

# Deep Learning for Speech Generation and Synthesis

*Yao Qian Frank K. Soong*

Speech Group  
MS Research Asia

# Background

- Deep learning has made a huge impact on automatic speech recognition (ASR) research, products and services
- Speech generation and synthesis is an inverse process of speech recognition  
Text-to-Speech (TTS) → Speech-to-Text (STT)
- Deep learning approaches to speech generation and synthesis  
Focus on TTS synthesis and voice conversion

# Outline

- Statistical parametric speech generation and synthesis
  - HMM-based speech synthesis
  - GMM-based voice conversion
- Deep learning
  - RBM, DBN, DNN, MDN and RNN
- Deep learning for speech generation and synthesis
  - Approaches to speech synthesis
  - Approaches to voice conversion
- Conclusions and future work

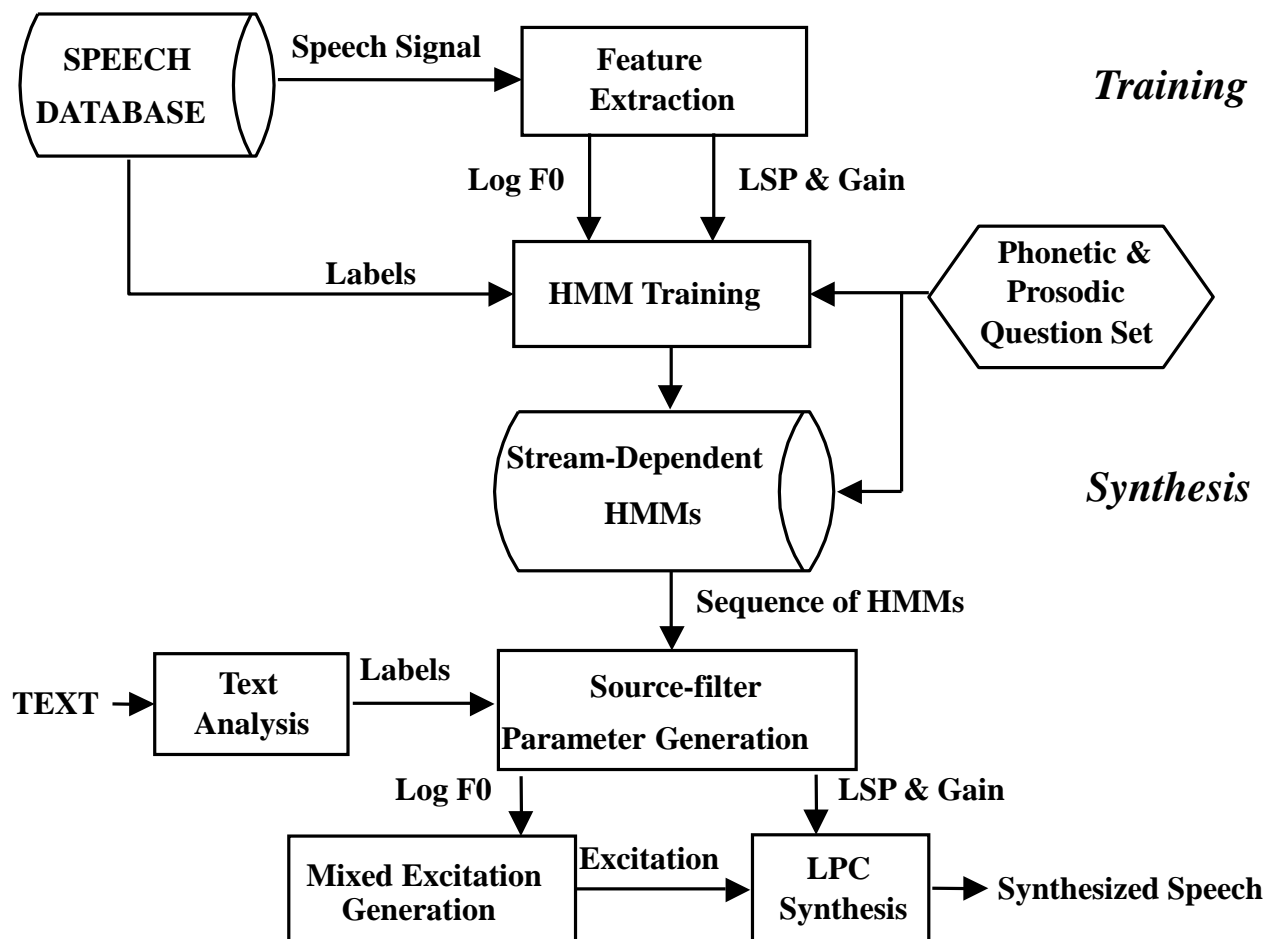
# Outline

- Statistical parametric speech generation and synthesis
  - HMM-based speech synthesis
  - GMM-based voice conversion
- Deep learning
  - RBM, DBN, DNN, MDN and RNN
- Deep learning for speech generation and synthesis
  - Approaches to speech synthesis
  - Approaches to voice conversion
- Conclusions and future work

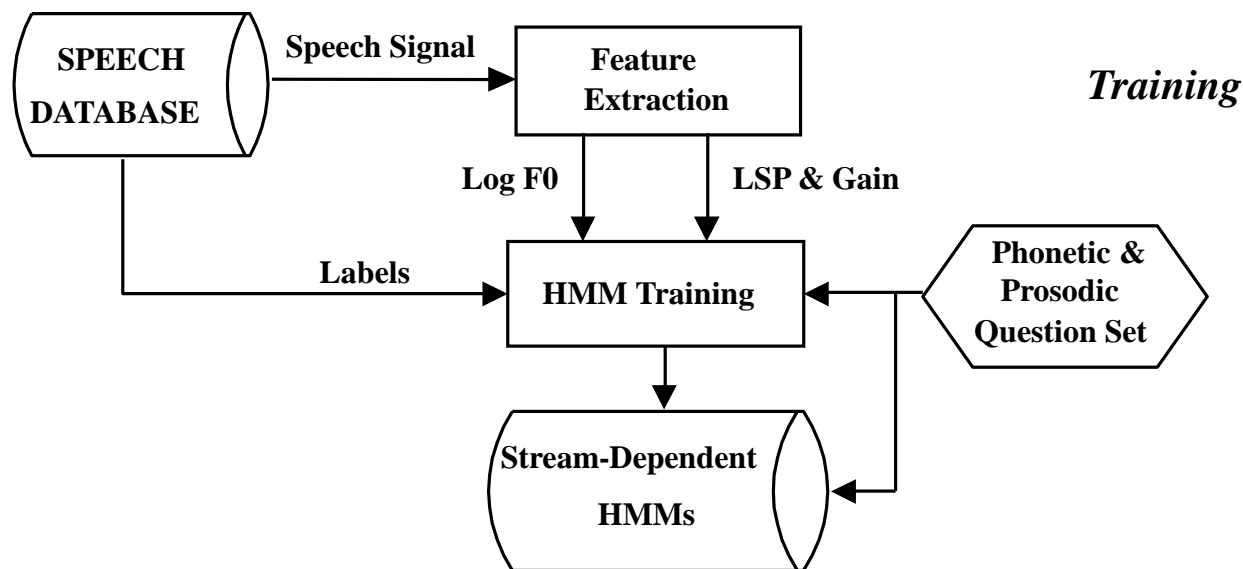
# HMM-based Speech Synthesis<sup>[1]</sup>

- By reversing the input and output of a Hidden Markov Model (HMM) in speech recognition, we then turn HMM into a generation model for text-to-speech (TTS) synthesis.
- Speech spectrum, fundamental frequency, voicing and duration are modeled simultaneously by HMM <sup>[2]</sup>.
- Models are trained and signals are generated by the universal Maximum Likelihood (ML) criterion.

# HMM-based Speech Synthesis<sup>[1]</sup>



# HMM-based Speech Synthesis<sup>[1]</sup>



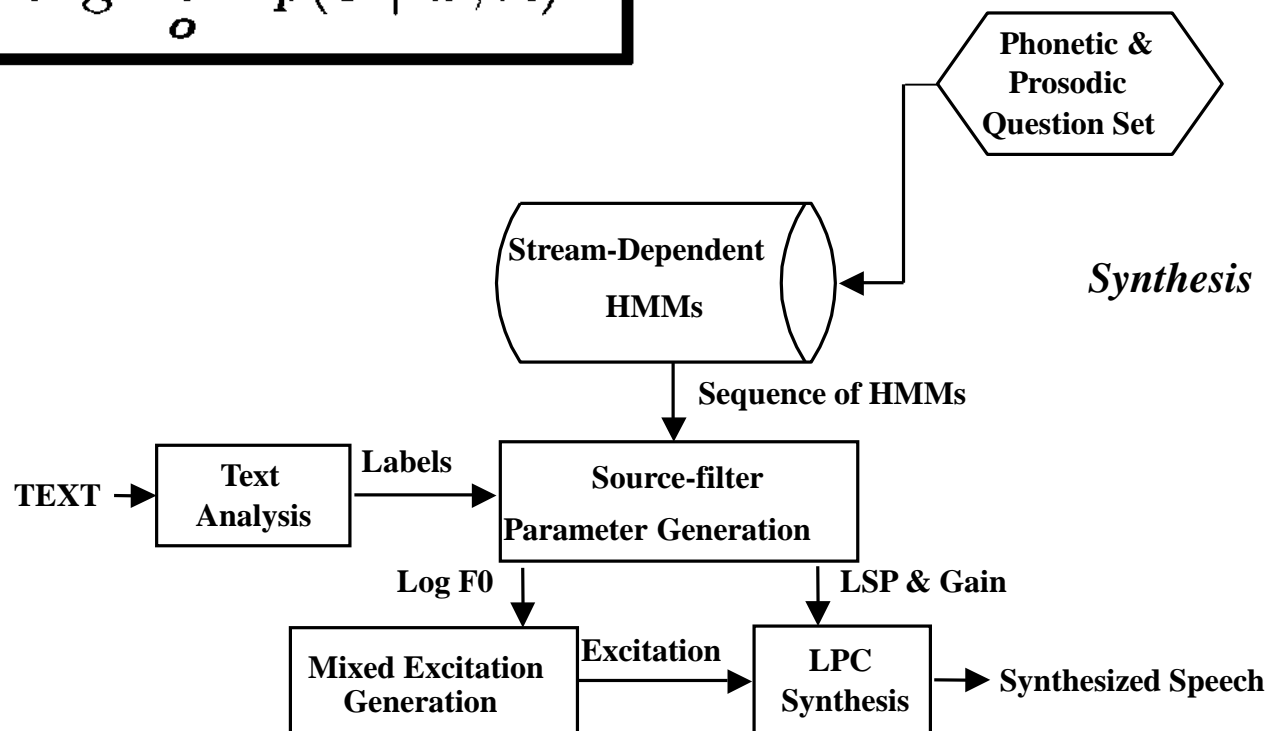
- Training

$$\hat{\lambda} = \arg \max_{\lambda} p(\mathbf{O} \mid \mathcal{W}, \lambda)$$

# HMM-based Speech Synthesis<sup>[1]</sup>

## - Synthesis

$$\hat{o} = \arg \max_o p(o | w, \hat{\lambda})$$





# Advantages of HMM-based TTS

- Statistical parametric model can be efficiently trained with recorded speech data and corresponding transcriptions.
- HMM-based statistical model is parsimonious (i.e., small model size) and data efficient.
- HMMs can generate “optimal” speech parameter trajectory in the ML sense.

