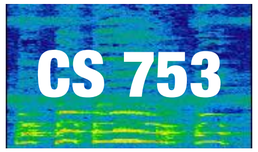
Hybrid/Tandem models + TDNNs + Intro to RNNs

Lecture 8



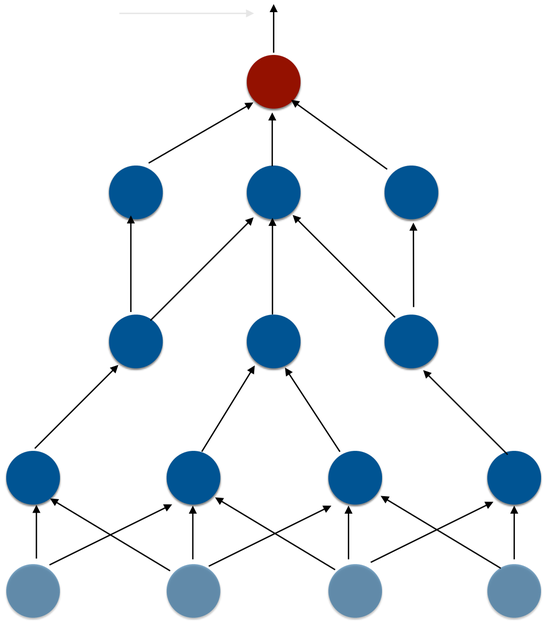
Instructor: Preethi Jyothi

Feedback from in-class quiz 2 (on FSTs)

- Common mistakes
  - Forgetting to consider subset of input alphabet
  - Not careful about only accepting non-empty strings
  - Non-deterministic machines that allow for a larger class of strings than what was specified

Recap: Feedforward Neural Networks

- Deep feedforward neural networks (referred to as DNNs) consist of an input layer, one or more hidden layers and an output layer



- Hidden layers compute non-linear transformations of its inputs.
- Can assume layers are fully connected. Also referred to as *affine* layers.
- Sigmoid, tanh, ReLU are commonly used activation functions

Feedforward Neural Networks for ASR

- Two main categories of approaches have been explored:

1. Hybrid neural network-HMM systems: Use DNNs to estimate HMM observation probabilities

2. Tandem system: NNs used to generate input features that are fed to an HMM-GMM acoustic model

Feedforward Neural Networks for ASR

- Two main categories of approaches have been explored:

1. Hybrid neural network-HMM systems: Use DNNs to estimate HMM observation probabilities

2. Tandem system: DNNs used to generate input features that are fed to an HMM-GMM acoustic model

Decoding an ASR system

- Recall how we decode the most likely word sequence W for an acoustic sequence O:

$$W^* = \arg\max_W \Pr(O|W)\Pr(W)$$

- The acoustic model Pr(O|W) can be further decomposed as (here, Q,M represent triphone, monophone sequences resp.):

$$\begin{aligned}\Pr(O|W) &= \sum_{Q,M} \Pr(O,Q,M|W) \\ &= \sum_{Q,M} \Pr(O|Q,M,W)\Pr(Q|M,W)\Pr(M|W) \\ &\approx \sum_{Q,M} \Pr(O|Q)\Pr(Q|M)\Pr(M|W)\end{aligned}$$

Hybrid system decoding

$$\Pr(O|W) \approx \sum_{Q,M} \Pr(O|Q) \Pr(Q|M) \Pr(M|W)$$

We've seen Pr(O|Q) estimated using a Gaussian Mixture Model.
Let's use a neural network instead to model Pr(O|Q).

