Neural Speech Recognition

Jim Glass / MIT 6.806-6.864 / Spring 2021

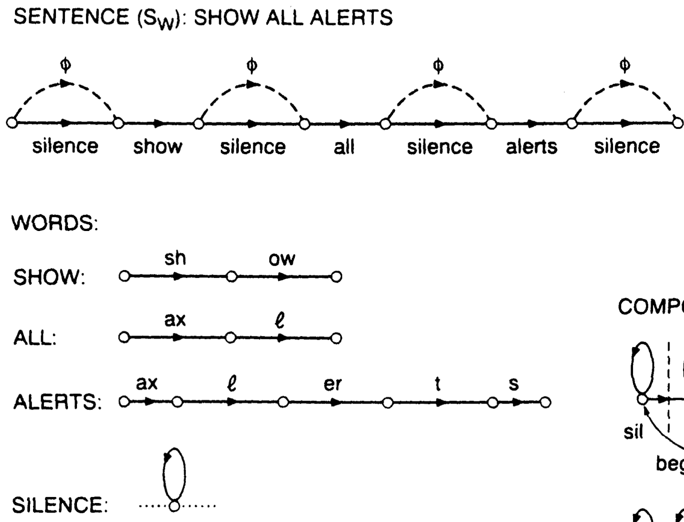
1

Today's ASR Topics

- Finishing up classic ASR from last lecture
  - Search space representation via FSTs
  - Corpora and evaluations
- Neural ASR methods
  - Hybrid and tandem ANN-HMM models
  - Listen, Attend, & Spell (LAS)
  - Connectionist temporal classification (CTC)
  - RNN-Transformer (RNN-T)

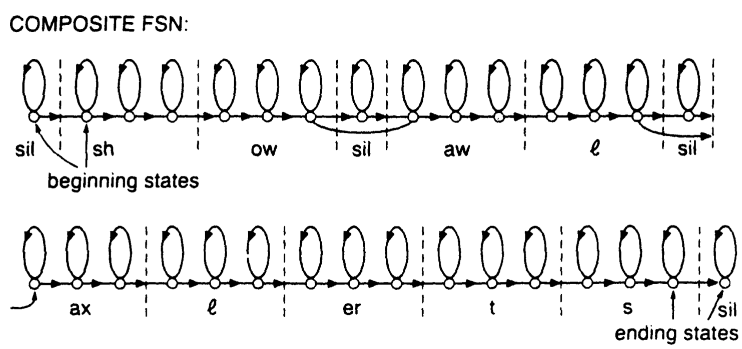
2

Recap: HMM ASR Formulation



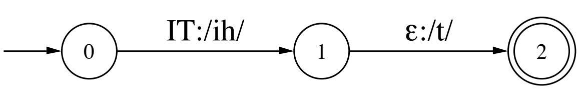
Words and word sequences can be represented as Markov chains

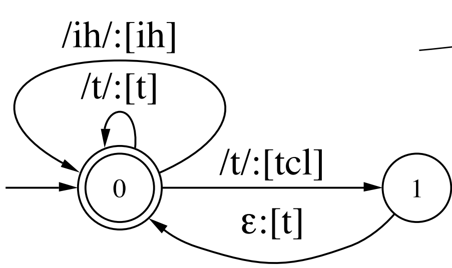
HMM decoding can be thought of as a giant directed graph search

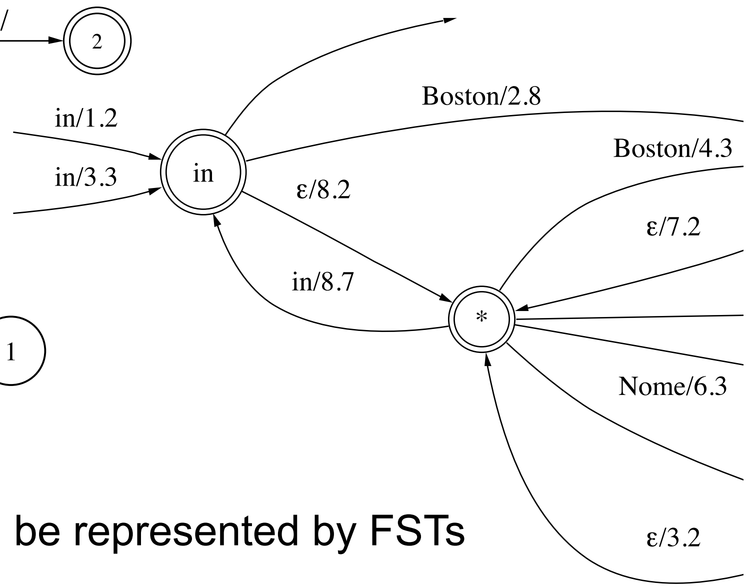


3

Finite State Transducer Representations for ASR







Many ASR constraints can be represented by FSTs

4

Speech Recognition as Cascade of FSTs

- ASR as a cascade of FSTs:

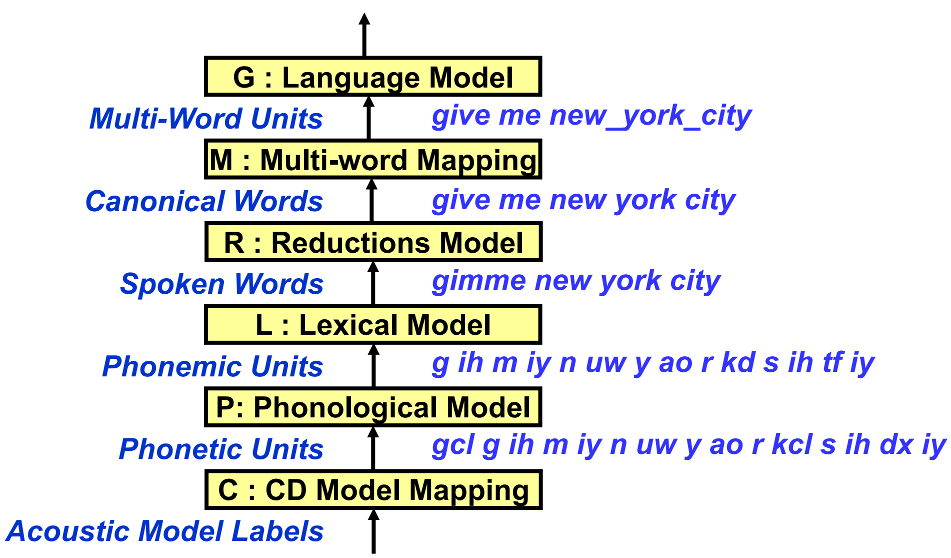
$$O \circ (M \circ P \circ L \circ G)$$

  - G: language model (weighted words ← words)
  - L: lexicon (phonemes ← words)
  - P: phonological rule application (phones ← phonemes)
  - M: model topology (e.g., HMM) (states ← phones)
  - O: observations with acoustic model scores
- (M ∘ P ∘ L ∘ G) is *single* FST seen by search
- Viterbi search performs composition of O with (M ∘ P ∘ L ∘ G)
- Gives great flexibility in how components are combined

