version of currently developed ASR technologies goes hand-in-hand with another domain of Machine Learning called Natural Language Processing (NLP). These variants of ASR come closest to allowing real conversation between people and machine intelligence and though there is still a long way to go before the apex of development, we're already seeing some remarkable results.

In this article, we discuss one such End-to-End ASR model based on the Transformer architecture (a state-of-the-art NLP model) called Mask-CTC. Researchers from the Center for Language and Speech Processing at Johns Hopkins University, Baltimore and the Department of Communications and Computer Engineering of Waseda University, Tokyo proposed this technique to reduce inference time in speech recognition and increase accuracy of current models and systems.

2 End-to-End ASR Systems

For most of the past fifteen years, ASR has been powered by classical Machine Learning methods like Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs). Though once the industry standard, accuracy of these classical models had plateaued in recent years, opening the door for new approaches powered by advanced Deep Learning technology that's also been behind the progress in other fields such as self-driving cars.

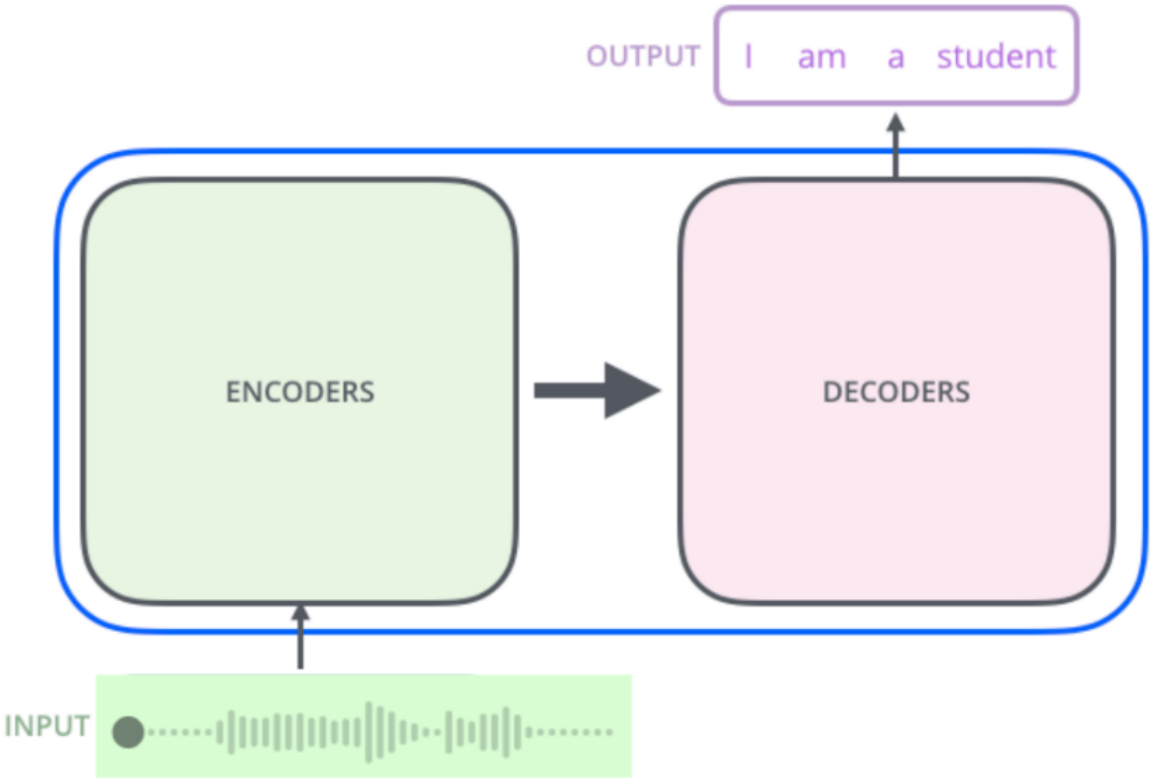
In 2014, Baidu published the famous paper, Deep Speech: Scaling up end-to-end speech recognition. In this paper, the researchers demonstrated the strength of applying Deep Learning research to power state-of-the-art, accurate systems. The paper kicked off a renaissance in the field of ASR, popularizing the Deep Learning approach and pushing model accuracy past the plateau and closer to human level performance. Not only has accuracy skyrocketed, but access to ASR technology has also improved dramatically since.

The main problem is to convert the audio sequence or speech directly to words uttered. ASR is a sequence-to-sequence problem and is a very hot field of research.

3 Transformers

The Transformer in NLP is a novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies with ease. For example the pronouns and verbs at the end of sentences may depend on objects and subjects which had occurred at the very start.

