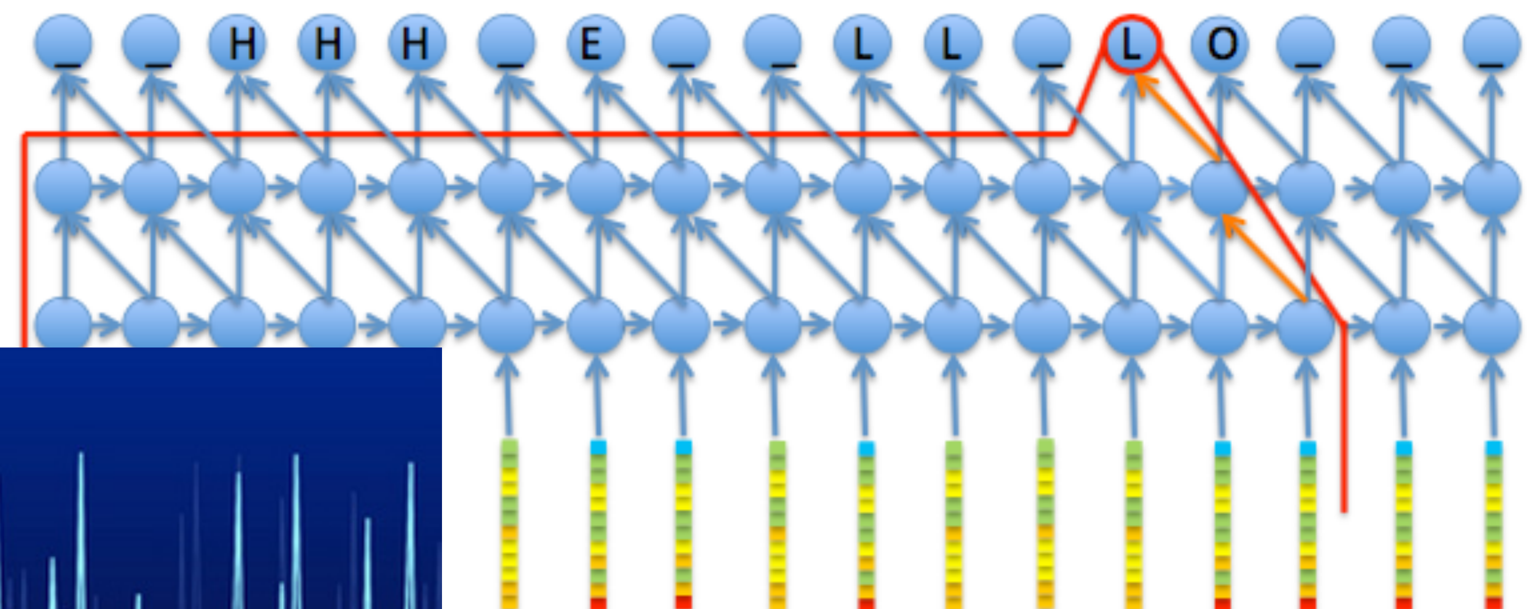
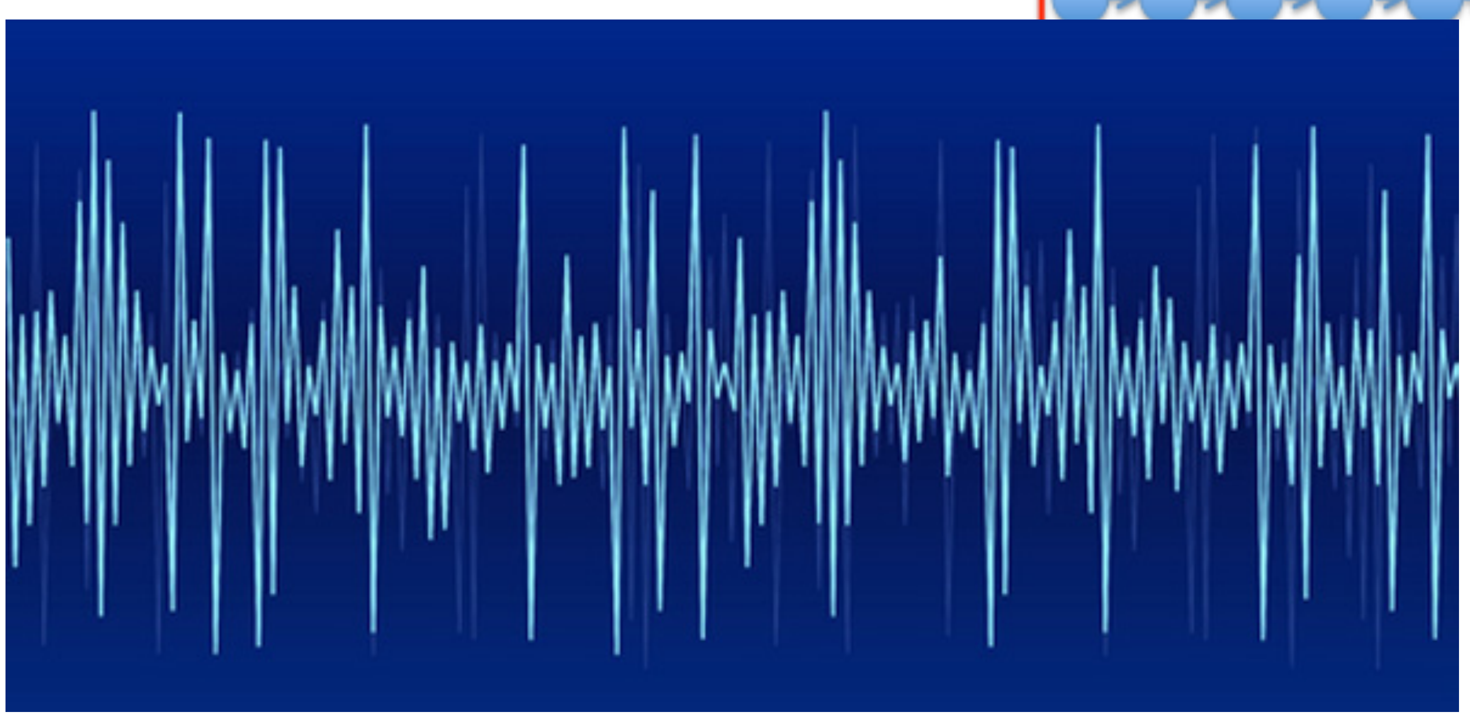
Speech Recognition with Deep Learning

Patrick Foldenauer

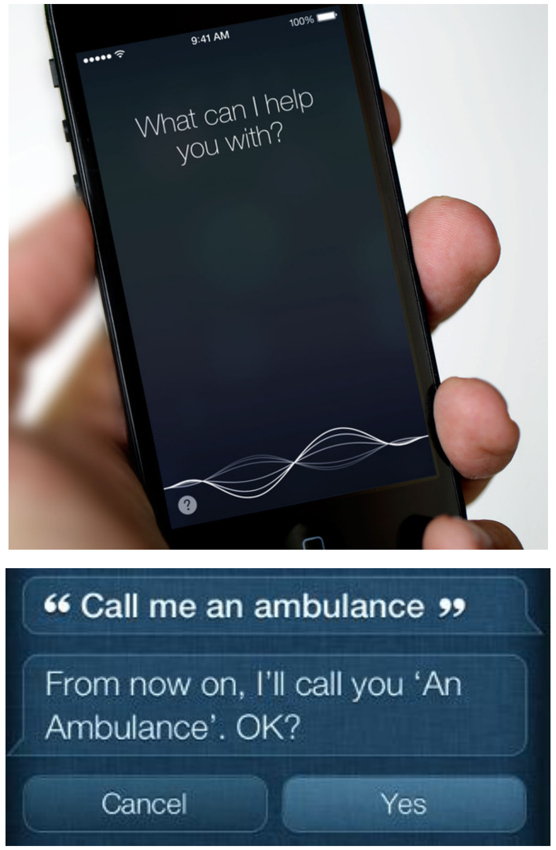
based on lecture by Adam Coates at Bay Area Deep Learning School 2016





- Motivation
- ASR Workchain
- Deep Learning ASR
    - Preprocessing
    - Connectionist Temporal Classification (CTC)
    - Decoding
- Example/Summary