5

Expanded FST Representation

- FST representation can be expanded for more efficient representation of lexical variation



6

Evaluating Speech Recognition

- The standard evaluation metric for ASR is word error rate (WER)
  - Based on a string alignment between hypothesis and reference text
  - Errors can include insertions, deletions, and substitutions

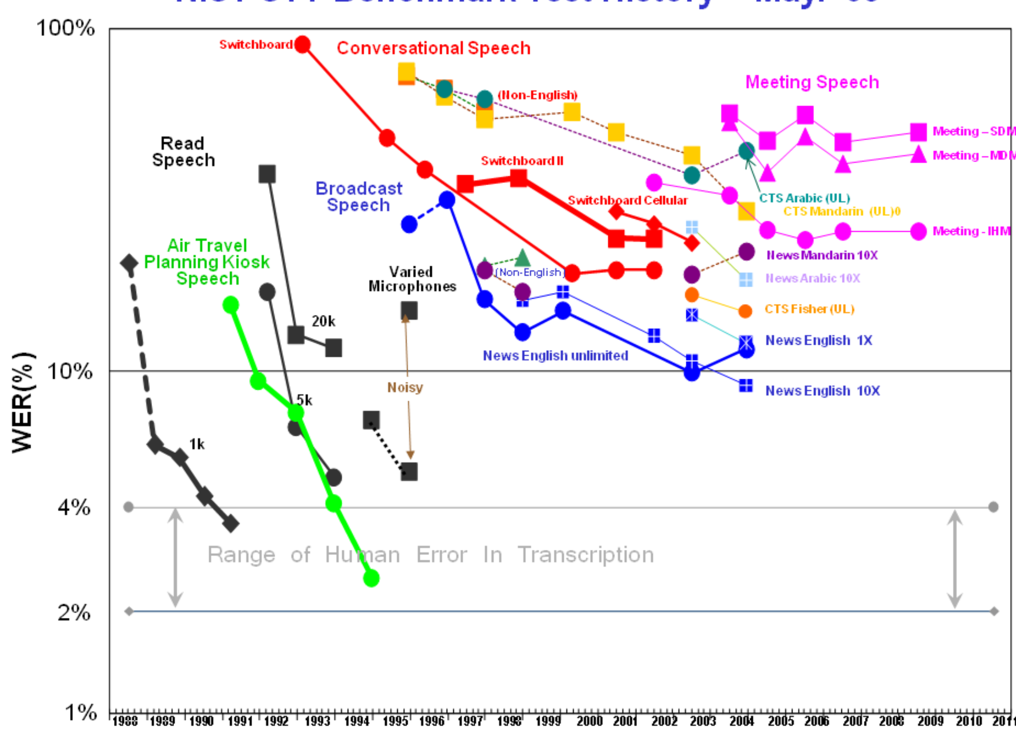
REF:	i	***	**	UM	the	PHONE	IS	i	LEFT	THE	portable	****	PHONE	UPSTAIRS	last night
HYP:	i	GOT	IT	TO	the	*****	FULLEST	i	LOVE	TO	portable	FORM	OF	STORES	last night
Eval:		I	I	S		D	S		S	S		I	S	S	

$$WER = 100 \text{ x } \frac{Insertions + Substitutions + Deletions}{Total\ Words\ in\ Correct\ Transcript}$$

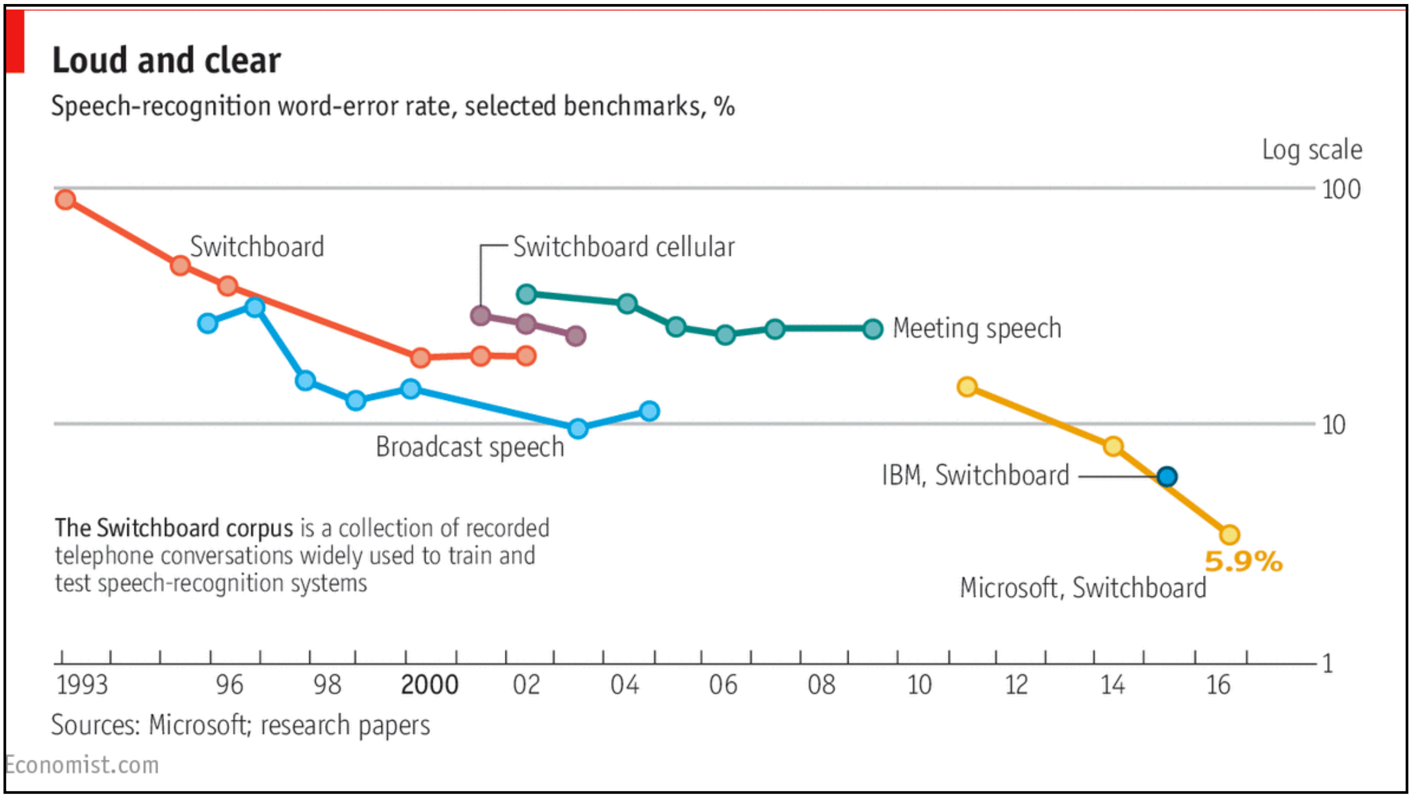
- The same metric can be applied to character error rate (CER) etc.
- String edit distance tends to underestimate errors at high WERs
- Standard NIST software (sclite) is available to measure WERs

