

# Algorithms for Speech and Natural Language Processing



Neil Zeghidour

Phd Student, FAIR Paris, CoML (INRIA/ENS)

 The image part with relationship ID rId2 was not found in the file.

 The image part with relationship ID rId2 was not found in the file.

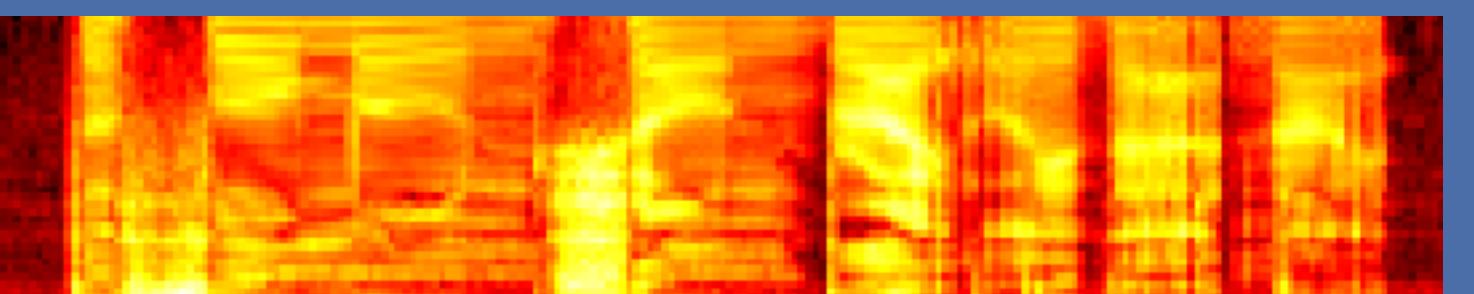
# End-to-end speech recognition



## HMM-DNN Recap



- Extract speech features

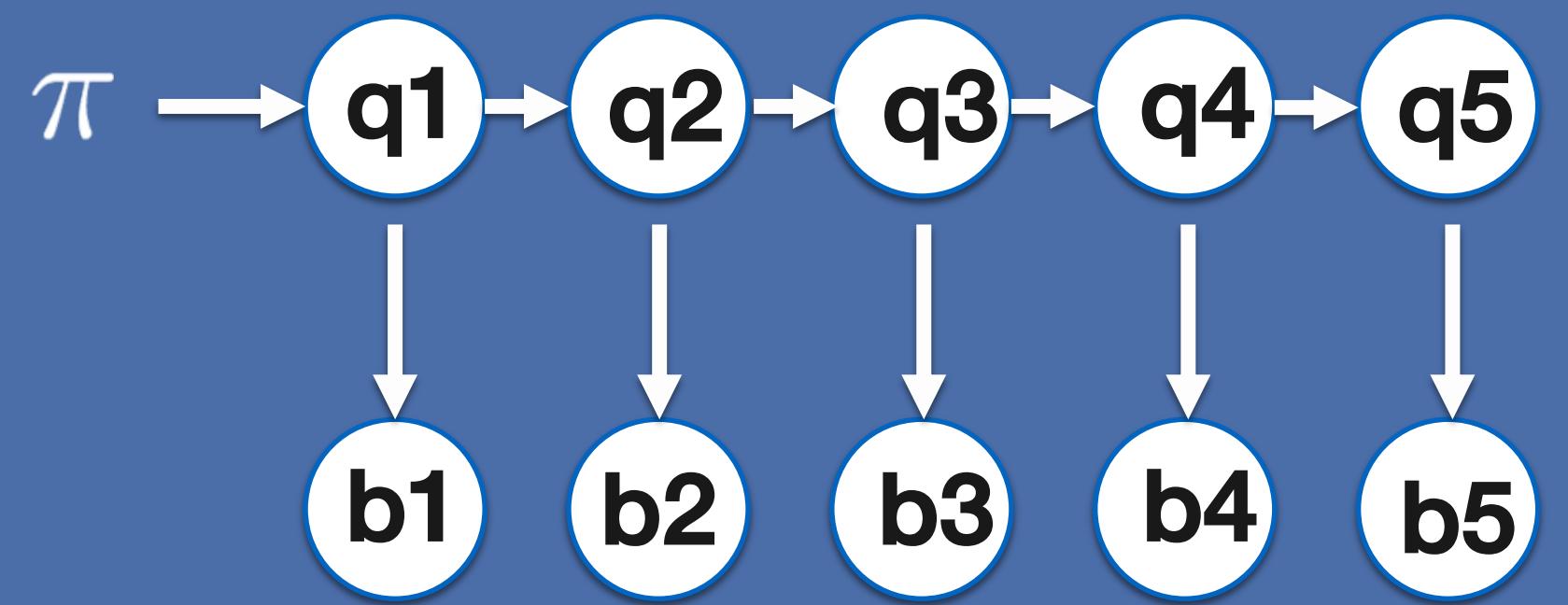




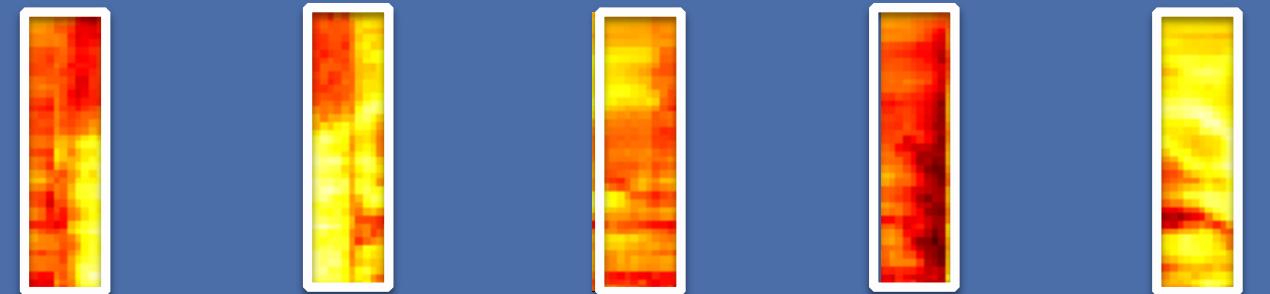
## HMM-DNN Recap

- Extract speech features
- Train a Hidden Markov Model with Gaussian Mixture Model for emissions

**K-Æ-B**

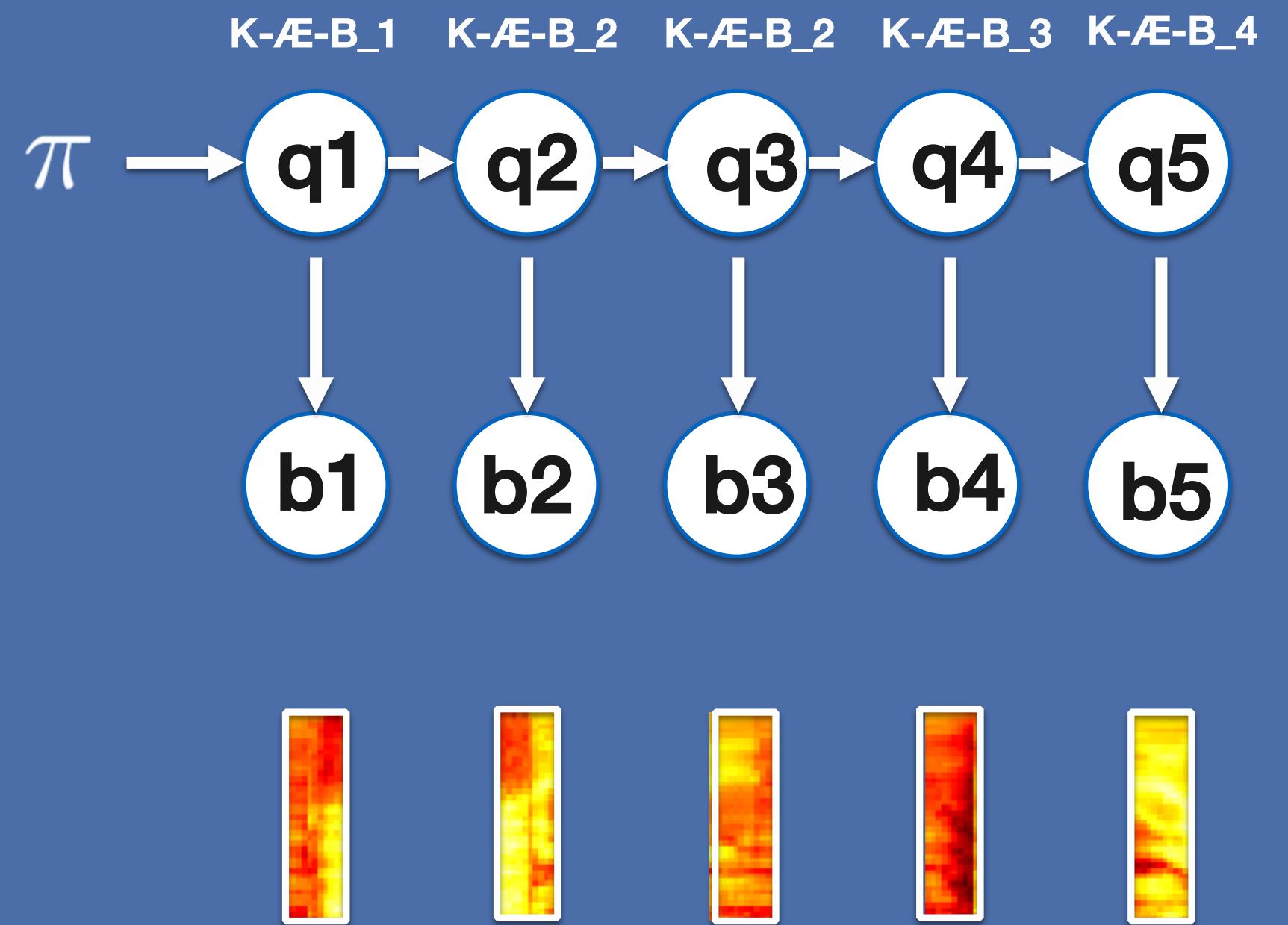


$$b_j \sim \sum_{m=1}^M c_{jm} \mathcal{N}(\mu_{jm}, \Sigma_{jm})$$



## HMM-DNN Recap

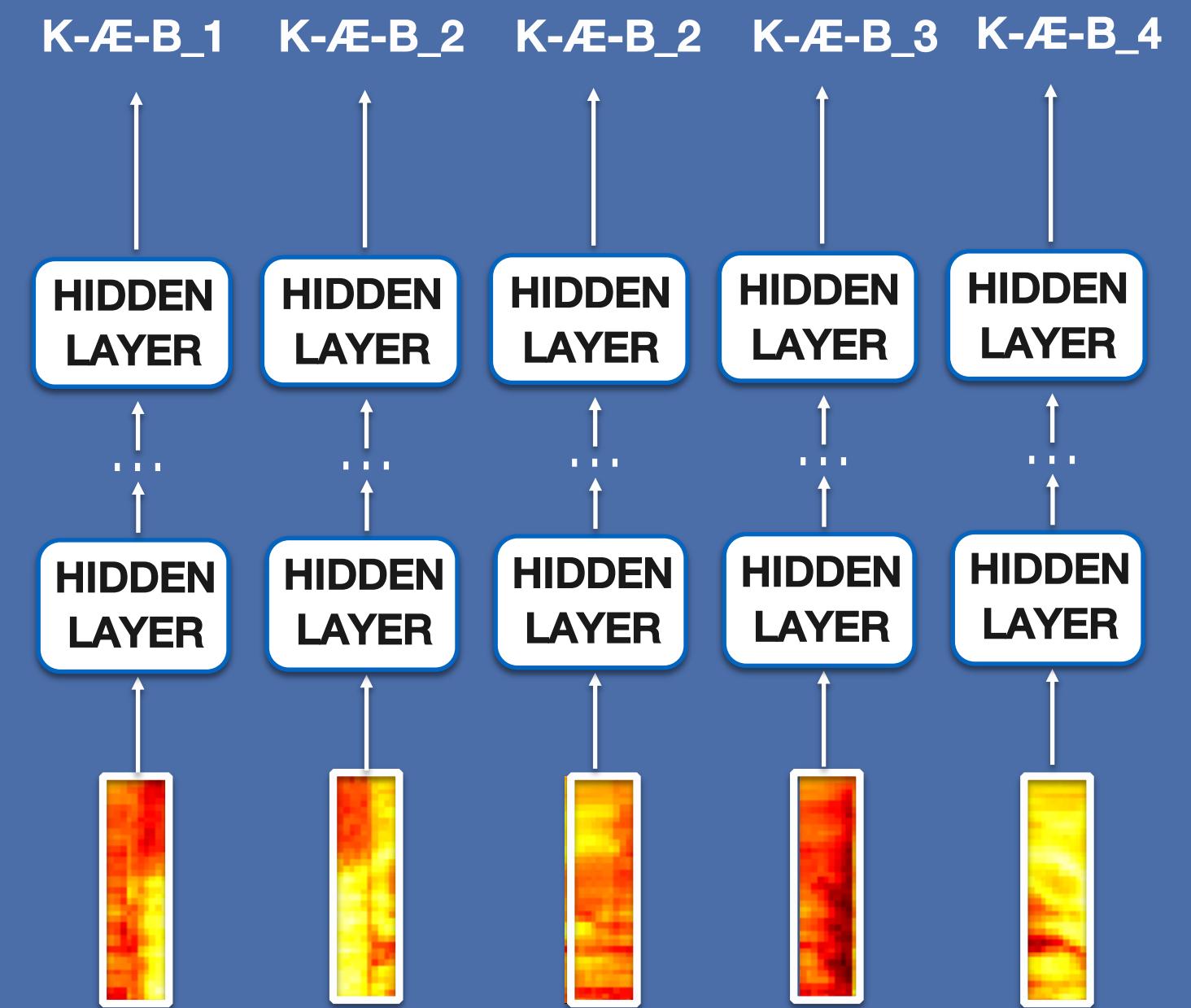
- Extract speech features
- Train a Hidden Markov Model with Gaussian Mixture Model for emissions
- Use the Viterbi algorithm to extract an alignment of feature frames with states





## HMM-DNN Recap

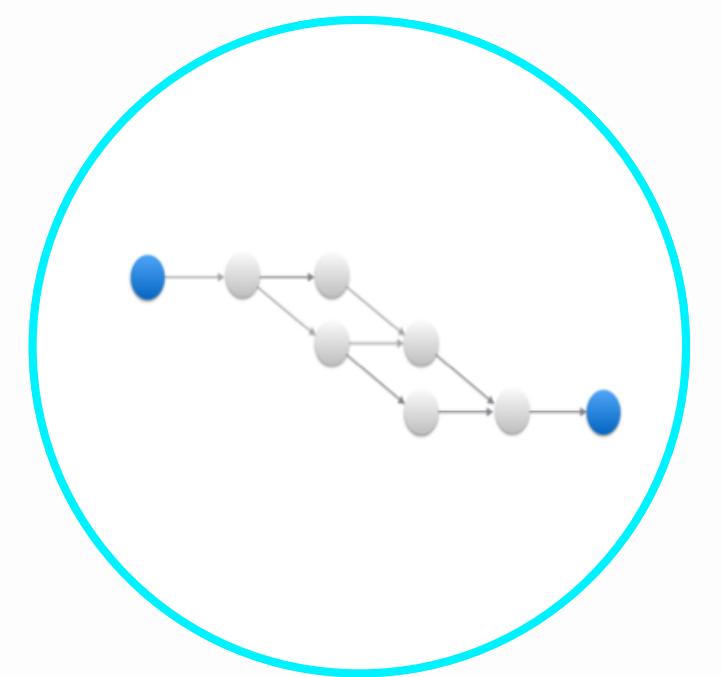
- Extract speech features
- Train a Hidden Markov Model with Gaussian Mixture Model for emissions
- Use the Viterbi algorithm to extract an alignment of feature frames with states
- Train a Neural Network to predict each state from its frame





## End-to-end training

- One neural network
- No alignment
- No speech features



## End-to-end speech recognition

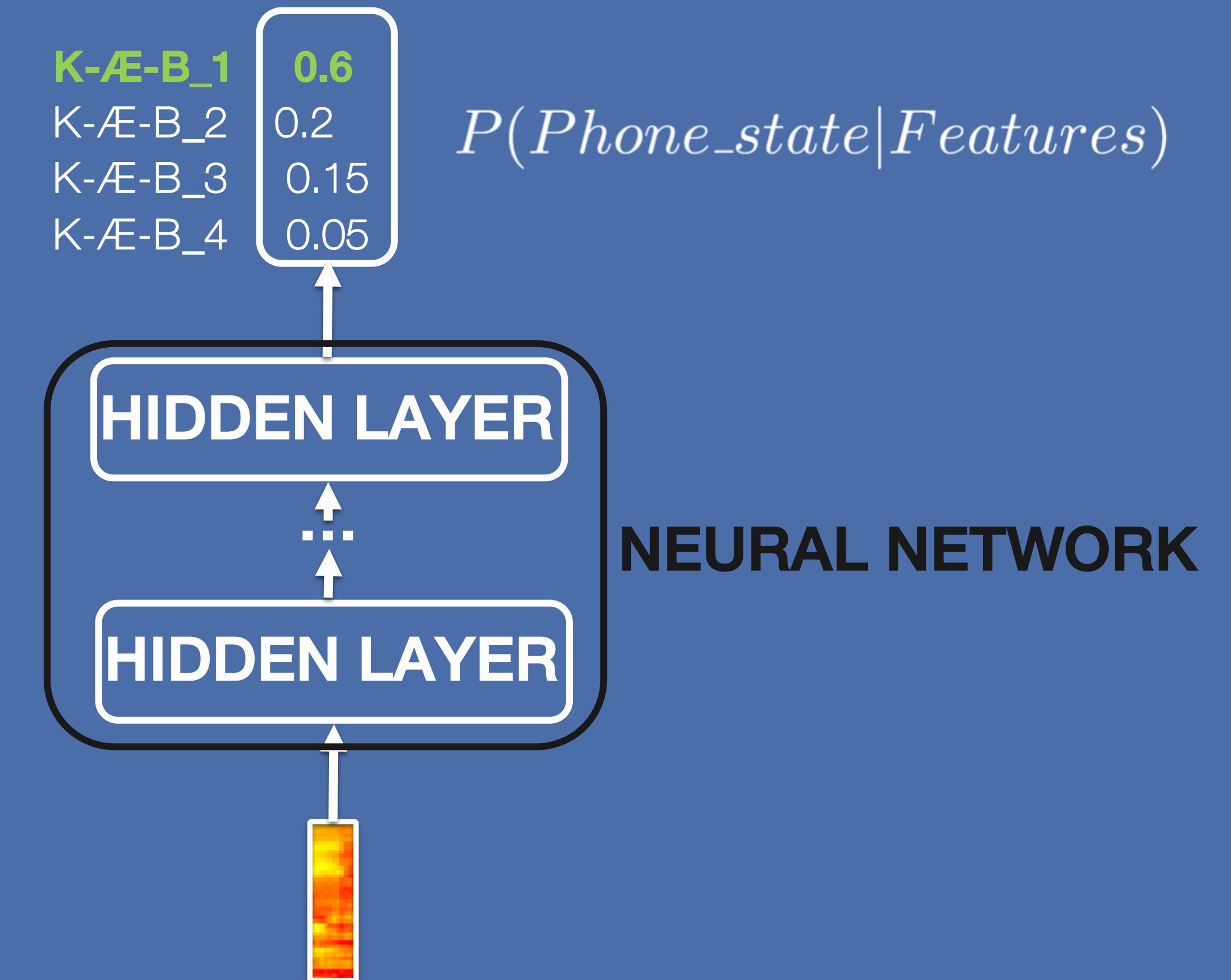
- Connectionist Temporal Classification
- Deep acoustic models for end-to-end speech recognition
- Speech recognition from raw representations



## Training a DNN without states?



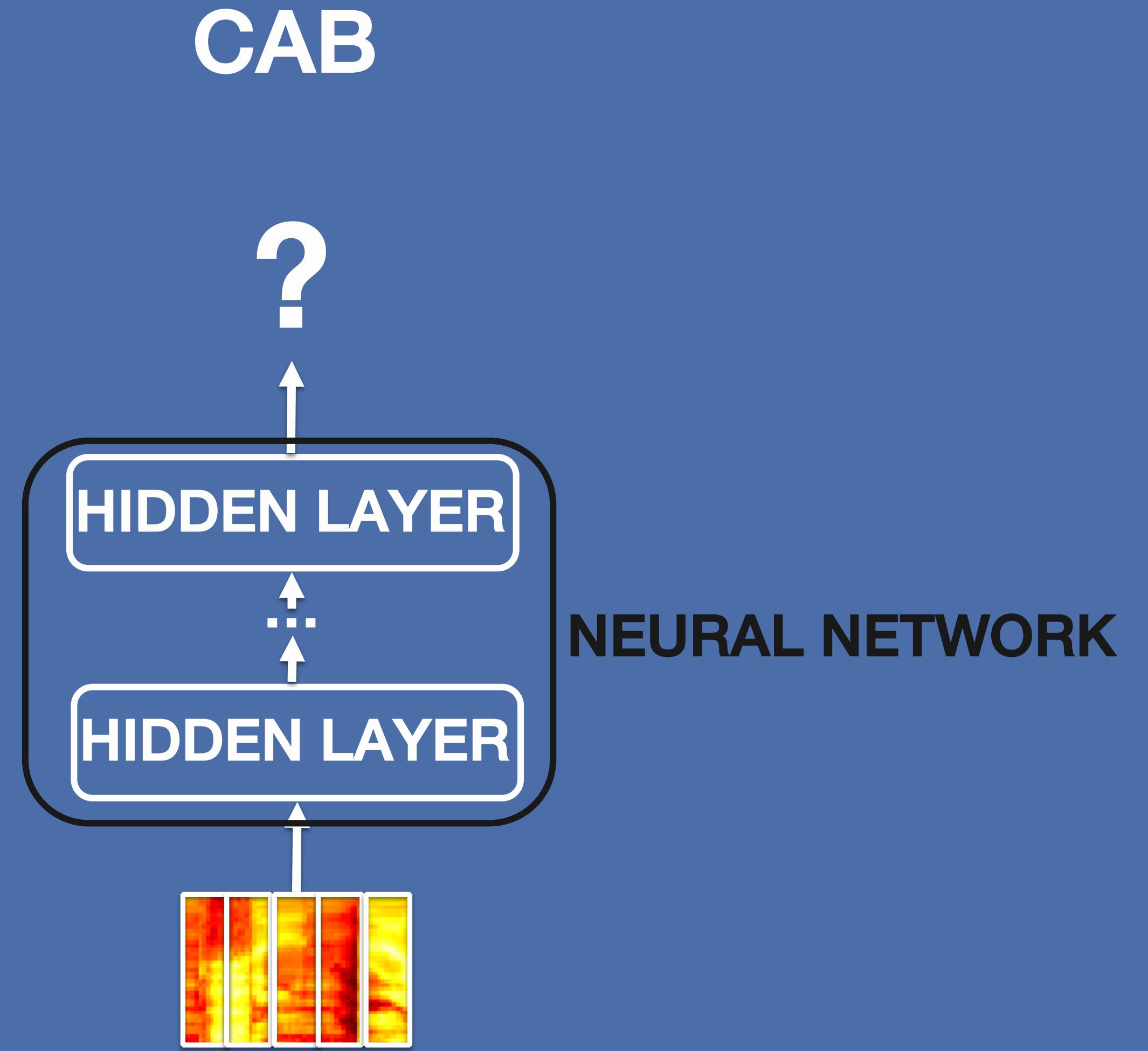
- When a state is attributed to each frame it is trivial to define the loss function: classification loss

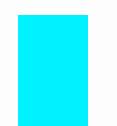




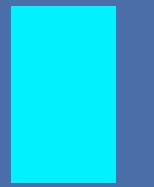
## Training a DNN without states?

