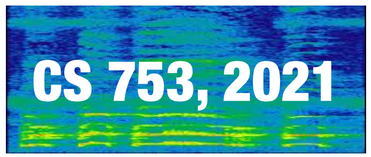
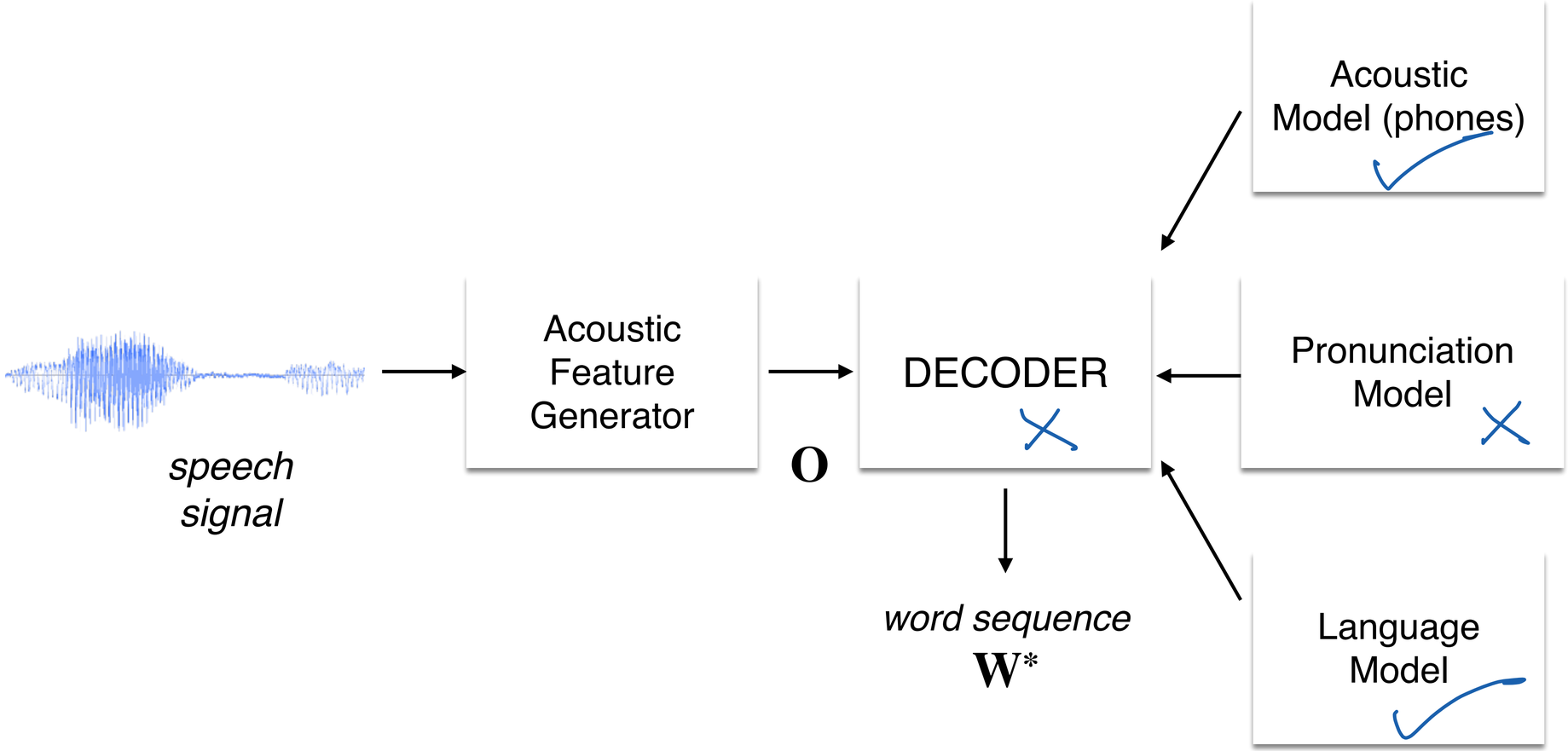
End-to-End Neural ASR Systems (I)

Lecture 7a



Instructor: Preethi Jyothi, IITB

Architecture of a Cascaded ASR system



Limitations of Cascaded ASR Systems

- Frame-level training targets derived from HMM-based alignments

Limitations of Cascaded ASR Systems

- Frame-level training targets derived from HMM-based alignments
- Pronunciation dictionaries are used to map from words to phonemes; expensive resource to create

Limitations of Cascaded ASR Systems

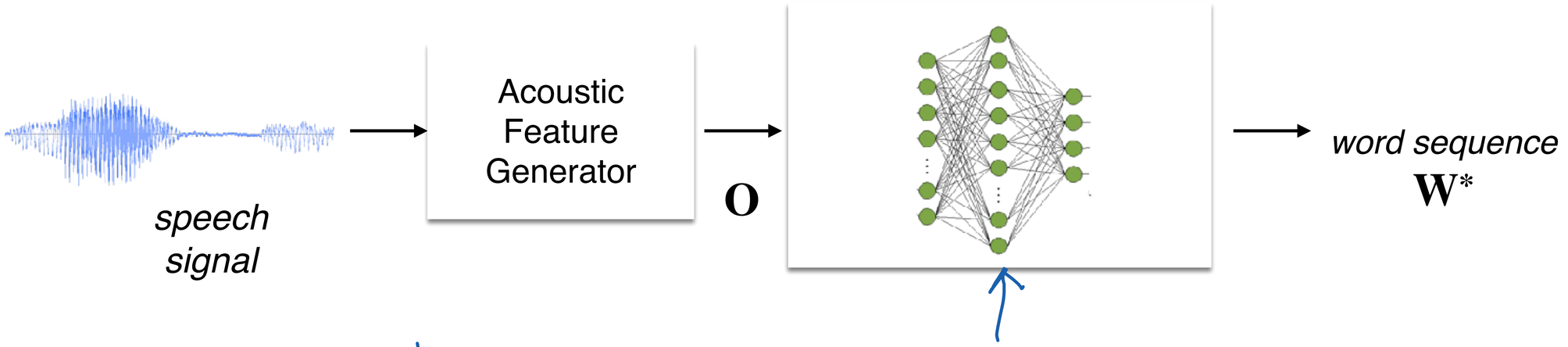
- Frame-level training targets derived from HMM-based alignments
- Pronunciation dictionaries are used to map from words to phonemes; expensive resource to create
- Complex training process, and difficult to globally optimise

Limitations of Cascaded ASR Systems

- Frame-level training targets derived from HMM-based alignments
- Pronunciation dictionaries are used to map from words to phonemes; expensive resource to create
- Complex training process, and difficult to globally optimise
- [Limitation not specific to cascaded systems per se]
  Objective function optimized in neural networks very different from final evaluation metric (i.e. word transcription accuracy)

Cascaded ASR ⇒ End-to-end ASR

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



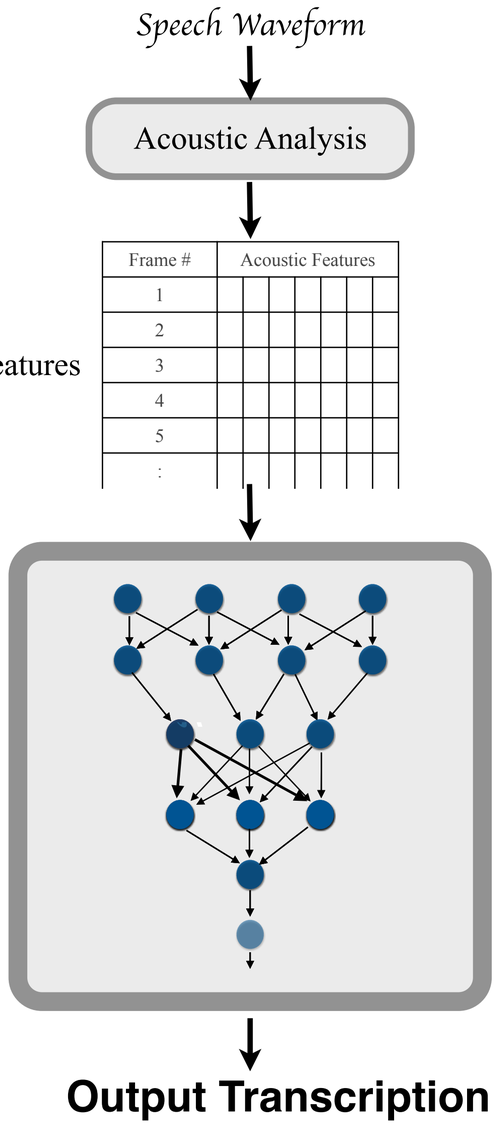
Single end-to-end model that directly learns a mapping from speech to text

End-to-End ASR Systems

- All components trained jointly as a single end-to-end model
- Trained using pairs of speech clips and their corresponding text transcripts

Pairs of speech clips and corr. grapheme sequences

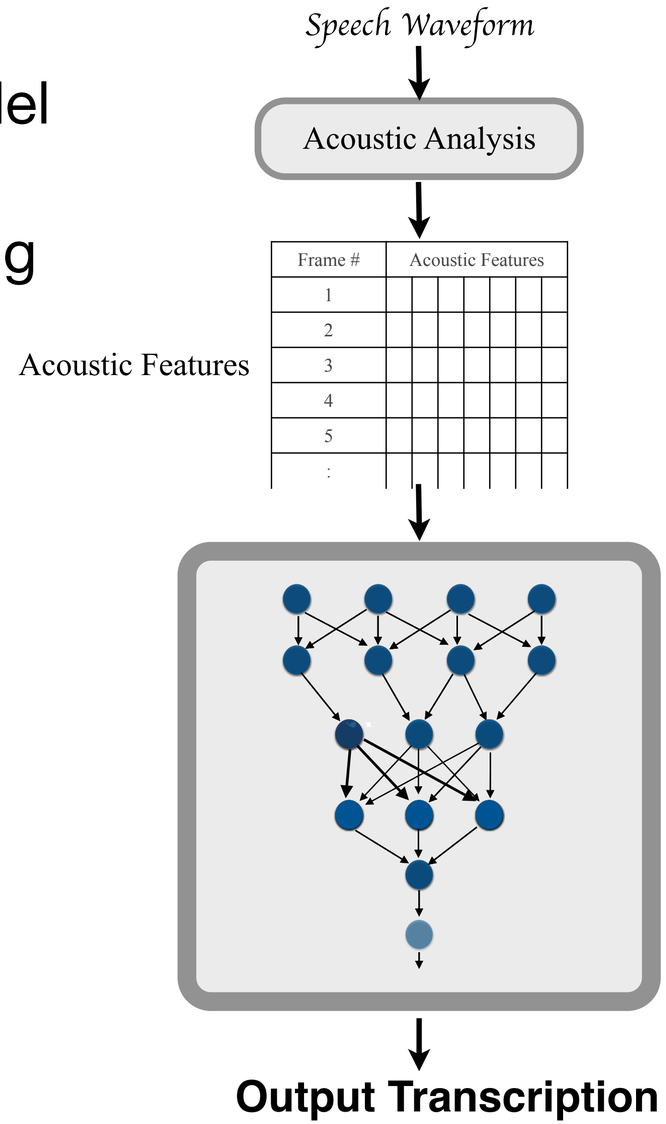
- End-to-end models, with sufficient data, sometimes outperform conventional ASR systems [1,2]



[1] "State-of-the-art speech recognition with sequence-to-sequence models", Chiu et al., 2018
[2] "RWTH ASR Systems for LibriSpeech: Hybrid vs Attention", Lüscher, Christoph, et al. , 2019

End-to-End ASR Systems

- All components trained jointly as a single end-to-end model
- Trained using pairs of speech clips and their corresponding text transcripts
- End-to-end models, with sufficient data, sometimes outperform conventional ASR systems [1,2]



	dev	test
DNN-HMM	4.0	4.4
E2E (Attention)	4.7	4.8

Librispeech-960

	dev	test
DNN-HMM	5.0	5.8
E2E (Attention)	14.7	14.7

Librispeech-100

[1] "State-of-the-art speech recognition with sequence-to-sequence models", Chiu et al., 2018
[2] "RWTH ASR Systems for LibriSpeech: Hybrid vs Attention", Lüscher, Christoph, et al. , 2019

A fully end-to-end system

- Build an end-to-end (e2e) model that subsumes all the standard ASR components (ideally, without any additional language model during decoding)

- Attention-based e2e model: Makes *no independence assumptions* (unlike the CTC models) about the probability distribution of the output sequences given the input

