Neural Methods in Automatic Speech Recognition

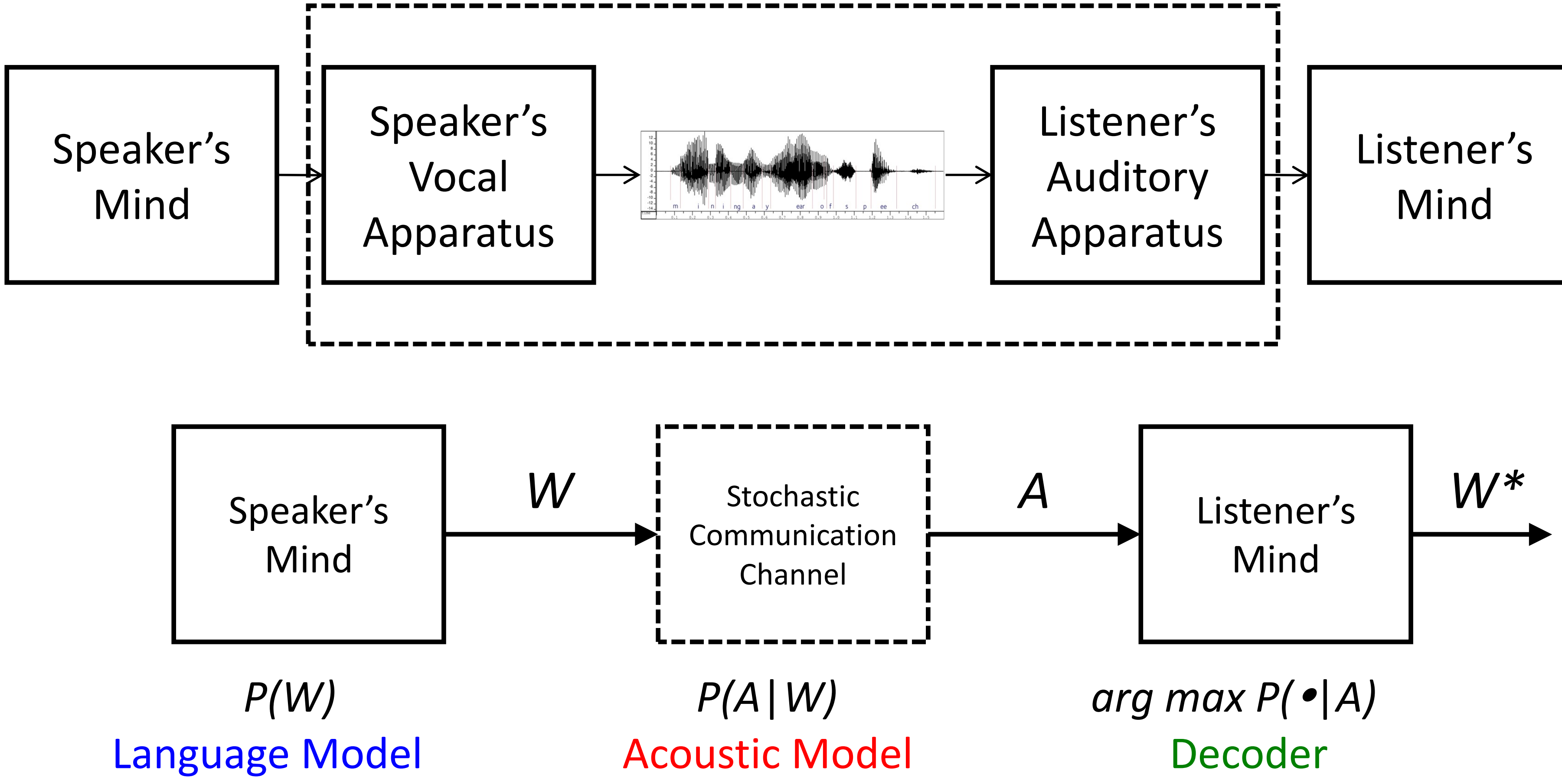
Neural Acoustic Models: Waibel et al (1988) to Povey et al (2016)

Neural Language Models: Nakamura et al (1989) to Sundermeyer et al (2012)

Connectionist Temporal Classification: Graves (2006) and Graves & Jaitly (2014)

Deep Speech 2: Hannun et al (2014) & Attention-Based Models: Chorowski et al (2015)

The "source-channel" model for automatic speech recognition (ASR)



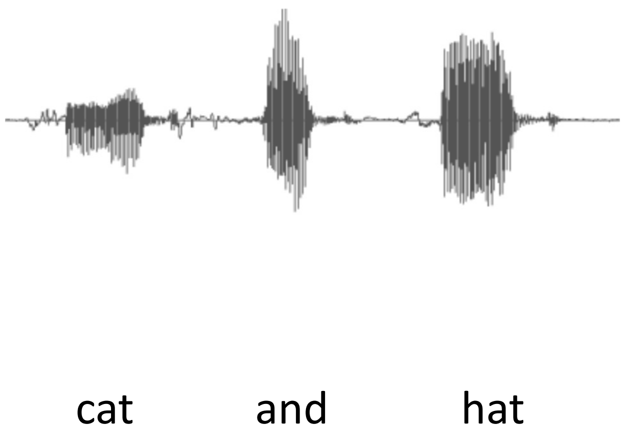
Hidden Markov models are popular as acoustic models

$$
\begin{aligned}
P(\mathbf{A} \mid \mathbf{W}) &= \sum_{\mathbf{S} \in \mathcal{S}(\mathbf{W})} P(\mathbf{A}, \mathbf{S} \mid \mathbf{W}) = \sum_{\mathbf{S} \in \mathcal{S}(\mathbf{W})} P(\mathbf{A} \mid \mathbf{S}, \mathbf{W}) P(\mathbf{S} \mid \mathbf{W}) \\
&\approx \sum_{\mathbf{S} \in \mathcal{S}(\mathbf{W})} P_E(\mathbf{A} \mid \mathbf{S}) P_T(\mathbf{S}) \\
&= \sum_{\mathbf{S} \in \mathcal{S}(\mathbf{W})} P_E(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_T \mid s_1, s_2, \ldots, s_T) P_T(s_1, s_2, \ldots, s_T) \\
&= \sum_{\mathbf{S} \in \mathcal{S}(\mathbf{W})} \prod_{t=1}^{T} P_E(\mathbf{a}_t \mid s_t) P_T(s_t | s_{t-1})
\end{aligned}
$$

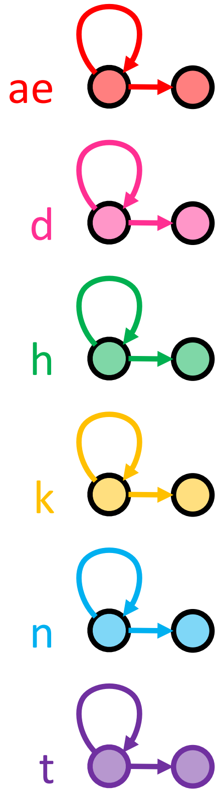
Dynamic programming is popular for "decoding," i.e. for hypothesis search

$$\begin{aligned}
\widehat{\mathbf{W}} &= \arg\max_{\mathbf{W}} P(\mathbf{A} \mid \mathbf{W}) P(\mathbf{W}) \\
&= \arg\max_{\mathbf{W}} \sum_{\mathbf{S} \in \mathcal{S}(\mathbf{W})} P(\mathbf{A} \mid \mathbf{S}) P(\mathbf{S}) P(\mathbf{W}) \\
&\approx \arg\max_{\mathbf{W}} \max_{\mathbf{S} \in \mathcal{S}(\mathbf{W})} P(\mathbf{A} \mid \mathbf{S}) P(\mathbf{S}) P(\mathbf{W}) \\
&= \arg\max_{\mathbf{W}} \max_{\mathbf{S} \in \mathcal{S}(\mathbf{W})} \log P(\mathbf{A} \mid \mathbf{S}) + \log P(\mathbf{S}) + \log P(\mathbf{W}) \\
&\equiv \texttt{Project}\left(\texttt{Bestpath}\left(\texttt{Compose}\left(\mathbf{A}_{\log P(\mathbf{A} \mid \mathbf{S})} \circ \mathbf{L}_{\log P(\mathbf{S})} \circ \mathbf{G}_{\log P(\mathbf{W})}\right)\right)\right)
\end{aligned}$$

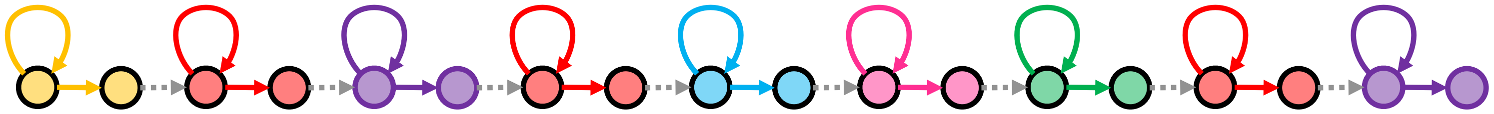
Composite HMM for “cat and hat”