- When a state is attributed to each frame it is trivial to define the loss function
- Without alignment, we have the word transcription and the feature frames
- Train a speech recognition system directly from the features to the transcription (no phonetic transcription, no pronunciation dictionary, no alignment)



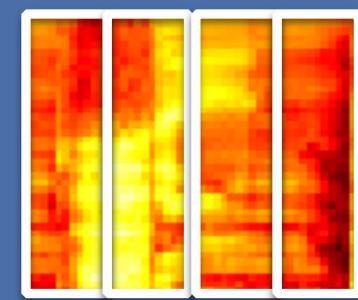


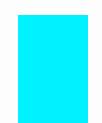
## Problem: no alignment available



- Phonemes last dozens to hundreds of milliseconds, while frames are sampled every 10ms
- Many more frames than phonemes/letters
- We need to learn **classification** and **alignment** jointly
- A loss function allows learning both at the same time: Connectionist Temporal Classification (CTC)

CAB





## Connectionist Temporal Classification

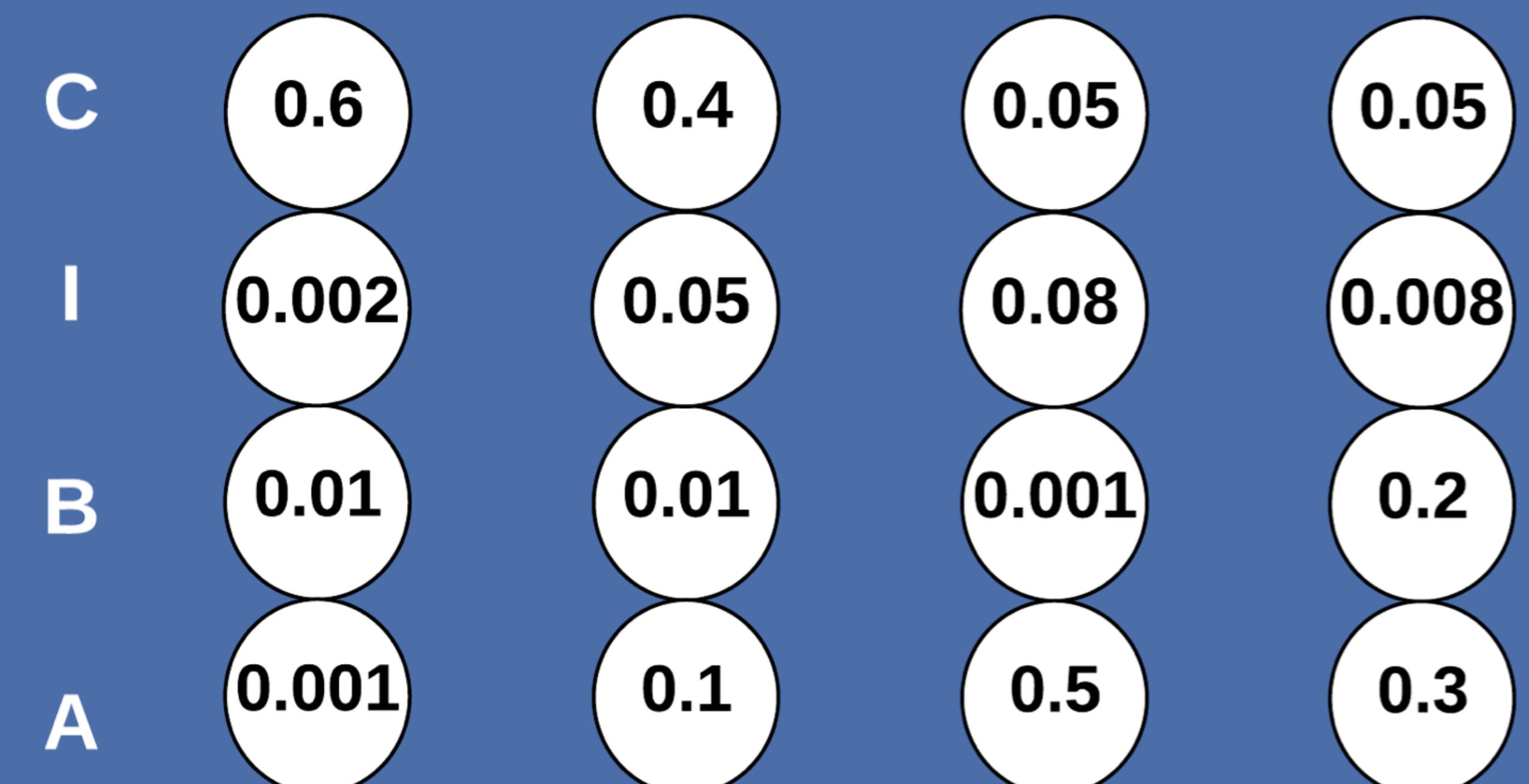


- Given that the final layer of your network is a softmax with as many dimensions as there are characters, it can be seen as a probability distribution over characters

$y_k^t$  the probability of character  $k$  at time  $t$

$$y_k^t = \text{Softmax}(h_{N-1}(x_t))$$

$$= \frac{e^{w_k^T h_{N-1}(x_t) + b}}{\sum_{k=1}^K e^{w_k^T h_{N-1}(x_t)}}$$



« Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks », Graves et al.



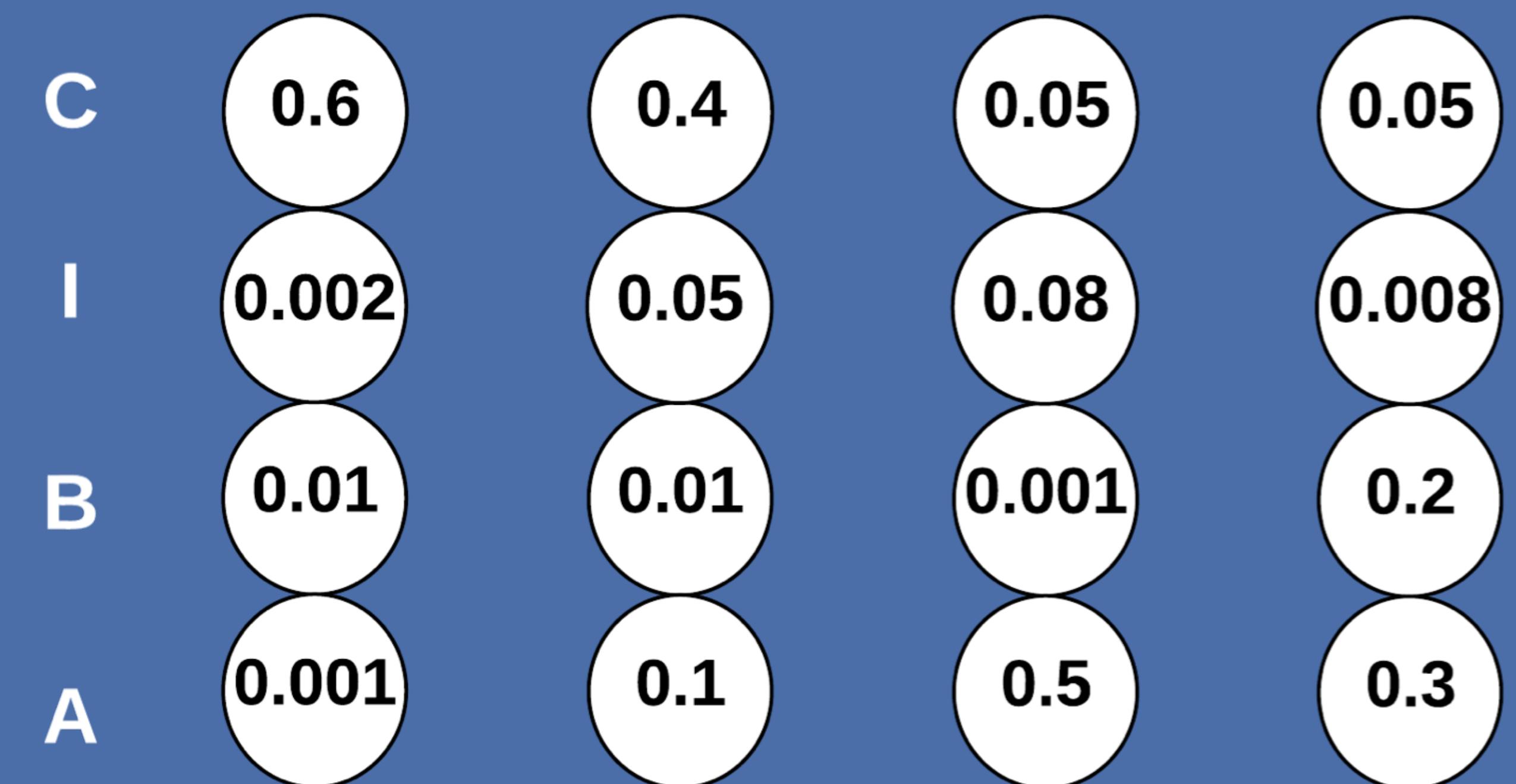
## Connectionist Temporal Classification

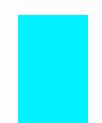


- Given that the final layer of your network is a softmax with as many dimensions as there are characters, it can be seen as a probability distribution over characters

$y_k^t$  the probability of character  $k$  at time  $t$

$Y = \{A, B, C, \dots, Z, \text{apostrophe}, \text{space}, \text{blank}\}$

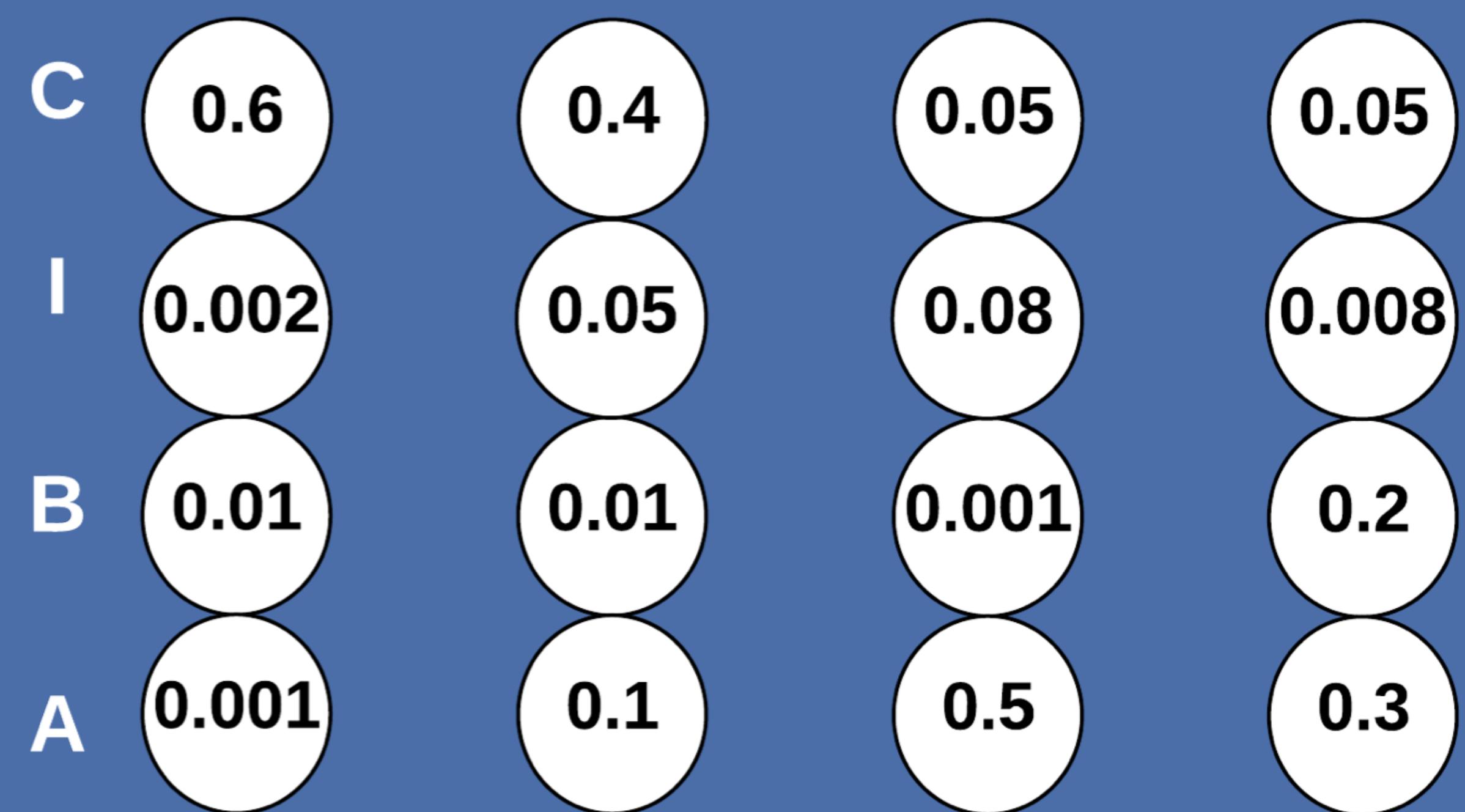




## Connectionist Temporal Classification



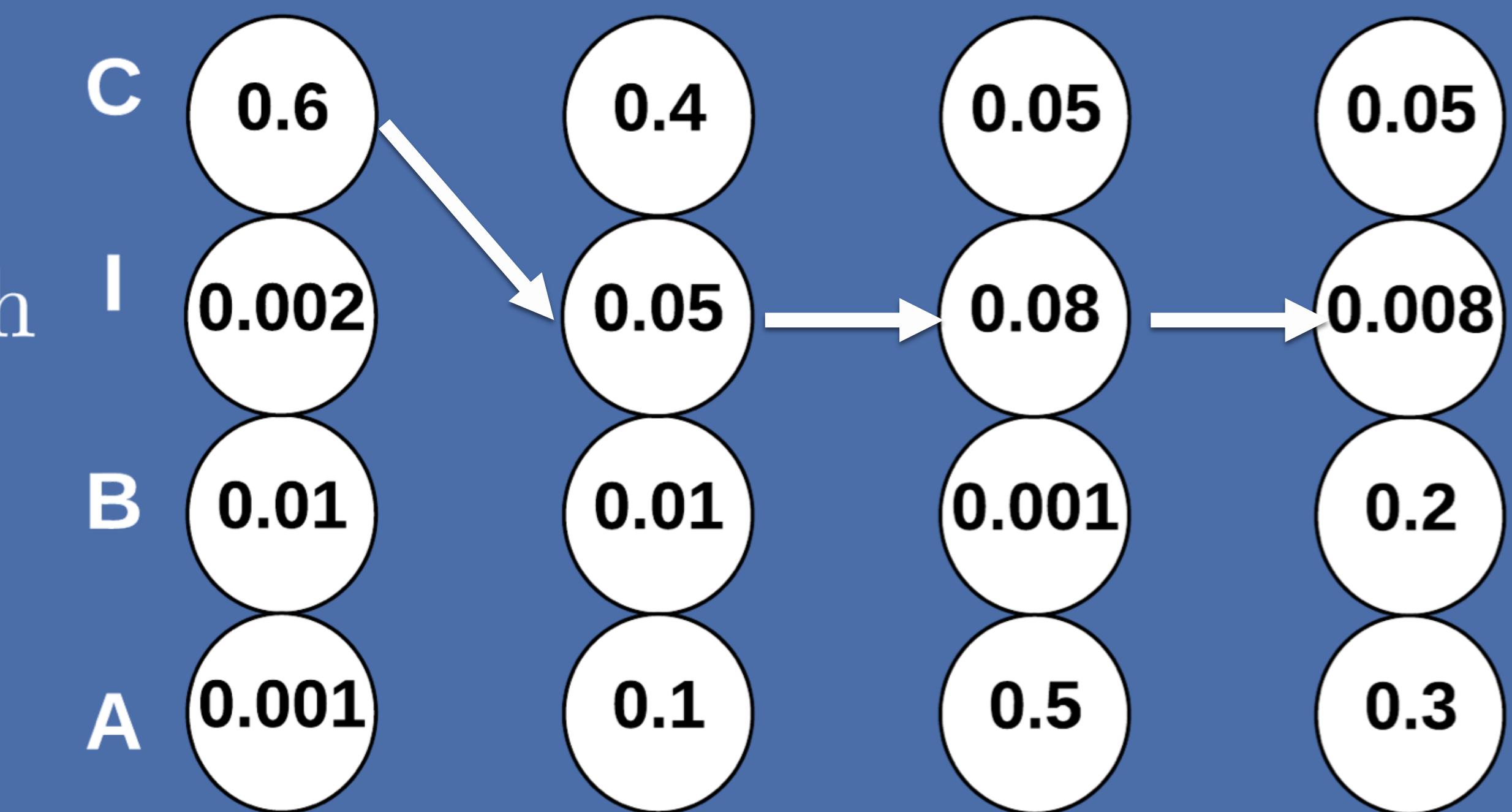
- Each output dimension is a node
- A path is a sequence of nodes
- Probability of a path is the product of the probability of the nodes (independence of time steps)



