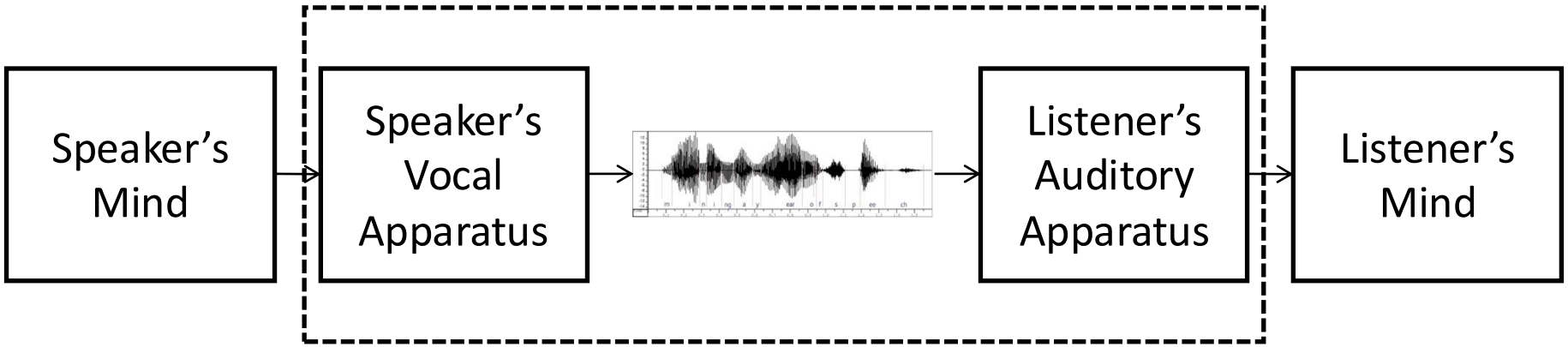
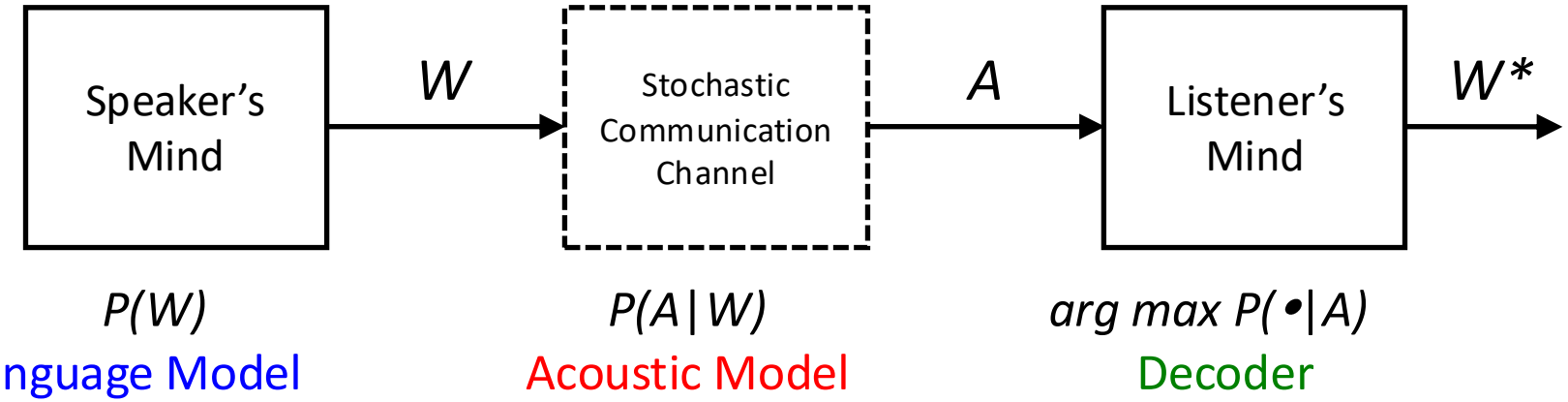
Neural Methods in Automatic Speech Recognition

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) and *Attention-Based Models*: Chorowski et al (2015)
Recurrent Neural Transduction: Graves (2012) to He et al (2019)

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



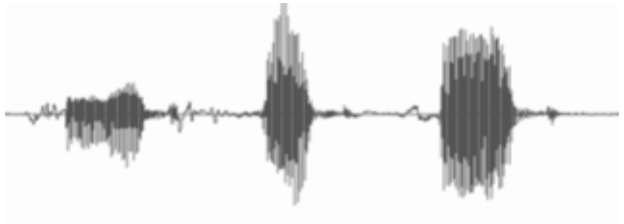


Hidden Markov models became popular as acoustic models

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

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

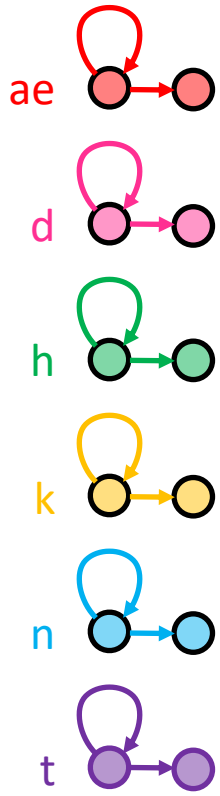
$$
\begin{aligned}
\widehat{\mathbf{W}} &= \arg\max_{\mathbf{W}} P(\mathbf{A} \mid \mathbf{W}) P(\mathbf{W}) \\
&= \arg\max_{\mathbf{W}} \sum_{\mathbf{S} \in \mathcal{S}(\mathbf{W})} P(\mathbf{A} \mid \mathbf{S}) P(\mathbf{S}) P(\mathbf{W}) \\
&\approx \arg\max_{\mathbf{W}} \max_{\mathbf{S} \in \mathcal{S}(\mathbf{W})}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