8

NIST STT Benchmark Test History – May. '09

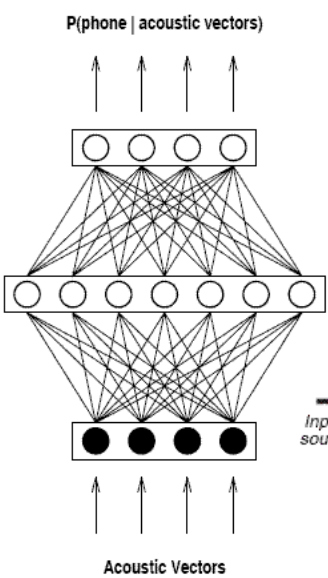


9

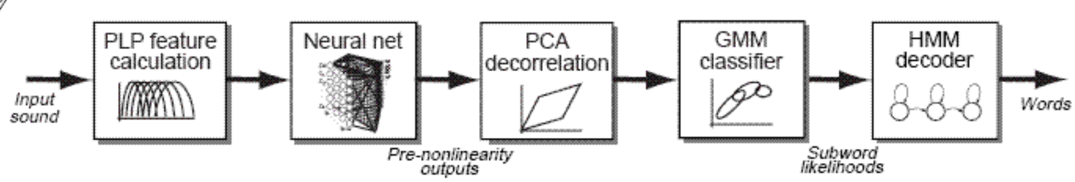


10

Neural Network-based ASR



- Artificial neural network (ANN) models were first explored for ASR in late 1980s (e.g., MLPs)
- Over the last decade ANNs have replaced GMMs as the dominant acoustic model for ASR
- ANN outputs can replace GMMs altogether (*Hybrid*), or be used as inputs to HMM-GMMs (*Tandem*)

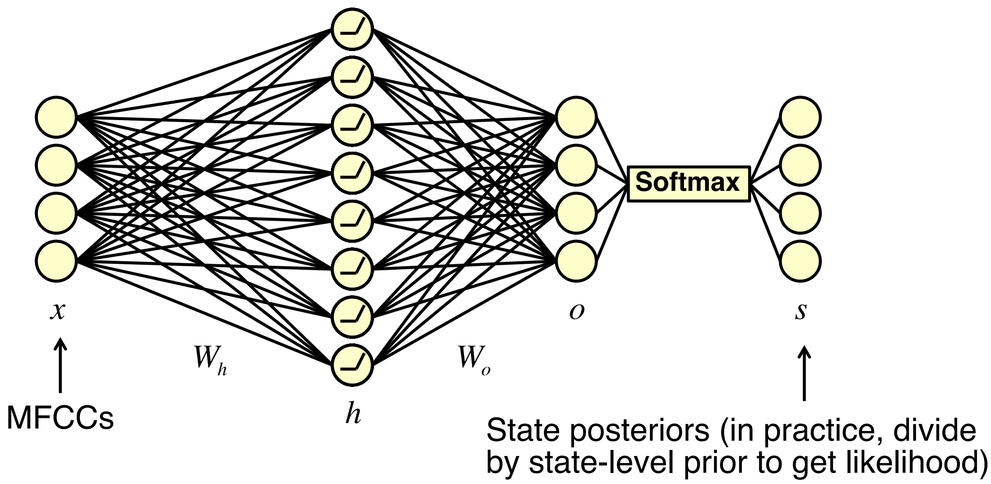


- Connectionist Temporal Classification (CTC) and attention-based Seq2Seq models replace HMMs

11

DNN Acoustic Models

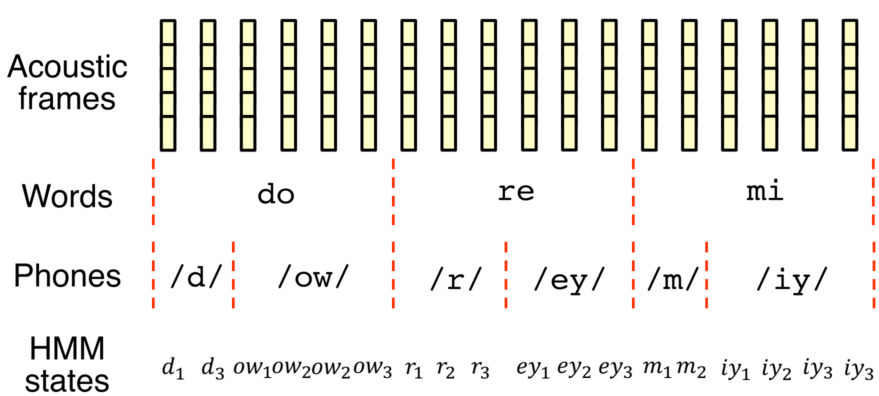
- Simplest approach: feed-forward frame-level classifier
  - Inputs are acoustic feature vectors (e.g. MFCCs, filterbanks, etc.)
  - Targets are HMM states (we're just replacing the GMM)



12

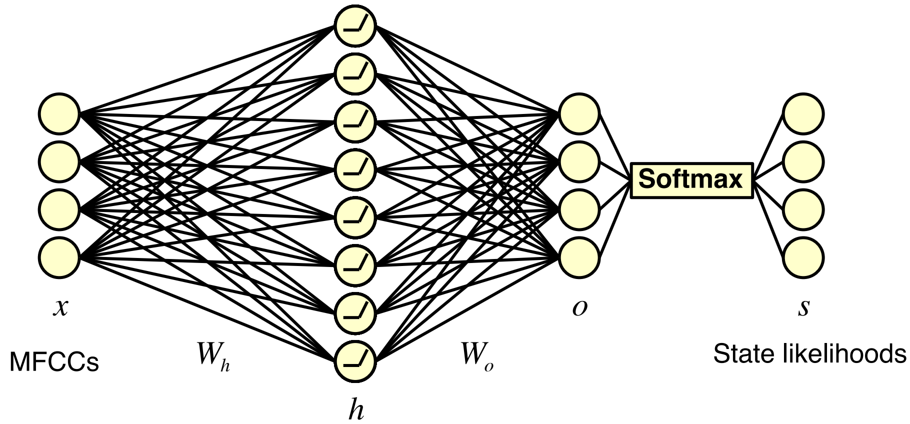
Viterbi Training

- Where do we get the frame-level DNN classification targets?
- Use Viterbi to find best alignment of acoustic frames to HMM states
- Use the HMM states as the classification targets



