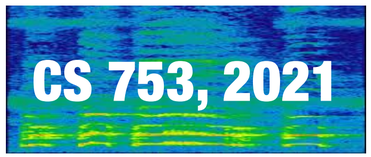
End-to-End Neural ASR Systems (II)

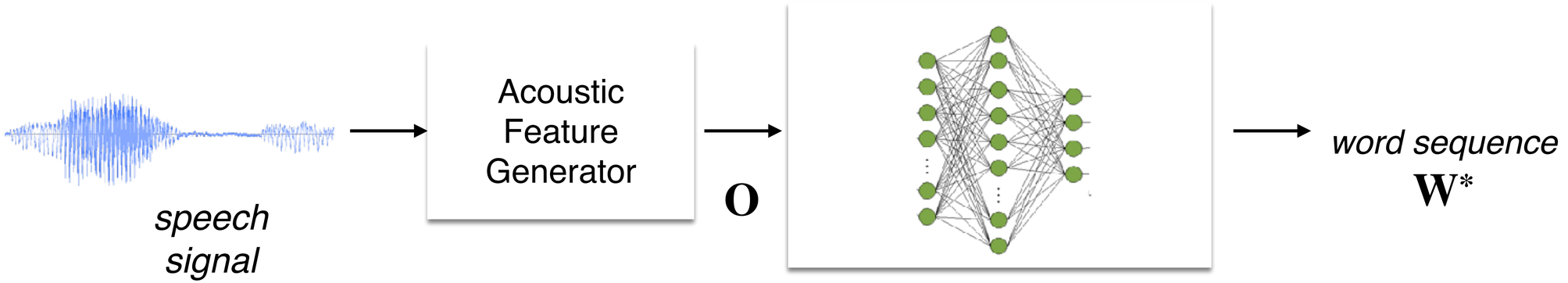
Lecture 8a



Instructor: Preethi Jyothi, IITB

Recap: Cascaded ASR ⇒ End-to-end ASR

$$\mathbf{W}^* = \arg\max_{W} \Pr(\mathbf{W} \mid \mathbf{O})$$



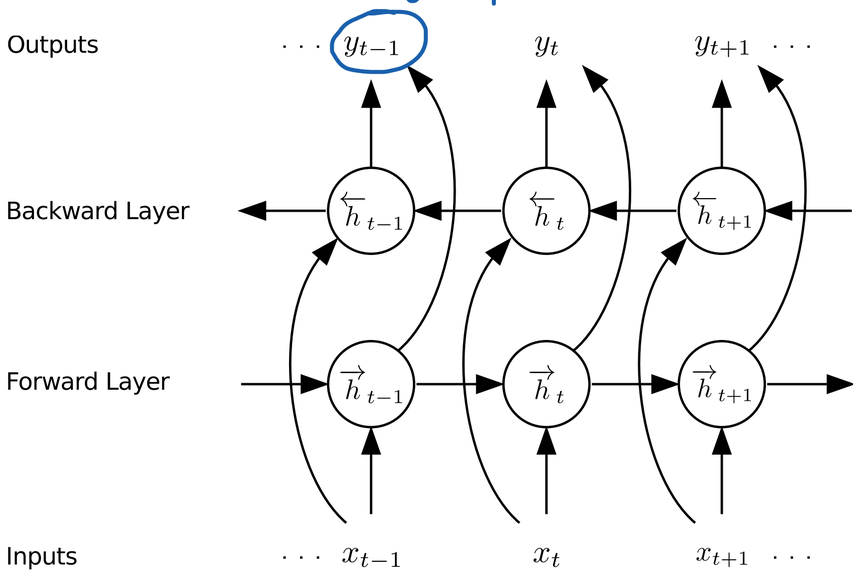
Single end-to-end model that directly learns a mapping from speech to text

1. Encoder-decoder models with attention

2. **CTC-based models**

Network Architecture

grapheme



forward

$$\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right)$$

$$\overleftarrow{h}_t = \mathcal{H}\left(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right)$$ backward

$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_o$$

$x_1 \ldots x_t \ldots x_T$

- Input: Acoustic feature vectors. Output: Characters
- Long Short-Term Memory (LSTM) units (with in-built memory cells) are used to implement $\mathcal{H}$ (in eqns above)
- Deep bidirectional LSTMs: Stack multiple bidirectional LSTM layers

Image from: Graves & Jaitley, Towards End-to-End Speech Recognition with Recurrent Neural Networks, ICML 14

Connectionist Temporal Classification (CTC)

- RNNs in ASR, if trained at the frame-level, will typically require alignments between the acoustics and the word sequence during training telling you which label (e.g. phone or character) should be output at each timestep.

Connectionist Temporal Classification (CTC)

- RNNs in ASR, if trained at the frame-level, will typically require alignments between the acoustics and the word sequence during training telling you which label (e.g. phone or character) should be output at each timestep.

- A new loss function, Connectionist Temporal Classification (CTC) tries to get around this.

Connectionist Temporal Classification (CTC)

- RNNs in ASR, if trained at the frame-level, will typically require alignments between the acoustics and the word sequence during training telling you which label (e.g. phone or character) should be output at each timestep.

- A new loss function, Connectionist Temporal Classification (CTC) tries to get around this.

- This is an objective function that allows RNN training without an explicit alignment step: CTC considers all possible alignments.

CTC Objective Function

sequence of chars

- CTC objective function is the probability of an output label sequence y given an utterance x → sequence of acoustic features

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

Σ alignment

[GJ14] Towards End-to-End Speech Recognition with Recurrent Neural Networks, ICML 14

CTC Objective Function

- CTC objective function is the probability of an output label sequence y given an utterance x

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

- Here, we sum over all possible alignments for y, enumerated by $B^{-1}(y)$

Is the underlying word BET or BEET?

BBBEEEETTT ← one possible alignment

i/p acoustic feature sequence of length 10 corresponding to the word BET

size 10

BBEEEEETTT

BBBBEETTTT

⋮

[GJ14] Towards End-to-End Speech Recognition with Recurrent Neural Networks, ICML 14

CTC: Mapping alignments to an output sequence

- Augment the output vocabulary with an additional “blank” (_) label

CTC: Mapping alignments to an output sequence

- Augment the output vocabulary with an additional “blank” (_) label
- For a given label sequence, there can be multiple alignments: (x, y, z) could correspond to (x, _, y, _, _, z) or (_, x, x, _, y, z)

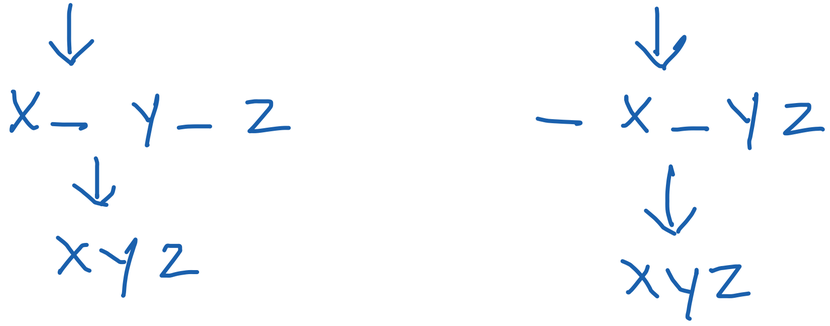
repeating chars have to be separated by a blank _