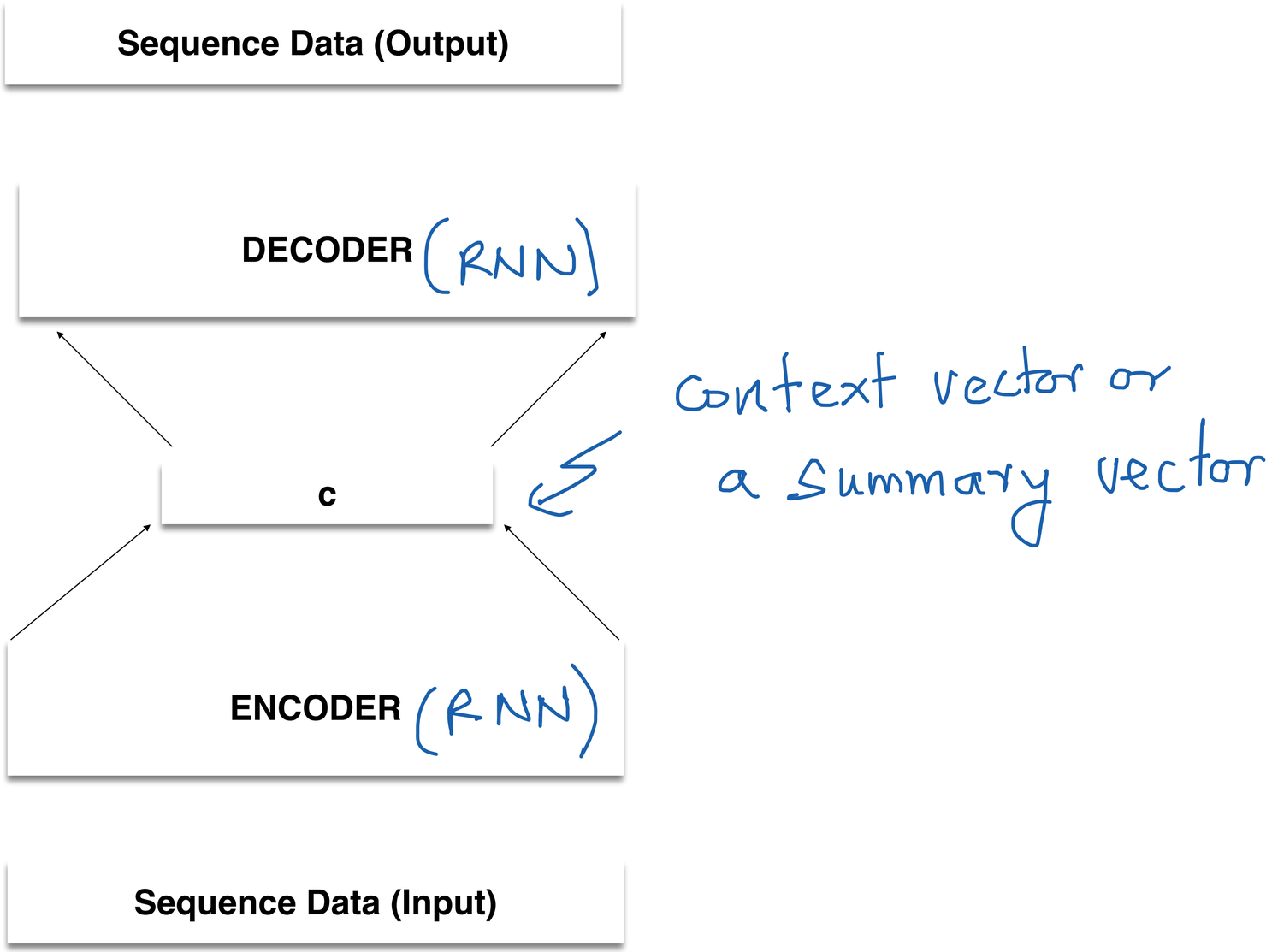
$$P(\mathbf{y}|\mathbf{x}) = \prod_i P(y_i|\mathbf{x}, y_{<i})$$