13

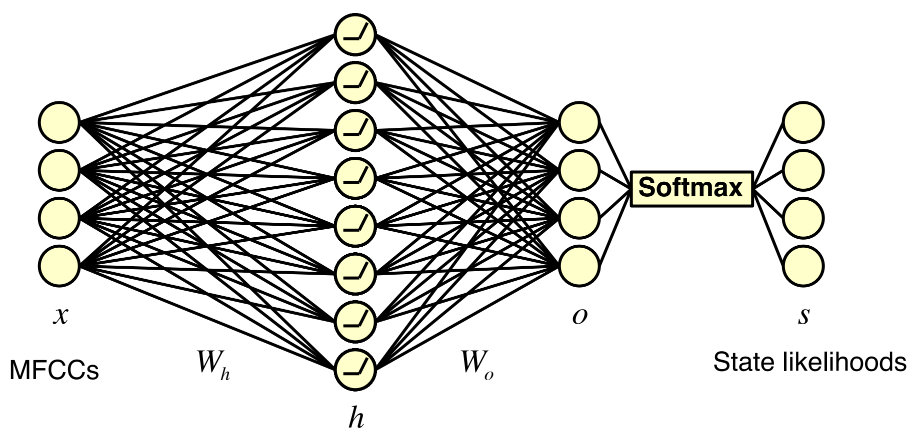
Tandem ANN-HMM Approach



1. Train DNN model to predict HMM state probabilities per frame
2. Extract o vector (often called “Bottleneck Features”) from the DNN
3. Retrain GMM-HMM system using bottleneck features

14

Hybrid ANN-HMM Approach

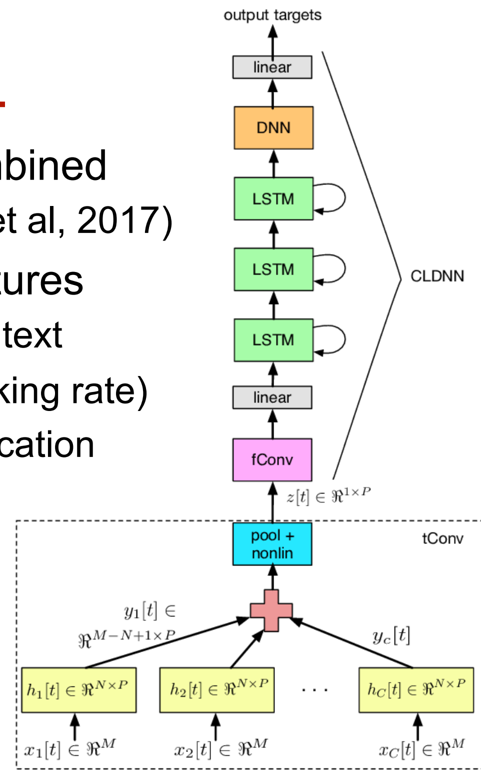


1. Derive frame-level forced alignments from an initial GMM-HMM
2. Train DNN to predict HMM state probabilities per frame
3. Directly use the predicted state likelihoods during decoding

15

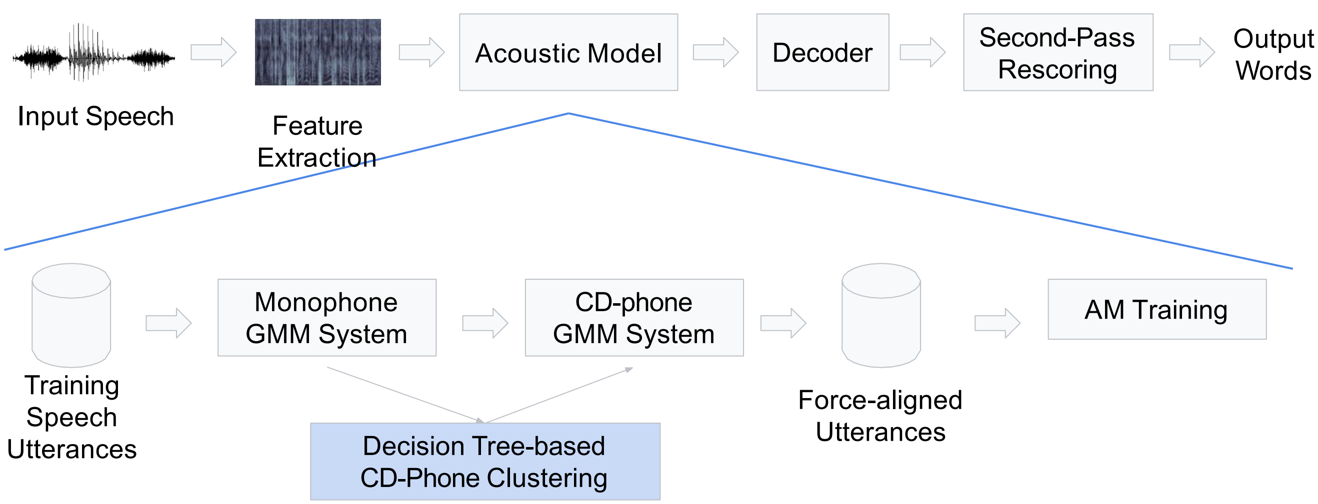
Hybrid ANN-HMM Models for ASR

- Different types of neural models can be combined
  - e.g., CNN + LSTM + DNN = CLDNN (Sainath et al, 2017)
- Exploit strengths of different neural architectures
  - CNNs for capturing local acoustic-phonetic context
  - RNNs for capturing context (and variable speaking rate)
  - DNNs for discriminative power for state classification