CTC: Mapping alignments to an output sequence

$B^{-1}(Y) = \{a_1, a_2, \ldots, a_M\}$ $\quad B(a_1) = B(a_2) = \ldots = B(a_M) = Y$

- Augment the output vocabulary with an additional “blank” (_) label
- For a given label sequence, there can be multiple alignments: (x, y, z) could correspond to (x, _, y, _, _, z) or (_, x, x, _, y, z)
- Define a 2-step operator B that reduces a label sequence by: first, removing repeating labels and second, removing blanks.
  B(“x, _, y, _, _, z”) = B(“_, x, x, _, y, z”) = “x, y, z”

an alignment



CTC Objective Function

- CTC objective function is the probability of an output label sequence y given an utterance x

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

- Here, we sum over all possible alignments for y, enumerated by $B^{-1}(y)$
- CTC assumes that $\Pr(a|x)$ can be computed as $\prod_{t=1}^{T} \Pr(a_t|x)$

$$\Pr(a|x) = \prod_{t} \Pr(a_t|x)$$

CTC Objective Function

- CTC objective function is the probability of an output label sequence y given an utterance x

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

- Here, we sum over all possible alignments for y, enumerated by $B^{-1}(y)$
- CTC assumes that $\Pr(a|x)$ can be computed as $\prod_{t=1}^{T} \Pr(a_t|x)$
  - i.e. CTC assumes that outputs at each time-step are conditionally independent given the input

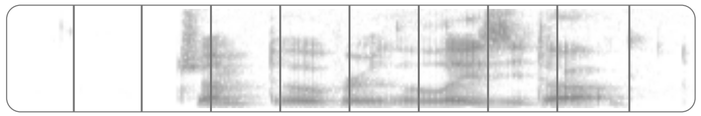
CTC Objective Function

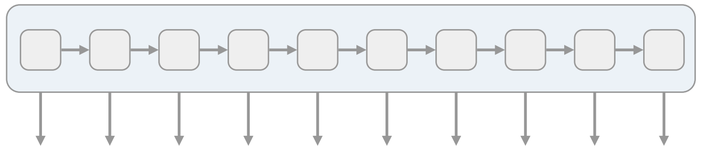
- CTC objective function is the probability of an output label sequence y given an utterance x

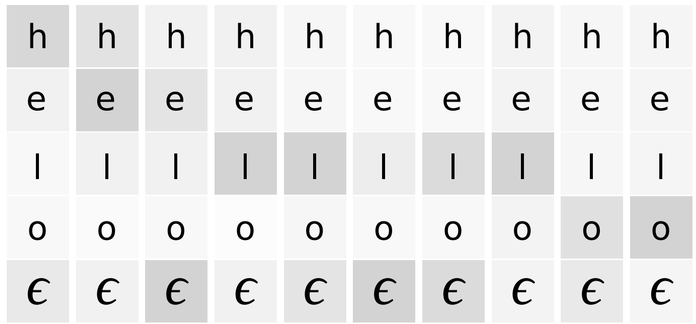
$$\mathrm{CTC}(x, y) = \Pr(y|x) = \sum_{a \in B^{-1}(y)} \Pr(a|x)$$

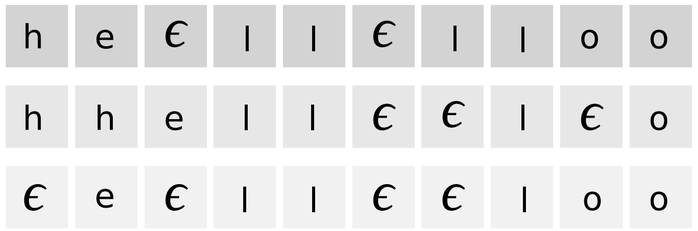
- Here, we sum over all possible alignments for y, enumerated by $B^{-1}(y)$
- CTC assumes that $\Pr(a|x)$ can be computed as $\prod_{t=1}^{T} \Pr(a_t|x)$
  - i.e. CTC assumes that outputs at each time-step are conditionally independent given the input
- Efficient dynamic programming algorithm to compute this loss function and its gradients

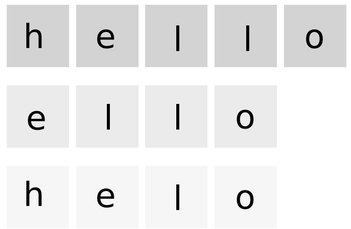
Connectionist Temporal Classification: Overview











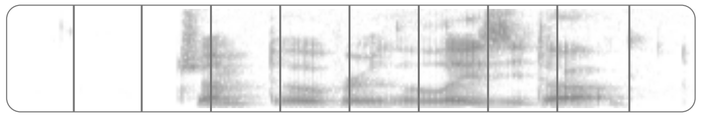
- CTC objective function is the probability of an output label sequence y given an utterance x (by summing over all possible alignments for y provided by $B^{-1}(y)$):

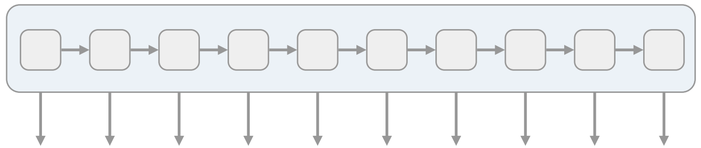
$$\mathrm{CTC}(x, y) = \Pr(y \mid x) = \sum_{a \in B^{-1}(y)} \Pr(a \mid x)$$

$$= \sum_{a \in B^{-1}(y)} \prod_{t=1}^{T} \Pr(a_t \mid x)$$

Image from: https://distill.pub/2017/ctc/

Connectionist Temporal Classification: Overview





h	h	h	h	h	h	h	h	h	h
e	e	e	e	e	e	e	e	e	e
l	l	l	l	l	l	l	l	l	l
o	o	o	o	o	o	o	o	o	o
ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ

h	e	ϵ	l	l	ϵ	l	l	o	o
h	h	e	l	l	ϵ	ϵ	l	ϵ	o
ϵ	e	ϵ	l	l	ϵ	ϵ	l	o	o

h	e	l	l	o
e	l	l	o	
h	e	l	o	

- CTC objective function is the probability of an output label sequence y given an utterance x (by summing over all possible alignments for y provided by $B^{-1}(y)$):

$$\mathrm{CTC}(x, y) = \Pr(y \mid x) = \sum_{a \in B^{-1}(y)} \Pr(a \mid x)$$

$$= \sum_{a \in B^{-1}(y)} \prod_{t=1}^{T} \Pr(a_t \mid x)$$

- Efficient forward+backward algorithm to compute this loss function and its gradients [GJ14]

[GJ14] Towards End-to-End Speech Recognition with Recurrent Neural Networks, ICML 14