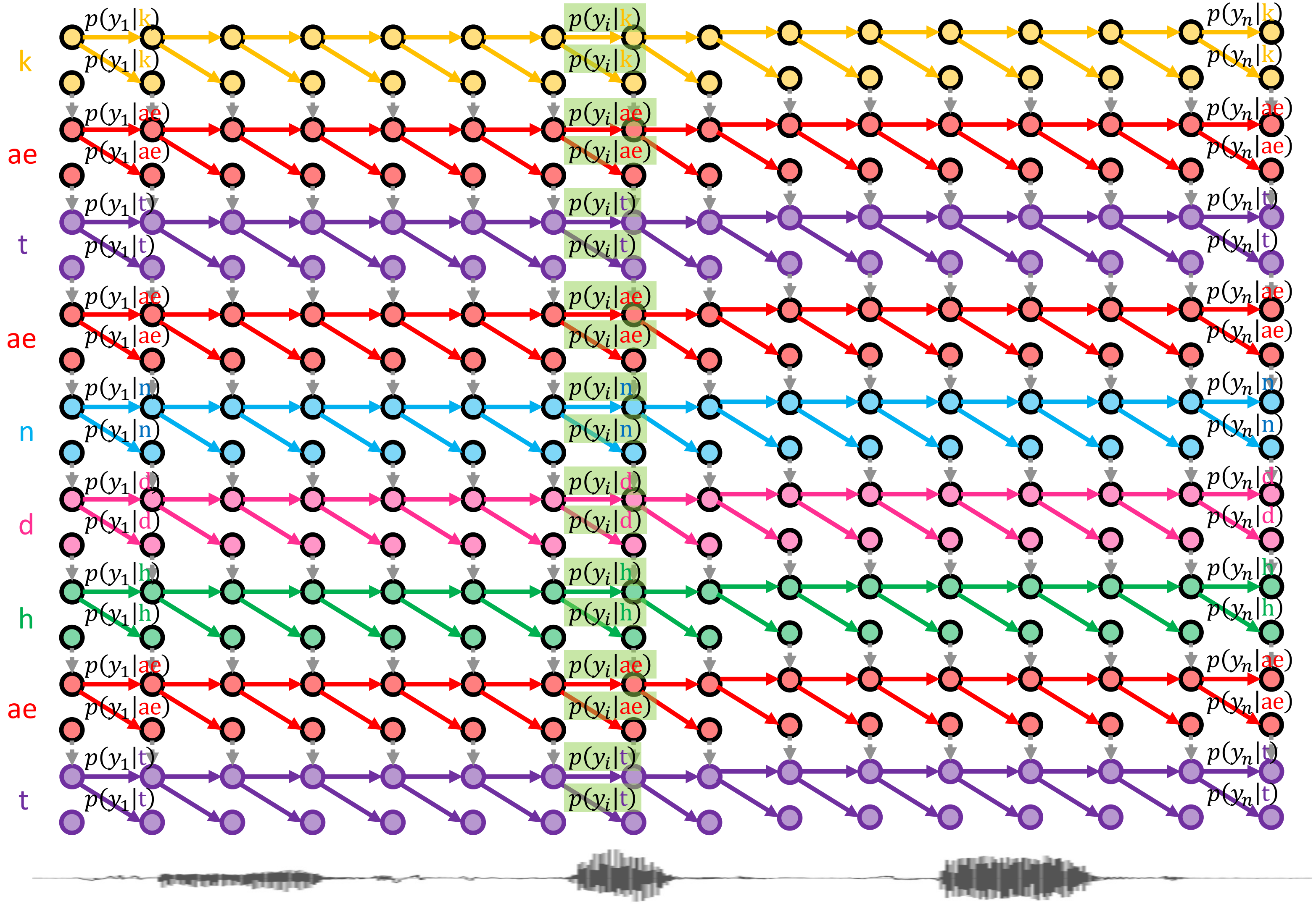


Phoneme HMMs

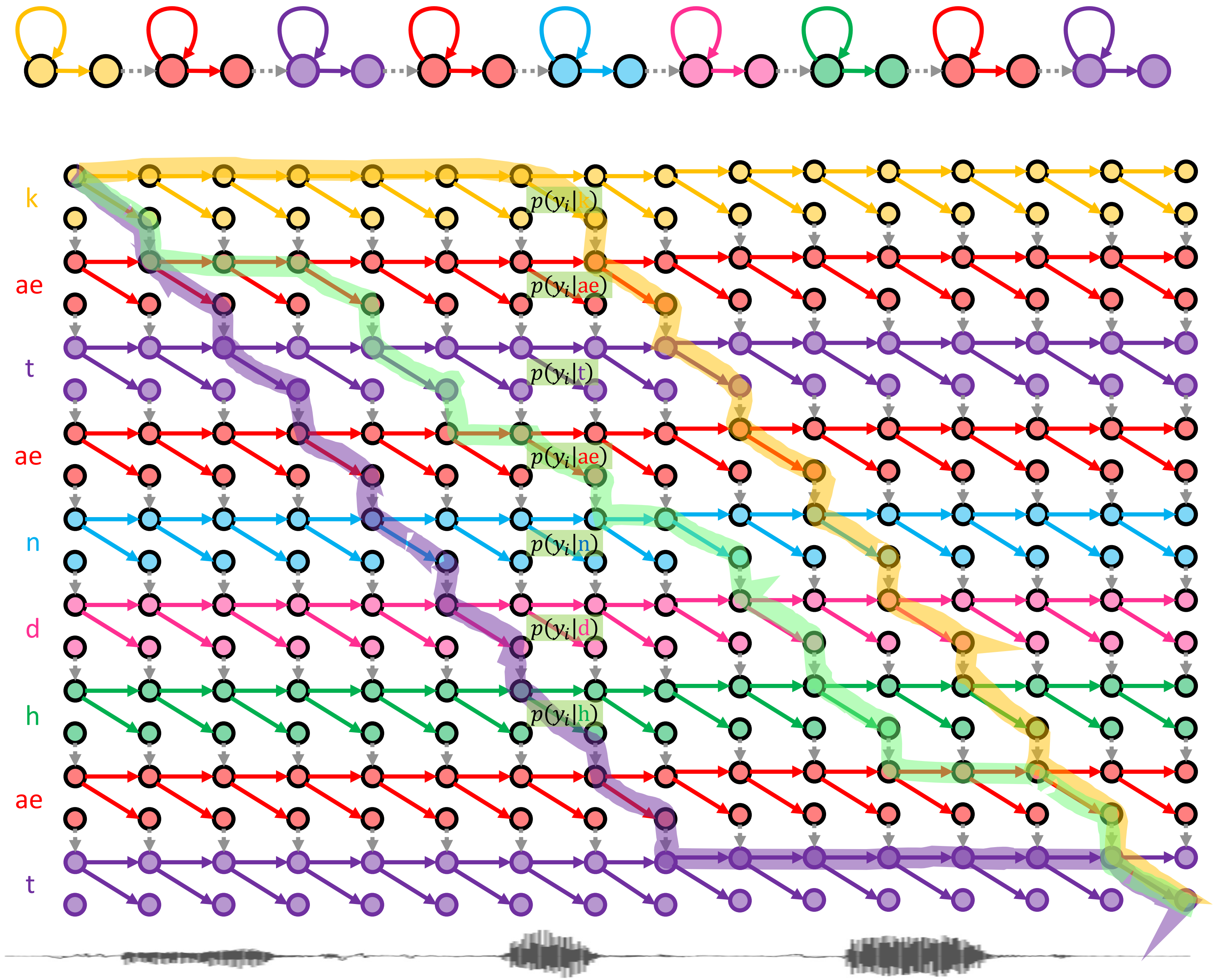


and	ae	n	d
cat	k	ae	t
hat	h	ae	t





Composite HMM for “cat and hat”



“Forward” Algorithm

$$P(\boldsymbol{y}|\boldsymbol{w}) = \sum_{\boldsymbol{s}\in\mathcal{S}(\boldsymbol{w})} P_\vartheta(\boldsymbol{y}|\boldsymbol{s})P_\tau(\boldsymbol{s})$$

$$= \sum_{\boldsymbol{s}\in\mathcal{S}(\boldsymbol{w})} \prod_{i=1}^{n} P_\vartheta(y_i|s_i)P_\tau(s_i|s_{i-1})$$

Viterbi Algorithm

$$\hat{\boldsymbol{s}} = \arg\max_{\boldsymbol{s}\in\mathcal{S}(\boldsymbol{w})} P(\boldsymbol{s}|\boldsymbol{y})$$

$$= \arg\max_{\boldsymbol{s}\in\mathcal{S}(\boldsymbol{w})} \frac{P(\boldsymbol{y},\boldsymbol{s})}{P(\boldsymbol{y})}$$

$$= \arg\max_{\boldsymbol{s}\in\mathcal{S}(\boldsymbol{w})} \prod_{i=1}^{n} P_\vartheta(y_i|s_i)P_\tau(s_i|s_{i-1})$$

Acoustic Modeling with Deep Neural Networks for Hybrid ASR Systems

Repurposing Algorithms Developed for HMM-based Architectures

A paper appeared in September 2011 …

INTERSPEECH 2011

Conversational Speech Transcription
Using Context-Dependent Deep Neural Networks

Frank Seide[1], *Gang Li,*[1] *and Dong Yu*[2]

[1]Microsoft Research Asia, Beijing, P.R.C.
[2]Microsoft Research, Redmond, USA
{fseide,g

ICASSP 1988

Phoneme Recognition: Neural Networks vs. Hidden Markov Models

A. Waibel T. Hanazawa G. Hinton * K. Shikano K. Lang †

ATR Interpreting Telephony Research Laboratories
*University of Toronto and Canadian Institute for Advanced Research
†Carnegie-Mellon University