16

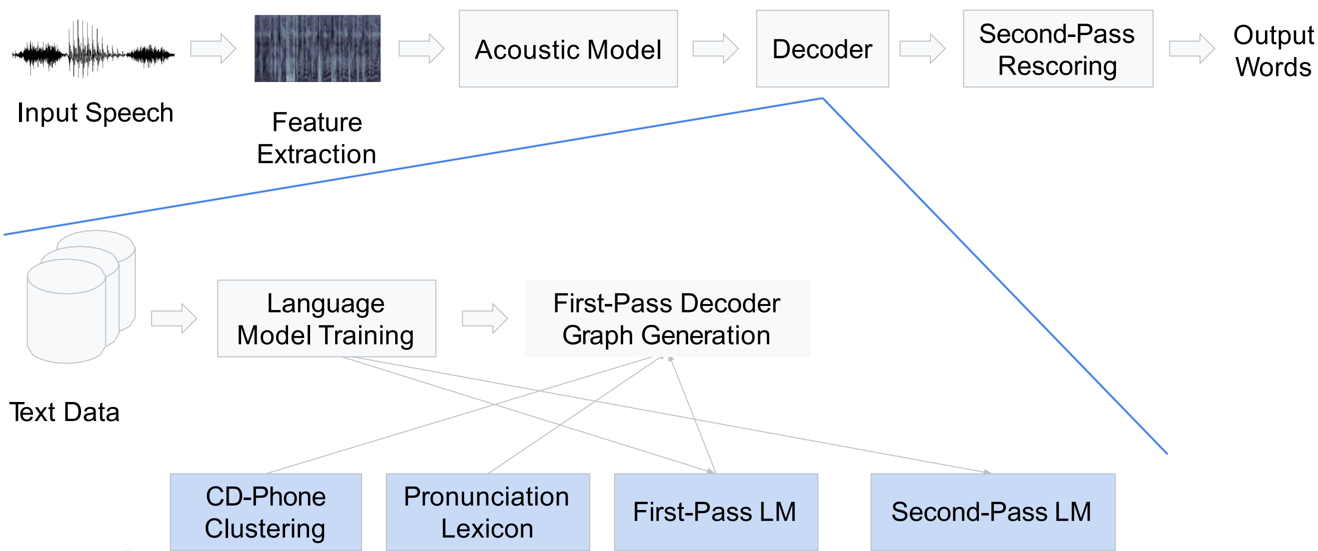
Conventional ASR Pipeline



[Sainath, 2019]

17

Conventional ASR Pipeline

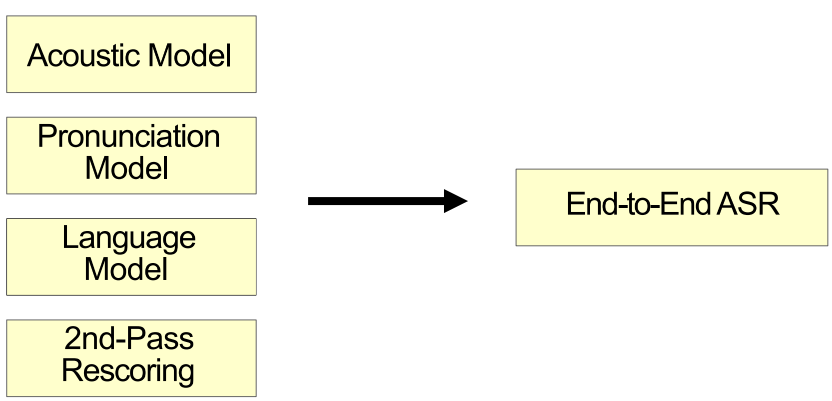


[Sainath, 2019]

18

End-to-End ASR

- A system which directly maps a sequence of input acoustic features into a sequence of graphemes or words
- A system which is trained to optimize criteria that are related to the final evaluation metric we are interested in (e.g., WER)



19

Attention-based Encoder-Decoder ASR Models

Listen, Attend and Spell

William Chan
Carnegie Mellon University
williamchan@cmu.edu

Navdeep Jaitly, Quoc V. Le, Oriol Vinyals
Google Brain
{ndjaitly,qvl,vinyals}@google.com

[Chan et al., CoRR, 2015]

Attention-Based Models for Speech Recognition

Jan Chorowski
University of Wrocław, Poland
jan.chorowski@ii.uni.wroc.pl

Dzmitry Bahdanau
Jacobs University Bremen, Germany

Dmitriy Serdyuk
Université de Montréal

Kyunghyun Cho
Université de Montréal

Yoshua Bengio
Université de Montréal
CIFAR Senior Fellow

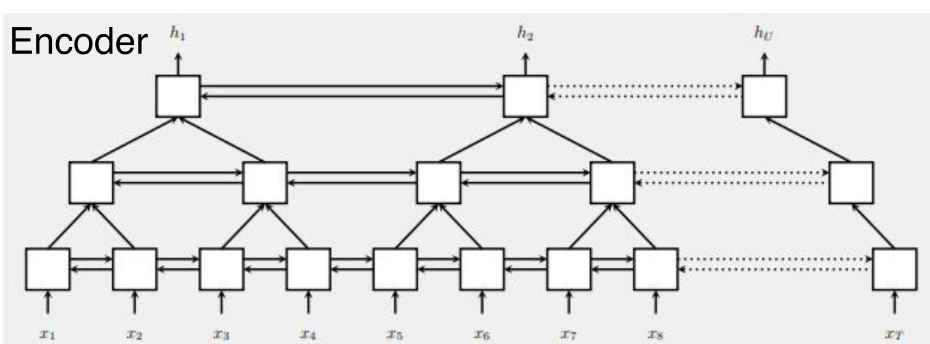
[Chorowski et al., NIPS, 2015]

- Attention-based encoder-decoder models emerged first in the context of neural machine translation; applied to ASR in 2015

20

Listen, Attend and Spell (LAS)

- Consists of an encoder (aka Listener) and a decoder (aka Speller)
  - Inspired by sequence-to-sequence modeling with attention
- Encoder based on multilayer (e.g., 3) bidirectional RNNs (LSTMs)
  - Each layer reduces time resolution by a factor of 2 (i.e., a net factor of 8)
  - Downsampling reduces computation complexity and speeds learning

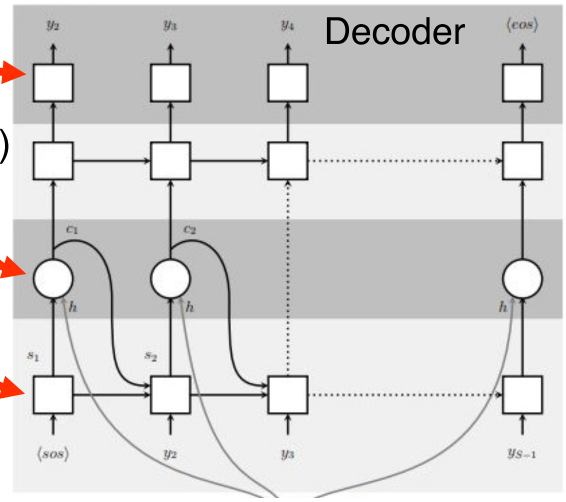


[Chan et al., 2015]

