# Paper reading list

* RNN-t first introduced : [Sequence Transduction with Recurrent Neural Networks](https://arxiv.org/pdf/1211.3711)
* RNN-t streaming, all-neural, sequence-to-sequence : EXPLORING ARCHITECTURES, DATA AND UNITS FOR STREAMING END-TO-END SPEECH RECOGNITION WITH RNN-TRANSDUCER
* Personalizing ASR for Dysarthric and Accented Speech with Limited Data

# Compare models

|  |  |
| --- | --- |
|  |  |

# 

# RNN-t first introduced

|  |  |
| --- | --- |
| previous work | RNN :  advantage:   * learning to represent both the input and output sequences in a way that is invariant to sequential distortions such as shrinking, stretching and translating * better at storing and accessing information over long periods of time   drawback:  require a pre-defined alignment between the input and output sequences to perform transduction. (alignment is hard to get) |
| RNN-T | * + end-to-end, probabilistic sequence transduction system, based entirely on RNNs |
| CTC | * + previous: CTC output length is not longer than input(can not work for task like text2speech)   + this paper: extends CTC by defining a distribution over output sequences of all lengths   + Conditional independent assumption     - Labels at each time index are conditionally independent (like HMMs) |

## RNN transducer, how it can be trained and applied to test data

|  |  |
| --- | --- |
| Formula | x = (x1,x2,...,xT) be a length T input  y = (y1, y2, . . . , yU ) be **a length U output sequence**  Elements as alignments; X \* as x sequence set  B is a function that remove null symbols from alignment  extended output space as Y ∪ ∅    Two recurrent neural networks are used to determine Pr(a ∈ |x)   1. Transcription network F, input sequence x and output sequence f=(f1,. . . , fT) of transcription vectors 2. Prediction network G, scans sequence y and output prediction vector sequence g=(g0, . . . ,gU) |
| Prediction network G | Role:   * The prediction network attempts to model each element of y given the previous ones; it is therefore similar to a standard next-step-prediction RNN * The RNN-T removes the conditional independence assumption in CTC by introducing a *prediction network*   Structure :   * G is a RNN : an input layer, an output layer and a single hidden layer.   Input layer of G:   * The length U + 1 input sequence = (∅,y1,...,yU) to G (and a ∅ as a starting point ) The inputs are encoded as one-hot vectors;   Hidden layer h:       1. H is the activation function :    1. Traditional rnn tends to use tanh or sigmoid    2. LSTM in this paper activation function :     : input gate  : forget gate  : output gate  s : state vectors  The weight matrices from the state to gate vectors are diagonal, so element m in each gate vector only receives input from element m of the state vector. |
| Transcription Network F | Role:   * to define a distribution over input-output alignments   Structure:   * Bidirectional RNN : scans input sequence x forwards and backwards with two separate hidden layers. Both of which feed forward to a single output layer.(for Y consists of K labels , the output layer is size K+1, with null symbol)   Input layer  Hidden layer (backwords) Hidden layer(forwards)  Output layer  Backward layer from T to 1    Forward layer from 1 to T |
| Output distribution |  |
| Decoding | Pr(k|t, u) (in Equation 13) is used to determine the transition probabilities in the lattice shown here. [each circle here is a K+1 classifier].  2.4 |
| Compare to LAS | the RNN-T model can be compared to LAS  Transcription network F => Encoder  Prediction network G + joint network => Decoder. |

