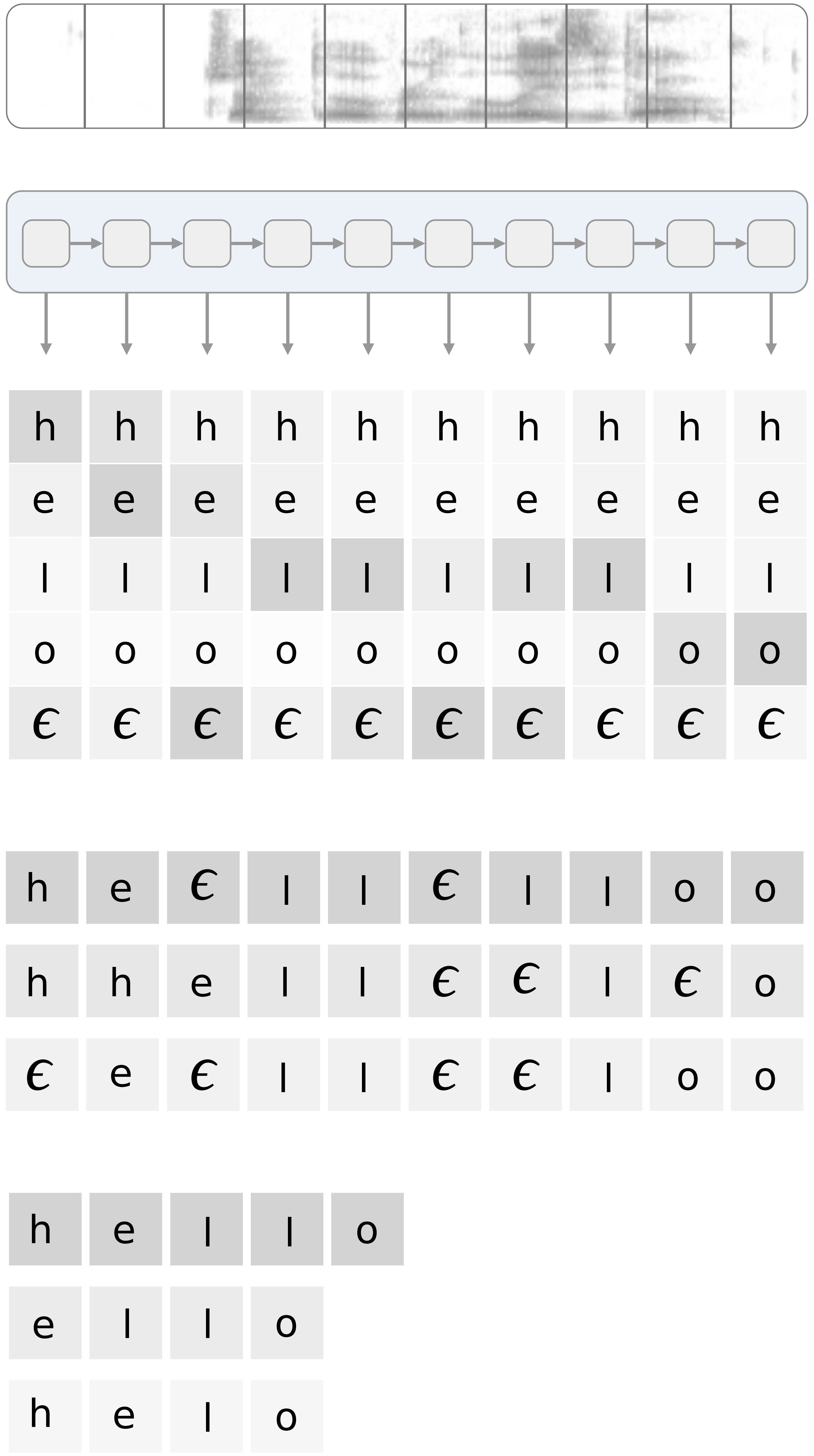
End-to-end Neural Architectures For ASR

Lecture 15

CS 753

Instructor: Preethi Jyothi

Connectionist Temporal Classification (CTC): Recap



- 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 $B^{-1}(y)$):