21

Listen, Attend and Spell: Speller

- Decoder output y_i based on internal state s_i and context vector c_i
  - Outputs consists of characters and special sentence markers <sos> <eos>
  - Output a simple MLP with *softmax*
- Context c_i attends all h based on state s_i
  - Various attention models (e.g., dot product)
  - Attention focuses on local observations
- State, s_i, based on s_{i-1}, y_{i-1}, and c_{i-1}
  - Based on a multilayer RNN (e.g., 2 LSTMs)
- Decoding ends when <eos> generated



Encoder Output $h = \{h_1, \ldots, h_U\}$

[Chan et al., 2015]

22

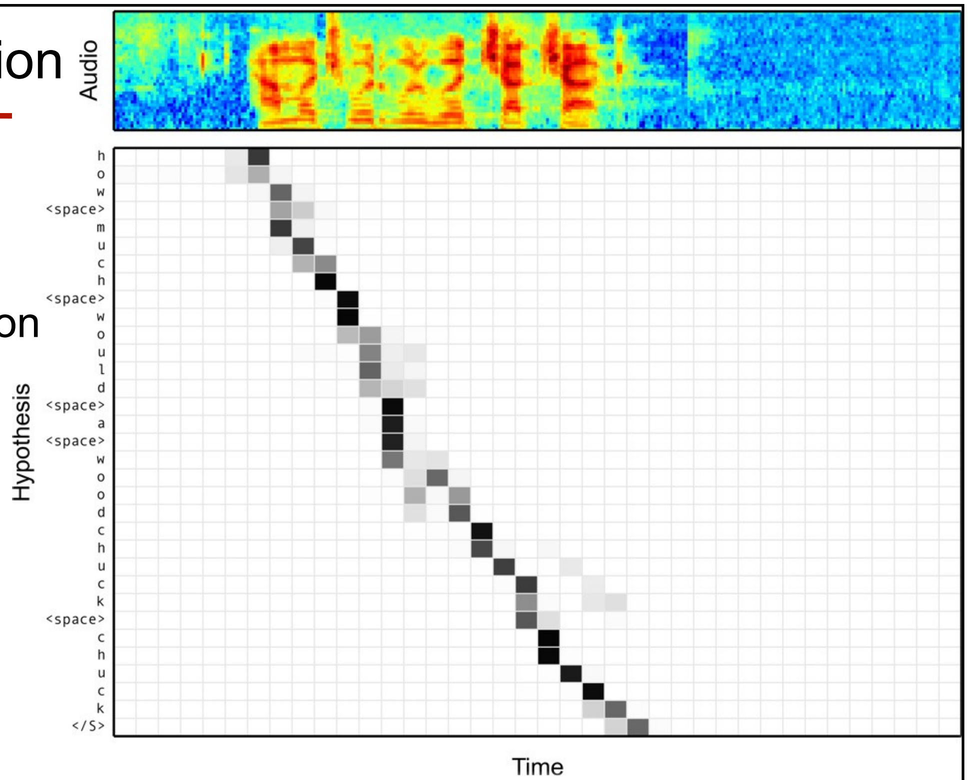
LAS: Training and Decoding

- LAS models are trained jointly to maximize $\sum_i \log P(y_i | x, y_{<i})$
- To improve robustness to bad predictions, a sampled output is sometimes used as input instead of the ground truth
- Decoding is performed with a left-to-right beam search
  - At each time step all partial hypotheses are expanded with all characters
  - Only the top scoring partial hypotheses are retained in beam (e.g., 32)
  - When <eos> token is encountered, path is removed and retained
- Another language model can be used to rescore final hypotheses

23

Visualizing Attention

- Attention is roughly monotonic in time
- Model can be confused by repetition



[Chan et al., 2016]

24

Connectionist Temporal Classification (CTC)

Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks

Alex Graves[1] ALEX@IDSIA.CH
Santiago Fernández[1] SANTIAGO@IDSIA.CH
Faustino Gomez[1] TINO@IDSIA.CH
Jürgen Schmidhuber[1,2] JUERGEN@IDSIA.CH
[1] Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Manno-Lugano, Switzerland
[2] Technische Universität München (TUM), Boltzmannstr. 3, 85748 Garching, Munich, Germany

[Graves et al., ICML, 2006]

Towards End-to-End Speech Recognition with Recurrent Neural Networks

Alex Graves GRAVES@CS.TORONTO.EDU
Google DeepMind, London, United Kingdom
Navdeep Jaitly NDJAITLY@CS.TORONTO.EDU
Department of Computer Science, University of Toronto, Canada

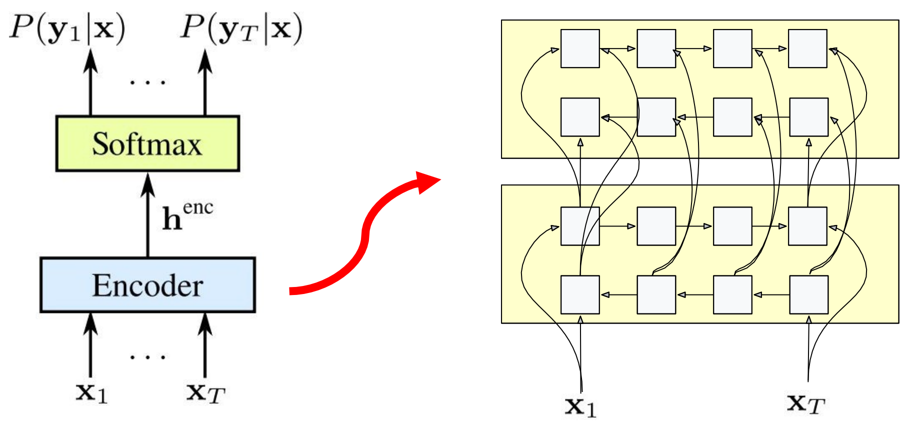
[Graves et al., ICML, 2014]

- Conventional neural models require labeled data for training
  – For speech recognition this requires frame-level labels (e.g., every 10ms)
- CTC allows for training without the need for frame-level alignments
  – Initially for phone recognition; later for end-to-end character & word ASR
- Appropriate when tasks labeling unsegmented data sequences

25

Connectionist Temporal Classification (CTC)

- CTC models generate a label at the same frame-rate as the input
  - Outputs phonetic or character symbols, then removes repeats
- Encoder consists of multiple layers of uni- or bidirectional RNNs
  - Bidirectional RNNs perform best; unidirectional enable streaming ASR