So, ~~a lot of~~ progress has been made since 1988

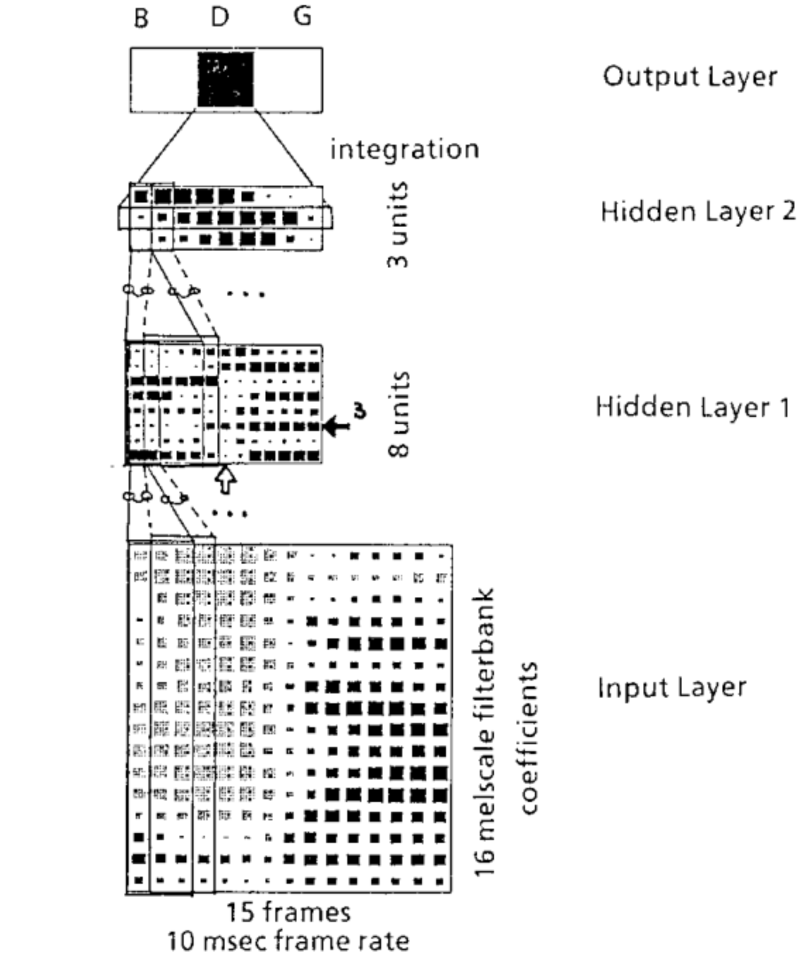
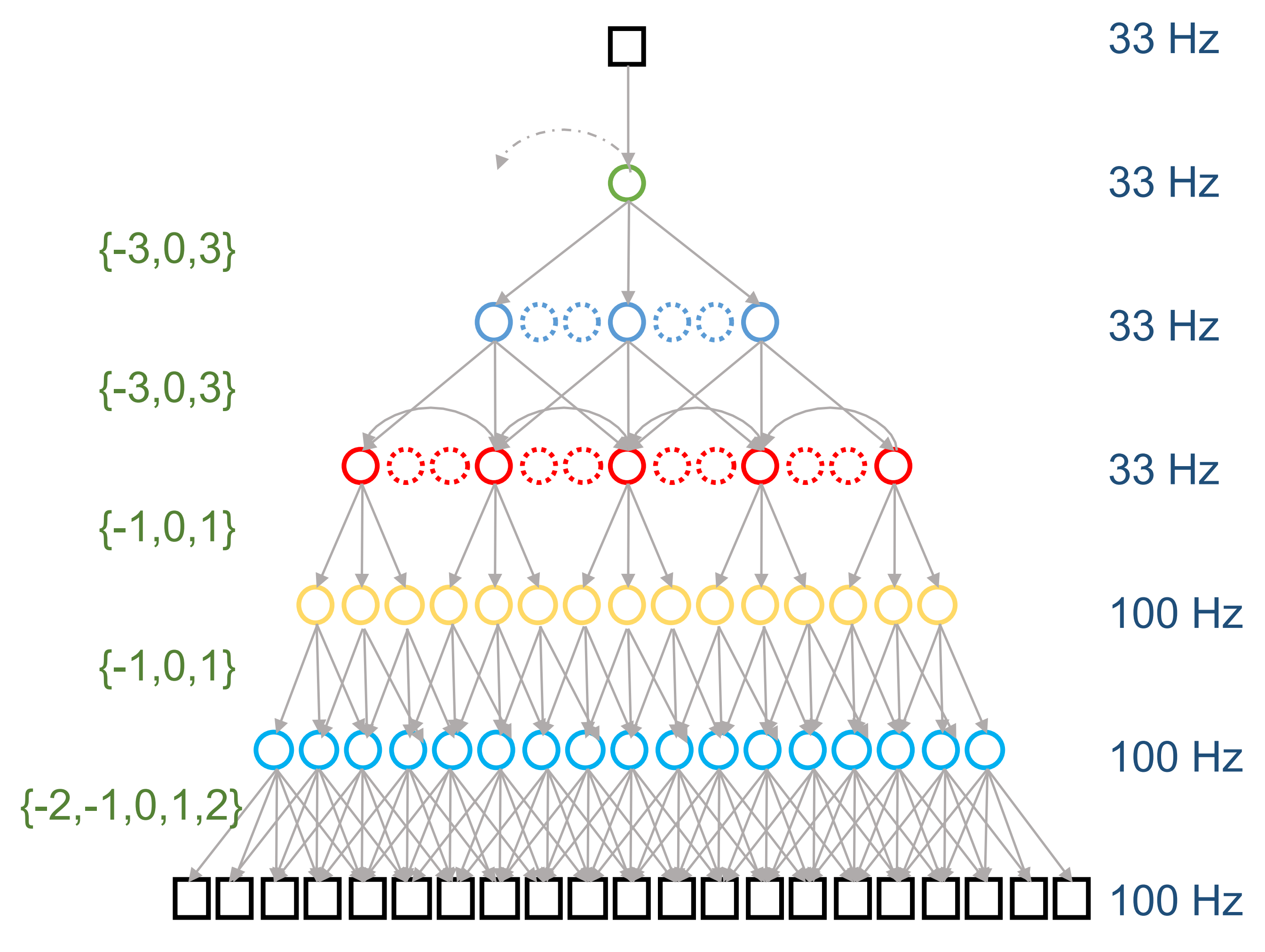
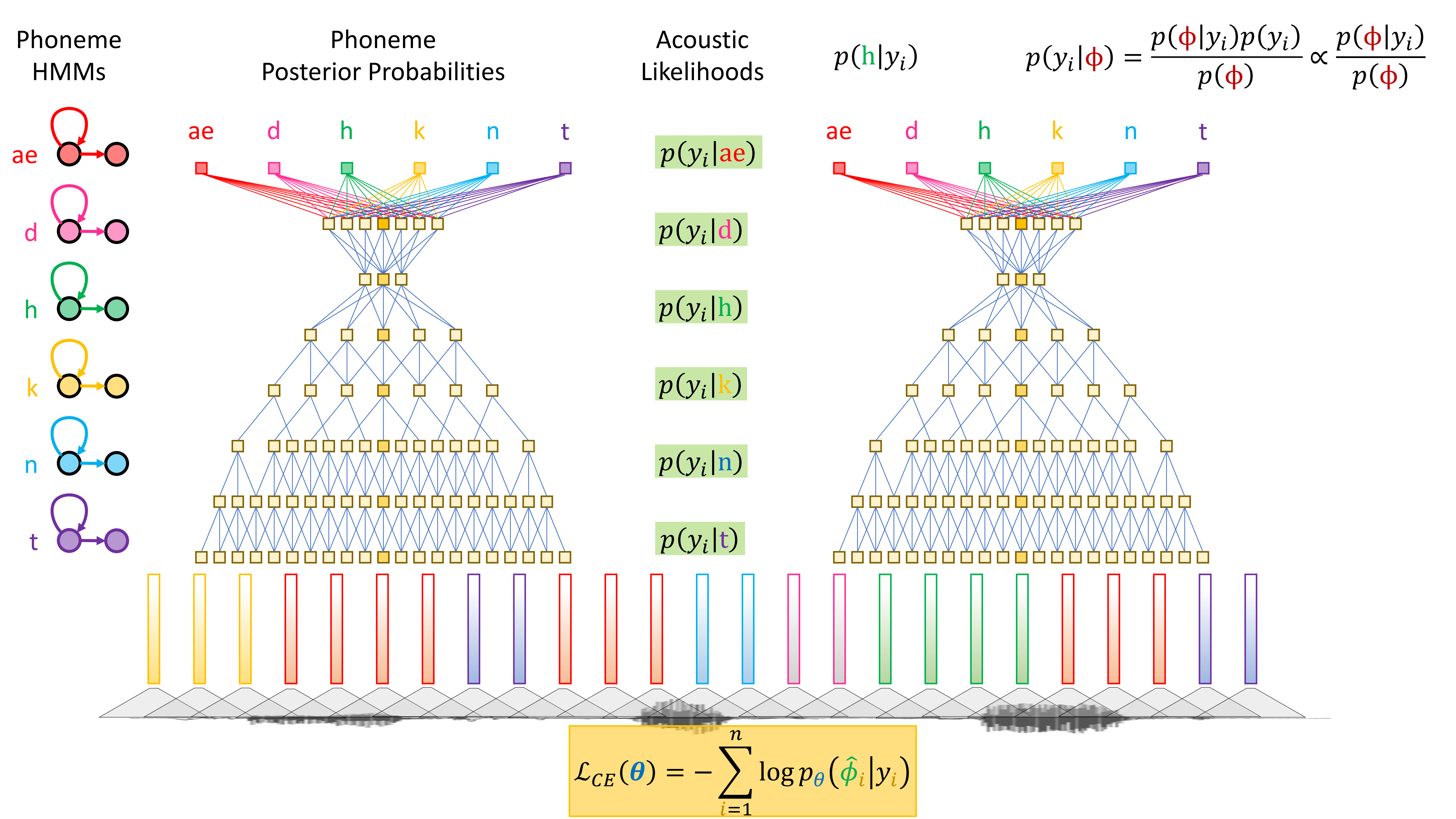
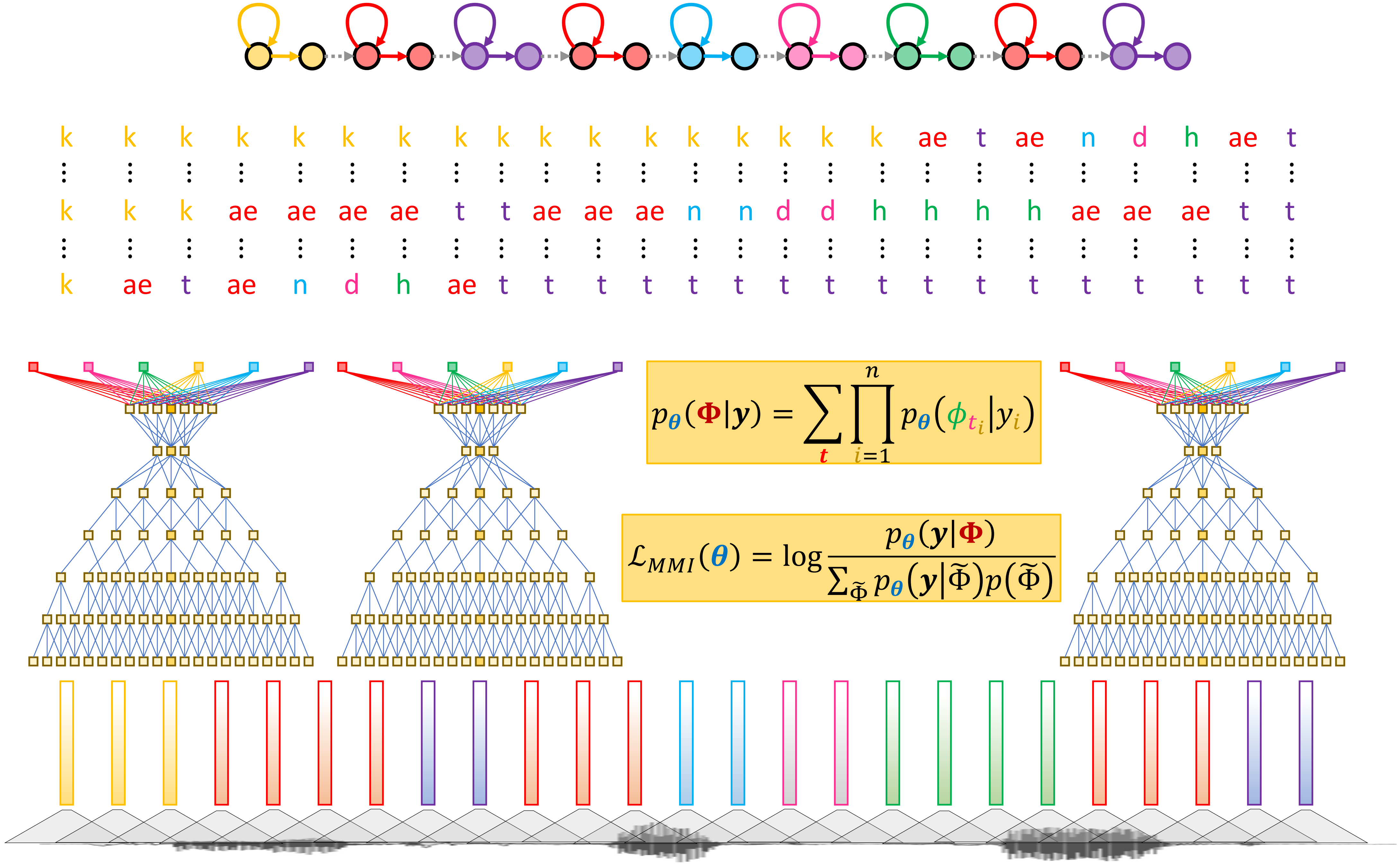


Figure 2: The TDNN architecture (input: "DA")




Phoneme HMMs
Phoneme Posterior Probabilities
Acoustic Likelihoods
$p(h|y_i)$
$p(y_i|\phi) = \frac{p(\phi|y_i)p(y_i)}{p(\phi)} \propto \frac{p(\phi|y_i)}{p(\phi)}$
ae
d
h
k
n
t
ae d h k n t
$p(y_i|ae)$
$p(y_i|d)$
$p(y_i|h)$
$p(y_i|k)$
$p(y_i|n)$
$p(y_i|t)$
ae d h k n t
$\mathcal{L}_{CE}(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \log p_\theta(\hat{\phi}_i|y_i)$



k k k k k k k k k k k k k k k k k ae t ae n d h ae t
k k k ae ae ae ae t t ae ae ae n n d d h h h h ae ae ae t t
k ae t ae n d h ae t t t t t t t t t t t t t t t t t
$p_{\boldsymbol{\theta}}(\boldsymbol{\Phi}|\boldsymbol{y}) = \sum_{t} \prod_{i=1}^{n} p_{\boldsymbol{\theta}}(\phi_{t_i}|y_i)$
$\mathcal{L}_{MMI}(\boldsymbol{\theta}) = \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{\Phi})}{\sum_{\tilde{\Phi}} p_{\boldsymbol{\theta}}(\boldsymbol{y}|\tilde{\Phi})p(\tilde{\Phi})}$

Maximum Mutual Information Estimation

INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Purely sequence-trained neural networks for ASR based on lattice-free MMI

Daniel Povey[1,2], *Vijayaditya Peddinti*[1], *Daniel Galvez*[3], *Pegah Ghahremani*[1],
Vimal Manohar[1], *Xingyu Na*[4], *Yiming Wang*[1], *Sanjeev Khudanpur*[1,2]

[1]Center for Language and Speech Processing, The Johns Hopkins University
[2] HLT CoE, The Johns Hopkins University
[3]Department of Computer Science, Cornell University
[4]Lele Innovation and Intelligence Technology (Beijing) Co., Ltd.