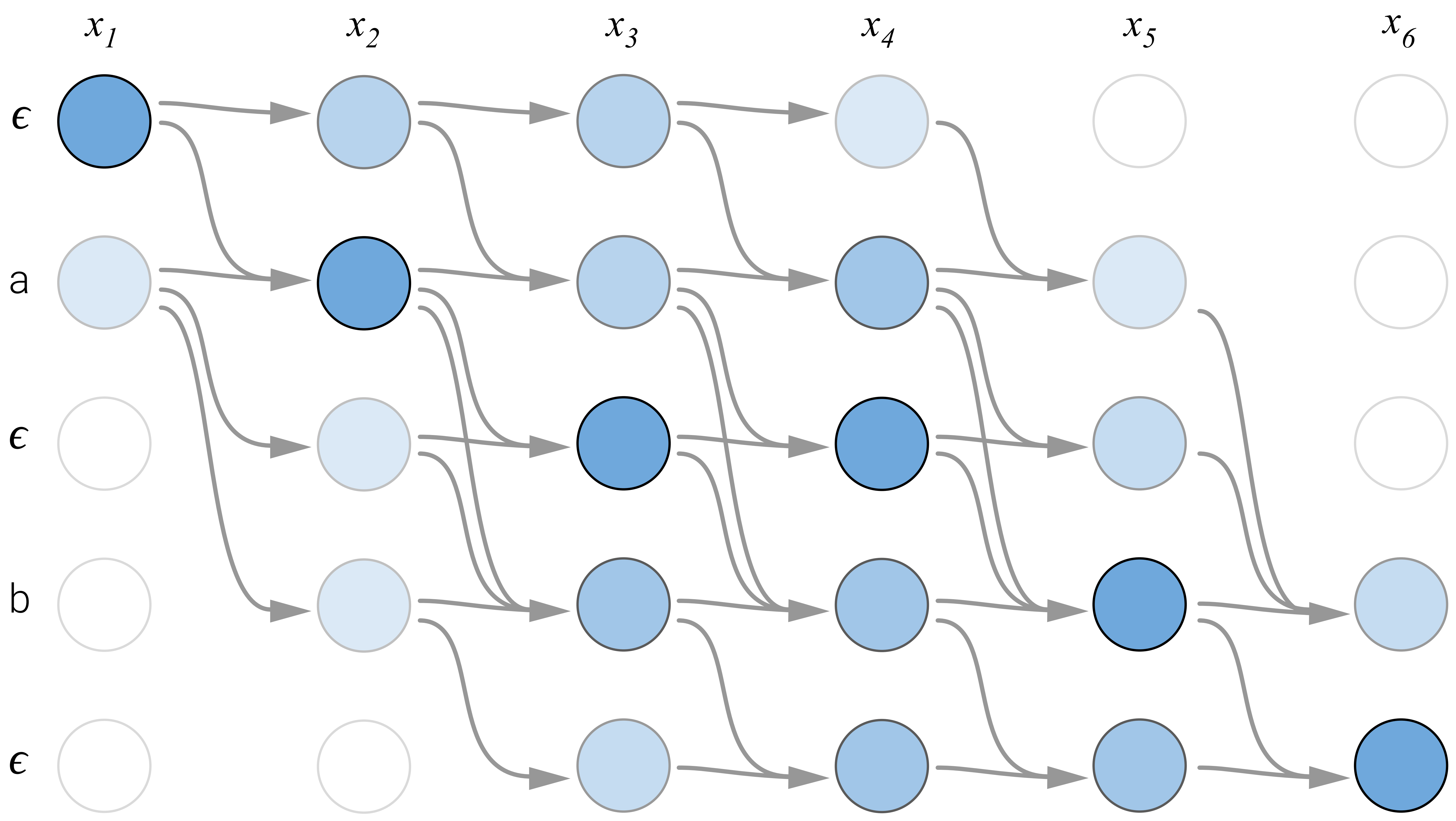
$$CTC(x, y) = \Pr(y \mid x) = \sum_{a \in B^{-1}(y)} \Pr(a \mid x)$$

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

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

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** $|x| = T$

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

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

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

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

CTC vs. LAS

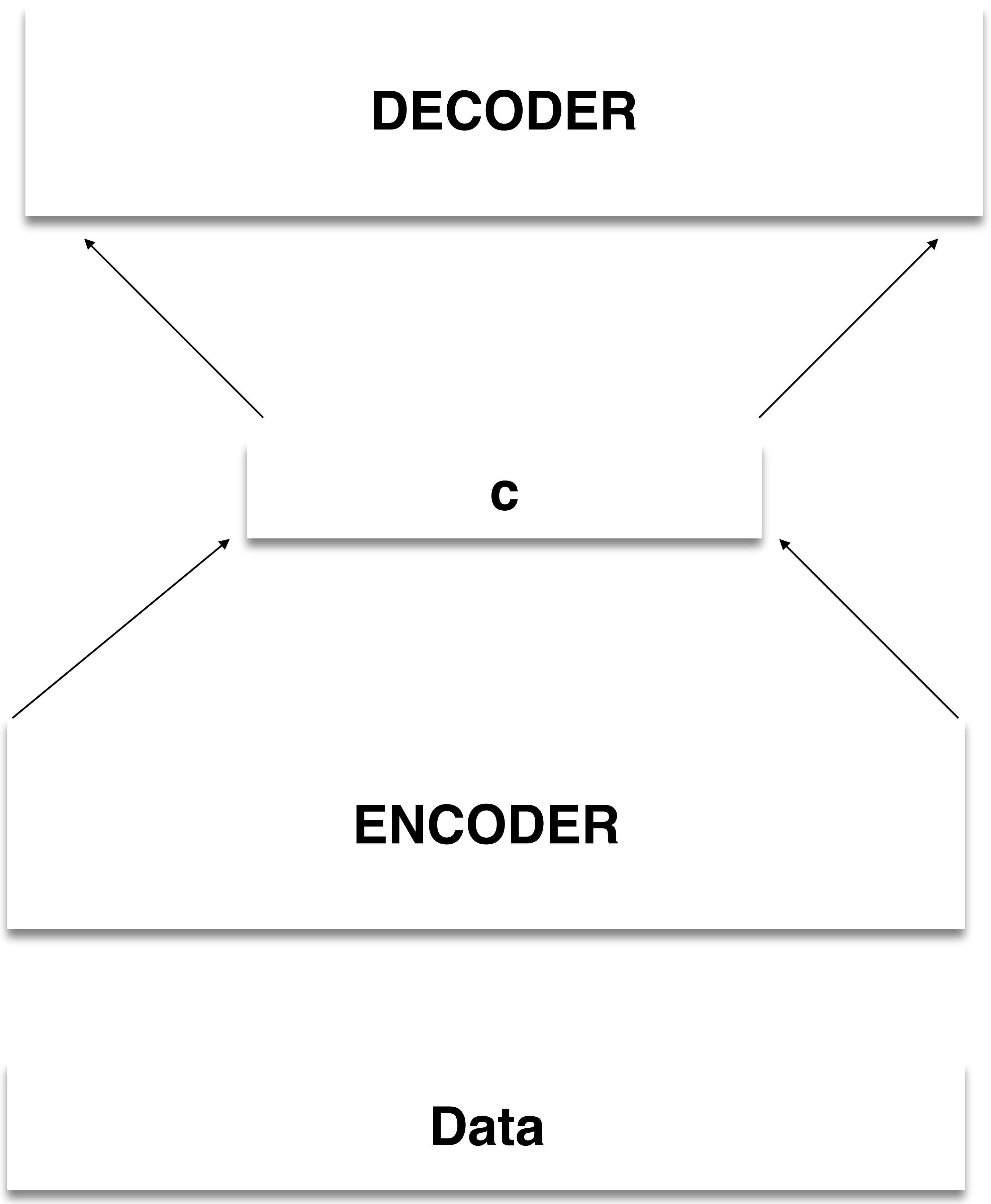
- Works well for end-to-end ASR systems
- CTC makes an assumption that the network outputs at different time steps are conditionally independent given the inputs
- The Listen, Attend and Spell [LAS] network makes no independence assumptions about the probability distribution of the output sequences given the input