26

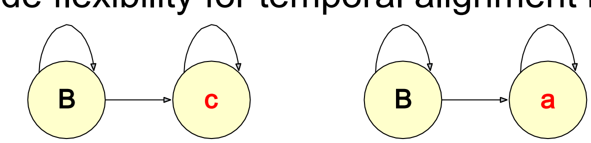
CTC Alignments

- CTC introduces a "blank" (B) symbol to distinguish repeated outputs
  - Repeating outputs removed, then blanks removed in final output
- Many possible frame alignments, $\hat{y}$, can produce the same output, y

Possible frame alignments, $\hat{y}$'s, for $y =$ "cat"

```
B B c B B a a B B t
B c c B a B B B B t
        . . .
B c B B a B B t t B
```

- CTC model corresponds to an HMM model with a shared initial state (blank), followed by a separate state for the actual output unit
  - Self-loops provide flexibility for temporal alignment between input & output



27

CTC Modeling

- CTC objective function maximizes probability of output label sequence, y, by marginalizing over all possible alignments, $\hat{y}$

```
B B c B B a a B B t
B c c B a B B B B t
        . . .
B c B B a B B t t B
```

$$P_{CTC}(y|x) = \sum_{\hat{y} \in \beta(y,x)} \prod_{t=1}^{T} P(\hat{y}_t|x)$$

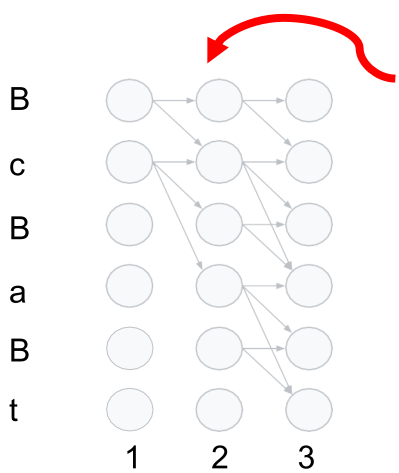
Where $\beta(y,x)$ is all possible alignments of y for input x

- CTC assumes outputs are conditionally independent from each other
  - CTC relies on external language models for sequential constraints
- Computing $P_{CTC}(y|x)$ is similar to computing $P(\boldsymbol{O})$ in HMMs
  - Can be computed recursively like the forward-backward algorithm!

28

CTC Training

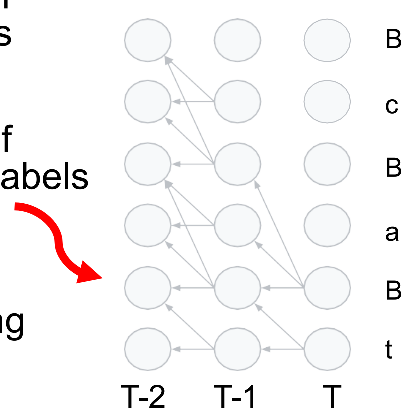
- Computing the gradients of the loss requires the computation of the alpha-beta variables using the forward-backward algorithm



$\alpha(t,u)$: probability of emitting first u labels in first t frames

$\beta(t,u)$: probability of emitting last $\mathrm{U}-u$ labels in last $\mathrm{T}-t$ frames

$\alpha(t,u)\beta(t,u)$: probability of emitting label u at frame t

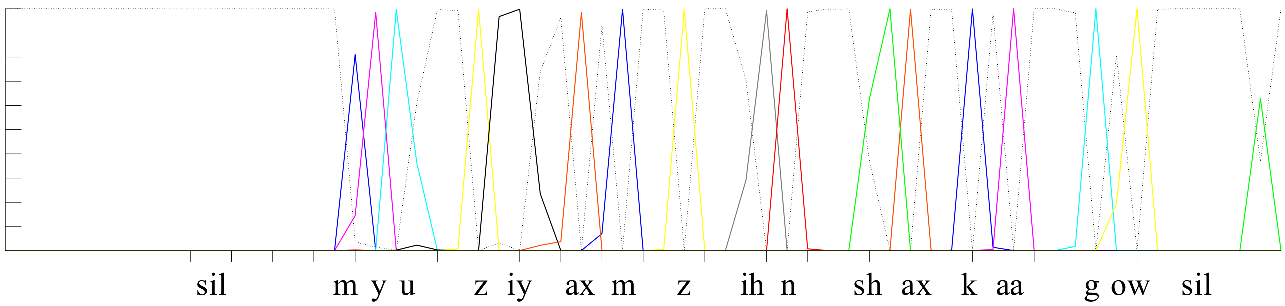


Frames, t

29

Visualizing Alignments

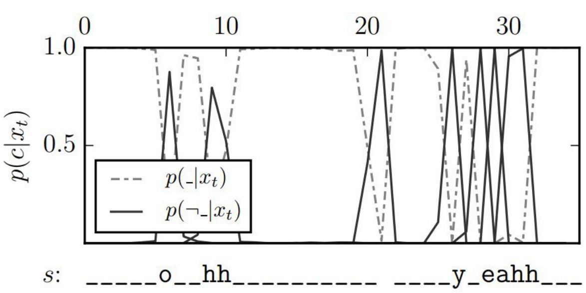
- CTC produces “spiky” and sparse activations; can sometimes read off the final transcript from the output even without an LM



[Sak et al., 2015]

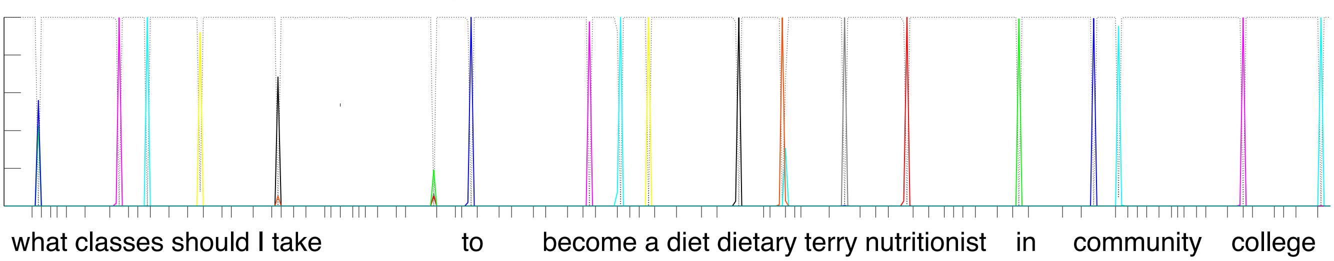
Visualizing Alignments

- CTC produces “spiky” and sparse activations; can sometimes read off the final transcript from the output even without an LM



[Maas et al., 2015]

[Sak et al., 2015]



Recurrent Neural Network Transducer (RNN-T)

Sequence Transduction with Recurrent Neural Networks [Graves ICML, 2012]