{dpovey,dt.galvez,asr.naxingyu}@gmail.com,
{vijay.p,vmanoha1,pghahre1,yiming.wang,khudanpur}@jhu.edu

Language Modeling with (Recurrent) Neural Networks

Efforts to Go Beyond *n*-gram Dependence in Language Models

Using Neural Networks to Estimate $P(w_t|h_t)$

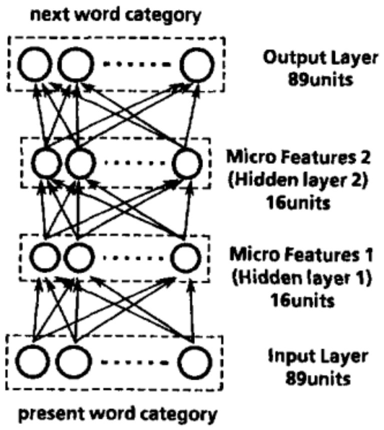


Fig.2 Basic Bigram Network for Word Category Prediction

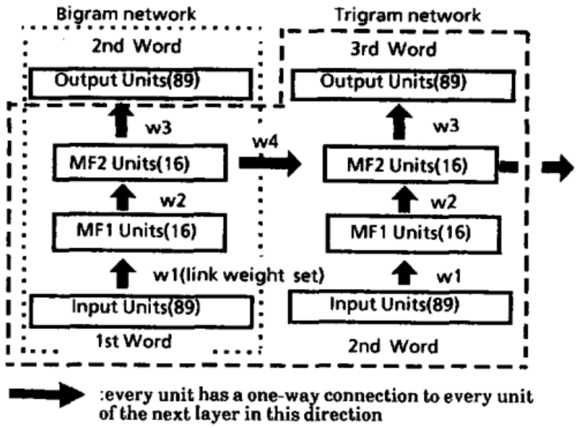


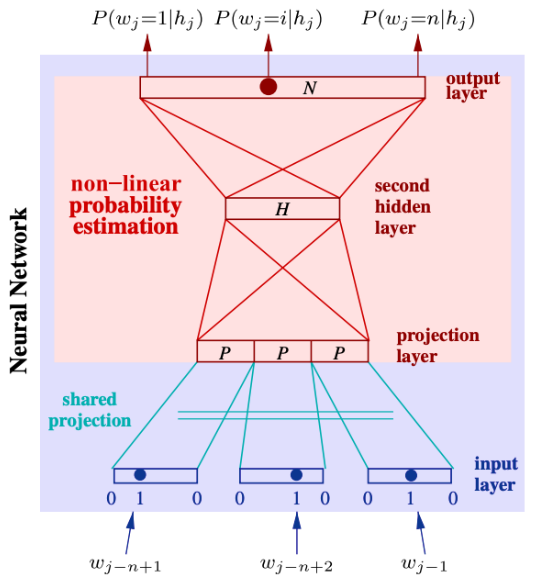
Fig.3 NETgram Model 1 for Word Category Prediction

A STUDY OF ENGLISH WORD CATEGORY PREDICTION BASED ON NEURAL NETWORKS

Masami NAKAMURA, Kiyohiro SHIKANO

ATR Interpreting Telephony Research Laboratories
Seika-chou, Souraku-gun, Kyoto 619-02, JAPAN

ICASSP 1989



ELSEVIER

Available online at www.sciencedirect

ScienceDirect

LANGUAGE

Computer Speech and Language 21 (2007) 492–518

www.elsevier.com/locate/csl

Continuous space language models ☆

Holger Schwenk

Spoken Language Processing Group, LIMSI-CNRS, BP 133, 91403 Orsay cedex, France

Received 19 December 2005; received in revised form 15 September 2006; accepted 15 September 2006
Available online 9 October 2006

A paper appeared in September 2010 …

COGNITIVE SCIENCE 14, 179–211 (1990)

Finding Structure in Time

JEFFREY L. ELMAN
University of California, San Diego

INTERSPEECH 2010

Recurrent neural network based language model

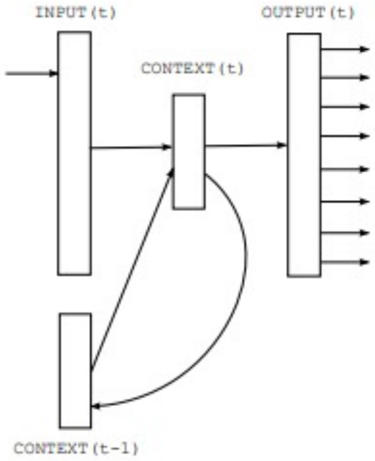
Tomáš Mikolov[1,2], *Martin Karafiát*[1], *Lukáš Burget*[1], *Jan "Honza" Černocký*[1], *Sanjeev Khudanpur*[2]

[1]Speech@FIT, Brno University of Technology, Czech Republic
[2] Department of Electrical and Computer Engineering, Johns Hopkins University, USA
{imikolov,karafiat,burget,cernocky}@fit.vutbr.cz, khudanpur@jhu.edu

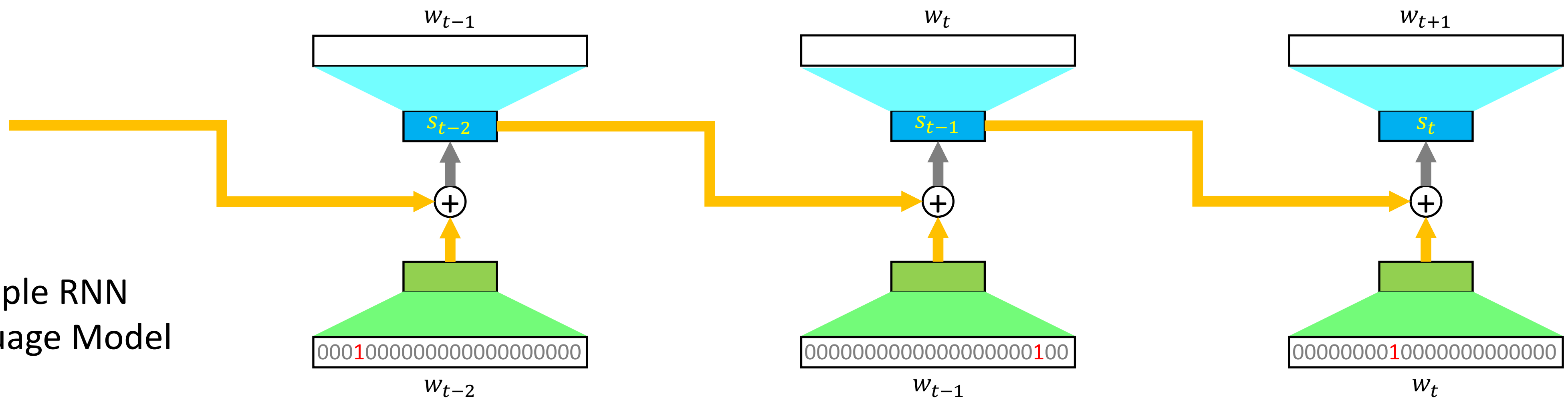
Abstract

A new recurrent neural network based language model (RNN LM) with applications to speech recognition is presented. Results indicate that it is possible to obtain around 50% reduction of perplexity by using mixture of several RNN LMs, compared to a state of the art backoff language model. Speech recognition experiments show around 18% reduction of word error rate on the Wall Street Journal task when comparing models trained on the same amount of data, and around 5% on the much harder NIST RT05 task, even when the backoff model is trained on much more data than the RNN LM. We provide ample empirical evidence to suggest that connectionist language models are superior to standard n-gram techniques, except their high computational (training) complexity.

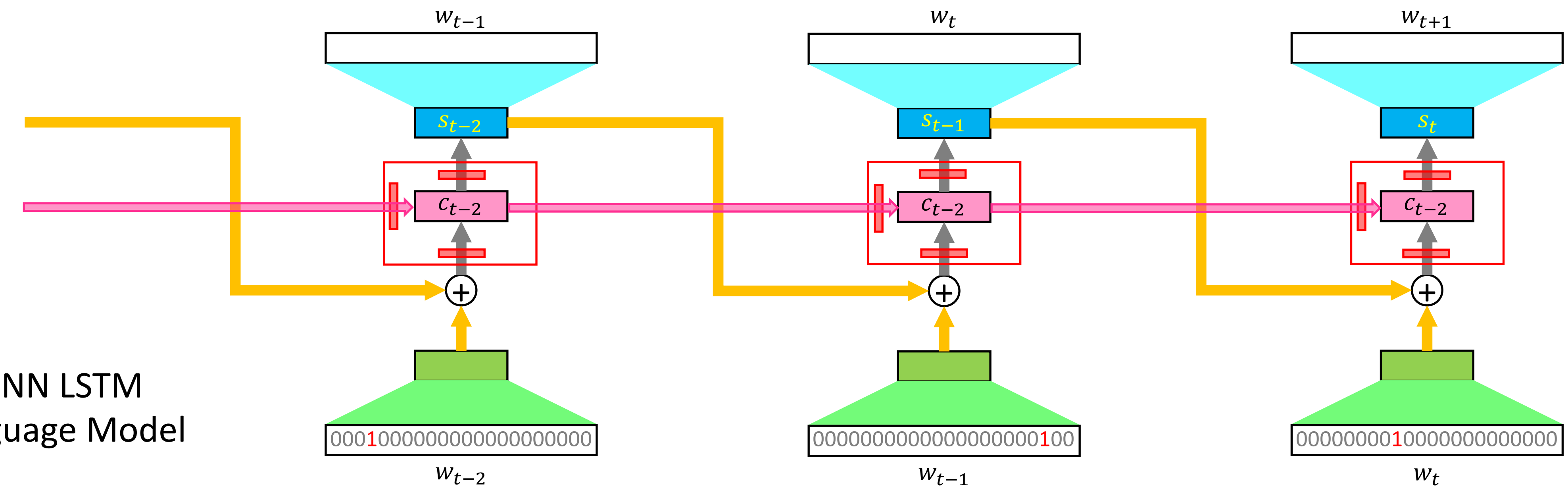
Index Terms: language modeling, recurrent neural networks, speech recognition