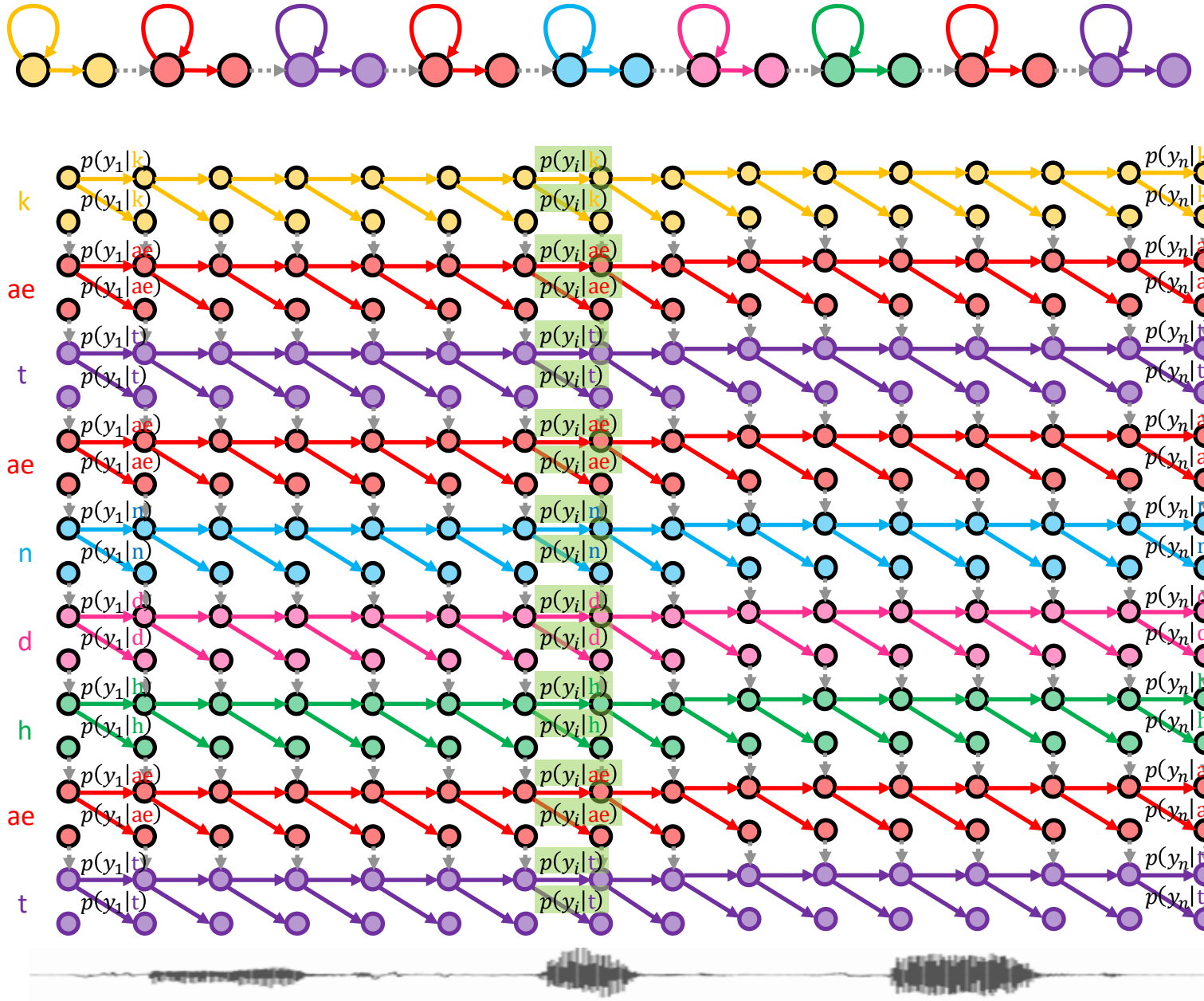
cat and hat

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

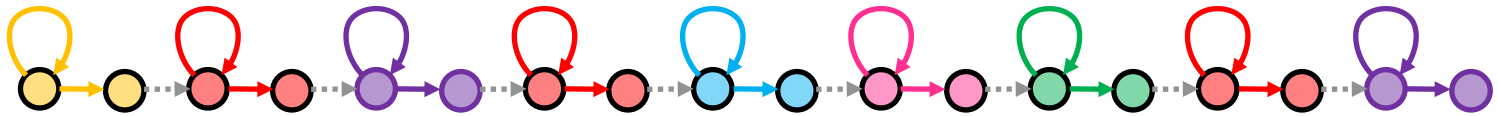
Phoneme HMMs

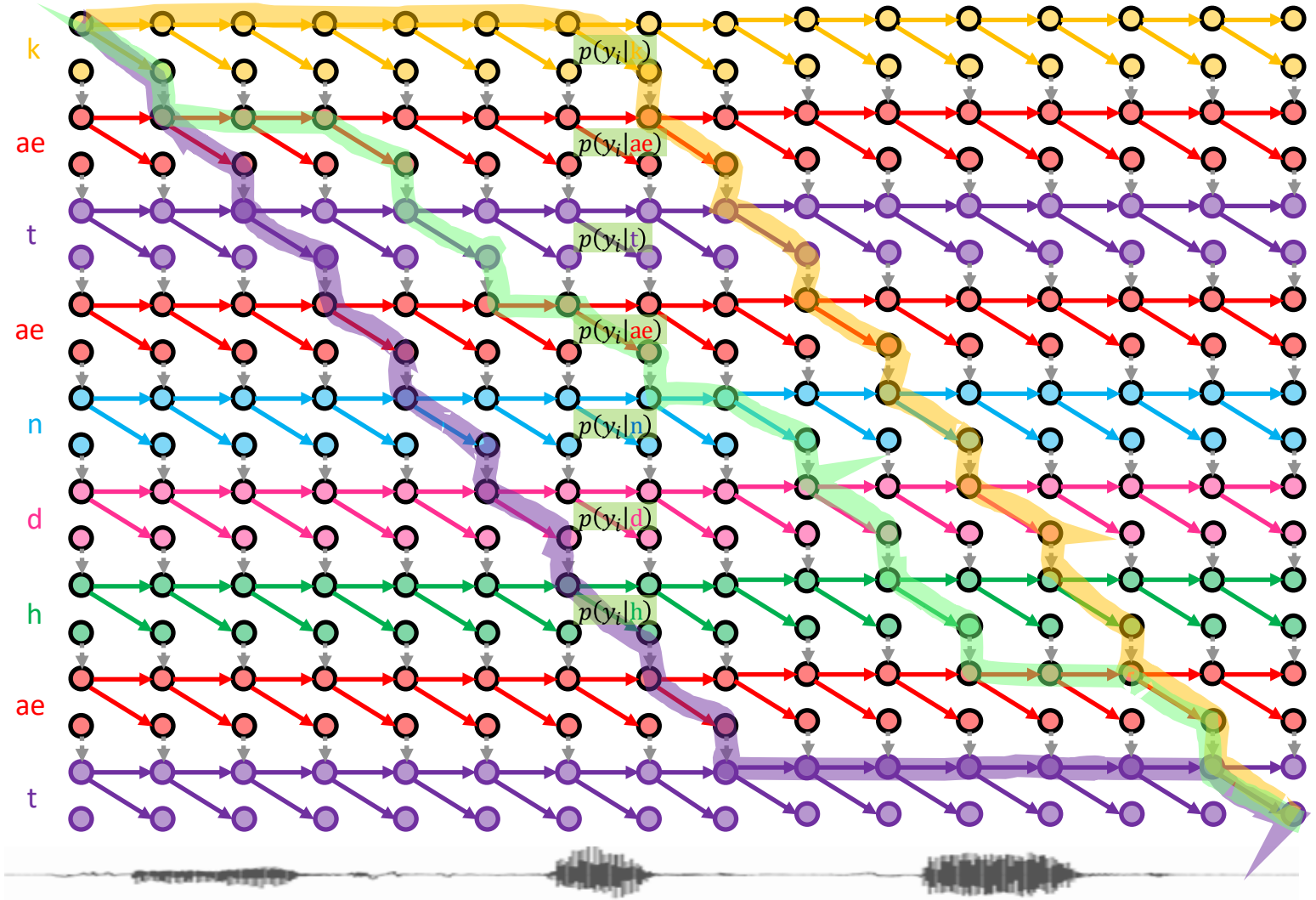


Composite HMM for “cat and hat”



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

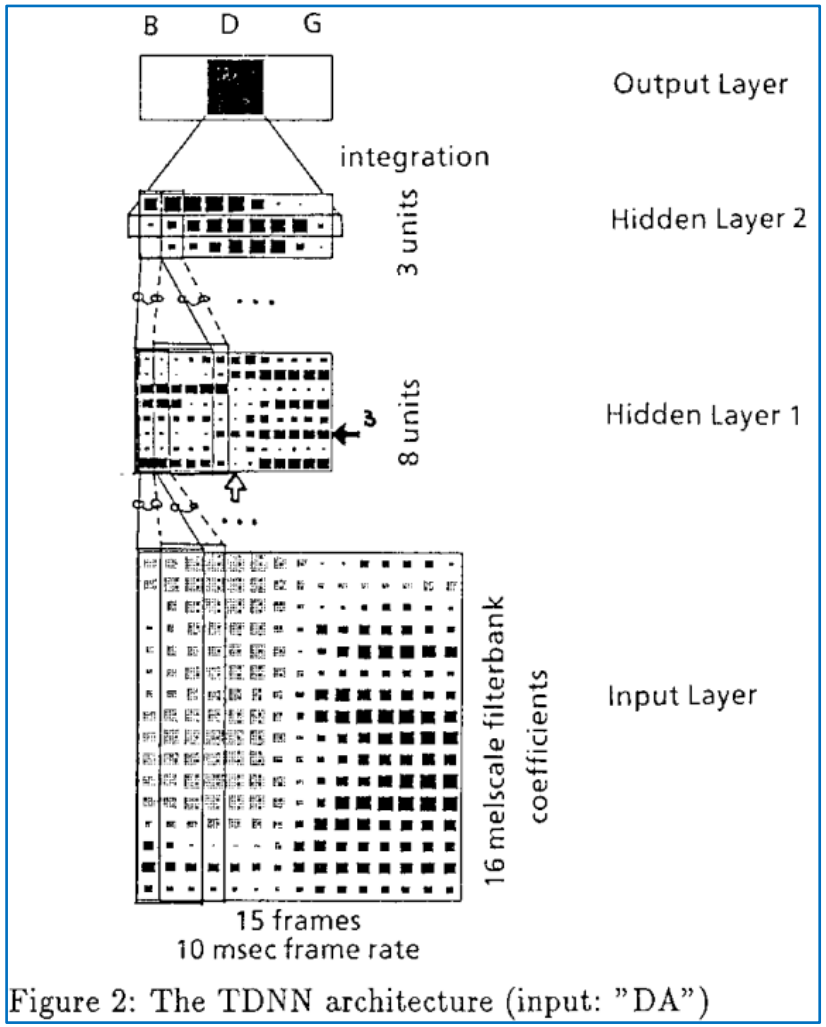
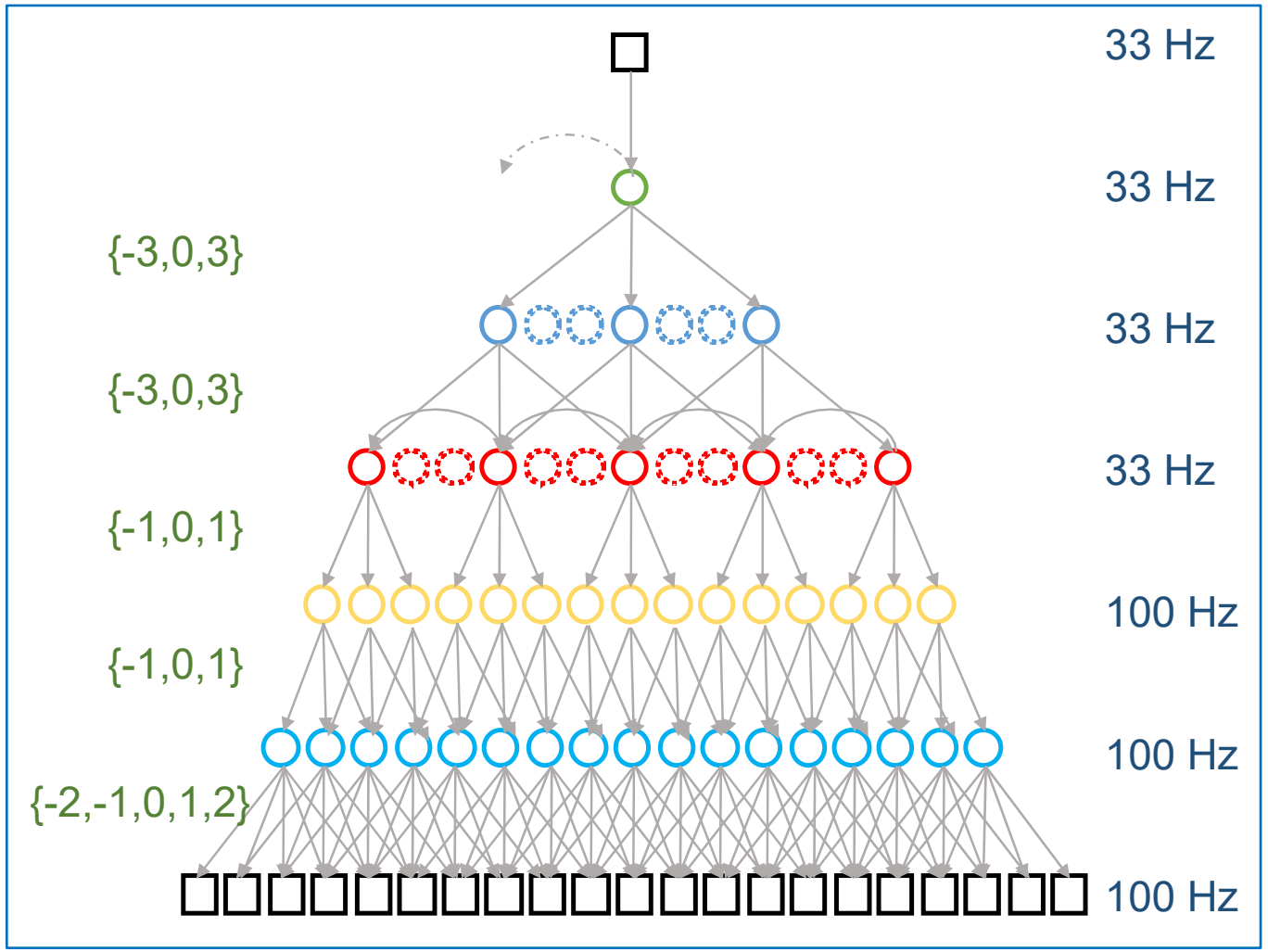
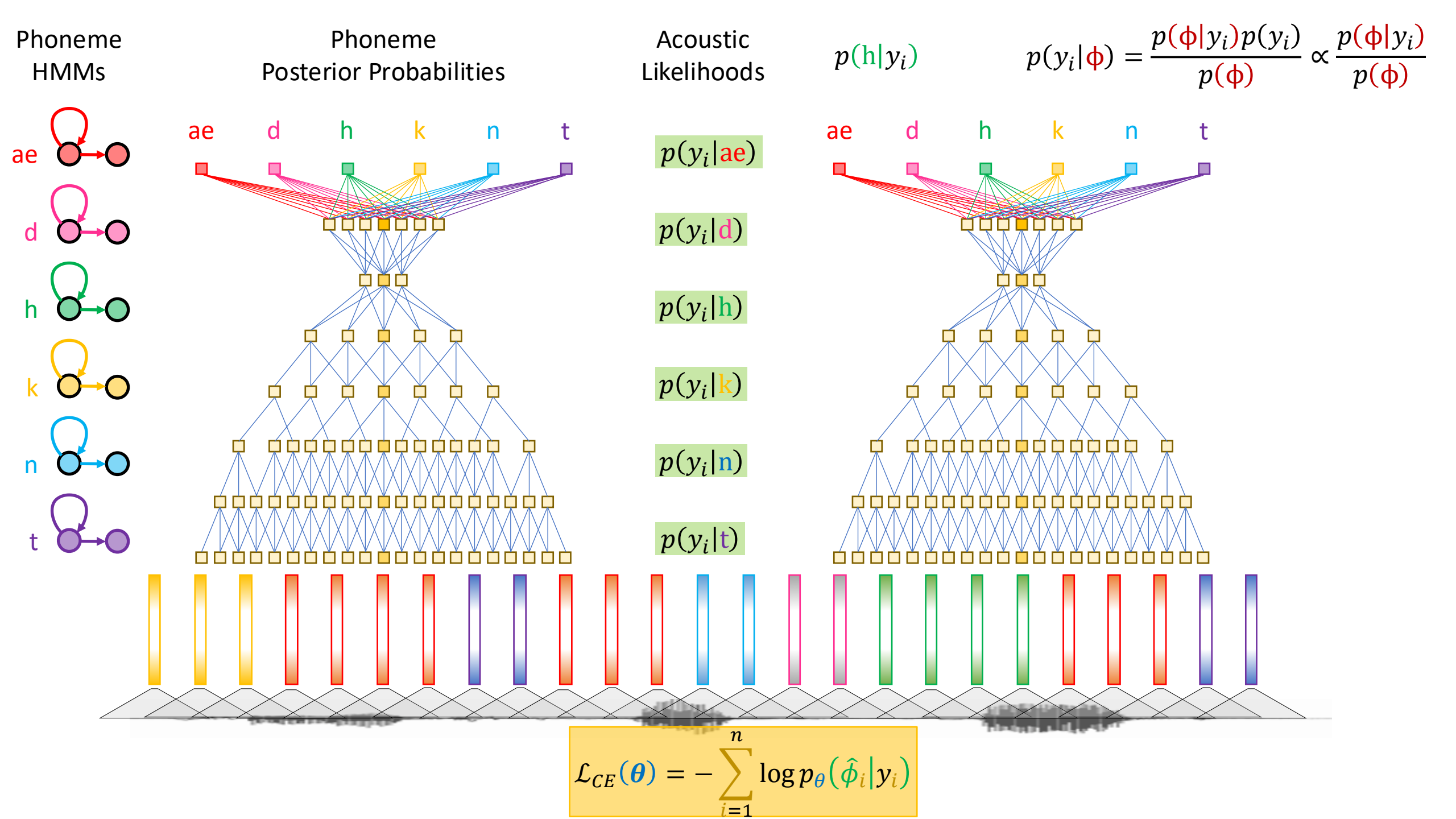