# Concatenative vs. HMM-based TTS synthesis

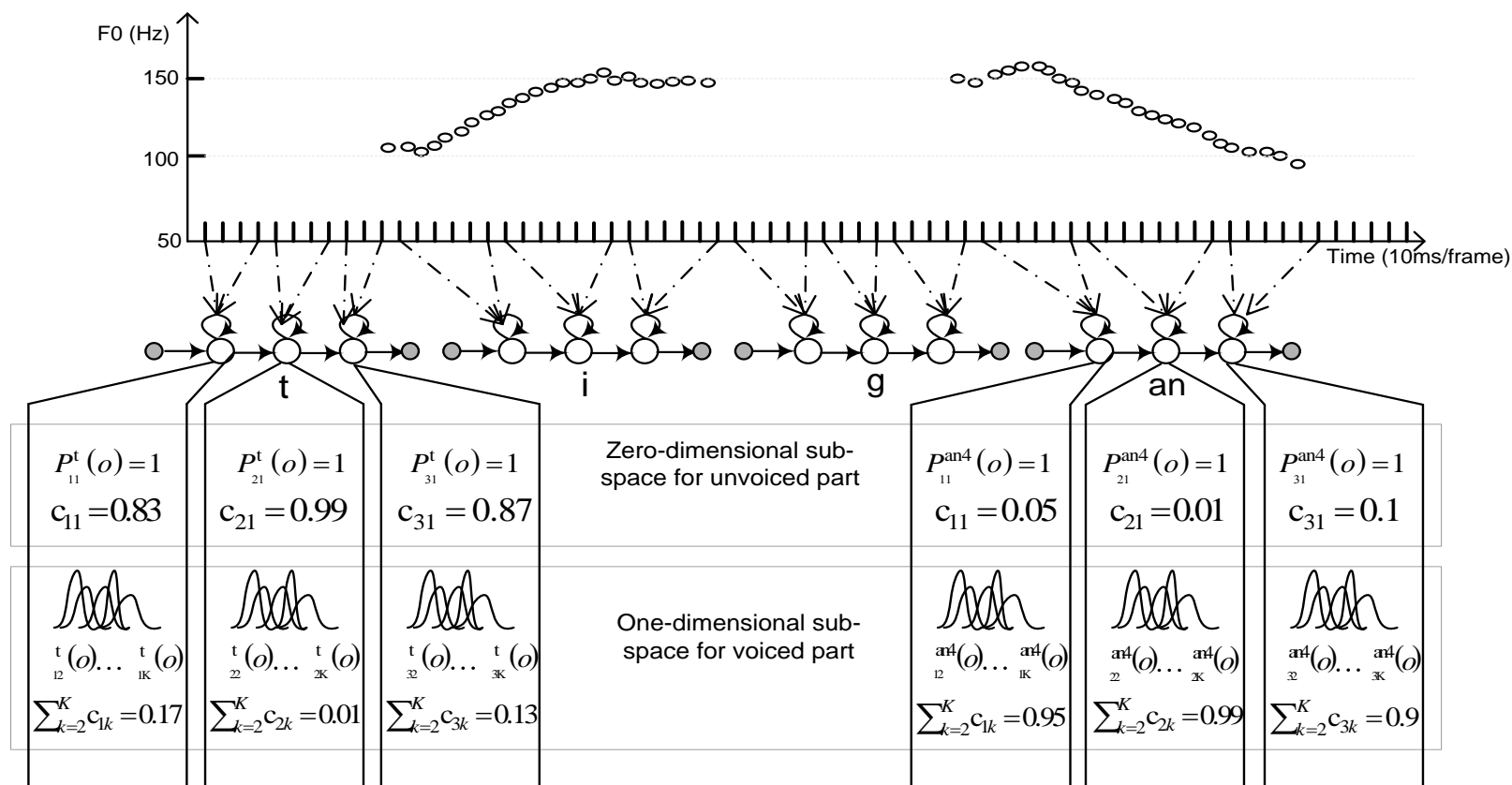
- HMM-based synthesis
  - Statistically trained
  - Vcoded speech (smooth & stable)
  - Small footprint (less than 2MB)
  - Easy to modify its voice characteristics
- Concatenative synthesis
  - Sample-based (unit selection)
  - High quality with occasional glitches
  - Large footprint
  - More difficult to modify its voice characteristics

# Technologies

- Multi-Space Distribution (MSD)-HMM for F0 and voicing modeling
- Contextual clustering of decision tree
- Parameter generation with dynamic features
- Mixed excitation
- Global Variance
- Minimum Generation Error Training

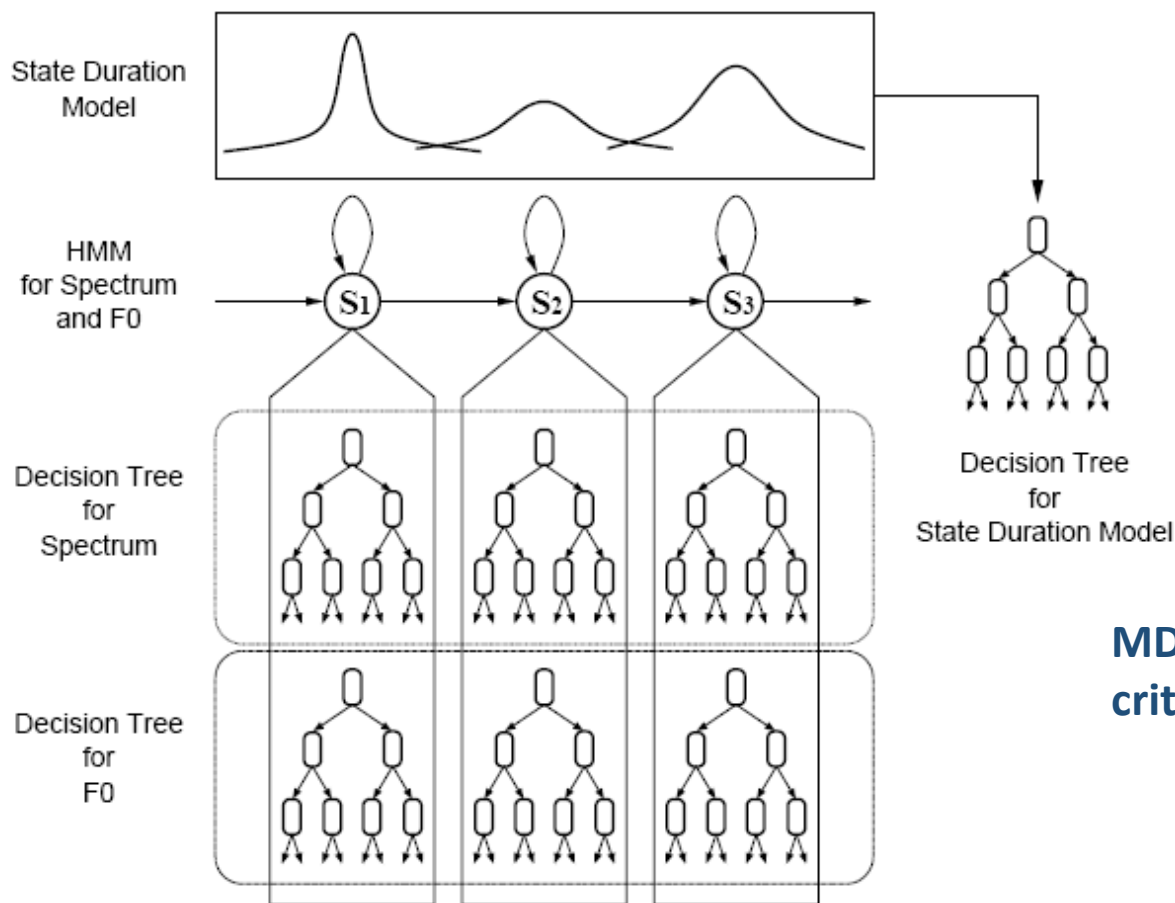
# MSD-HMM for F0 Modeling<sup>[3]</sup>

- A schematic representation of using MSD-HMM for f0 modeling



# Context Clustering<sup>[4,5]</sup>

- Each state of model has three trees: trees of duration, spectrum and pitch



# An Example of Decision Trees

Question set

C=fricative?

C=Vowel?

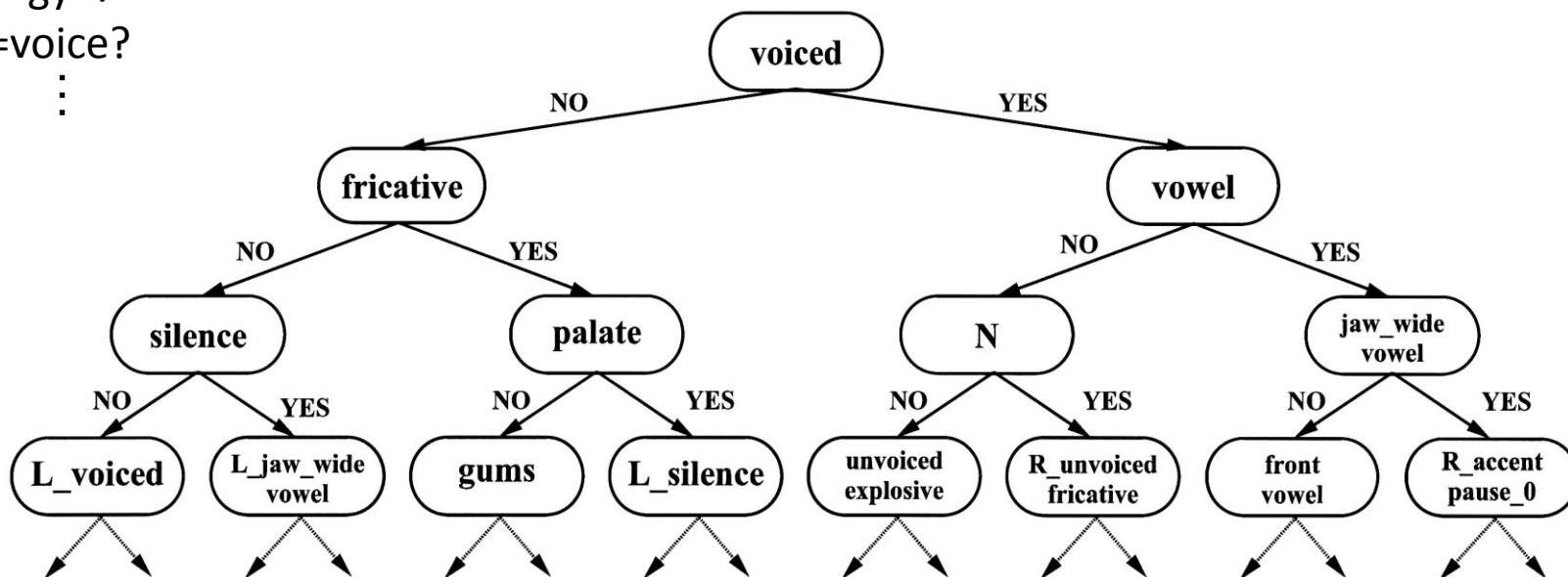
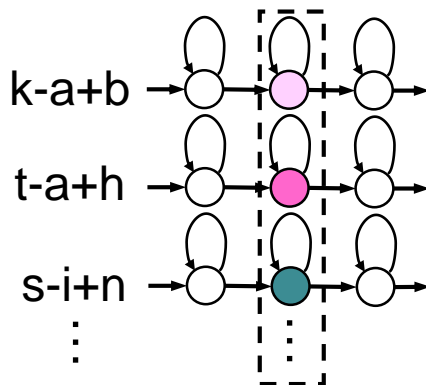
R=silence?

R="cl"?

L="gy"?

L=voice?

⋮



# Parameter Generation<sup>[6]</sup>

Speech parameter generation from HMM with dynamic constraint

For a given HMM  $\lambda$ , determine a speech parameter vector sequence  $O = [o_1^T, o_2^T, o_3^T, \dots, o_T^T]^T$ ,  $o_t = [c_t^T, \Delta c_t^T, \Delta^2 c_t^T]^T$ , which maximizes:

$$P(O | \lambda) = \sum_{all\ Q} P(O, Q | \lambda)$$

If given  $Q$ , maximizes  $P(O | Q, \lambda)$  with respect to  $O = WC$

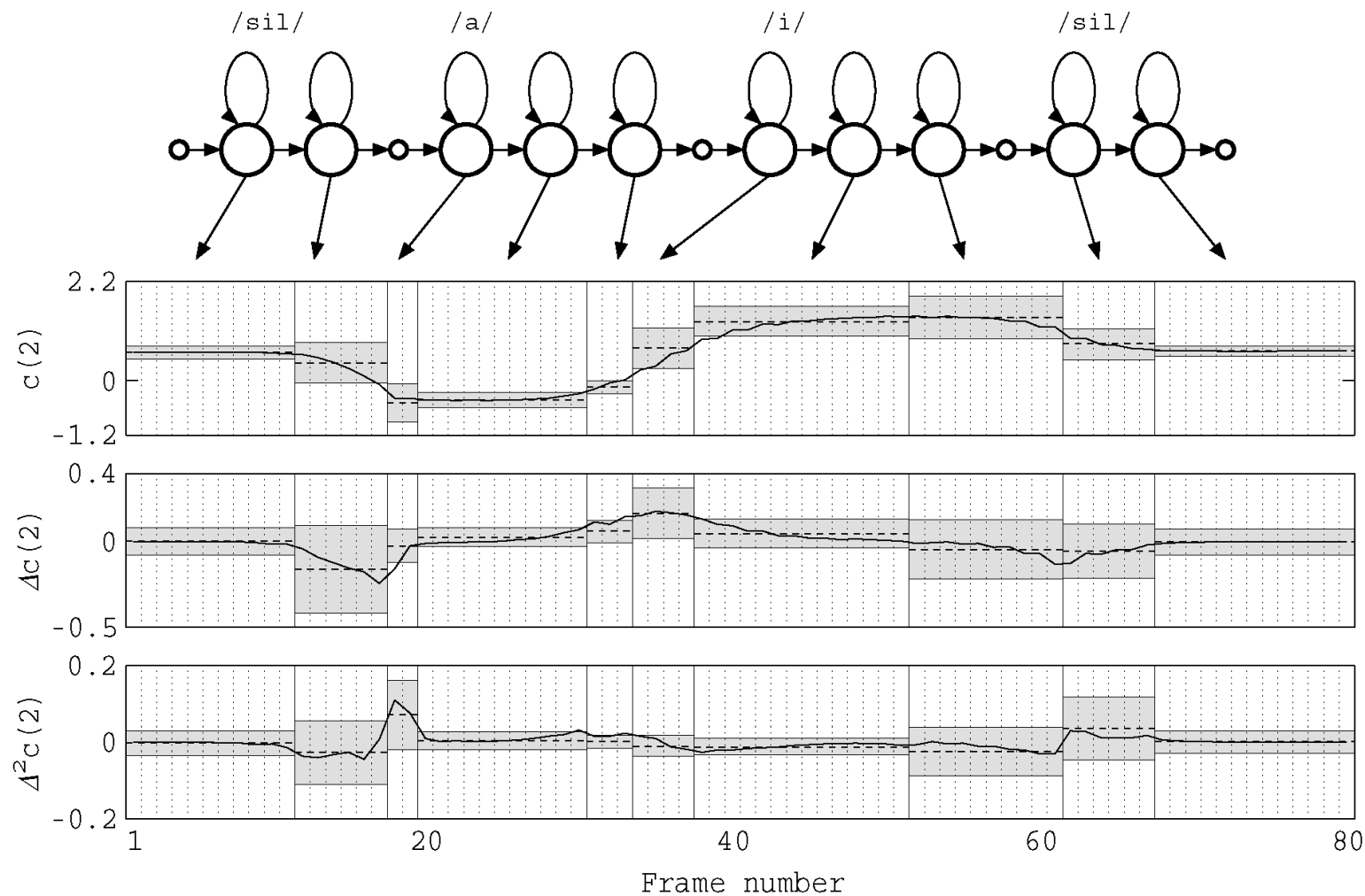
$$\log P(O | Q, \lambda) = -\frac{1}{2} O^T U^{-1} O + O^T U^{-1} M + K$$

$$\frac{\partial \log P(WC | Q, \lambda)}{\partial C} = 0$$

$$W^T U^{-1} W C = W^T U^{-1} M^T$$

$$\begin{pmatrix} c_1 \\ \Delta c_1 \\ \Delta^2 c_1 \\ \vdots \\ c_T \\ \Delta c_T \\ \Delta^2 c_T \end{pmatrix} = \begin{pmatrix} \begin{matrix} 1 & 0 & 0 & \cdots \\ 0 & \frac{1}{2} & 0 & \cdots \\ 2 & -1 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \end{matrix} \\ \vdots \\ \begin{matrix} \cdots & 0 & 0 & 1 \\ \cdots & 0 & -\frac{1}{2} & 0 \\ \cdots & 0 & -1 & 2 \end{matrix} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_T \end{pmatrix}$$

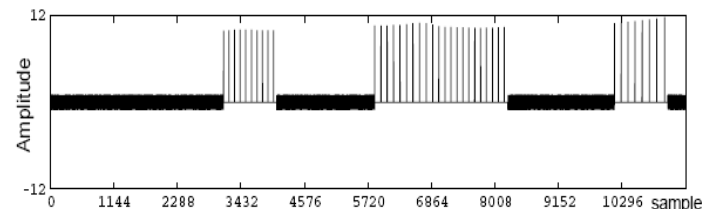
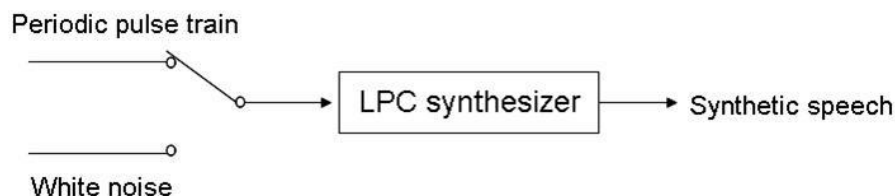
# Generated Speech Parameter Trajectory [7]



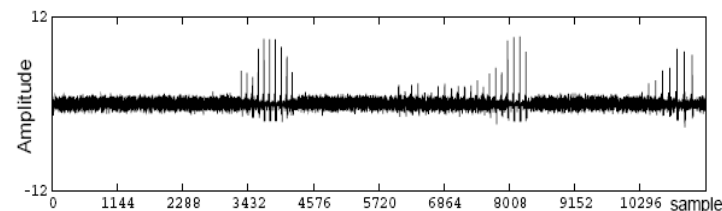
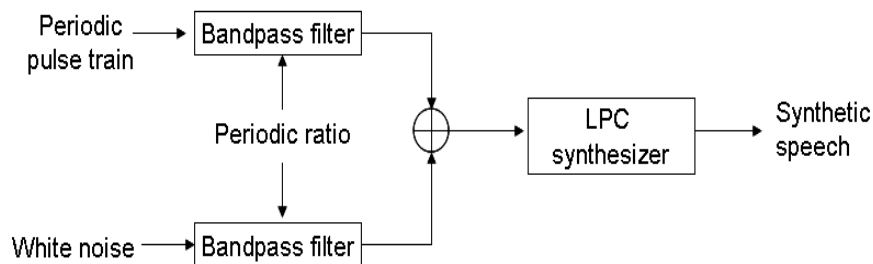


# Mixed Excitation [8,9]

- Traditional excitation model (hiss-buzz model)



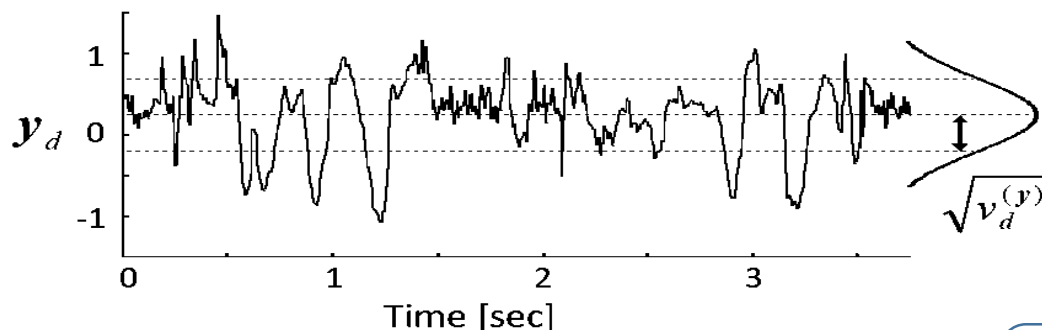
- A better excitation model (mixed excitation model)



# Global Variance (GV) <sup>[10]</sup>

- “Global variance” of features over an utterance

GV of  $d^{\text{th}}$  coefficient: 
$$v_d^{(y)} = \frac{1}{T} \sum_{t=1}^T \left( y_{t,d} - \frac{1}{T} \sum_{\tau=1}^T y_{\tau,d} \right)^2$$



- Generation

$$L = \log \{ p(\mathbf{O}|\mathbf{Q}, \boldsymbol{\lambda})^\omega \cdot p(\mathbf{v}(\mathbf{C})|\boldsymbol{\lambda}_v) \}$$

A Single Gaussian  
with  $\mu_v$  and  $\Sigma_v$

# Minimum Generation Error Training <sup>[11]</sup>

- **Training model** by maximizing the likelihood of training data

$$\lambda^* = \arg \max_{\lambda} \sum_{all Q} P(O, Q / \lambda) \quad w.r.t \quad O = WC$$

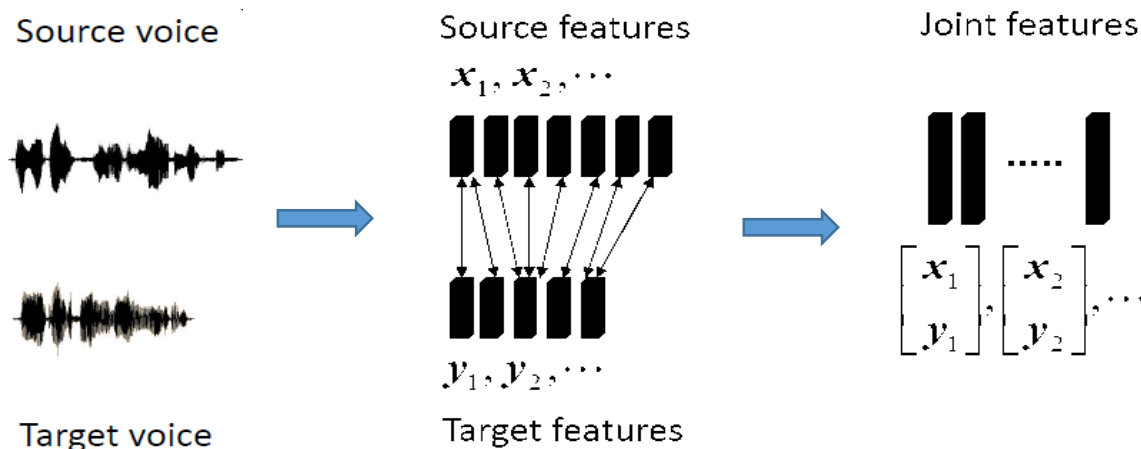
- **Refining model** by minimizing weighted generation error

$$\lambda^{i+1} = \arg \min_{\lambda} \sum_{all Q} P(Q / \lambda^i, O) D(C, \tilde{C})$$

# GMM-based Voice Conversion [12,15]

- Modeling joint feature vector (source  $\mathbf{x}_t$  and target  $\mathbf{y}_t$  speakers) by GMM

$$p(\mathbf{x}_t, \mathbf{y}_t | \lambda) = \sum_{m=1}^M \alpha_m \mathcal{N} \left( \begin{bmatrix} \mathbf{x}_t \\ \mathbf{y}_t \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_m^{(x)} \\ \boldsymbol{\mu}_m^{(y)} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_m^{(xx)} & \boldsymbol{\Sigma}_m^{(xy)} \\ \boldsymbol{\Sigma}_m^{(yx)} & \boldsymbol{\Sigma}_m^{(yy)} \end{bmatrix} \right)$$



# GMM-based Voice Conversion [12,13]

- Maximum likelihood conversion

$$p(\mathbf{y}_t | \mathbf{x}_t, \lambda) = \sum_{m=1}^M p(m | \mathbf{x}_t, \lambda) \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_{m,t}^{(y|x)}, \boldsymbol{\Sigma}_m^{(y|x)})$$

$$p(m | \mathbf{x}_t, \lambda) = \frac{\alpha_m \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_m^{(x)}, \boldsymbol{\Sigma}_m^{(xx)})}{\sum_{n=1}^M \alpha_n \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_n^{(x)}, \boldsymbol{\Sigma}_n^{(xx)})}$$

$$\begin{aligned} \boldsymbol{\mu}_{m,t}^{(y|x)} &= \boldsymbol{\mu}_m^{(y)} + \boldsymbol{\Sigma}_m^{(yx)} \boldsymbol{\Sigma}_m^{(xx)^{-1}} (\mathbf{x}_t - \boldsymbol{\mu}_m^{(x)}) \\ &= \mathbf{A}_m^{(y|x)} \mathbf{x}_t + \mathbf{b}_m^{(y|x)} \end{aligned}$$

$$\boldsymbol{\Sigma}_m^{(y|x)} = \boldsymbol{\Sigma}_m^{(yy)} - \boldsymbol{\Sigma}_m^{(yx)} \boldsymbol{\Sigma}_m^{(xx)^{-1}} \boldsymbol{\Sigma}_m^{(xy)}$$

# Limitations

- Vocoder
  - Synthesized voice with some traditional vocoder flavor
- Acoustic modeling
  - “Wrong” model assumptions out of necessity, e.g., GMM, diagonal covariance
  - Greedy, hence suboptimal, search derived decision tree state clustering (tying)