$$P(\mathbf{y}|\mathbf{x}) = \prod_i P(y_i|\mathbf{x}, y_{<i})$$

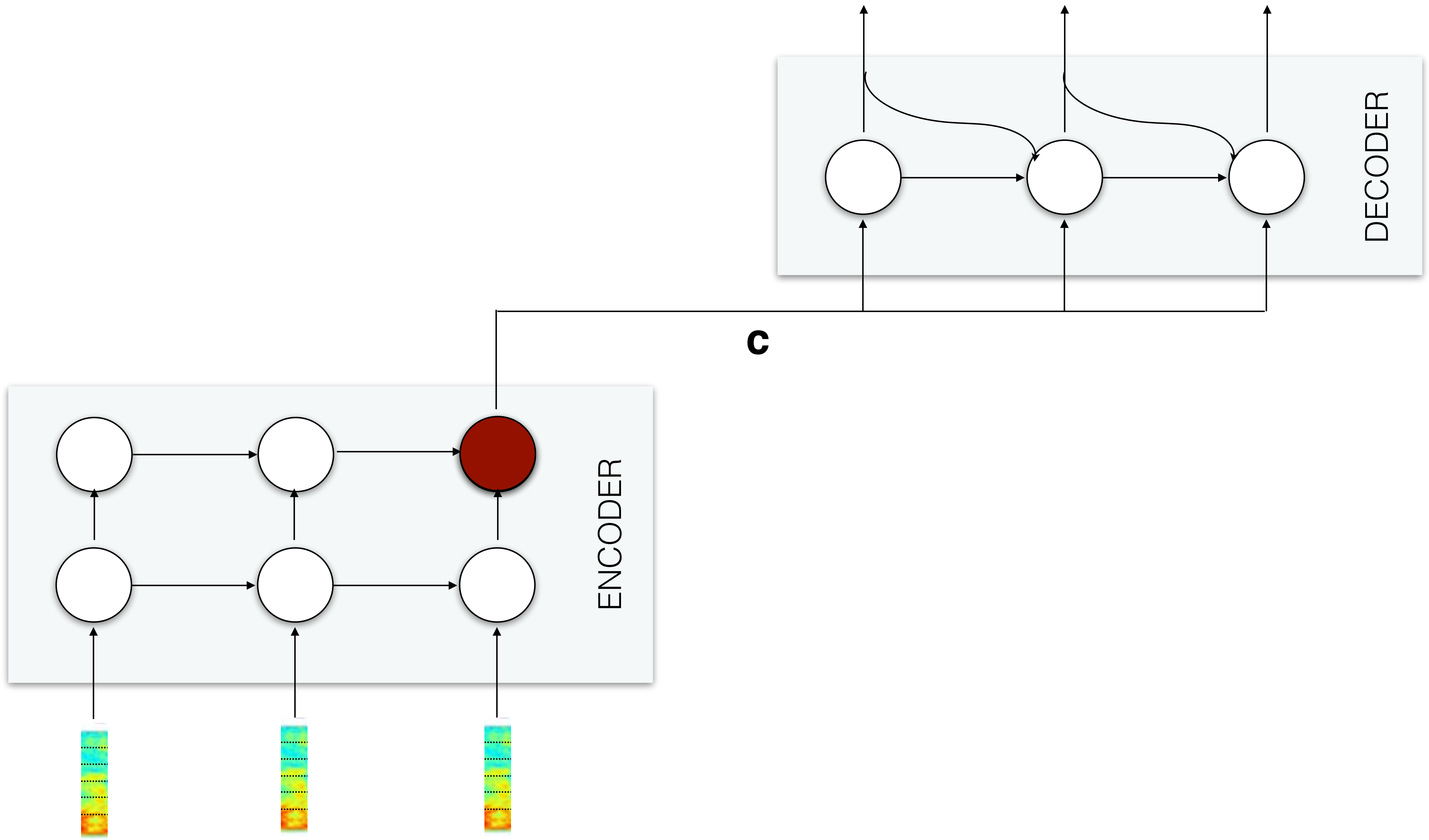
- Based on the sequence-to-sequence with attention framework