Figure 2: The TDNN architecture (input: "DA")




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

Maximum Mutual Information Estimation

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

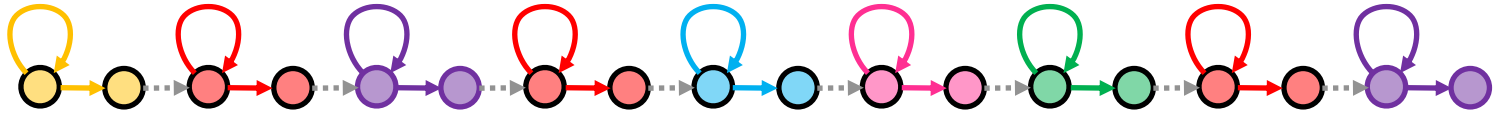


Purely sequence-trained neural networks for ASR based on lattice-free MMI

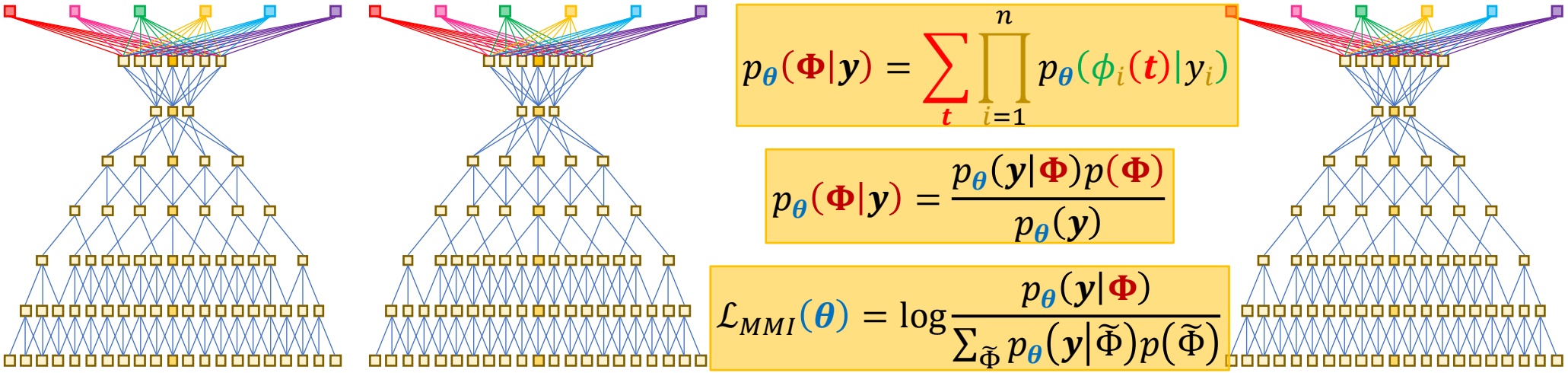
Daniel Povey[1,2], *Vijayaditya Peddinti*[1], *Daniel Galvez*[3], *Pegah Ghahremani*[1],
Vimal Manohar[1], *Xingyu Na*[4], *Yiming Wang*[1], *Sanjeev Khudanpur*[1,2]

[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



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



$$p_\theta(\mathbf{\Phi}|\boldsymbol{y}) = \sum_{\boldsymbol{t}} \prod_{i=1}^{n} p_\theta(\phi_i(\boldsymbol{t})|y_i)$$

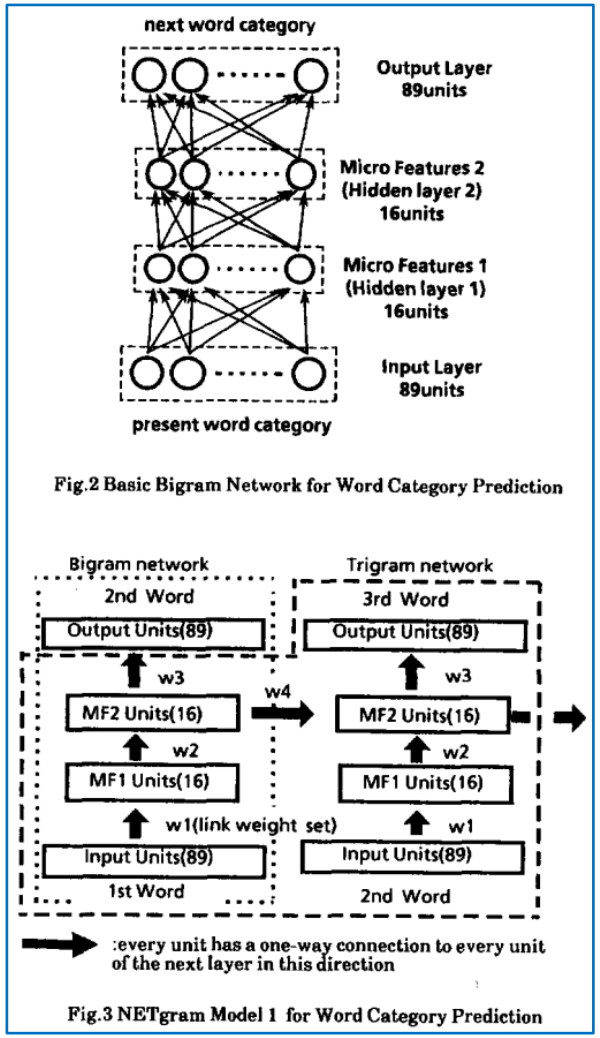
$$p_\theta(\mathbf{\Phi}|\boldsymbol{y}) = \frac{p_\theta(\boldsymbol{y}|\mathbf{\Phi})p(\mathbf{\Phi})}{p_\theta(\boldsymbol{y})}$$

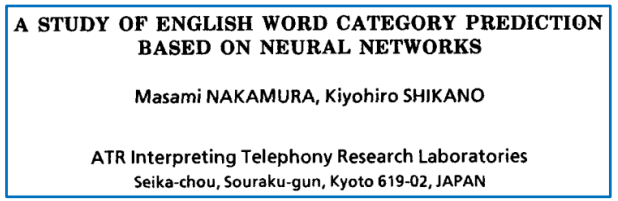
$$\mathcal{L}_{MMI}(\boldsymbol{\theta}) = \log \frac{p_\theta(\boldsymbol{y}|\mathbf{\Phi})}{\sum_{\tilde{\Phi}} p_\theta(\boldsymbol{y}|\tilde{\Phi})p(\tilde{\Phi})}$$

Language Modeling with Recurrent Neural Networks

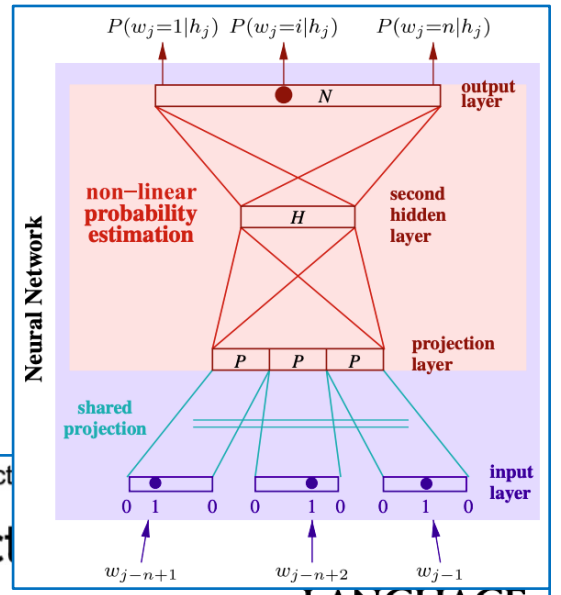
Efforts to Go Beyond *n*-gram Dependence in Language Models

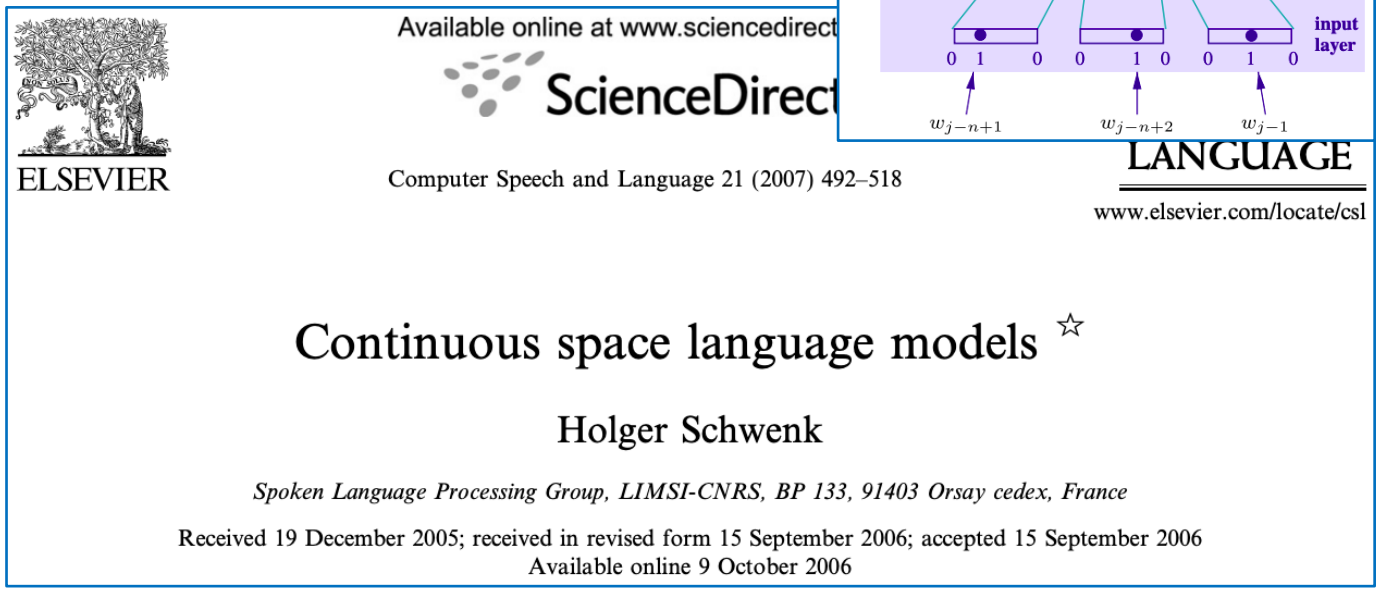
Using Neural Networks to Estimate $P(w_t|h_t)$





ICASSP 1989





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

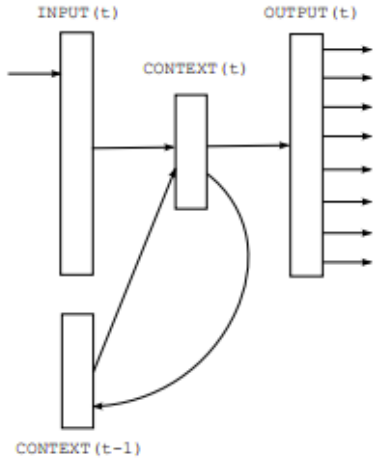
Tomáš Mikolov[1,2], *Martin Karafiát*[1], *Lukáš Burget*[1], *Jan "Honza" Černocký*[1], *Sanjeev Khudanpur*[2]