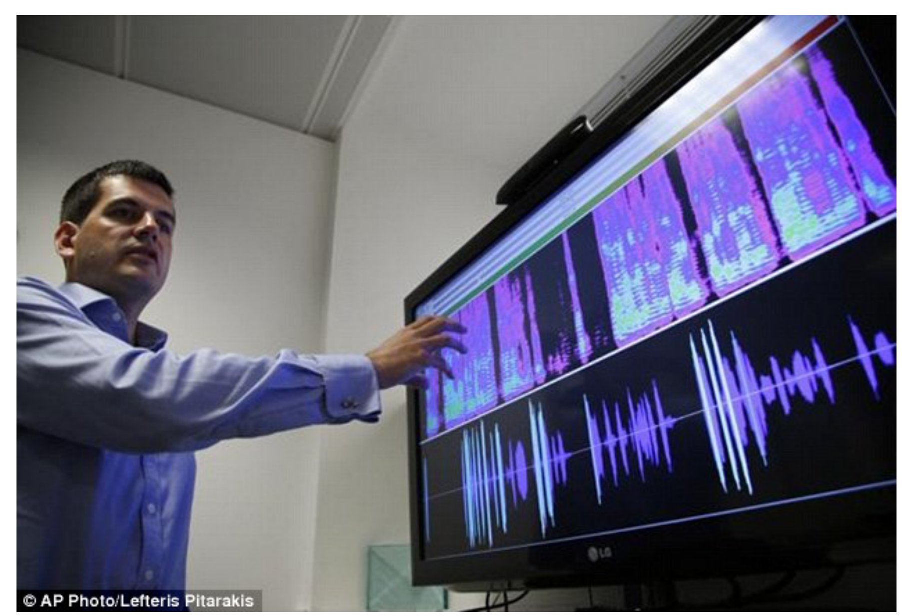
Speech recognition applications



Mobile devices


Hands-free interaction

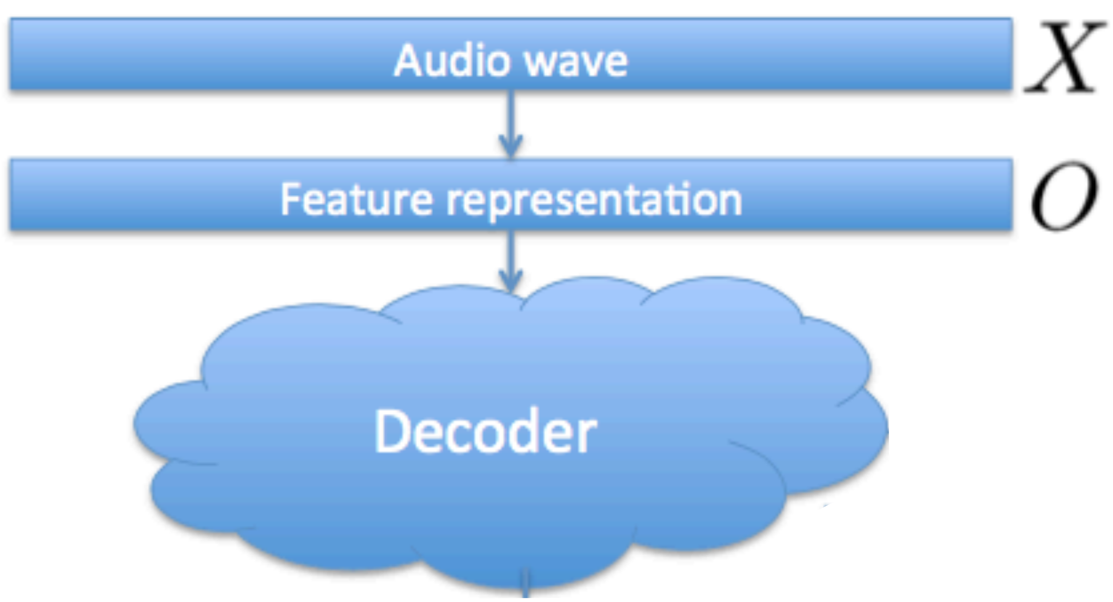
Content captioning



Voice verification

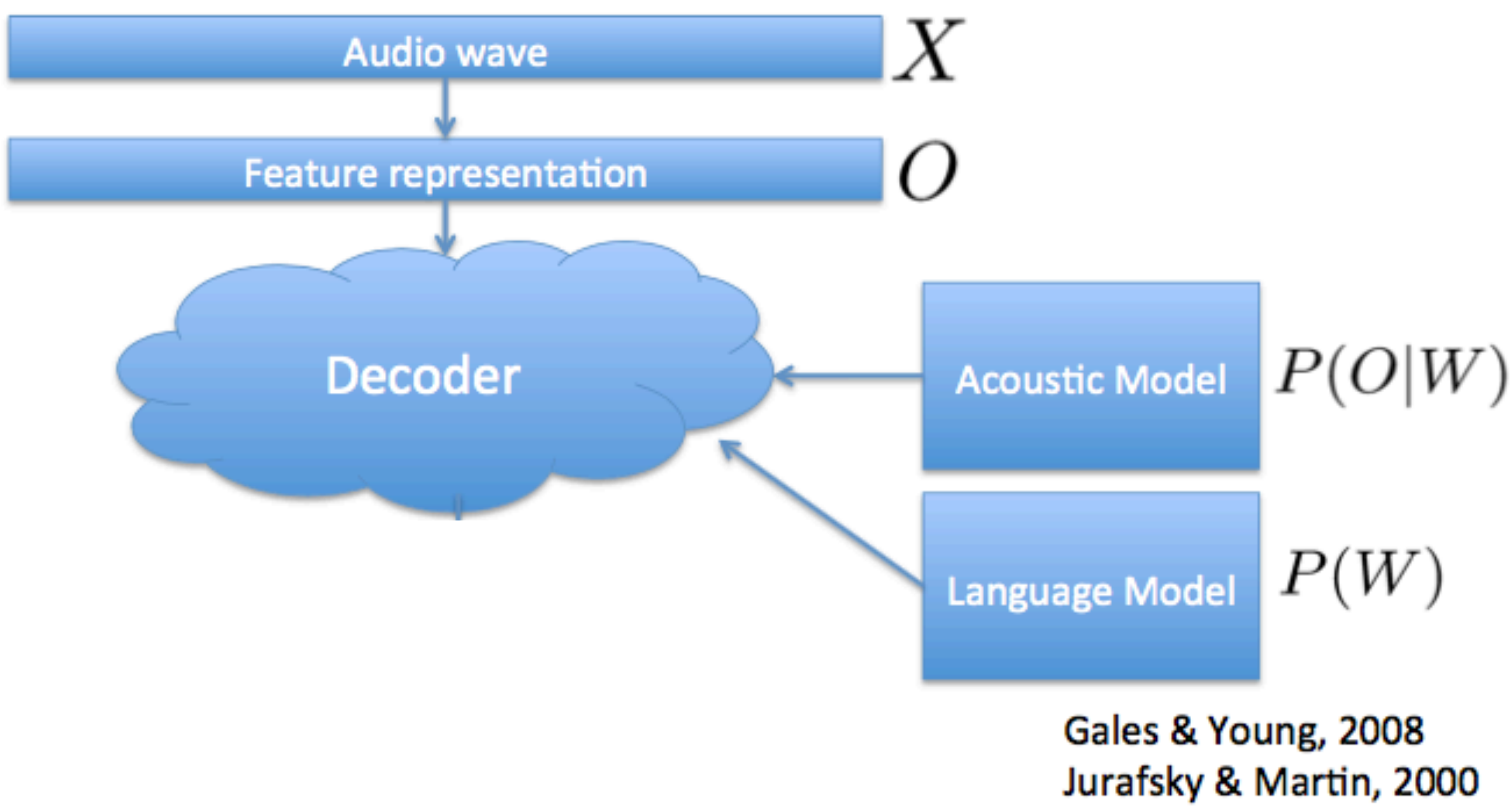
ASR Workchain

- Split up the task into different components:



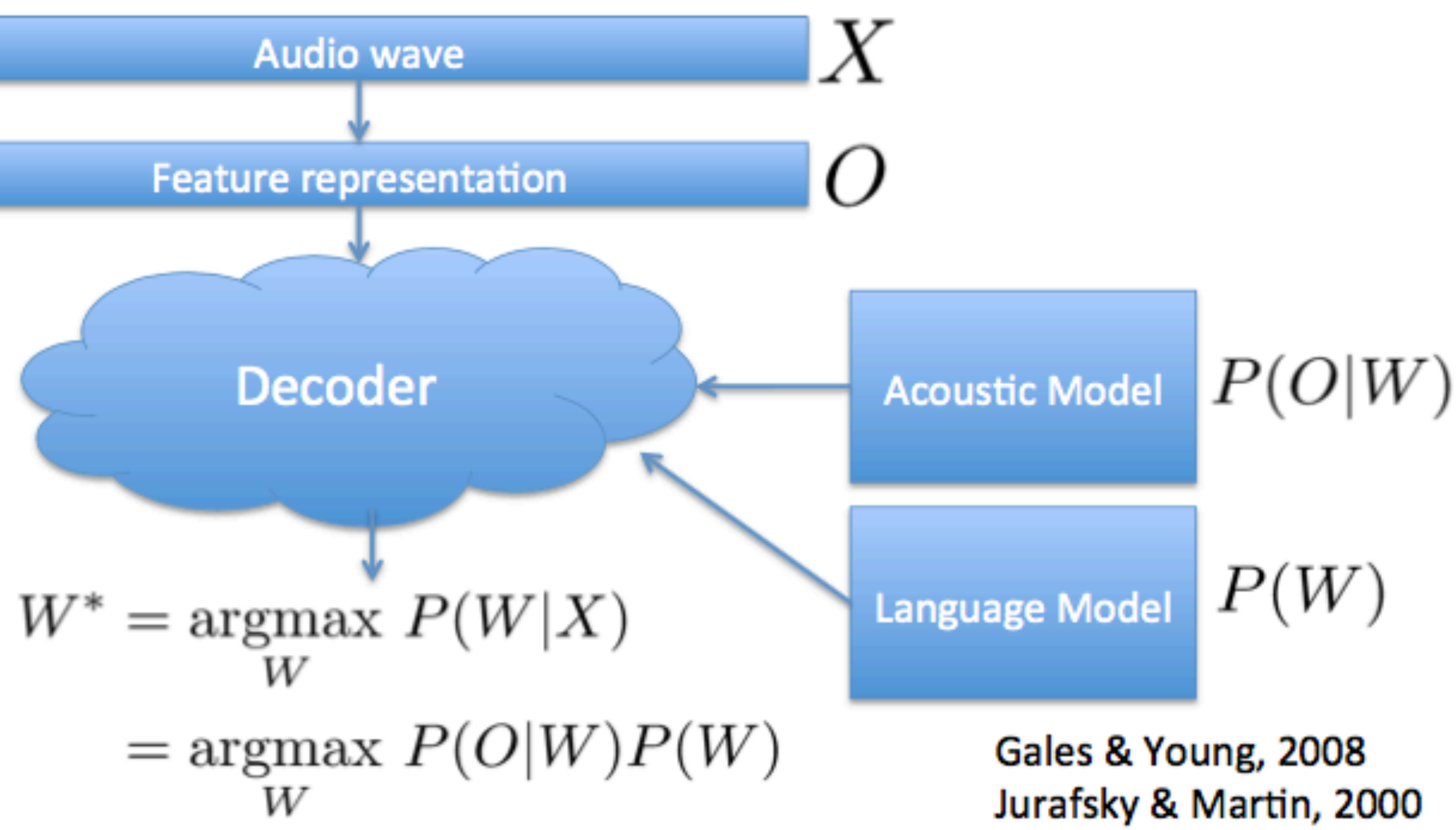
ASR Workchain

- Split up the task into different components:



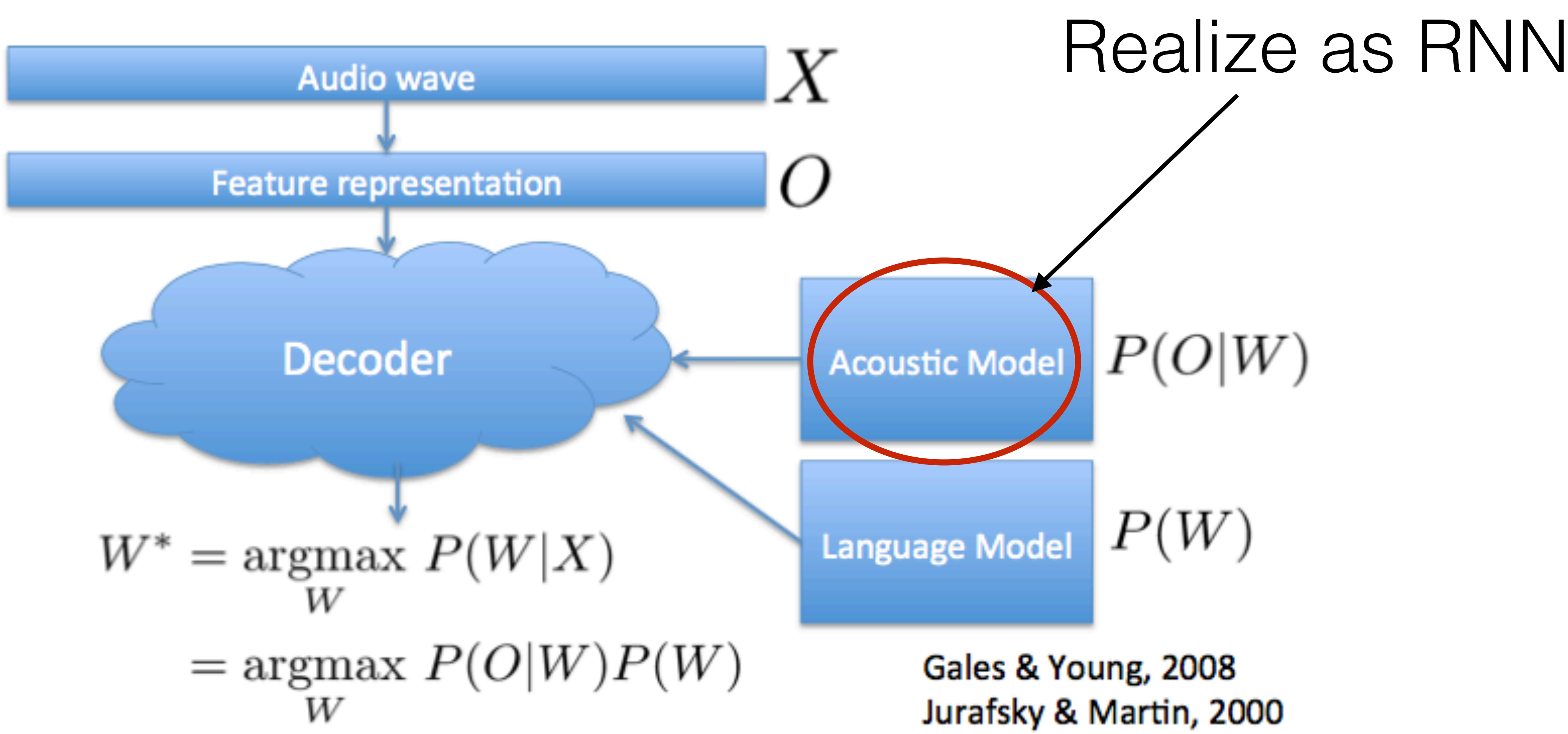
ASR Workchain

- Split up the task into different components:

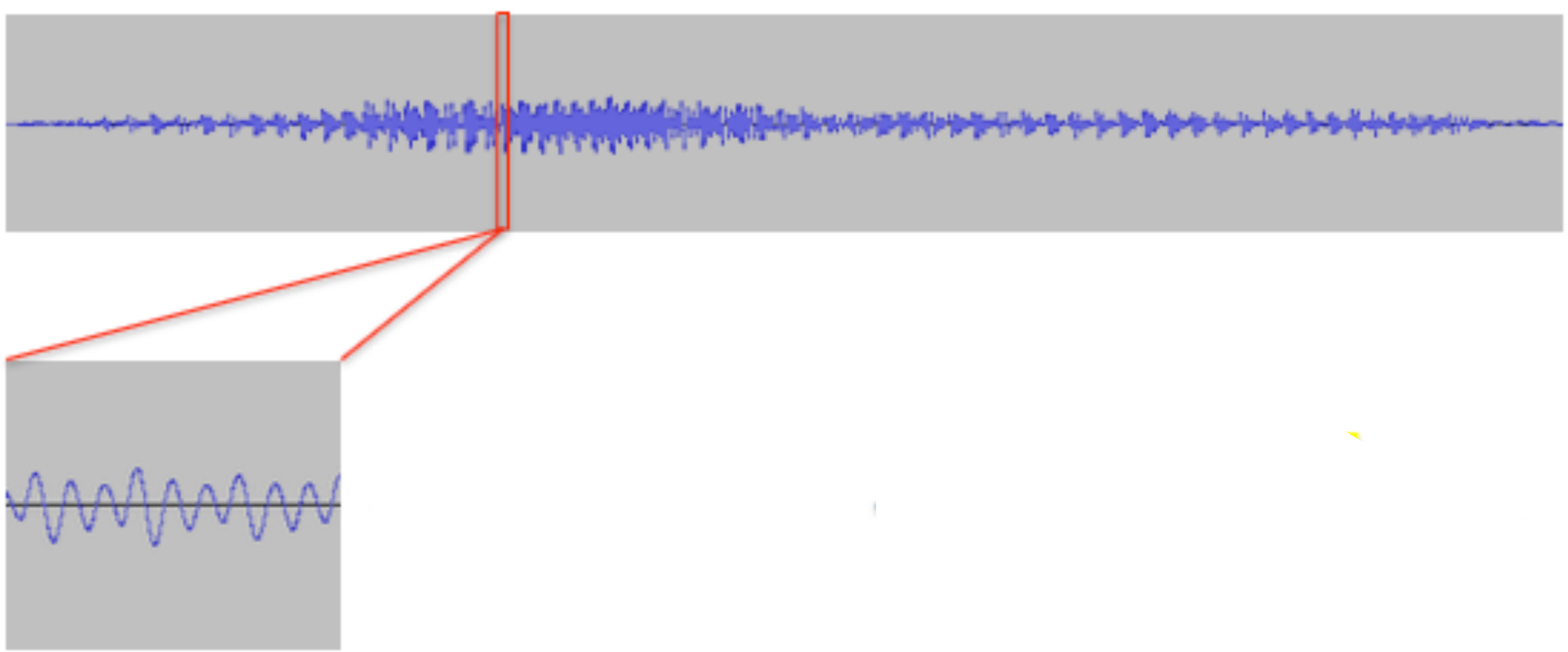


ASR Workchain

- Split up the task into different components:



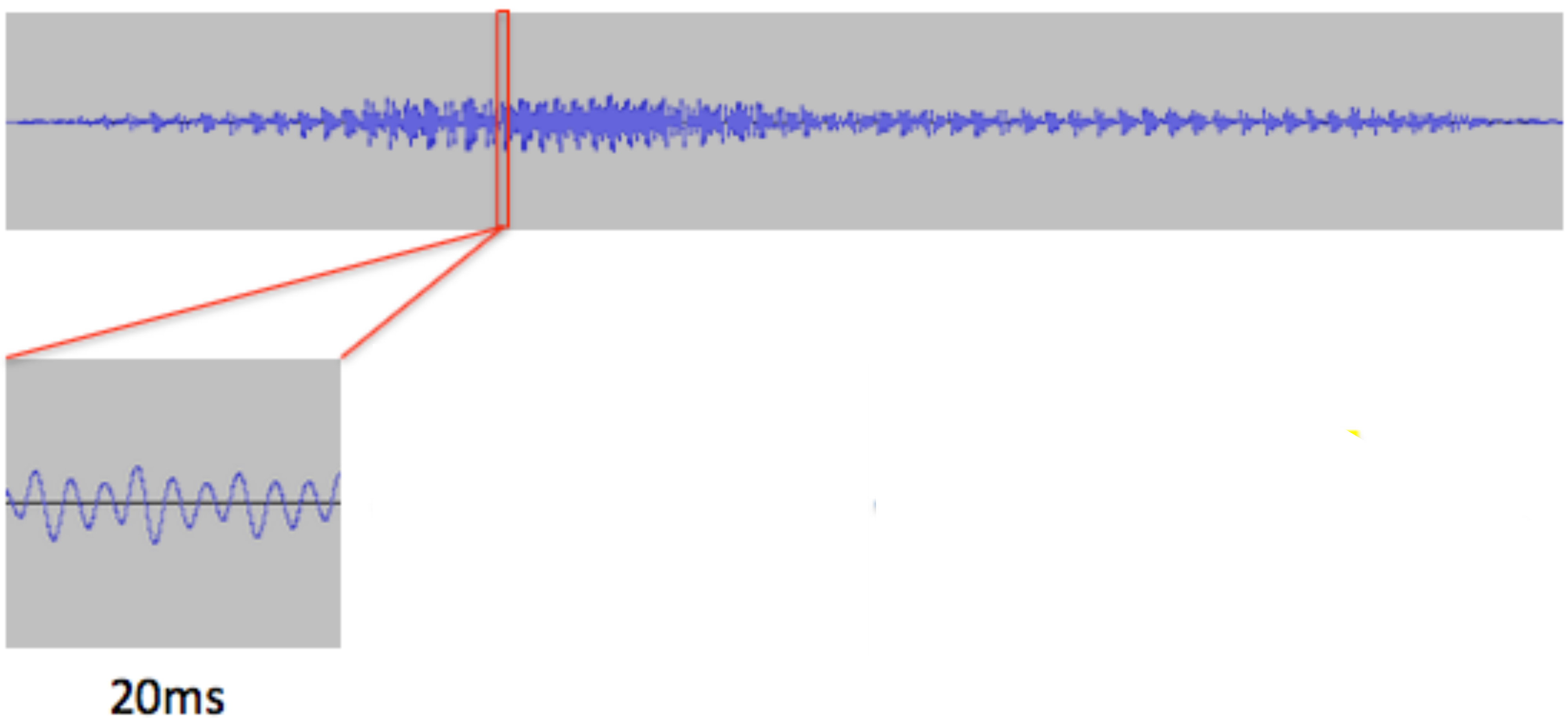
Preprocessing



20ms

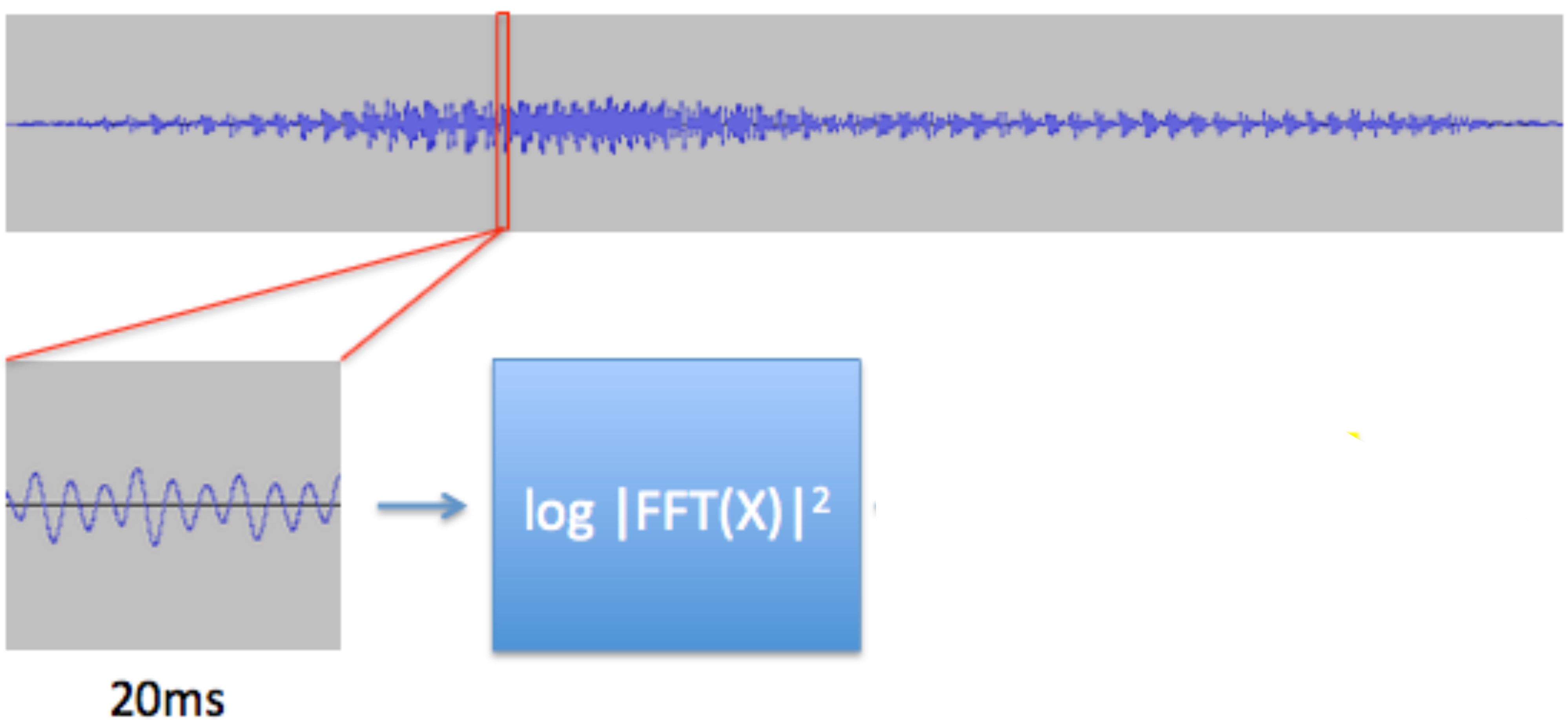
Preprocessing

- Sample audio signal/“discretize”



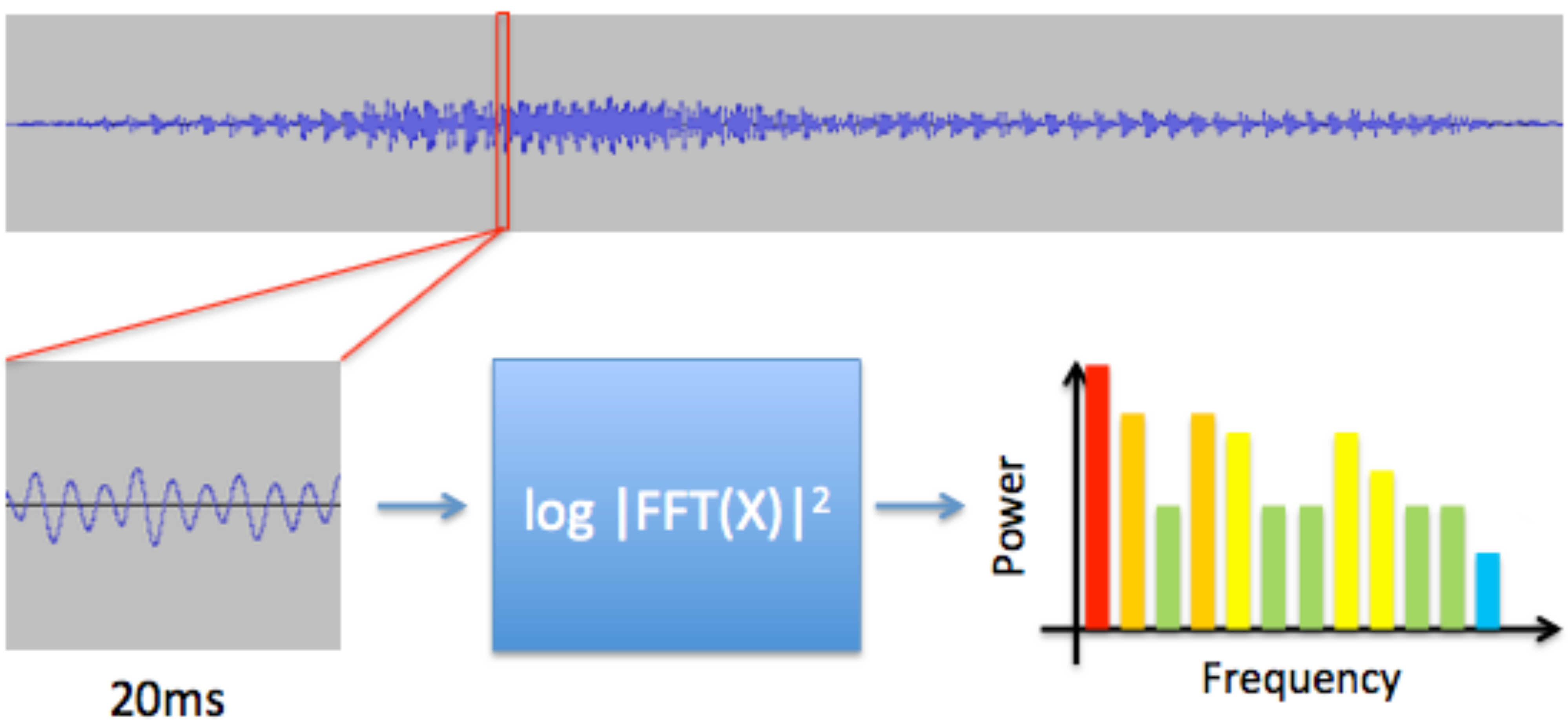
Preprocessing

- Sample audio signal/“discretize”
- Extract power spectrum (i.e. the features)



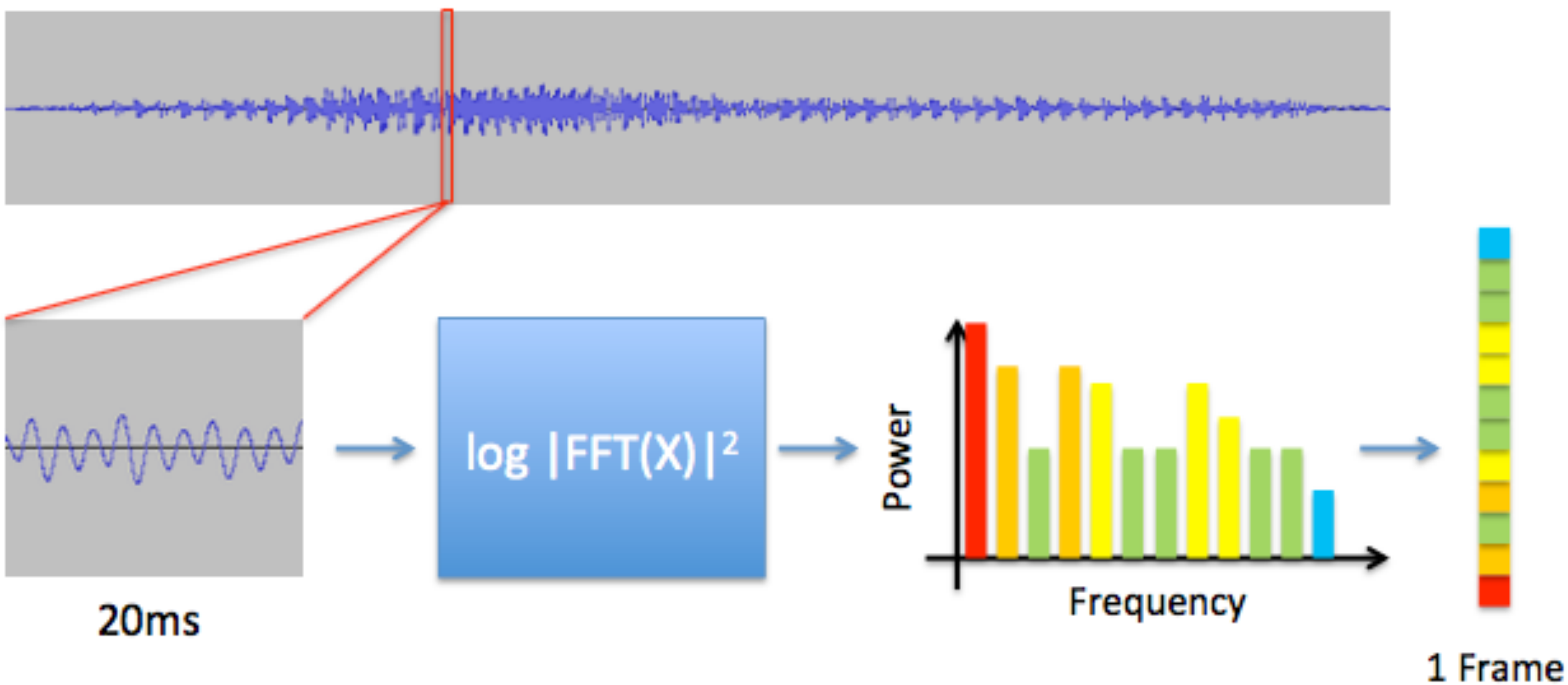
Preprocessing

- Sample audio signal/“discretize”
- Extract power spectrum (i.e. the features)