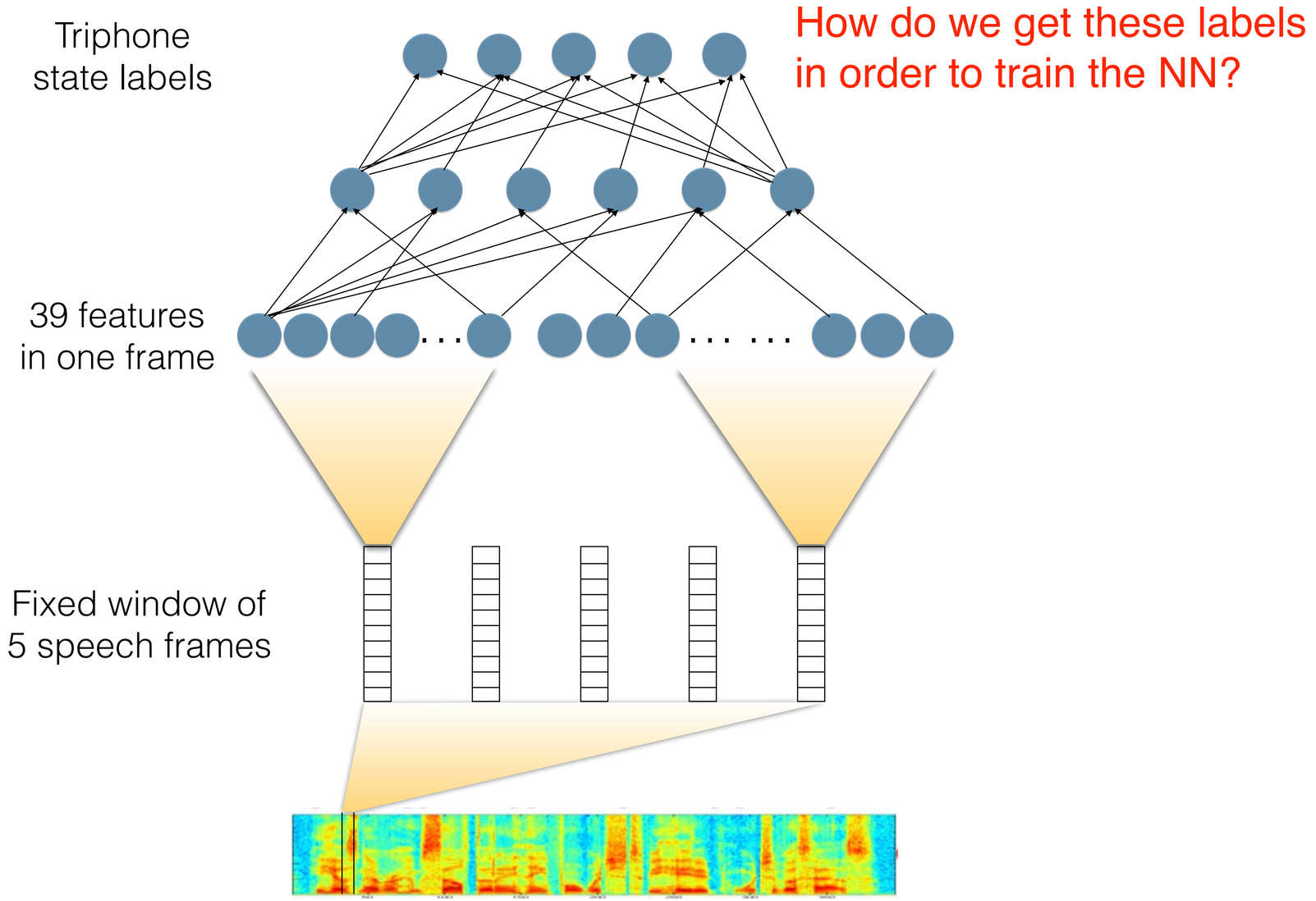
$$\Pr(O|Q) = \prod_t \Pr(o_t|q_t)$$

$$\Pr(o_t|q_t) = \frac{\Pr(q_t|o_t) \Pr(o_t)}{\Pr(q_t)}$$

$$\propto \frac{\Pr(q_t|o_t)}{\Pr(q_t)}$$