[LAS]: Chan et al., Listen, Attend and Spell: A NN for LVCSR, ICASSP 2016

Sequence to sequence models
Encoder-decoder architecture

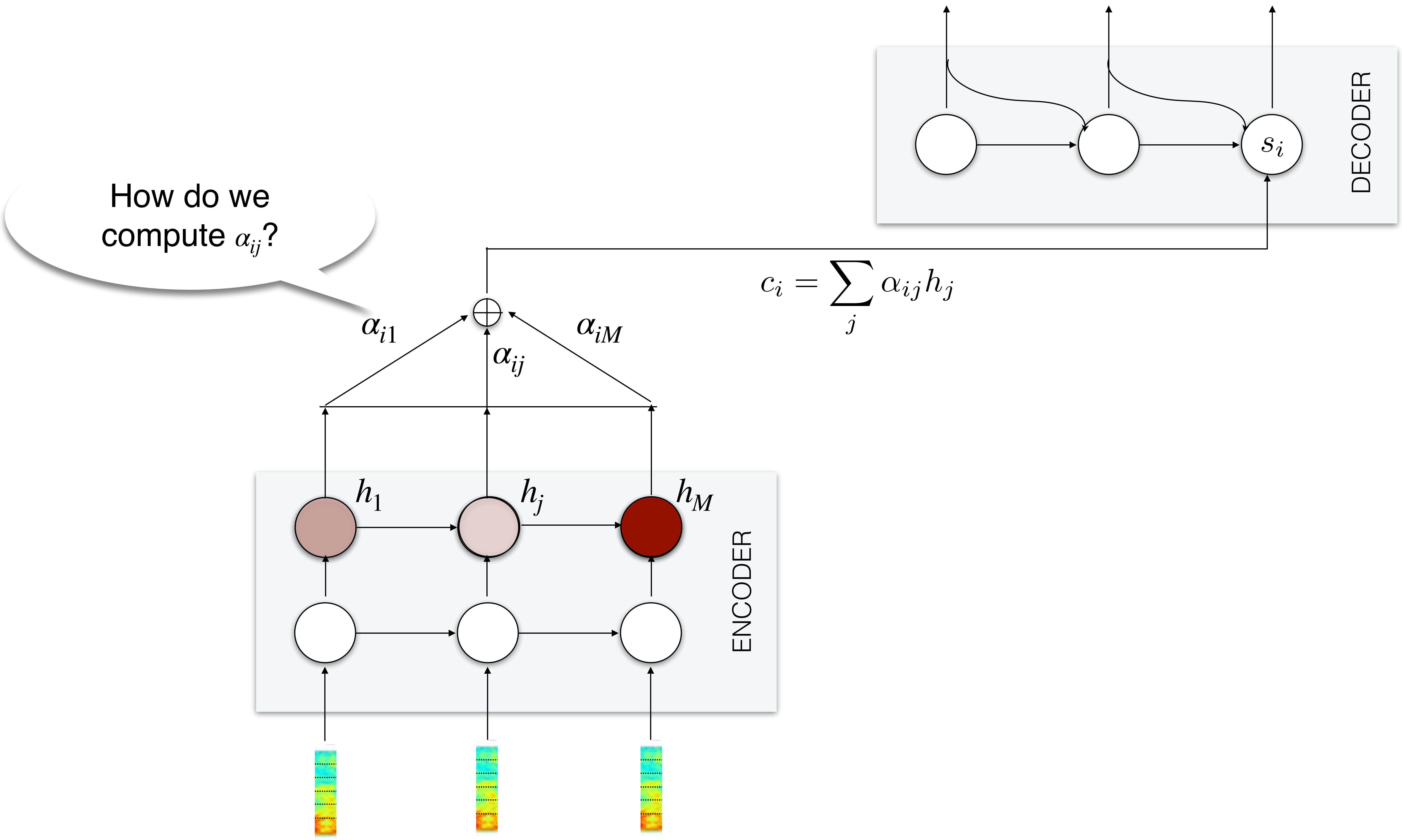


Sequence to sequence models
Encoder-decoder architecture



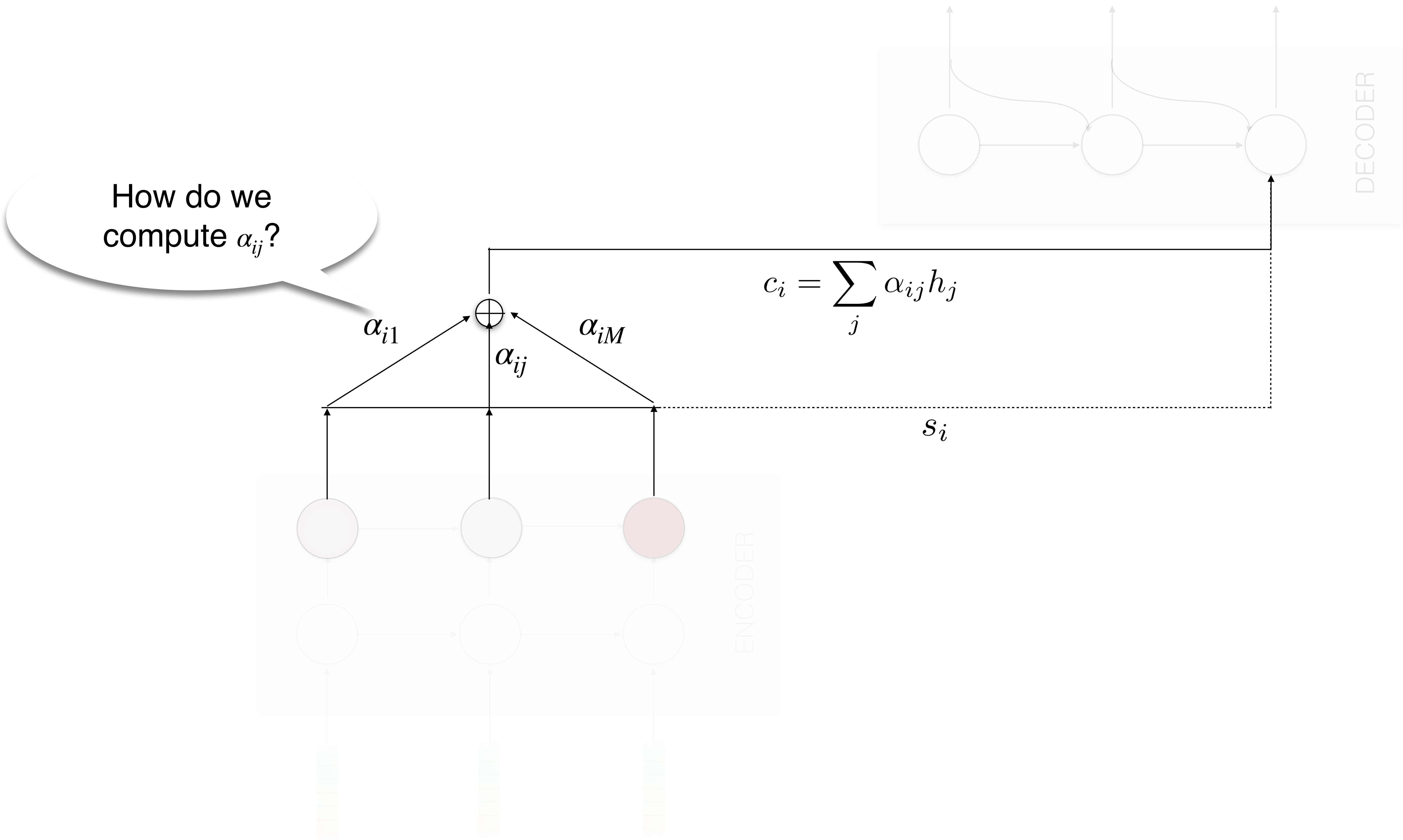
Sequence to sequence models

Encoder-decoder with attention



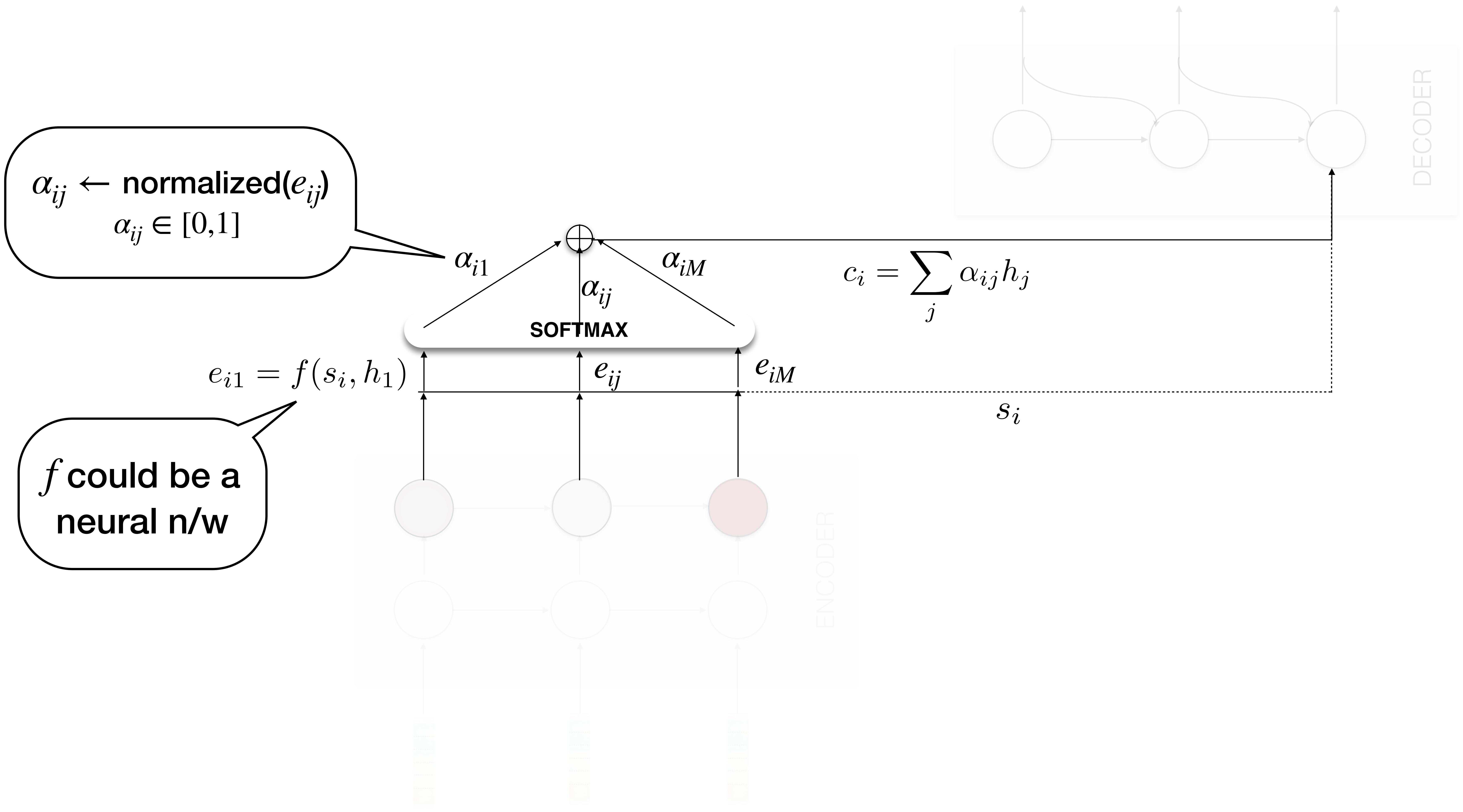
Sequence to sequence models

Encoder-decoder with attention

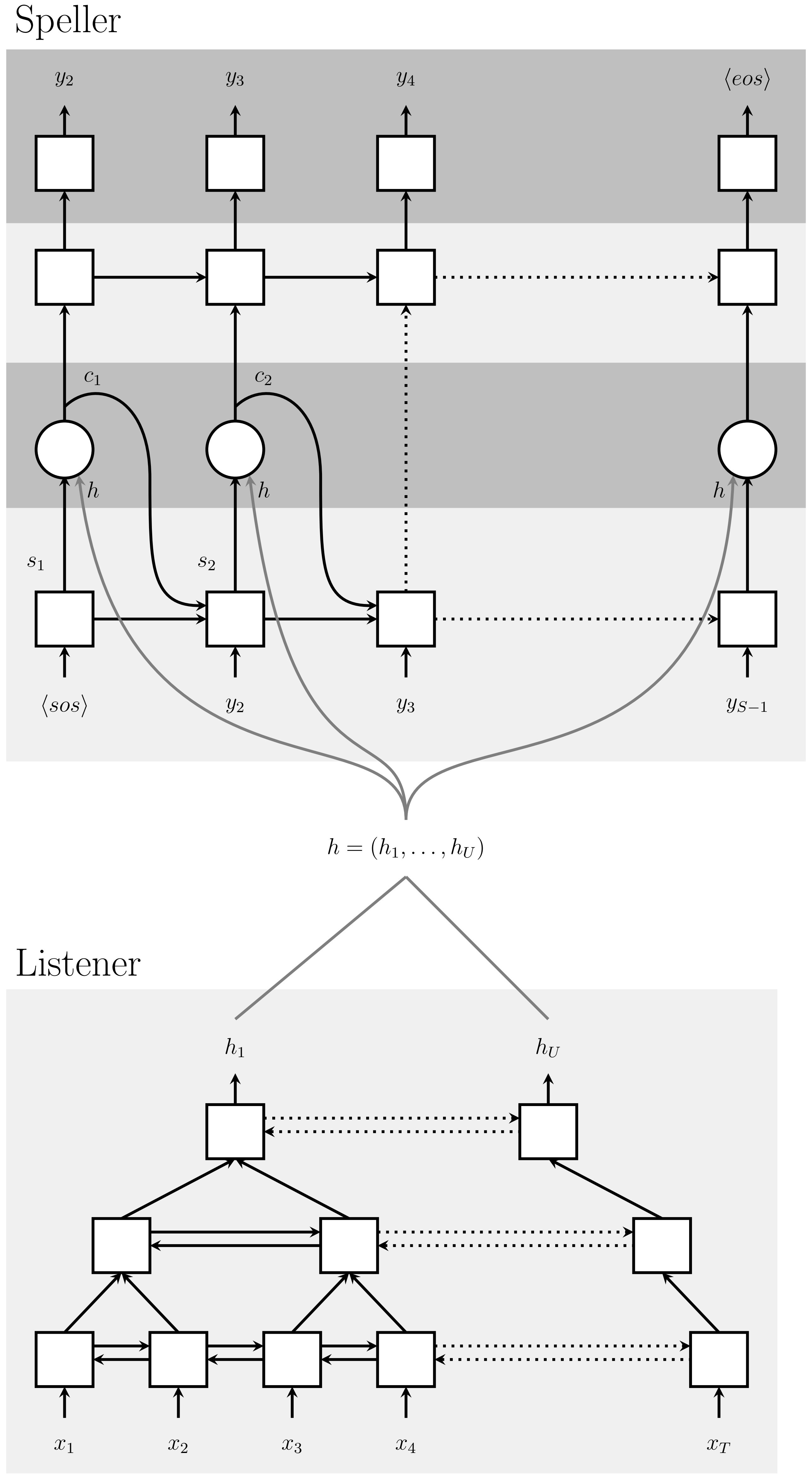


Sequence to sequence models

Encoder-decoder with attention



The Model



- The Listen, Attend & Spell (LAS) architecture is a sequence-to-sequence model consisting of
  - a Listener (Listen): An acoustic model encoder. Deep BLSTMs with a pyramidal structure: reduces the time resolution by a factor of 2 in each layer.