# Outline

- Statistical parametric speech generation and synthesis
  - HMM-based speech synthesis
  - GMM-based voice conversion
- Deep learning
  - RBM, DBN, DNN, MDN and RNN
- Deep learning for speech generation and synthesis
  - Approaches to speech synthesis
  - Approaches to voice conversion
- Conclusions and future work

# Deep Learning<sup>[16]</sup>

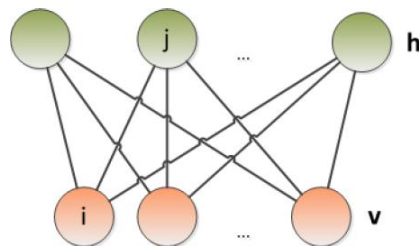
- A machine learning algorithm to learn high-level abstractions in data by using multiple-layered models
  - Typical neural nets with 3 or more hidden layers
- Motivated by human brain, which organize ideas and concepts hierarchically
- Difficulties to train DNN in 1980s
  - Training slowly , vanishing gradients (RNN)



# Deep Learning<sup>[16]</sup>

- Improving DNN training significantly since 2006
  - GPU
  - Big data
  - Pre-training
- Different architectures, e.g. DNN, CNN, RNN and DBN, for computer vision, speech recognition and synthesis, and natural language processing

# Restricted Boltzmann Machine (RBM) <sup>[17]</sup>



**Restricted:** No interaction between hidden variables

No interaction between visible variables

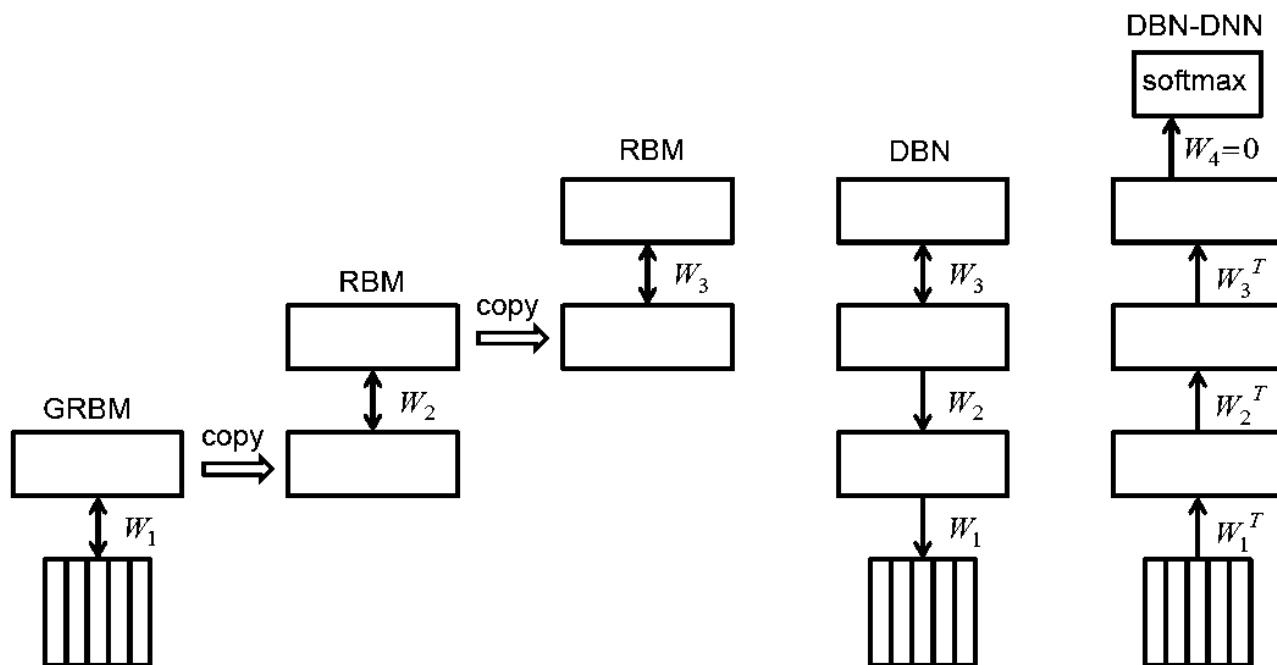
- Joint distribution  $p(\mathbf{v}, \mathbf{h} \mid \mathbf{W})$  defined in terms of an energy function  $E(\mathbf{v}, \mathbf{h}; \mathbf{W})$

$$p(\mathbf{v}, \mathbf{h} \mid \mathbf{W}) = \frac{1}{Z(\mathbf{W})} \exp \{-E(\mathbf{v}, \mathbf{h}; \mathbf{W})\}$$
$$E(\mathbf{v}, \mathbf{h}; \mathbf{W}) = - \sum_i b_i v_i - \sum_j c_j h_j - \sum_{i,j} v_i w_{ij} h_j$$

- Parameters can be estimated by contrastive divergence learning <sup>[18]</sup>

# Deep Belief Network (DBN) <sup>[17]</sup>

- Each successive pair of layers is treated as an RBM
- Stack RBMs to form a DBN



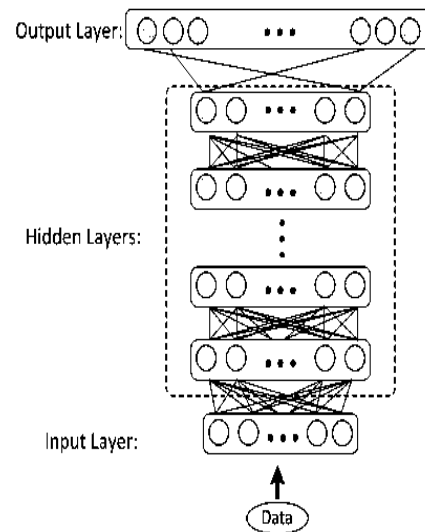
# Deep Neural Net (DNN)

- A feed-forward, artificial neural networks with multiple hidden layers

$$y_j = f(x_j)$$

where

$$x_j = b_j + \sum_i y_i w_{ij}$$



- A nonlinear activation function

$$f(x_j) = \frac{1}{1 + e^{-x_j}}$$

Sigmoid function

$$f(x_j) = \frac{e^{x_j} - e^{-x_j}}{e^{x_j} + e^{-x_j}}$$

Hyperbolic tangent  
(or tanh) function

$$f(x_j) = \max(x_j, 0)$$

Rectified linear  
function

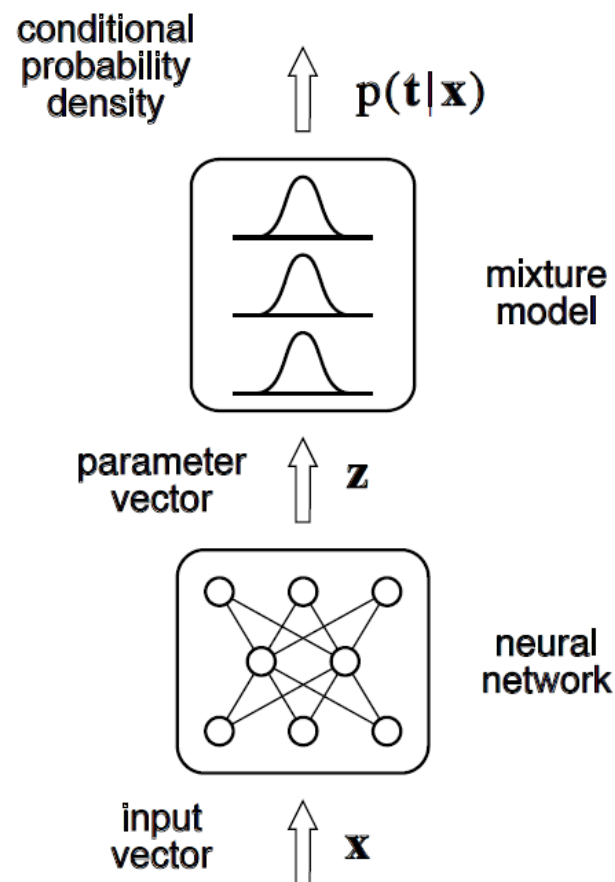
- Train discriminative models via back-propagation (BP) <sup>[19]</sup> with an appropriate cost function
- A “mini-batch” based stochastic gradient descent algorithm
- Improved acoustic model for enhancing speech recognition performance significantly <sup>[17]</sup>

# DNN Pre-training

- Initialize the weights to a better starting point than random initialization, prior to BP
  - DBN<sup>[17]</sup>
    - Layer-by-layer trained RBMs
    - Unsupervised training via Contrastive Divergence
  - Layer-wise BP<sup>[20]</sup>
    - Add one hidden layer after another
    - Supervised training with BP

# Mixture Density Networks (MDN) <sup>[21]</sup>

- MDN combines a conventional neural network with a mixture density model
- Conventional neural network can represent arbitrary functions
- In principle, MDN can represent arbitrary conditional probability distributions



# Mixture Density Networks (MDN) [21]

- Output is a mixture density of target
  - Make neural network a generative model rather than regression model
- Activation function
  - Sigmoid or hyperbolic tangent for hidden nodes
  - Softmax for mixture weights, exponential for variance and identity for mean
- Criterion
  - Maximum likelihood (MLE)

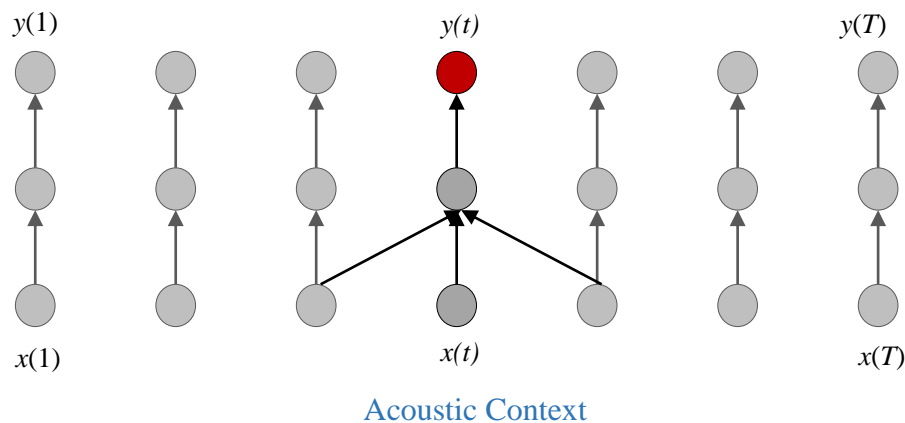


# Recurrent Neural Network (RNN) [22]

$$h_t = \mathcal{H}(\mathbf{W}_{xh}x_t + b_h)$$

$$y_t = h_t$$

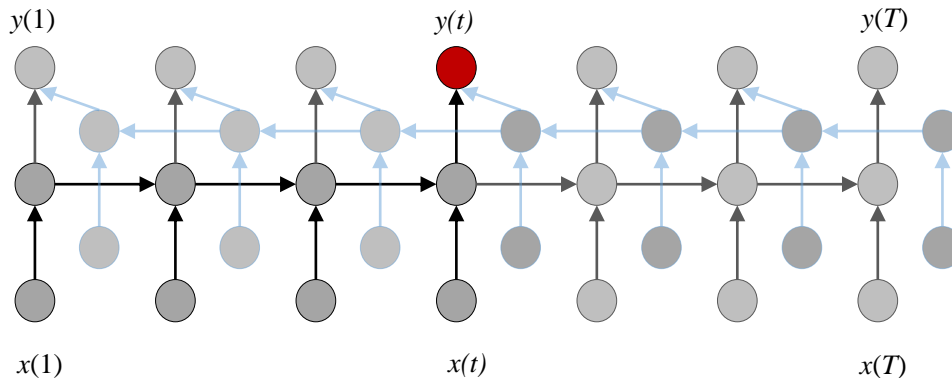
**Feed Forward Neural Network**



**Recurrent Neural Network**

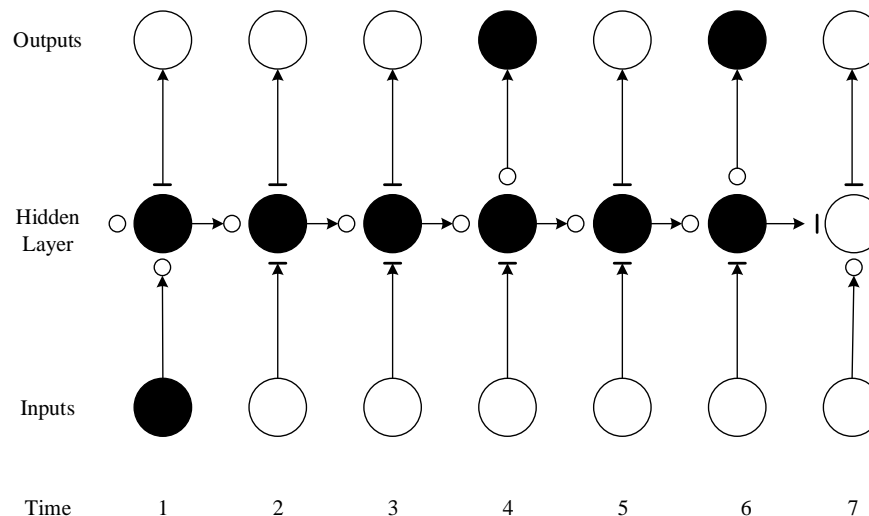
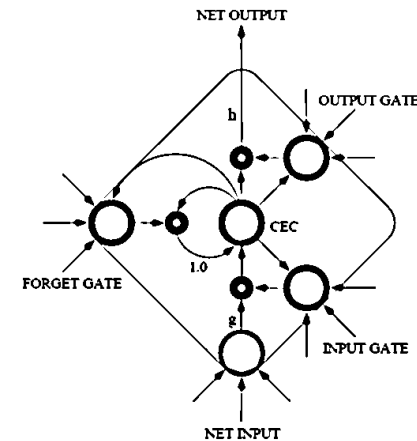
$$h_t = \mathcal{H}(\mathbf{W}_{xh}x_t + \mathbf{W}_{hh}h_{t-1} + b_h)$$