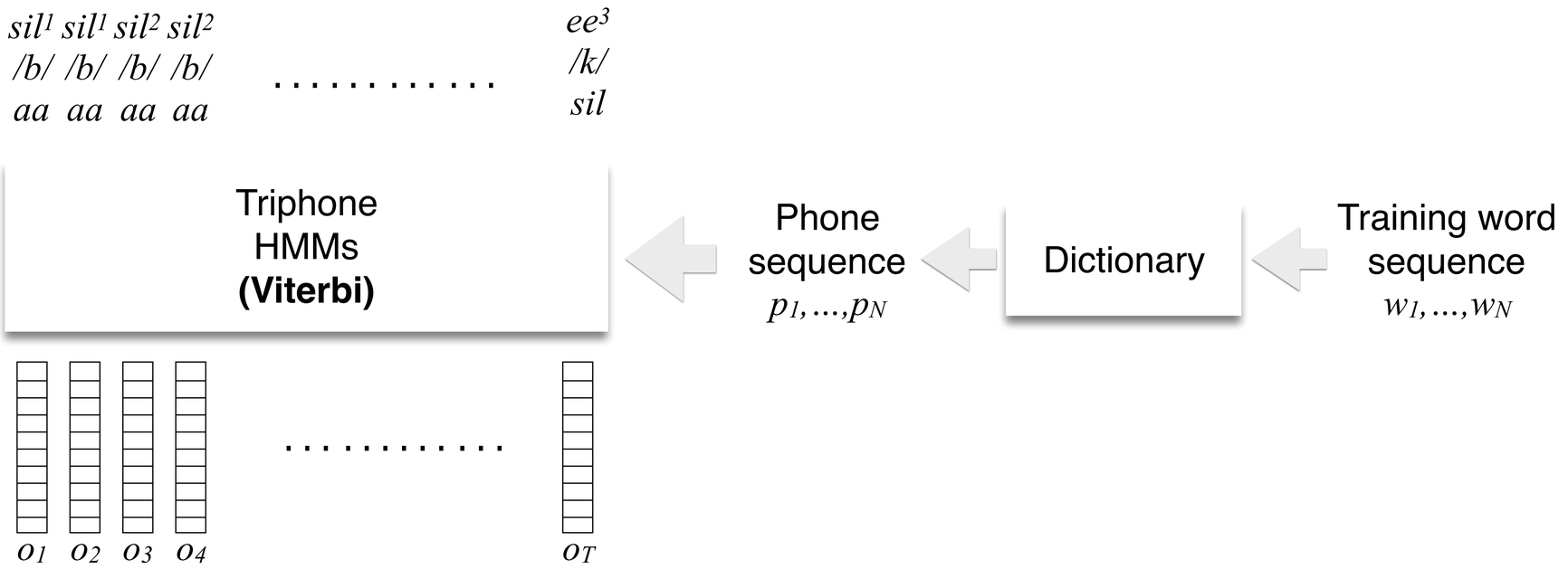
where o_t is the acoustic vector at time t and q_t is a triphone HMM state
Here, Pr(q_t|o_t) are posteriors from a trained neural network.
Pr(o_t|q_t) is then a scaled posterior.