[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

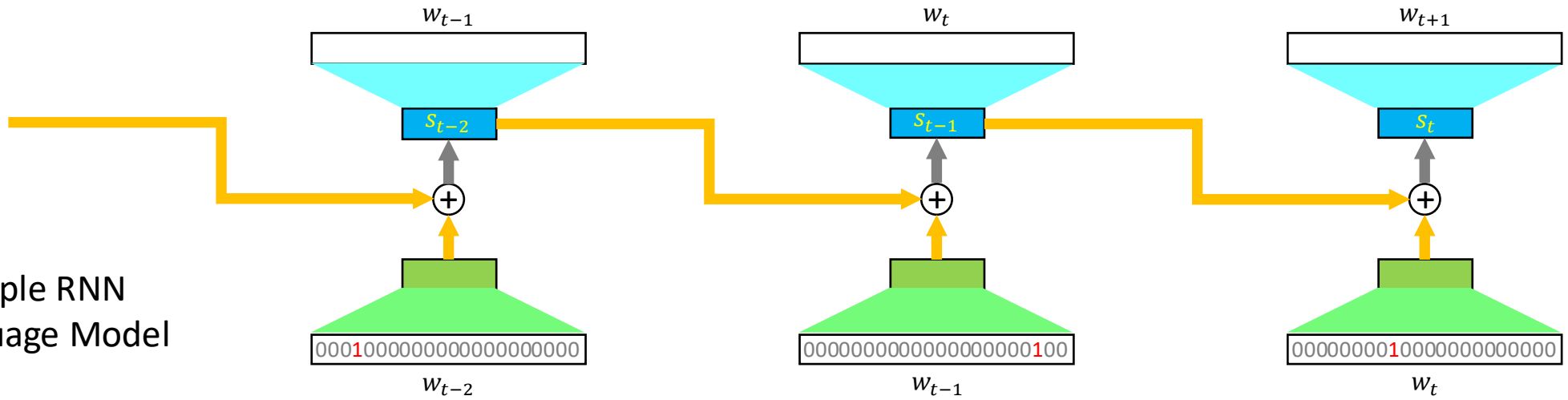
Abstract

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.

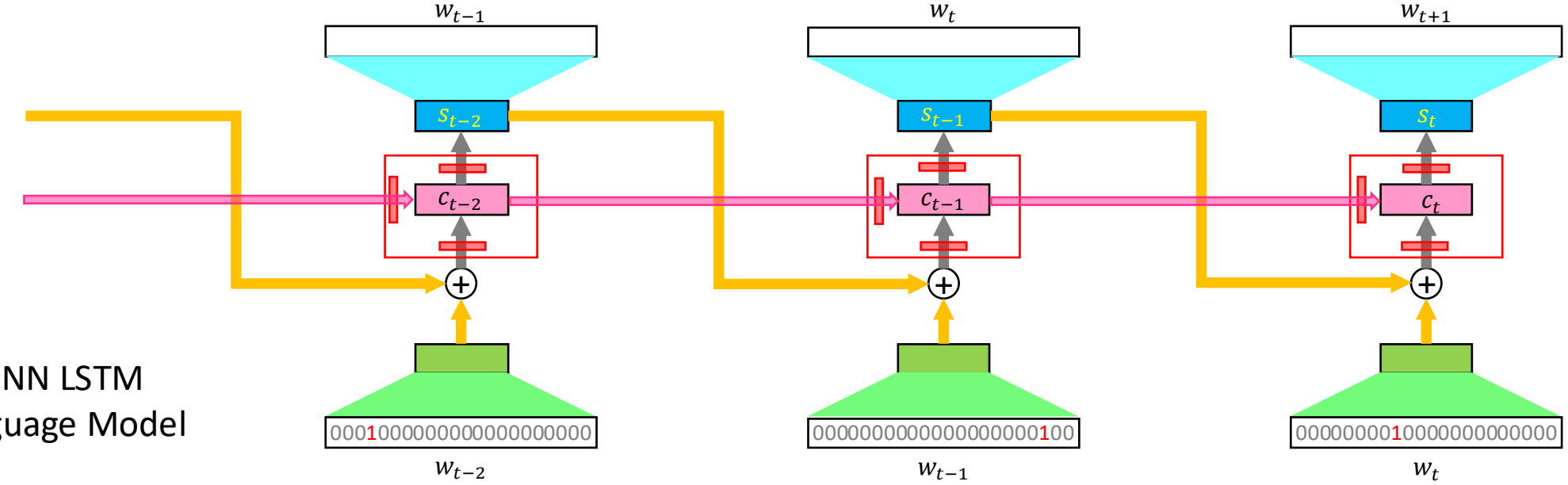
Index Terms: language modeling, recurrent neural networks, speech recognition



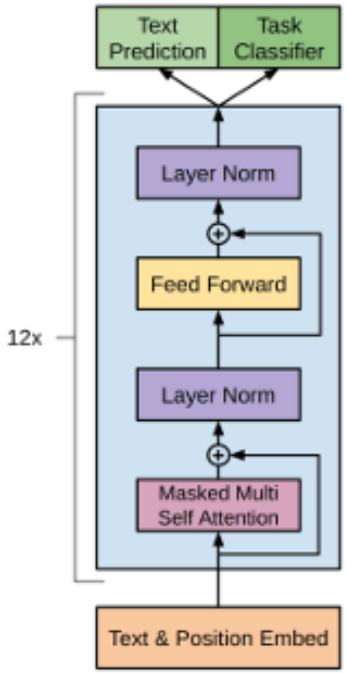
A Simple RNN Language Model

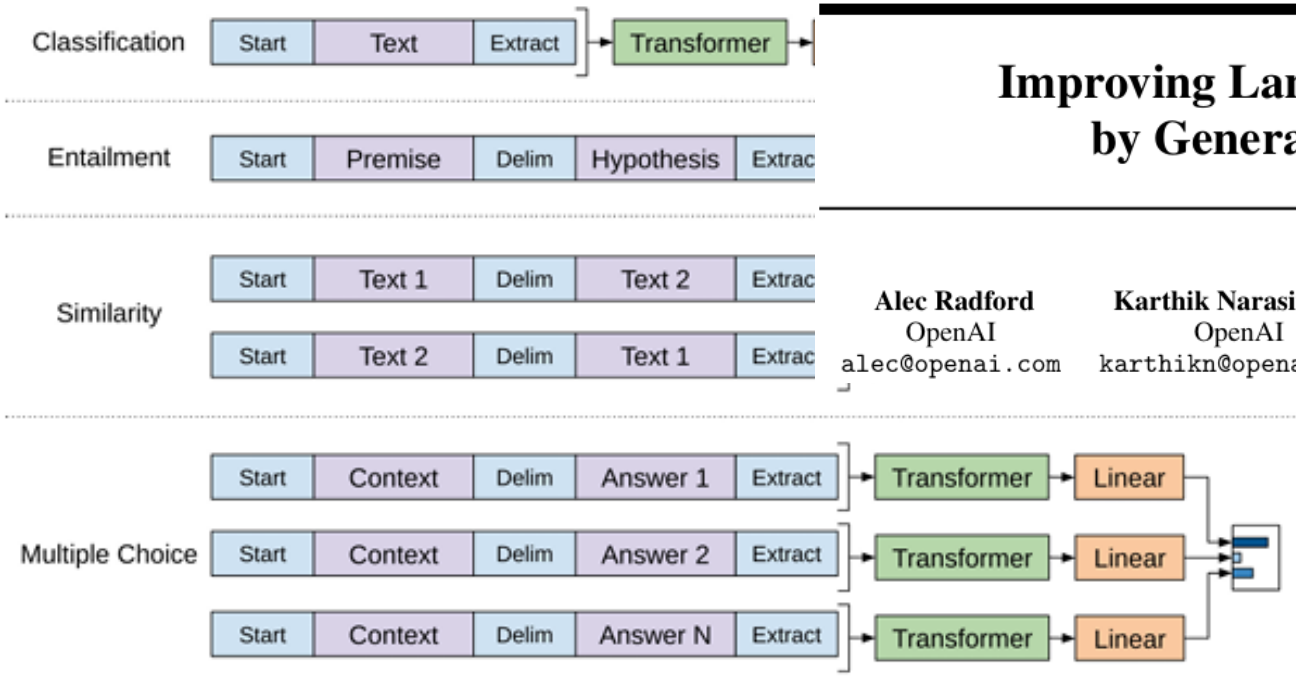


An RNN LSTM Language Model



Elman $\xrightarrow{2010}$ RNN-LM $\xrightarrow{2012}$ LSTM-LM $\xrightarrow{2018}$ GPT





Improving Language Understanding by Generative Pre-Training

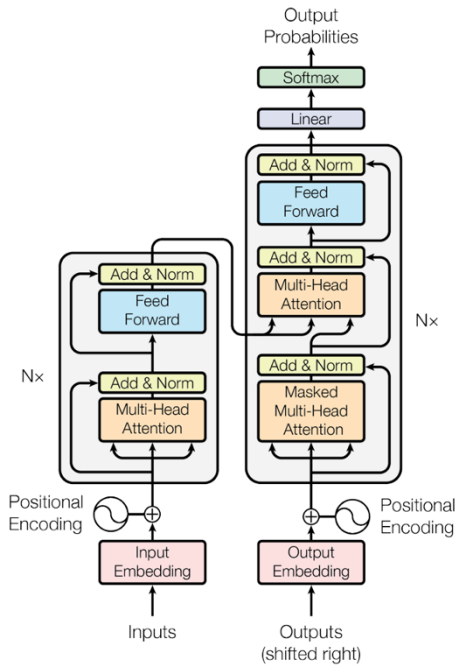
Alec Radford OpenAI alec@openai.com
Karthik Narasimhan OpenAI karthikn@openai.com
Tim Salimans OpenAI tim@openai.com
Ilya Sutskever OpenAI ilyasu@openai.com

3.1 Unsupervised pre-training

Given an unsupervised corpus of tokens $\mathcal{U} = \{u_1, \ldots, u_n\}$, we use a standard language modeling objective to maximize the following likelihood:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta) \quad (1)$$

where k is the size of the context window, and the conditional probability P is modeled using a neural network with parameters Θ. These parameters are trained using stochastic gradient descent [51].



Training Neural Networks without using HMM-Based Alignments

Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks

Alex Graves[1] ALEX@IDSIA.CH
Santiago Fernández[1] SANTIAGO@IDSIA.CH
Faustino Gomez[1] TINO@IDSIA.CH
Jürgen Schmidhuber[1,2] JUERGEN@IDSIA.CH