End-to-end training



## Connectionist Temporal Classification



$\pi = \pi_1, \dots, \pi_T$  a path in the graph

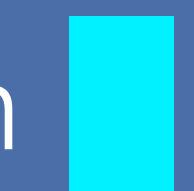
$$p(\pi|x) = \prod_{t=1}^T y_{\pi_t}^t, \forall \pi \in L^T$$

$$\text{Probability(CIII)} = 0.6 * 0.05 * 0.08 * 0.008 = 0.00000192$$

End-to-end training

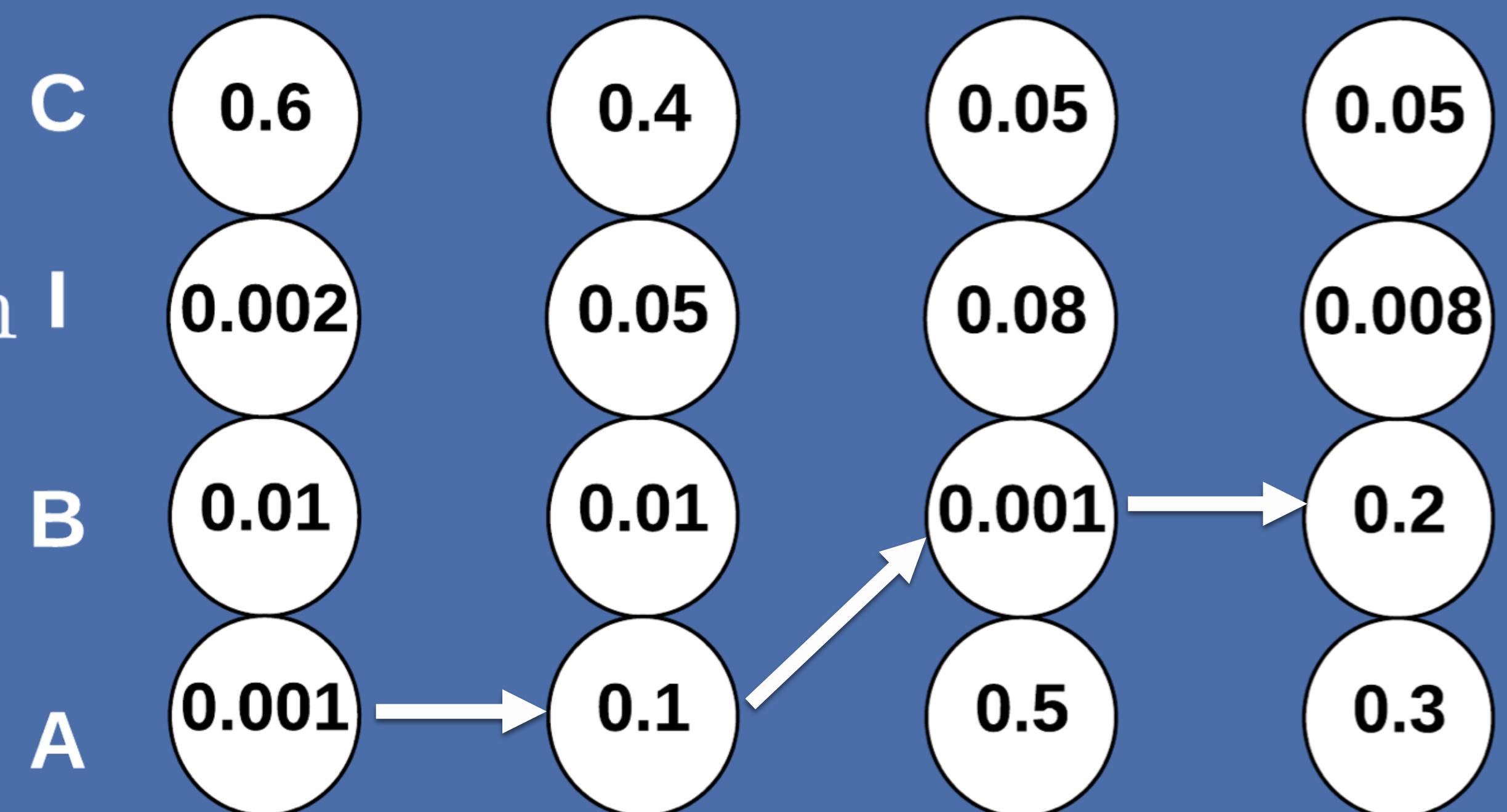


## Connectionist Temporal Classification

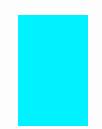


$\pi = \pi_1, \dots, \pi_T$  a path in the graph I

$$p(\pi|x) = \prod_{t=1}^T y_{\pi_t}^t, \forall \pi \in L^T$$



$$\text{Probability(AABB)} = 0.001 * 0.1 * 0.001 * 0.2 = 0.00000002$$



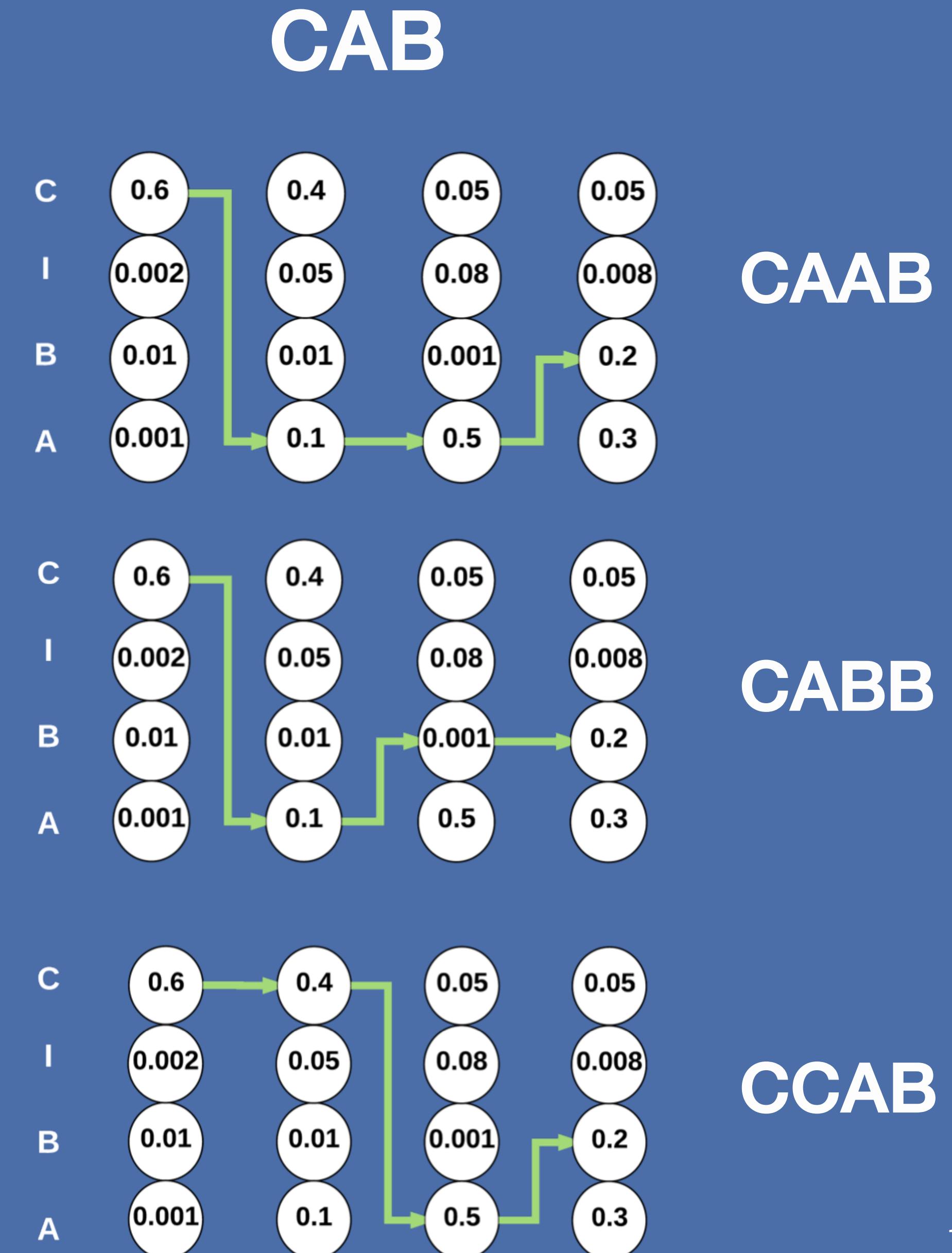
Training criterion



- Loss function:

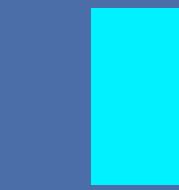
$$- \sum_{\pi \in \text{Valid paths}} P(\pi|x)$$

- Maximize the valid paths
- Nothing for the other paths
- Train the entire network with backpropagation

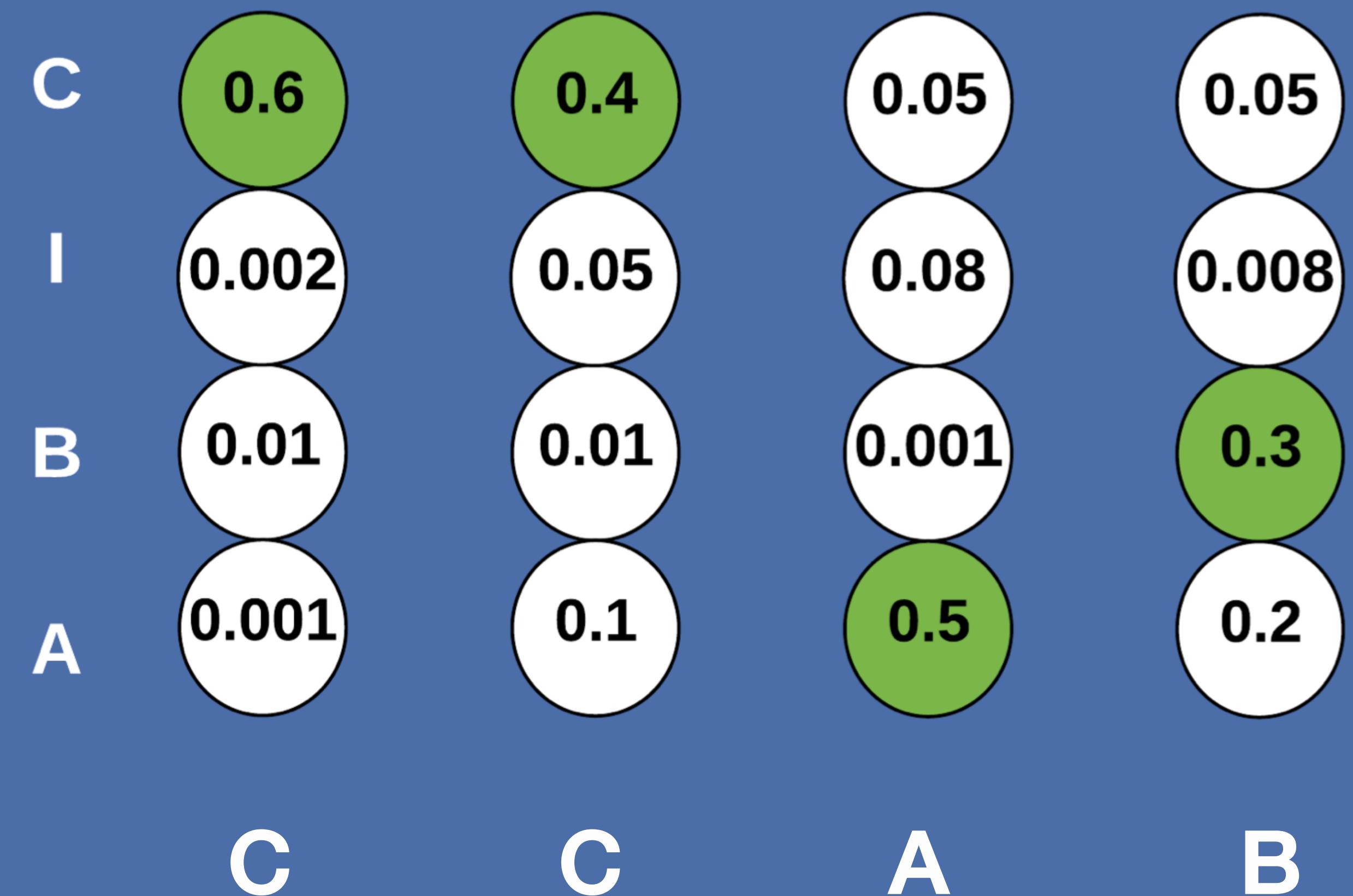




## Decoding



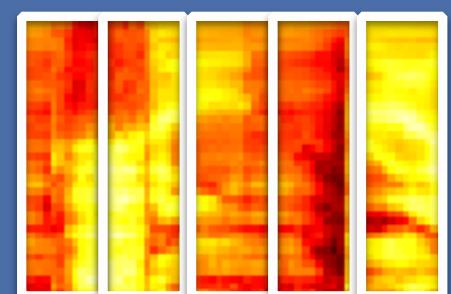
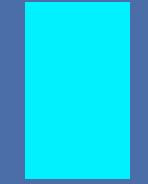
- The basic decoding is just to take the most likely character at each step
- It can incorporate a language model by adding the probability of words



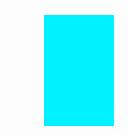
End-to-end training



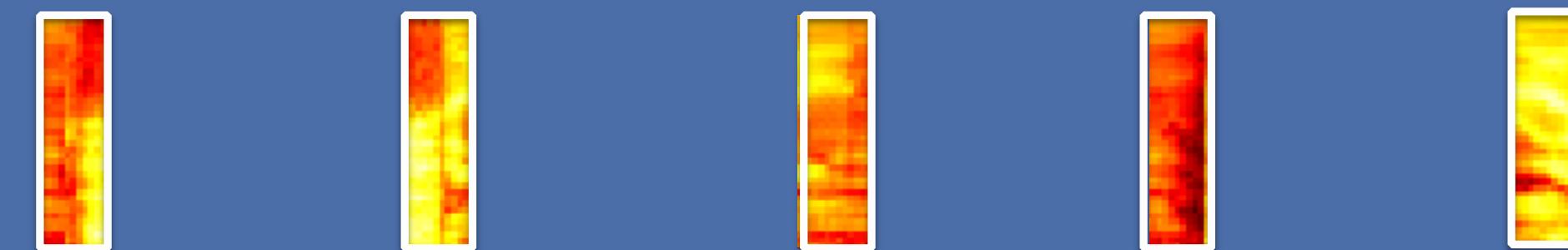
Recurrent acoustic model



End-to-end training



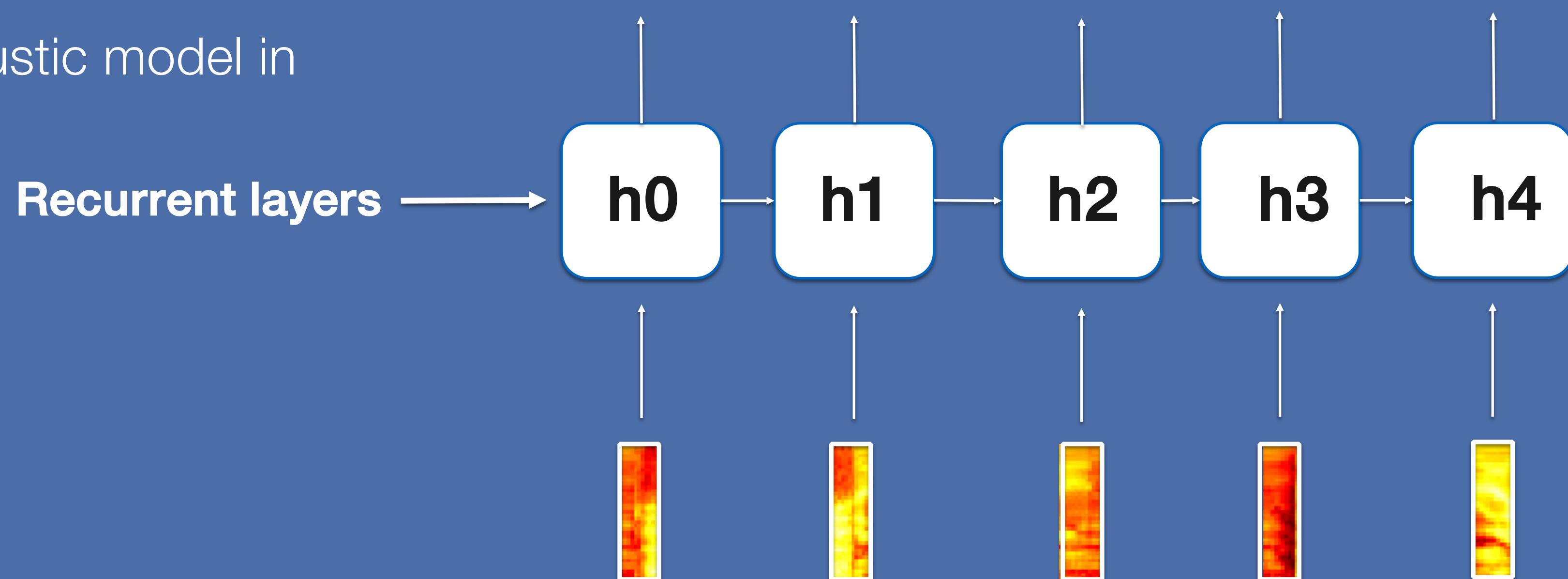
Recurrent acoustic model





## Recurrent acoustic model

- Recurrent Neural Network used to output the distributions step by step
- Exploits the dependencies along the sequence (compensates for CTC)
- Most standard acoustic model in the past years