$$y_t = \mathbf{W}_{hy}h_t + b_y$$

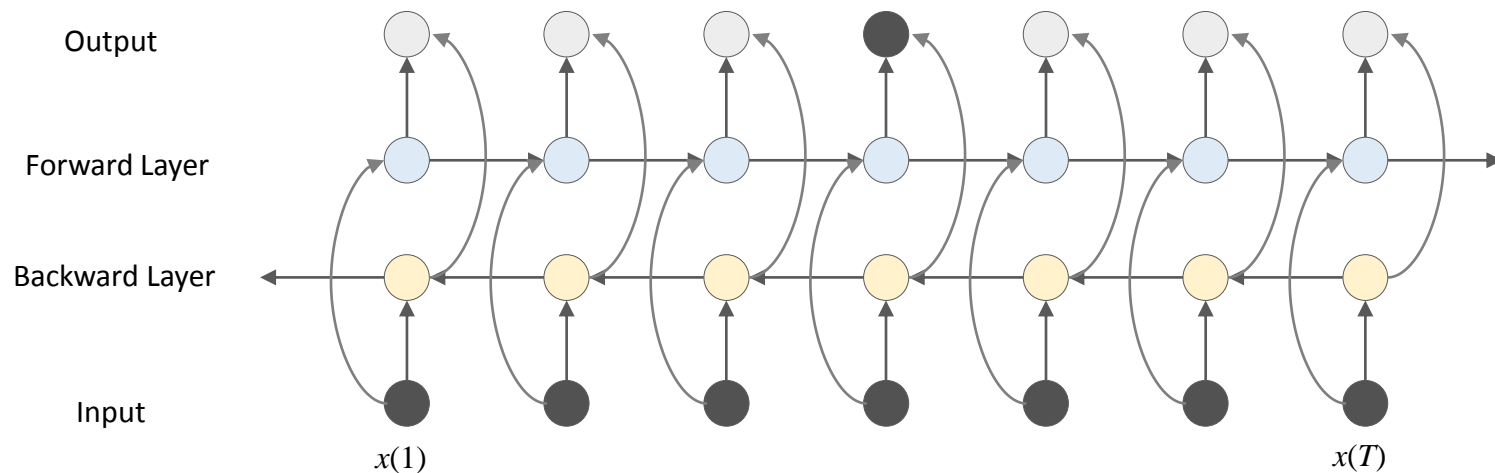


# Long Short Term Memory (LSTM)<sup>[22,23]</sup>

$$\begin{aligned}
 i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
 c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
 o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
 h_t &= o_t \tanh(c_t)
 \end{aligned}$$



# Bidirectional RNN<sup>[22,24]</sup>



$$\vec{h}_t = \mathcal{H}(\mathbf{W}_{x\vec{h}}x_t + \mathbf{W}_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}})$$

$$x(t) \leftarrow x(1), x(2), \dots, x(t)$$

Forward direction: past and current

$$\overleftarrow{h}_t = \mathcal{H}(\mathbf{W}_{x\overleftarrow{h}}x_t + \mathbf{W}_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}})$$

$$x(t) \leftarrow x(T), x(T-1), \dots, x(t)$$

Backward direction: future and current

$$y_t = \mathbf{W}_{\vec{h}y}\vec{h}_t + \mathbf{W}_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y$$

$$x(t) \leftarrow x(1), x(2), \dots, x(t), \dots, x(T-1), x(T)$$

Bidirection: all

# Outline

- Statistical parametric speech generation and synthesis
  - HMM-based speech synthesis
  - GMM-based voice conversion
- Deep learning
  - RBM, DBN, DNN, MDN and RNN
- Deep learning for speech generation and synthesis
  - Approaches to speech synthesis
  - Approaches to voice conversion
- Conclusions and future work

# Deep Learning for Speech Generation and Synthesis

- Motivated by the DNN's superior performance in ASR, we investigate potential advantages of NN in speech synthesis and voice conversion
- Model long-span, high dimensional and correlated features as its input
  - MCEP or LSP  $\rightarrow$  Spectral envelop, 1 frame  $\rightarrow$  multi frames
- Non-linear mapping between input and output features with a deep-layered , hierarchical structure
  - Representing highly-variable mapping function compactly
  - Simulating human speech production system

# Deep Learning for Speech Generation and Synthesis

- Distributed representation
  - Generating observed data by the interactions of many different factors on different levels
  - Replacing HMM states and decision tree, which decompose the training data into small partitions
- “Discriminative” and “predictive” in generation sense, with appropriate cost function(s), e.g. generation error
  - Training criteria is closer to the objectives than ML

# Approaches of Deep Learning to Speech Synthesis

- RBM/DBN
  - CUHK <sup>[25]</sup>, USTC/Microsoft <sup>[26]</sup>
- DNN
  - Google <sup>[27]</sup>, CSTR<sup>[28]</sup>, Microsoft <sup>[29]</sup>
- DBN/DNN for feature transformation
  - IBM <sup>[30]</sup>
- MDN
  - Google <sup>[31]</sup>
- RNN with BLSTM
  - Microsoft <sup>[32]</sup>, IBM <sup>[33]</sup>

# Generative models: RBM vs. GMM [34]

- GMM

$$p(\mathbf{v}) = \sum_{i=1}^M m_i N(\mathbf{u}_i, \Sigma_i)$$

- Mixture models can't generate distributions sharper than the individual components
- Need lots of data to estimate parameters

- RBM

$$p(\mathbf{v}) \propto e^{-\frac{1}{2}(\mathbf{v}-\mathbf{b})^T(\mathbf{v}-\mathbf{b})} \prod_j (1 + e^{c_j + \mathbf{v}^T \mathbf{W}_{*,j}})$$

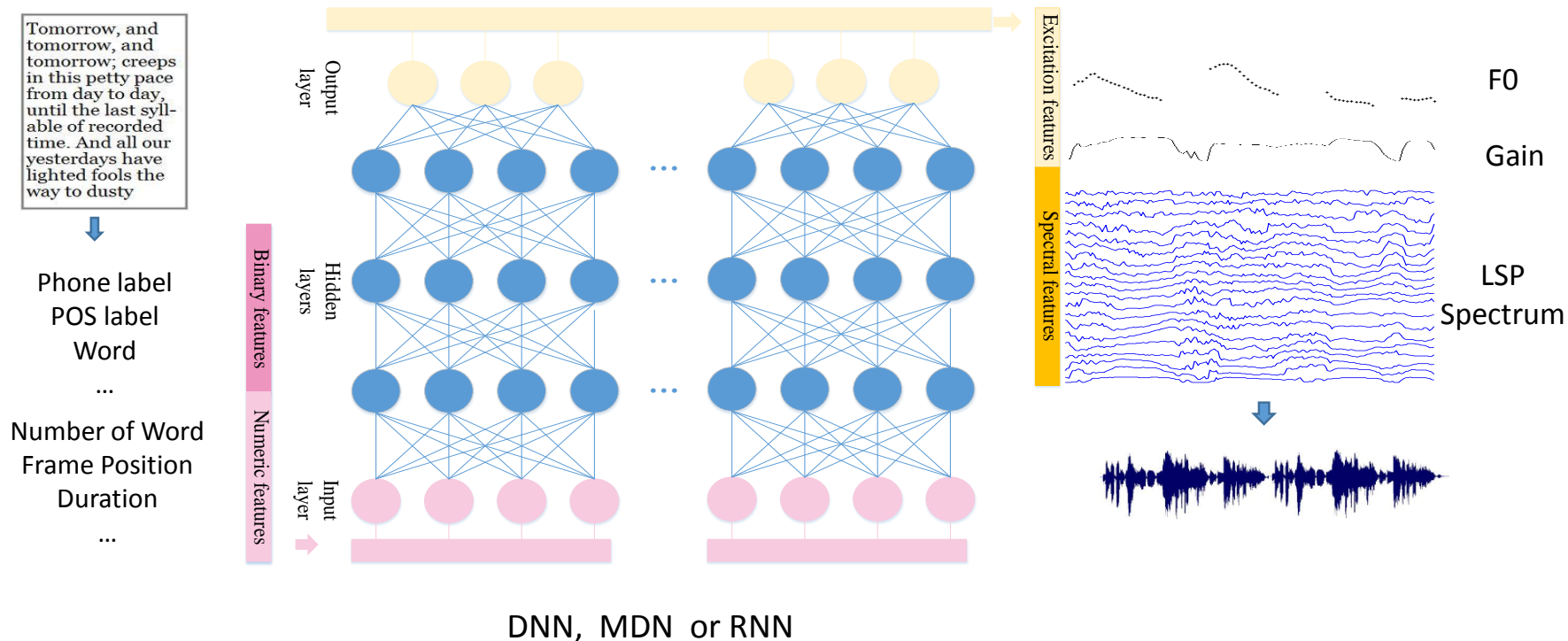
- Product models can generate distributions sharper than the individual components
- Need less data to estimate parameters



# DBN for Speech Synthesis [25,26]

- Modeling joint distribution of linguistic and acoustic features
  - Fully utilize generative nature to generate speech parameter from DBN
- DBNs replaces GMMs for the distribution of the spectral envelopes at each decision tree-clustered HMM state
  - The estimated model of spectral RBM has much sharper formant structure than Gaussian mean