Image from: https://distill.pub/2017/ctc/

Illustration: Forward Algorithm to compute $\alpha_t(j)$

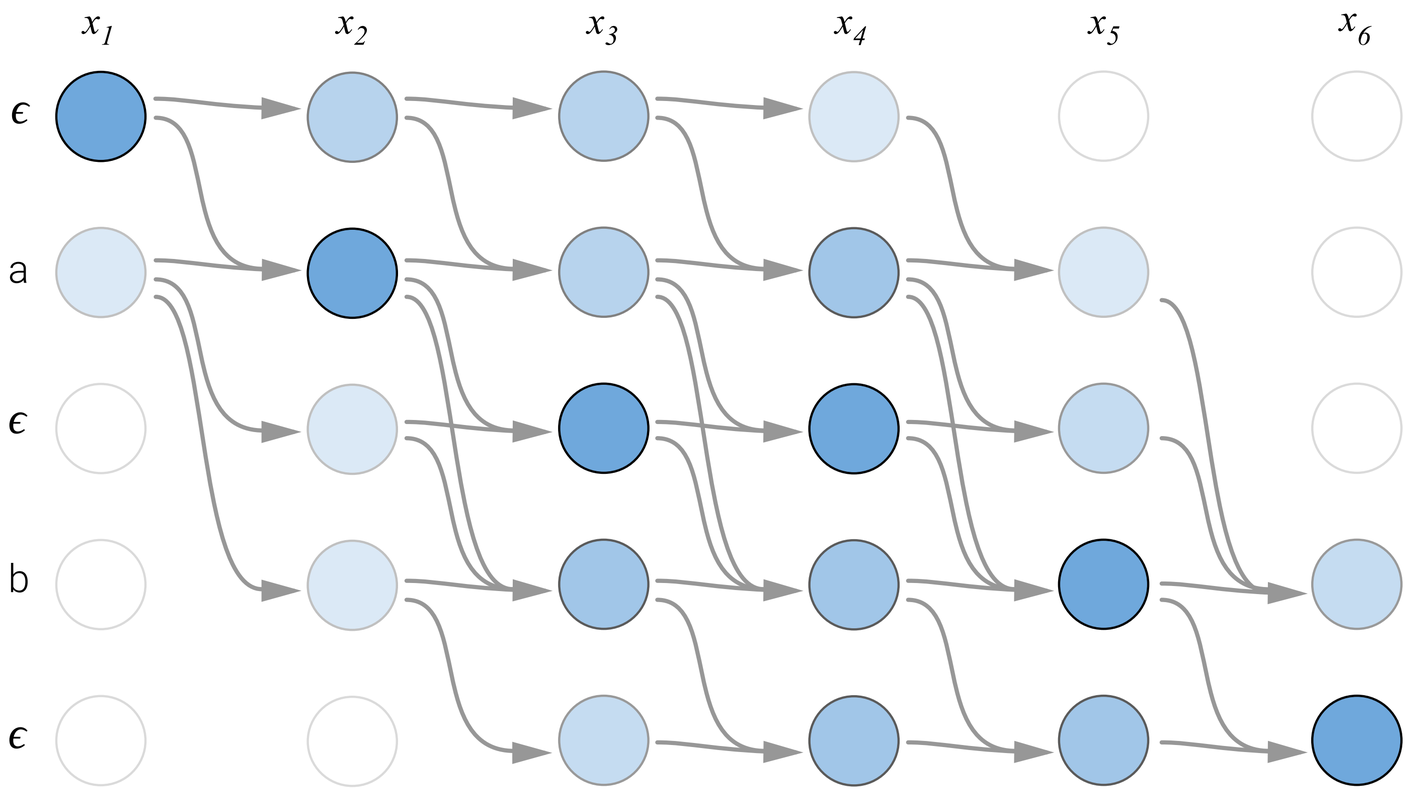


Image from: https://distill.pub/2017/ctc/

Illustration: Forward Algorithm to compute $\alpha_t(j)$

= Prob of $Y_{1:j}$ at time t

forward probability $\alpha_t(j)$

$$\alpha_t(j) = \sum_{i=j-2}^{j} \alpha_{t-1}(i) a_{ij} b_t(y'_j)$$