A Simple RNN
Language Model

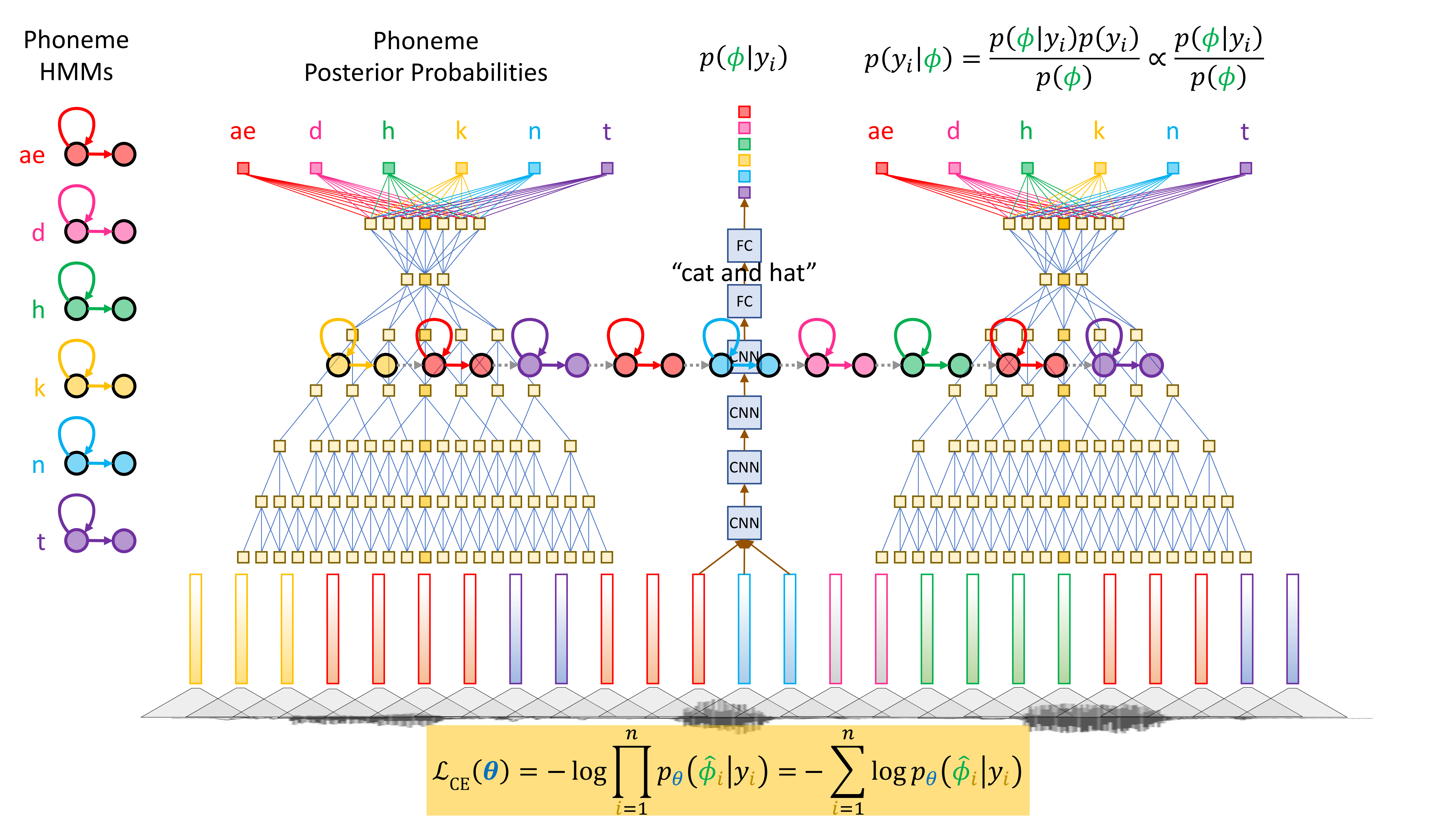


An RNN LSTM
Language Model

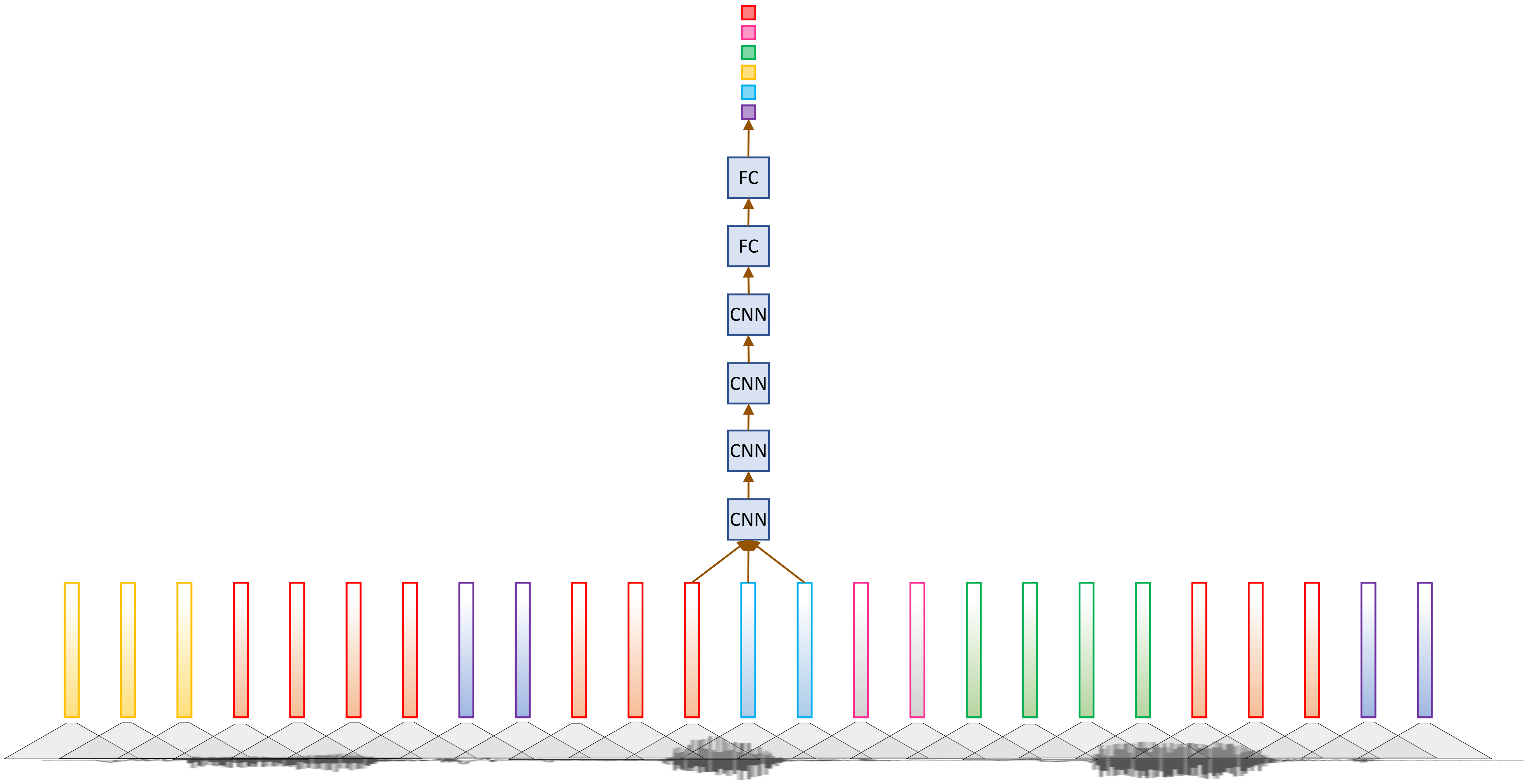


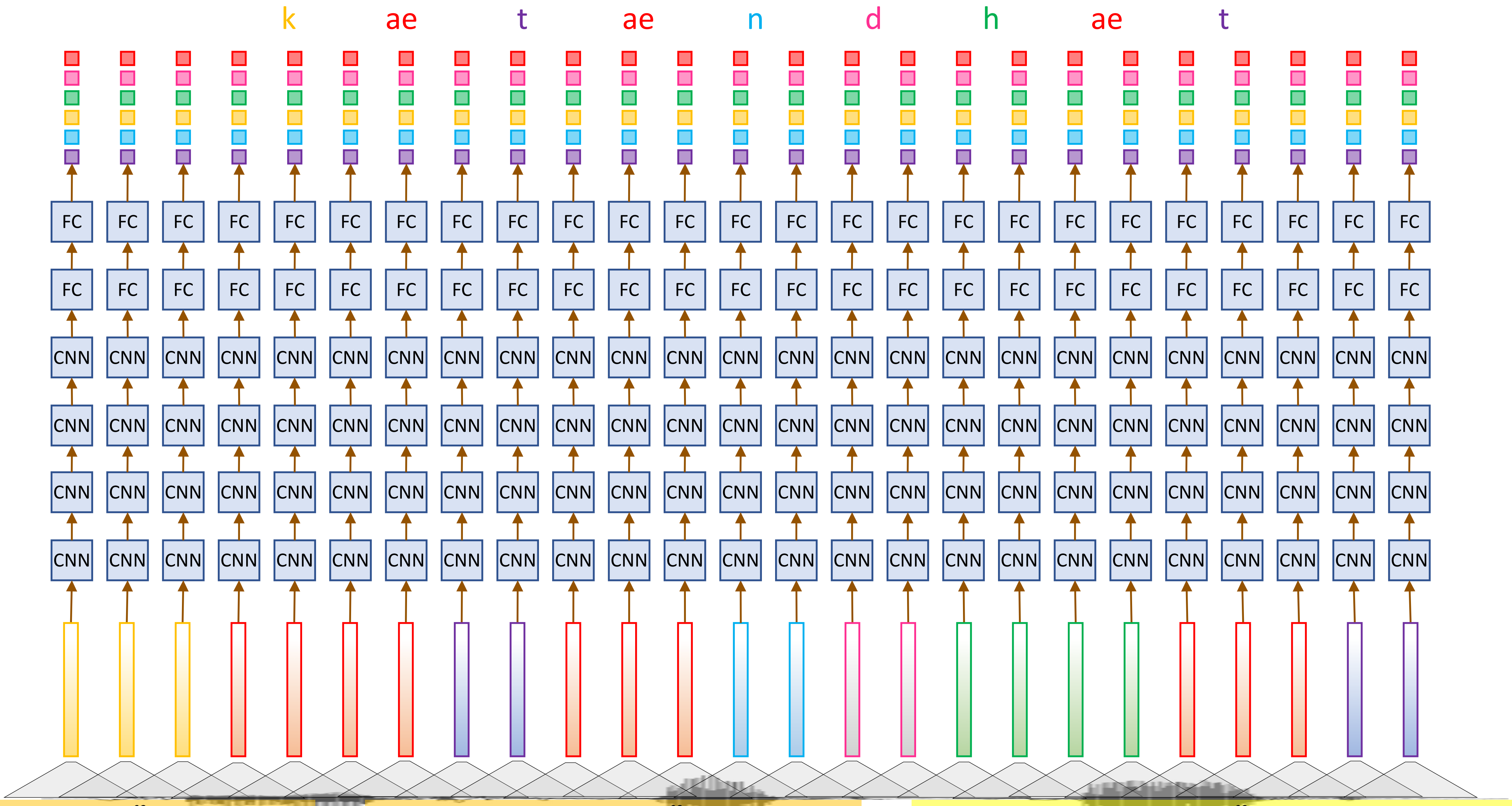
Training Neural Networks without using HMM-Based Alignments

Connectionist Temporal Classification



Phoneme HMMs
Phoneme Posterior Probabilities
$p(\phi|y_i)$
$p(y_i|\phi) = \frac{p(\phi|y_i)p(y_i)}{p(\phi)} \propto \frac{p(\phi|y_i)}{p(\phi)}$
ae
d
h
k
n
t
ae d h k n t
ae d h k n t
FC
FC
CNN
CNN
CNN
CNN
“cat and hat”
$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{\theta}) = -\log \prod_{i=1}^{n} p_\theta(\hat{\phi}_i|y_i) = -\sum_{i=1}^{n} \log p_\theta(\hat{\phi}_i|y_i)$