  - a Speller (AttendAndSpell): An attention-based decoder. Consumes $\mathbf{h}$ and produces a probability distribution over characters.

$$\mathbf{h} = \text{Listen}(\mathbf{x})$$
$$P(y_i|\mathbf{x}, y_{<i}) = \text{AttendAndSpell}(y_{<i}, \mathbf{h})$$

Image from: Chan et al., Listen, Attend and Spell: A NN for LVCSR, ICASSP 2016

Attend and spell

- Produces a distribution over characters conditioned on all characters seen previously

$$\begin{aligned} c_i &= \text{AttentionContext}(s_i, \mathbf{h}) \\ s_i &= \text{RNN}(s_{i-1}, y_{i-1}, c_{i-1}) \\ P(y_i|\mathbf{x}, y_{<i}) &= \text{CharacterDistribution}(s_i, c_i) \end{aligned}$$

- At each decoder time-step i, AttentionContext computes a score for each encoder step u, which is then converted into softmax probabilities that are linearly combined to compute c_i

$$\begin{aligned} e_{i,u} &= \langle \phi(s_i), \psi(h_u) \rangle \\ \alpha_{i,u} &= \frac{\exp(e_{i,u})}{\sum_{u'} \exp(e_{i,u'})} \\ c_i &= \sum_u \alpha_{i,u} h_u \end{aligned}$$

Training and Decoding

- Training
  - Train the parameters of the model to maximize the log probability of the training instances

$$\tilde{\theta} = \max_{\theta} \sum_{i} \log P(y_i|\mathbf{x}, \tilde{y}_{<i}; \theta)$$

- Decoding
  - Simple left-to-right beam search
  - Beams can be rescored with a language model

Experiments

Table 1: WER comparison on the clean and noisy Google voice search task. The CLDNN-HMM system is the state-of-the-art, the Listen, Attend and Spell (LAS) models are decoded with a beam size of 32. Language Model (LM) rescoring can be beneficial.

Model	Clean WER	Noisy WER
CLDNN-HMM [22]	8.0	8.9
LAS	14.1	16.5
LAS + LM Rescoring	10.3	12.0

- Listen function used 3 layers of BLSTM (512 nodes); AttendAndSpell used a 2-layer LSTM (256 nodes)