# Deep Learning (DNN, MDN, RNN) for Speech Synthesis



# DNN for TTS Training

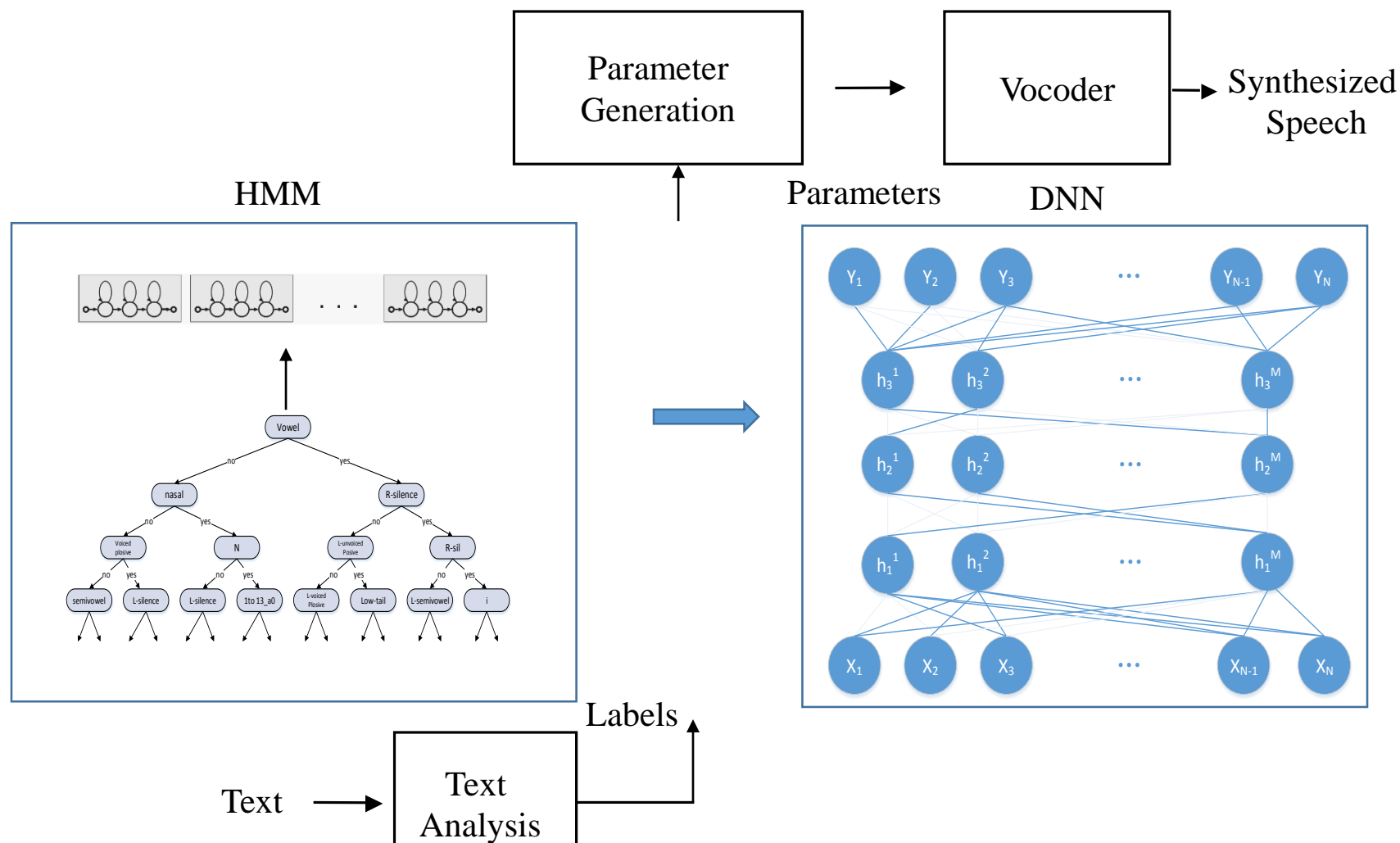
- Input features  $\mathbf{x}$  and output features  $\mathbf{y}$  are time-aligned frame-by-frame by well-trained HMM models
- L2 cost function with Back-Propagation (BP)

$$C = \frac{1}{2T} \sum_{t=1}^T \|f(x^{(t)}) - y^{(t)}\|^2 + \frac{\lambda}{2} \sum_{i,j,l} (w_{i,j}^l)^2$$

- A “mini-batch” based stochastic gradient descent algorithm

$$(W^l, b^l) \leftarrow (W^l, b^l) + \varepsilon \frac{\partial C}{\partial (W^l, b^l)} , \quad 0 \leq l \leq L$$

# DNN vs. HMM for TTS Synthesis



# Deep MDN (DMDN) based Speech Synthesis<sup>[31]</sup>

- MDN, In principle, can represent arbitrary conditional probability distributions
- To address the limitations of DNN
  - Unimodal nature of its objective function
  - Lack of ability to predict variances

# DMDN for TTS training<sup>[31]</sup>

- Training Criterion

$$p(y | x, \mathcal{M}) = \sum_{m=1}^M w_m(x) \cdot \mathcal{N}(y; \mu_m(x), \sigma_m^2(x))$$

$$w_m(x) = \frac{\exp(z_m^{(w)}(x, \mathcal{M}))}{\sum_{l=1}^M \exp(z_l^{(w)}(x, \mathcal{M}))}$$

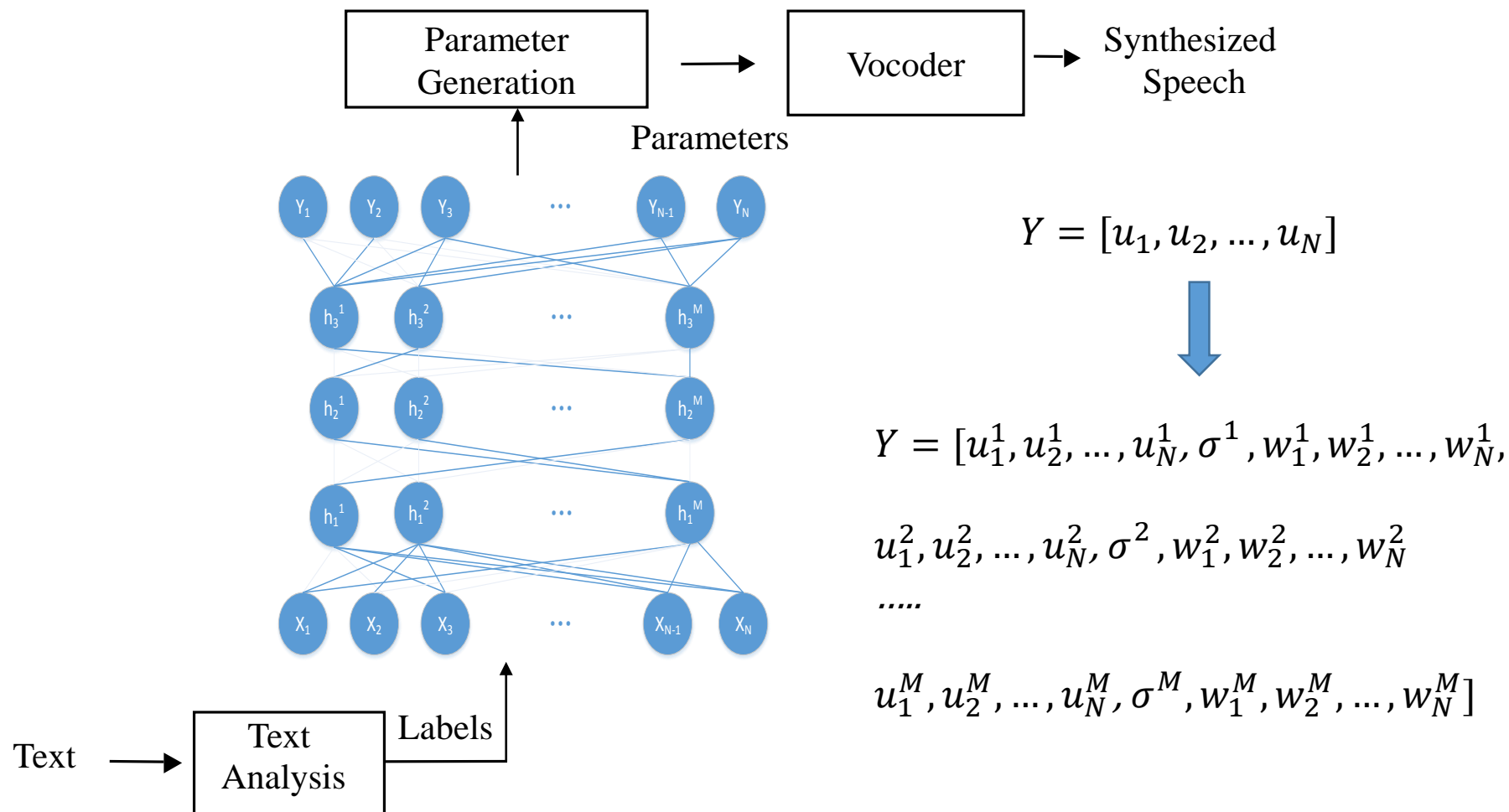
$$\sigma_m(x) = \exp(z_m^{(\sigma)}(x, \mathcal{M}))$$

$$\mu_m(x) = z_m^{(\mu)}(x, \mathcal{M})$$

- Consistent with generation algorithm

$$p(y | x, \lambda) = p(y | Q, \lambda)$$

# DMDN for TTS Synthesis [31]



# DBLSTM-RNN based Speech Synthesis

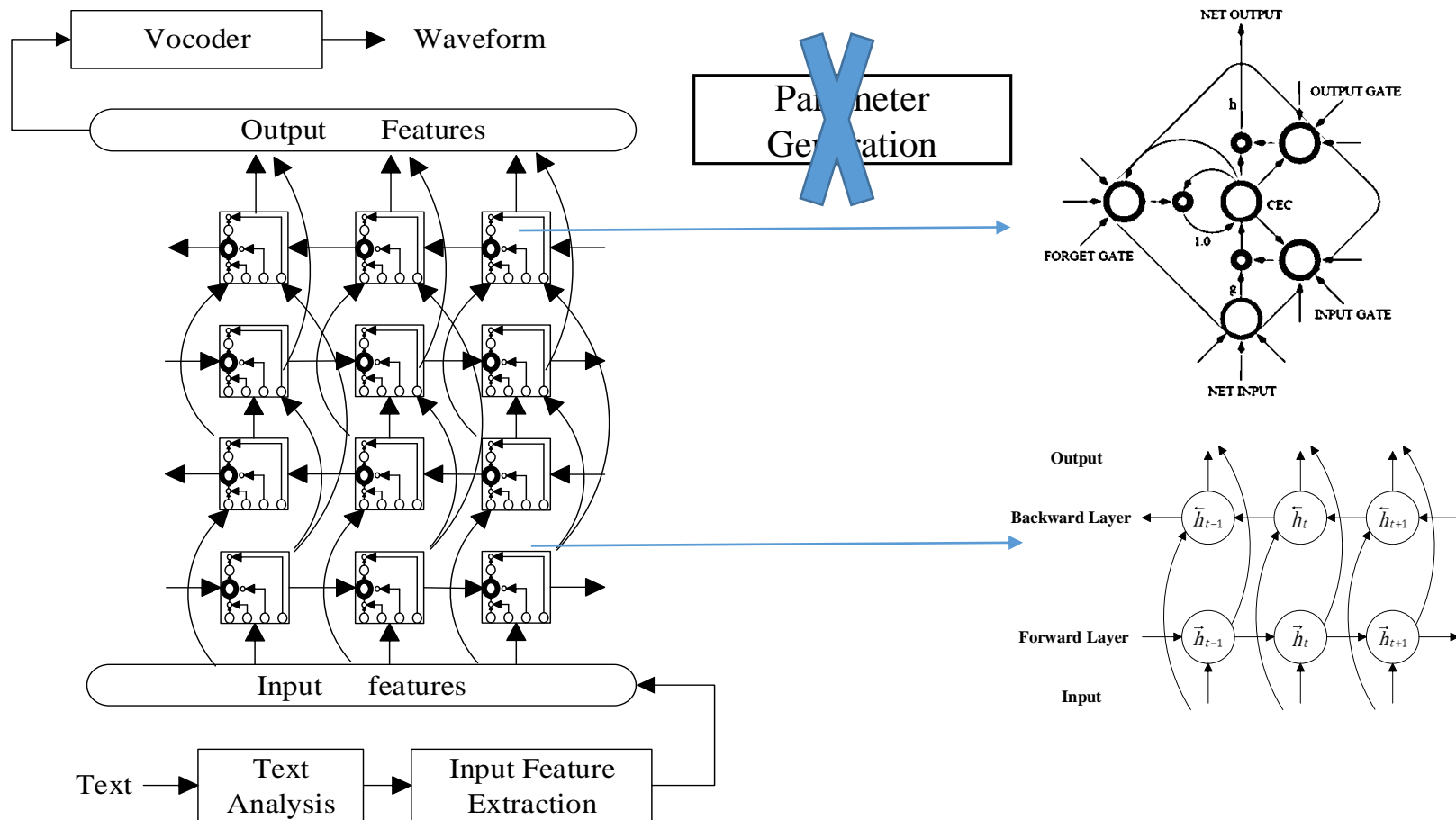
- Speech production is a continuous dynamic process
  - Consider semantic and syntactic info, select words, formulate phonetics, articulate sounds
  - These features interact with complex contextual effects from neighboring and distant input
- RNN with LSTM, in principle, can capture information from anywhere in the feature sequence
- Bi-directional, to access long-range context in both forward and backward directions
  - A whole sentence is given as input



# DBLSTM-RNN for TTS training

- Training criterion
  - L2 cost function
- Back-propagation through time (BPTT)
  - Unfold the RNN into feed-forward network through time
  - Train the unfolded network with back-propagation
- Sequence mapping between input and output
  - The weight gradients are computed over the entire utterance
  - Utterance-level randomization
  - Tens of utterances for each “mini-batch”

# DBLSTM-RNN for TTS synthesis



# DBLSTM-RNN based Speech Synthesis

- Pros
  - Bidirectional, deep-in-time and deep-in-layer structure
  - Embedded sequence training
  - Matched criteria for training and testing
- Cons
  - High computational complexity
  - Slow to converge