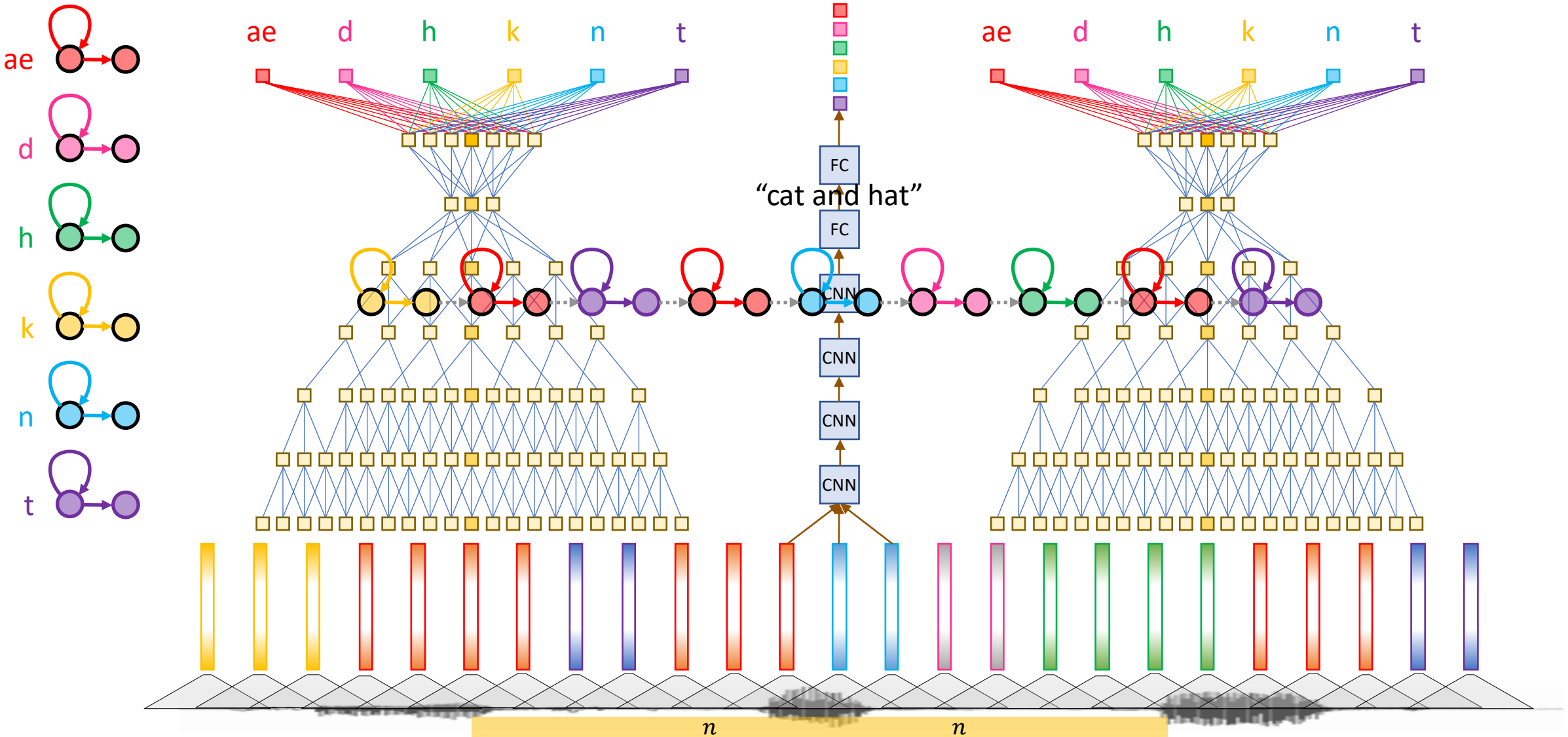
[1] Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Manno-Lugano, Switzerland
[2] Technische Universität München (TUM), Boltzmannstr. 3, 85748 Garching, Munich, Germany

Phoneme HMMs

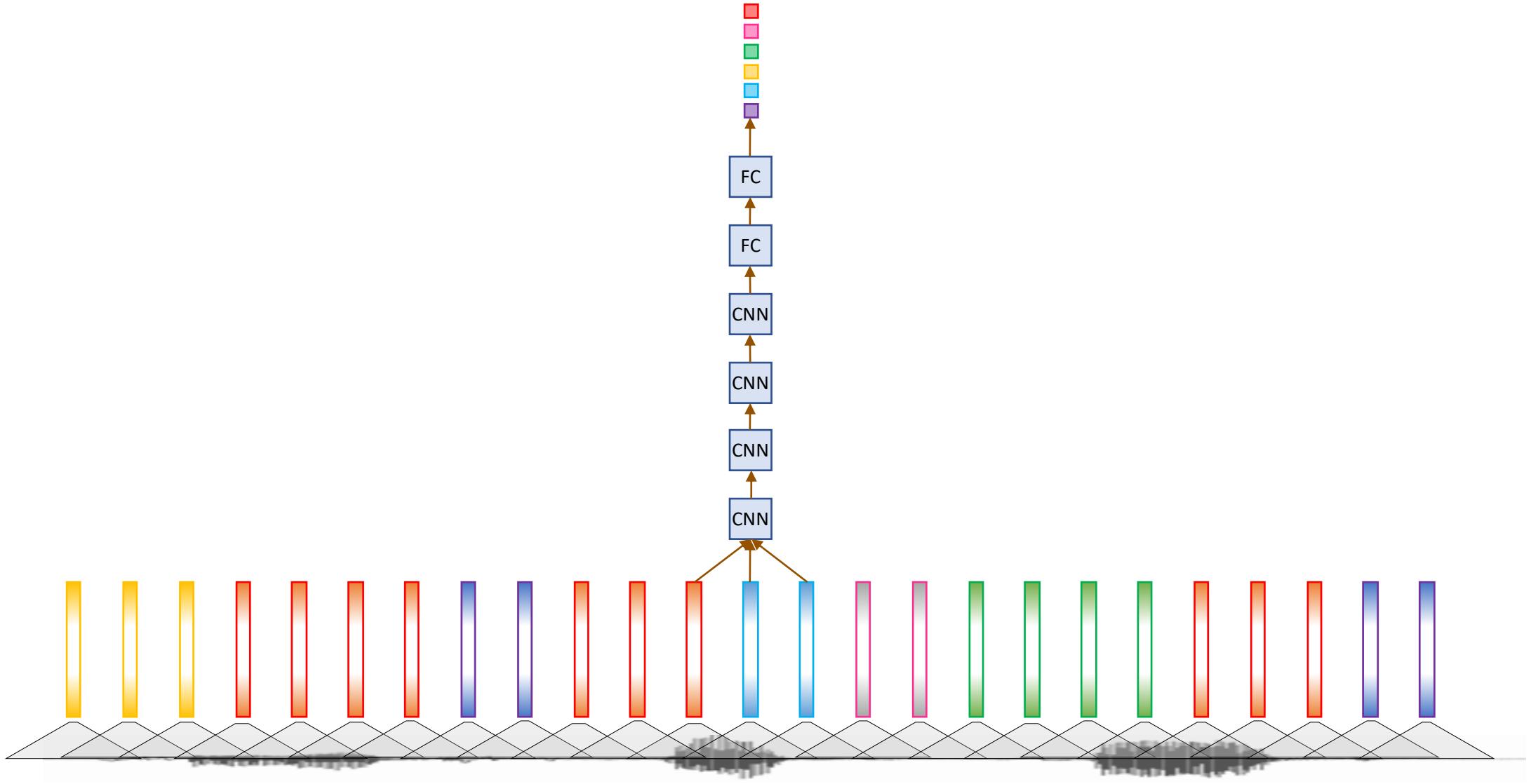
Phoneme Posterior Probabilities

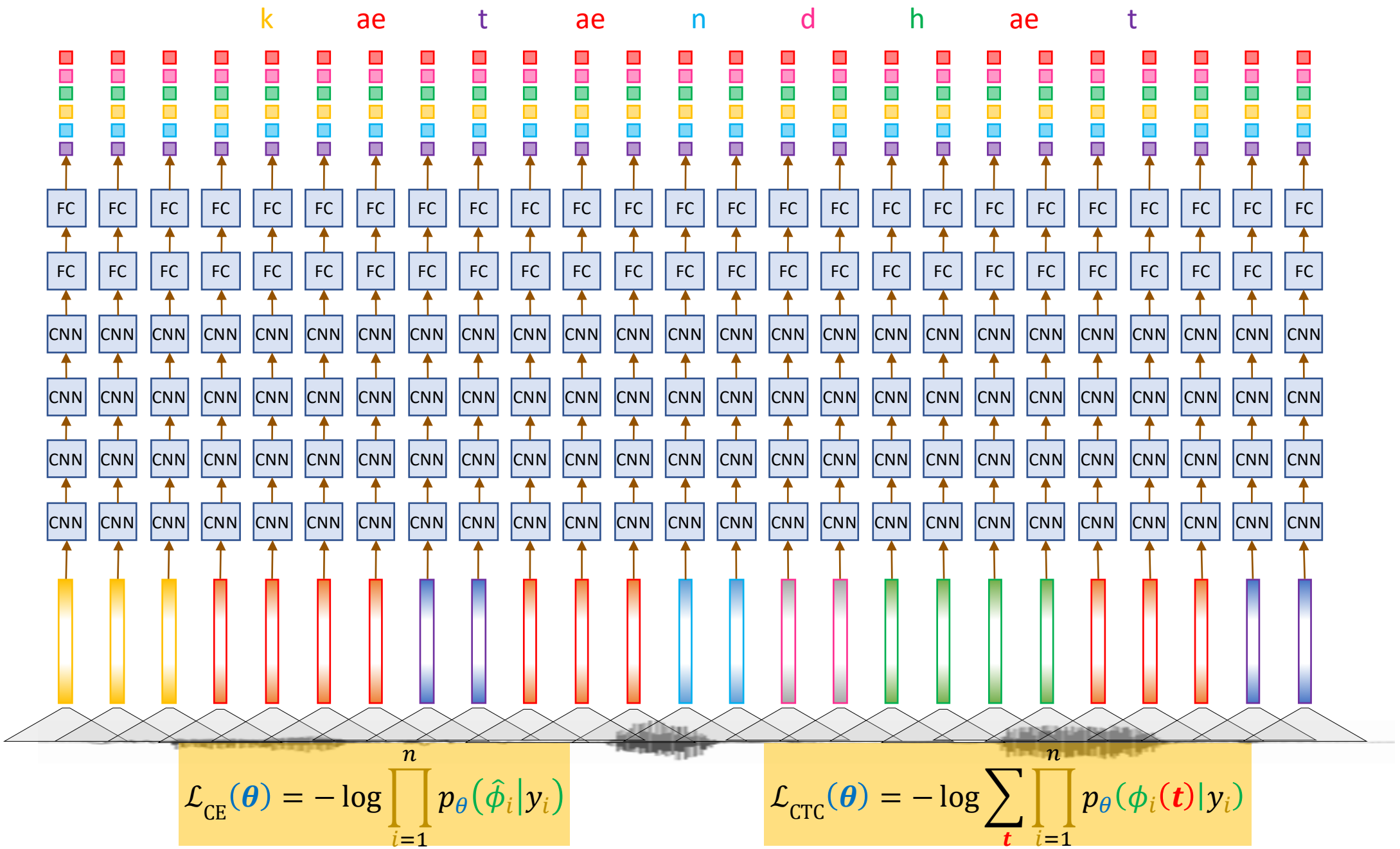
$p(\phi|y_i)$

$$p(y_i|\phi) = \frac{p(\phi|y_i)p(y_i)}{p(\phi)} \propto \frac{p(\phi|y_i)}{p(\phi)}$$