- Constraining the beam search with a dictionary had no impact on WER

Analysis

Alignment between the Characters and Audio

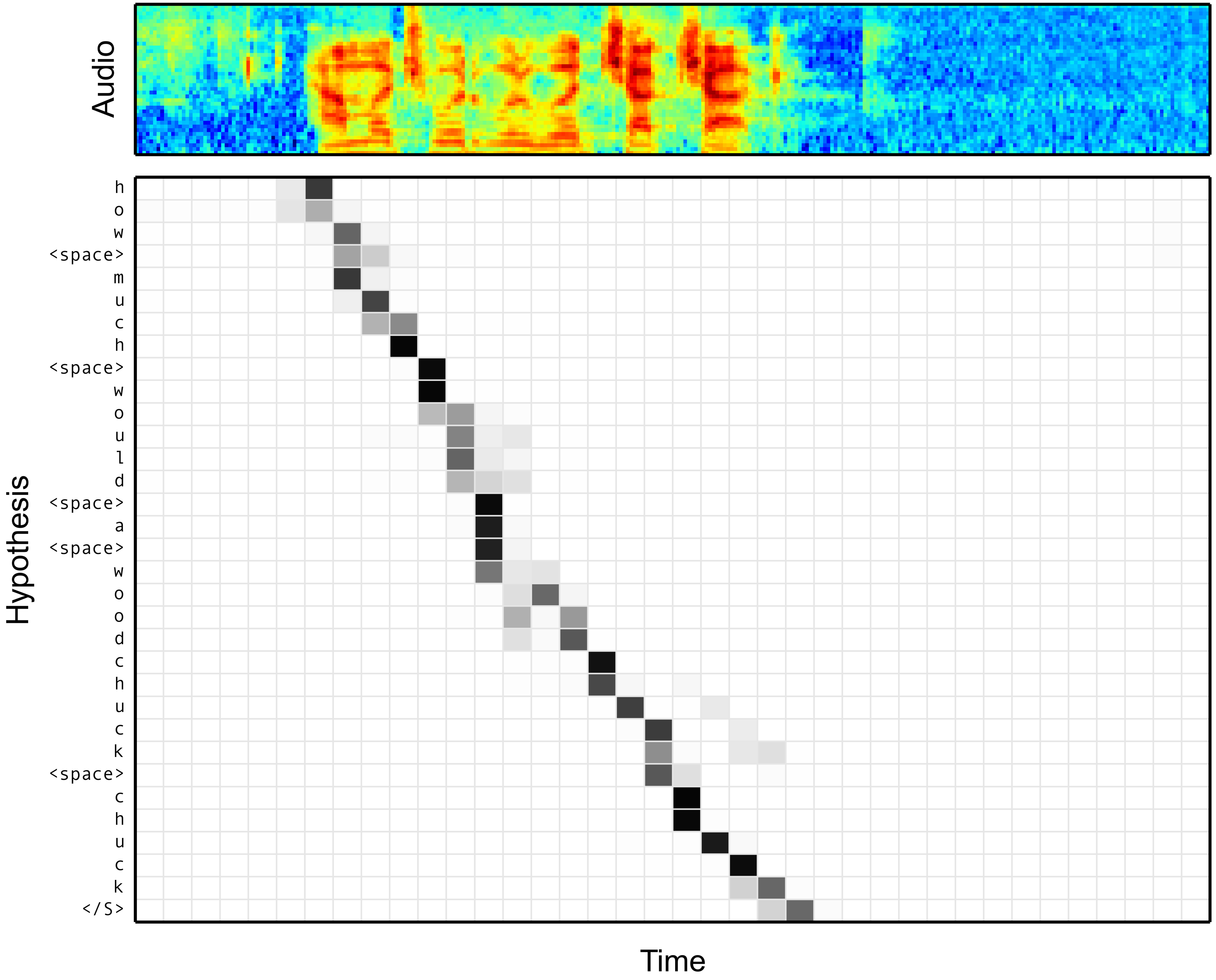
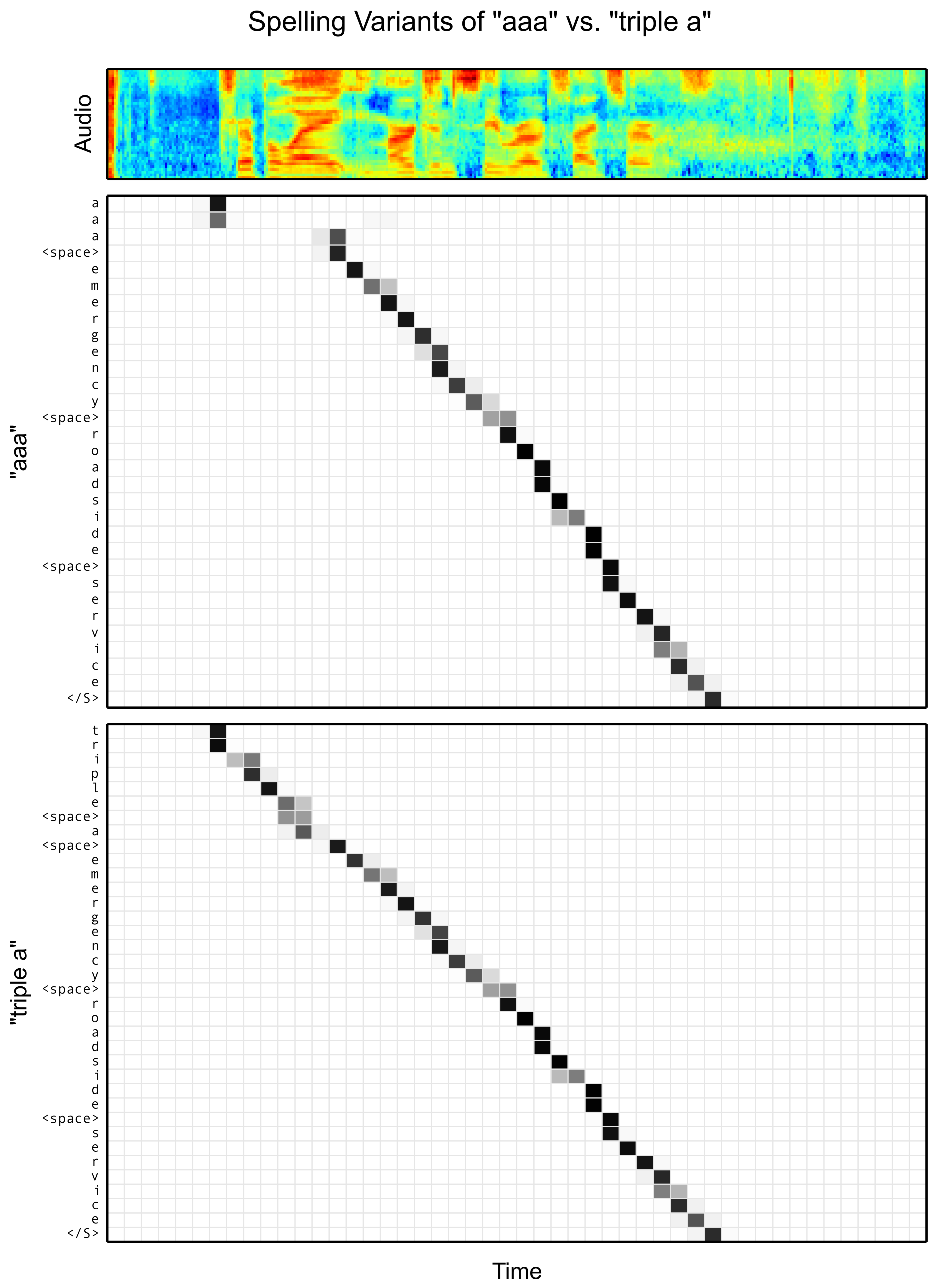