Computing $\Pr(q_t|o_t)$ using a deep NN

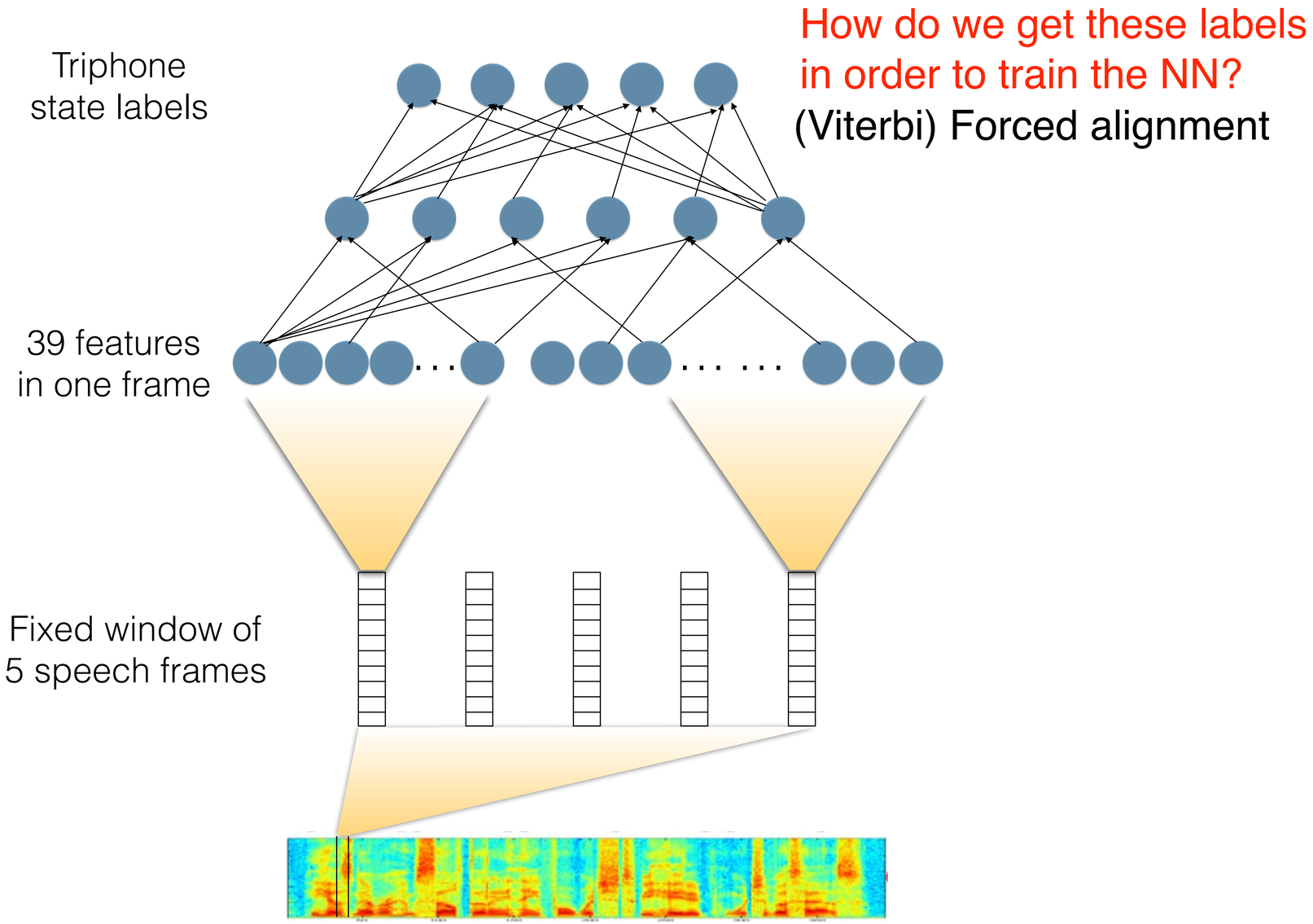


Triphone labels

- Forced alignment: Use current acoustic model to find the most likely sequence of HMM states given a sequence of acoustic vectors. (Algorithm to help compute this?)
- The “Viterbi paths” for the training data, are also referred to as forced alignments



Computing $Pr(q_t|o_t)$ using a deep NN



How do we get these labels in order to train the NN?

(Viterbi) Forced alignment

Computing priors Pr(q_t)

- To compute HMM observation probabilities, Pr($o_t|q_t$), we need both Pr($q_t|o_t$) and Pr(q_t)
- The posterior probabilities Pr($q_t|o_t$) are computed using a trained neural network
- Pr(q_t) are relative frequencies of each triphone state as determined by the forced Viterbi alignment of the training data

Hybrid Networks

- The networks are trained with a minimum cross-entropy criterion

$$L(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i)$$

- Advantages of hybrid systems:

1. Fewer assumptions made about acoustic vectors being uncorrelated: Multiple inputs used from a window of time steps

2. Discriminative objective function used to learn the observation probabilities

Summary of DNN-HMM acoustic models
Comparison against HMM-GMM on different tasks

[TABLE 3] A COMPARISON OF THE PERCENTAGE WERs USING DNN-HMMs AND GMM-HMMs ON FIVE DIFFERENT LARGE VOCABULARY TASKS.

TASK	HOURS OF TRAINING DATA	DNN-HMM	GMM-HMM WITH SAME DATA	GMM-HMM WITH MORE DATA
SWITCHBOARD (TEST SET 1)	309	18.5	27.4	18.6 (2,000 H)
SWITCHBOARD (TEST SET 2)	309	16.1	23.6	17.1 (2,000 H)
ENGLISH BROADCAST NEWS	50	17.5	18.8	
BING VOICE SEARCH (SENTENCE ERROR RATES)	24	30.4	36.2	
GOOGLE VOICE INPUT	5,870	12.3		16.0 (>> 5,870 H)
YOUTUBE	1,400	47.6	52.3	

Hybrid DNN-HMM systems consistently outperform GMM-HMM systems (sometimes even when the latter is trained with lots more data)

Table copied from G. Hinton, et al., “Deep Neural Networks for Acoustic Modeling in Speech Recognition”, IEEE Signal Processing Magazine, 2012.

Neural Networks for ASR

- Two main categories of approaches have been explored:

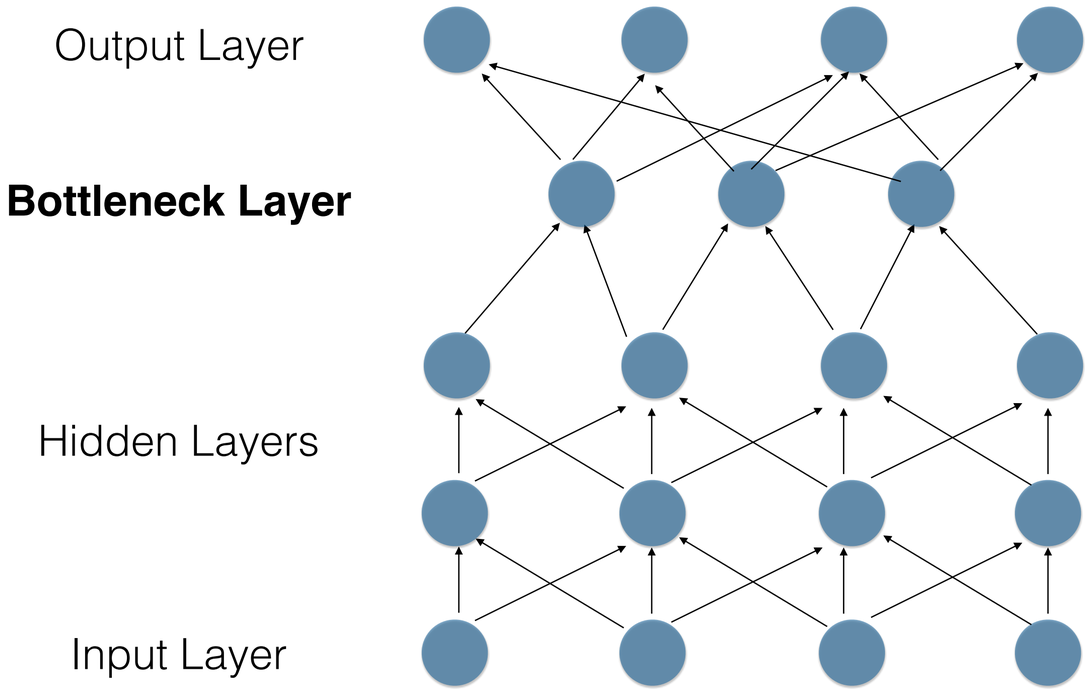
1. Hybrid neural network-HMM systems: Use DNNs to estimate HMM observation probabilities

2. Tandem system: NNs used to generate input features that are fed to an HMM-GMM acoustic model

Tandem system

- First, train a DNN to estimate the posterior probabilities of each subword unit (monophone, triphone state, etc.)
- In a hybrid system, these posteriors (after scaling) would be used as observation probabilities for the HMM acoustic models
- In the tandem system, the DNN outputs are used as “feature” inputs to HMM-GMM models