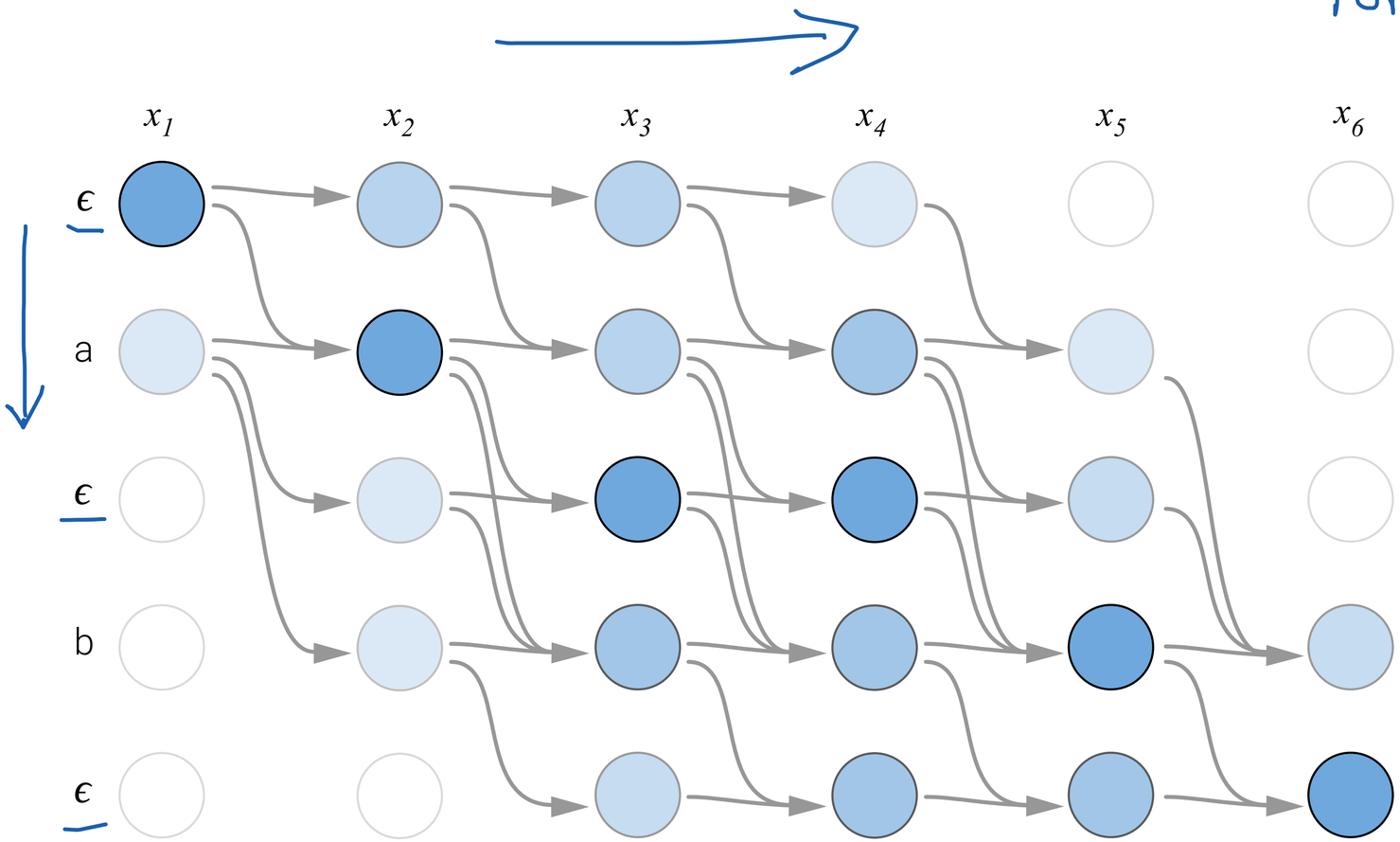
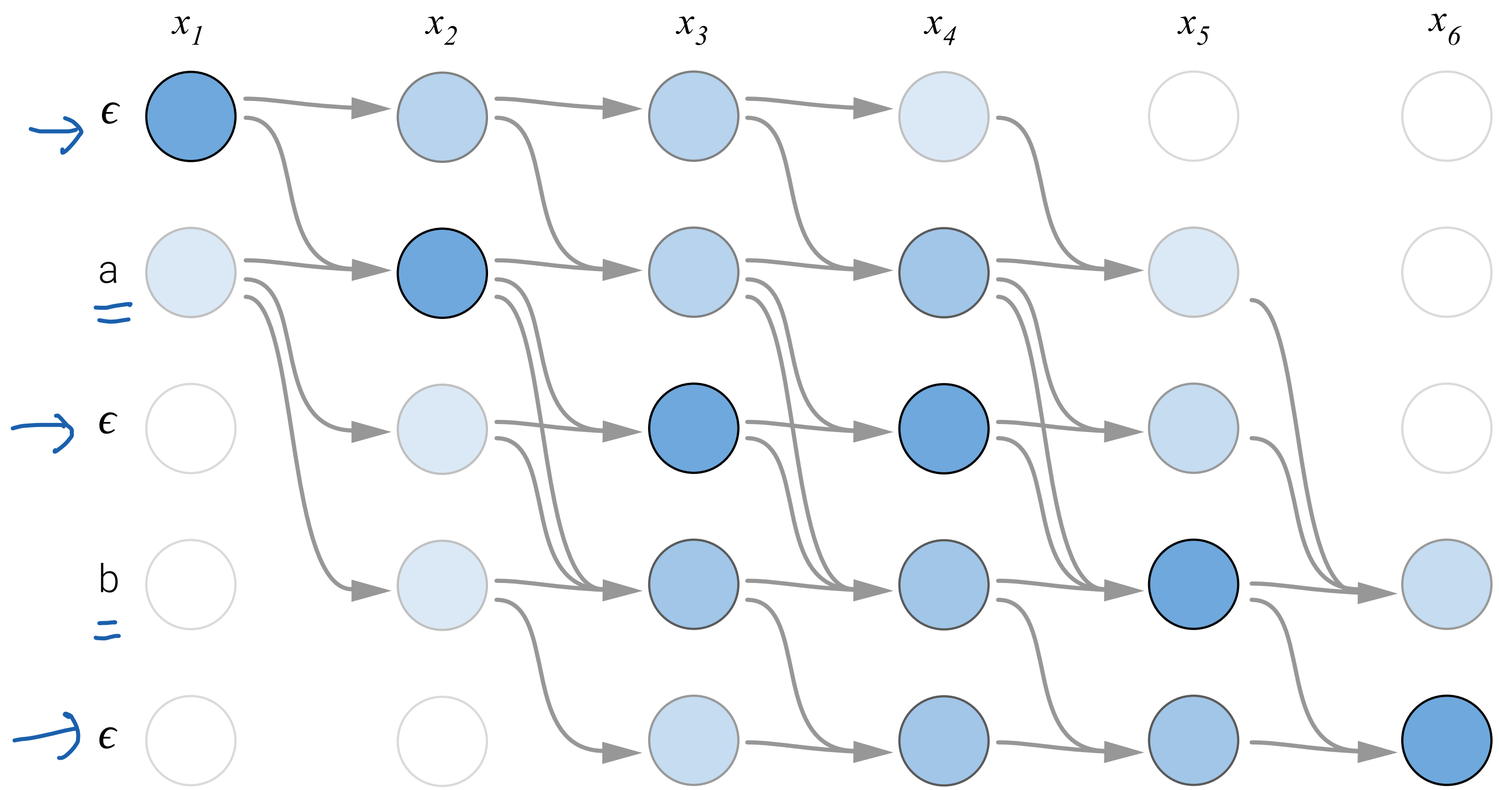


Image from: https://distill.pub/2017/ctc/

Illustration: Forward Algorithm to compute $\alpha_t(j)$



$$\alpha_t(j) = \sum_{i=j-2}^{j} \alpha_{t-1}(i) a_{ij} b_t(y'_j)$$

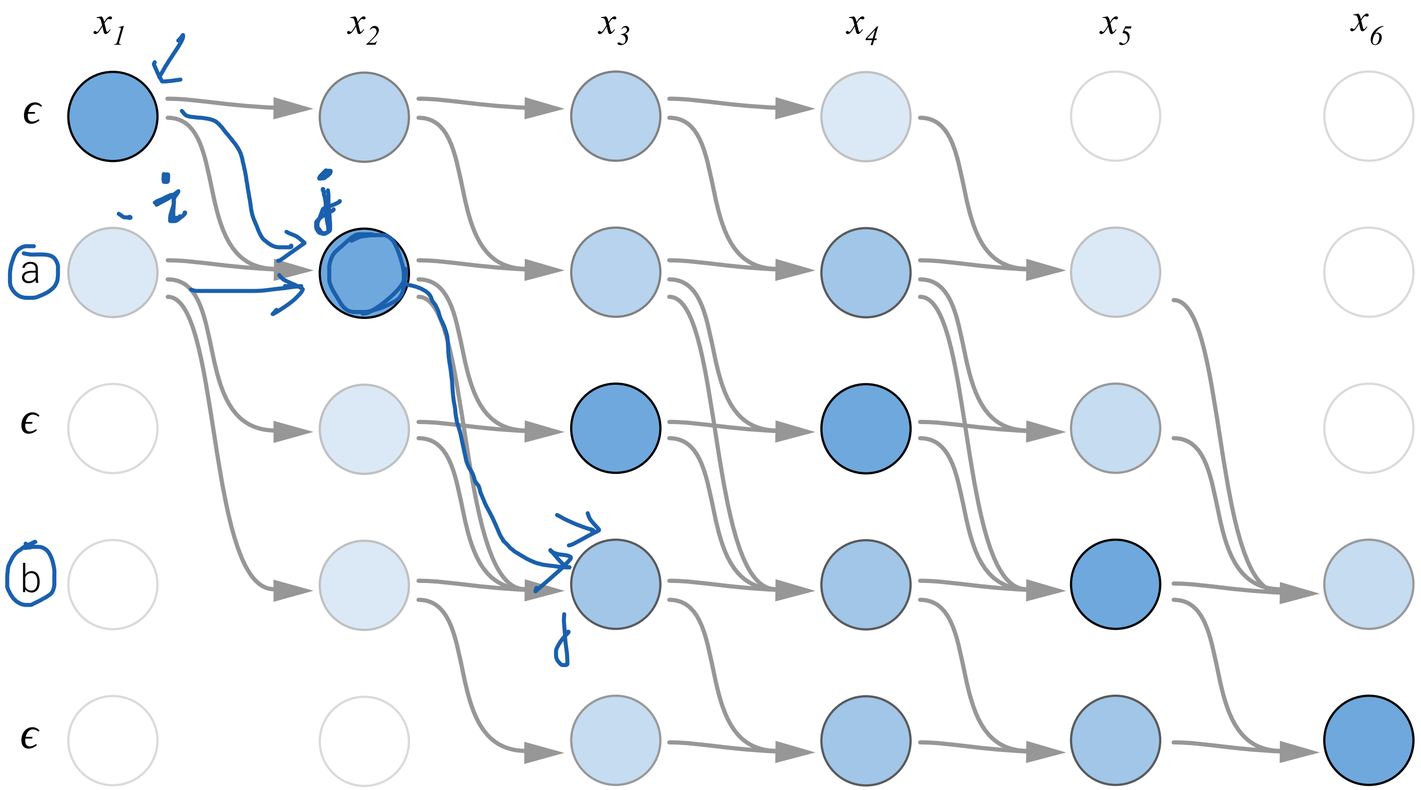
where

$b_t(y'_j)$ is the probability given by NN to the symbol y'_j for $t = 1 \ldots T$, when $|x| = T$