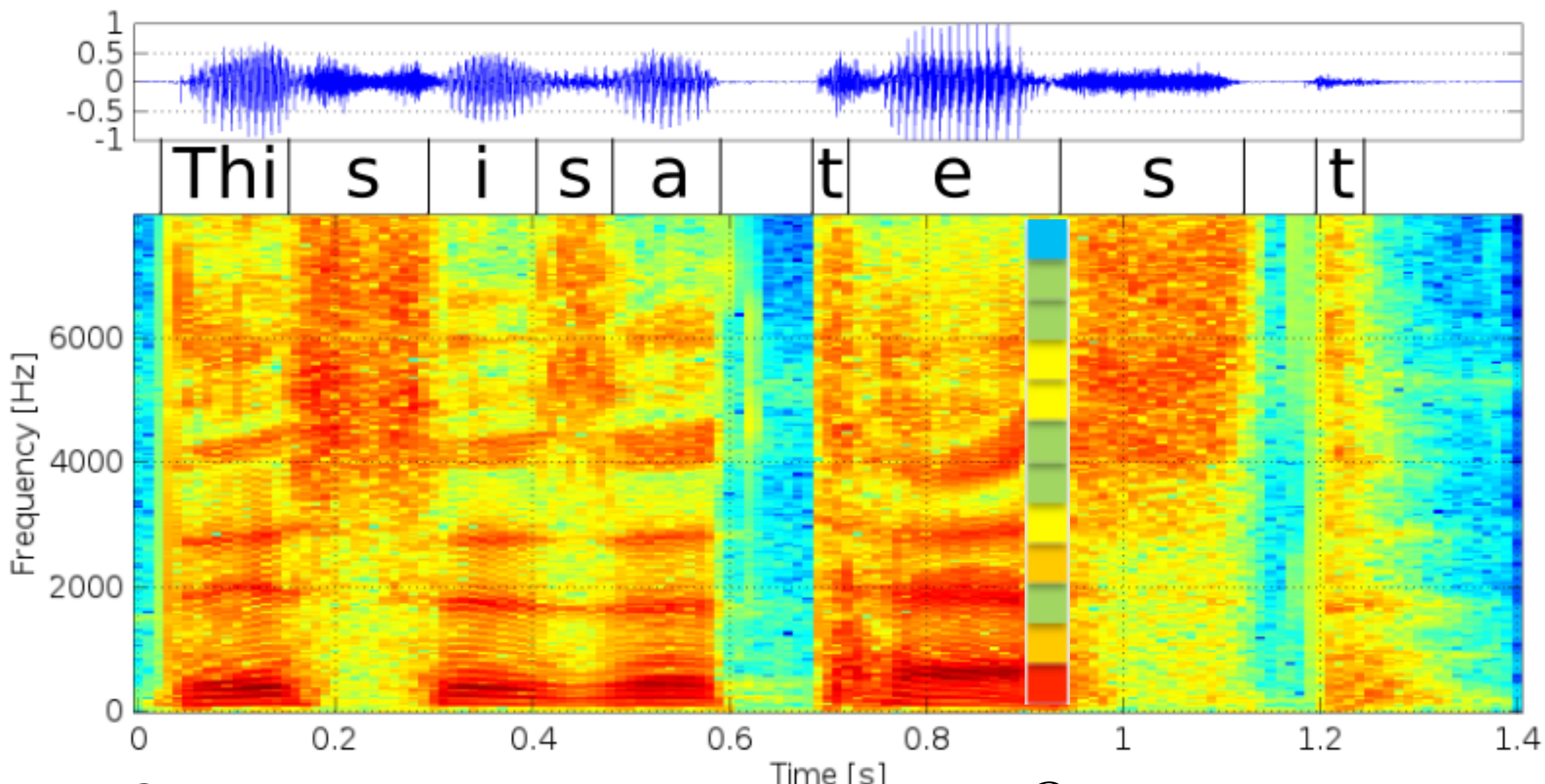
Preprocessing

- Sample audio signal/"discretize"
- Extract power spectrum (i.e. the features)
- Construct local feature vector from power spectrum



Preprocessing

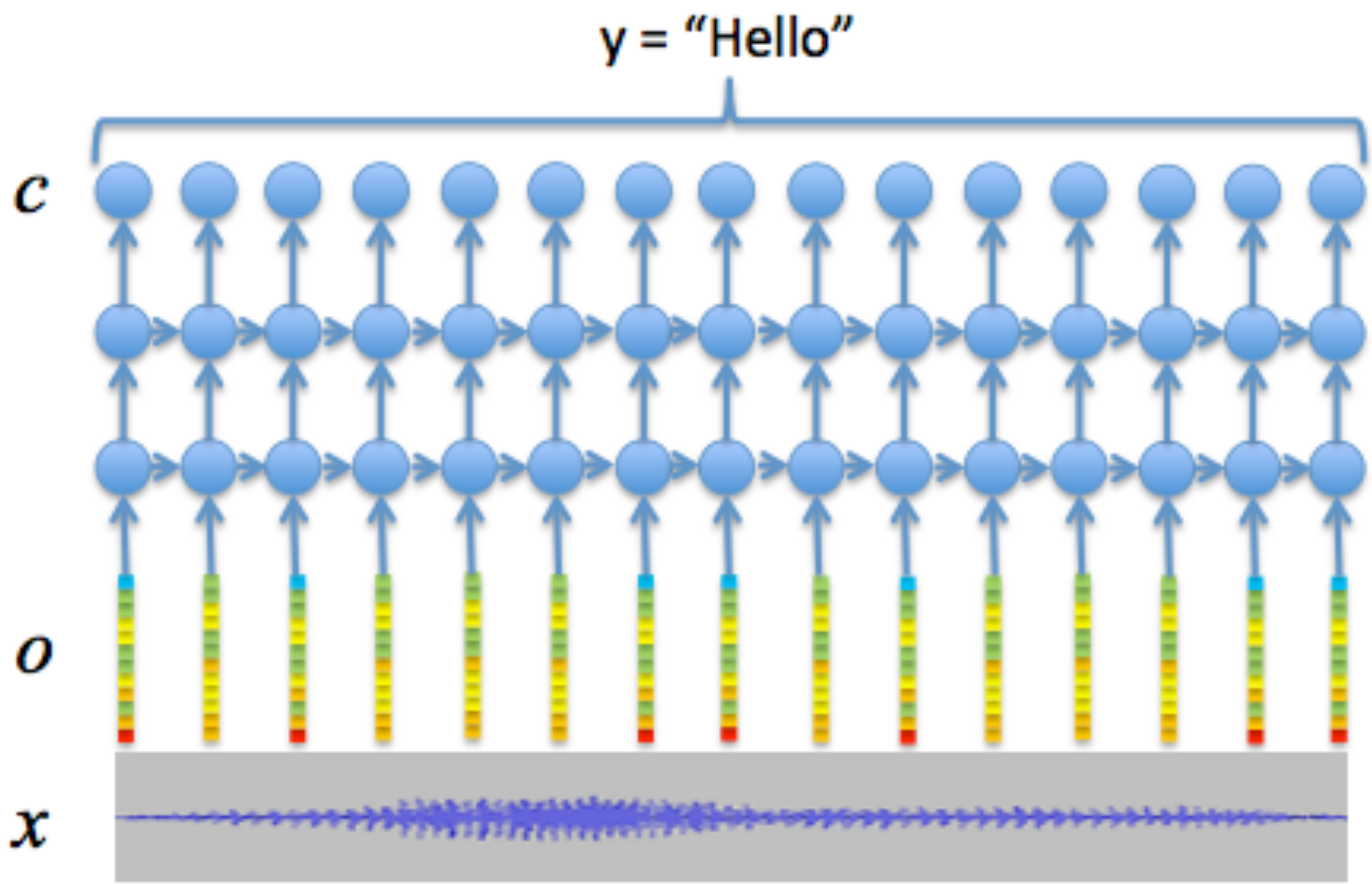
Generate full spectrogram for the audio sequence



O_1 $\dots$ O_t $\dots$ O_T

RNN Acoustic Model

Create RNN that outputs for sequence x transcription y



Connectionist Temporal Classification

Main issue: $\mathrm{length}(y) \leq \mathrm{length}(x)$

Connectionist Temporal Classification

Main issue: $\text{length}(y) \leq \text{length}(x)$

Addressed by CTC [Graves *et al.*,2006]:

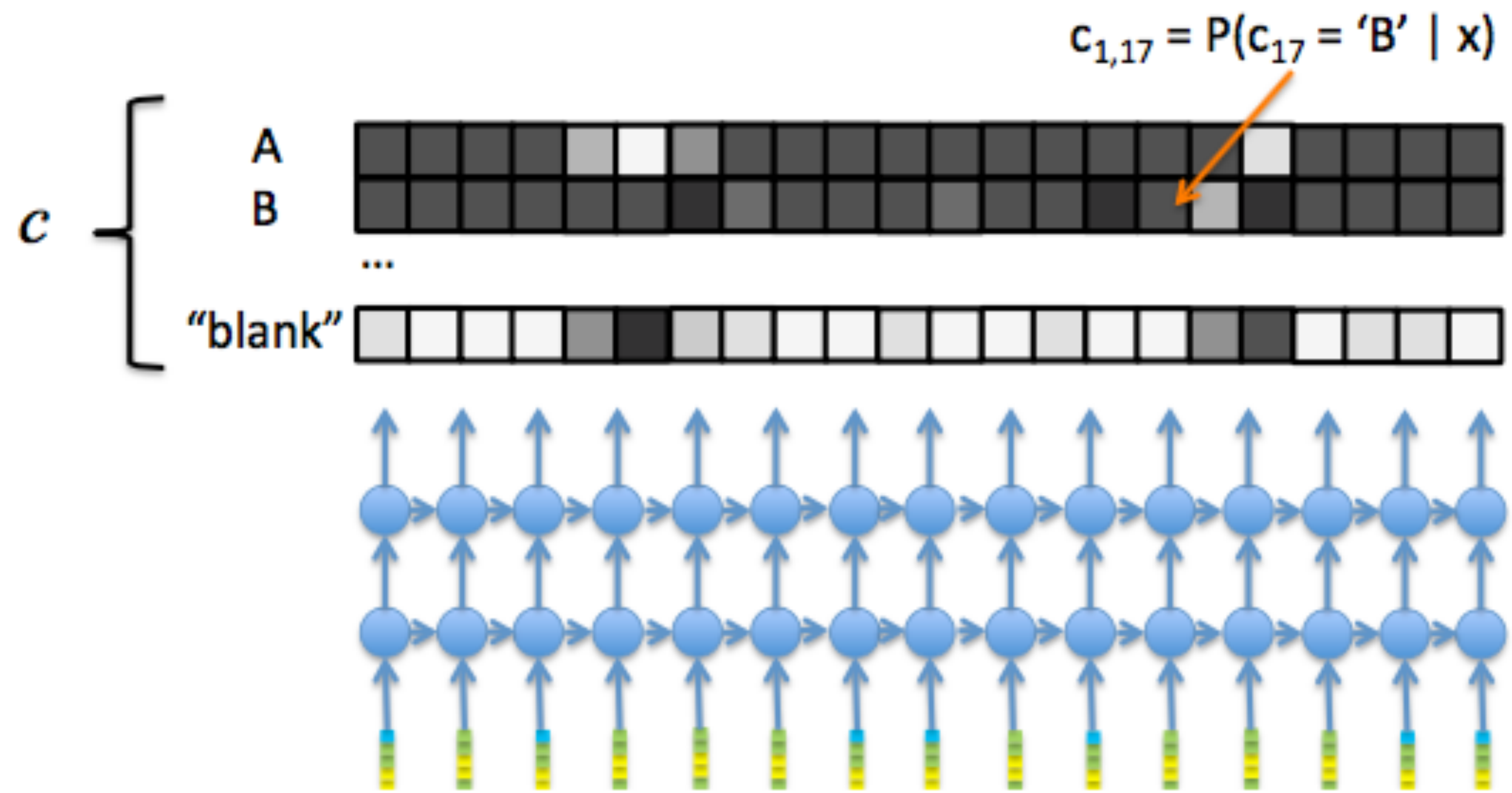
1. Output neurons *c* encode distribution over letters: $(\text{length}(c) = \text{length}(x)):$

$$c \in \{A, B, C, \ldots, Z, \text{blank}, \text{space}\}$$

2. Define a mapping $\beta(c) \rightarrow y$
3. Maximize the likelihood of true labels y^* for given input x under this model

Connectionist Temporal Classification

1. RNN output:

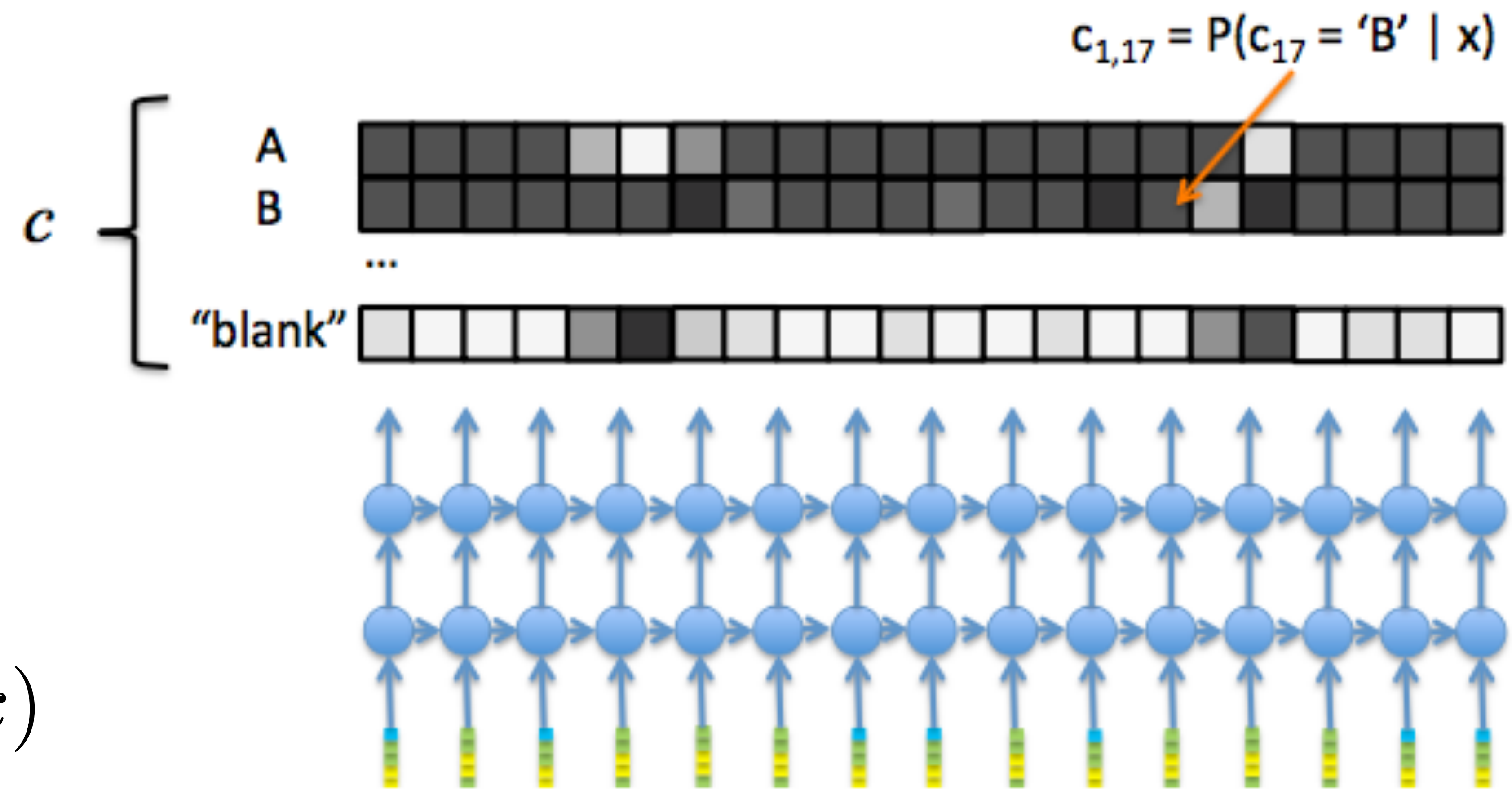


Connectionist Temporal Classification

1. RNN output:

Under assumption of independence, output defines distribution over whole sequences of characters *c*:

$$P(c|x) = \prod_{i=1}^{N} P(c_i|x)$$



$$P(c = \text{HHH\_E\_\_LL\_LO\_\_\_}|x) = P(c_1 = \text{H}|x)P(c_2 = \text{H}|x)\ldots P(c_1 = \text{blank}|x)$$

Connectionist Temporal Classification

2. Define mapping:

- Eliminate duplicate characters then remove blanks:

$$\beta(c = \mathrm{HHH\_E\_\_LL\_LO\_\_\_}) = \text{"HELLO"}$$

Connectionist Temporal Classification

2. Define mapping:

- Eliminate duplicate characters then remove blanks:

$$\beta(c = \text{HHH\_E\_\_LL\_LO\_\_\_}) = \text{"HELLO"}$$

- Mapping implies distribution over *transcriptions y:*

$P(c\|x) = \{$ 0.1	HHH_E__LL_LO___	"HELLO" v.1
0.02	HH__E__LL_LO___	"HELLO" v.2
0.01	HHH_E__L_L␣OH__	"HELL OH"
0.01	HHH_EE_LL_L_O__	"HELLO" v.3
…	YY__E__LL_LO_W_	"YELLOW"

$$P(y|x) = \sum_{c:\beta(c)=y} P(c|x)$$

$P(\text{"HELLO"}) = 0.1 + 0.02 + 0.01 + \ldots$

Connectionist Temporal Classification

3. Maximize likelihood:

Update model parameters θ such that correct label y^* maximizes the log likelihood:

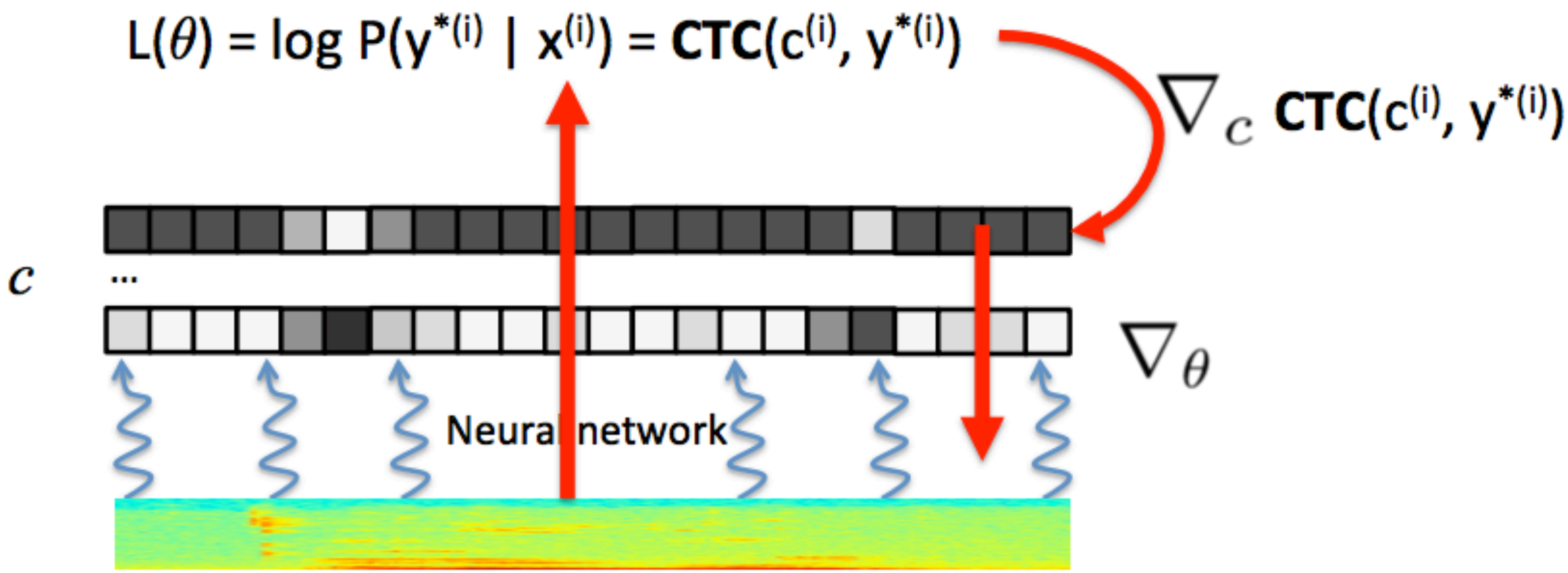
$$\theta^\star = \arg\max_\theta \sum_i \log P(y^{\star(i)}|x^{(i)})$$
$$= \arg\max_\theta \sum_i \log \sum_{c:\beta(c)=y^{\star(i)}} P(c|x^{(i)})$$