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

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

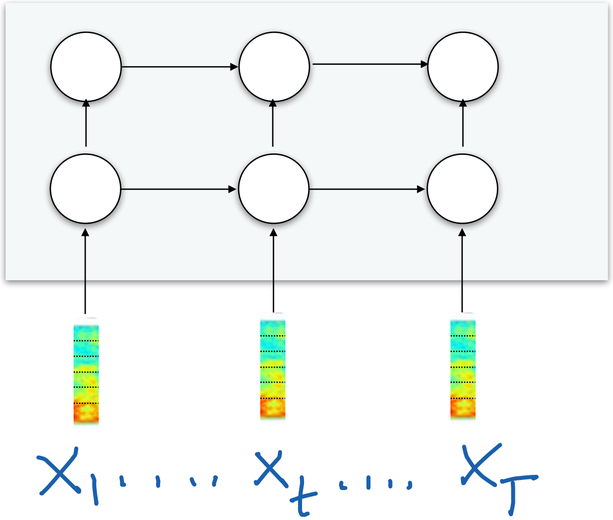
Sequence to sequence models

Encoder-decoder architecture



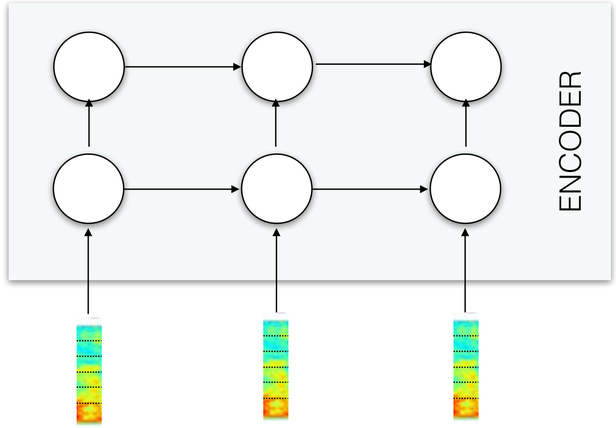
Sequence to sequence models

Encoder-decoder architecture



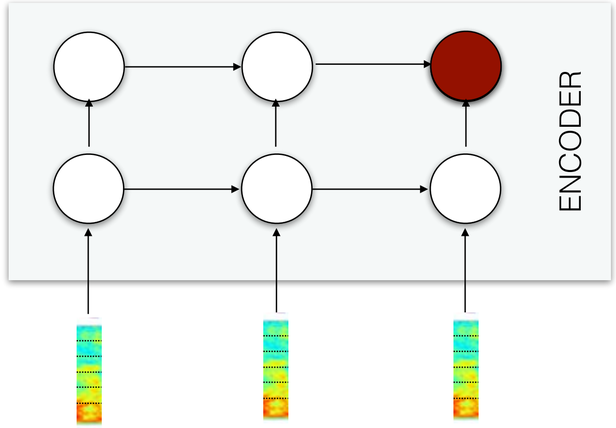
Sequence to sequence models

Encoder-decoder architecture



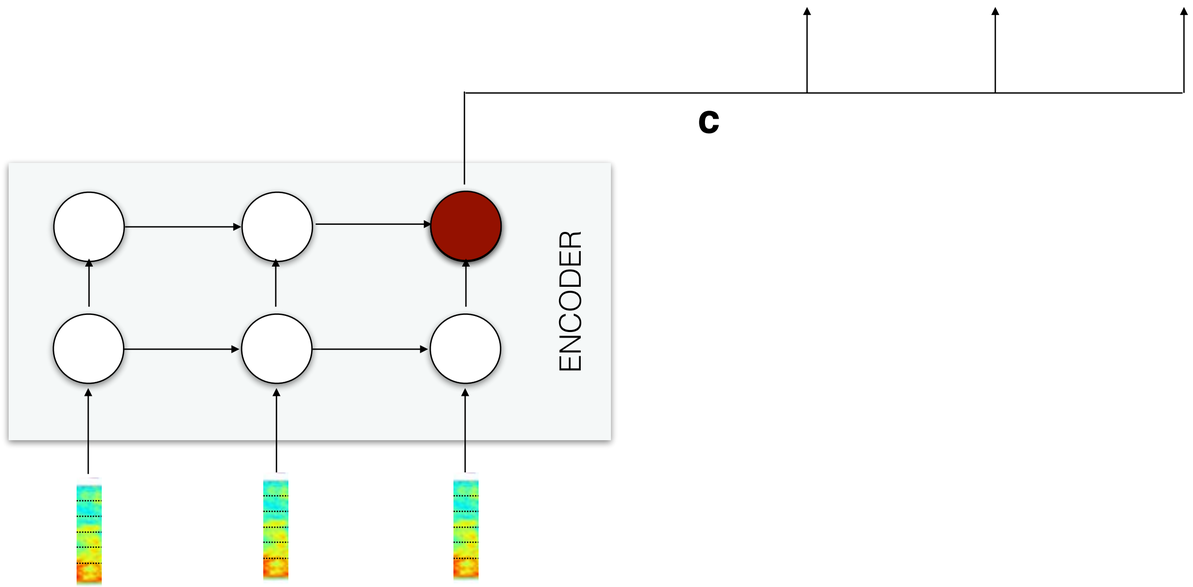
Sequence to sequence models

Encoder-decoder architecture



Sequence to sequence models

Encoder-decoder architecture



Sequence to sequence models

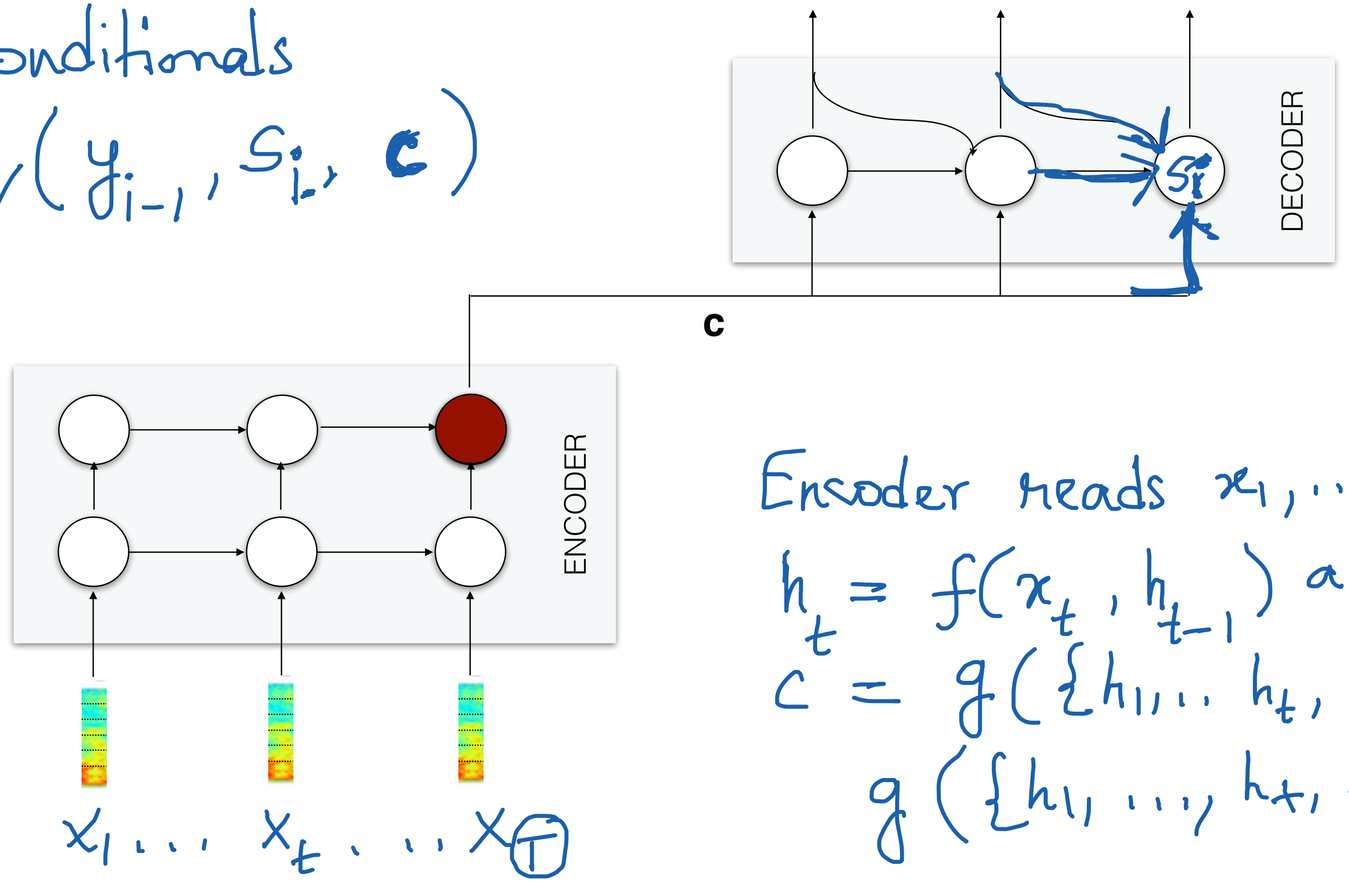
Encoder-decoder architecture

$$\log P(y|x) = \sum_{i=1}^{M} \log P(y_i | y_{1..i-1}, c) \quad (A)$$

Training objective: Minimize $\sum_{(x,y)\in\mathcal{D}} -\log P(y|x)$

(A) Each of these conditionals

$$P(y_i | y_{1..i-1}, c) = g(y_{i-1}, s_i, c)$$



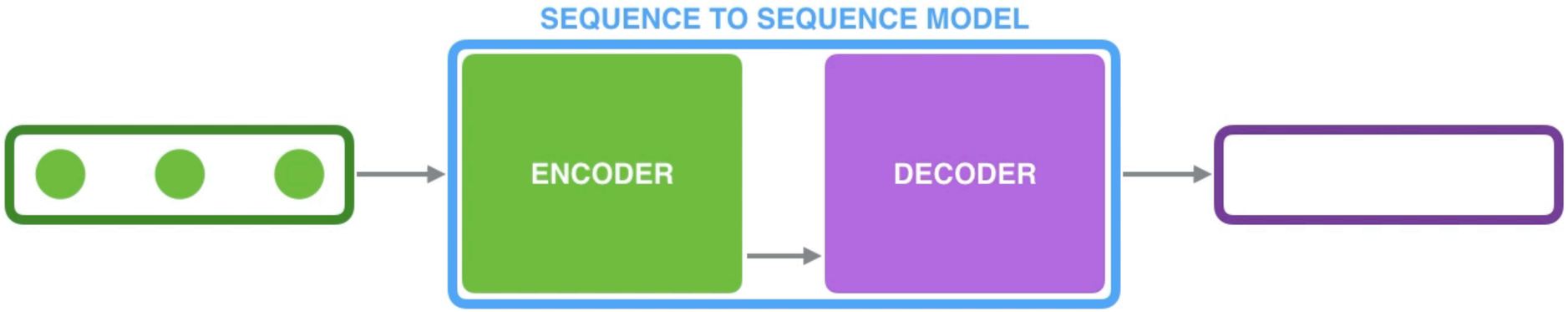
Encoder reads $x_1, \ldots, x_T$ & produces

$$h_t = f(x_t, h_{t-1}) \text{ and}$$

$$c = g(\{h_1, \ldots h_t, \ldots, h_T\})$$

$$g(\{h_1, \ldots, h_t, \ldots h_T\}) = h_T$$

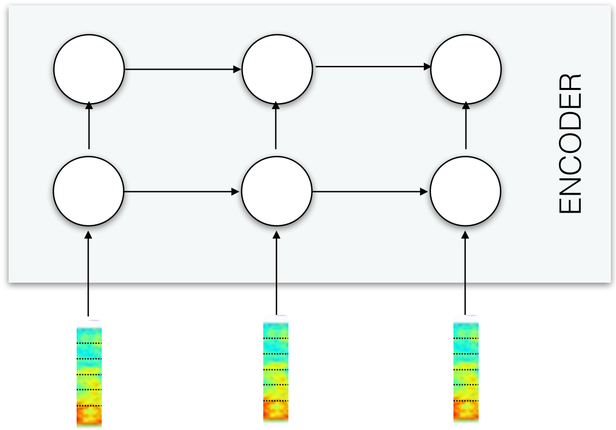
Sequence to sequence model



Jay Alammar, Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention), 2018. http://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

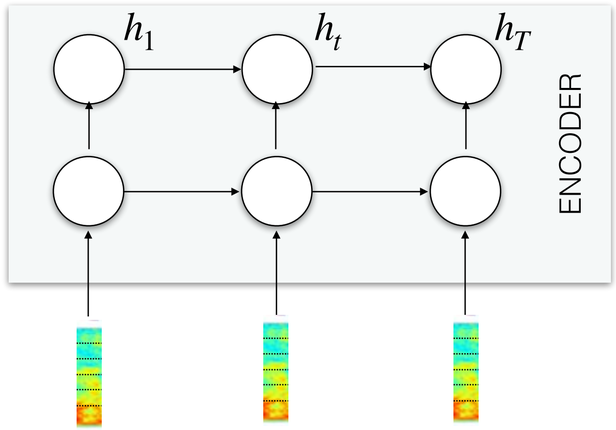
Sequence to sequence models

Encoder-decoder architecture with attention



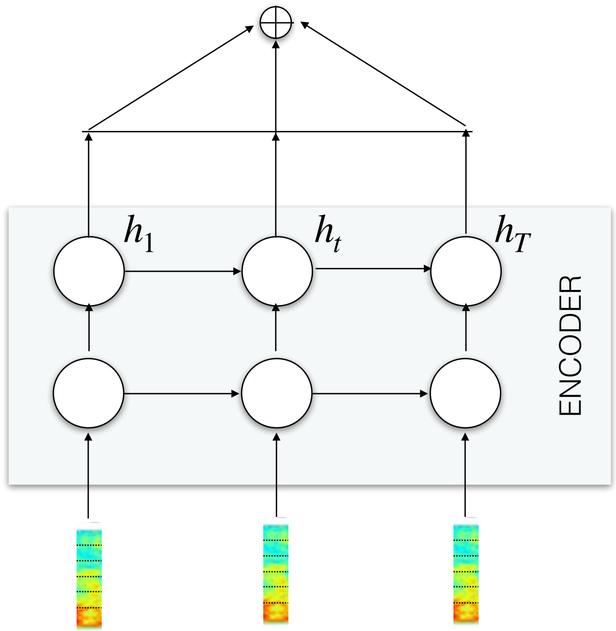
Sequence to sequence models

Encoder-decoder architecture with attention



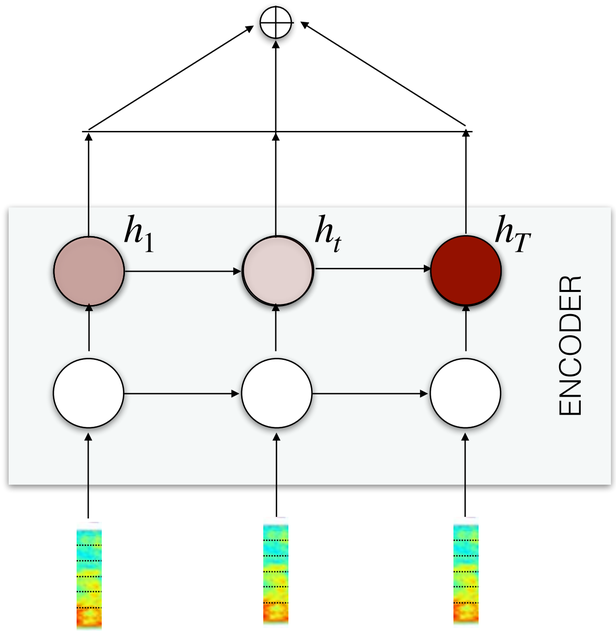
Sequence to sequence models

Encoder-decoder architecture with attention



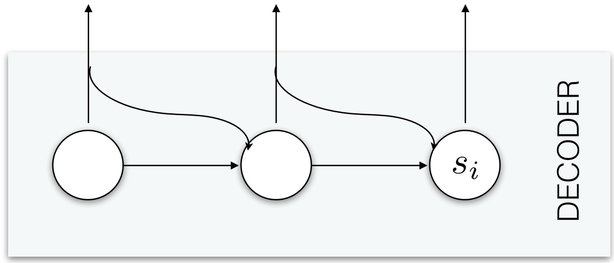
Sequence to sequence models

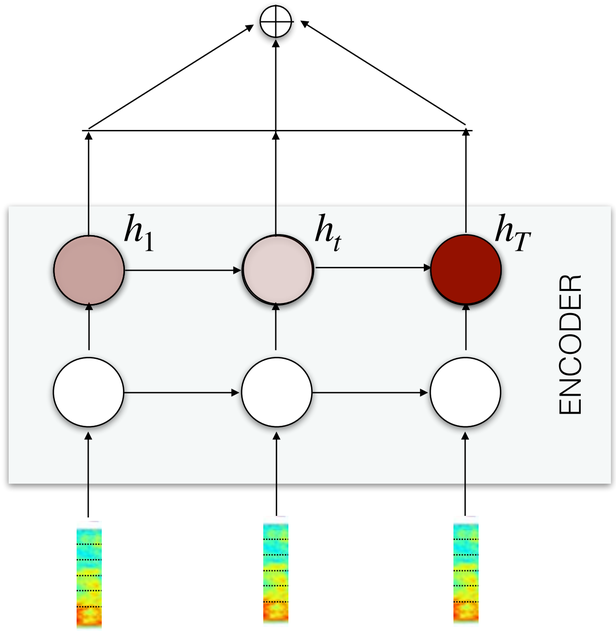
Encoder-decoder architecture with attention



Sequence to sequence models

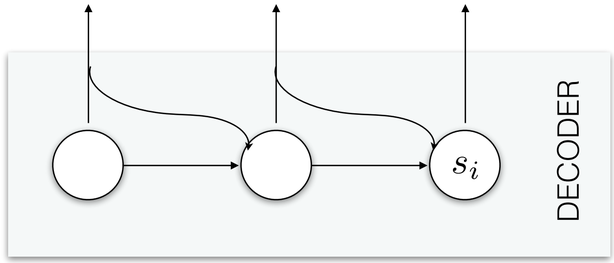
Encoder-decoder architecture with attention