Image from: Chan et al., Listen, Attend and Spell: A NN for LVCSR, ICASSP 2016

Attention Distributions



Beam	**Text**	$\log P$	**WER**
Truth	call aaa roadside assistance	-	-
1	call aaa roadside assistance	-0.57	0.00
2	call triple a roadside assistance	-1.54	50.00
3	call trip way roadside assistance	-3.50	50.00
4	call xxx roadside assistance	-4.44	25.00

Image from: https://arxiv.org/pdf/1508.01211.pdf

Attention Distributions

Spelling Variants of "st" vs. "saint"

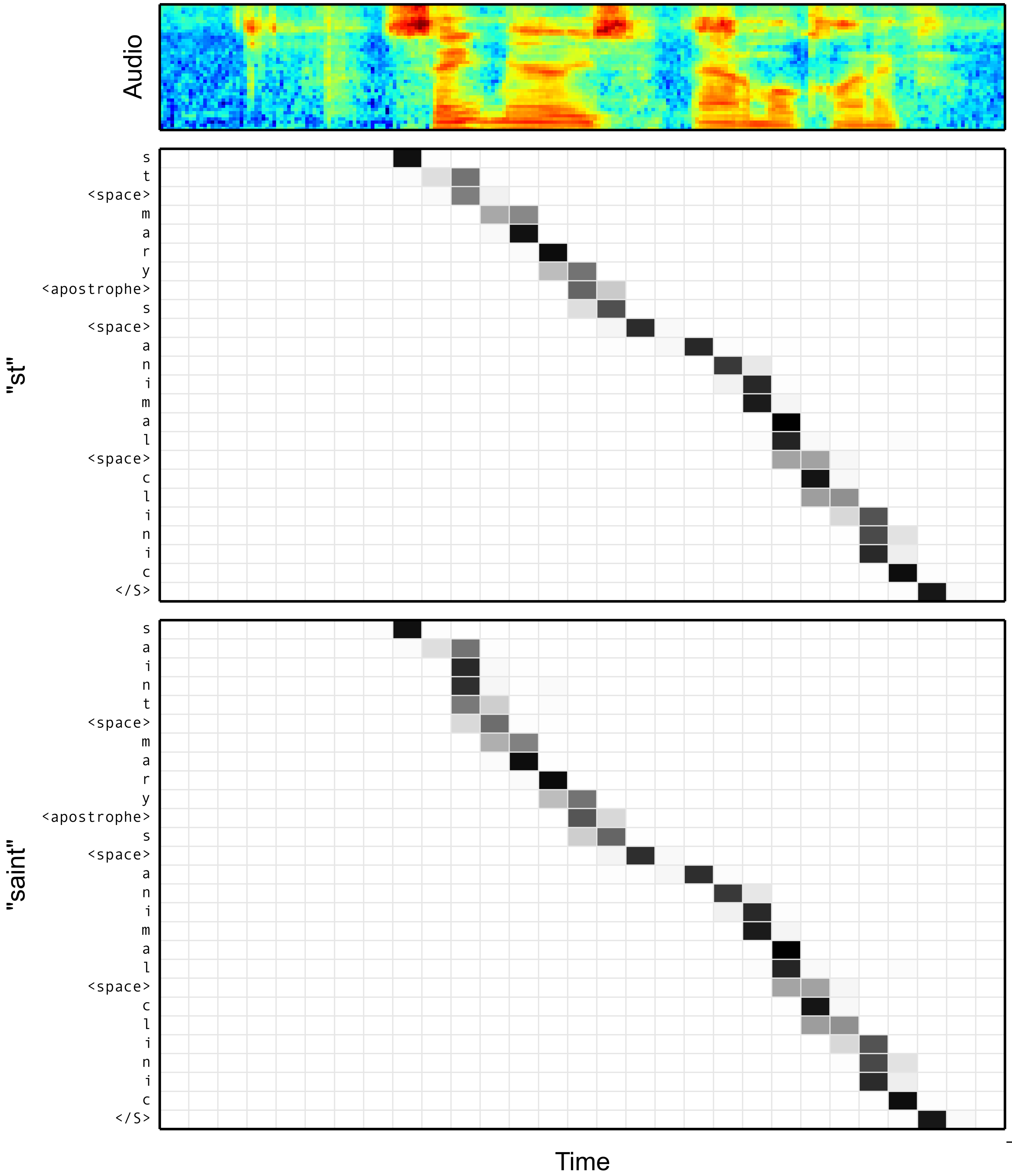


Image from: https://arxiv.org/pdf/1508.01211.pdf