Bottleneck Features

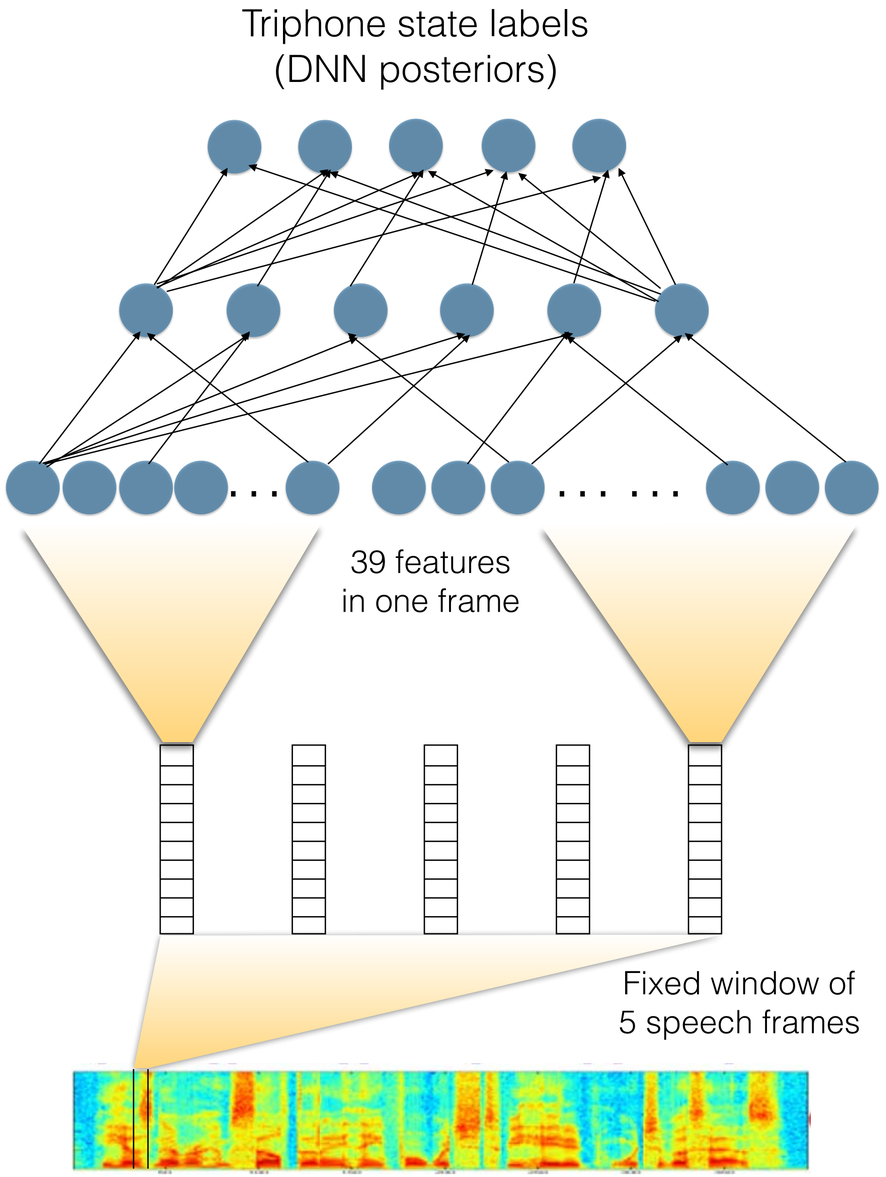


Use a low-dimensional bottleneck layer representation to extract features

These bottleneck features are in turn used as inputs to HMM-GMM models

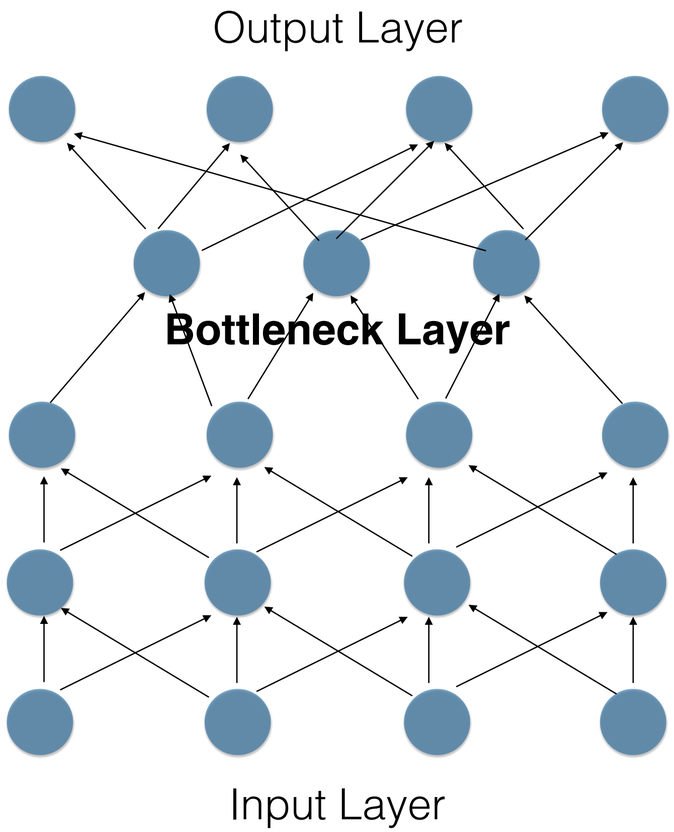
Recap: Hybrid DNN-HMM Systems

- Instead of GMMs, use scaled DNN posteriors as the HMM observation probabilities
- DNN trained using triphone labels derived from a forced alignment "Viterbi" step.
  - Forced alignment: Given a training utterance $\{O, W\}$, find the most likely sequence of states (and hence triphone state labels) using a set of trained triphone HMM models, M. Here M is constrained by the triphones in W.



Recap: Tandem DNN-HMM Systems

- Neural networks are used as “feature extractors” to train HMM-GMM models
- Use a low-dimensional bottleneck layer representation to extract features from the bottleneck layer
- These bottleneck features are subsequently fed to GMM-HMMs as input