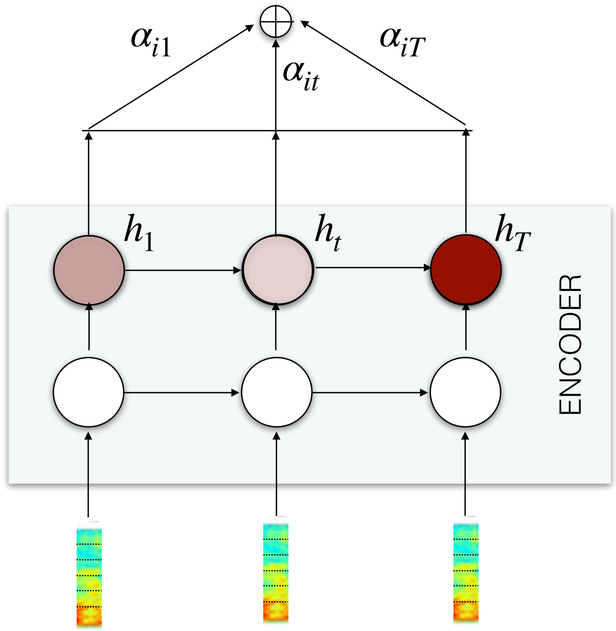




Sequence to sequence models

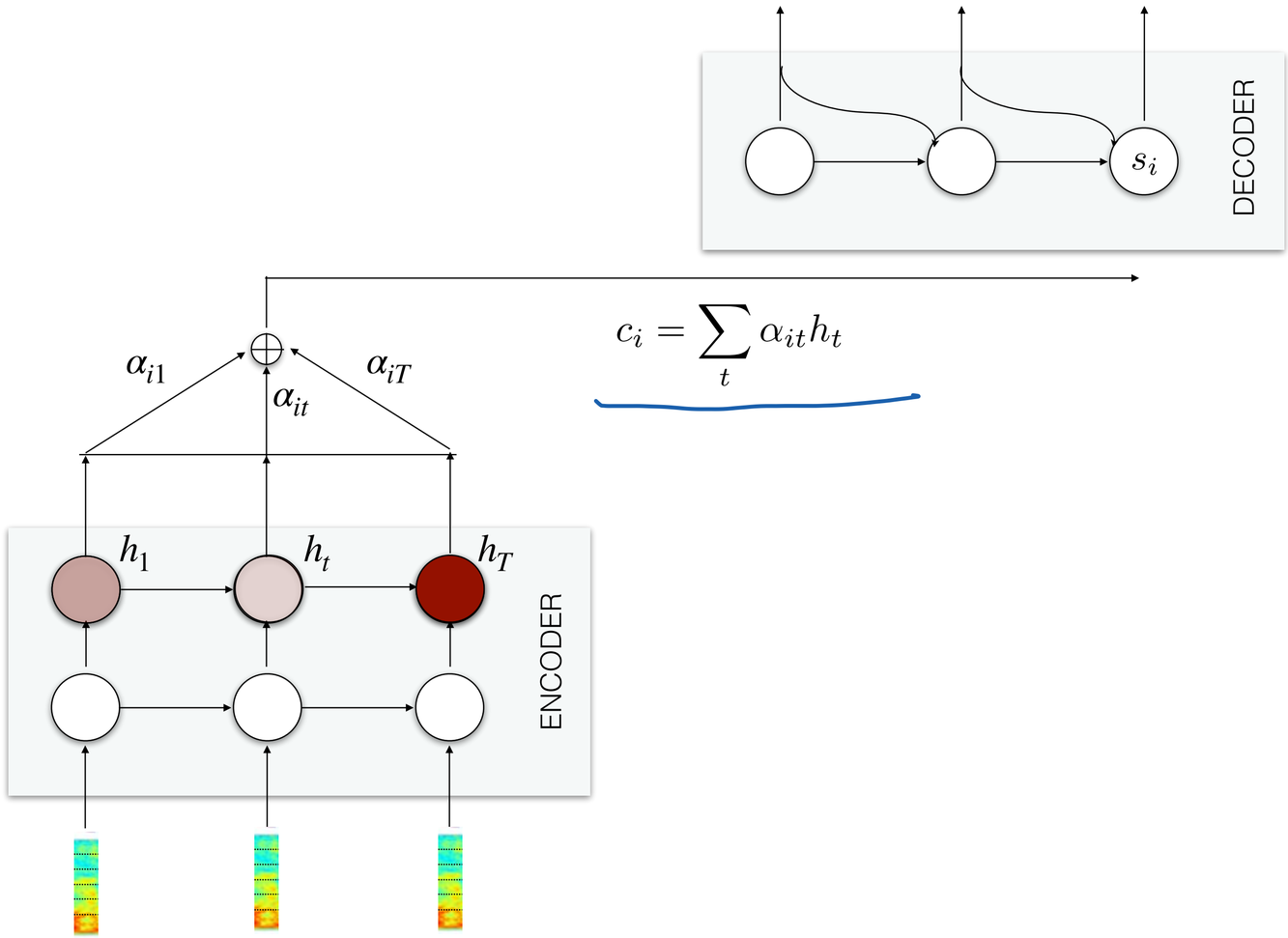
Encoder-decoder architecture with attention





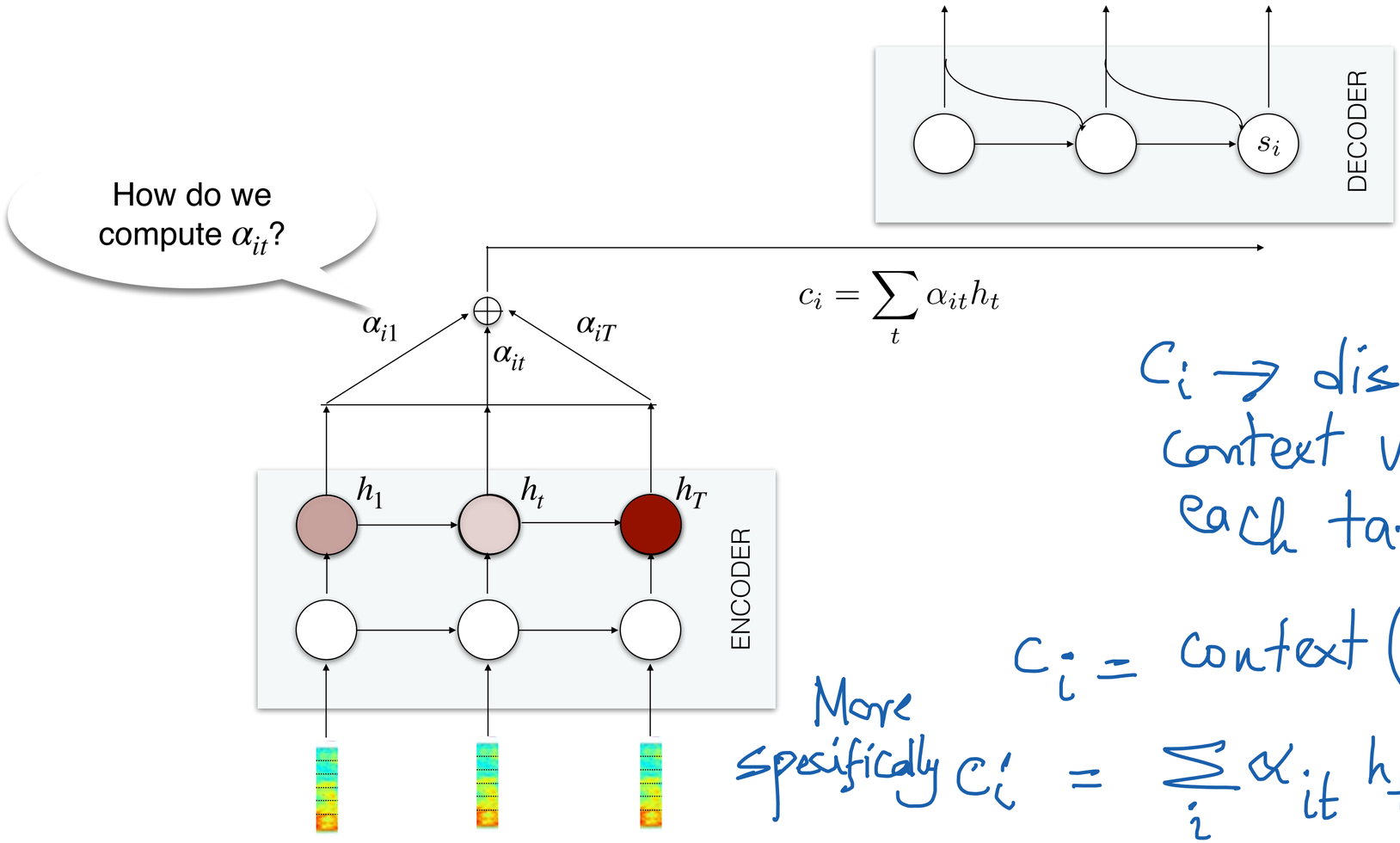
Sequence to sequence models

Encoder-decoder architecture with attention



Sequence to sequence models

Encoder-decoder architecture with attention



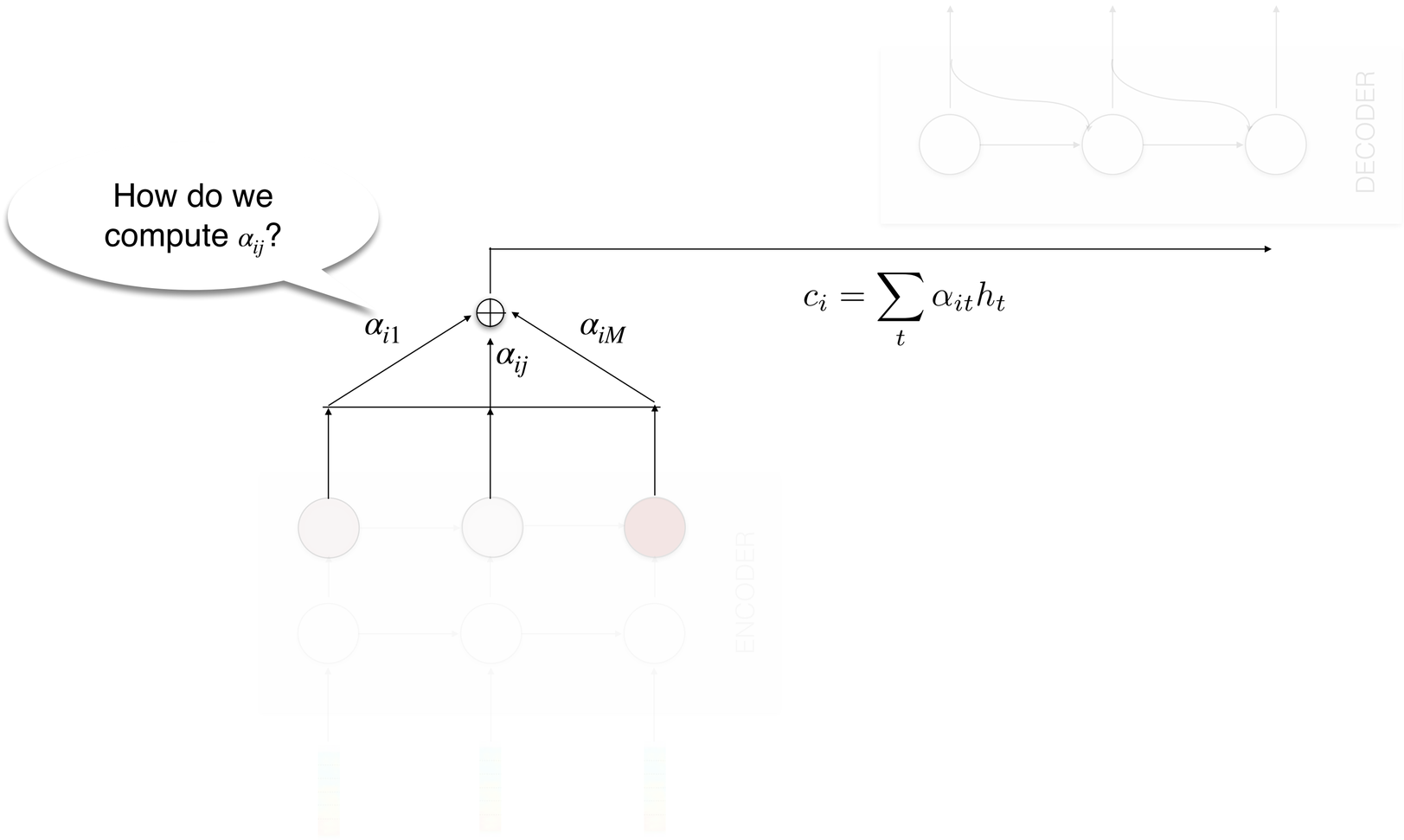
$c_i \rightarrow$ distinct context vector for each target output y_i