Alex Graves GRAVES@CS.TORONTO.EDU
Department of Computer Science, University of Toronto, Canada

SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORKS [Graves et al., ICASSP, 2013]

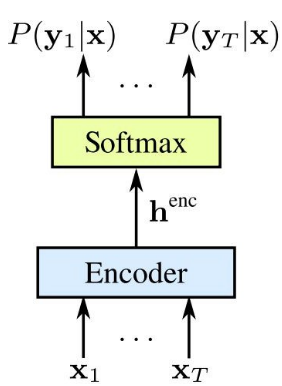
Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton

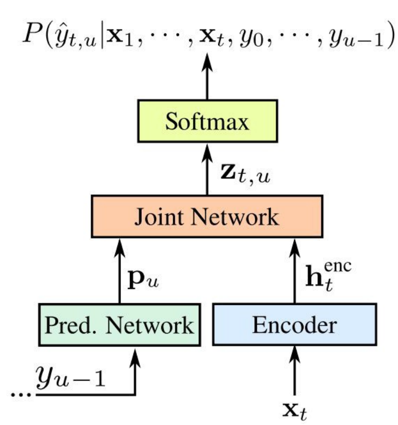
Department of Computer Science, University of Toronto

- RNN-T augments a CTC-based model with an RNN LM
- Both components are trained jointly on speech training data
- As with CTC, RNN-T does not require aligned training data

RNN-Transducer ASR

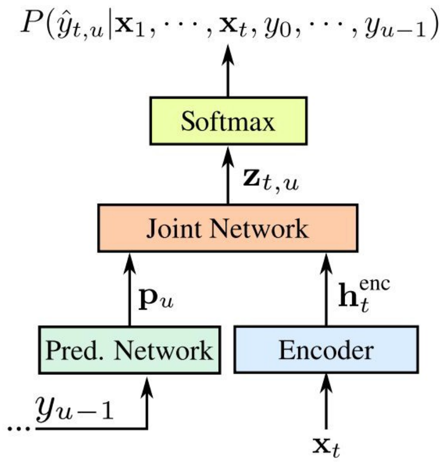
- RNN-T augments the CTC “transcriber” with a “prediction” RNN LM
  - A joint network combines LM predictions, p_u, with CTC predictions
- Decoding uses beam search and proceeds time-synchronously
  - Inference terminates when all frames have been consumed

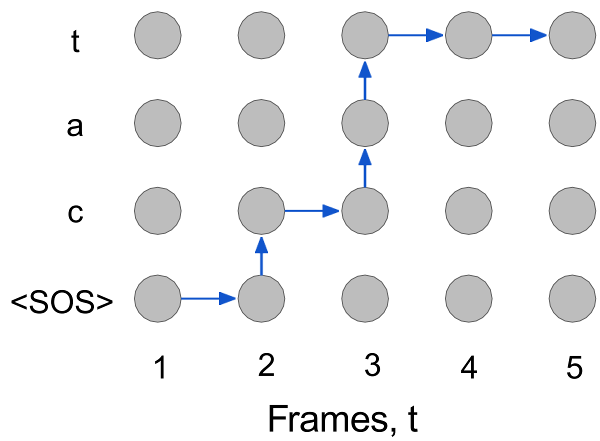




RNN-T Training

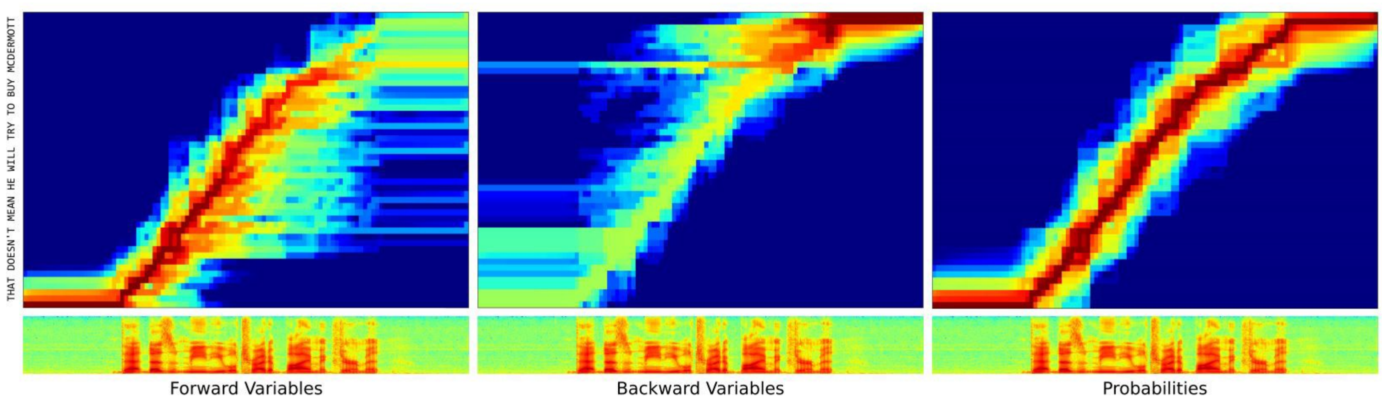
- During training feed the true label sequence to the LM
- Given a target sequence of length U and T, generate $U \times T$ *softmax*





34

Recurrent Neural Network Transducer (RNN-T)



$\alpha(t,u)$: probability of emitting first u labels in first t frames

$\beta(t,u)$: probability of emitting last $\mathrm{U} - u$ labels in last $\mathrm{T} - t$ frames

$\alpha(t,u)\beta(t,u)$: probability of emitting label u at frame t

[Graves, ICML, 2012]

35

Comparing LAS and CTC Models

- Attention-based encoder-decoder ASR models like LAS perform extremely well when given a large speech dataset
  - Have unlimited flexibility to increase parameter sizes
  - Learn language model constraints along with acoustic constraints
  - Best used for off-line or cloud-based speech processing
- CTC-based models are well matched to ASR and other tasks labeling unsegmented data sequences (e.g., handwriting)
  - Assume the output label sequence is shorter than the input sequence
  - Assume monotonic label progression (in contrast to attention models)
  - Assume label output conditionally independent given inputs
  - Can be integrated with language model predictor in RNN-T
  - Effective for streaming ASR decoding with unidirectional models

36

References

- Readings:
  - Jurafsky and Martin, "Speech and Language Processing", Chp. 26

- Extra readings:
  - Chan et al., "Listen, Attend and Spell," arXiv:1508.01211v2, 2015
  - Graves et al., "Connectionist Temporal Classification," ICML, 2006
  - Graves, "Sequence Transduction with RNNs," arXiv:1211.3711v1, 2012

37