End-to-end training



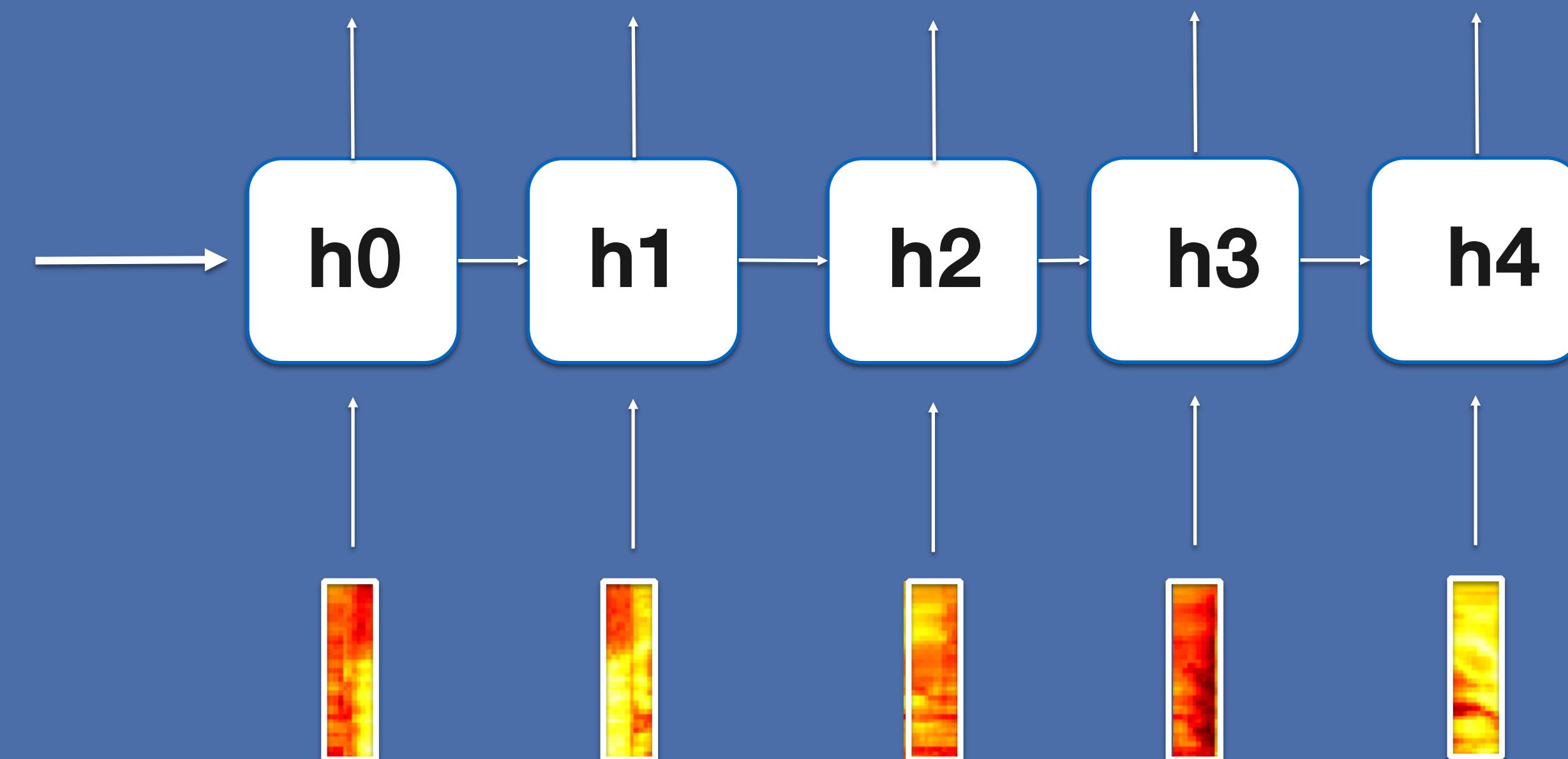
Recurrent acoustic model



$$h_0 = \tanh(W_x x_0)$$

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h) \text{ for } t > 0$$

$$y^t = \text{Softmax}(W_{hy} h_t + b_y)$$



End-to-end training



Recurrent acoustic model



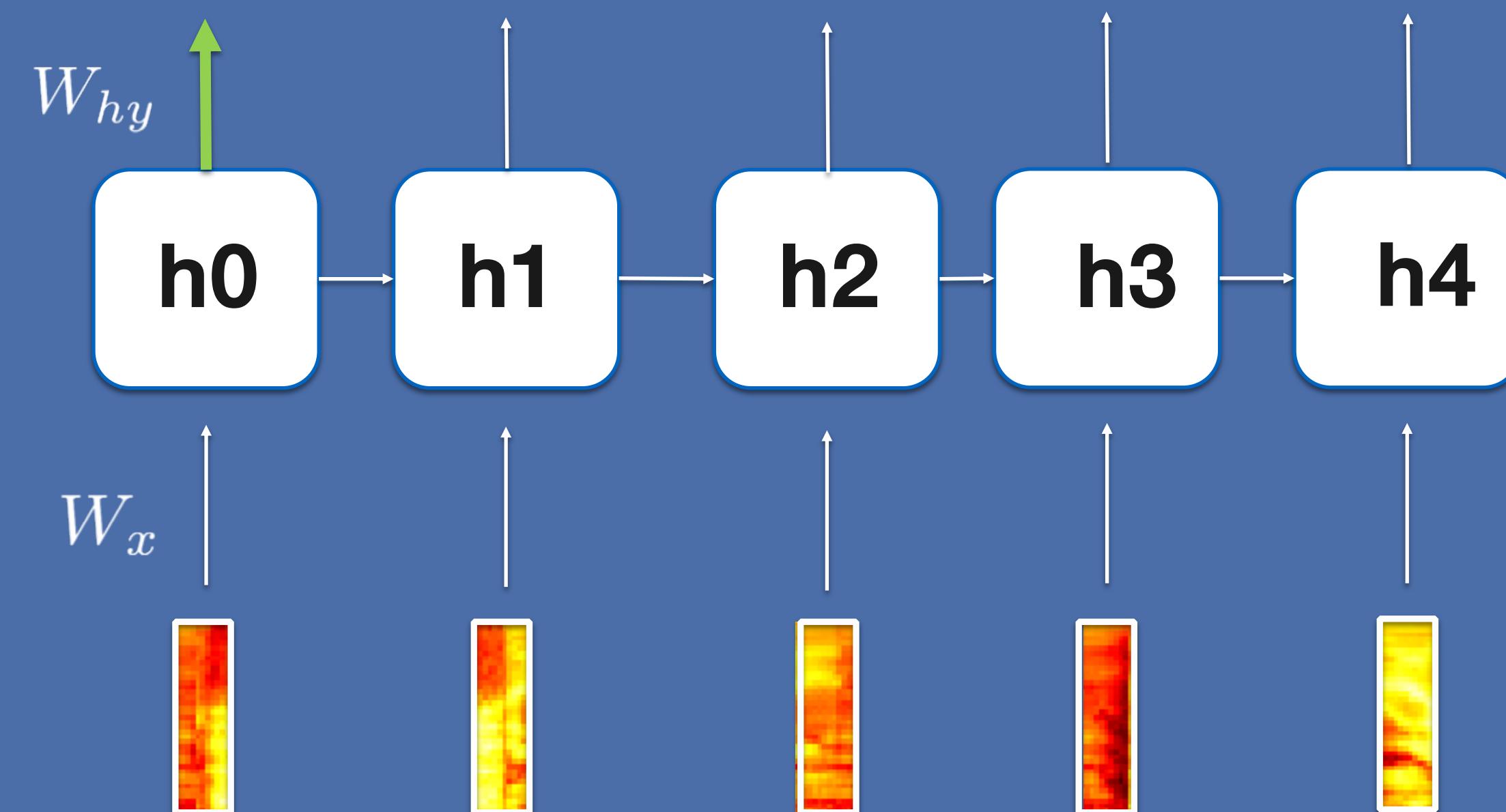
$$h_0 = \tanh(W_x x_0)$$

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h) \text{ for } t > 0$$

$$y^t = \text{Softmax}(W_{hy} h_t + b_y)$$

$$P(y^0|x_0)$$

A	0.6
B	0.002
C	0.01
D	0.001
...	
Z	0.04



End-to-end training



Recurrent acoustic model



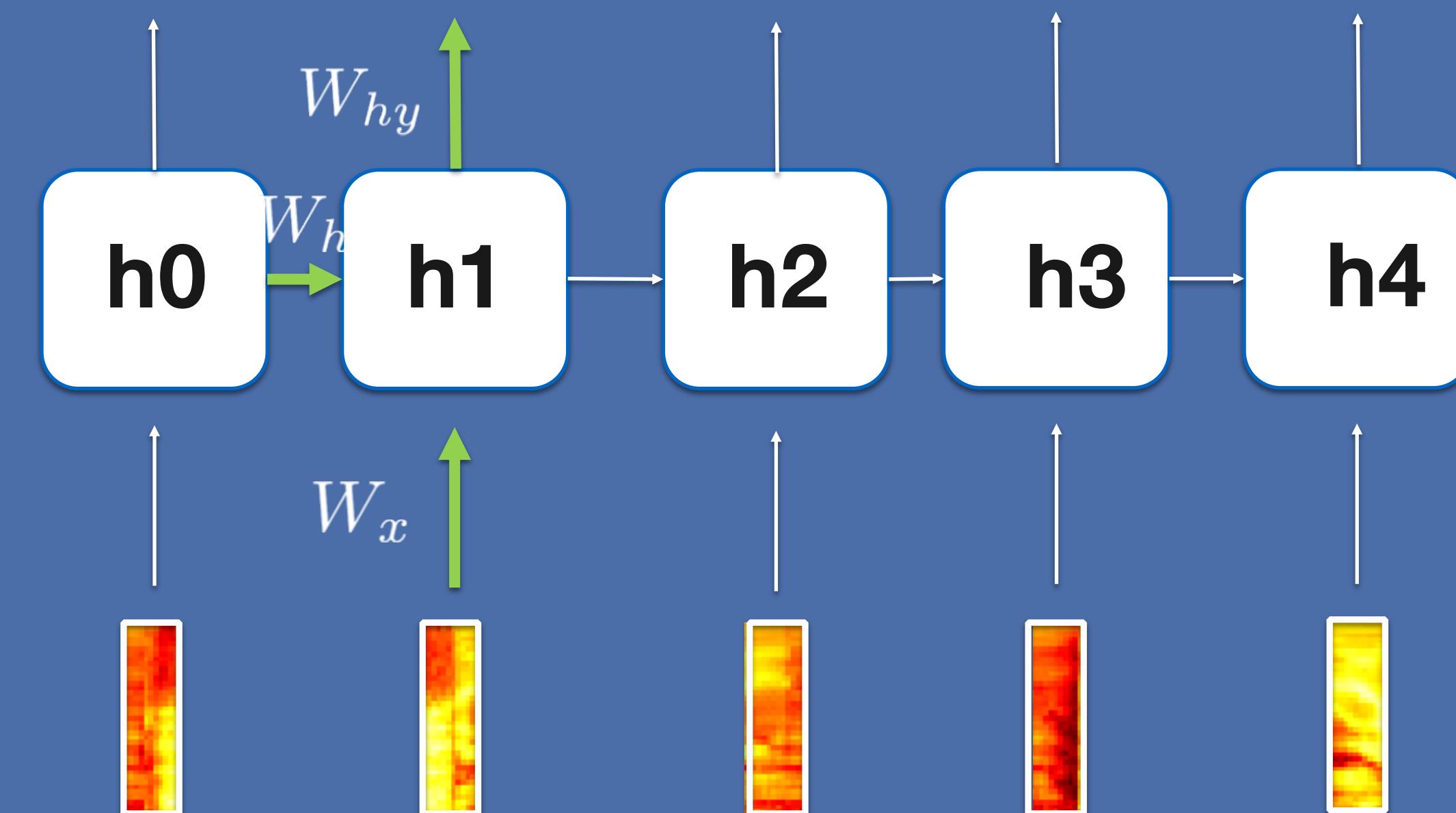
$$h_0 = \tanh(W_x x_0)$$

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h) \text{ for } t > 0$$

$$y^t = \text{Softmax}(W_{hy} h_t + b_y)$$

$$P(y^1|x_1, h_0)$$

A	0.6
B	0.002
C	0.01
D	0.001
...	
Z	0.04



End-to-end training



Recurrent acoustic model



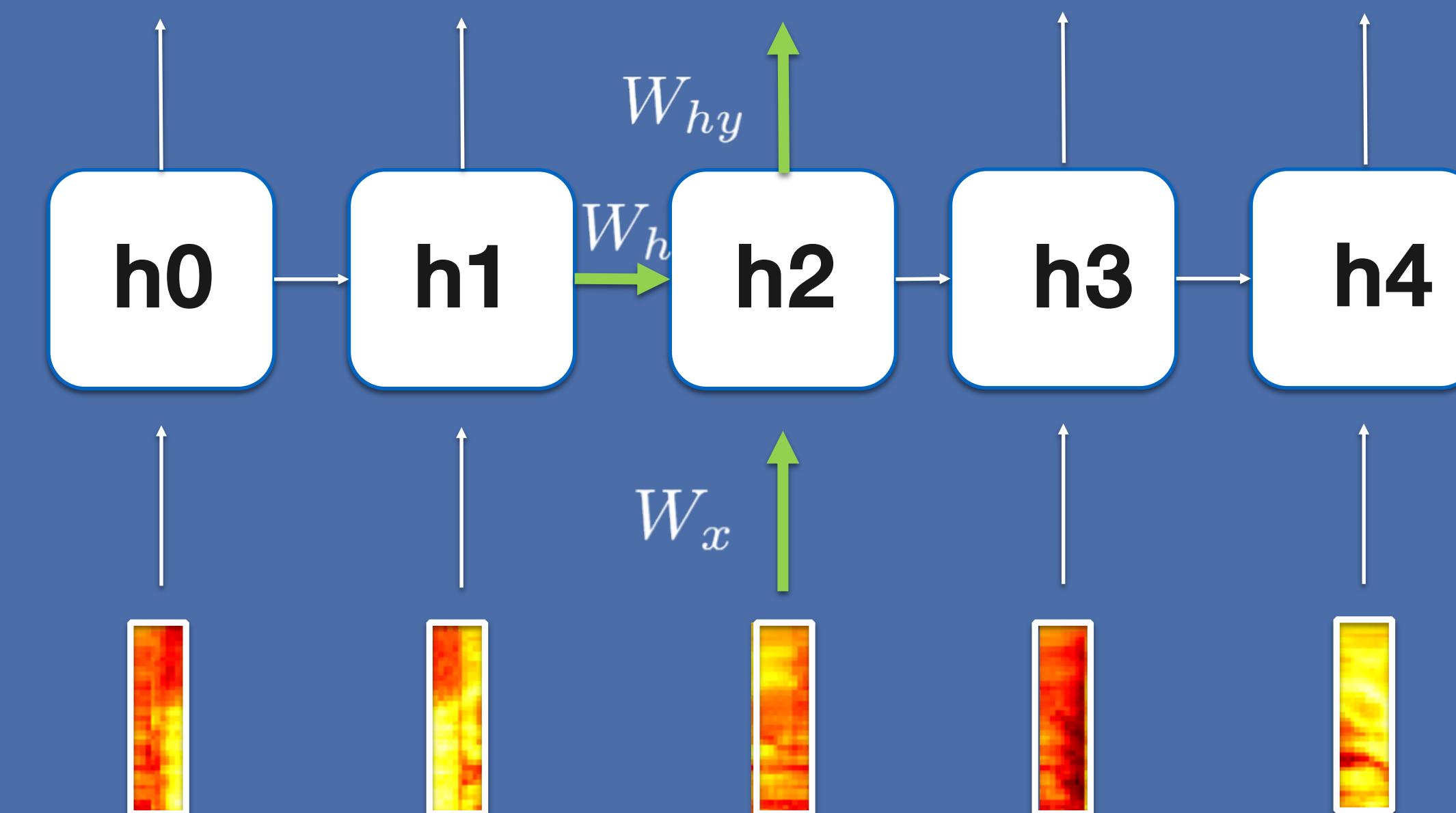
$$h_0 = \tanh(W_x x_0)$$

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h) \text{ for } t > 0$$

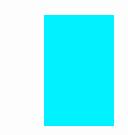
$$y^t = \text{Softmax}(W_{hy} h_t + b_y)$$

$$P(y^2|x_2, h_1)$$

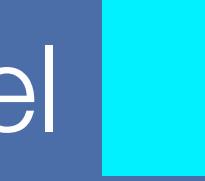
A	0.6
B	0.002
C	0.01
D	0.001
...	
Z	0.04



End-to-end training



Recurrent acoustic model



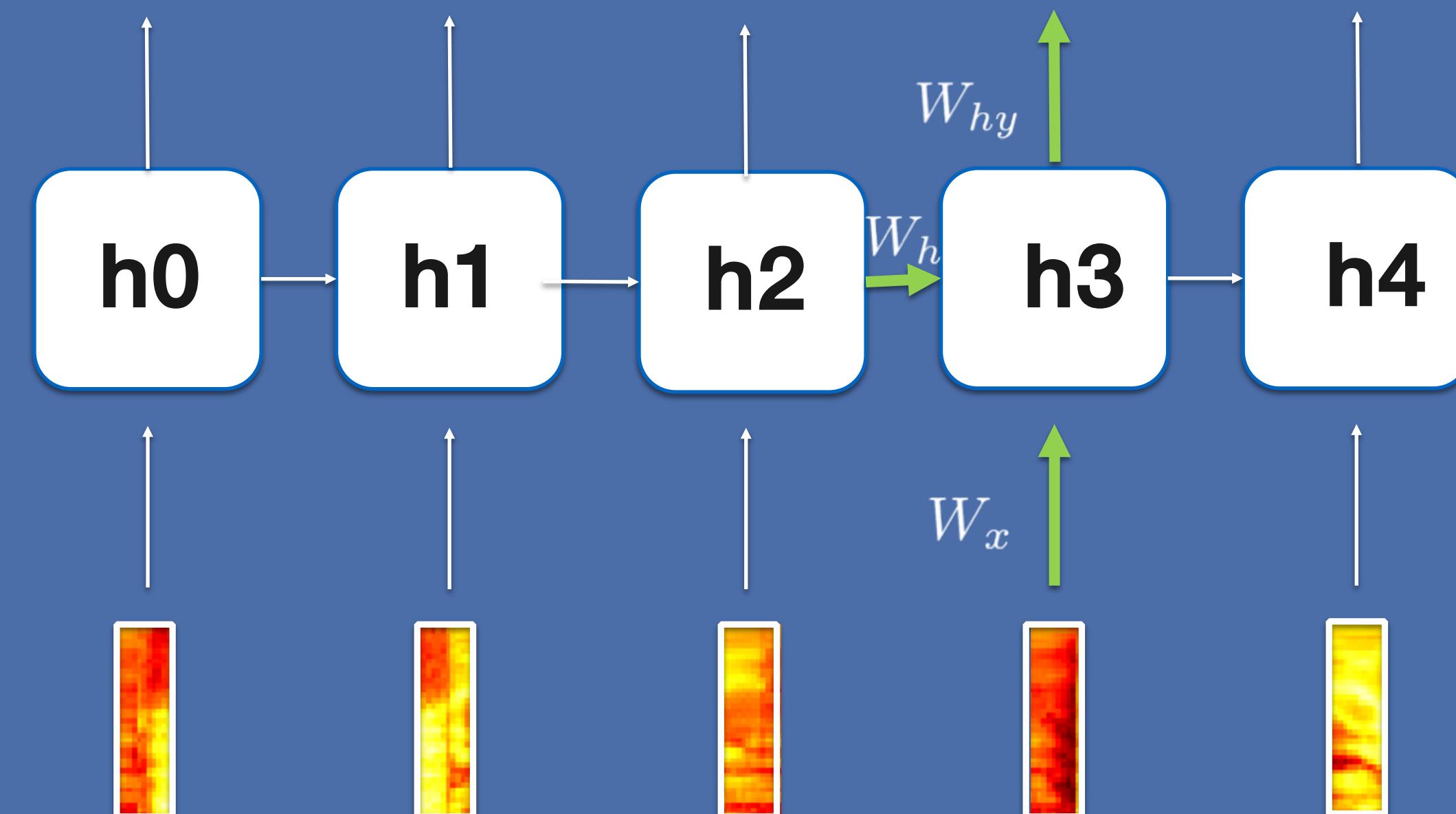
$$h_0 = \tanh(W_x x_0)$$

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h) \text{ for } t > 0$$

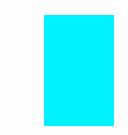
$$y^t = \text{Softmax}(W_{hy} h_t + b_y)$$

$$P(y^3|x_3, h_2)$$

A	0.6
B	0.002
C	0.01
D	0.001
...	
Z	0.04



End-to-end training



Recurrent acoustic model



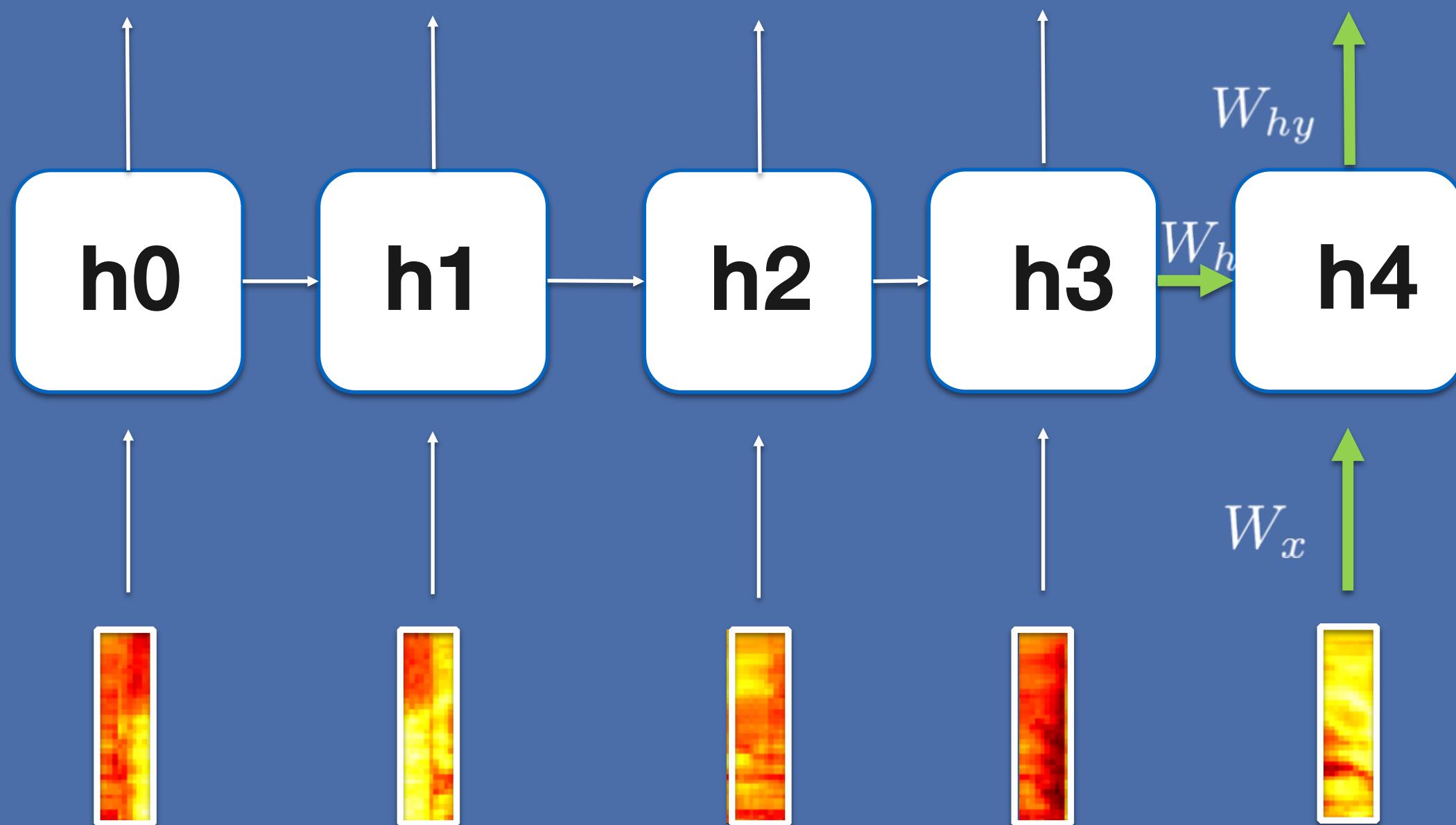
$$h_0 = \tanh(W_x x_0)$$

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h) \text{ for } t > 0$$

$$y^t = \text{Softmax}(W_{hy} h_t + b_y)$$

$$P(y^4|x_4, h_3)$$

A	0.6
B	0.002
C	0.01
D	0.001
...	
Z	0.04



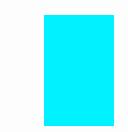
## Question

Recurrent Neural Networks are good acoustic models as they allow modelling a sequence. However there is one aspect of speech they cannot model, which one is it?