Connectionist Temporal Classification

How do we maximize:

- Use gradient descent methods on the CTC objective function (based on log likelihood)

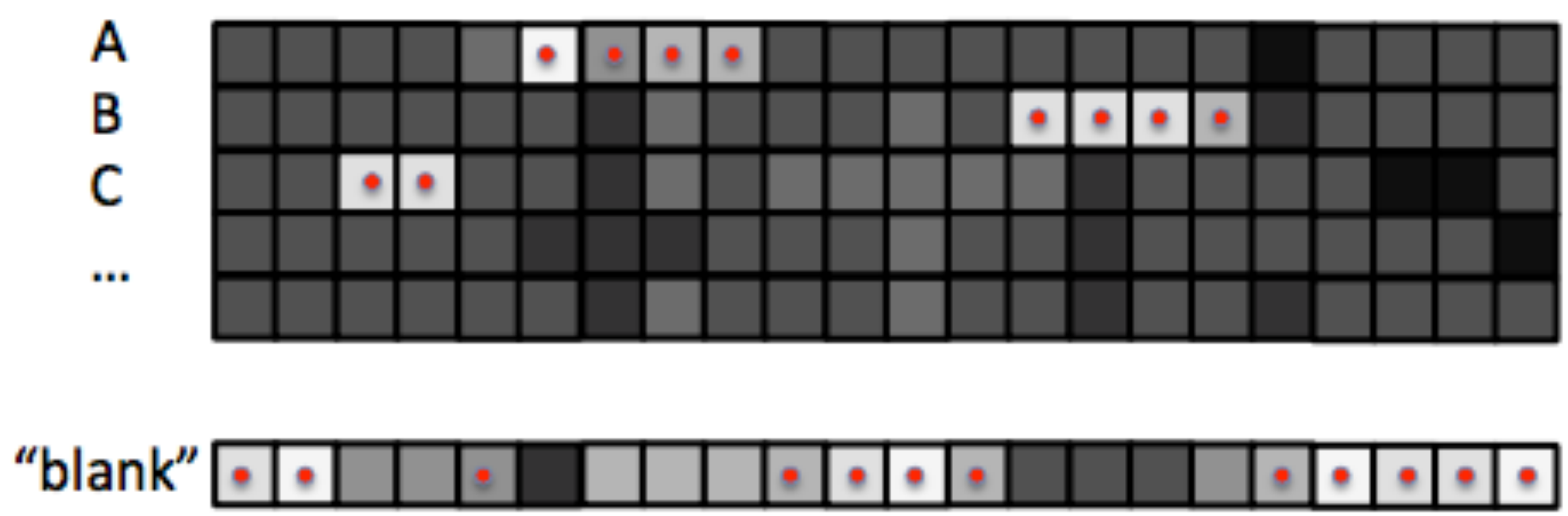
- Backpropagate error and adjust model parameters θ



Decoding

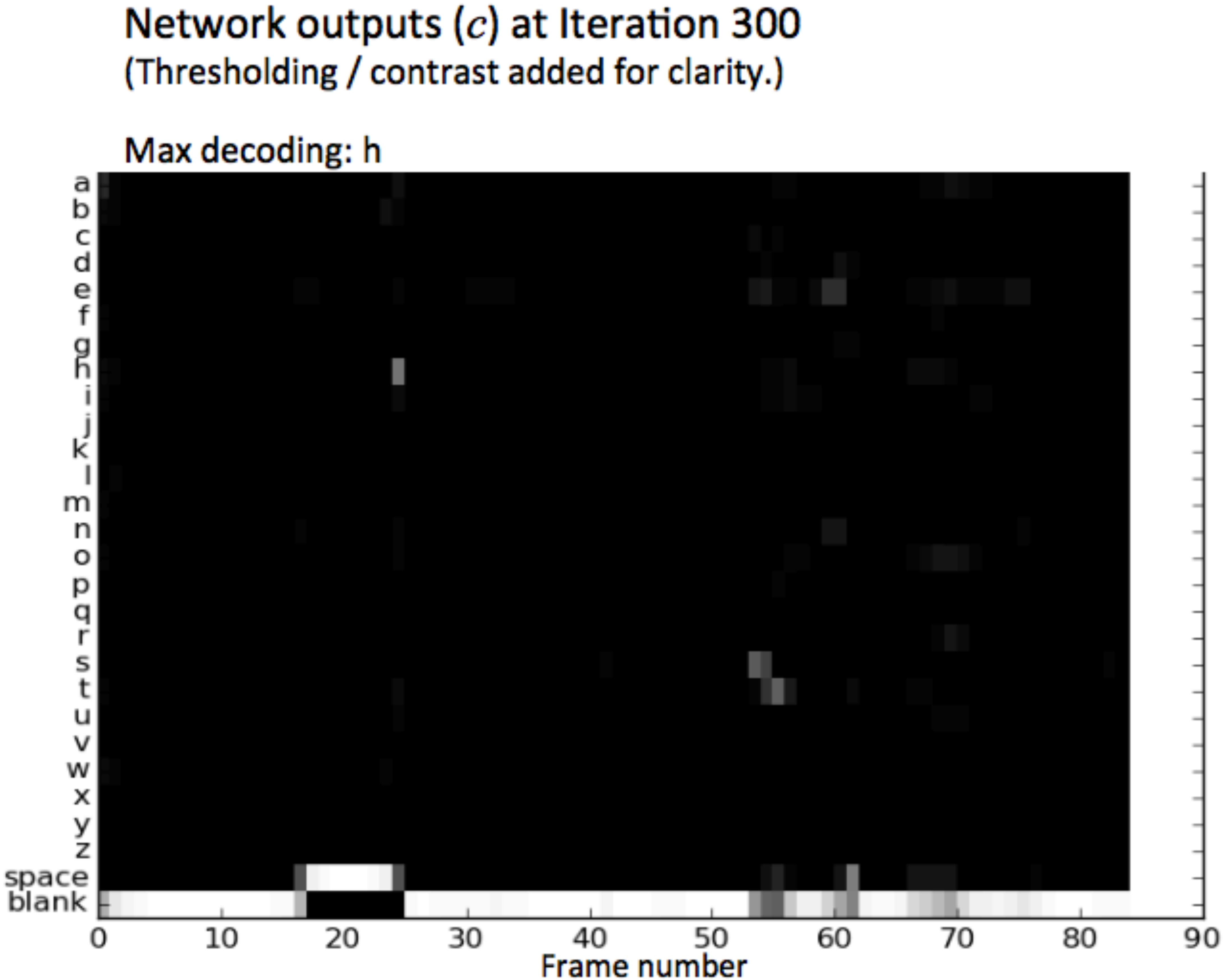
- For unseen data trained RNN outputs $P(c|x)$
- How do we find "most likely" transcription *y* ?
  ➡ Simplest method: "max decoding"

$$\beta\left(\arg\max_{c} P(c|x)\right)$$



"CAB"

Example:



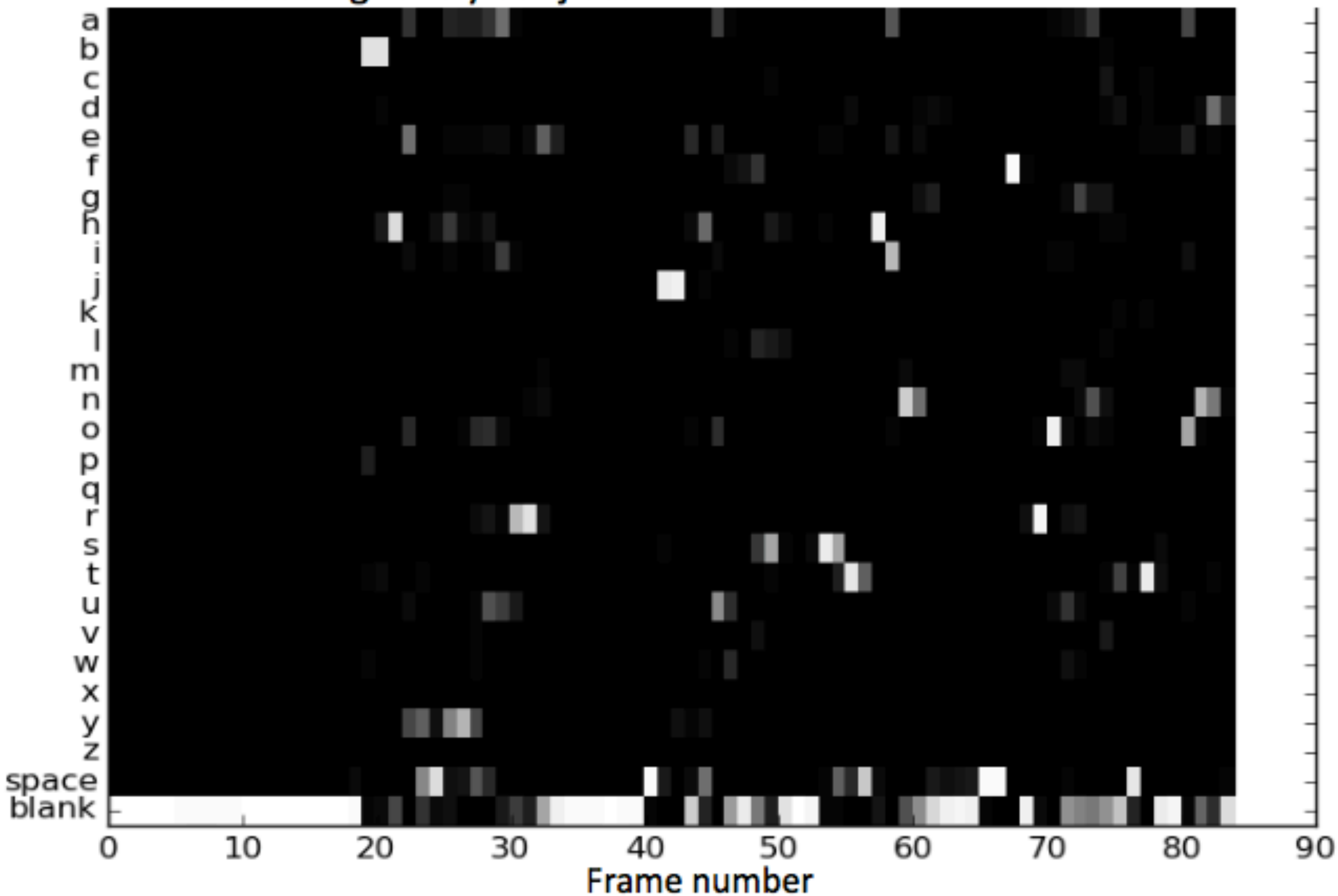
Network outputs (c) at Iteration 300
(Thresholding / contrast added for clarity.)
Max decoding: h
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
space
blank
0
10
20
30
40
50
60
70
80
90
Frame number