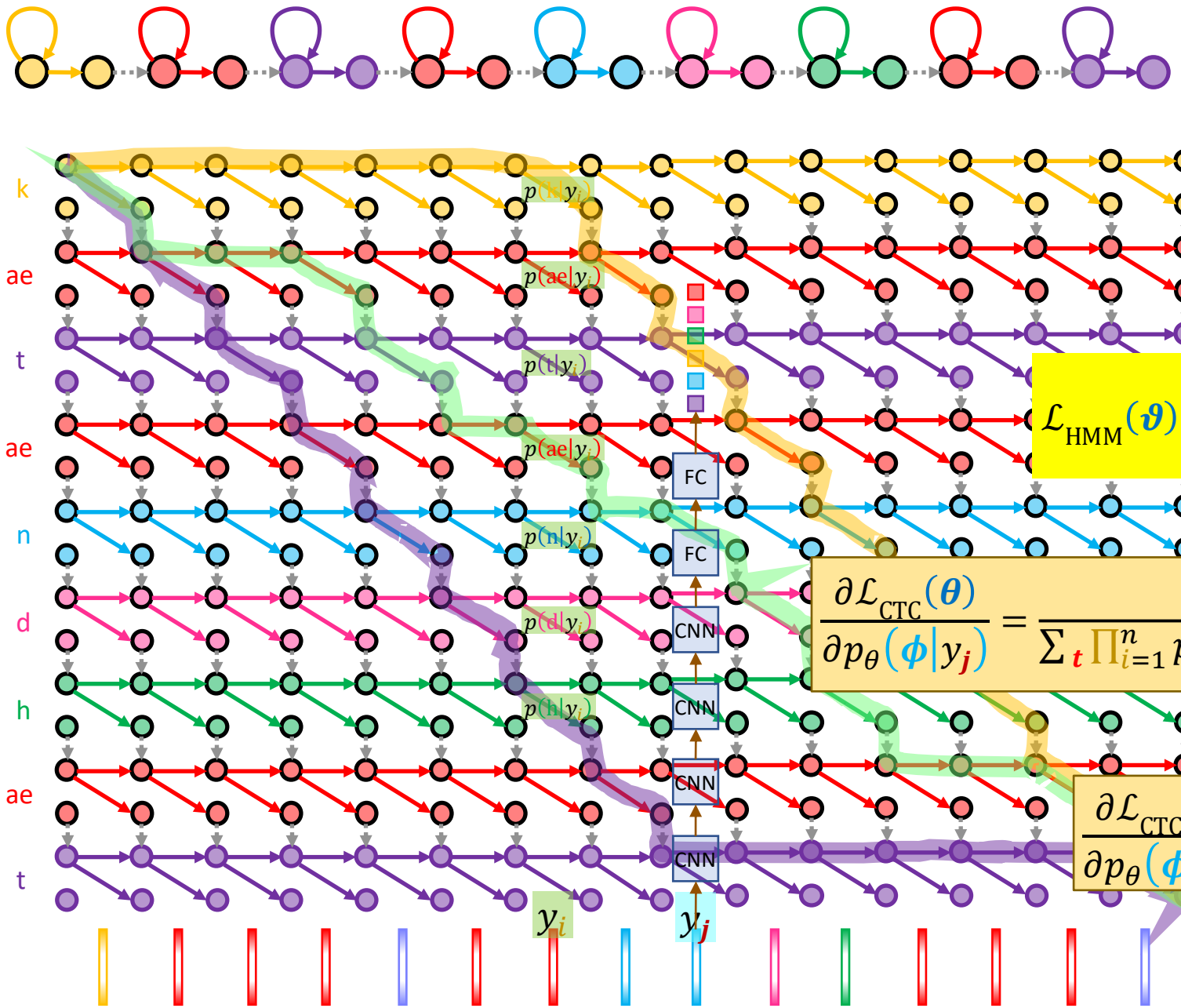
$$\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) = -\log \prod_{i=1}^{n} p_\theta(\hat{\phi}_i|y_i) = -\sum_{i=1}^{n} \log p_\theta(\hat{\phi}_i|y_i)$$


FC
FC
CNN
CNN
CNN
CNN


k
ae
t
ae
n
d
h
ae
t
FC
CNN
$\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) = -\log \prod_{i=1}^{n} p_\theta(\hat{\phi}_i | y_i)$
$\mathcal{L}_{\text{CTC}}(\boldsymbol{\theta}) = -\log \sum_{\boldsymbol{t}} \prod_{i=1}^{n} p_\theta(\phi_i(\boldsymbol{t}) | y_i)$

Calculating the CTC loss for “cat and hat”

Calculating the gradient of the CTC loss

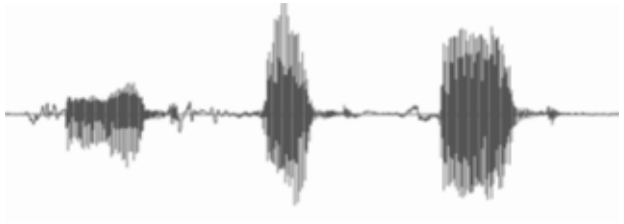


$$\mathcal{L}_{\text{CTC}}(\boldsymbol{\theta}) = -\log \sum_{\boldsymbol{t}} \prod_{i=1}^{n} p_\theta(\phi_i(\boldsymbol{t})|y_i)$$

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

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

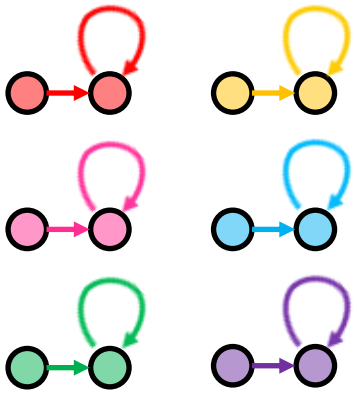
$$\frac{\partial \mathcal{L}_{\text{CTC}}(\boldsymbol{\theta})}{\partial p_\theta(\boldsymbol{\phi}|y_j)} = -\frac{1}{p_\theta(\boldsymbol{\phi}|y_j)} \frac{\sum_{\boldsymbol{t}:\phi_j(\boldsymbol{t})=\boldsymbol{\phi}} \prod_{i=1}^{n} p_\theta(\phi_i(\boldsymbol{t})|y_i)}{\sum_{\boldsymbol{t}} \prod_{i=1}^{n}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



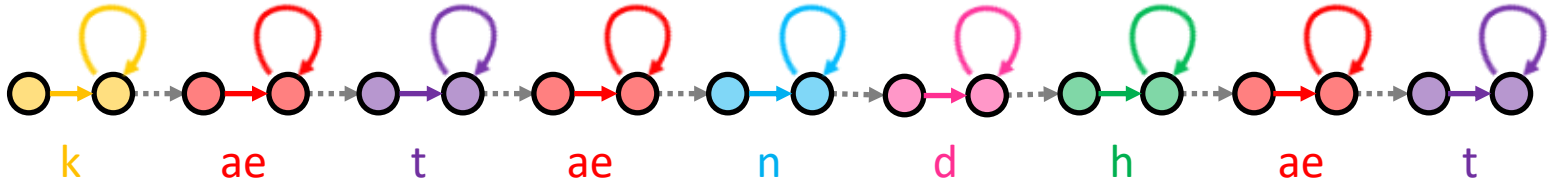
cat and hat

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

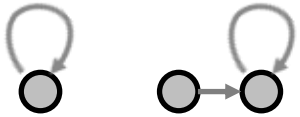
Phoneme HMMs



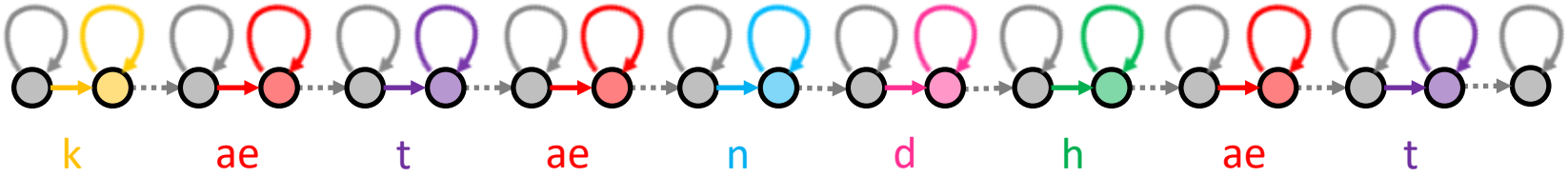
Composite HMM for “cat and hat”

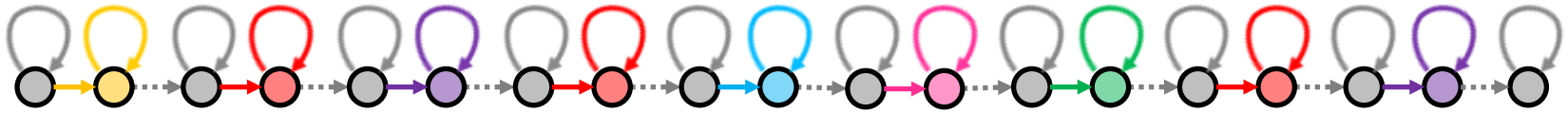


The CTC “Blank” Symbol (β)



FSA of permissible CTC strings for “cat and hat”