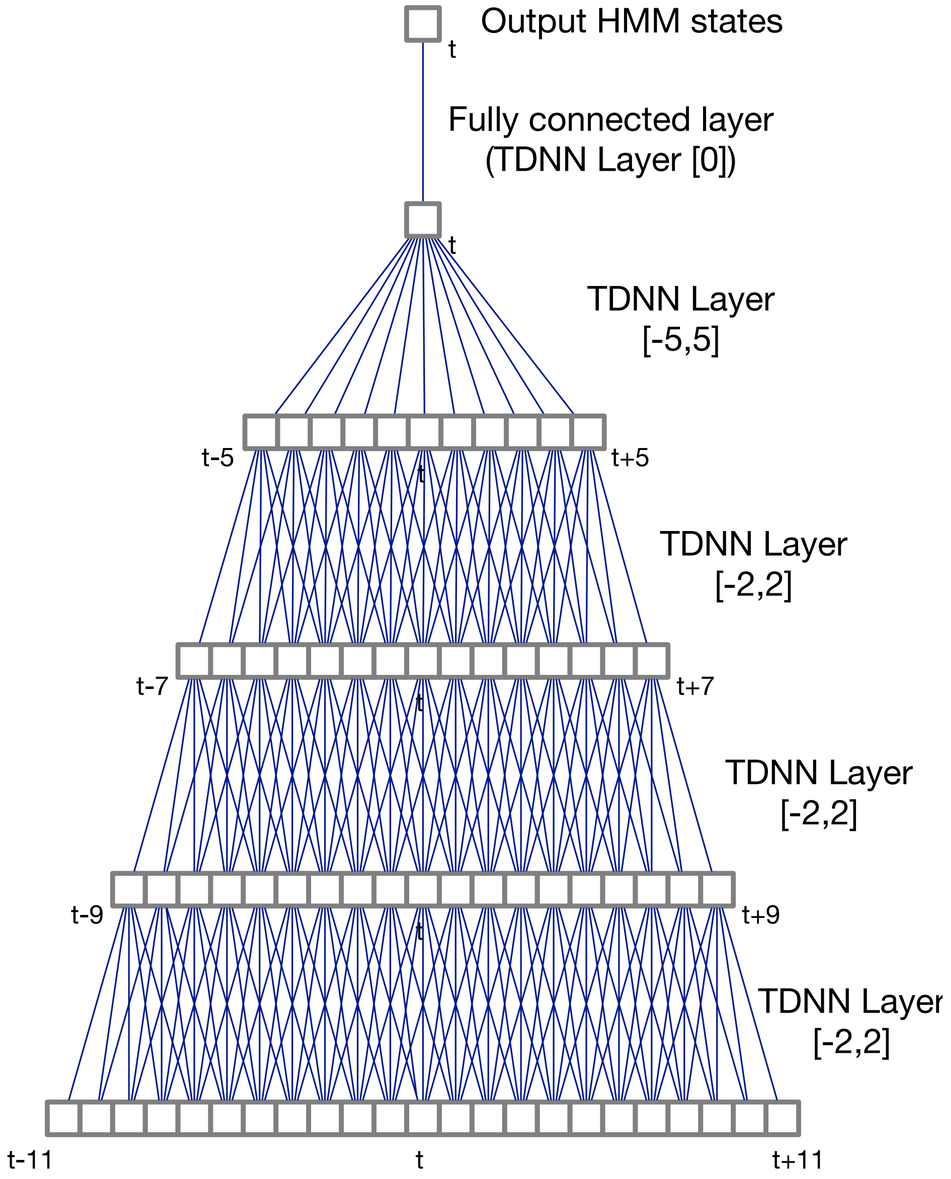


Feedforward DNNs we've seen so far…

- Assume independence among the training instances (modulo the context window of frames)
  - Independent decision made about classifying each individual speech frame
  - Network state is completely reset after each speech frame is processed
- This independence assumption fails for data like speech which has temporal and sequential structure
- Two model architectures that capture longer ranges of acoustic context:
  1. **Time delay neural networks (TDNNs)**