The main idea behind Transformers is Attention. Similar to how we humans understand language, sentences and the overall context by paying more attention to some parts of the sentence than others, Attention models and Transformer architecture try to correlate different parts of the sentence. First, the input instance is encoded by some "layers" of Encoders, which one after the other, incorporate attention details about the input. These details get more coarse and broad as we move to the topmost layer. Then comes the Decoder, which takes data about attention and one-by-one generates the expected output. Training is often done in a masked fashion. We mask or hide some output instances and



force the model to predict them correctly through a loss function. This also allows great speed because we can parallelize these predictions. However, a major flaw such systems suffer with is their general inability to deal with long output sequences.

4 CTC

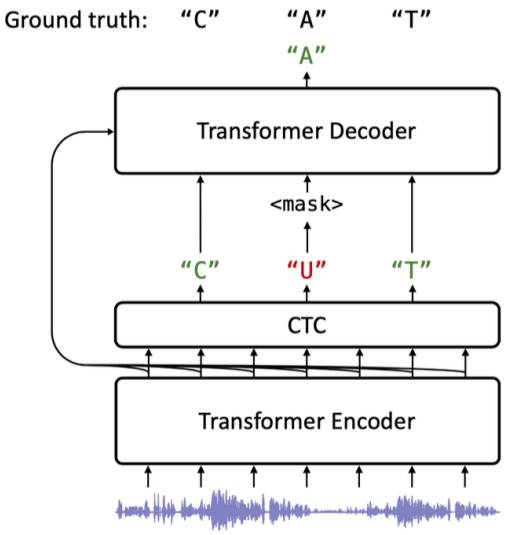
Normally, to train sequence-to-sequence models, alignments between the acoustic features and output labels are necessary. After all, not all speakers are the same. Your grandmom might take much more time than you to speak certain words. Siri still must cater to her demands, right? However, we don't have these alignments when dealing with data on-the-fly. So what alignment do we choose?

The Connectionist Temporal Classification (CTC) loss function doesn't think much, it just starts churning. It considers all possible alignments of the input with the output. You might think this will be dreadfully slow. It isn't! Dynamic Programming comes to the rescue, helping us compute stuff fast. Yes, it all looks great on paper but there is again a major drawback of this method. The very basis of this method forces you to assume that output tokens are independent of each other. But we know this is false for data like sound or language. The result of this flaw is seen in practice - performance is often not as good as other famous ASR systems.

5 Mask-CTC

Neural sequence-to-sequence models are usually **autoregressive**, meaning each output word is generated based on previously generated words or utterances and thus require as many iterations as the output length. On the other hand, **non-autoregressive** models can simultaneously generate tokens within a constant number of iterations, which results in significant inference time reduction and better suits end-to-end ASR models for real-world scenarios.

The authors of the Mask-CTC paper proposed a modification to speed up inference and improve accuracy by taking the best of the two worlds - use length prediction from the CTC method, then refine it's outputs using transformers.



Because a model like Transformer suffers with prediction of the output length, we use CTC first to predict a sequence and freeze the length of the obtained sequence at this stage. However, because CTC's predictions are weak, Transformers come in to boost them. All the predictions where CTC had low confidence are masked and given to the Transformer, which then uses

Attention and Context information from the decoder to strengthen these predictions over a fixed number of iterations. The above figure gives a lucid example -

Overall, we see gains both in the inference time, because of parallelism employed by Transformers and in accuracy, strengthened by CTC specifically for long input instances.

6 Experiments and Results

As in any work of science, results speak for the method. The authors experimented with their model in the following setting -

- The model was trained for three different languages - English, Spanish & Japanese.
- Mask-CTC is found to be a whopping 116 faster than autoregressive ASR models. It also showed improvement in Character/Word/Sentence Error Rates over CTC.
- It was observed that increasing number of iterations during decoding increased performance.

Here is an example of how the usage of Transformer strengthened CTC's predictions -

Ground truth
instead they favor unannounced checks by roving rather than in house inspectors focusing on critical control points in seafood processing
Greedy CTC inference
instead they favor un anounced checks by roving rather than in house anspectors focusing on critical control points in sefood processing
Proposed CTC masking & Iterative decoding ($P_{\text{thres}} = 0.999, K = 3$)
instead they favor un__noun__d ch__ks by roving rather than _n_house _nspectors focusing on crit_cal control points in __food processing
instead they favor unannoun__d ch__ks by roving rather than _n_house inspectors focusing on critical control points in __food processing
instead they favor unannounc_d ch_cks by roving rather than in house inspectors focusing on critical control points in __food processing
instead they favor unannounced checks by roving rather than in house inspectors focusing on critical control points in sifood processing

7 Conclusion

The Mask-CTC model combines two lines of work to form a single model which is strong and fast. However, this is still not perfect as the above example illustrates. Future work can invoke better models for representing language and improved strategies for decoding. Automatic Speech Recognition continues to be unsolved but this strategy does represent a great step forward towards human level performance.

-Images and examples related to Mask-CTC were taken from the original paper.
-Inspiration for the Transformer illustration was drawn from this beautiful article.