1. Speech recognition

1.1 Problem definition

Speech recognition or **Automatic Speech Recognition** field of knowledge which develops techniques for converting captured audio signal into transcript. As input we have a raw audio signal, which can be captured by microphones. The signal is represented as features $\mathbb{X} = \{x_1, x_2, \ldots, x_T\}$, where **T** is the number of frames we divide our audio signal into and corresponding output is represented by text sequence $\mathbb{Y} = \{y_1, y_2, \ldots, y_L\}$, where **L** is the length of vocabulary $\{a, b, c, , \ldots z, !, ?, \ldots\}$. However, we do not know how the characters in the transcript align to each frame of audio signal. The classic goal is to build a generative model which would maximize the following function:

$$Y^* = argmax_Y \mathbf{p}(Y|X)\mathbf{p}(Y) \tag{1.1}$$

where $\mathbf{p}(Y|X)$ refers to acoustic model and $\mathbf{p}$(Y) to language model.

1.2 Existing methods

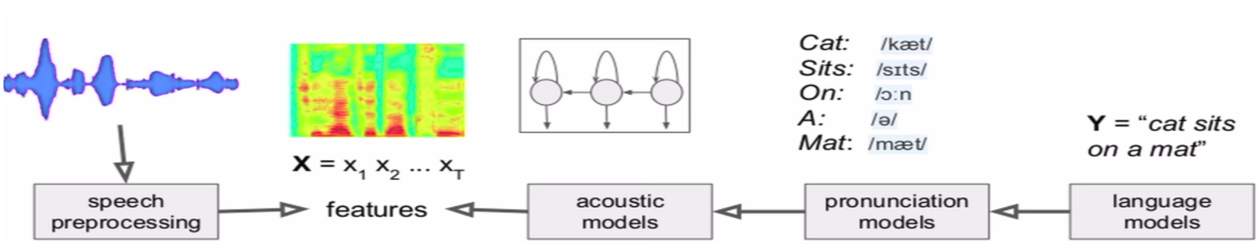

Figure 1.1 — Classical scheme of speech recognition process

In Figure 1.1 [1] we can see the general process of speech recognition. Existing approaches in this field can be generalized as follows:

Table 1.1

Existing methods

Components	Traditional	based on Artificial Neural Networks(NN)
Speech processing	Classical speech processing	Convolutional models on raw signals
Acoustic model	Gaussian mixture models	LSTMS Hiden Markov Models
Pronunciation models(PM)	Pronunciation tables	NN based PM
Language models	N grams models	Neural language models

Methods which are based on Neural Networks preform better than tradition al ones, however they have drawbacks as well: there is separate NN in every component, but each one optimizes its own objective which may not result in a better overall performance. Therefore, so called end to end models were introduced. The most famous of them are:

- Connectionist Temporal Classification (CTC)
- Listen Attend and Spell (LAS)

1.3 Connectionist Temporal Classification

Connectionist Temporal Classification a probabilistic model $\mathbf{p}(Y|X)$. It is widely used for problems in speech and handwriting recognition. We have our fea tures $\mathbb{X}$ spectrogram and outputs $\mathbb{Y}$ transcripts. CTC model gives us an output distribution over all possible $\mathbb{Y}$ for a given $\mathbb{X}$. Our goal is to maximize the proba bility of the right answer for all x_i. We should define the loss function that allows a bidirectional RNN 1.2 [1] to be trained for sequence transcription tasks without requiring any prior alignment between the input and target sequences. However, to compute the probability of an output, CTC perform a sum over the probability of all possible alignments.