# RNN-t streaming

|  |  |
| --- | --- |
| this paper | * + jointly(联合地) learns acoustic and language model components from transcribed acoustic data     - encoder(acoustic model) : CTC acoustic model     - decoder(language model) : a recurrent neural network language model trained on text data alone.     - entire model trained with RNN-T loss   + Experiment best with:     - sub-word units (‘wordpieces’)     - a twelve-layer LSTM encoder with a two-layer LSTM decoder trained with 30,000 wordpieces as output targets     - ASR 5.2% |
| formula | W = w1 , ..., wn ,the most likely word sequence,  x = x1 , ..., xT ,given an acoustic input sequence  where T represents the number of frames in the utterance |
| Joint network | Role: takes in the output of encoder and decoder , and output a distribution over next symbol  detail in A. Graves, A.-R. Mohamed, and G. E. Hinton, “Speech recognition with deep recurrent neural networks,” 2013. |
| RNNT loss | which marginalizes over all alignments of target labels with blanks as in CTC, and is computed using dynamic programming.  detail in A. Graves, “Sequence transduction with recurrent neural networks,” in *Proc. of ICASSP*, 2012. |
| pre-training | Encoder:  It has been previously shown that initializing RNN-T encoder parameters from a model trained with the CTC loss is beneficial for the phoneme recognition task  decoder  the prediction network is not conditioned on the encoder output. This allows for the pre-training of the decoder as a RNN language model on text-only data .  trained with cross-entropy loss. |
| word piece | word piece advantage:  tunable number of labels  better on longer context than graphemes(single character too slow to infer)    For the wordpiece models which have longer duration than graphemes, we employ an additional ’time-convolution’ in the encoder network to reduce the sequence length of encoded activations which is similar to the pyramidal sequence length reduction.    Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, Q. Macherey, M. Krikun, Y. Cao, Q. Gao, and K. Macherey et al., “Google’s neural machine translation system: Bridging the gap between human and machine translation,” 2016. |
| training | To compare with conventional ASR:  18,000 h training data => acoustic model  150 million sentence => 5gram lm  RNN-T is trained with the same data as baseline  The CTC encoder network is pre-trained with acoustic transcribed data and as with the baseline acoustic model the pronunciation model is used to generate phoneme transcriptions for the acoustic data. |
| Evaluation | a beam of 100 for grapheme models and 25 for wordpiece models and a temperature of 1.5 on the softmax  test set number:15,000 utterances |
|  |  |

# Finetuning

## Personalizing ASR for Dysarthric and Accented Speech with Limited Data

|  |  |
| --- | --- |
| Previous methods | 1, appended articulatory features to the usual acoustic ones to achieve a 4-8% relative WER improvement,  2, adapts ASR models trained on open source datasets to a dataset of dysarthric speech.  Drawbacks:  limited by the quality of the base model  lack of data available for finetuning. |

|  |  |
| --- | --- |
| steps overview | * 1. We create personalized ASR models by starting with a base model trained on standard, unaccented speech. (RNN-T LAS googleCloud)   2. For each speaker : finetune different layer combinations on different amounts of data, on both ALS and accents datasets. |

1, training basic model

|  |  |
| --- | --- |
| base model | train data  During training, we distorted the audio using a room simulator derived from YouTube data. The average SNR of the added noise is 12dB. |
| RNN-T | It has a 5 layer bidirectional convolutional LSTM encoder, a 2 layer LSTM decoder, and a joint layer. It has 49.6M parameters in total. ref |
| LAS | It is a sequence- to-sequence with attention model that maps acoustic frames to graphemes.  We use an encoder with 4 layers of bidrectional convolutional LSTMs and a 2 layer RNN decoder.  The model has a total of 132M parameters.    The base LAS model was trained to 1M steps on all 960 hours of the Librispeech dataset.  It achieved a 5.5% WER on the clean test split and 15.5% on the non-clean |
| Google Cloud | Google cloud ASR model only looks backwards in time. (streaming) [RNN-T and LAS look whole audio] |

2, fine tuning

|  |  |
| --- | --- |
| Fine tune -RNNT | Ei : denote the ith layer of the encoder  training from E0 up with or without the joint layer was always better than the other methods,  we exhaustively searched our finetuning space    By exhastively search here means => finetuning each of E0, E0-E1, E0-E2, etc.. the entire encoder, with or without the joint layer. |
| Fine tune LAS | best result achieved by finetuning all layers |

|  |  |
| --- | --- |
| training / testing set | L2   * For each of the 20 speakers, we split the data into 90/10 train and test.   \* All of the sentences which contains proper nouns are used in the training set, in order to remove the possibility of the model to artificially achieve better results on the test set by memorizing them |
| finetune time / machine | All our finetuning uses four Tesla V100 GPUs for no more than four hours. |

3. Test

|  |  |
| --- | --- |
| Test result |  |