1. Tonal Languages
2. Co-articulation
3. Speaker variations

<https://api.socrative.com/rc/KBcrE4>

End-to-end training



Bi-directional RNN



$$\vec{h}_0 = \tanh(W_x x_0)$$

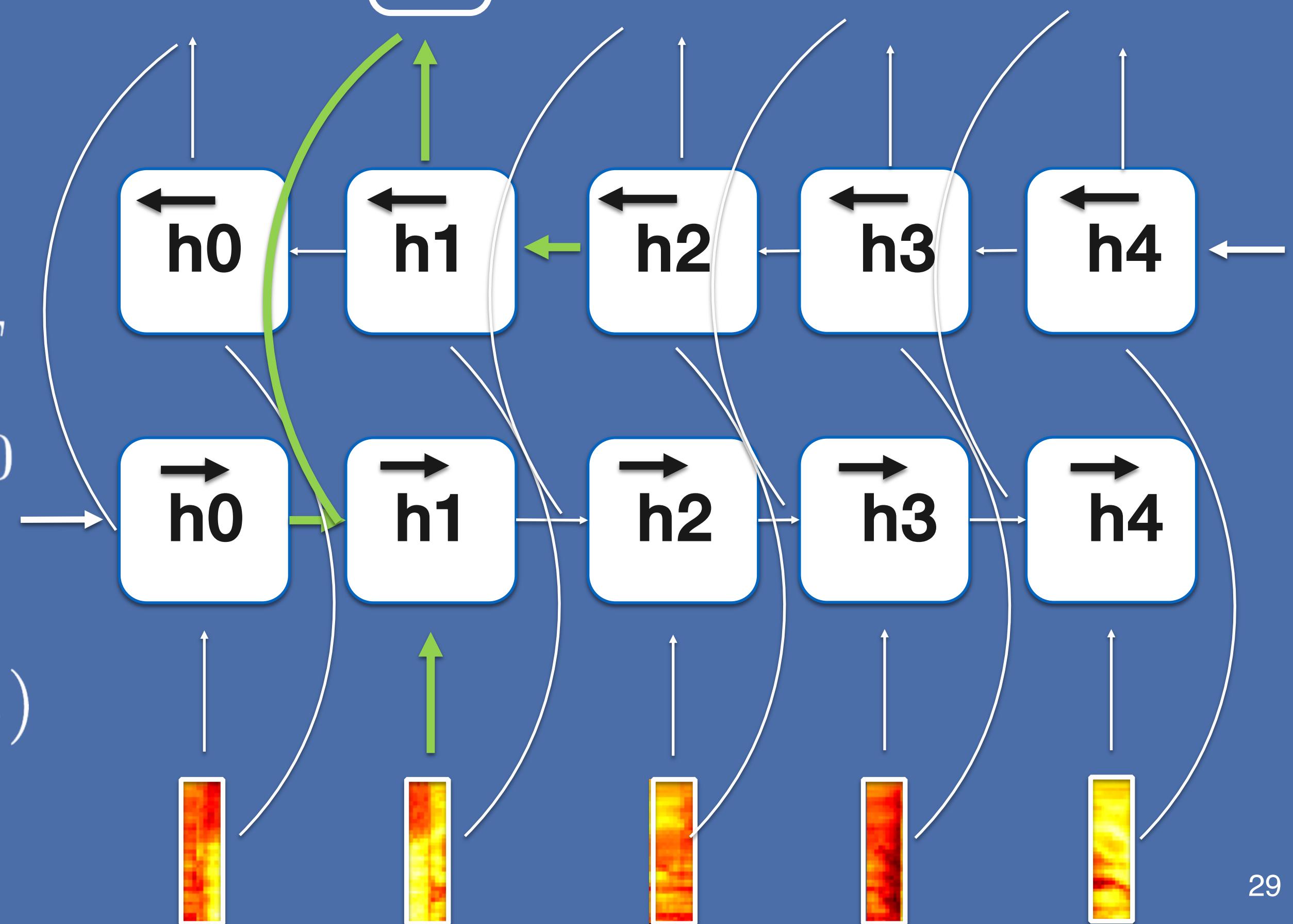
$$\leftarrow h_T = \tanh(W_x x_T)$$

$$\leftarrow h_t = \tanh(W_x x_t + W_h^\leftarrow \leftarrow h_{t-1} + b_h^\leftarrow) \text{ for } t < T$$

$$\vec{h}_t = \tanh(W_x x_t + W_h^\rightarrow \vec{h}_{t-1} + b_h^\rightarrow) \text{ for } t > 0$$

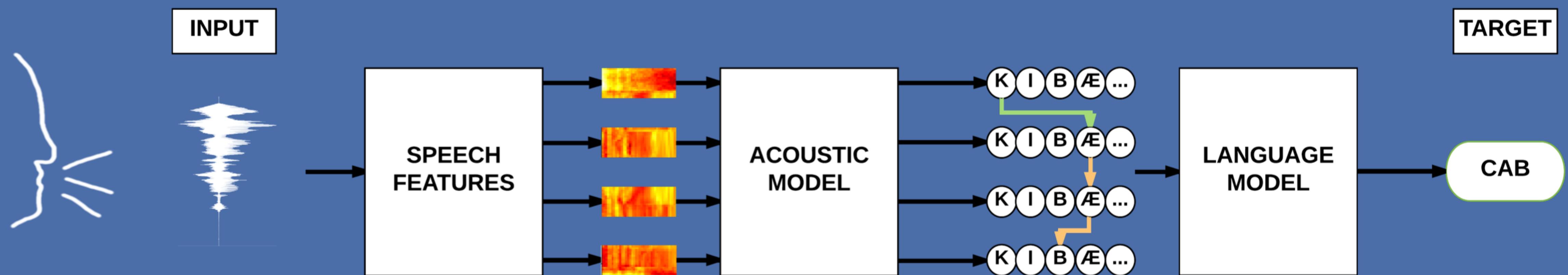
$$y^t = \text{Softmax}(W_{\vec{h} y} \vec{h}_t + W_{\leftarrow h y} \leftarrow h_t + b_y)$$

A	0.6
B	0.002
C	0.01
D	0.001
...	
Z	0.04



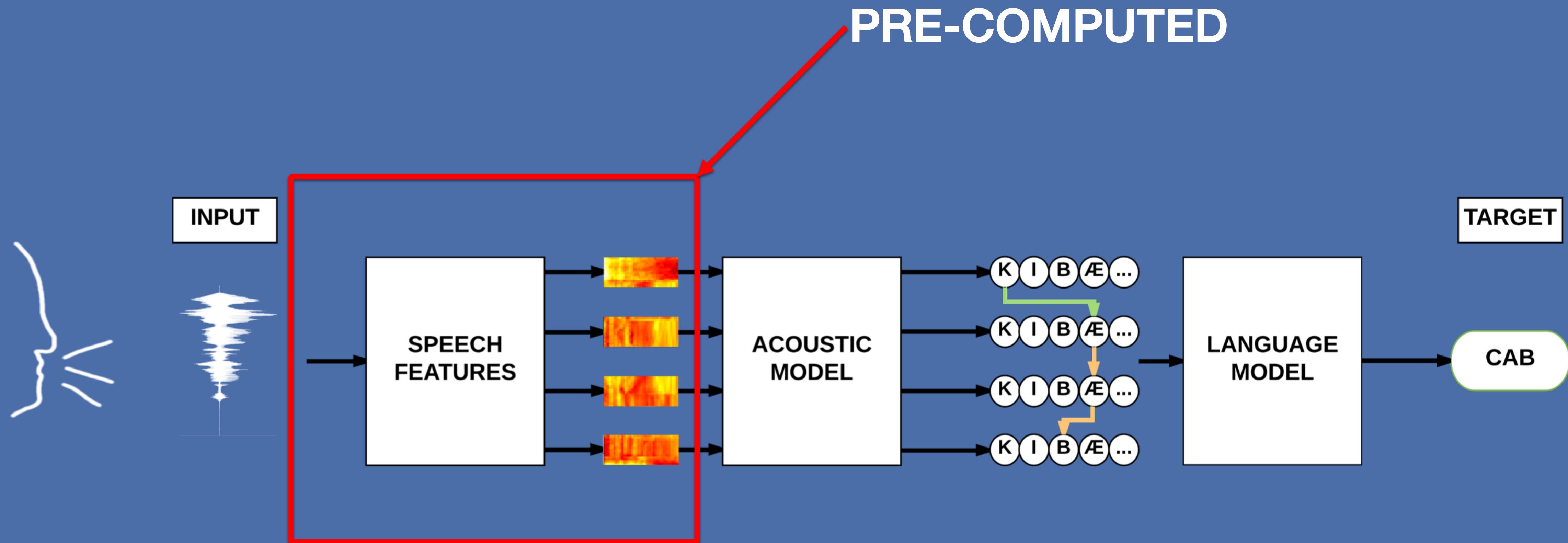
End-to-end training

Is this « end-to-end » learning?



End-to-end training

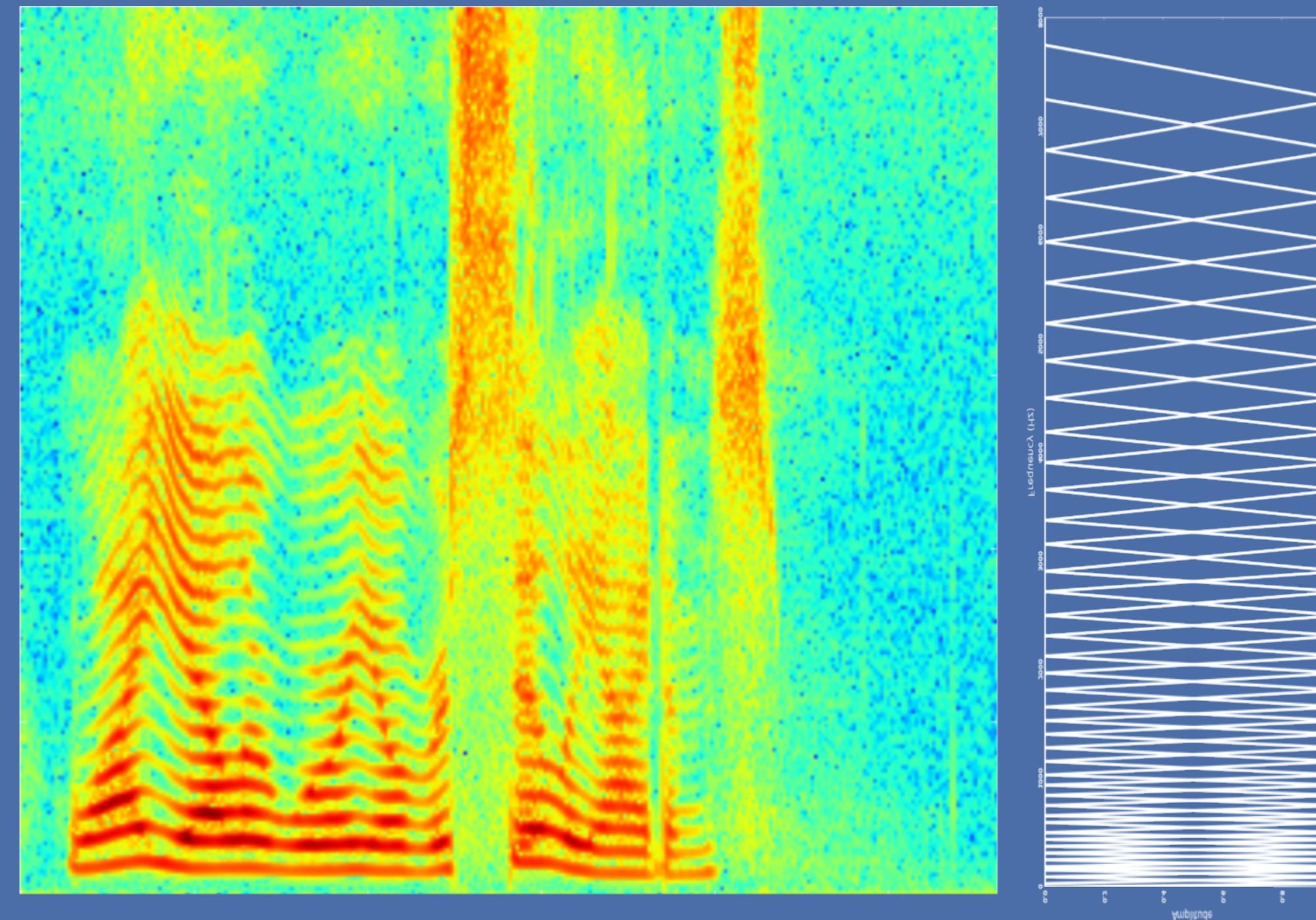
Is this « end-to-end » learning?





Can we learn the mel-filterbanks?

$$Melfbank_j(k) = \sum_{\omega=0}^{256} \text{Spectrogram}(k, \omega) Melfilter_j(\omega)$$



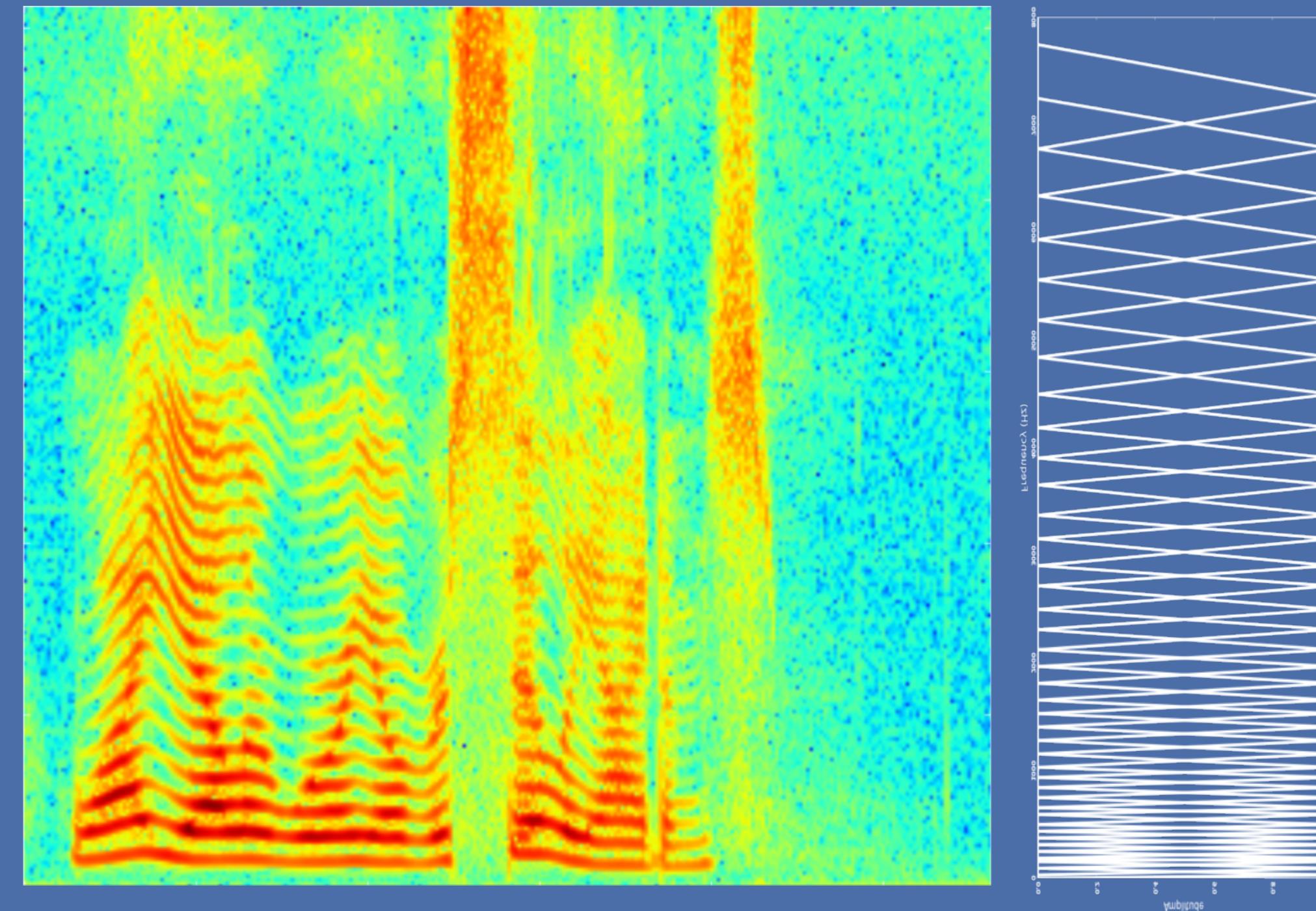


## Can we learn the mel-filterbanks?



- This can be computed by a convolution over the frequency axis

« Learning Filter Banks  
Within a Deep Neural  
Network Framework»,  
Sainath et al.

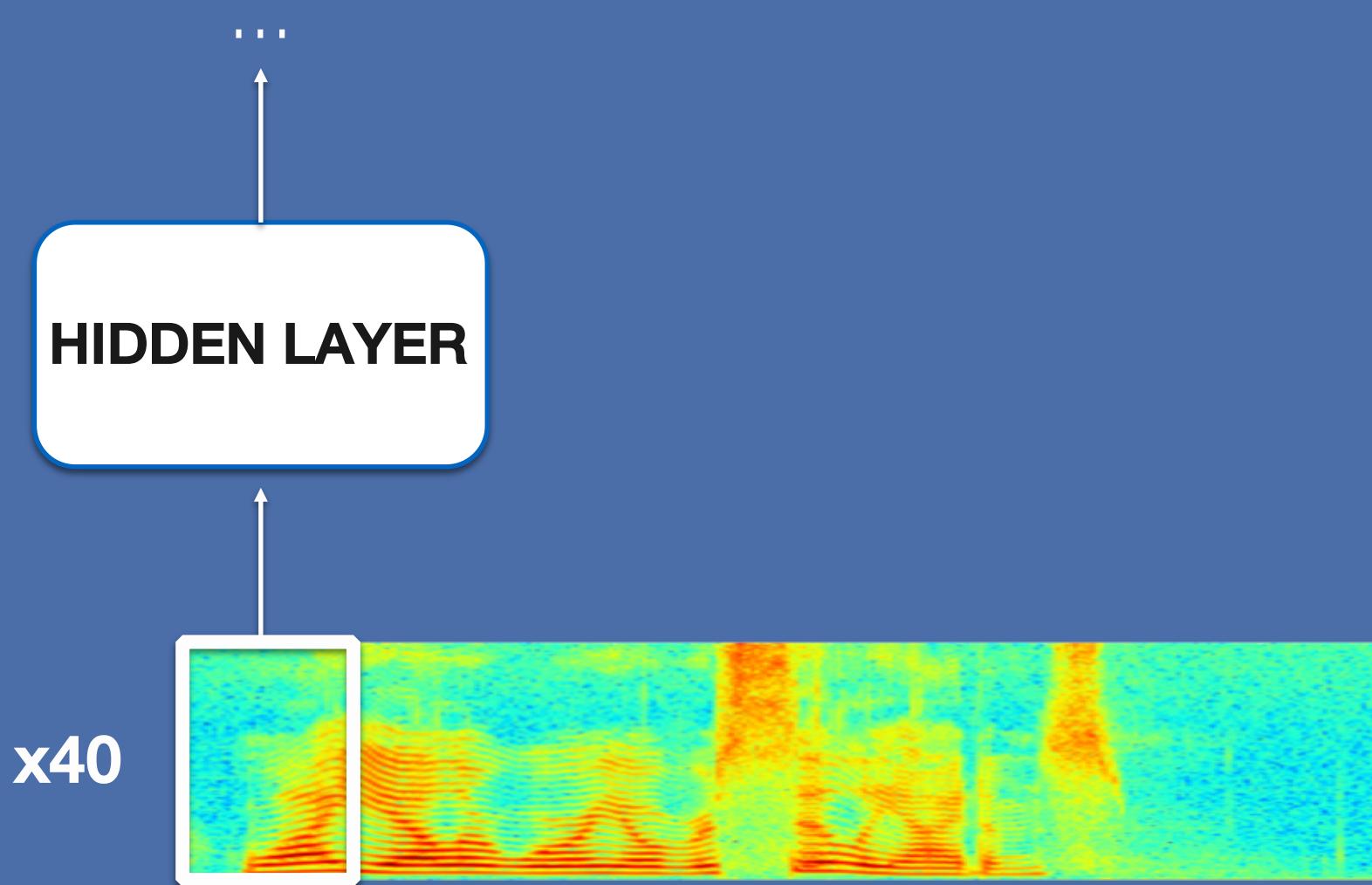




Can we learn the mel-filterbanks?