$c_i = \text{context}(s_i, \{h_1, \ldots h_t \ldots, h_T\})$

More specifically $c_i = \sum_i \alpha_{it} h_t$

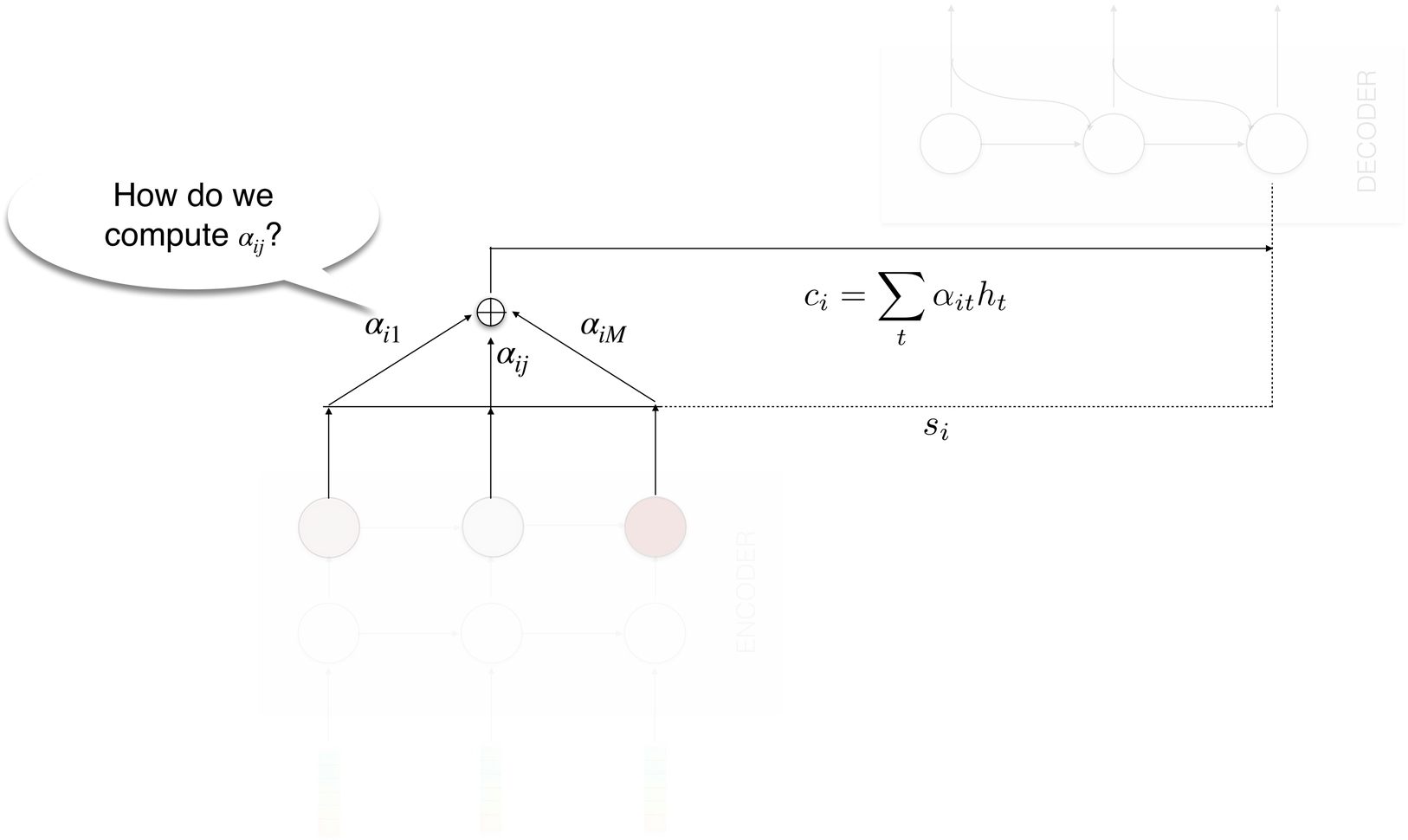
Sequence to sequence models

Encoder-decoder architecture with attention



Sequence to sequence models

Encoder-decoder architecture with attention



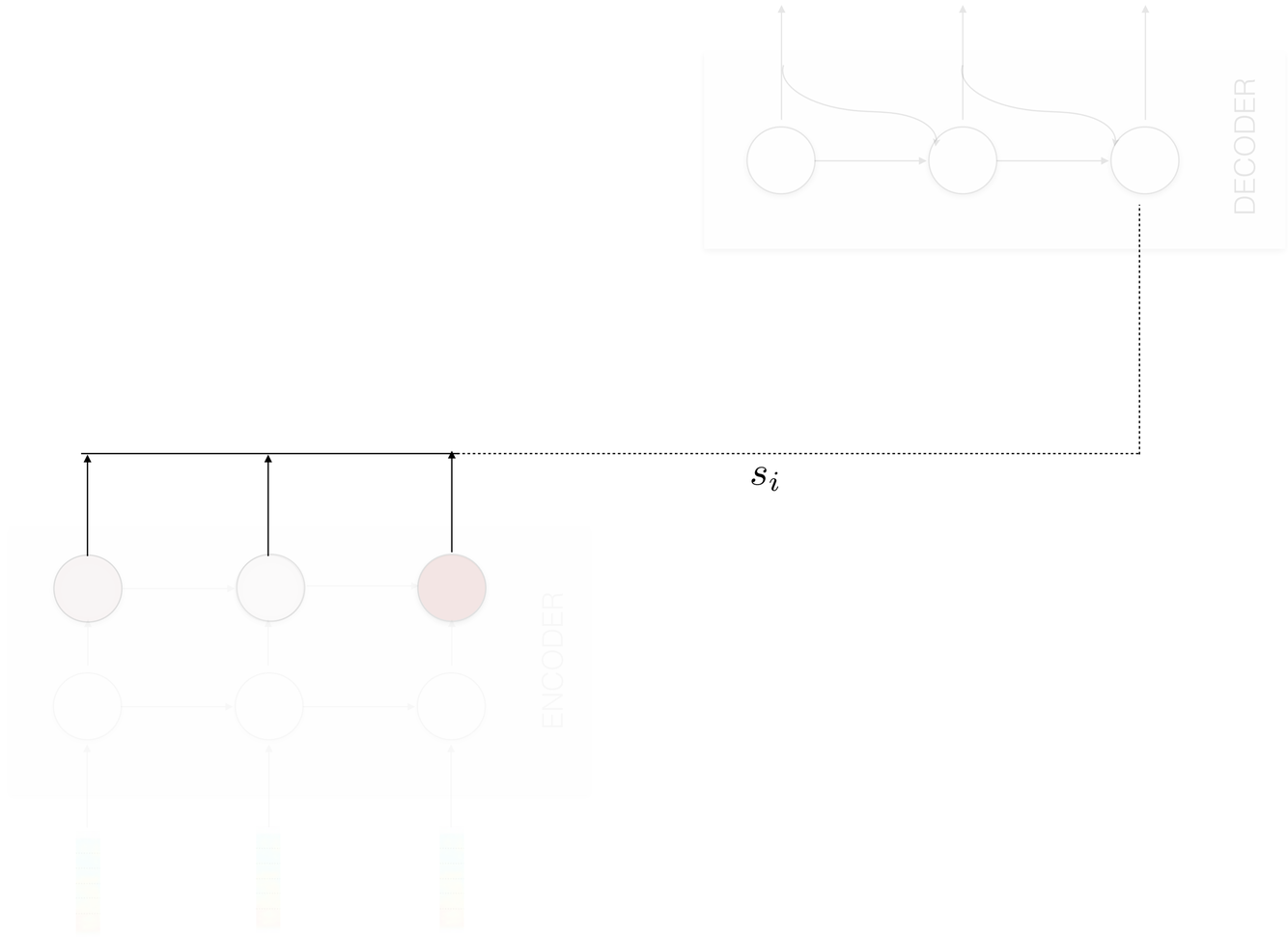
Sequence to sequence model with Attention

Attention at time step 4

Jay Alammar, Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention), 2018.
http://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