$$y'_j = \begin{cases} y_{j/2} \text{ if } j \text{ is even} \\ \epsilon \text{ otherwise} \end{cases} \quad (j = 1 \ldots 2l+1 \textbf{ when } |y| = l)$$

Image from: https://distill.pub/2017/ctc/

Illustration: Forward Algorithm to compute $\alpha_t(j)$



$$\alpha_t(j) = \sum_{i=j-2}^{j} \alpha_{t-1}(i) a_{ij} b_t(y'_j)$$

where

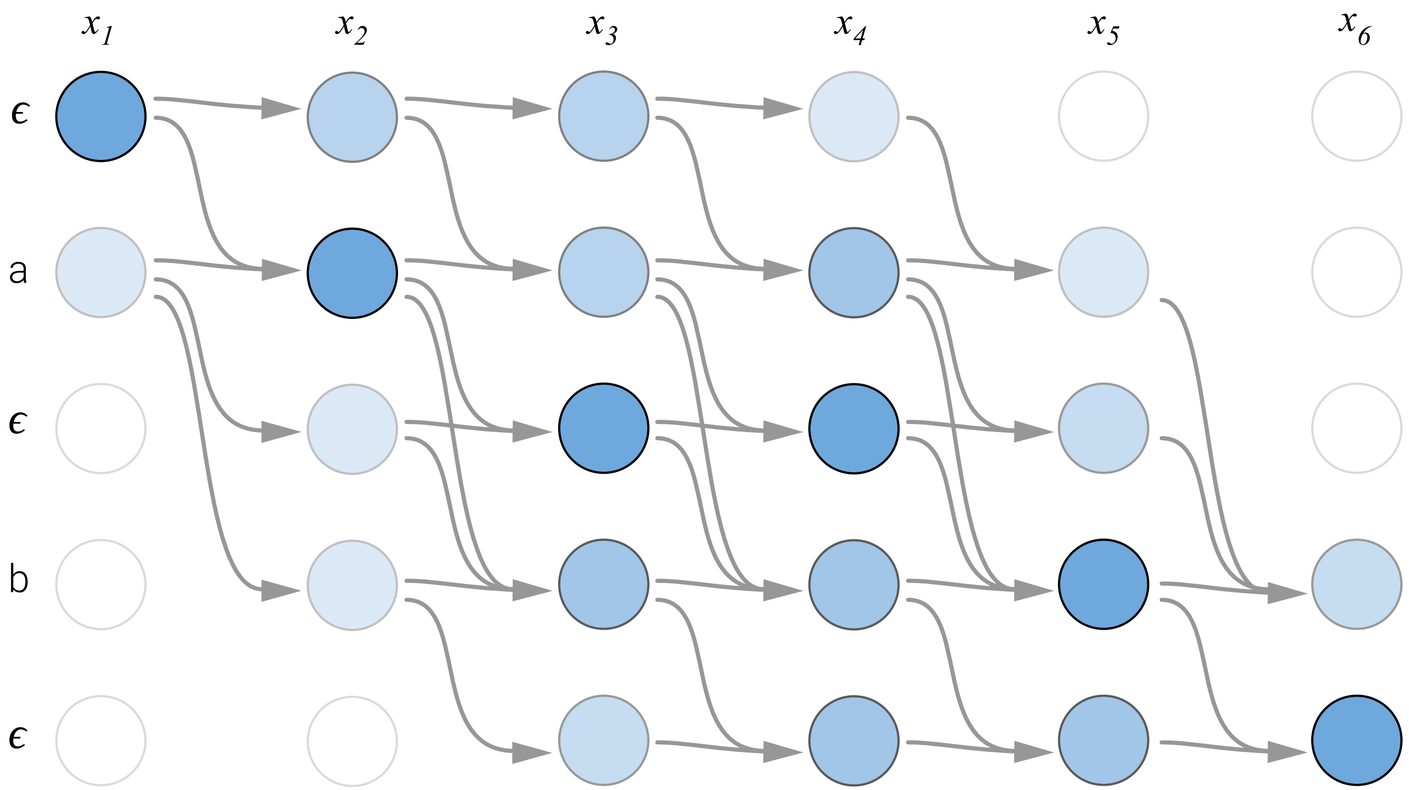
$b_t(y'_j)$ is the probability given by NN to the symbol y'_j for $t = 1 \ldots T$, when $|x| = T$

$$y'_j = \begin{cases} y_{j/2} \text{ if } j \text{ is even} \\ \epsilon \text{ otherwise} \end{cases} \quad (j = 1 \ldots 2l+1 \textbf{ when } |y| = l)$$

$$a_{ij} = \begin{cases} 1 \textbf{ if } i = j \textbf{ or } i = j-1 \\ 1 \textbf{ if } i = j-2 \textbf{ and } y'_j \neq y'_{j-2} \\ 0 \textbf{ otherwise} \end{cases}$$

Image from: https://distill.pub/2017/ctc/

Illustration: Forward Algorithm to compute $\alpha_t(j)$



$$\alpha_t(j) = \sum_{i=j-2}^{j} \alpha_{t-1}(i) a_{ij} b_t(y'_j)$$

where

$b_t(y'_j)$ is the probability given by NN to the symbol y'_j for $t = 1 \ldots T$, when $|x| = T$

$$y'_j = \begin{cases} y_{j/2} \text{ if } j \text{ is even} \\ \epsilon \text{ otherwise} \end{cases} \quad (j = 1 \ldots 2l+1 \textbf{ when } |y| = l)$$

$$a_{ij} = \begin{cases} 1 \textbf{ if } i = j \textbf{ or } i = j-1 \\ 1 \textbf{ if } i = j-2 \textbf{ and } y'_j \neq y'_{j-2} \\ 0 \textbf{ otherwise} \end{cases}$$

$$CTC(x, y) = \sum_{a \in B^{-1}(y)} \Pr(a \,|\, x) = \alpha_T(2l) + \alpha_T(2l+1)$$

$\Pr(y|x)$

Image from: https://distill.pub/2017/ctc/

Decoding

Greedy decoding

- Pick the single most probable output at every time step

$$\arg\max_{y} \Pr(y|x) \approx B(\arg\max_{a} Pr(a|x))$$

Decoding

- Pick the single most probable output at every time step

$$\arg\max_{y} \Pr(y|x) \approx B(\arg\max_{a} Pr(a|x))$$

- Use a beam search algorithm to integrate a dictionary and a language model

Experimental Results

Table 1. **Wall Street Journal Results.** All scores are word error rate/character error rate (where known) on the evaluation set. 'LM' is the Language model used for decoding. '14 Hr' and '81 Hr' refer to the amount of data used for training.