- This can be computed by a convolution over the frequency axis

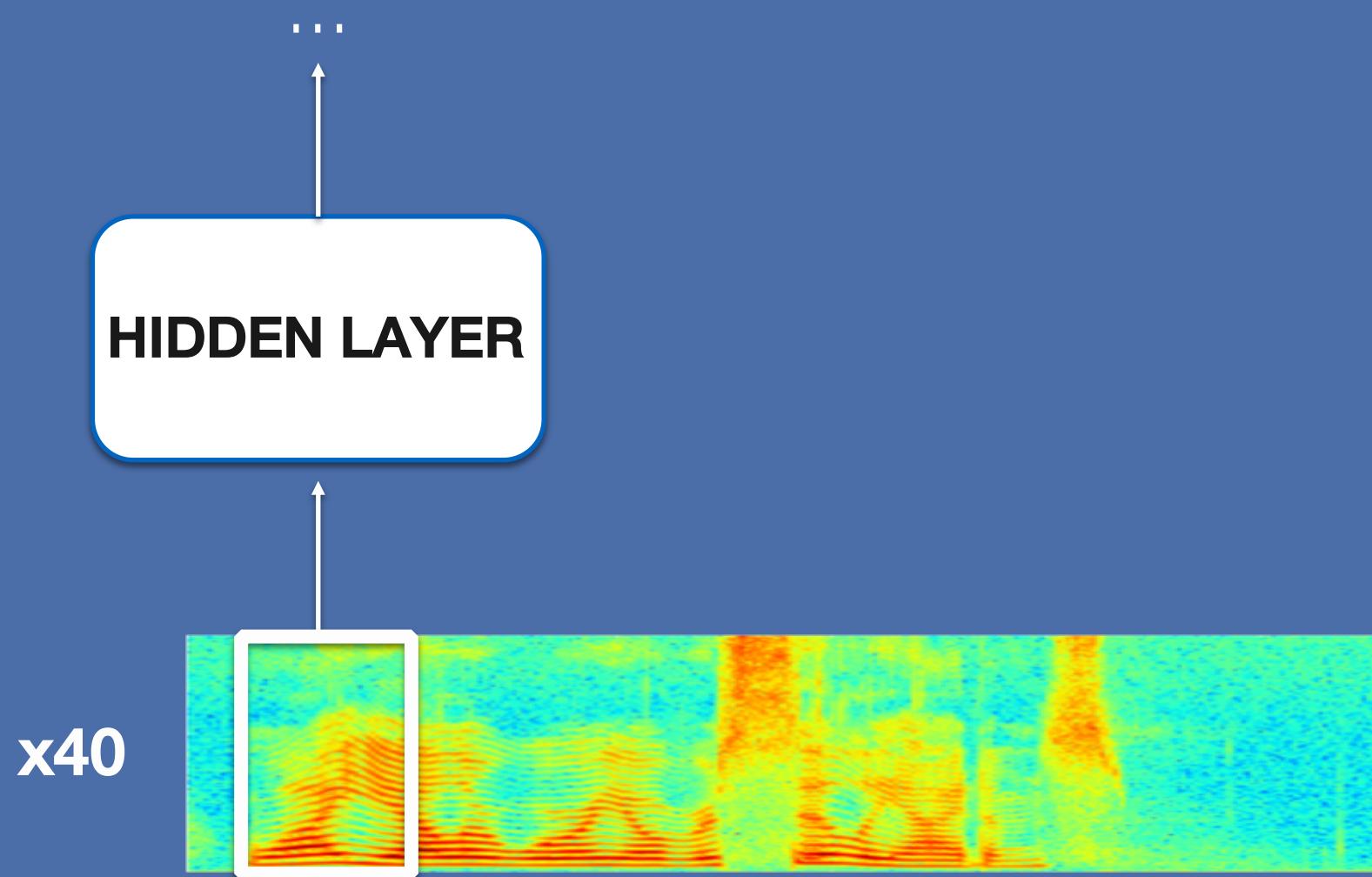




## Can we learn the mel-filterbanks?

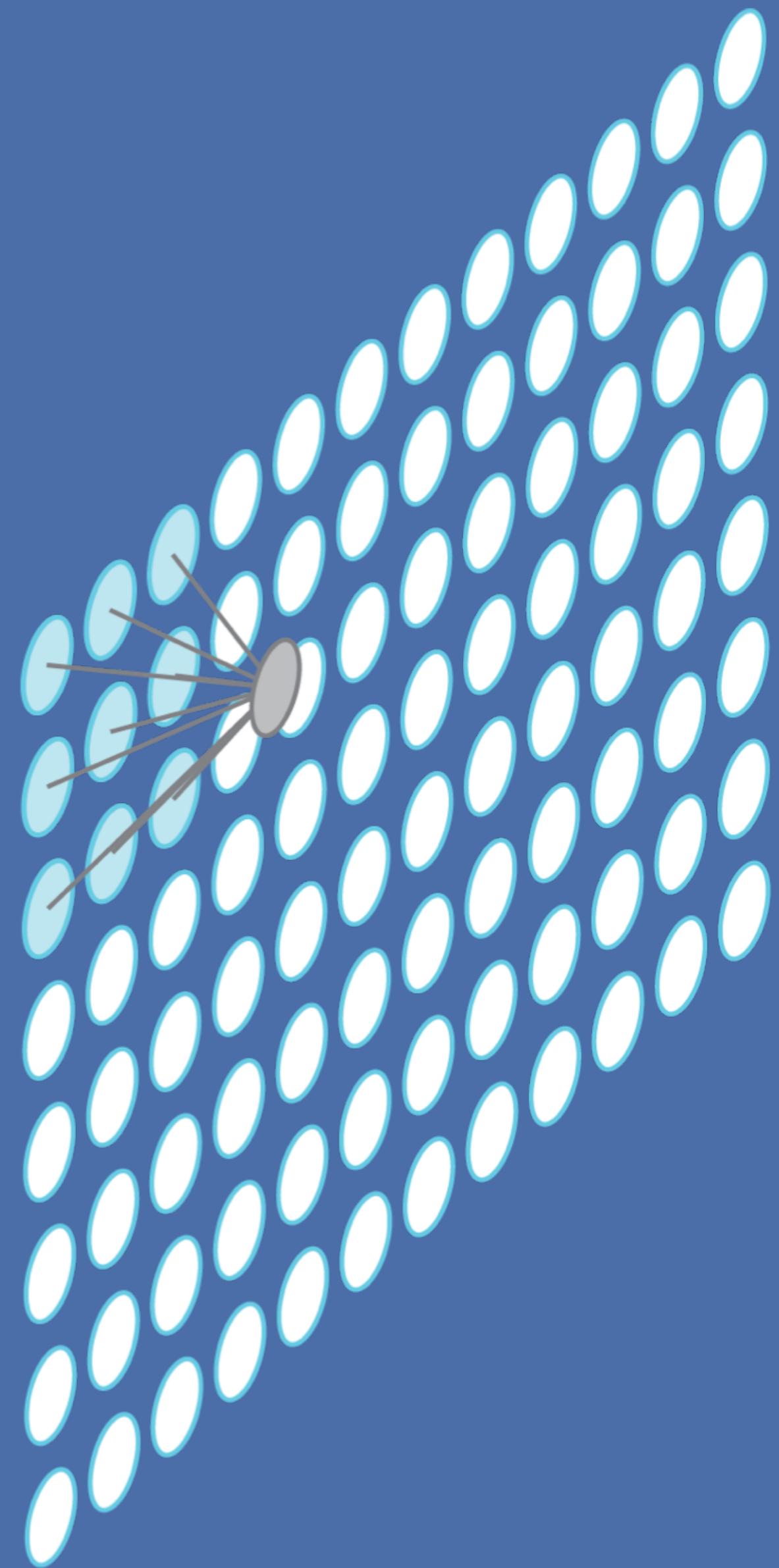


- This can be computed by a convolution over the frequency axis



End-to-end training

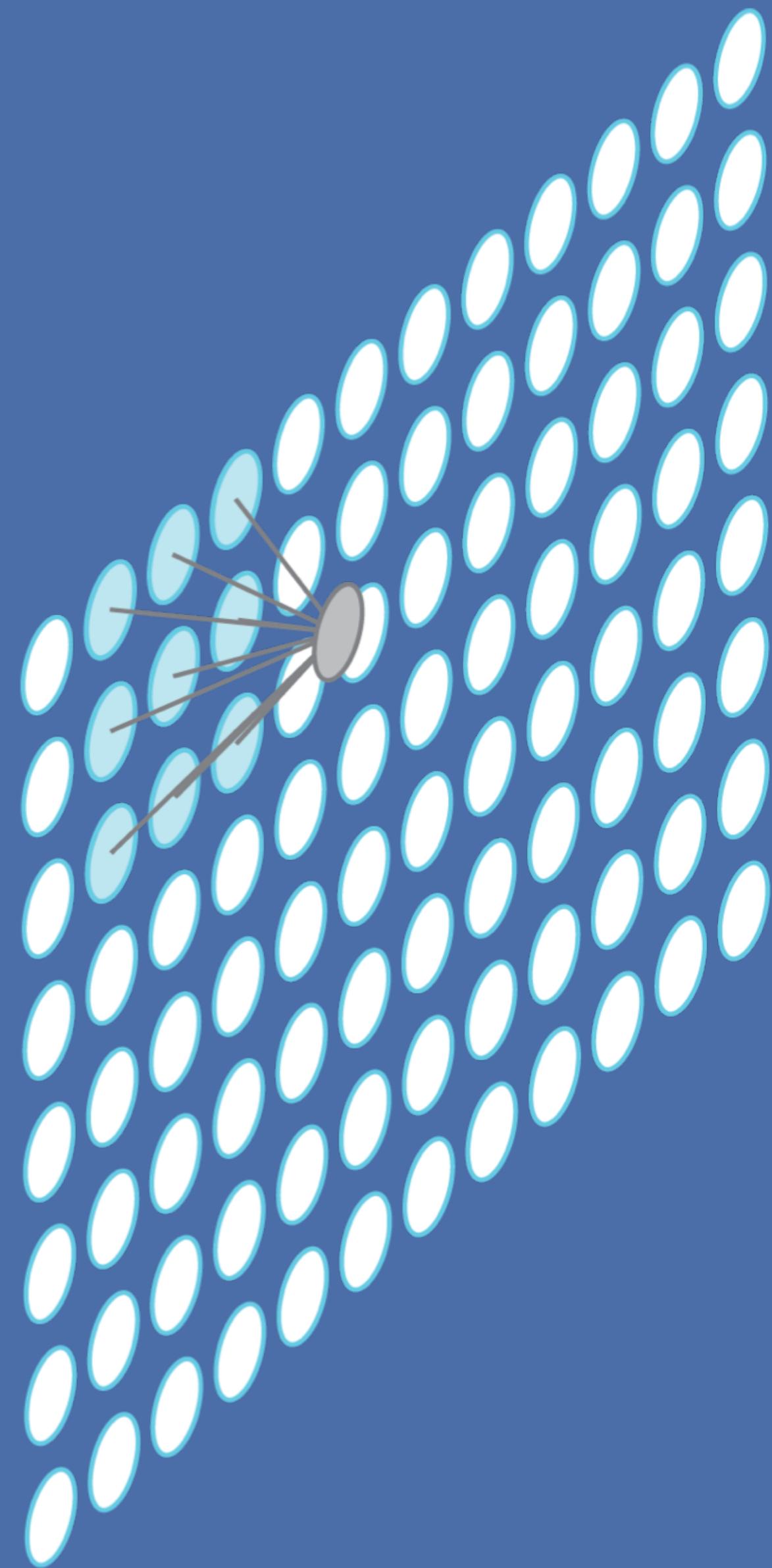
## 2-D Convolutions



Kernel size :  $1 \times 3 \times 3$   
Number of kernels : 1

End-to-end training

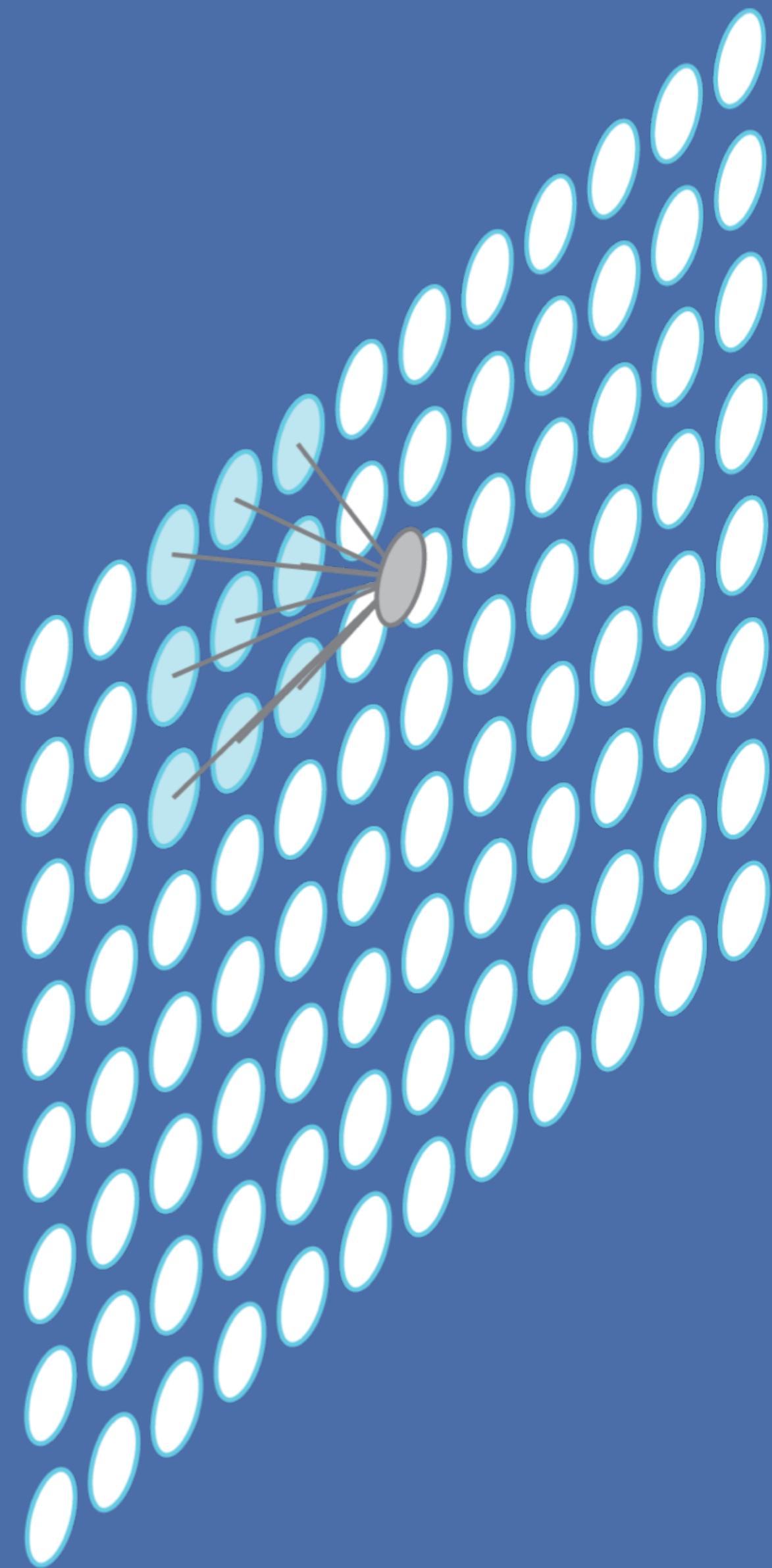
## 2-D Convolutions



Kernel size :  $1 \times 3 \times 3$   
Number of kernels : 1

End-to-end training

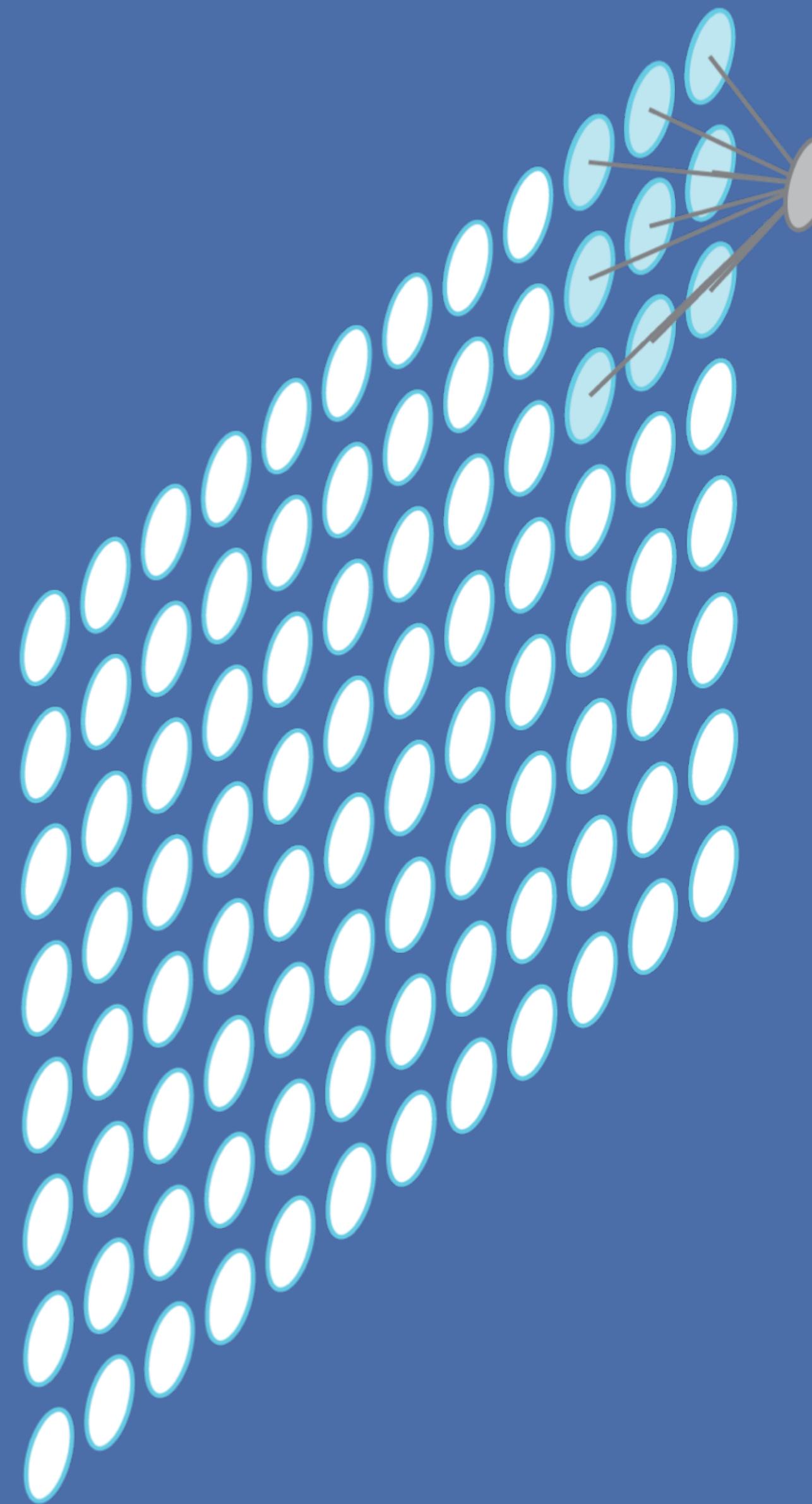
## 2-D Convolutions



Kernel size :  $1 \times 3 \times 3$   
Number of kernels : 1

End-to-end training

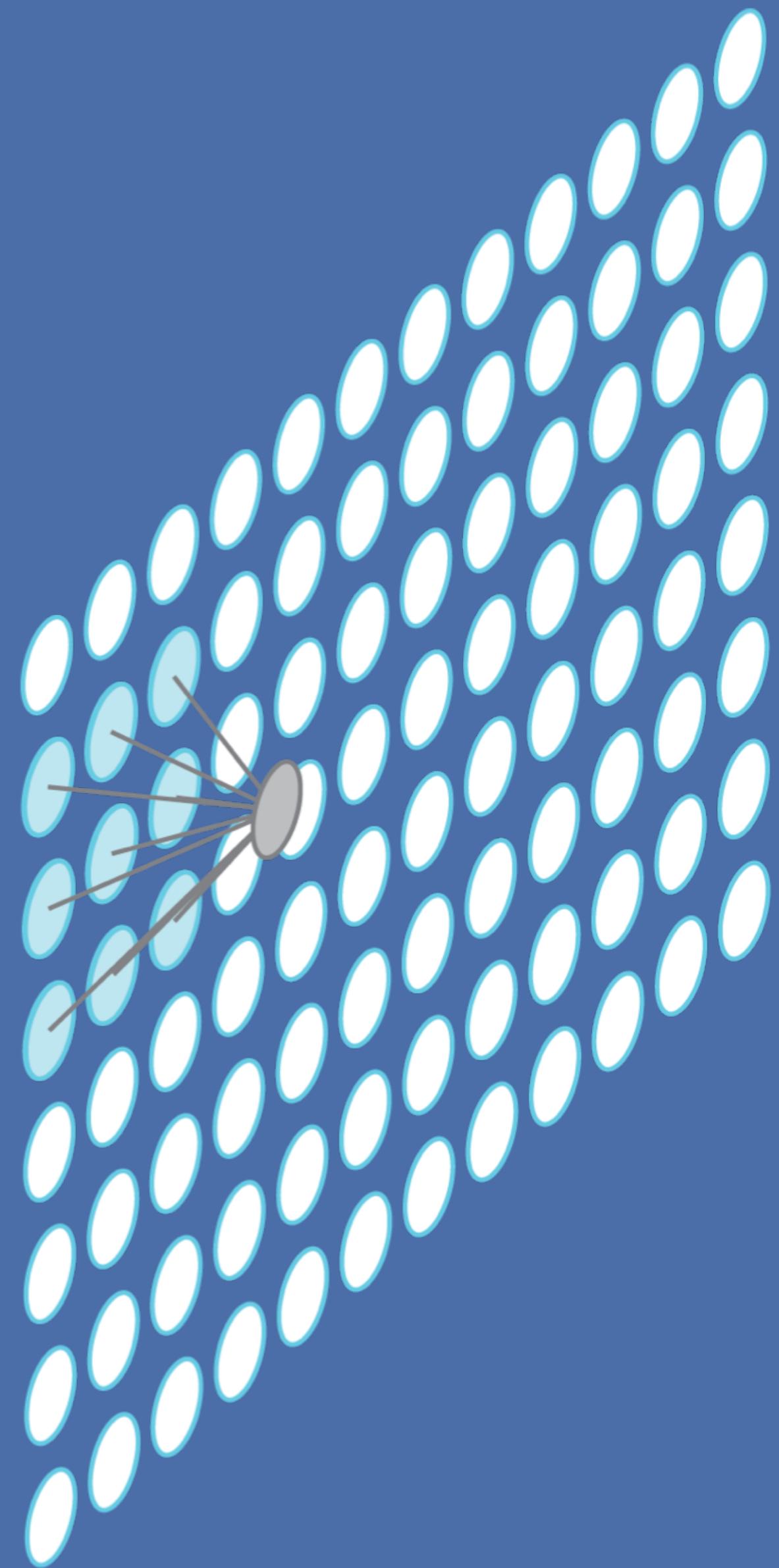
## 2-D Convolutions



Kernel size :  $1 \times 3 \times 3$   
Number of kernels : 1

End-to-end training

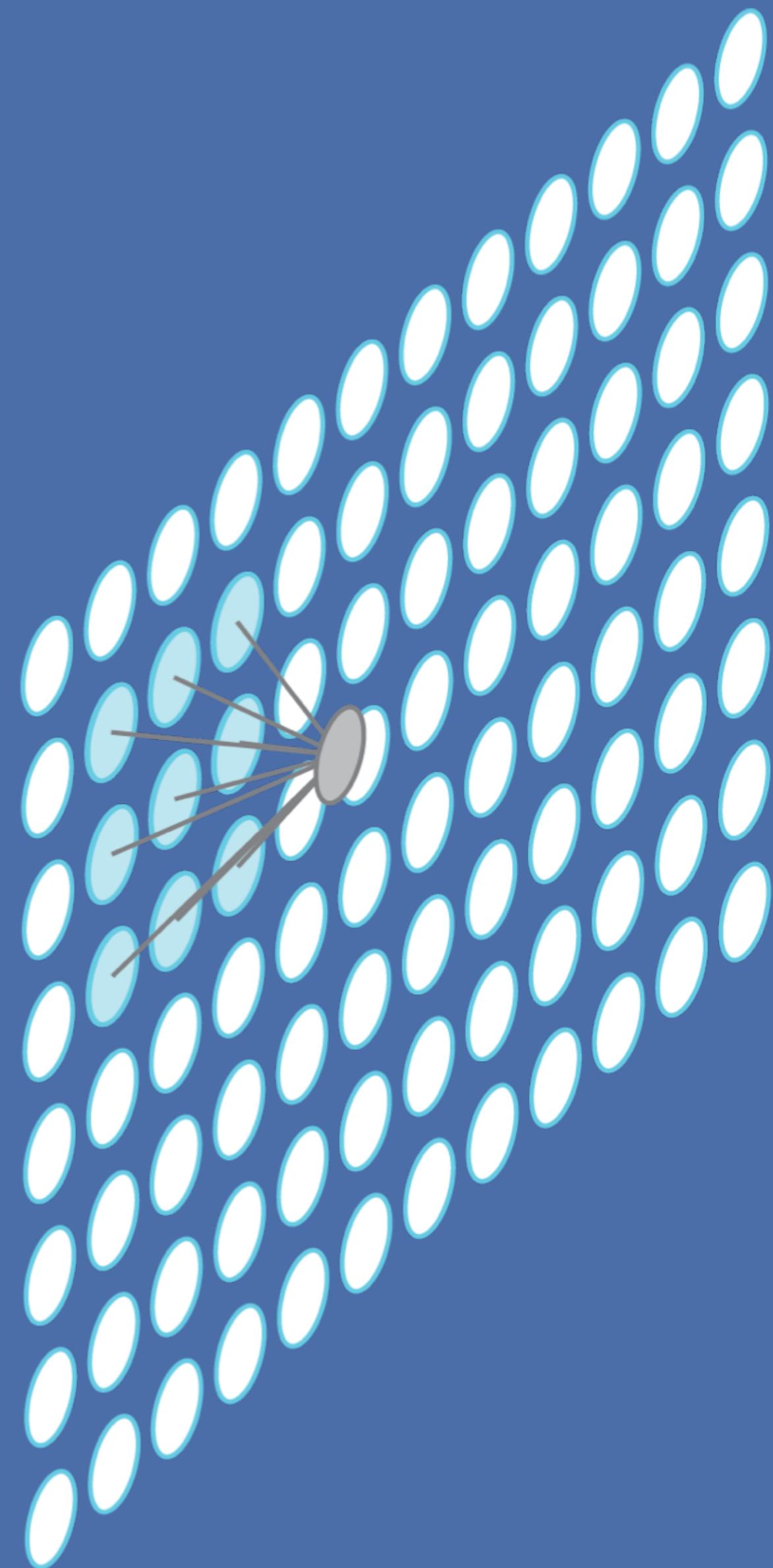
## 2-D Convolutions



Kernel size :  $1 \times 3 \times 3$   
Number of kernels : 1

End-to-end training

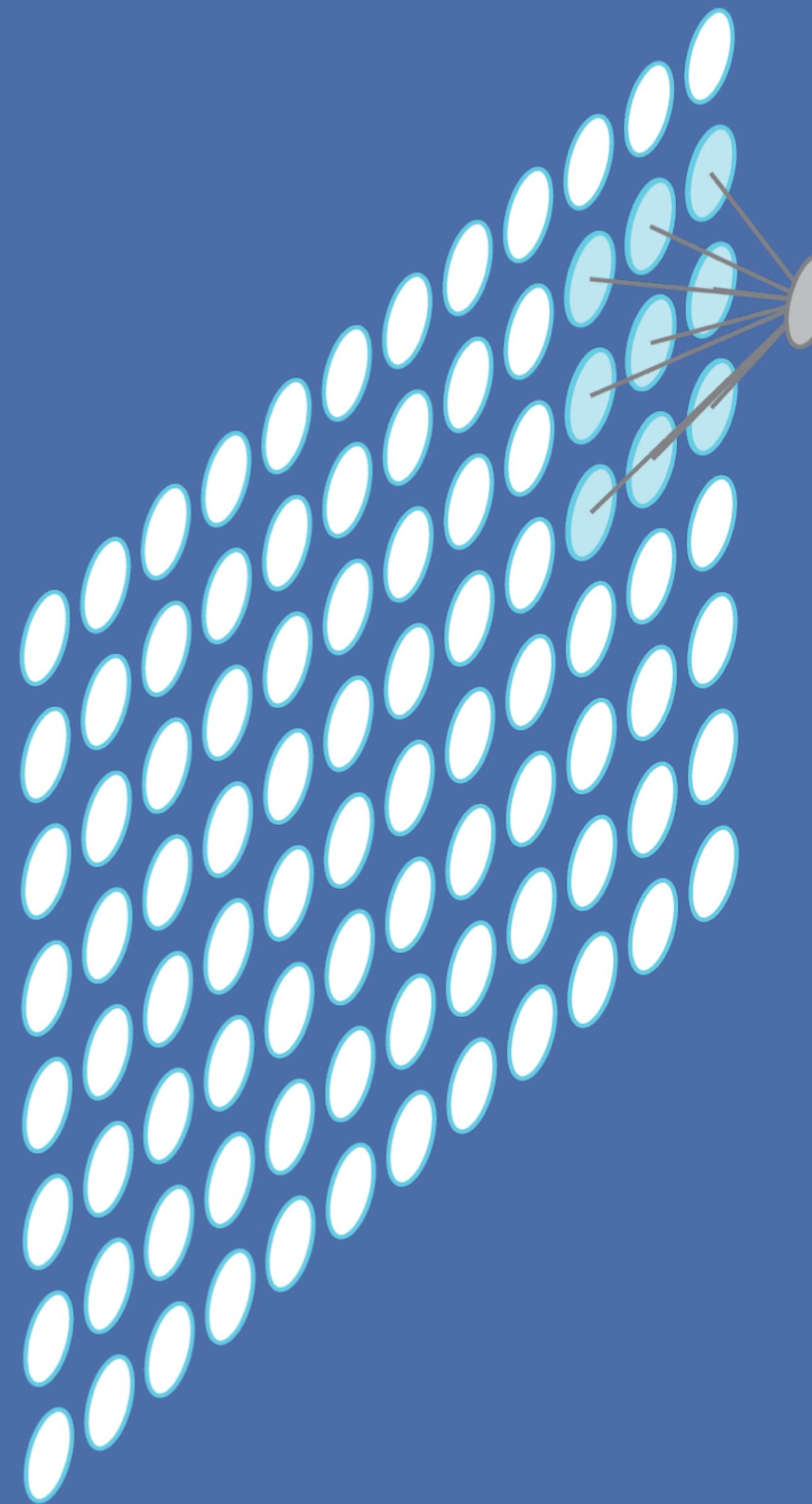
## 2-D Convolutions



Kernel size :  $1 \times 3 \times 3$   
Number of kernels : 1

End-to-end training

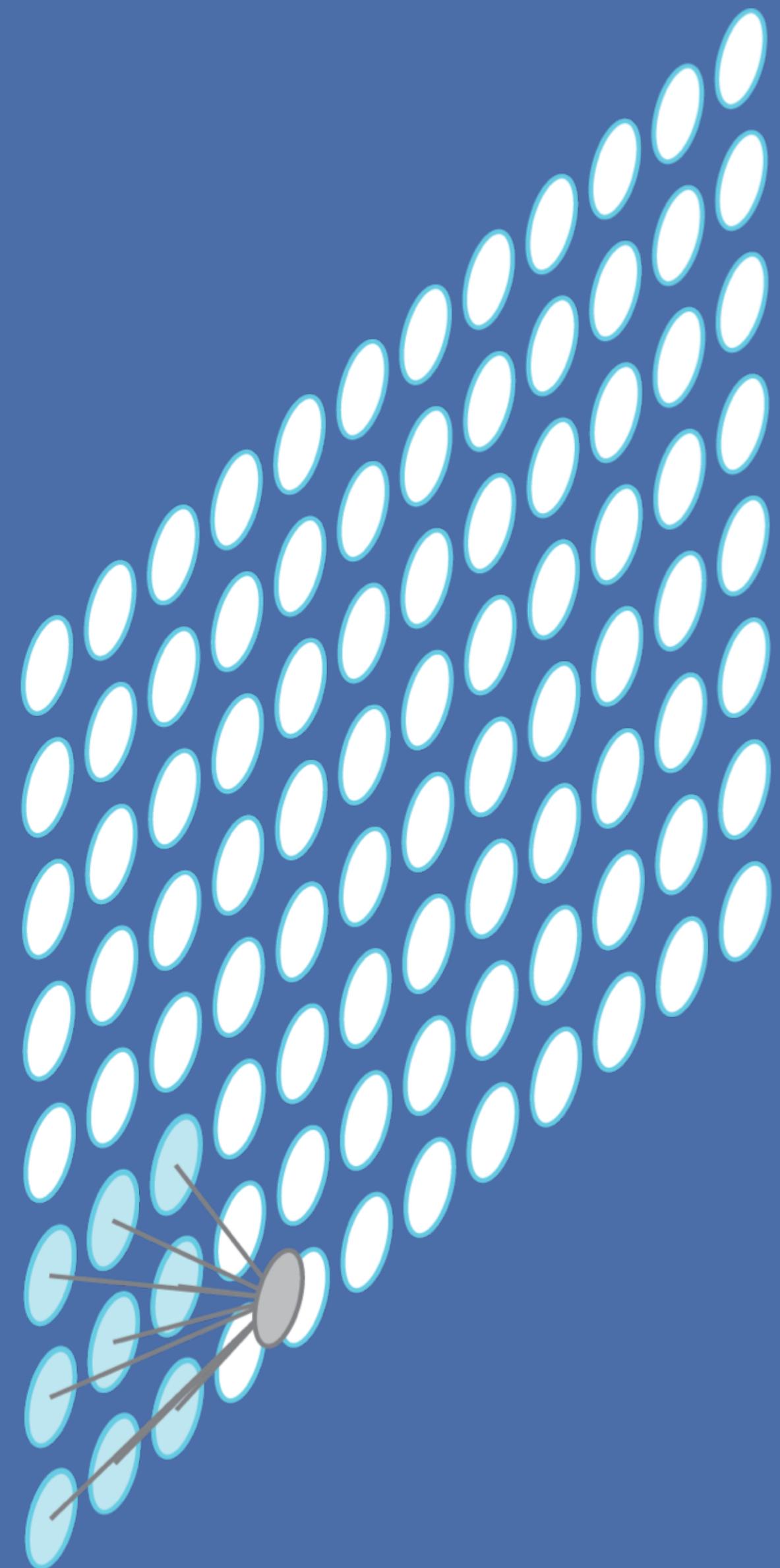
## 2-D Convolutions



Kernel size :  $1 \times 3 \times 3$   
Number of kernels : 1

End-to-end training

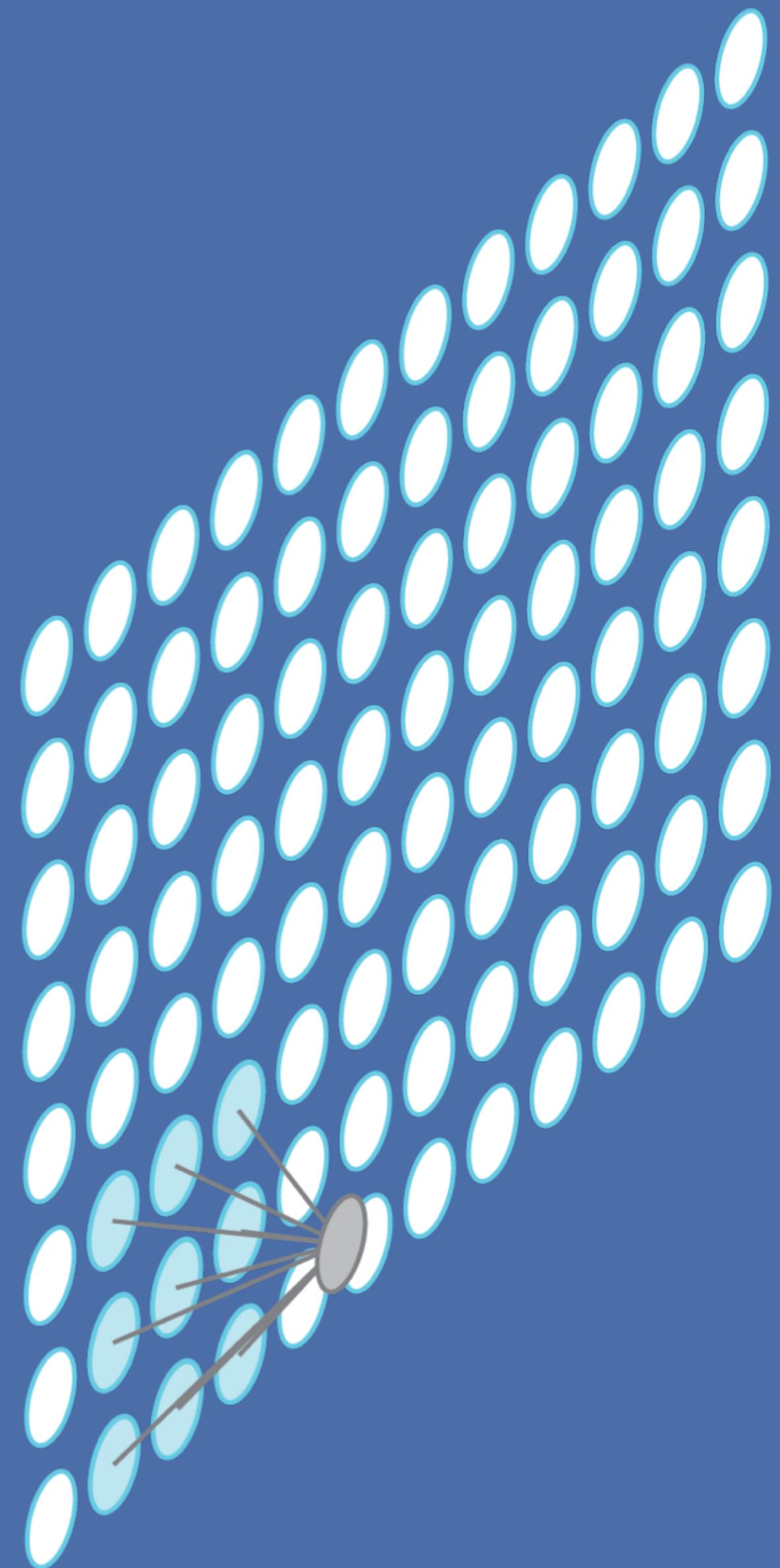
## 2-D Convolutions



Kernel size :  $1 \times 3 \times 3$   
Number of kernels : 1

End-to-end training

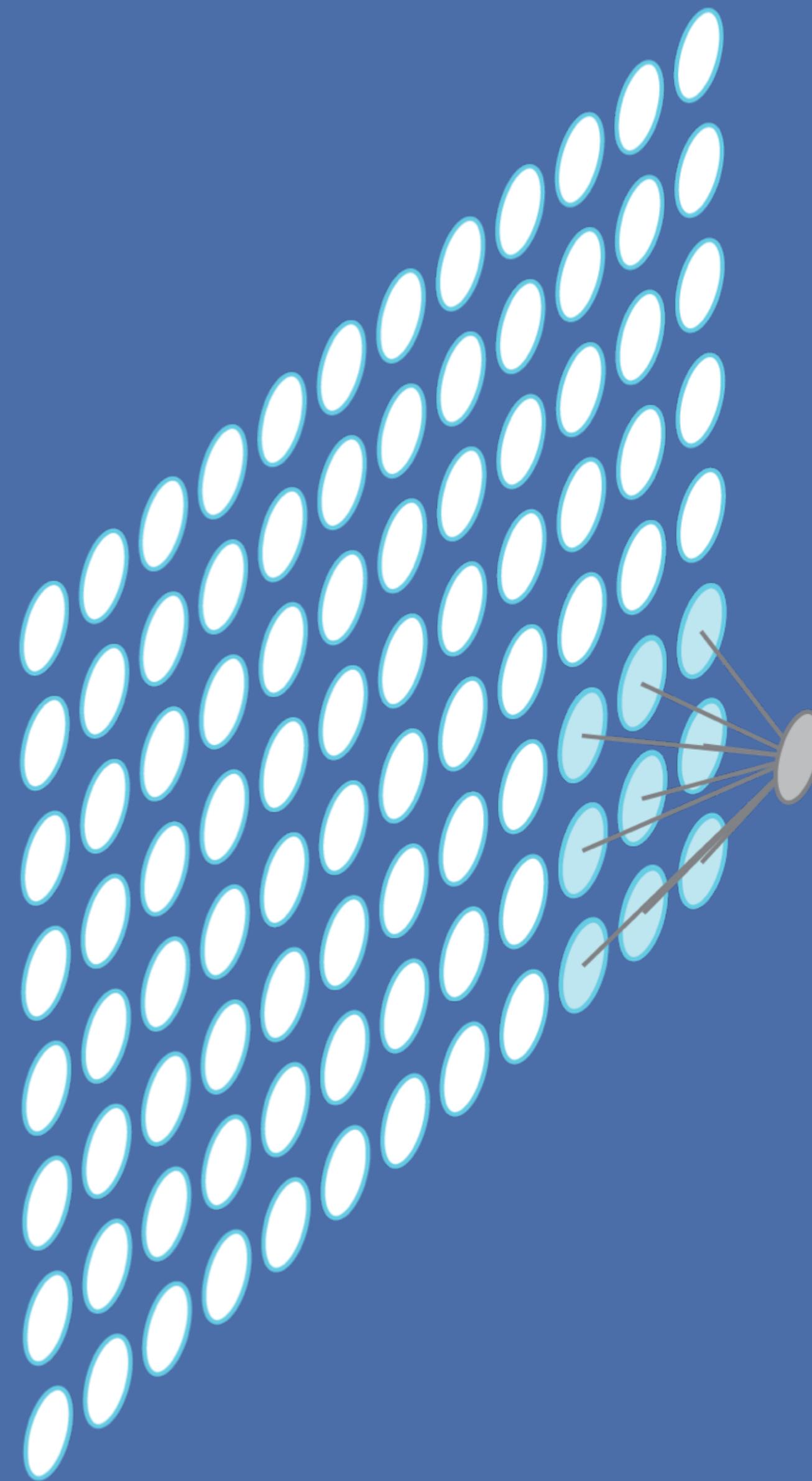
## 2-D Convolutions



Kernel size :  $1 \times 3 \times 3$   
Number of kernels : 1

End-to-end training

## 2-D Convolutions

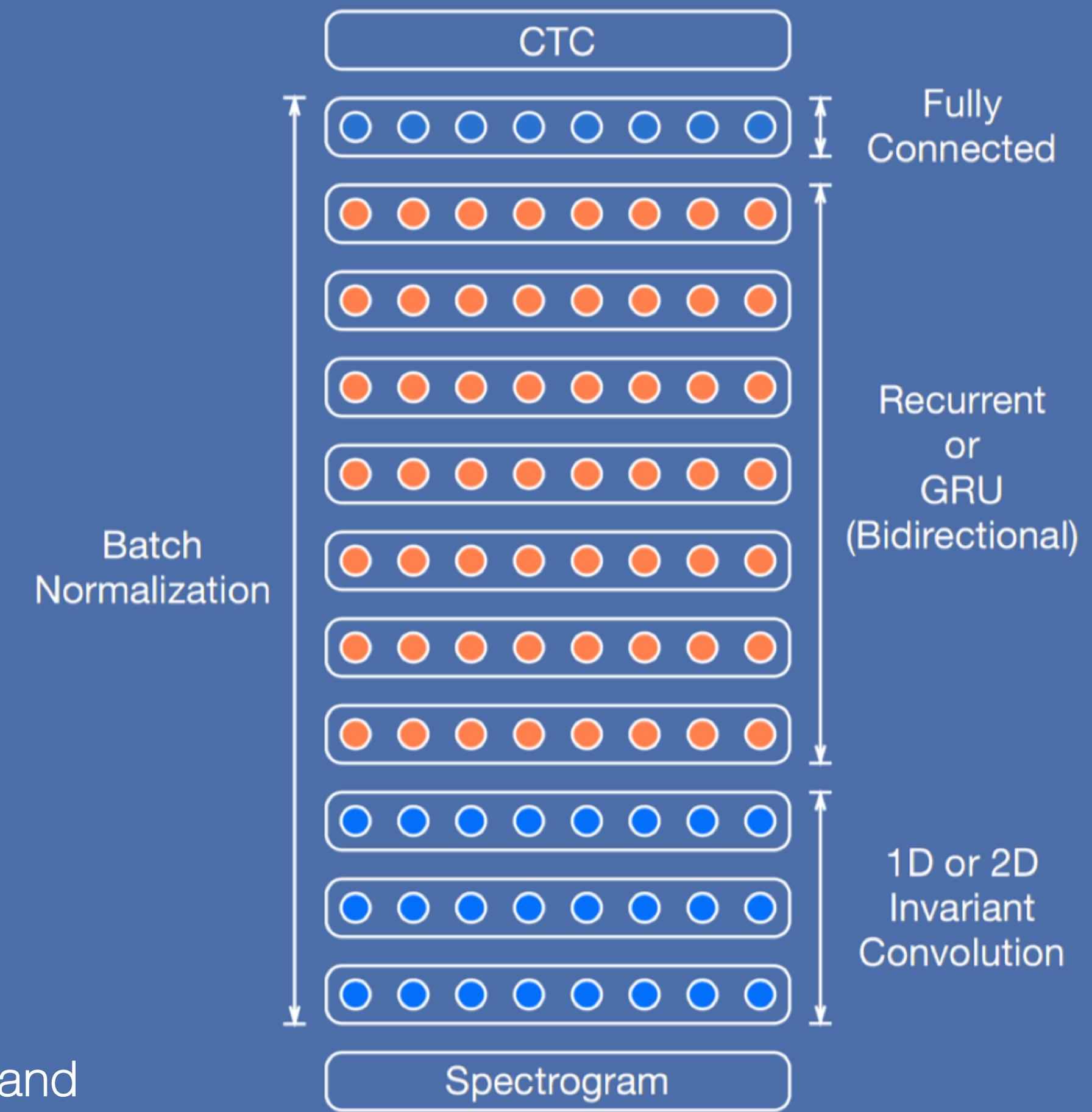


Kernel size :  $1 \times 3 \times 3$   
Number of kernels : 1



## Deep Speech 2

- Baidu system
- Combines convolutions on spectrograms, bi-directional RNN and CTC
- 100m parameters (best models)
- 16 GPUs (50TFlops)
- 3 to 5 days of training
- No speaker normalization (enough data to learn invariances!)