Example:

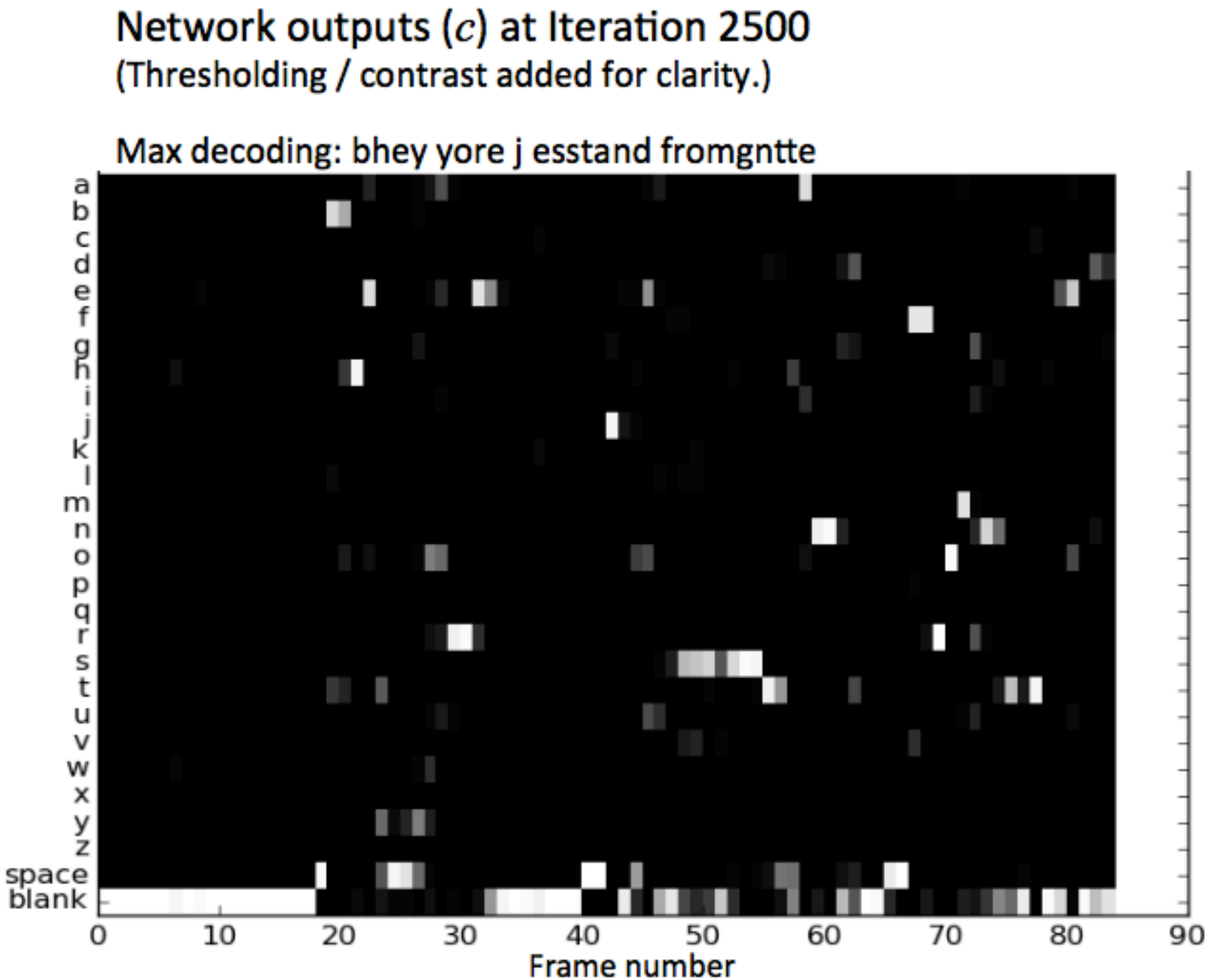
Network outputs (c) at Iteration 1500

(Thresholding / contrast added for clarity.)

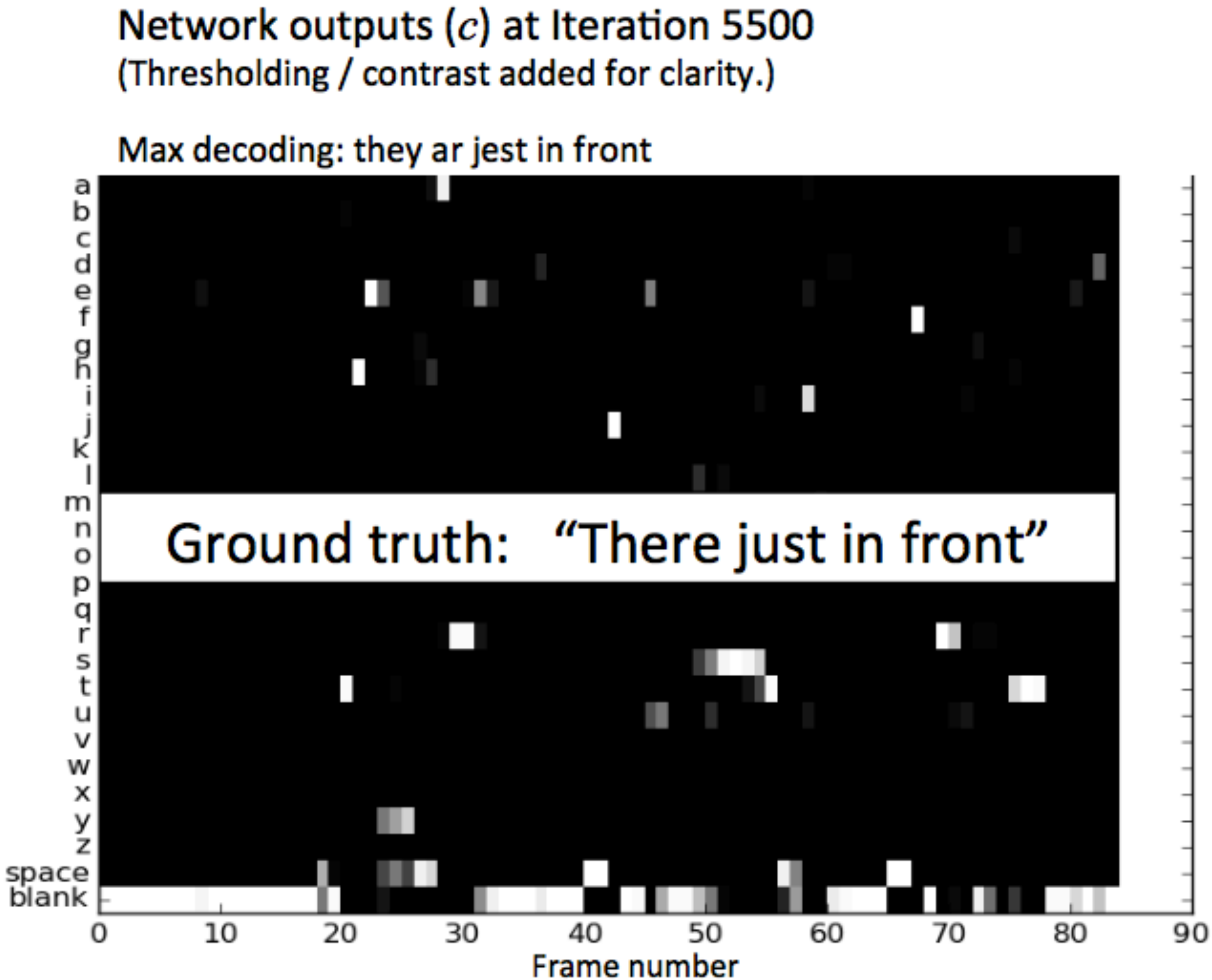
Max decoding: bhe y uar j usst hin fro ton



Example:


Network outputs (c) at Iteration 2500
(Thresholding / contrast added for clarity.)
Max decoding: bhey yore j esstand fromgntte
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
space
blank
0
10
20
30
40
50
60
70
80
90
Frame number

Example:


Network outputs (c) at Iteration 5500
(Thresholding / contrast added for clarity.)
Max decoding: they ar jest in front
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
space
blank
Ground truth: “There just in front”
0
10
20
30
40
50
60
70
80
90
Frame number

Summary

- Deep Learning is becoming more and more important for state-of-the-art speech recognition
- Success of DL RNNs based on CTC training algorithm
- Still a lot of engineering required for high accuracy speech system (language models, huge data sets, …)

References

- A Coates. “Lecture at Bay Area Deep Learning School 2016” https://youtu.be/9dXiAecyJrY?t=13874
- A Geitgey. “Machine Learning is Fun!” https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a
- A Graves, S Fernández, F Gomez, J Schmidhuber. “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks.” ICML, 2006.
- D Kriesel. “A Brief Introduction to Neural Networks” http://www.dkriesel.com/en/science/neural_networks