SYSTEM	LM	14 HR	81 HR
RNN-CTC	NONE	74.2/30.9	30.1/9.2
RNN-CTC	DICTIONARY	69.2/30.0	24.0/8.0
RNN-CTC	MONOGRAM	25.8	15.8
RNN-CTC	BIGRAM	15.5	10.4
RNN-CTC	TRIGRAM	13.5	8.7
BASELINE	NONE	—	—
BASELINE	DICTIONARY	56.1	51.1
BASELINE	MONOGRAM	23.4	19.9
BASELINE	BIGRAM	11.6	9.4
BASELINE	TRIGRAM	9.4	7.8
COMBINATION	TRIGRAM	—	6.7

Table from: Graves & Jaitley, Towards End-to-End Speech Recognition with Recurrent Neural Networks, ICML 14

Sample Character-level Transcripts

target: *TO ILLUSTRATE THE POINT A PROMINENT MIDDLE EAST ANALYST IN WASHINGTON RECOUNTS A CALL FROM ONE CAMPAIGN*
output: *TWO ALSTRAIT THE POINT A PROMINENT MIDILLE EAST ANALYST IM WASHINGTON RECOUNCACALL FROM ONE CAMPAIGN*

target: *T. W. A. ALSO PLANS TO HANG ITS BOUTIQUE SHINGLE IN AIRPORTS AT LAMBERT SAINT*
output: *T. W. A. ALSO PLANS TOHING ITS BOOTIK SINGLE IN AIRPORTS AT LAMBERT SAINT*

target: *ALL THE EQUITY RAISING IN MILAN GAVE THAT STOCK MARKET INDIGESTION LAST YEAR*
output: *ALL THE EQUITY RAISING IN MULONG GAVE THAT STACRK MARKET IN TO JUSTIAN LAST YEAR*

target: *THERE'S UNREST BUT WE'RE NOT GOING TO LOSE THEM TO DUKAKIS*
output: *THERE'S UNREST BUT WERE NOT GOING TO LOSE THEM TO DEKAKIS*

Another end-to-end system

- Decoding is at the word level if we use a dictionary+LM. Out-of-vocabulary (OOV) words cannot be handled.

Maas et al., “Lexicon Free Conversational Speech Recognition with Neural Networks”, NAACL 15

Another end-to-end system

- Decoding is at the word level if we use a dictionary+LM. Out-of-vocabulary (OOV) words cannot be handled.
- Build a system that is trained and decoded entirely at the character-level [Maas et al.]
  - This would enable the transcription of OOV words, disfluencies, etc.

Maas et al., “Lexicon Free Conversational Speech Recognition with Neural Networks”, NAACL 15

Another end-to-end system

- Decoding is at the word level if we use a dictionary+LM. Out-of-vocabulary (OOV) words cannot be handled.
- Build a system that is trained and decoded entirely at the character-level [Maas et al.]
  - This would enable the transcription of OOV words, disfluencies, etc.
- Shows results on the Switchboard task. Matches a GMM-HMM baseline system but underperforms compared to an HMM-DNN baseline.

Maas et al., “Lexicon Free Conversational Speech Recognition with Neural Networks”, NAACL 15

Model Specifics

- Approach consists of two neural models:
  - A deep bidirectional RNN (DBRNN) mapping acoustic features to character sequences (Trained using CTC.)
  - A neural network character language model

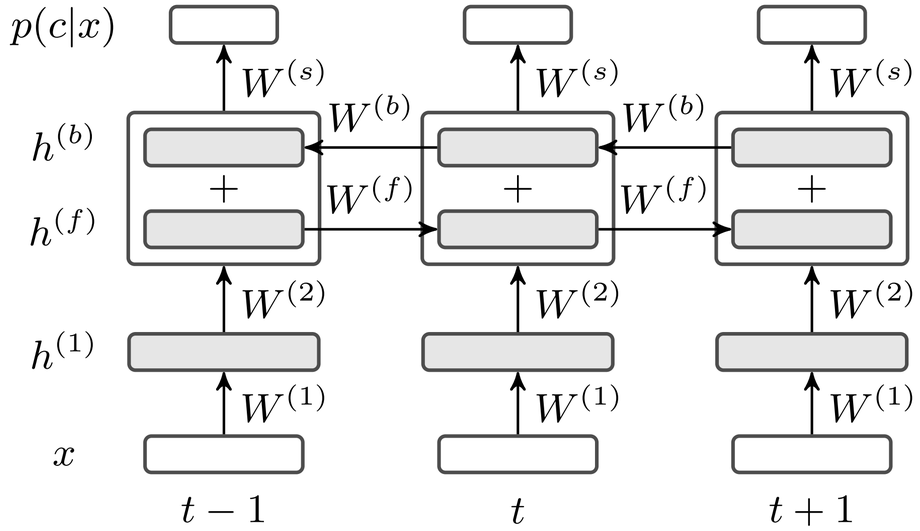


Image from Maas et al., “Lexicon Free Conversational Speech Recognition with Neural Networks”, NAACL 15

Decoding

- Simplest form: Decode without any language model

Decoding

- Simplest form: Decode without any language model
- Beam Search decoding:
  - Combine DBRNN outputs with a char-level language model
  - Char-level language model applied at every time step (unlike word models)
  - Circumvents the issue of handling OOV words during decoding

Sample Test Utterances

#	Method	Transcription
(1)	Truth	yeah i went into the i do not know what you think of *fidelity* but
	HMM-GMM	yeah when the i don't know what you think of fidel it even them
	CTC+CLM	yeah i went to i don't know what you think of fidelity but um
(2)	Truth	no no speaking of weather do you carry a altimeter slash *barometer*
	HMM-GMM	no i'm not all being the weather do you uh carry a uh helped emitters last brahms her
	CTC+CLM	no no beating of whether do you uh carry a uh a time or less barometer
(3)	Truth	i would ima- well yeah it is i know you are able to stay home with them
	HMM-GMM	i would amount well yeah it is i know um you're able to stay home with them
	CTC+CLM	i would ima- well yeah it is i know uh you're able to stay home with them

Image from Maas et al., "Lexicon Free Conversational Speech Recognition with Neural Networks", NAACL 15