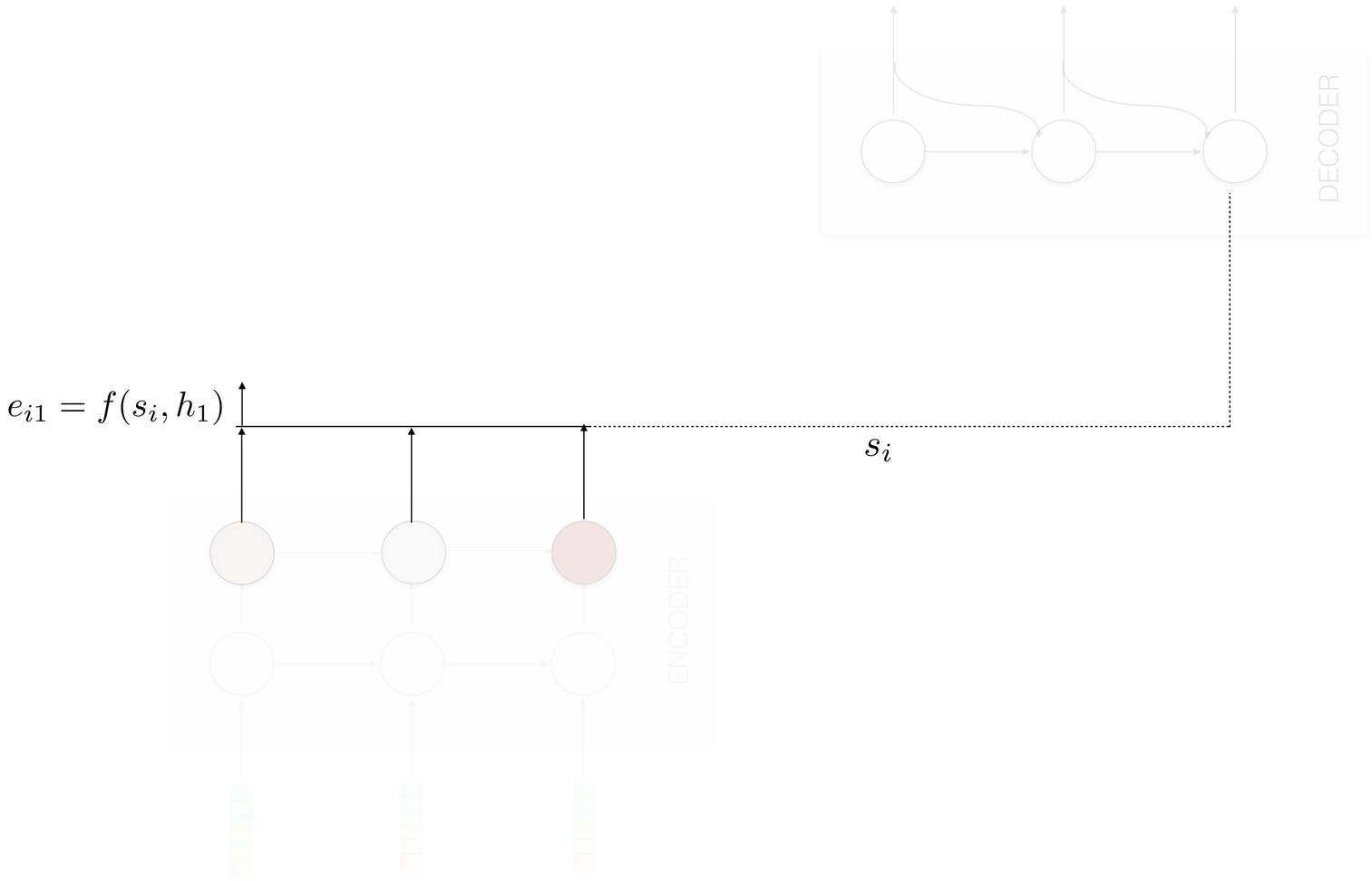
Sequence to sequence models

Encoder-decoder architecture with attention



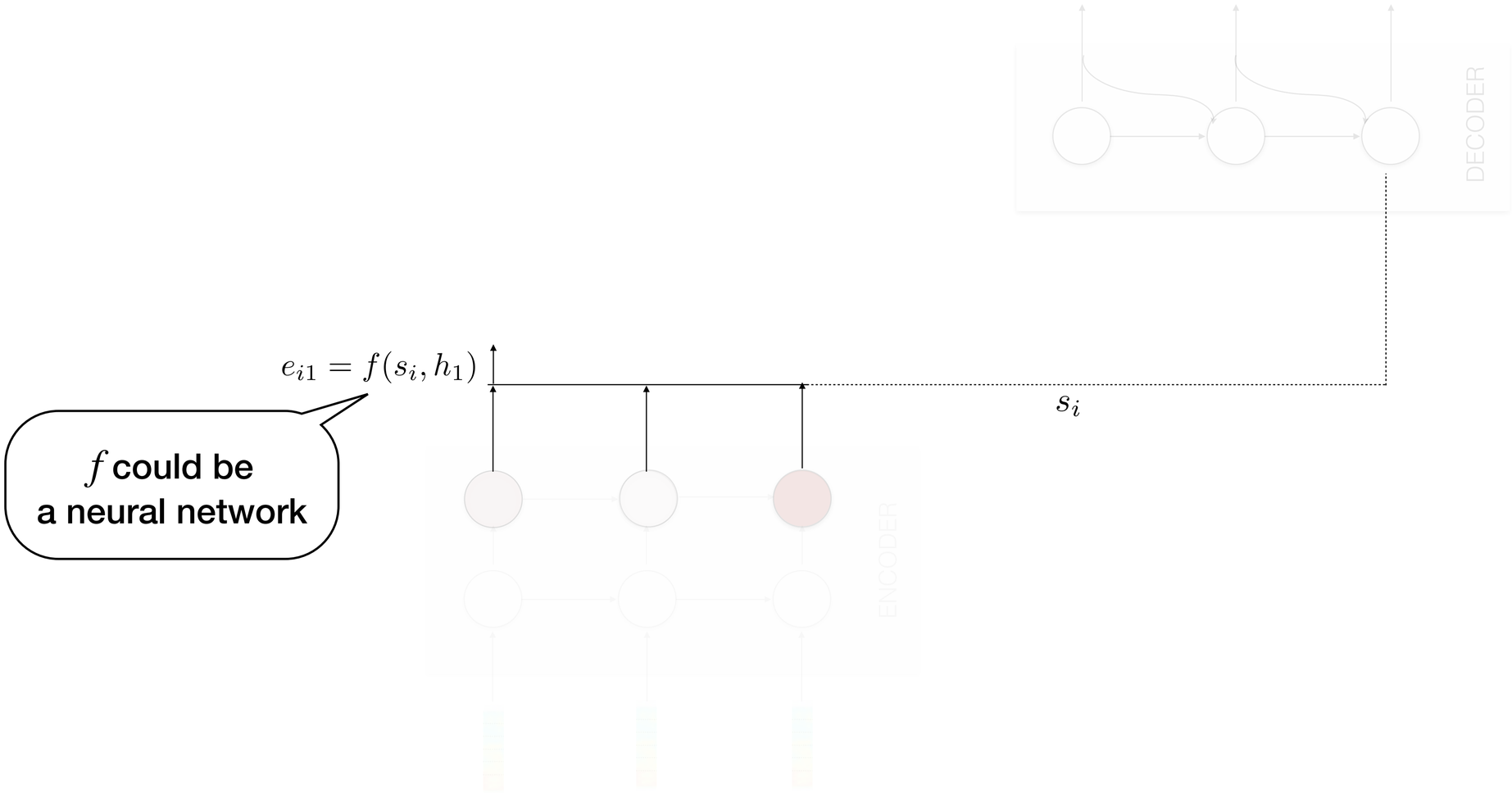
Sequence to sequence models

Encoder-decoder architecture with attention



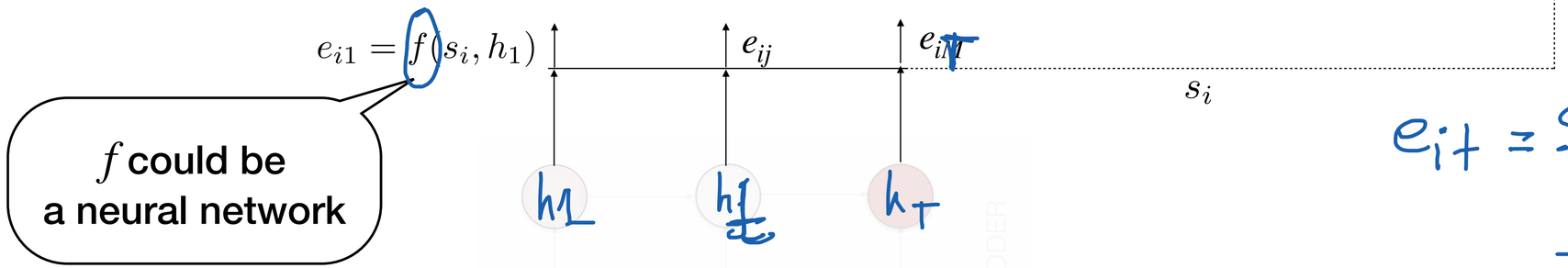
Sequence to sequence models

Encoder-decoder architecture with attention



Sequence to sequence models

Encoder-decoder architecture with attention



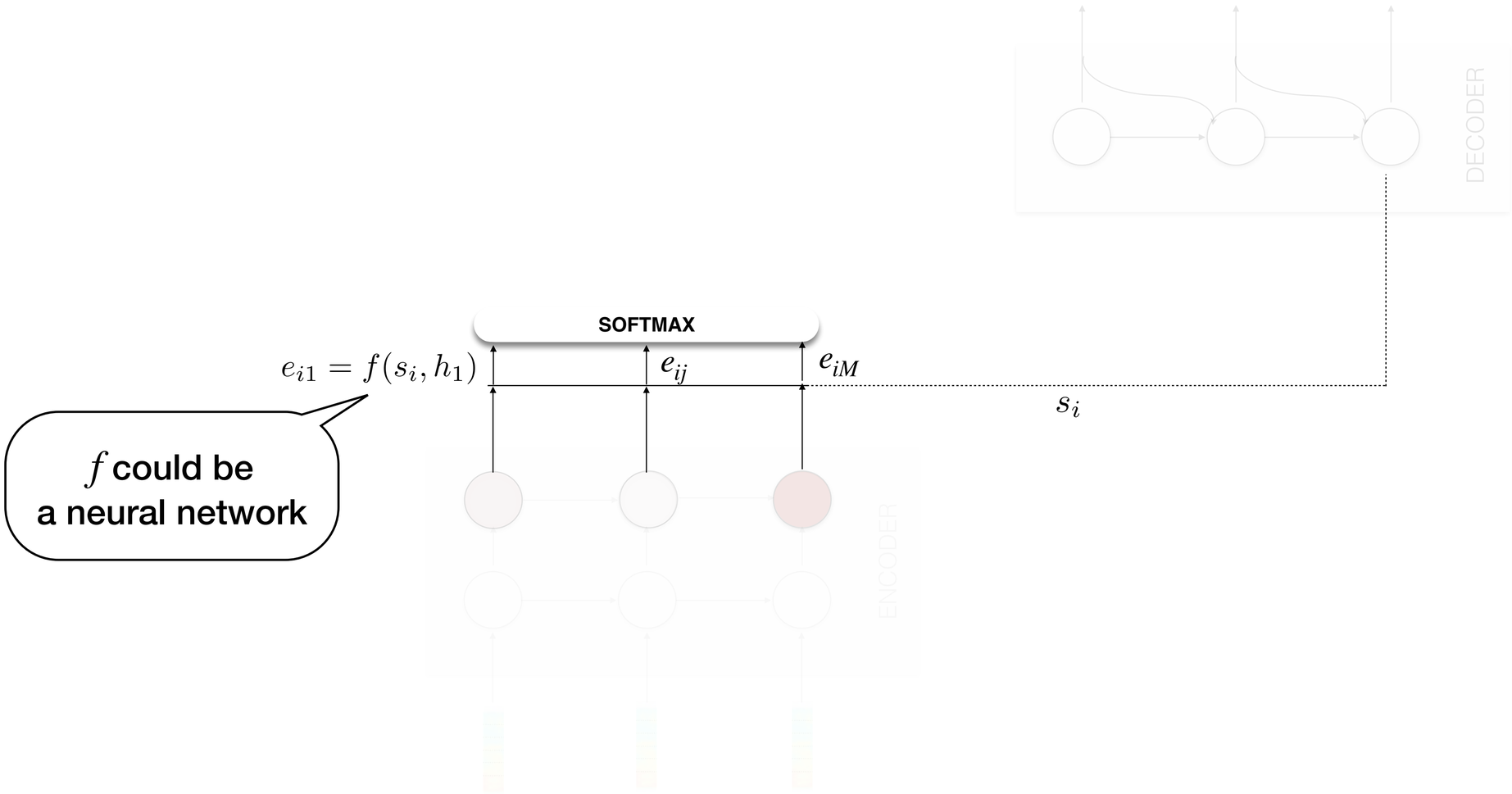
first compute scores $e_{it} = f(s_i, h_t)$

$$e_{it} = \text{score}(s_i, h_t)$$

$$= \begin{cases} s_i^T h_t & \text{dot} \\ s_i^T W_a h_t & \text{general} \\ v_a^T \tanh(W_a [s_i ; h_t]) & \text{concat} \end{cases}$$