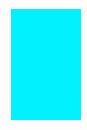
End-to-end training



Current state-of-the-art



- The measure of Performance in speech recognition is called WER (Word Error Rate): how many words the system failed to recognize (in %) -> **the lower the better**
- Best systems are now on par with humans on conversational speech in English (~11% WER)



## Resources



- Kaldi: <http://kaldi-asr.org/>
- Wav2letter (Torch): <https://github.com/facebookresearch/wav2letter>



## Limitations of current approaches

- Challenging types of speech
- Need of data: humans vs machines
- Threats on speech recognition systems

What about accented/noisy speech?

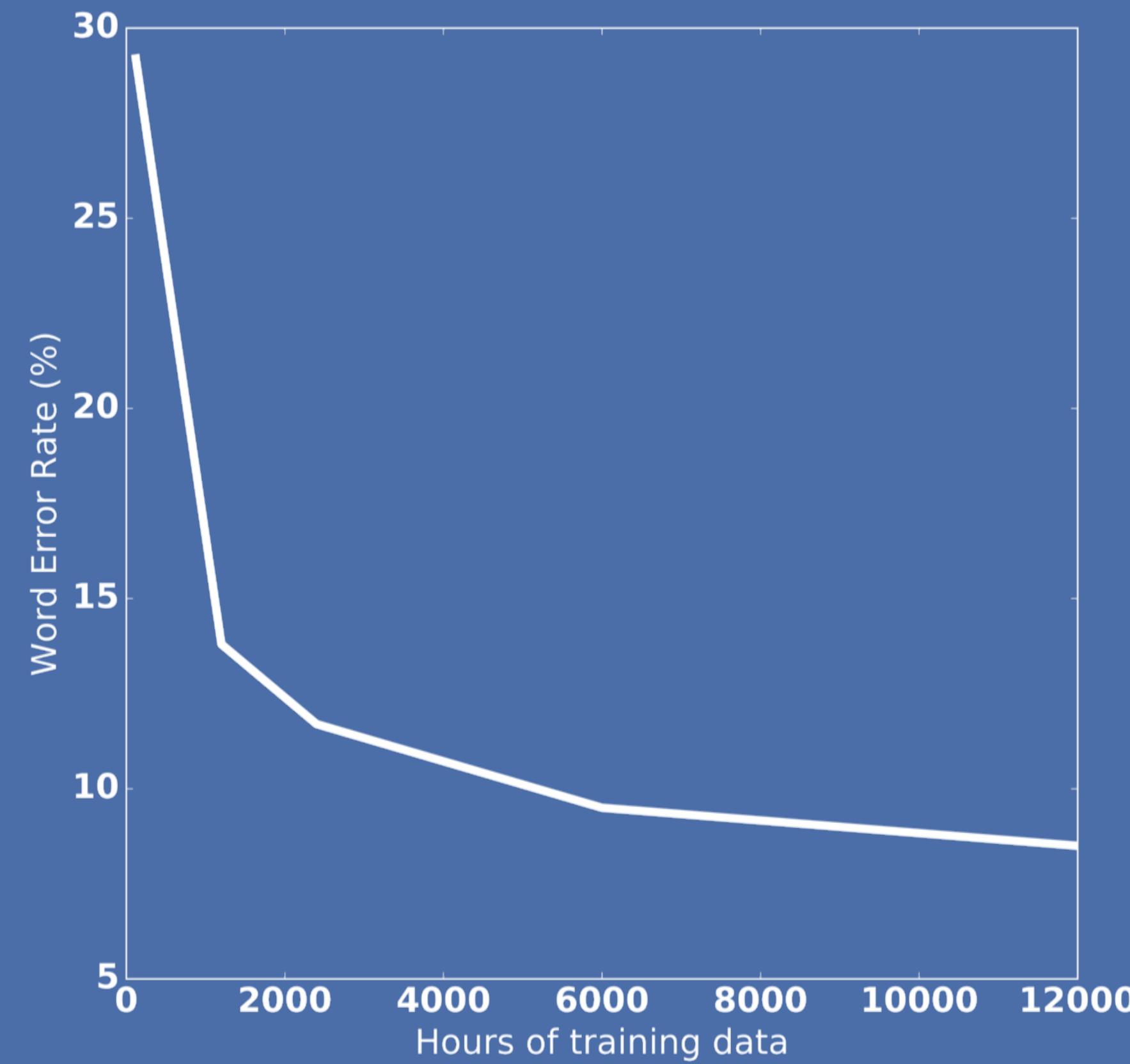
Accented Speech			
Test set	DS1	DS2	Human
VoxForge American-Canadian	15.01	7.55	4.85
VoxForge Commonwealth	28.46	13.56	8.15
VoxForge European	31.20	17.55	12.76
VoxForge Indian	45.35	22.44	22.15

**Word error rate of DeepSpeech 2 on  
accented speech**

Noisy Speech			
Test set	DS1	DS2	Human
CHiME eval clean	6.30	3.34	3.46
CHiME eval real	67.94	21.79	11.84
CHiME eval sim	80.27	45.05	31.33

**Word error rate of DeepSpeech 2 on  
noisy speech**

How much data do we need?

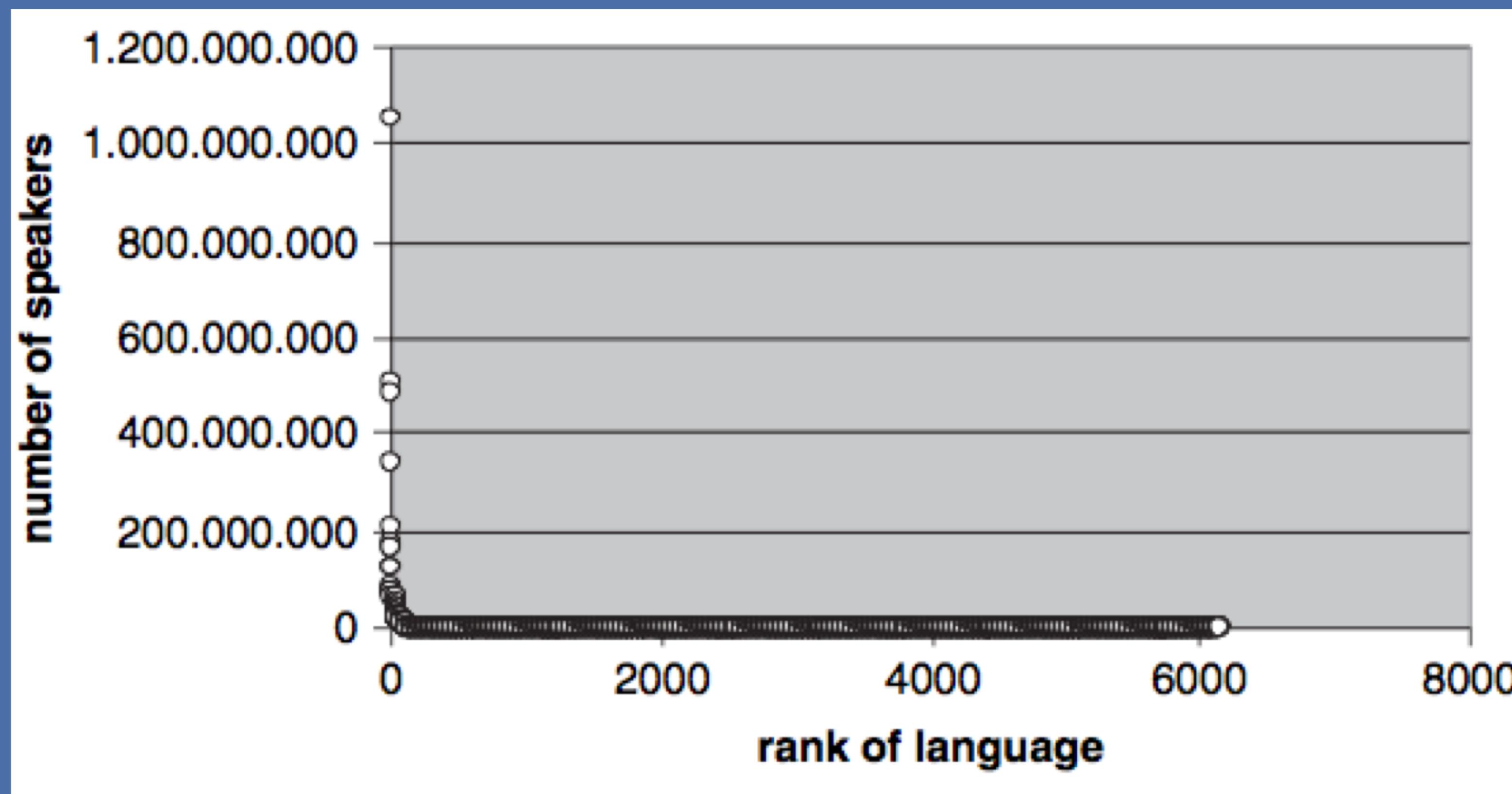


## Challenges

- Low-resource languages
- Languages without standard orthography  
**Half of the ~7000 languages in the world**
- Evolving vocabulary, new categories of users, new accents



## Low-resource languages

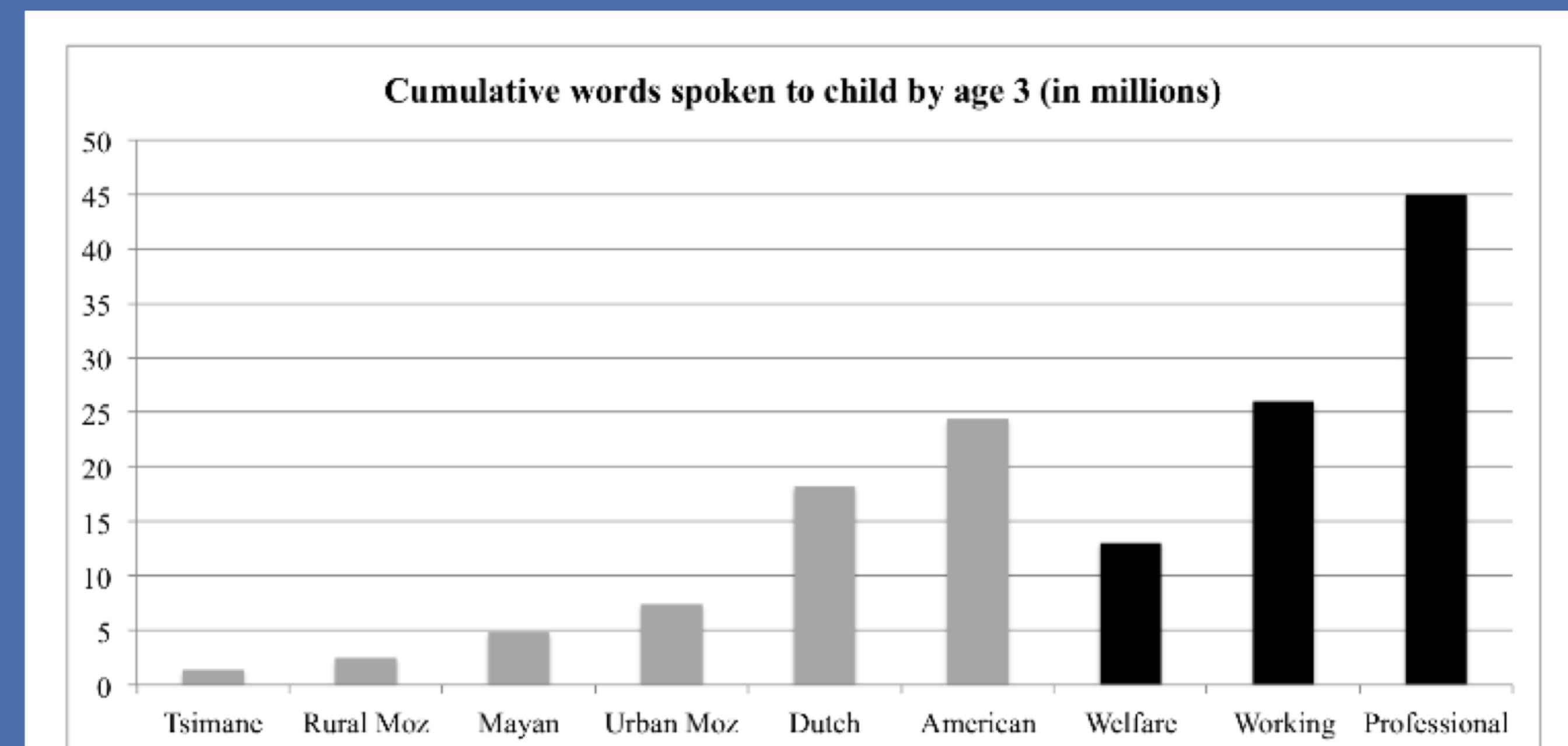


## Limitations of current approaches

- Small and varying quantity of data
- No annotations!
- Two speakers!
- We learn speech recognition, robustness to noise, quick adaptation to new accents

## Efficiency of human learning

From Cristia et al. (2017) Child Development.



↑  
66 h/year

↑  
800 h/year

↑  
1200 h/year

Deep Speech 2  
12000h



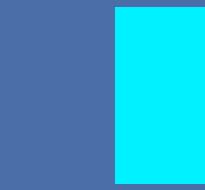
...



Limitations of current approaches



Adversarial examples