Analysing character probabilities

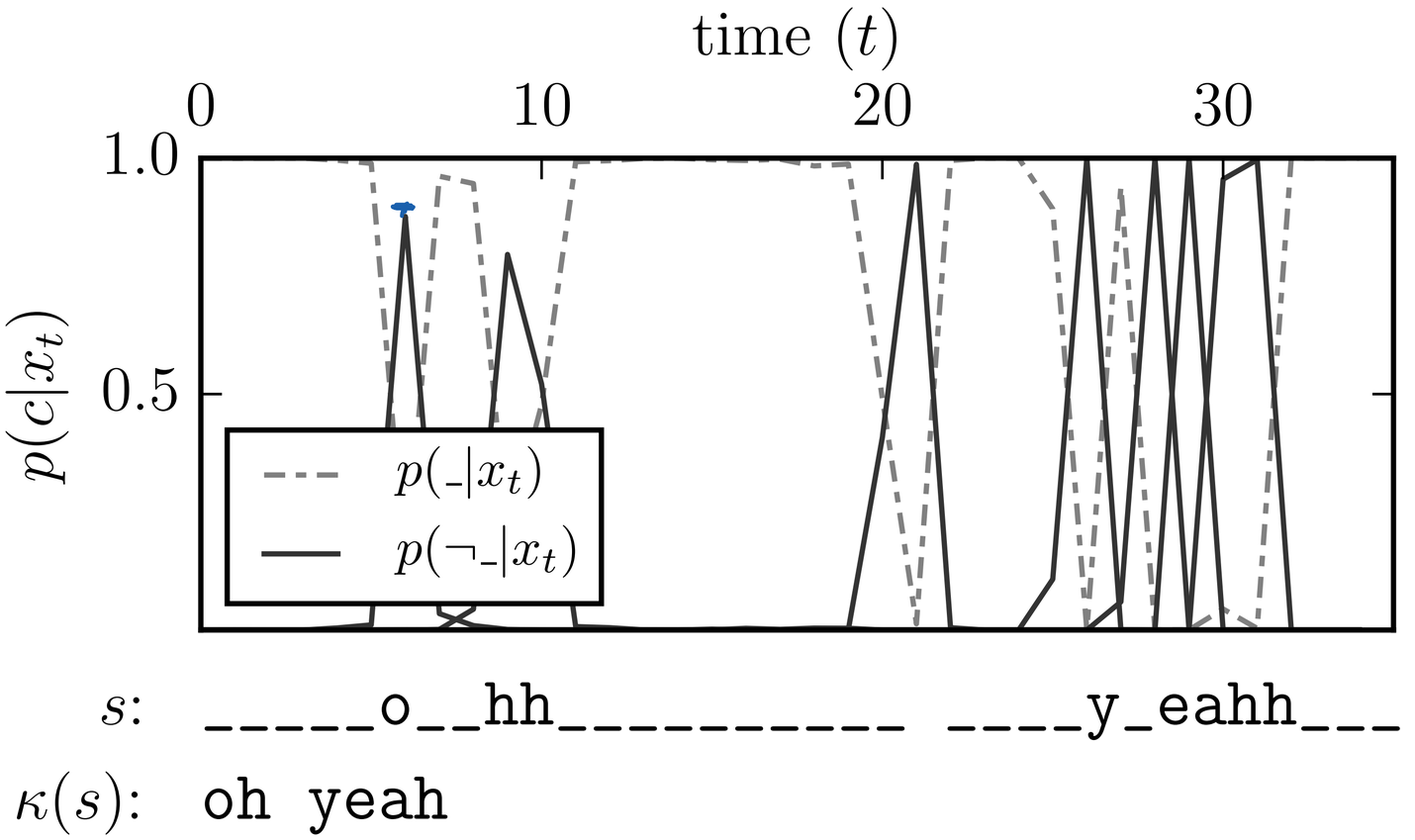
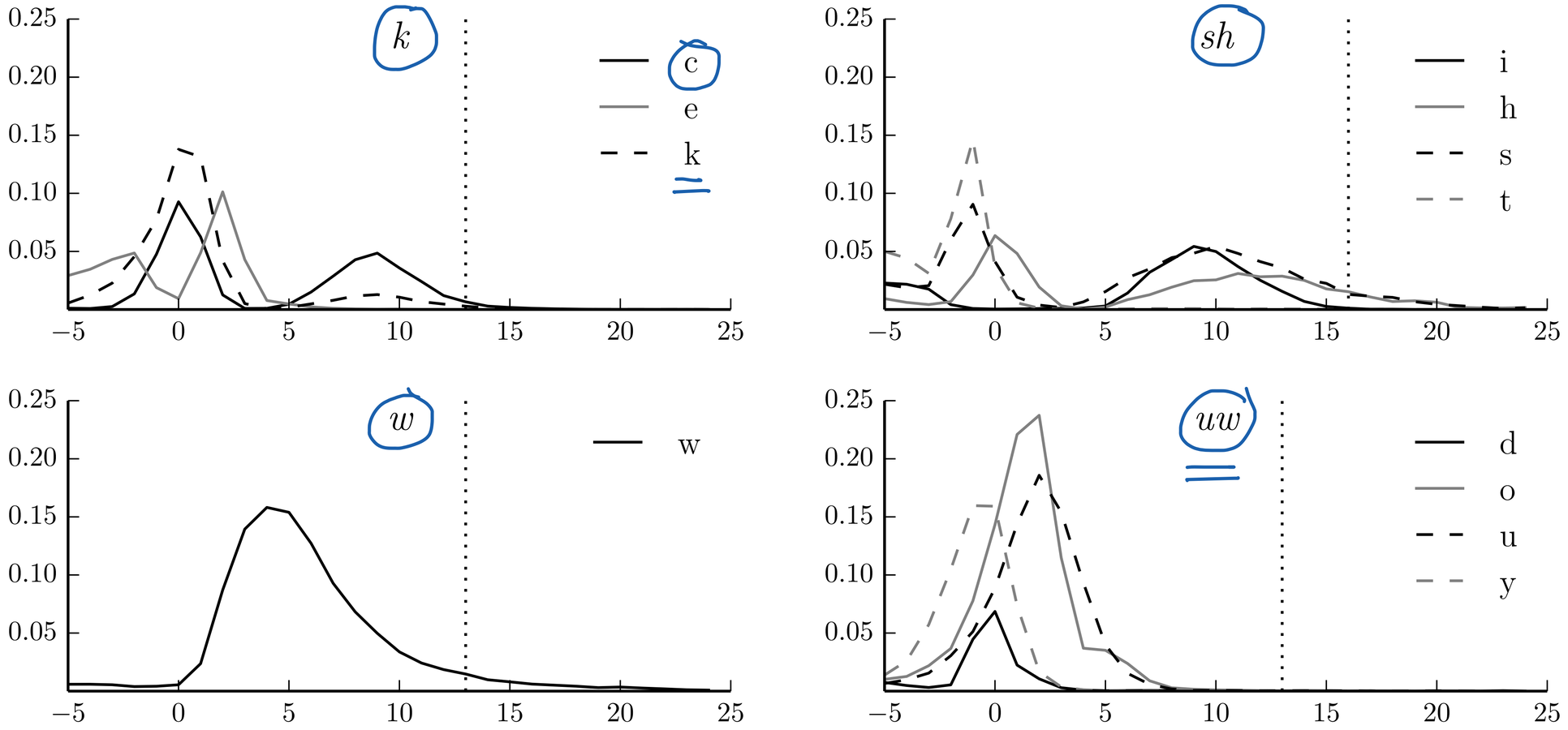


Image from Maas et al., “Lexicon Free Conversational Speech Recognition with Neural Networks”, NAACL 15

Character probabilities for various monophones



boon
put

Image from Maas et al., “Lexicon Free Conversational Speech Recognition with Neural Networks”, NAACL 15

Hybrid CTC/Attention-based End-to-End ASR

- The objective to be maximised is a linear combination of the CTC and attention objective functions.