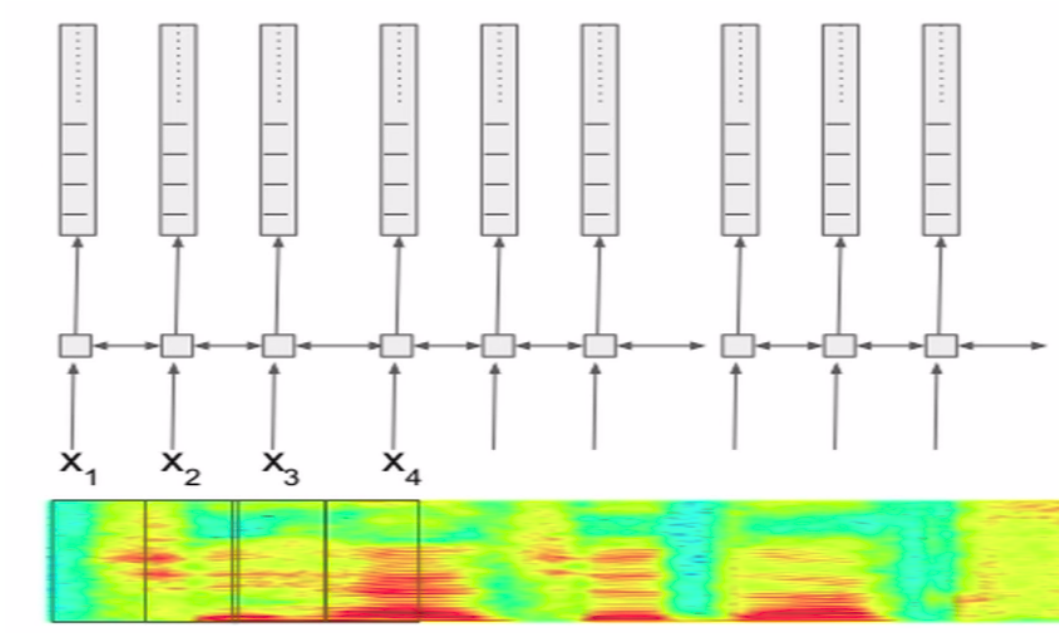

Figure 1.2 — Bidirectional Recurrent Neural network

The output layer of RNN contains a single unit for each of the vocabulary characters, plus an extra unit referred to as the 'blank' which corresponds to a null emission. The length of $\mathbb{Y}$ is the same or shorter than the length of $\mathbb{X}$. Given a length T input sequence x, the output vectors y_t are normalised with the softmax function 1.3, then interpreted as the probability of emitting the label (or blank) with index k at time t 1.3.

$$\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1} e^{x_j}} \tag{1.2}$$

$$\Pr(k, t|\mathbf{x}) = \frac{e^{y_t^k}}{\sum_{k'=1} e^{y_t^{k'}}} \tag{1.3}$$

where y_t^k is element k of y_t. A CTC alignment a is a length T sequence label indices. The probability $Pr(a|x)$ of a is the product of the emission probabilities at every time step:

$$\Pr(a|\mathbf{x}) = \prod_{t=1}^{T} \Pr(a_t, t|\mathbf{x}) \tag{1.4}$$

For a given transcription sequence, there are as many possible alignments as there are different ways of separating the labels with blanks. Denoted with B is an operator that removes repeated labels. The CTC alignment gives us a mechanism to go from probabilities at each time step to the probability of an output sequence. We can rewrite the probability as follows:

$$\Pr(\mathrm{y}|\mathrm{x}) = \sum_{a \in B^{-1}(y)} Pr(a|x) \tag{1.5}$$

Given a target transcription y^*, the network can then be trained to minimize the CTC objective function [2, p.4]:

$$CTC(X) = -logPr(y^*|x) = -\sum_{a \in B^{-1}(y)} logPr(a|x) \tag{1.6}$$

To optimize the function we use stochastic gradient descent. The CTC loss function is differentiable with respect to the per time step output probabilities. Therefore, we can use back propagation algorithm to update weights of our BRNN.

Properties of CTC

- Conditional independence the model assumes that every output is conditionally independent of the other outputs given the input, which is a bad assumption for many seq2seq problems.
- Alignment free
- Alignments are many to one. This property implies that output cannot have more time steps than the input. [3]

1.4 Applications for Speech Recognition

- Direct translation [4]
- Interactive voice response [5]
- Multi speaker multimodal models give a possibility for distinct output with the same input [6]
- Virtual assistant
- Hands free computing

Bibliography

1. Stanford University School of Engineering (2016, April, 3). Lecture 12: End to End Models for Speech Processing[Video file]. Retrieved from https://www.youtube.com/watch?v=3MjIkWxXigM&t=0s&index=13&list=PL3FW7Lu3i5Jsnh1rnUwq_TcylNr7EkRe6

2. Graves Alex and Jaitly Navdeep. Towards End to End Speech Recognition with Recurrent Neural Networks. ICML, 2014.

3. Hannun, "Sequence Modeling with CTC Distill, 2017.

4. Berard, A. and Pietquin, O. and Servan, C. and Besacier, L., "Listen and Trans late: A Proof of Concept for End to End Speech to Text Translation , 2016. Re trieved from http://adsabs.harvard.edu/abs/2016arXiv161201744B

5. Kraft, M. R., Androwich, I. (2012). Interactive Voice Response Technology: A Tool for Improving Healthcare. NI 2012: Proceedings of the 11th International Congress on Nursing Informatics, 2012, 224.

6. D'Ulizia, A.; Ferri, F.; Grifoni P. (2011). "A Learning Algorithm for Multimodal Grammar Inference IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics, Vol. 41 (6), pp. 1495

7. CS224S / LINGUIST285 Spoken Language Processing. Retrieved from http://web.stanford.edu/class/cs224s/syllabus.html

8. Andrew Maas (2016). Neural Networks in Speech Recognition [Lecture notes]. Retrieved from http://cs224d.stanford.edu/syllabus.html