# Experimental Setup

Corpus	U.S. English utterances (female, general domain)
Training/ Test set	5,000/200 sentences
Linguistic features	319 (binary), e.g. phone, POS 36 (numerical), e.g. word position
Acoustic features	1 U/V, LSP + gain (40+1), F0, $\Delta$ , $\Delta^2$
HMM topology	5-state, left-to-right HMM, MSD F0, MDL, MGE
DNN, MDN, RNN* architecture	1 - 6 layers, 512 or 1024 nodes/layer Sigmoid/Tanh, continuous F0
Post-processing	formant sharpening of LSP frequencies

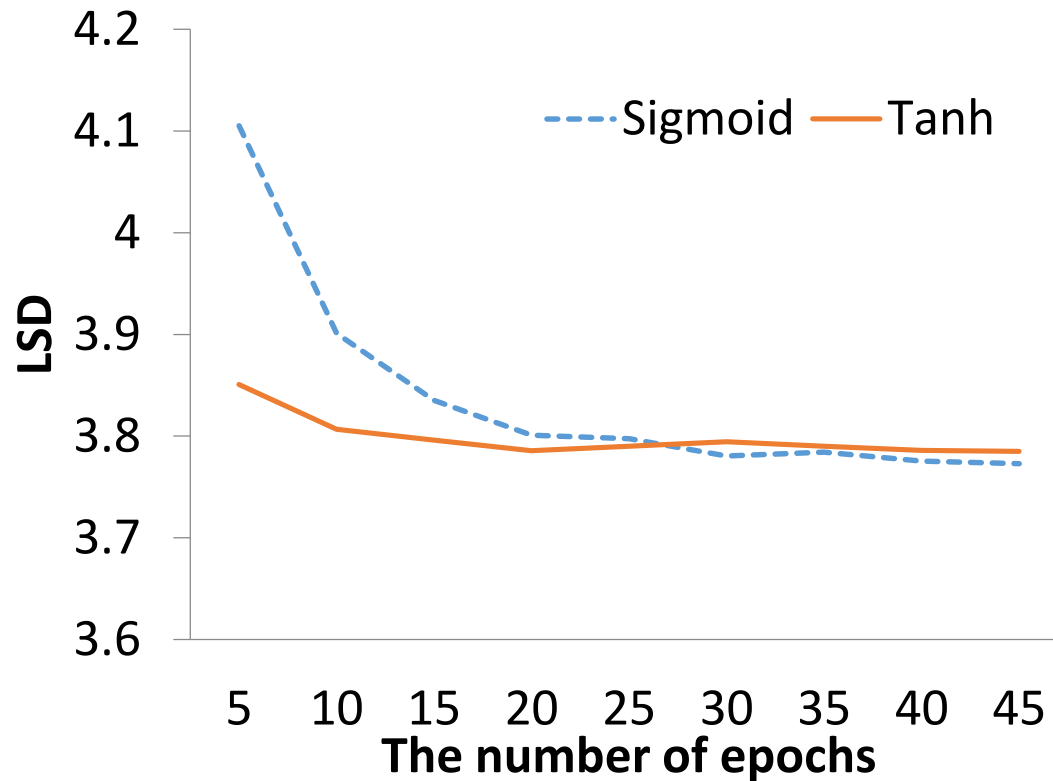
\* CURRENT <sup>[40]</sup>, Open source code under GPLv3

# Evaluation Metrics

- Objective measures
  - Log Spectral Distance (LSD)
  - Voiced/Unvoiced error rate
  - Root Mean Squared Error (RMSE) of F0
- Subjective measure
  - AB preference test
  - Crowdsourcing (20 subjects, 50 pairs of sentences for each)

# Experimental Results

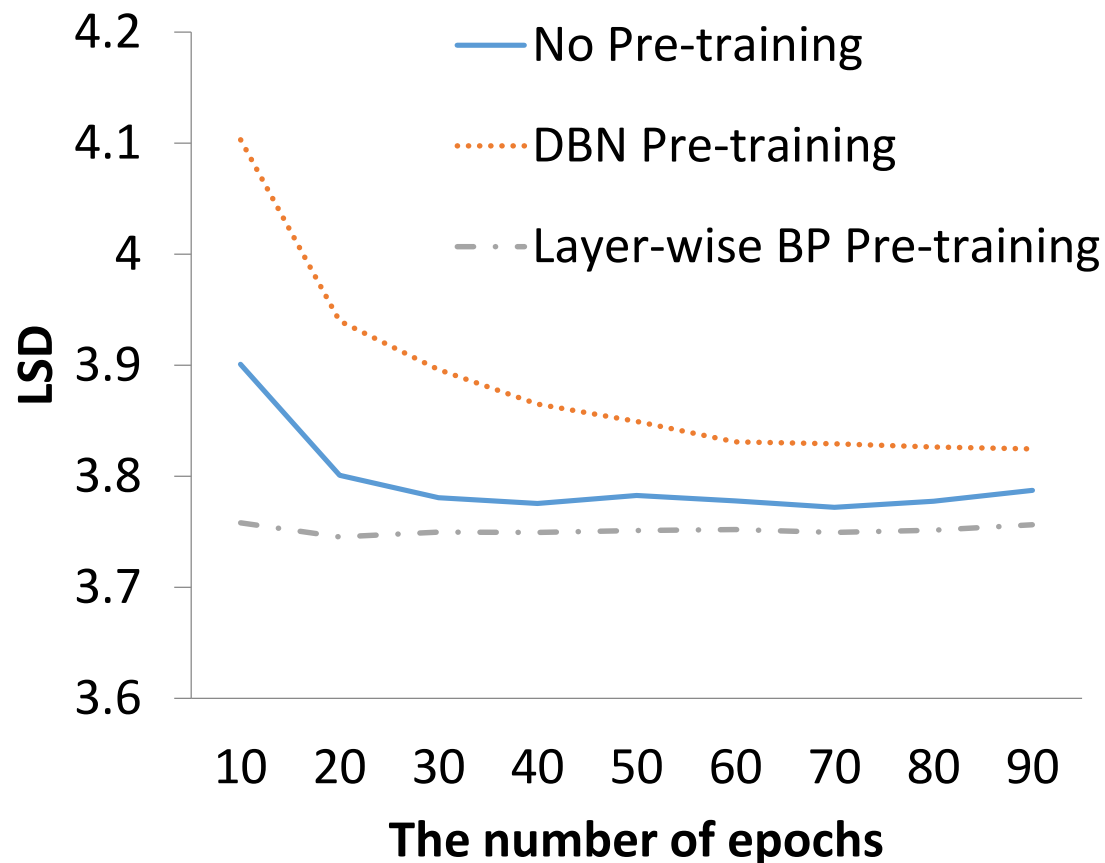
- Activation Functions (Sigmoid vs. Tanh)



\* Rectifier linear activation function is better in objective measures <sup>[31]</sup>

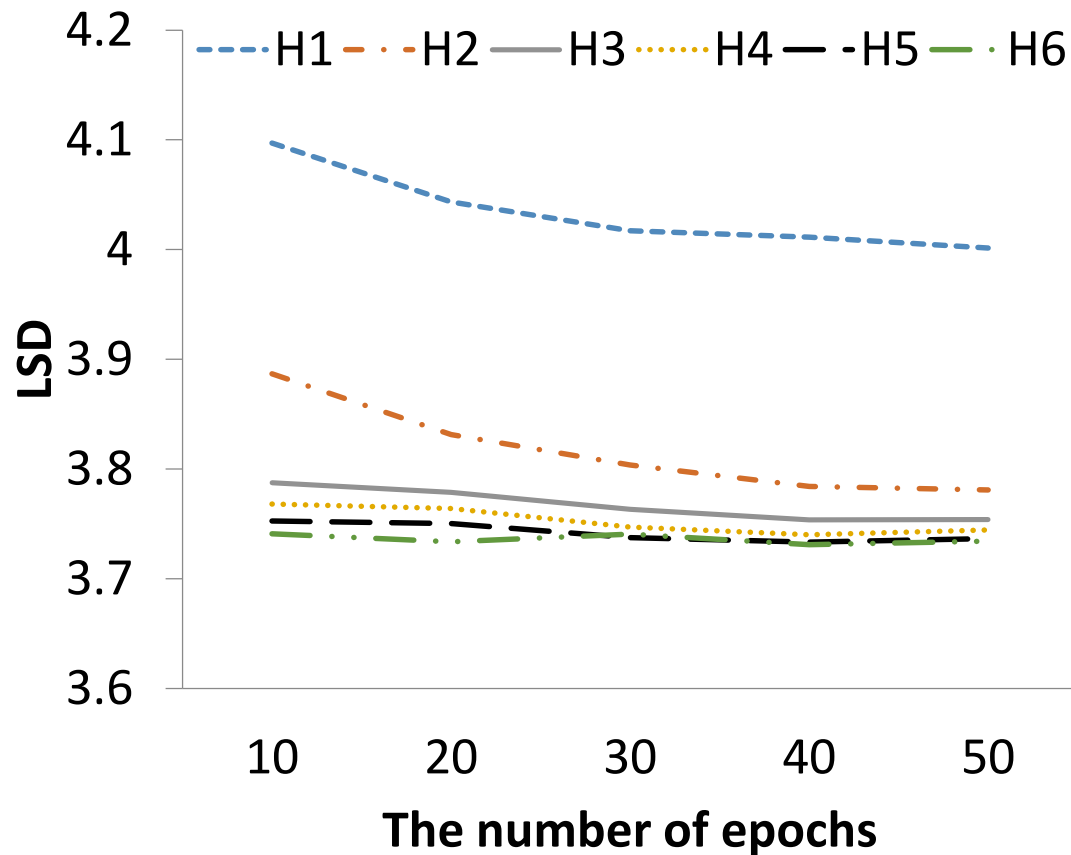
# Experimental Results

## ■ Pre-training



# Experimental Results

- The Number of Hidden Layers





# Evaluation Results (DNN)

Measure DNN Structure	LSD (dB)	V/U Error rate (%)	F0 RMSE (Hz)
512 *3 (0.77 M)	3.76	5.9	15.8
512 *6 (1. 55M)	3.73	5.8	15.8
1024*3 (2.59 M)	3.73	5.9	15.9

Measure HMM MDL Factor	LSD (dB)	V/U Error rate (%)	F0 RMSE (Hz)
1 (2.89 M)	3.74	5.8	17.7
1.6 (1.52M)	3.85	6.1	18.1
3 (0.85M)	3.91	6.2	18.4

46% HMM (MDL=1)	10% Neutral	44% DNN (512*3)	$P=0.45$
--------------------	----------------	--------------------	----------

23% HMM (MDL=1)	10% Neutral	67% DNN (1024*3)	$P<0.001$
--------------------	----------------	---------------------	-----------

# Evaluation Results (DNN)

- Alignment boundaries: state vs. phone

Measure DNN 1024 *3	LSD (dB)	V/U Error rate (%)	F0 RMSE (Hz)
State'	3.73	5.9	15.9
State*	3.63	5.9	15.4
Phone*	3.81	5.8	15.7

36% State *	31% Neutral	33% Phone*	$p=0.549$
----------------	----------------	---------------	-----------

State': 5 dimensional binary features, which indicate the current frame belonging to the state position of current phone

State\*: State' + 1 dim for state duration +1 dim for frame index (fraction 0~1)

Phone\*: 1 dim for phone duration +1 dim for frame index (fraction 0~1)

# Evaluation Results (DMDN)

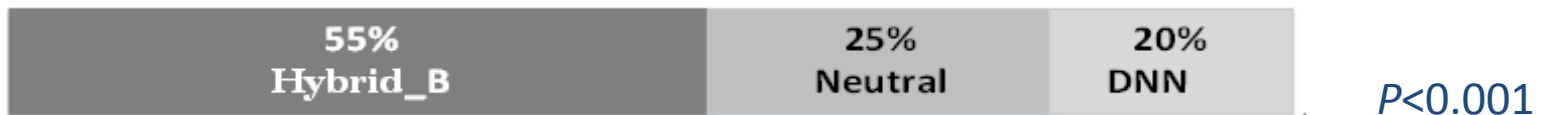
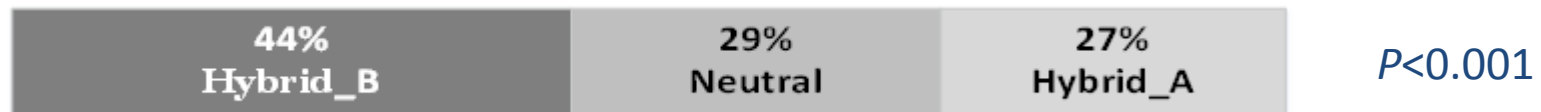
Measure 1024 *3 State*	LSD (dB)	V/U Error rate (%)	F0 RMSE (Hz)
MDN 1mix	3.70	5.9	15.8
MDN 4mix	3.74	5.9	15.5
DNN	3.63	5.9	15.4