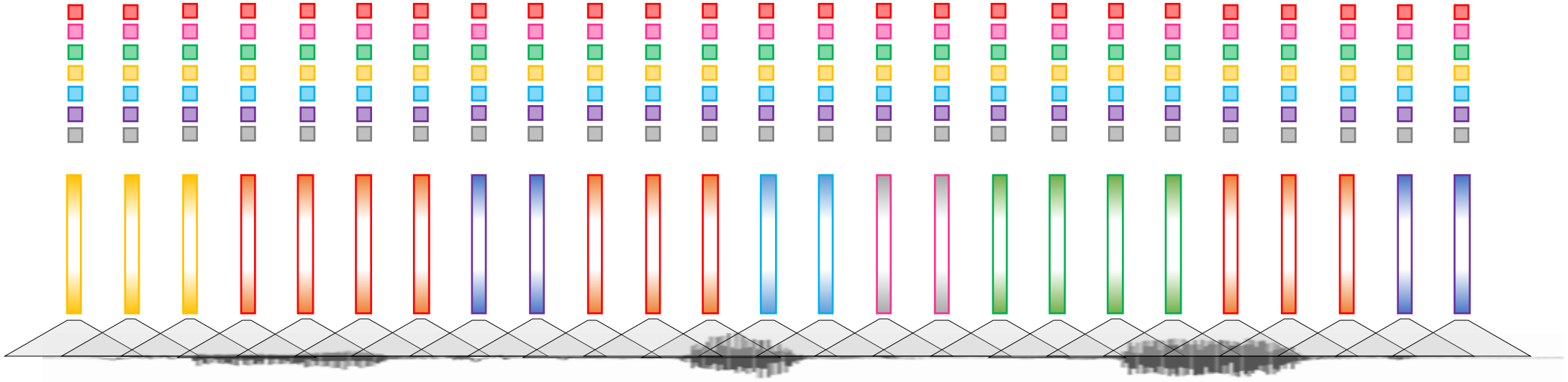


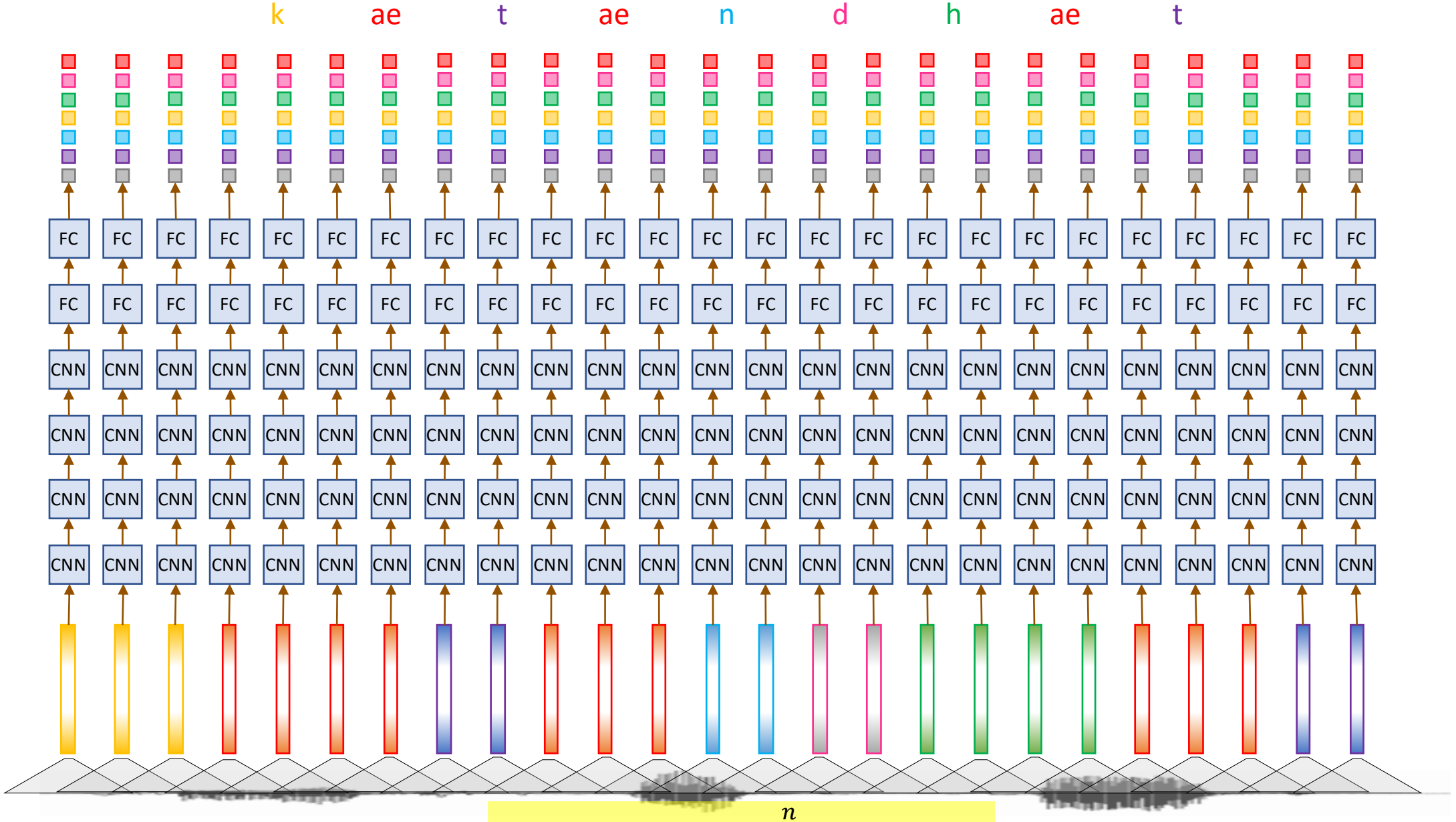
HMM State Sequences

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





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

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

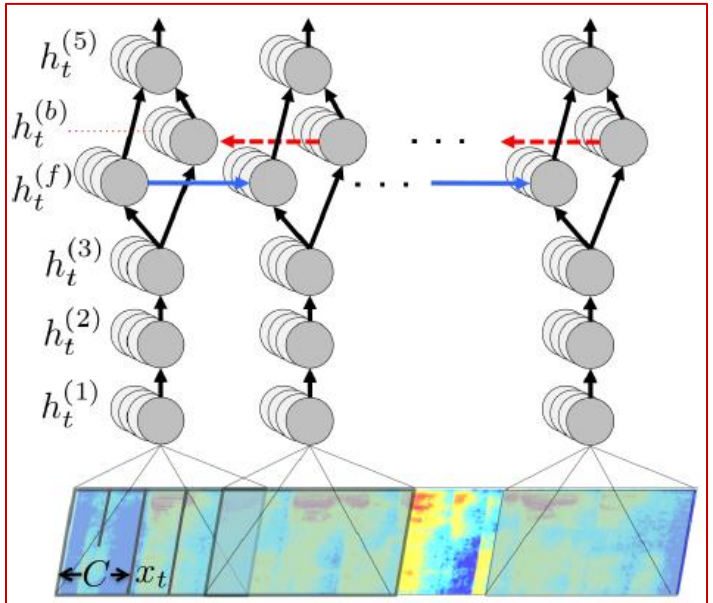
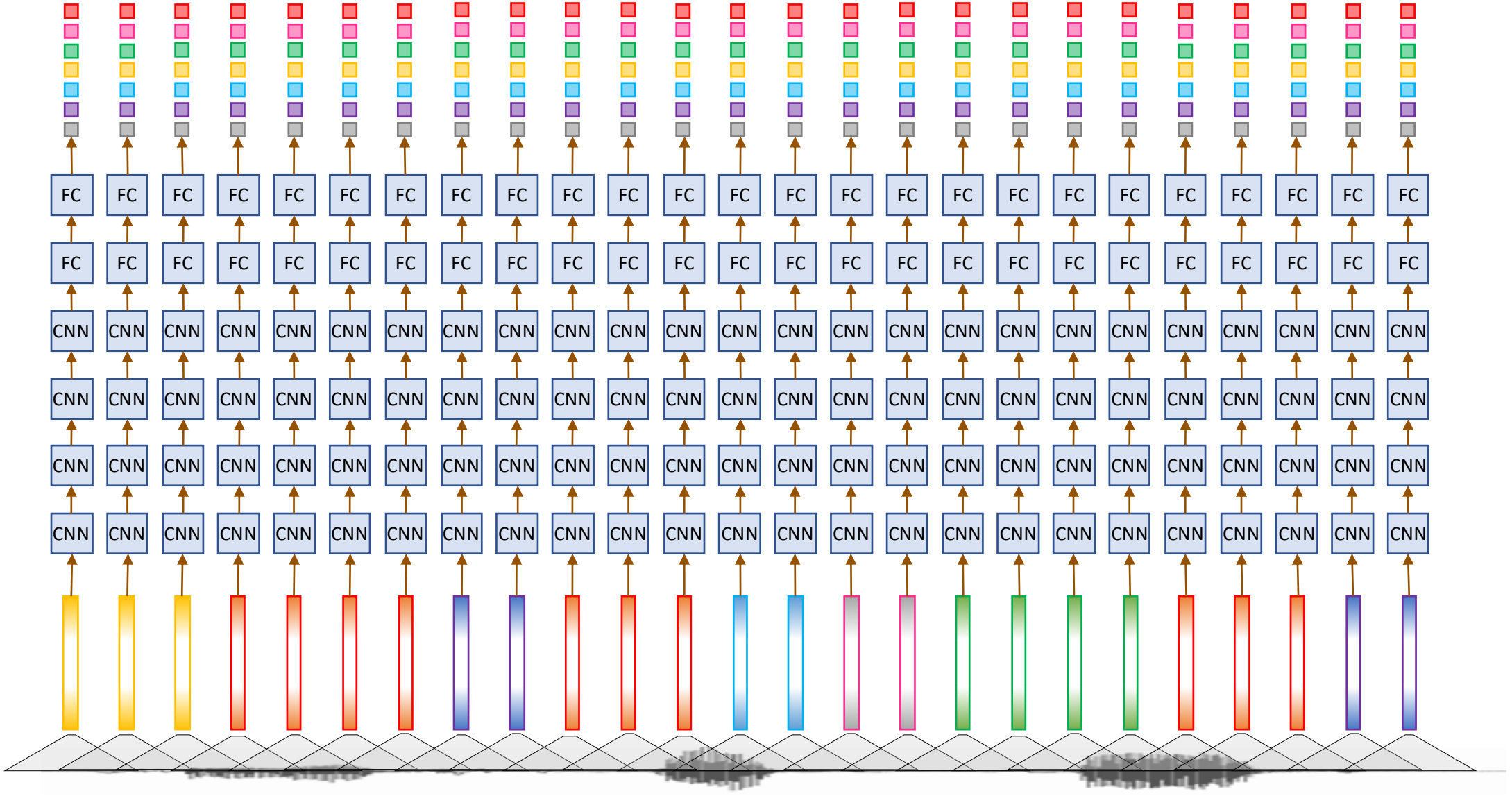
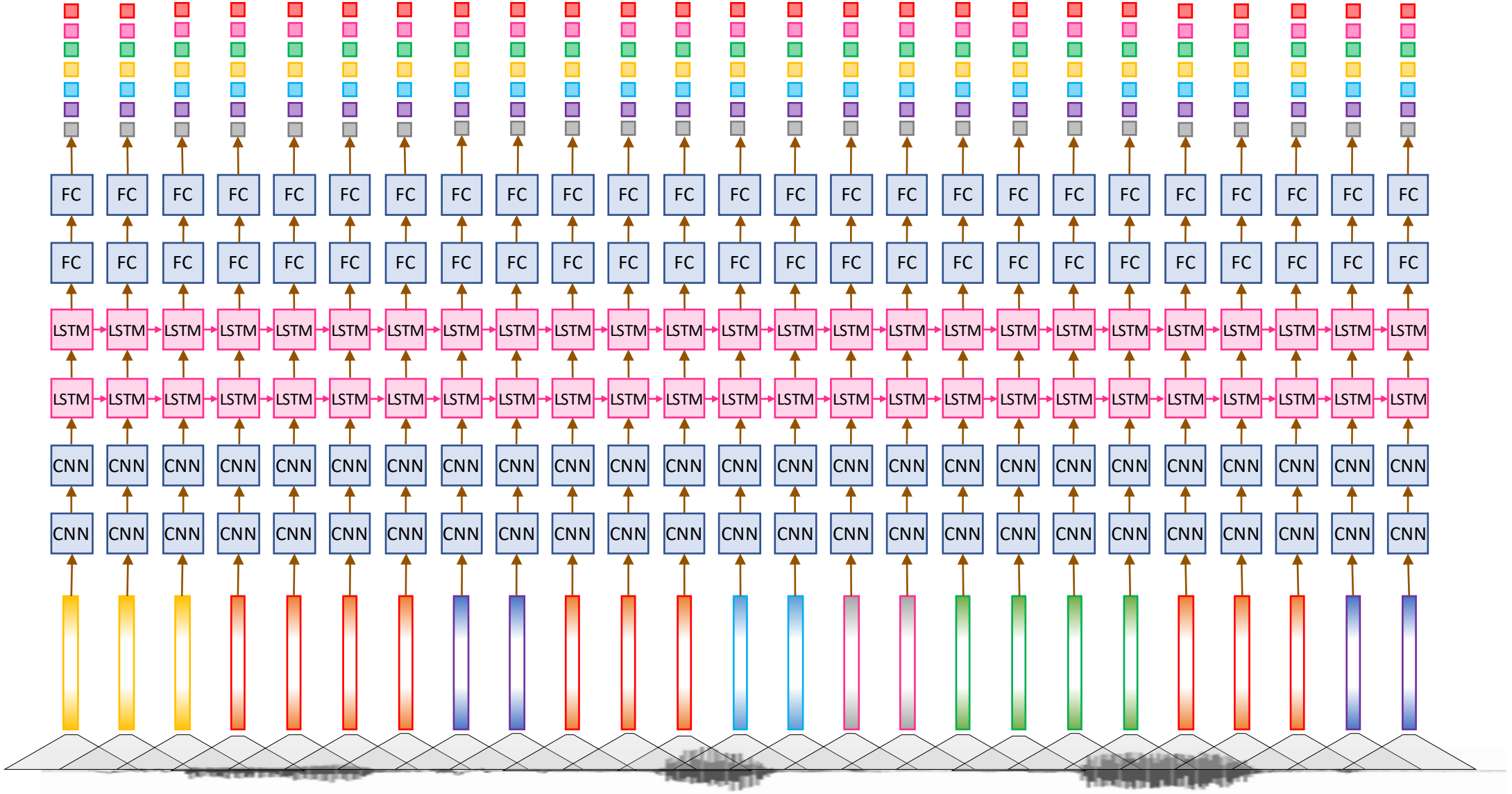


Figure 1: Structure of our RNN model and notation.