$$\tilde{x} = \operatorname{argmax}_{\tilde{x}: \|\tilde{x}-x\|_p \leq \epsilon} \ell(g_\theta(\tilde{x}), y)$$

Perturbation minimal in input space (pixels of an image, waveform of speech)  
but maximal for the classifier



## Adversarial examples



“The bearing was graceful an animated  
she let her son by the hand and before he  
walks two maids with wax lights and silver  
candlesticks.”



## Adversarial examples



“The bearing was graceful an animated  
she let her son by the hand and before he  
walks two maids with wax lights and silver  
candlesticks.”

Adversarial examples



“The bearing was graceful an animated  
she let her son by the hand and before he  
walks two maids with wax lights and silver  
candlesticks.”

Add a bit of noise...

Adversarial examples

Add a bit of noise...



“The bearing was graceful an animated  
she let her son by the hand and before he  
walks two maids with wax lights and silver  
candlesticks.”



“Mary was grateful then admitted she let  
her son before the walks to Mays would  
like slice furnace filter count six.”

## Adversarial examples

Add a bit of noise...



“The bearing was graceful an animated  
she let her son by the hand and before he  
walks two maids with wax lights and silver  
candlesticks.”



“Mary was grateful then admitted she let  
her son before the walks to Mays would  
like slice furnace filter count six.”



## « DolphinAttack »: Inaudible attacks on ASR systems

- Researchers have managed to create voice commands that are carried by a high frequency signal (>20kHz)
- These commands are inaudible for humans, but the microphone of ASR systems retrieves the command
- Without anyone hearing they could:
  - Fool 16 speech recognition systems (Siri, Amazon Echo, Google Nexus, etc.)
  - Launching Facetime, calling desired numbers
  - Playing Music or opening the back door with Amazon Echo
  - Put smartphones in airplane mode
  - Manipulate the navigation system of an Audi