Sequence to sequence models

Encoder-decoder architecture with attention

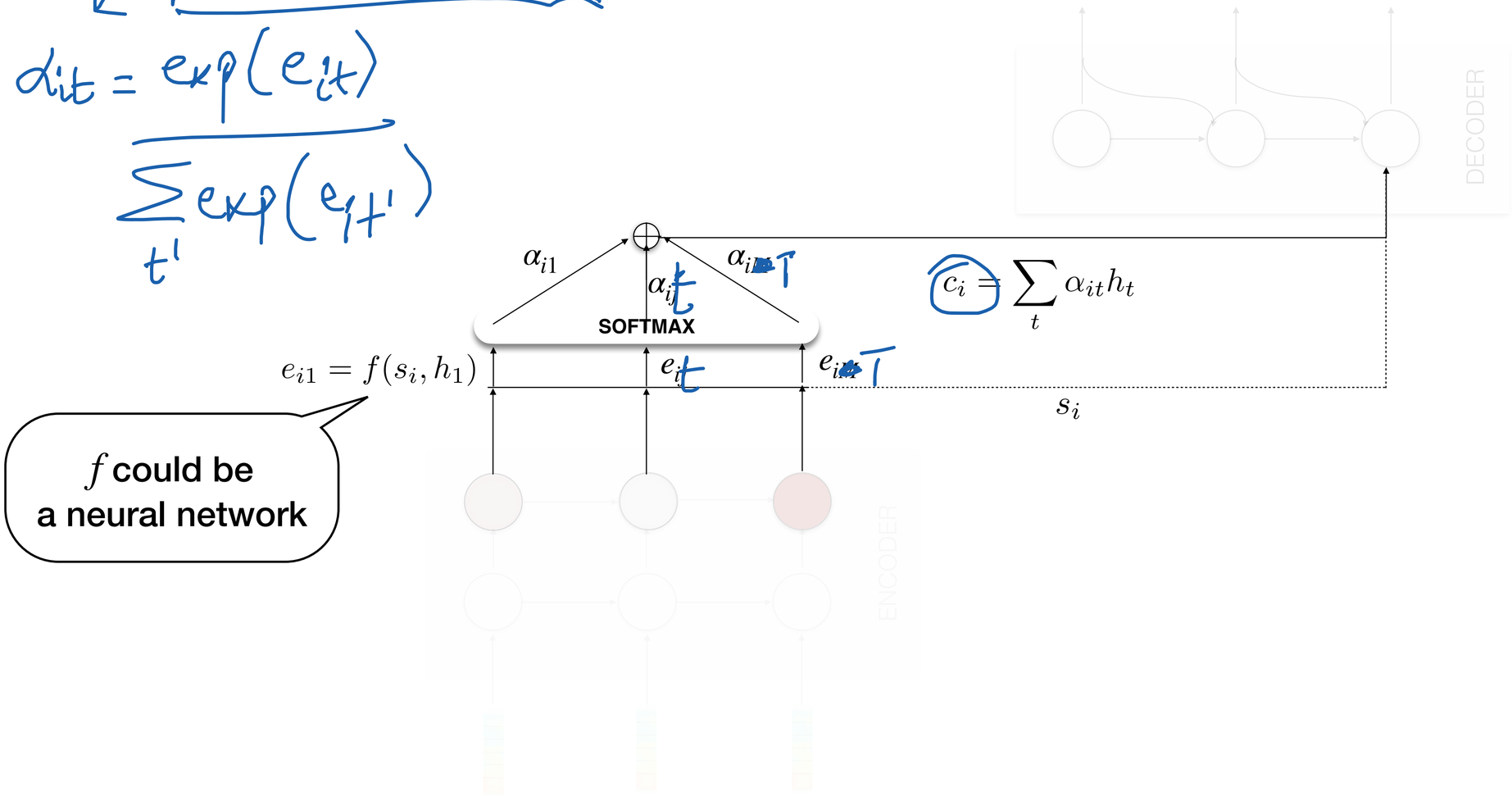


Sequence to sequence models

Encoder-decoder architecture with attention

attention distribution

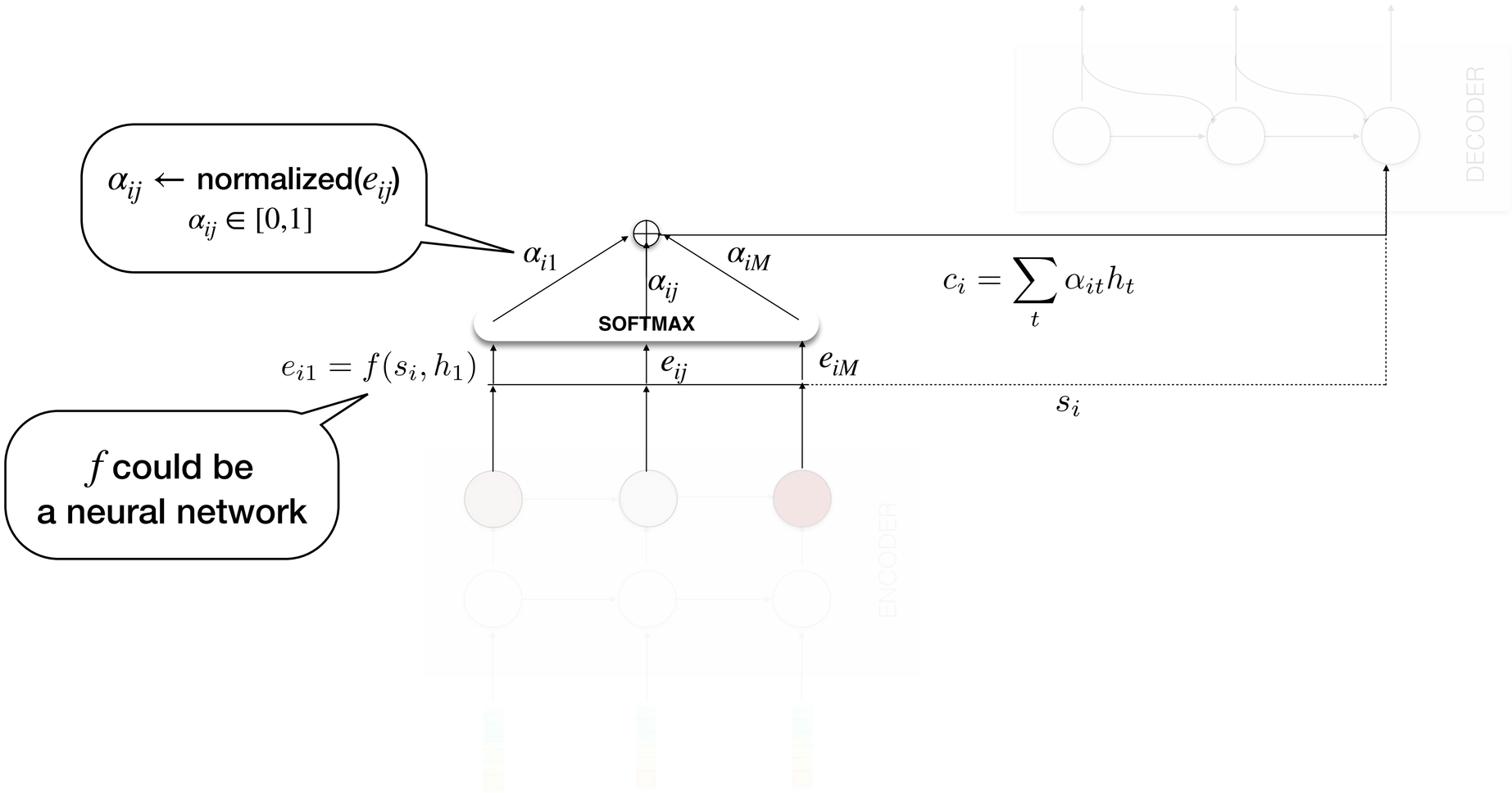
$$\alpha_{it} = \frac{\exp(e_{it})}{\sum_{t'} \exp(e_{it'})}$$



f could be a neural network

Sequence to sequence models

Encoder-decoder architecture with attention



The Model

- The Listen, Attend & Spell (LAS) architecture is a sequence-to-sequence model consisting of

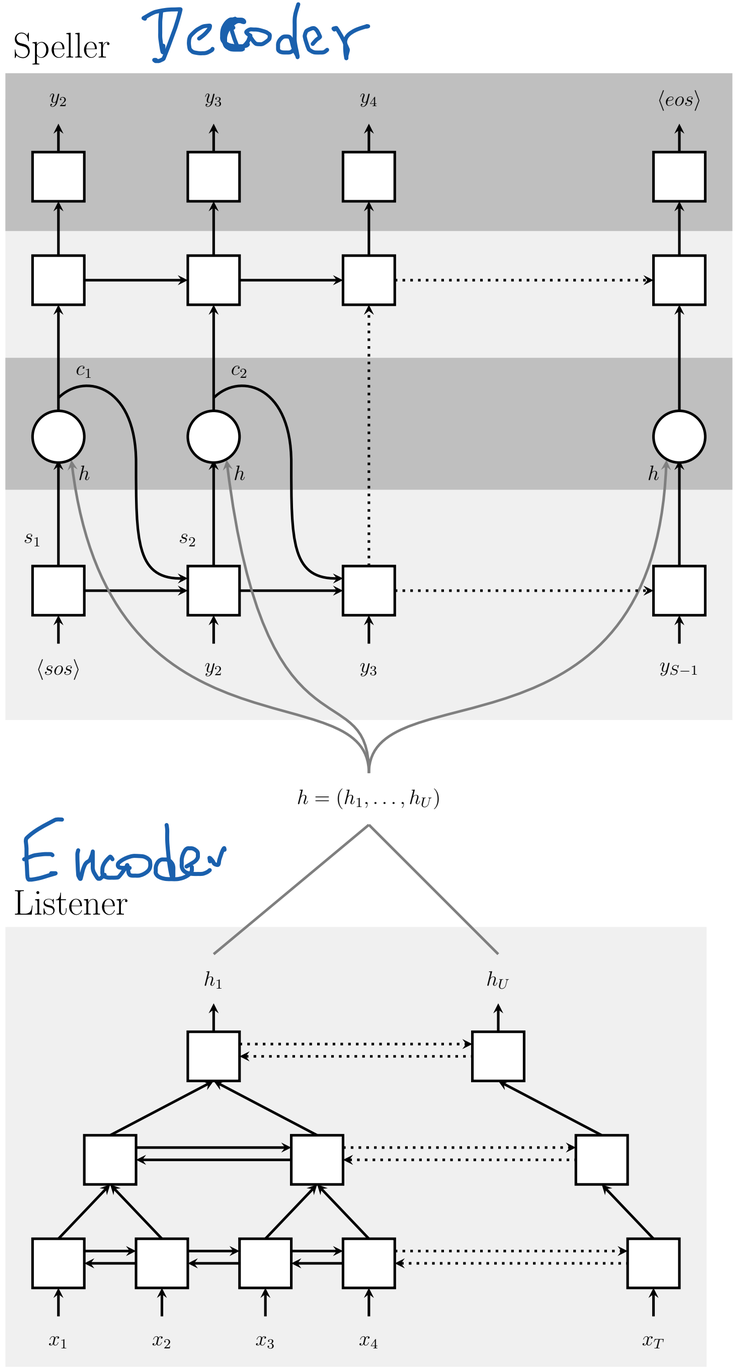
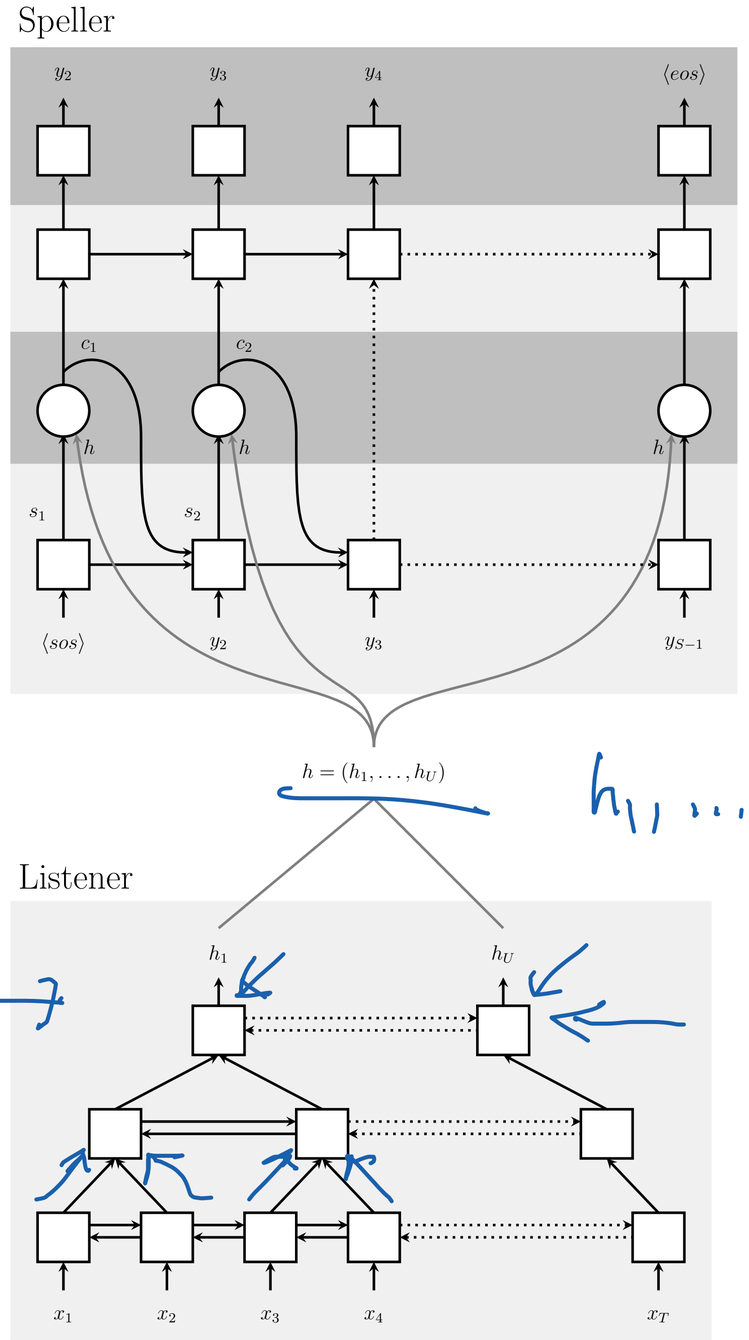


Image from: Chan et al., Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, ICASSP 2016

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.

$h_1, \ldots h_U$

for a typical BiLSTM :

Output at the t^{th} time-step from the j^{th} layer

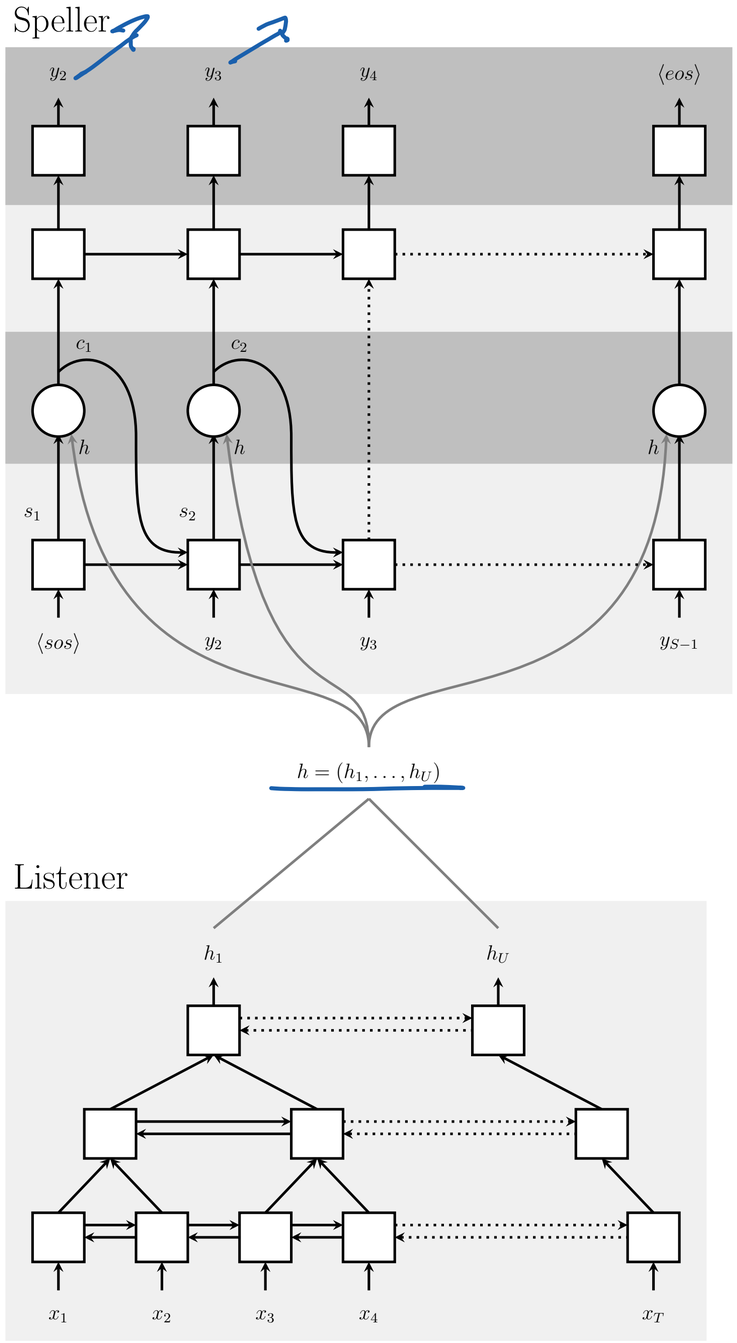
$$h_t^j = \text{BiLSTM}(h_{t-1}^j, h_t^{j-1})$$

For a pyramidal BiLSTM:

$$\mathbf{h} = \text{Listen}(\mathbf{x})$$

$$h_t^j = \text{PyBiLSTM}(h_{t-1}^j, [h_{2t}^{j-1}, h_{2t+1}^{j-1}])$$

Image from: Chan et al., Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, ICASSP 2016



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 ($\mathrm{AttendAndSpell}$): An attention-based decoder. Consumes $\mathbf{h}$ and produces a probability distribution over characters.