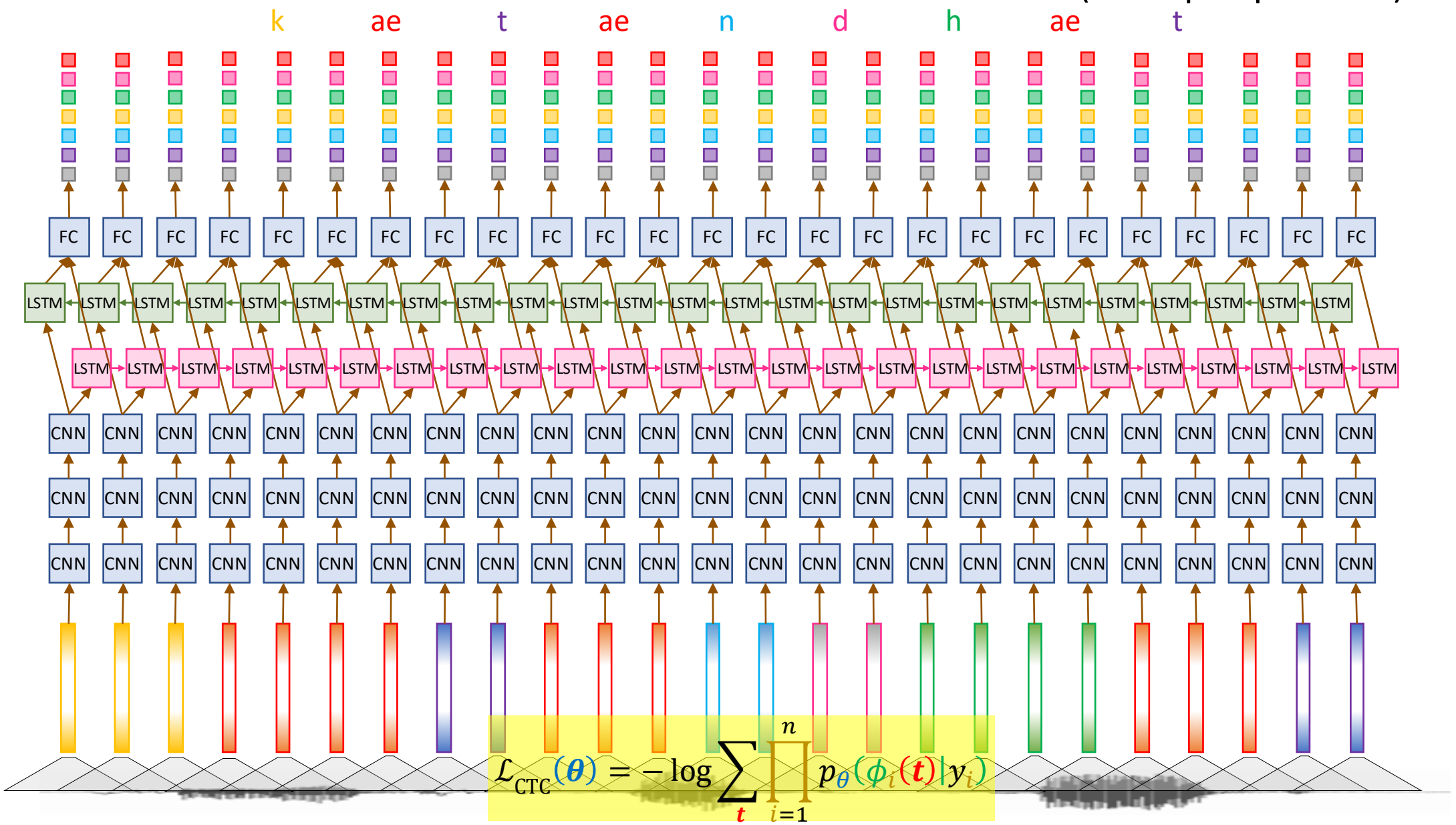
The CNN Architecture



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

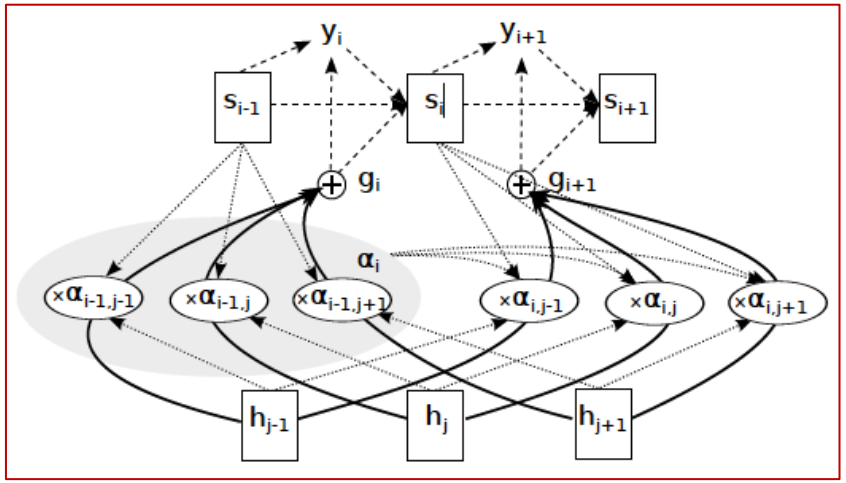
Jan Chorowski
University of Wrocław, Poland
jan.chorowski@ii.uni.wroc.pl

Dzmitry Bahdanau
Jacobs University Bremen, Germany

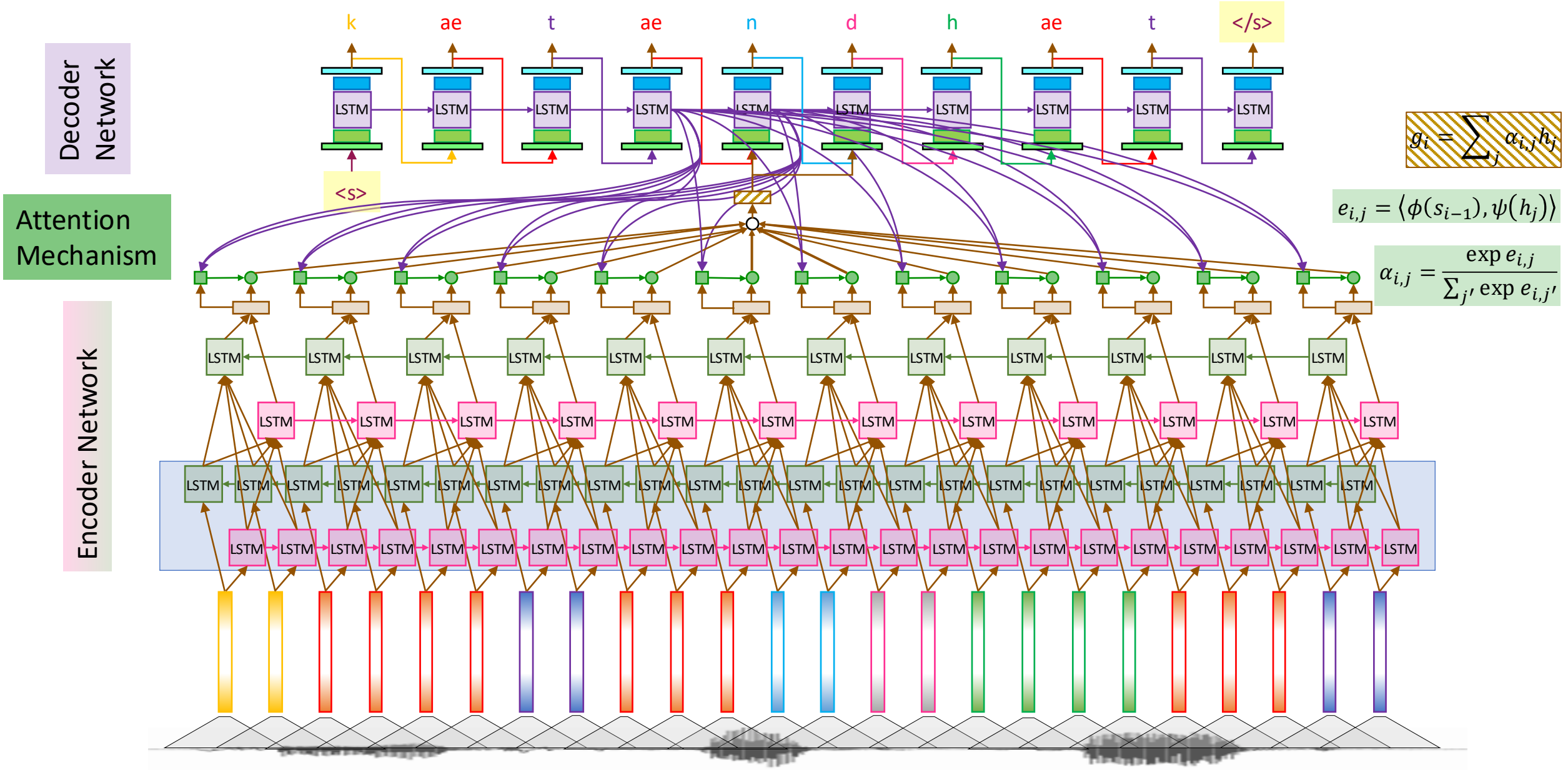
Dmitriy Serdyuk
Université de Montréal

Kyunghyun Cho
Université de Montréal

Yoshua Bengio
Université de Montréal
CIFAR Senior Fellow



An Encoder-Decoder Architecture with Attention



Recurrent Neural Network Transducers

Coming full circle to

the basic mathematical formulation of automatic speech recognition

There was another paper in 2012 …

Sequence Transduction with Recurrent Neural Networks

Alex Graves GRAVES@CS.TORONTO.EDU
Department of Computer Science, University of Toronto, Canada

ICML 2012 W

ICASSP 2019

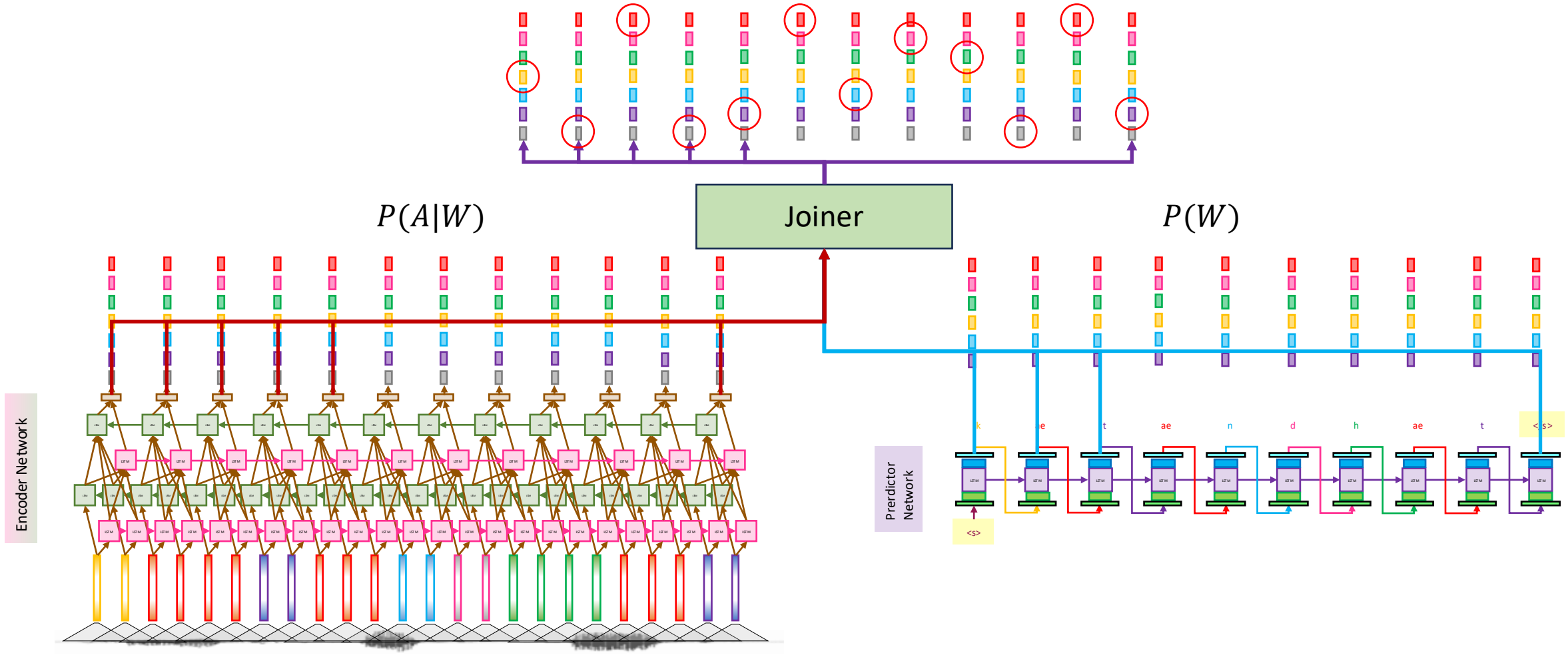
STREAMING END-TO-END SPEECH RECOGNITION FOR MOBILE DEVICES

Yanzhang He, Tara N. Sainath*, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, Qiao Liang, Deepti Bhatia, Yuan Shangguan, Bo Li, Golan Pundak, Khe Chai Sim, Tom Bagby, Shuo-yiin Chang, Kanishka Rao, Alexander Gruenstein*

Google, Inc., USA

{yanzhanghe, tsainath}@google.com

Combining a fully neural acoustic model with a fully neural language model



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)
- Recurrent Neural Network Transducers: Graves (2012) to He et al (2019)