FC
FC
CNN
CNN
CNN
CNN



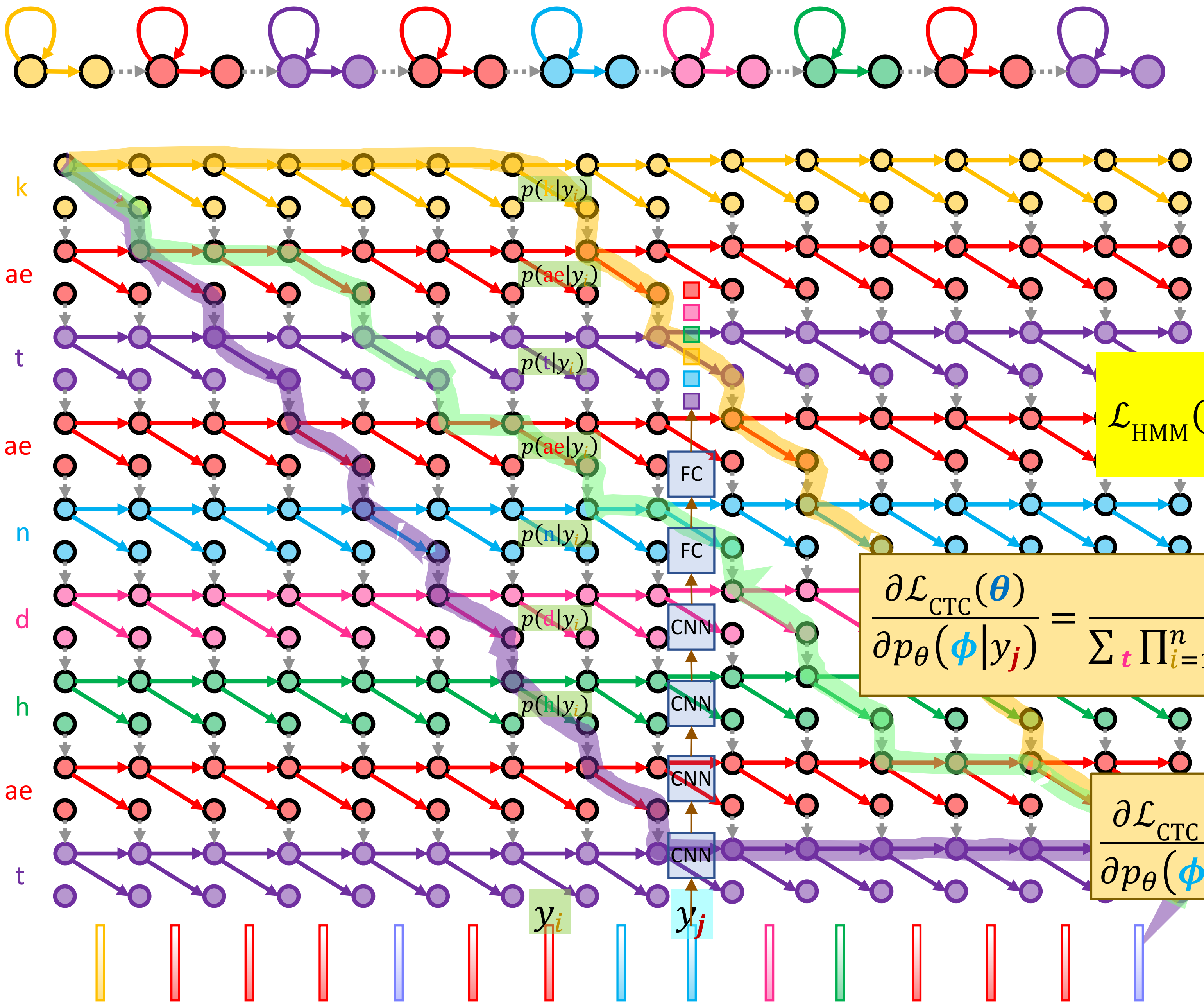
$$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{\theta}) = -\log \prod_{i=1}^{n} p_{\theta}(\hat{\phi}_i | y_i)$$

$$\mathcal{L}_{\mathrm{CTC}}(\boldsymbol{\theta}) = -\log \sum_{t} \prod_{i=1}^{n} p_{\theta}(\phi_{t_i} | y_i)$$

$$\mathcal{L}_{\mathrm{HMM}}(\boldsymbol{\vartheta}) = -\log \sum_{t} \prod_{i=1}^{n} p_{\vartheta}(y_i | \phi_{t_i}) p_{\vartheta}(\phi_{t_i} | \phi_{t_{i-1}})$$

Calculating the CTC loss for "cat and hat"

Calculating the gradient of the CTC loss

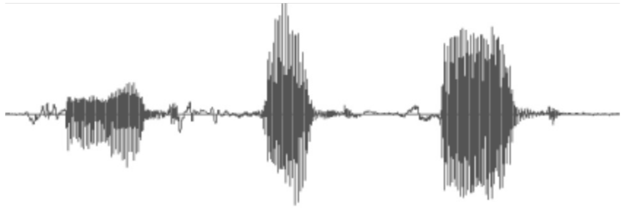


$$\mathcal{L}_{\mathrm{CTC}}(\boldsymbol{\theta}) = -\log \sum_{t} \prod_{i=1}^{n} p_{\theta}(\phi_{t_i}|y_i)$$

$$\mathcal{L}_{\mathrm{HMM}}(\boldsymbol{\vartheta}) = -\log \sum_{t} \prod_{i=1}^{n} p_{\vartheta}(y_i|\phi_{t_i})\, p_{\vartheta}(\phi_{t_i}|\phi_{t_{i-1}})$$

$$\frac{\partial \mathcal{L}_{\mathrm{CTC}}(\boldsymbol{\theta})}{\partial p_{\theta}(\boldsymbol{\phi}|y_j)} = \frac{-1}{\sum_{t} \prod_{i=1}^{n} p_{\theta}(\phi_{t_i}|y_i)} \sum_{t:\phi_{t_j}=\phi} \frac{1}{p_{\theta}(\boldsymbol{\phi}|y_j)} \prod_{i=1}^{n} p_{\theta}(\phi_{t_i}|y_i)$$

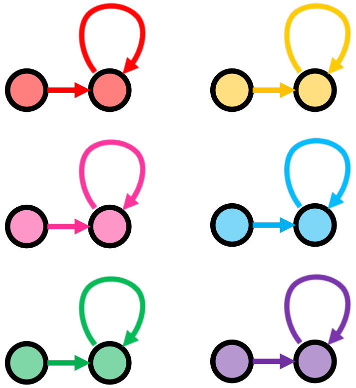
$$\frac{\partial \mathcal{L}_{\mathrm{CTC}}(\boldsymbol{\theta})}{\partial p_{\theta}(\boldsymbol{\phi}|y_j)} = -\frac{1}{p_{\theta}(\boldsymbol{\phi}|y_j)} \frac{\sum_{t:\phi_{t_j}=\phi} \prod_{i=1}^{n} p_{\theta}(\phi_{t_i}|y_i)}{\sum_{t} \prod_{i=1}^{n} p_{\theta}(\phi_{t_i}|y_i)}$$



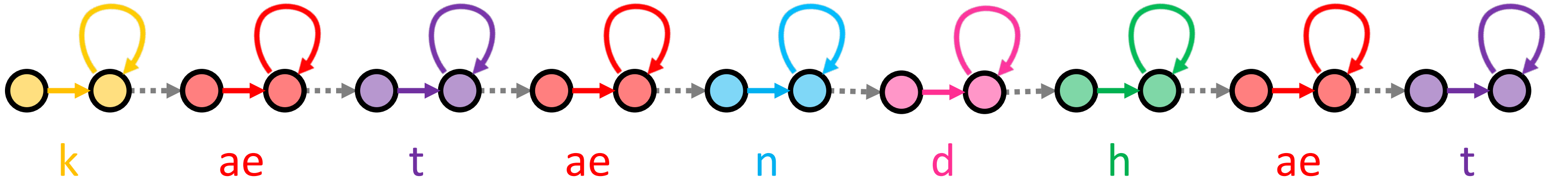
cat and hat

and	ae n d
cat	k ae t
hat	h ae t

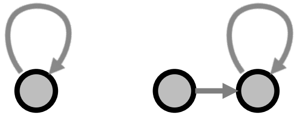
Phoneme HMMs



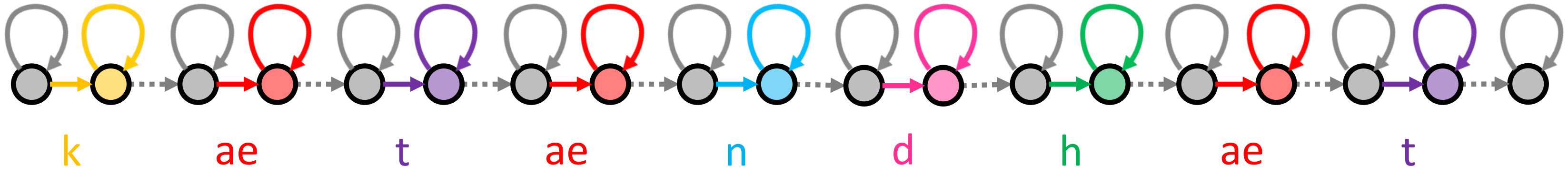
Composite HMM for “cat and hat”

