CS 753: Assignment 3

Vamsi Krishna Reddy	Yaswanth Kumar	Vighnesh Reddy
160050064	160050066	160050090

1 Clearly describe the RNN transducer (RNN-T) model. You can use a diagram to accompany your description.

To account for space limitation, we assume to adopt the notation similar to the paper [2] except explicitly defined otherwise. Figure 1 shows the three major components of an RNN-T model *i.e.* encoder (called transcription network in [2]), prediction and joint network. We note that joint network is not there in the original paper [2], but instead combined using a deterministic function with no learnable parameters as described later in the section.

We assume that $\mathcal{H}$ denotes the well known unidirectional RNN or LSTM module as needed. We don't reiterate the exact formulations here.

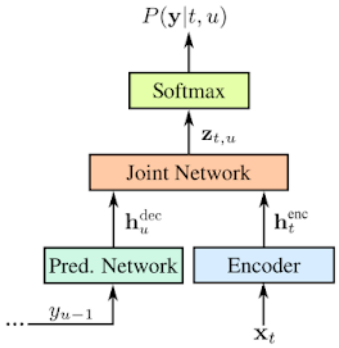

Figure 1: RNN-T architecture from [1]

Encoder / Transcription Network The encoder $\mathcal{F}$ converts the input x_t to a high-level representation h_t^{enc}. In the context of Speech Recognition, this is similar to an acoustic model where the input could be MFCC features of the audio signal. The paper [2] suggests encoder's architecture to be an bidirectional RNN (or LSTM) model.

$$\overrightarrow{h}_t = \mathcal{H}(\overrightarrow{h}_{t-1}) \quad | \quad \overleftarrow{h}_t = \mathcal{H}(\overleftarrow{h}_{t+1}) \quad | \quad h_t^{enc} = f_t = \mathcal{F}(x_t) = W_{\overrightarrow{h}o}\overrightarrow{h}_t + W_{\overleftarrow{h}o}\overleftarrow{h}_t + b_o$$

Prediction Network The prediction network $\mathcal{P}$ produces a high-level representation h_u^{pre} by conditioning over previous non-blank target predicted by RNN-T. In the context of Speech Recognition, this is similar to an RNN language model. The paper [2] suggests its architecture to be an (unidirectional forward) RNN (or LSTM) model (*i.e.* $\mathcal{P}$ is an instance of $\mathcal{H}$).

$$h_u^{pre} = g_u = \mathcal{P}(y_{u-1}) \quad | \quad y_0 = \emptyset$$

Output Distribution The *output density function* and hence the output distribution is obtained using (here superscript k denotes the kth element of the vector)

$$h(t,k,u) = exp(f_t^k + g_u^k) \quad | \quad Pr(k \in \hat{\mathcal{Y}}|t,u) = \frac{h(k,t,u)}{\sum_{k' \in \hat{\mathcal{Y}}} h(k',t,u)}$$

$Pr(k \in \hat{\mathcal{Y}}|t,u)$ determine the transition probabilities in the lattice (Figure 2). Every path in lattice from bottom-left to top-right denotes an alignment and hence $Pr(\boldsymbol{y}|\boldsymbol{x})$ is the sum of probabilities of all such paths which are an alignment of $\boldsymbol{y}$. This is efficiently calculated using the forward-backward algorithm.

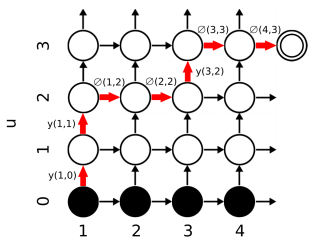

Figure 2: Lattice diagram from [2]

1.1 Training

Once we have $Pr(\boldsymbol{y}|\boldsymbol{x})$, the loss function of RNN-T is $L = -ln(Pr(\boldsymbol{y}|\boldsymbol{x}))$

The architecture until calculation of $Pr(k \in \hat{\mathcal{Y}}|t,u)$ is completely differentiable and similar to modern RNN based models. Also the gradients of loss L *wrt.* $Pr(\boldsymbol{y}|\boldsymbol{x})$ is known using Equation 20 of [2].

Thus the whole pipeline is completely differentiable and hence backpropagation through time is used to learn the parameters of various modules in RNN-T.

1.2 Inference

During inference, we need to find the mode of output sequence distribution represented by the lattice. Since the number of output sequences and hence paths in latticde are exponentially huge, direct estimation of mode is intractable. A standard approach in this scenario is using N-width beam search which keeps track of the N best output sequences for the input sequence seen till now and updating accordingly. During beam-search, we explore the lattice dynamically and choose the width N under computational capacity at hand.

2 How does RNN-T relax CTC's frame independence assumption?

It can be observed by comparing Figure 1 and 3 that in CTC, the output predictions are independent of each other given the input whereas in RNN-T, the prediction network utilizes the previous prediction of output to determine the next output token in the sequence.

RNN-T makes use of the prediction network to learn context information, which functions as a language model thereby relaxing the frame independence assumption that CTC makes. Moreover the probability of an output sequence is equal to the forward variable at the right terminal node in the lattice corresponding to the output sequence which is not equal to the product of probabilities of output symbols given input symbols.

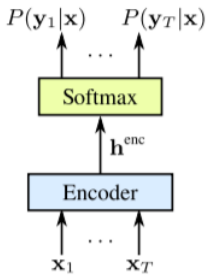

Figure 3: CTC

3 How does RNN-T make it easier to do online streaming? (That is, it can continuously process input samples and stream output symbols.)

During inference we can generate the lattice until the observed number of frames we have seen so far and then generating the output symbols, Later if new frames are observed we extend the lattice generated previously by generating the corresponding number of new transcription vectors from the transcription network by feeding in the new observed frames. Based on the new extended lattice we generate the remaining output symbols.

Hence inference in the RNN transducer is performed in a frame-synchronous manner, and please note that the model can be used to perform streaming recognition if a unidirectional transcription network is used.

4 List one limitation of the RNN transducer model.

From a practical point of view, a limitation observed is the significant complexity in proper training of RNN-T as cited in the literature [3]. However, it is to be noted that many optimizations have been made to the training procedures of RNN-T in literature [1].

It has also been observed to require more memory footprint since the encoder and prediction network compose a grid of alignments, thus involving a three-dimensional tensor during forward-backward training and hence requires more runtime memory compared to CTC training.

References

[1] Google AI Blog, "An all-neural on-device speech recognizer," March 2019. [Online]. Available: https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html

[2] A. Graves, "Sequence transduction with recurrent neural networks," 2012.

[3] T. Bagby, K. Rao, and K. C. Sim, "Efficient implementation of recurrent neural network transducer in tensorflow," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2018, pp. 506–512.