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

Image from: Chan et al., Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, ICASSP 2016

Attend and spell

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

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

decoder state

seq of encoder states

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

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

ϕ, ψ being MLPs

Training and Decoding

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

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

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

Maintain a set of (B) hypothesis

Expand each hyp in the beam with every possible char & retain the B most likely hyps.

Experiments

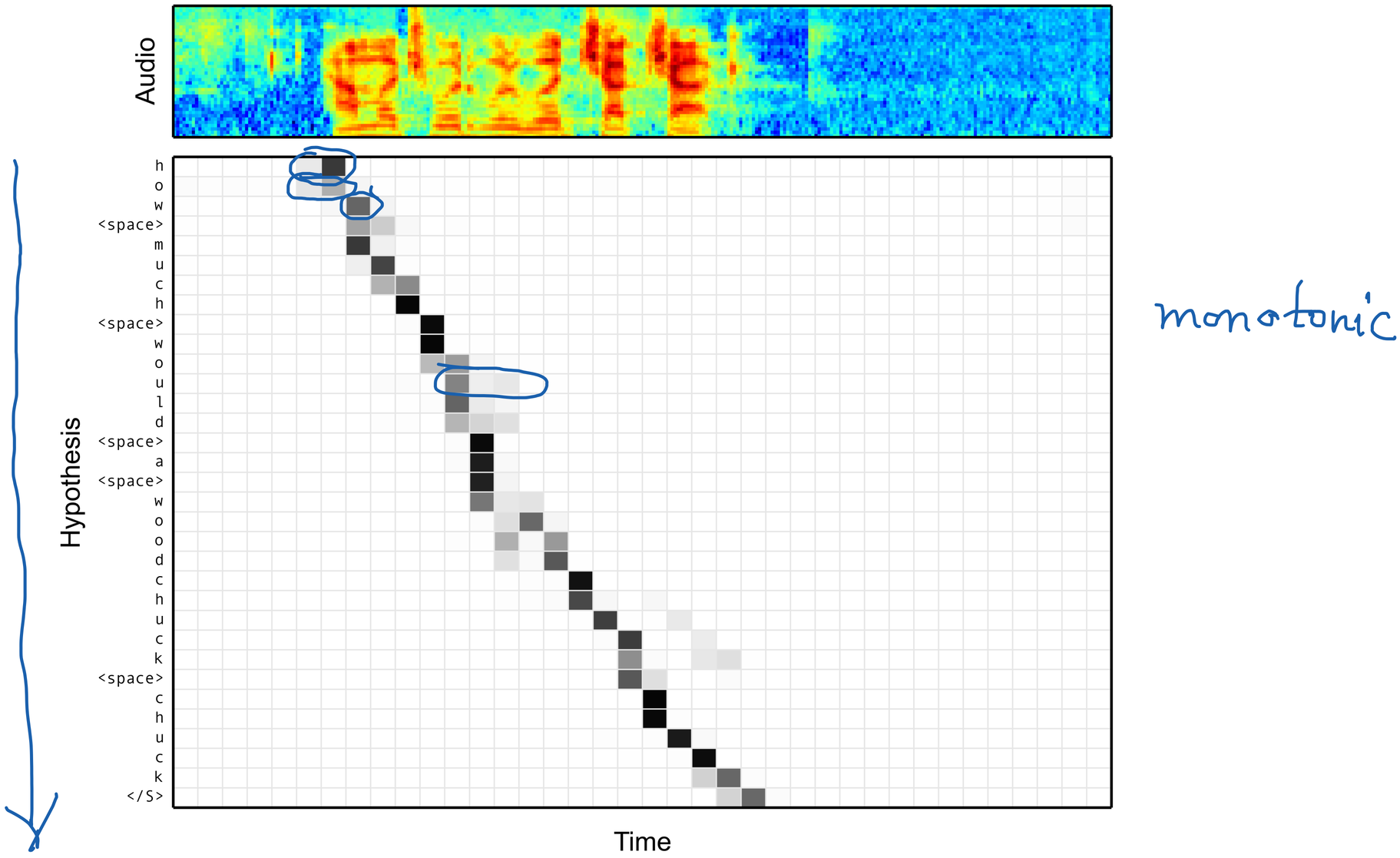
Table 1: WER comparison on the clean and noisy Google voice search task. The CLDNN-HMM system is the state-of-the-art, the Listen, Attend and Spell (LAS) models are decoded with a beam size of 32. Language Model (LM) rescoring can be beneficial.

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

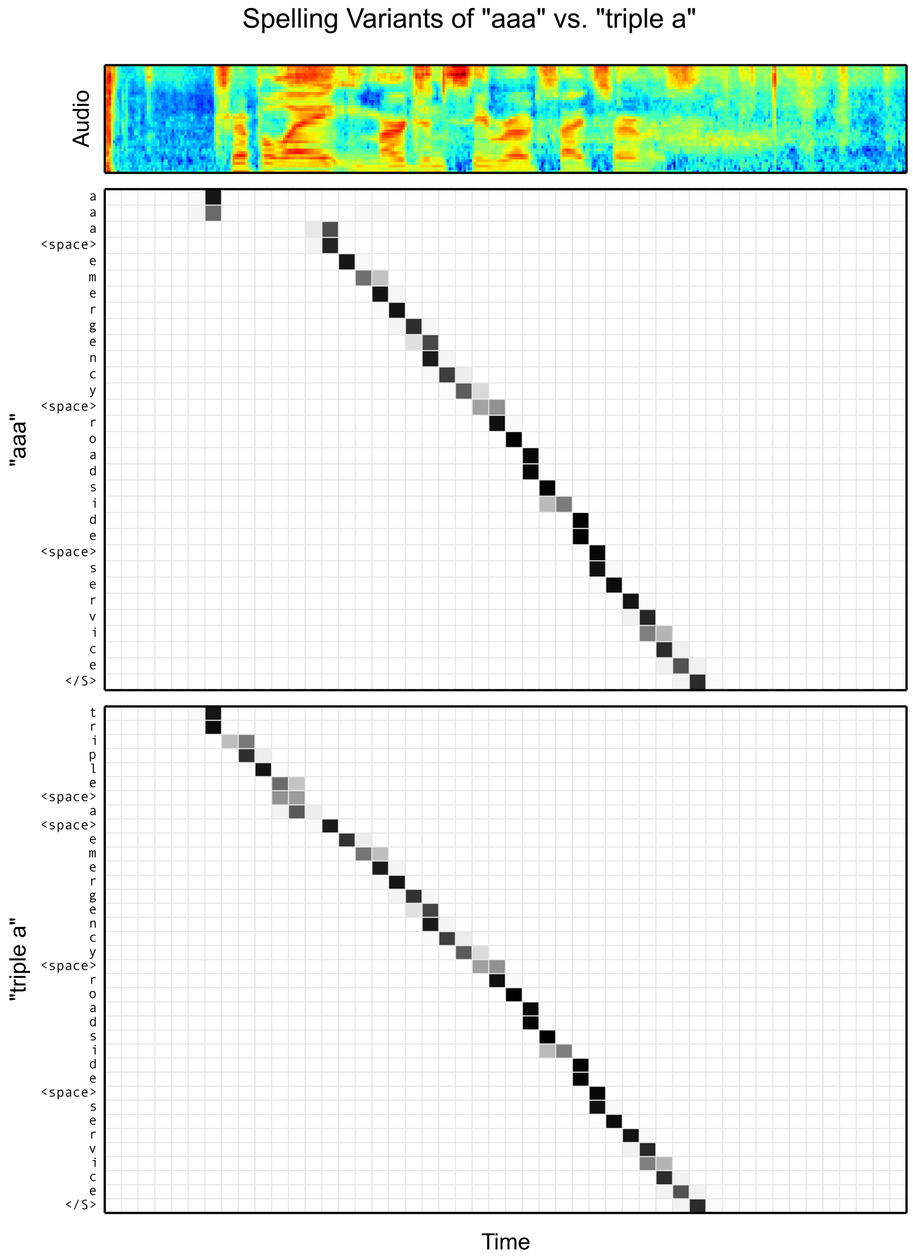
Alignment between the Characters and Audio



monotonic

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"

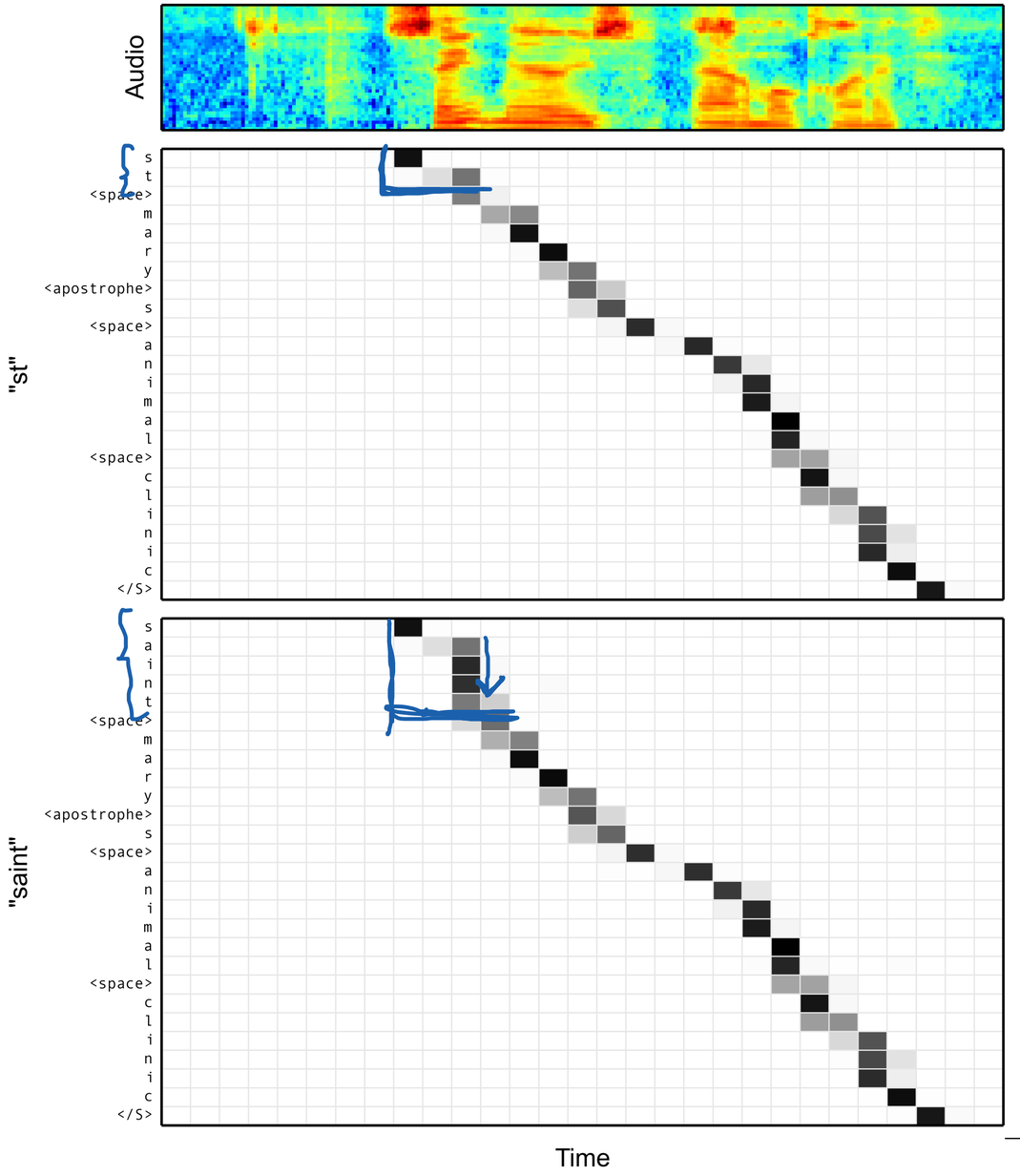


Image from: https://arxiv.org/pdf/1508.01211.pdf