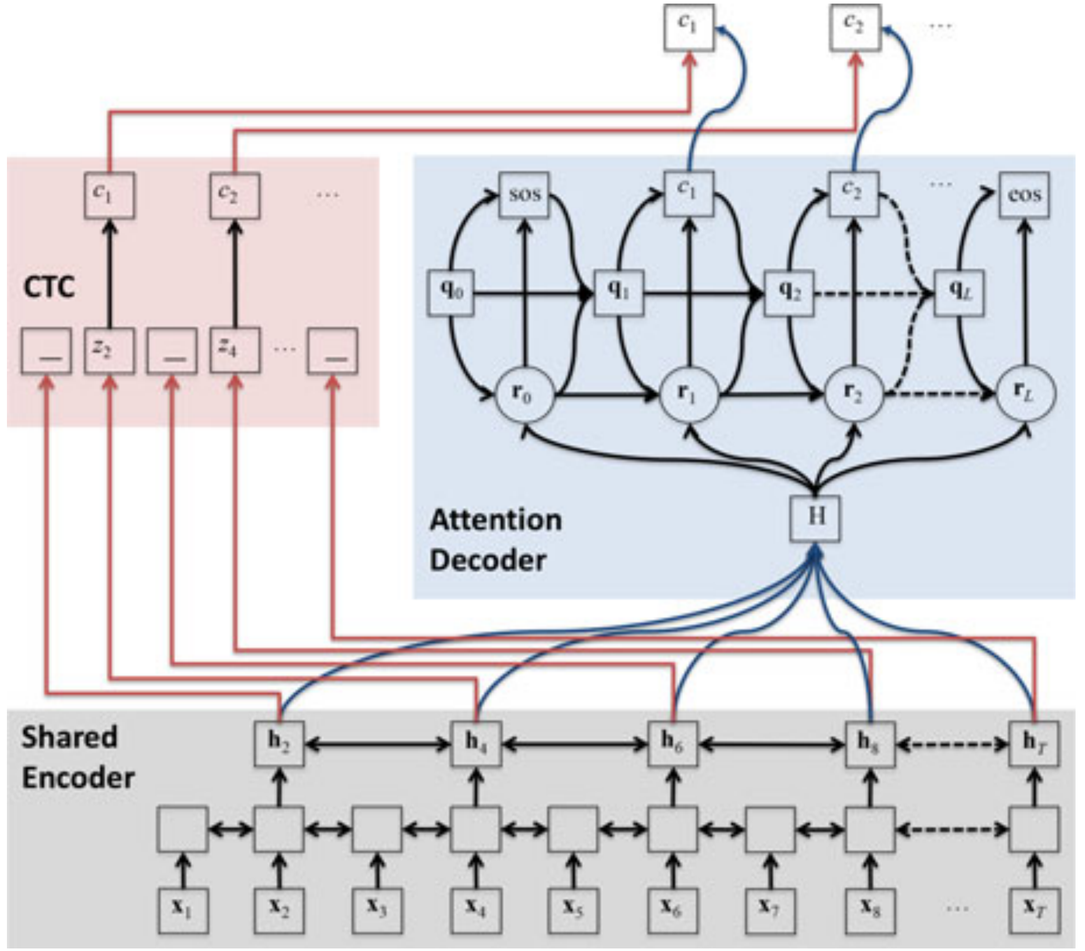


Image from: Watanabe et al.,Hybrid CTC/Attention Architecture for End-to-end Speech Recognition, 2017

Hybrid CTC/Attention-based End-to-End ASR

- The objective to be maximised is a linear combination of the CTC and attention objective functions.

	Task1	Task2	Task3
Cascaded GMM/ HMM Model	11.2	9.2	12.1
Cascaded HMM/ DNN Model	9.0	7.2	9.6
Hybrid CTC/ Attention	8.4	6.1	6.9

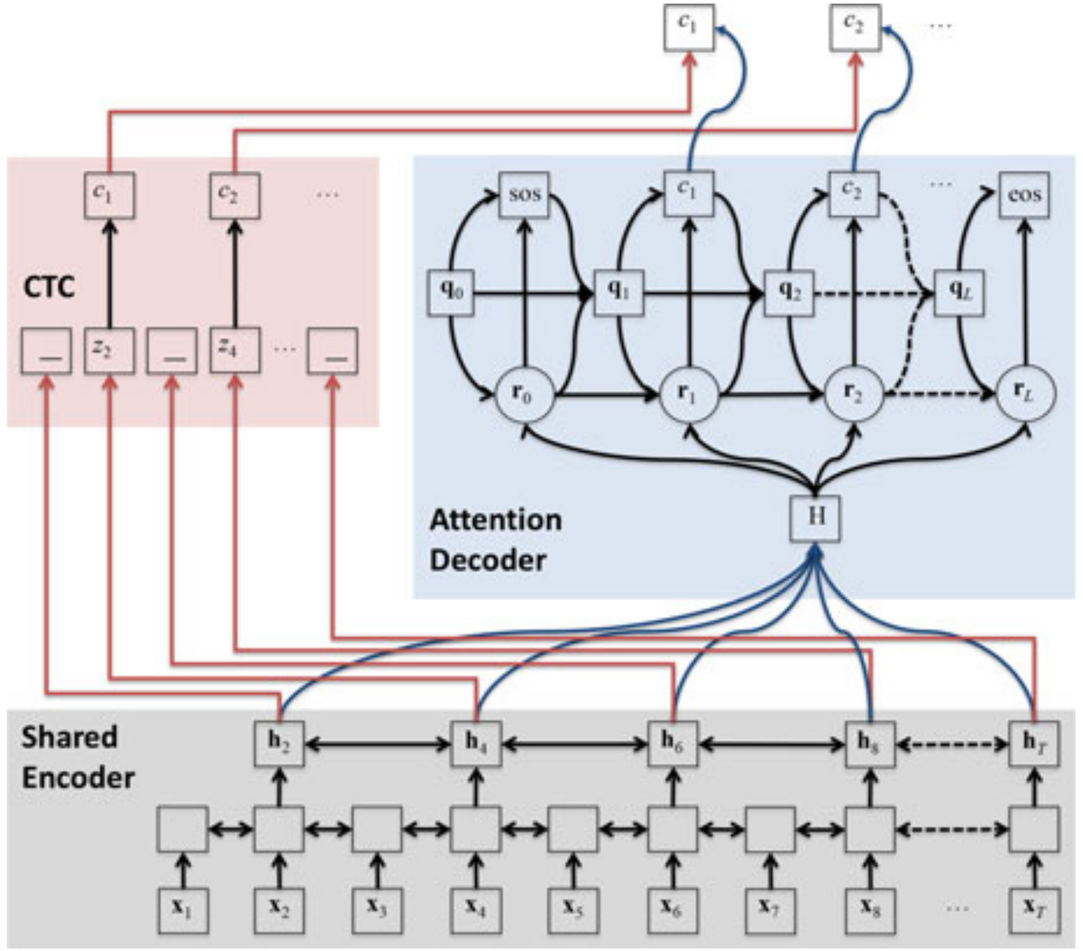


Image from: Watanabe et al.,Hybrid CTC/Attention Architecture for End-to-end Speech Recognition, 2017