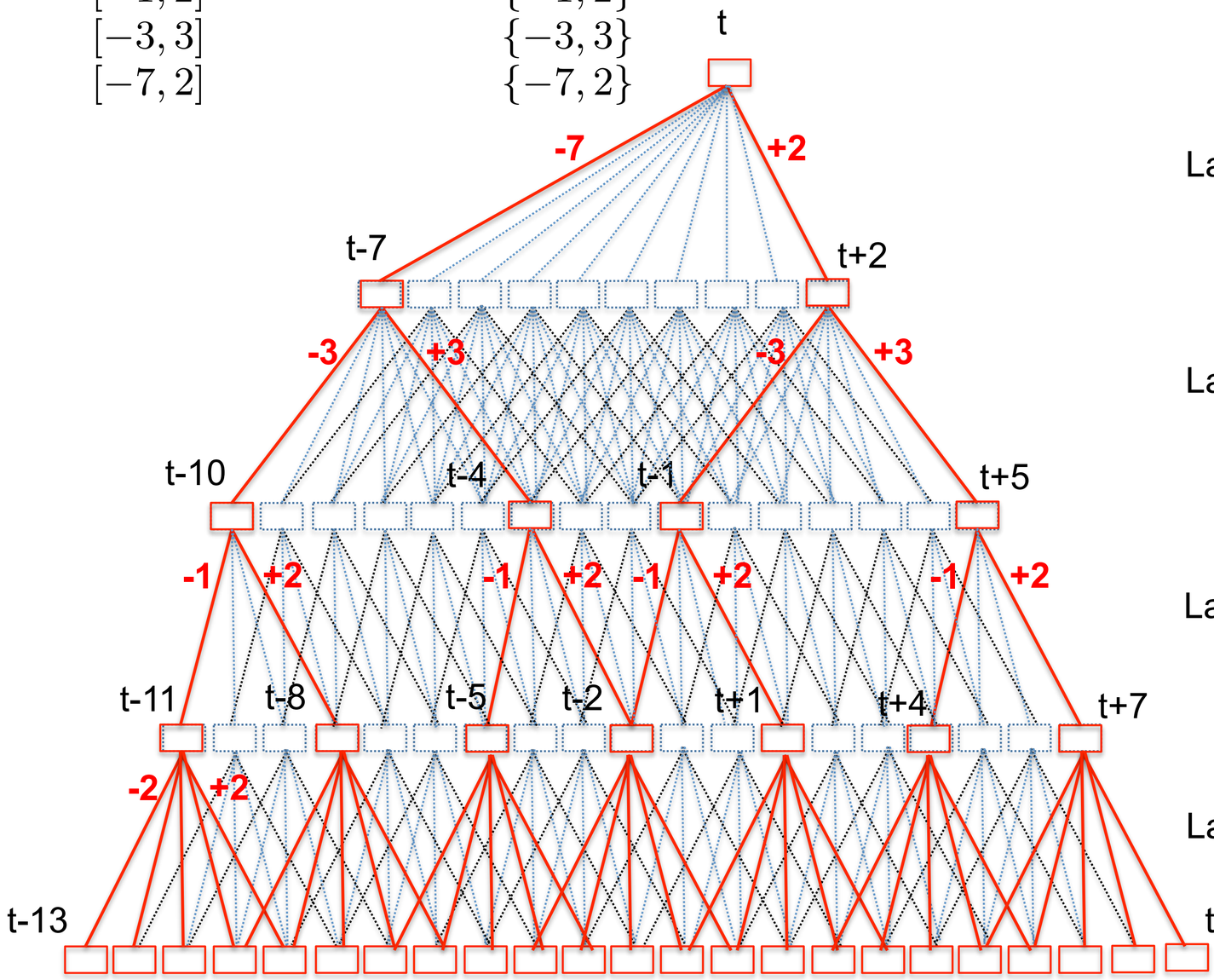
Time Delay Neural Networks

- Each layer in a TDNN acts at a different temporal resolution
  - Processes a context window from the previous layer
  - Higher layers have a wider receptive field into the input
- However, a lot more computation needed than DNNs!



Time Delay Neural Networks

Layer	Input context	Input context with sub-sampling
1	$[-2, +2]$	$[-2, 2]$
2	$[-1, 2]$	$\{-1, 2\}$
3	$[-3, 3]$	$\{-3, 3\}$
4	$[-7, 2]$	$\{-7, 2\}$



- Large overlaps between input contexts computed at neighbouring time steps
- Assuming neighbouring activations are correlated, how do we exploit this?
  - Subsample by allowing gaps between frames.
  - Splice increasingly wider context in higher layers.

Time Delay Neural Networks

Model	Network Context	Layerwise Context					WER	
		1	2	3	4	5	Total	SWB
DNN-A	$[-7, 7]$	$[-7, 7]$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	22.1	15.5
DNN-A_2	$[-7, 7]$	$[-7, 7]$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	**21.6**	**15.1**
DNN-B	$[-13, 9]$	$[-13, 9]$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	22.3	15.7
DNN-C	$[-16, 9]$	$[-16, 9]$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	22.3	15.7
TDNN-A	$[-7, 7]$	$[-2, 2]$	$\{-2, 2\}$	$\{-3, 4\}$	$\{0\}$	$\{0\}$	21.2	14.6
TDNN-B	$[-9, 7]$	$[-2, 2]$	$\{-2, 2\}$	$\{-5, 3\}$	$\{0\}$	$\{0\}$	21.2	14.5
TDNN-C	$[-11, 7]$	$[-2, 2]$	$\{-1, 1\}$	$\{-2, 2\}$	$\{-6, 2\}$	$\{0\}$	20.9	14.2
TDNN-D	$[-13, 9]$	$[-2, 2]$	$\{-1, 2\}$	$\{-3, 4\}$	$\{-7, 2\}$	$\{0\}$	**20.8**	**14.0**
TDNN-E	$[-16, 9]$	$[-2, 2]$	$\{-2, 2\}$	$\{-5, 3\}$	$\{-7, 2\}$	$\{0\}$	20.9	14.2

Feedforward DNNs we've seen so far…

- Assume independence among the training instances
  - Independent decision made about classifying each individual speech frame
  - Network state is completely reset after each speech frame is processed
- This independence assumption fails for data like speech which has temporal and sequential structure
- Two model architectures that capture longer ranges of acoustic context:
  1. Time delay neural networks (TDNNs)
  2. **Recurrent neural networks (RNNs)**