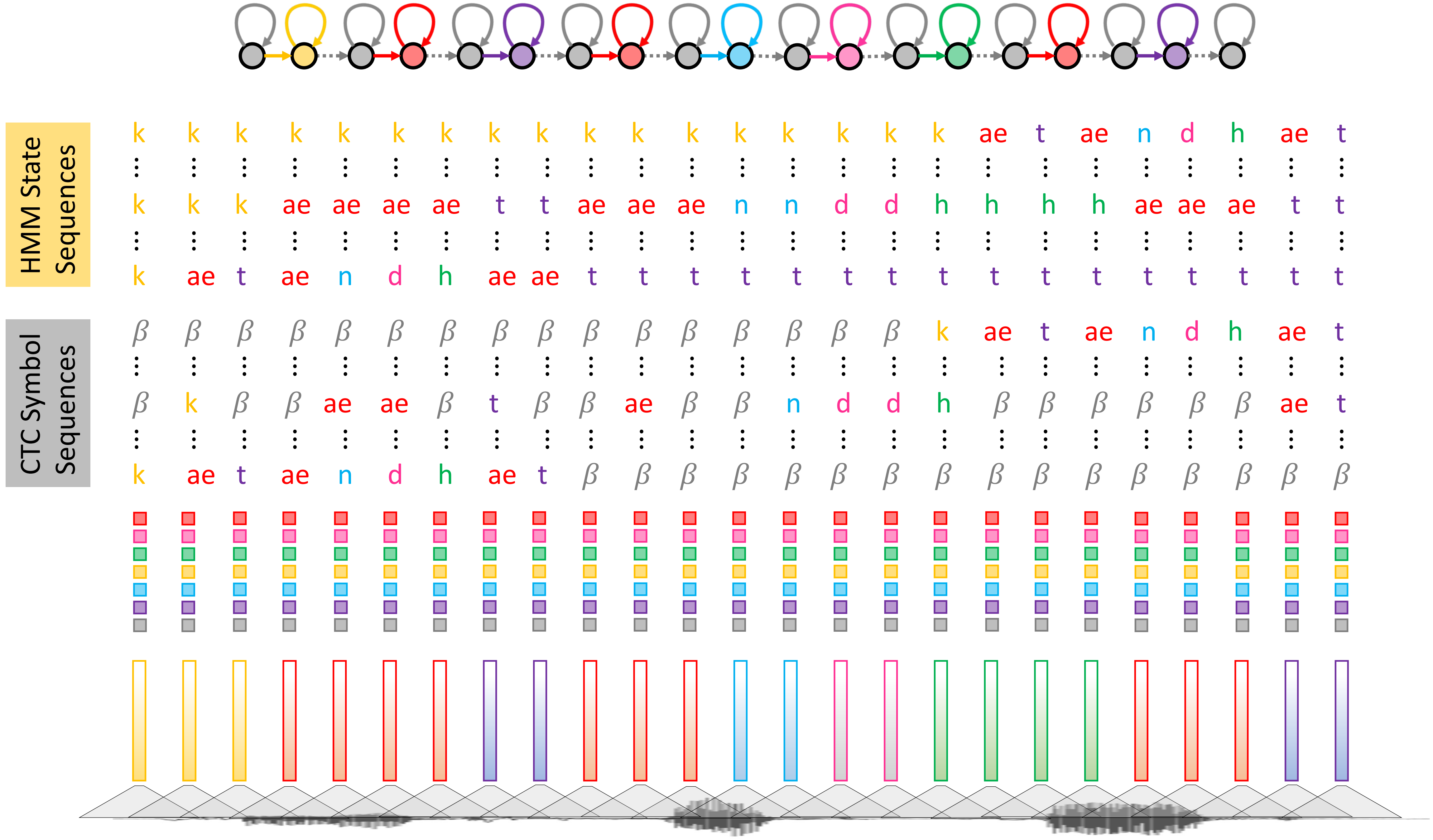


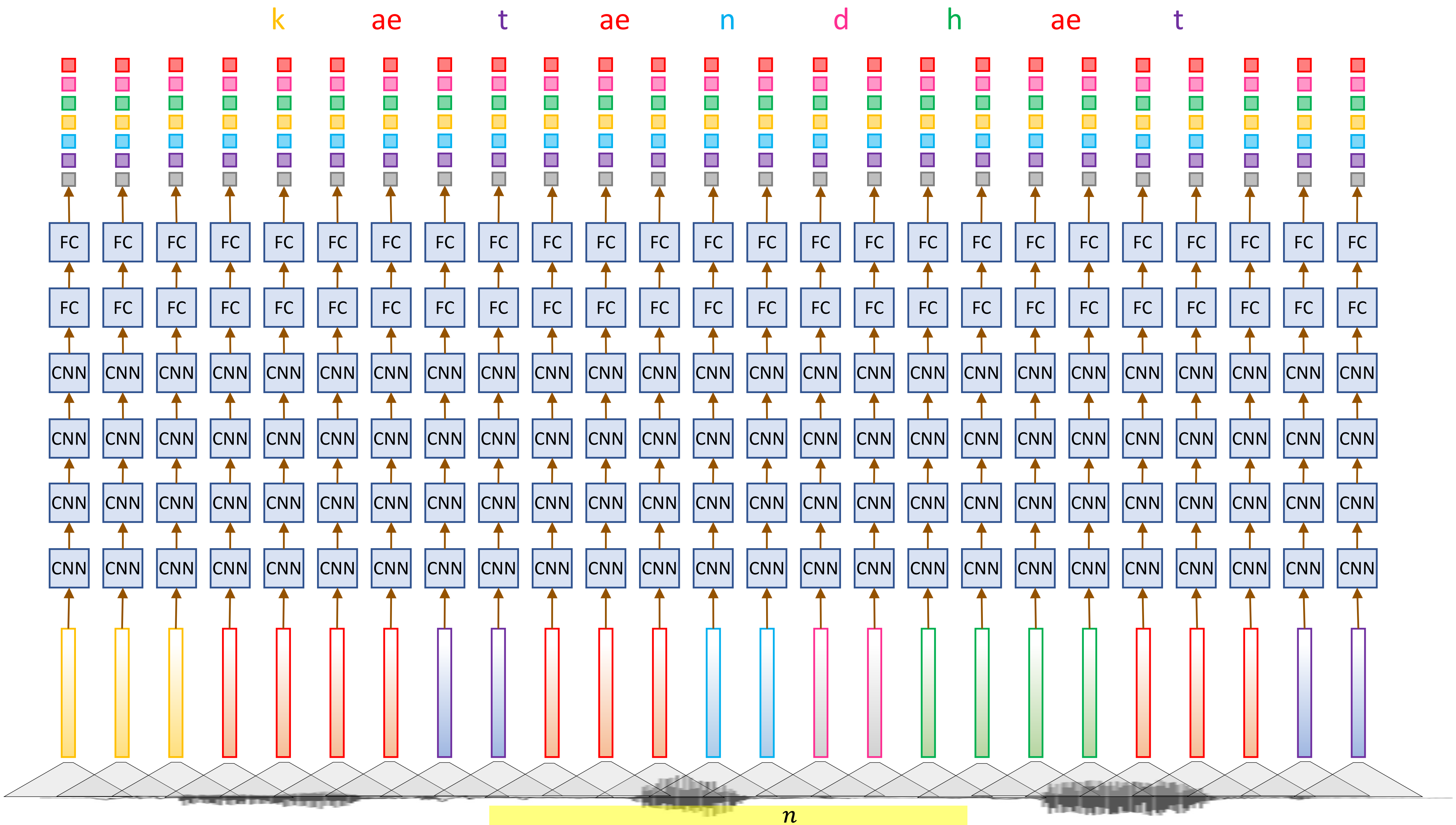
The CTC “`Blank`” Symbol (β)



FSA of permissible CTC strings for “cat and hat”




HMM State Sequences
k k k k k k k k k k k k k k k k k ae t ae n d h ae t
k k k ae ae ae ae t t ae ae ae n n d d h h h h ae ae ae t t
k ae t ae n d h ae ae t t t t t t t t t t t t t t t t
CTC Symbol Sequences
β β β β β β β β β β β β β β β β k ae t ae n d h ae t
β k β β ae ae β t β β ae β β n d d h β β β β β β ae t
k ae t ae n d h ae t β β β β β β β β β β β β β β β β



$$\mathcal{L}_{\mathrm{CTC}}(\boldsymbol{\theta}) = -\log \sum_{t} \prod_{i=1}^{n} p_{\theta}(\phi_{t_i} | y_i)$$

Neural Speech Recognition without HMMs (aka End2End ASR)

“Purely” CTC-Based Speech Recognition Architectures

End-to-End Speech Recognition

arXiv

Deep Speech: Scaling up end-to-end speech recognition

Awni Hannun[*], Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. Ng

Baidu Research – Silicon Valley AI Lab

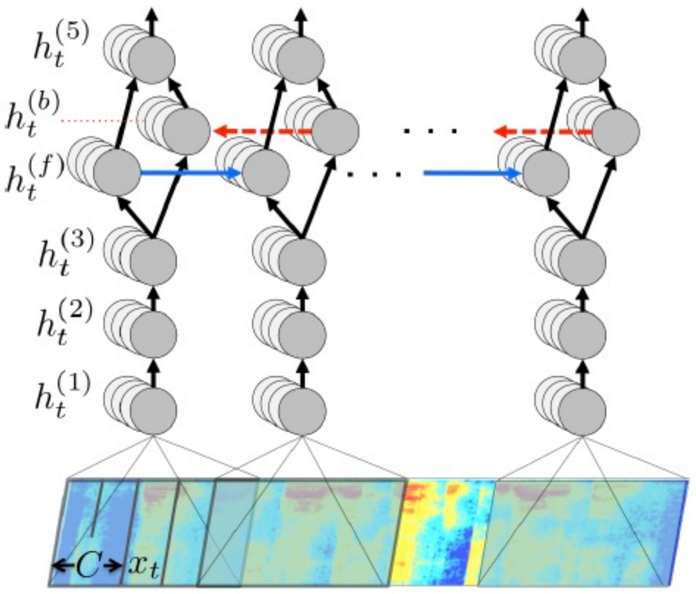
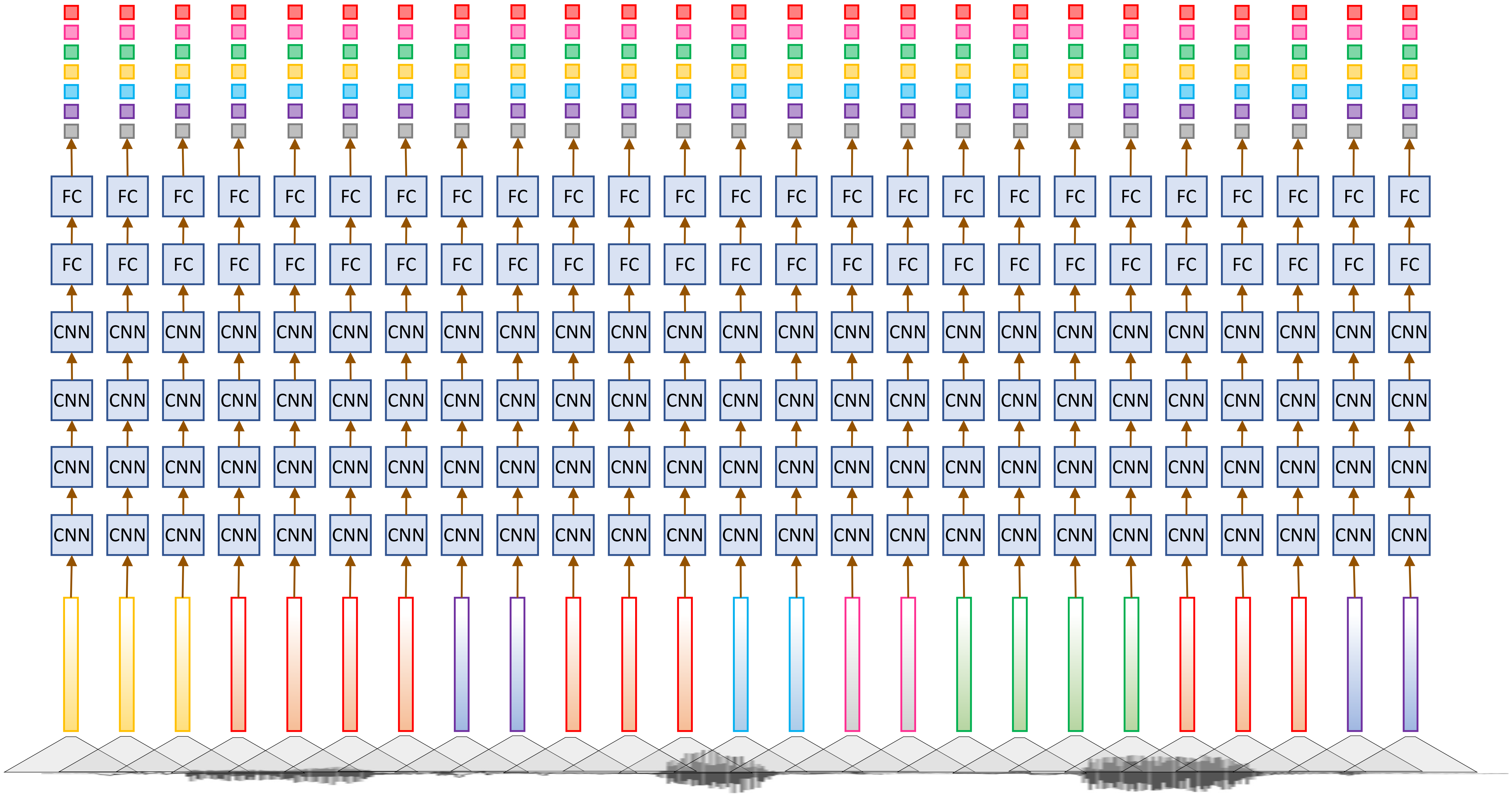
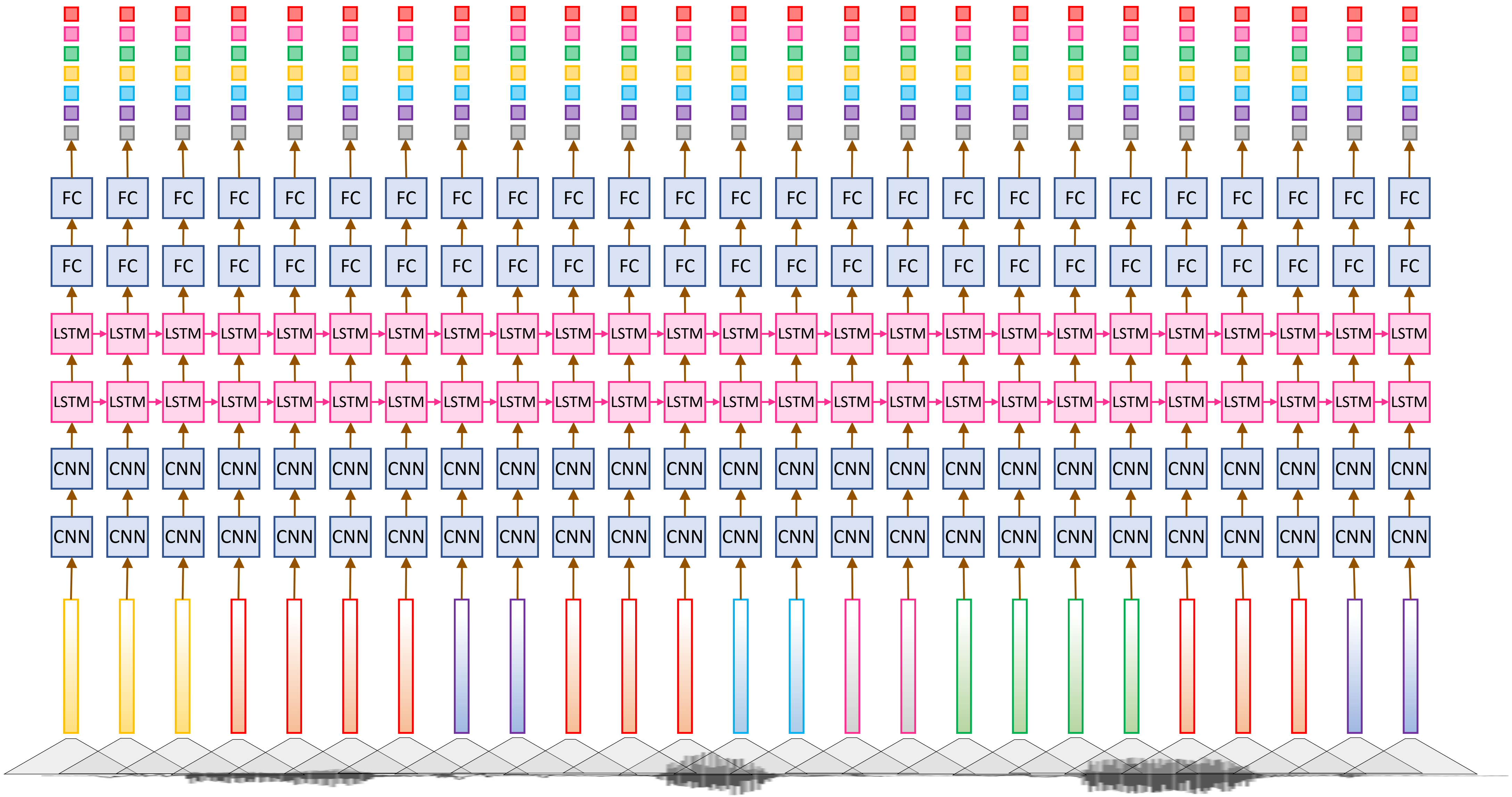


Figure 1: Structure of our RNN model and notation.

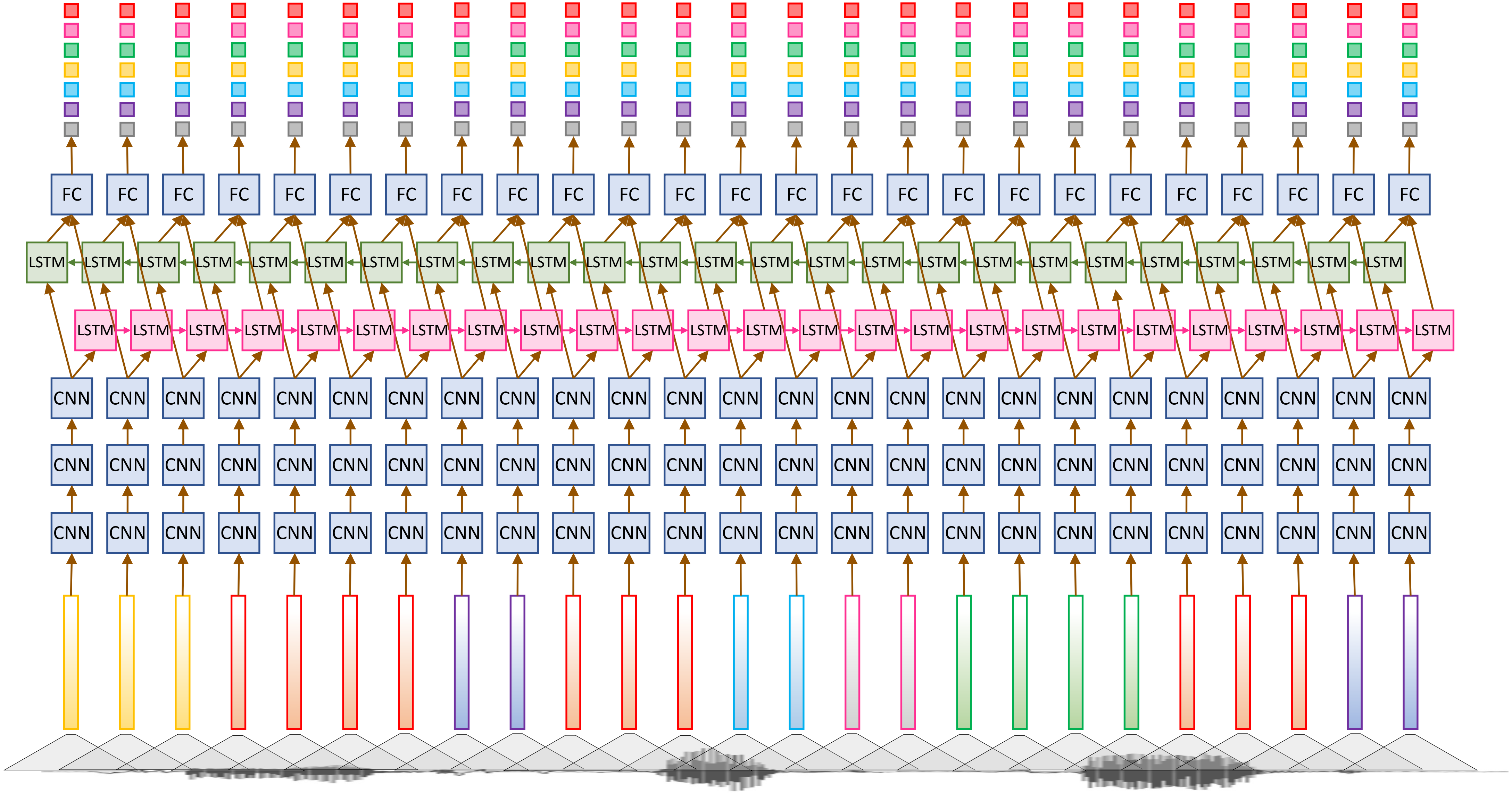
The CNN Architecture



The CNN+LSTM Architecture



A Bidirectional LSTM Architecture (Deep Speech)



End-to-End Speech Recognition using Neural Networks with Attention

Applying ideas from machine translation to speech recognition

Attention-Based Speech Recognition

NeurIPS 2015

Attention-Based Models for Speech Recognition

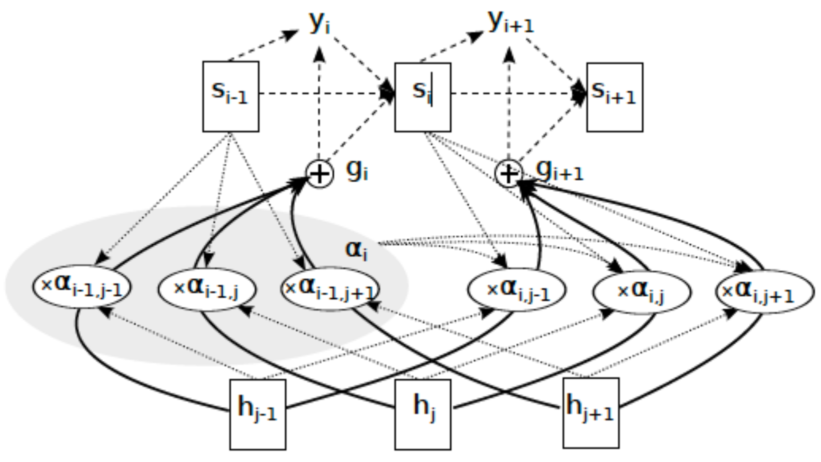
Jan Chorowski
University of Wrocław, Poland
jan.chorowski@ii.uni.wroc.pl

Dzmitry Bahdanau
Jacobs University Bremen, Germany

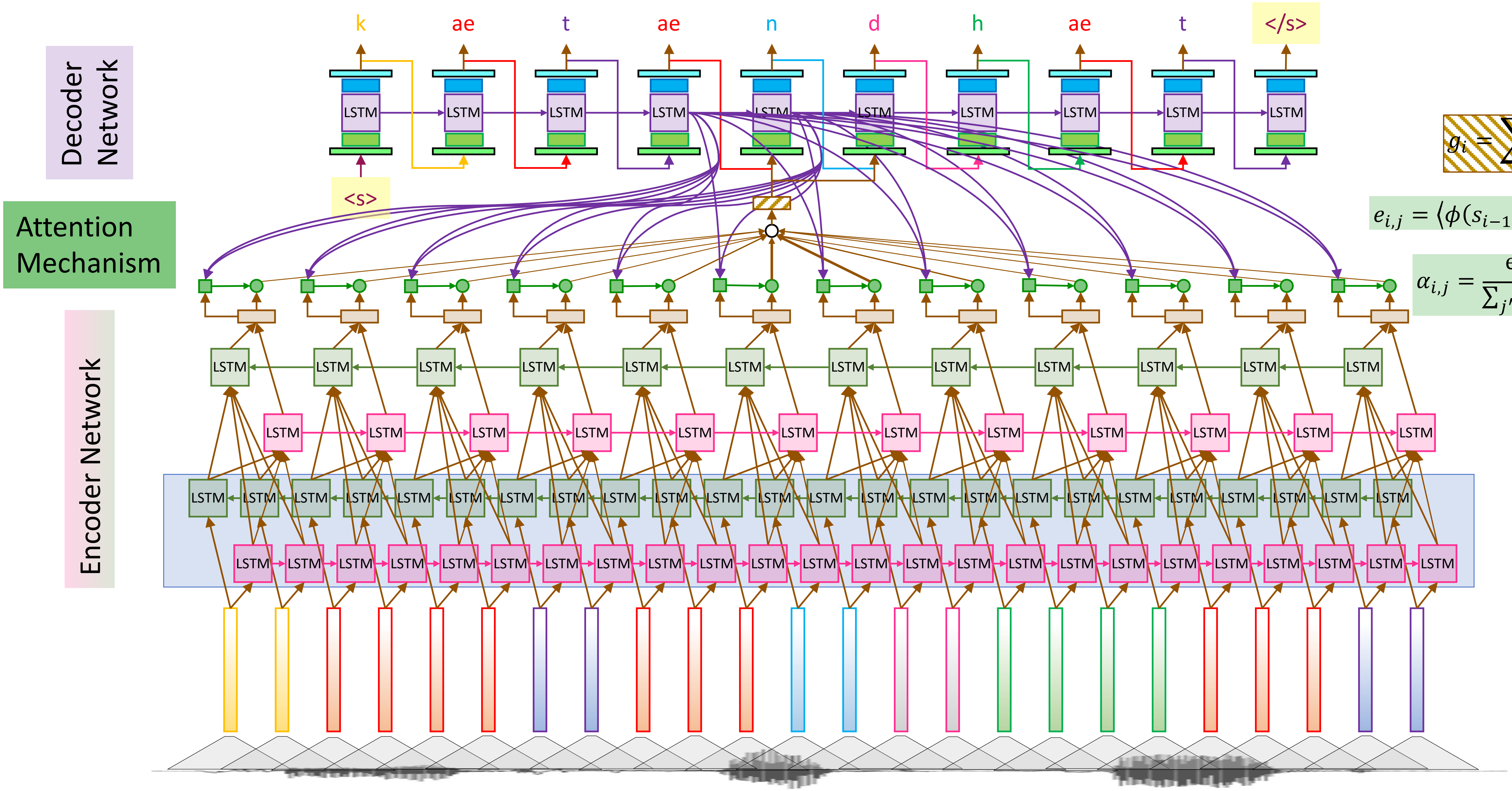
Dmitriy Serdyuk
Université de Montréal

Kyunghyun Cho
Université de Montréal

Yoshua Bengio
Université de Montréal
CIFAR Senior Fellow



An Encoder-Decoder Architecture with Attention



$g_i = \sum_j \alpha_{i,j} h_j$

$e_{i,j} = \langle \phi(s_{i-1}), \psi(h_j) \rangle$

$\alpha_{i,j} = \frac{\exp e_{i,j}}{\sum_{j'} \exp e_{i,j'}}$

Summary + Q&A

- Neural Acoustic Models: Waibel et al (1988) to Povey et al (2016)
- Neural Language Models: Nakamura et al (1989) to Sundermeyer et al (2012)
- Connectionist Temporal Classification: Graves (2006)
- CTC-Based Models: Hannun et al (2014) and Graves & Jaitly (2014)
- Attention-Based Models: Chorowski et al (2015)