38% MDN 1mix	26% Neutral	36% MDN 4mix	$p=0.350$
41% DNN	33% Neutral	26% MDN 4mix	$p<0.001$

- Much less data
- Very rich contextual information used in input feature vector

# Evaluation Results (DBLSTM-RNN)

Measure Model	LSD (dB)	V/U Error rate (%)	F0 RMSE (Hz)
DNN 1024*3 (2.59 M)	3.73	5.9%	15.9
Hybrid_A : 512*3_Sigmoid +512_BLSTM (2.30M)	3.61	5.7%	16.4
Hybrid_B : 512*2_Sigmoid +512*2_BLSTM (3.61M)	3.54	5.6%	15.8



# Evaluation Results (DBLSTM-RNN)

Measure 512*2_Sigmoid +512*2_BLSTM (3.61M)	LSD (dB)	V/U Error rate (%)	F0 RMSE (Hz)
No context (123 dim)	3.61	5.7%	16.1
Rich context (355 dim)	3.54	5.6%	15.8

33% No Context	32% Neutral	35% Rich context	$p=0.581$
-------------------	----------------	---------------------	-----------

\* DBLSTM-RNN uses 43 dims (U/V, log F0, LSP 40, gain) as output features instead of 127 dims ( with dynamic counterparts) used in DNN and MDN

# Evaluation Results (HMM vs. DNN vs. RNN)

Measure Structure	LSD (dB)	V/U Error rate (%)	F0 RMSE (Hz)
HMM MDL=1 (2.89 M)	3.74	5.8	17.7
DNN 1024*3 (2.59 M)	3.73	5.9	15.9
RNN (512*2_Sigmoid +512*2_BLSTM) (3.61 M)	3.54	5.6	15.8



# Demo

Hyperion, however, tumbles erratically as gravity from nearby moons tugs on its irregular shape.

They've all dried out; it's all carrot juice.

That's why Kathy could not change Ruby's behavior.

But to hear South African coach Kitch Christie talk, it's Lomu who should be worried.

When coaxing failed, the child's nose was plugged.

**HMM**    **DNN**    **RNN**



More information and the samples of synthesized speech

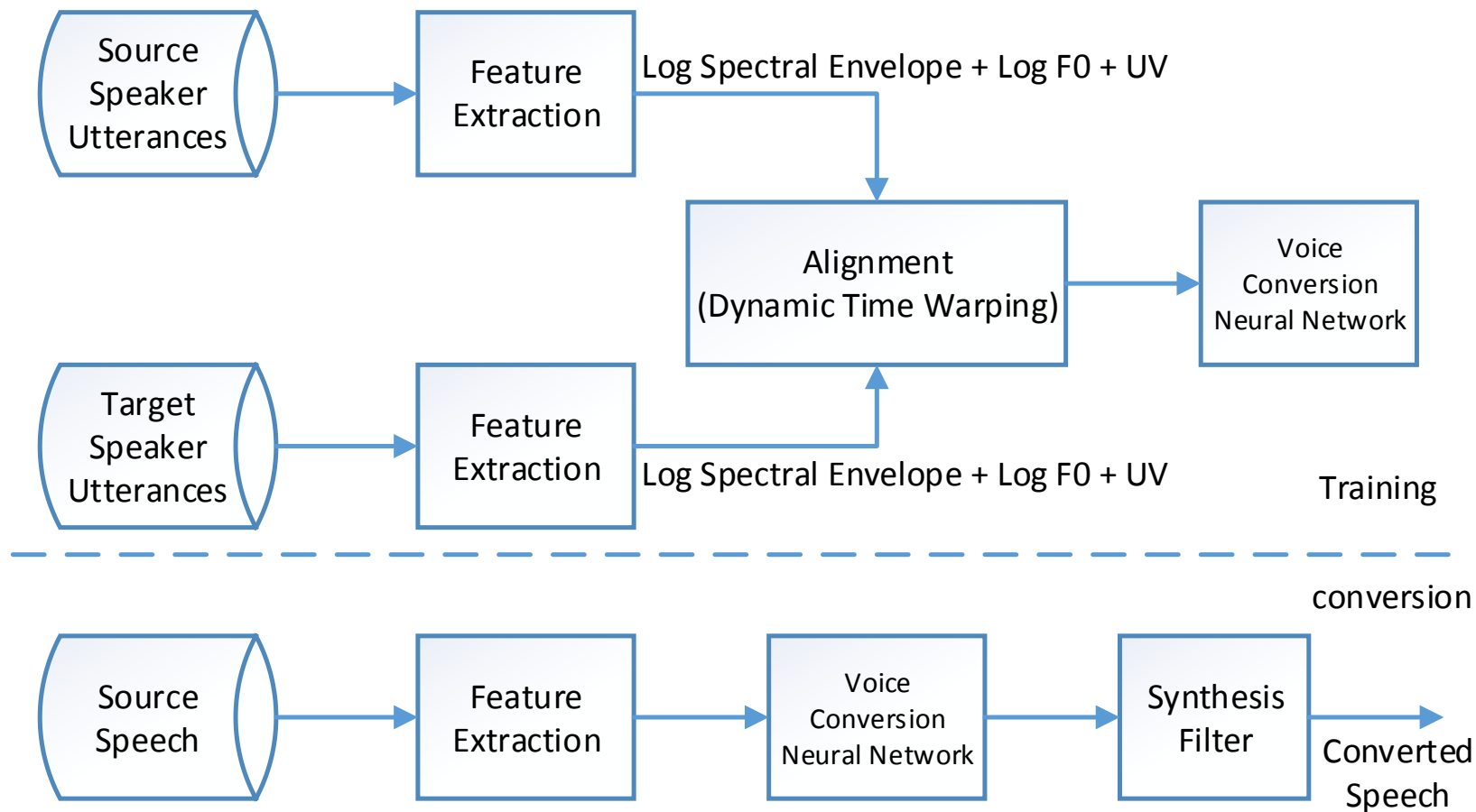
<http://research.microsoft.com/en-us/projects/dnntts/default.aspx>

# Neural Network Based Voice Conversion

- ANN
  - CMU <sup>[35]</sup>, Microsoft <sup>[36]</sup>
- RBM/Conditional RBM
  - USTC <sup>[37]</sup>, NTU <sup>[38]</sup>
- DBN
  - Kobe University <sup>[39]</sup>



# ANN for Voice Conversion



# ANN for Voice Conversion

- A multi-layer feed forward neural network for training the mapping functions between the source  $X$  and the target  $Y$  feature vectors.

$$\tilde{Y} = F(X) = f(h_{W,b}(X))$$

$$(W, b) = (W^1, b^1, W^2, b^2, W^3, b^3)$$

$$f(z) = \frac{1}{1 + e^{-z}}$$

- Nonlinear mapping function to simulate speech production/perception

# F0 Conversion

- Conventional method in GMM and NN based voice conversion [12,13,35,37]

$$f_t = \frac{f_t - u_s}{\sigma_s} \sigma_t + u_s$$



- Embedded F0 modeling in NN [41]
  - F0s are appended to spectral features in modeling
  - A longer window of observations

# Sequence Generation Error (SGE) Minimization [36]

- Incorporating parameter generation of the whole sequence (i.e., trajectory) into training

$$D(\mathbf{Y}, \tilde{\mathbf{Y}}) = \|\mathbf{Y} - \tilde{\mathbf{Y}}\|^2 = \frac{1}{T} \sum_{t=1}^T \|\mathbf{y}_t - \tilde{\mathbf{y}}_t\|^2$$

Frame error



$$\tilde{\mathbf{c}} = (\mathbf{W}^T \mathbf{U}^{-1} \mathbf{W})^{-1} \mathbf{W}^T \mathbf{U}^{-1} \tilde{\mathbf{y}}$$

$$D(\mathbf{c}, \tilde{\mathbf{c}}) = \|\mathbf{c} - \tilde{\mathbf{c}}\|^2 = \sum_{t=1}^T \|\mathbf{c}_t - \tilde{\mathbf{c}}_t\|^2$$

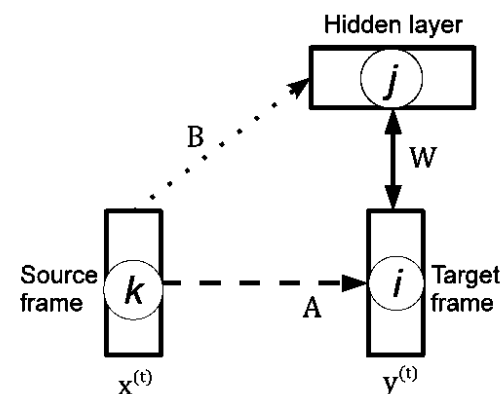
Sequence error

# Sequence Generation Error (SGE) Minimization [36]

- Training minimum sequence error
  - Mini-batched SGD is not suitable
- Performing randomization at the sentence level.
- Requiring more computational cost
  - Very hard to take the matrix inverse operation in parallel
  - The same global variance for each frame

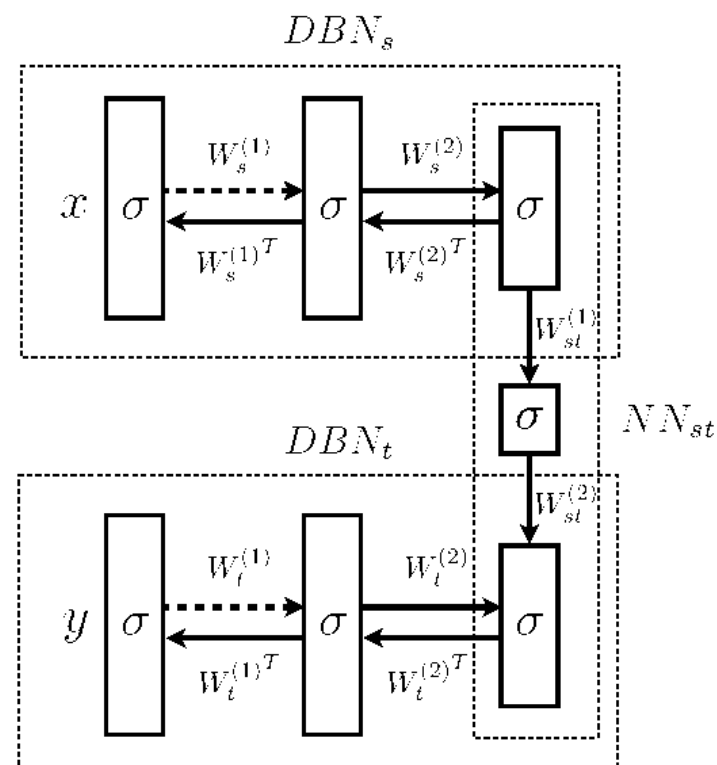
# RBM/CRBM for Voice Conversion [37, 38]

- Using RBM to model the spectral feature space instead of Gaussian
- Divide acoustic space by a GMM and train the samples in each sub-space by RBM
- CRBM to model the relationship between source target speech
- perform linear and non-linear transformations simultaneously



# DBN for Voice Conversion [39]

- Two different DBNs for source and target speakers
  - Feature transformation instead of MCEP in unsupervised training
- Concatenating NNs
  - Back-propagation using parallel data in supervised training



# Experimental Setup

- Speech corpus:
  - CMU ARCTIC database: 2 male (BDL and RMS,) 2 female (CLB and SLT), all US English speakers
- 6 Conversions, same gender and cross-gender:
  - SLT (F) to BDL (M)      BDL (M) to SLT (F)
  - SLT (F) to CLB (F)      CLB (F) to SLT (F)
  - BDL (M) to RMS (M)      RMS(M) to BDL (M)
- Training and Testing
  - Training: 100 parallel utterances
  - Testing: 100 utterances



# Experimental Setup

- Feature Vector
  - 256-dim spectral envelope<sup>1</sup> with delta and delta-delta
  - Log F0 (continuous) with delta and delta-delta and their contexts
  - 1-dim UV(unvoiced/voiced)
  - Normalized to zero mean and unity variance

<sup>1</sup>The performance of VC by using spectral envelope is better than those by using MCEP and LSP in [36,37]

# Experimental Setup

- Neural Network<sup>\*</sup>
  - Different NN structure, pre-training and frame error vs. sequence error trainings
  - Training procedure:
    - Pre-training → minimum frame error training → minimum sequence error training

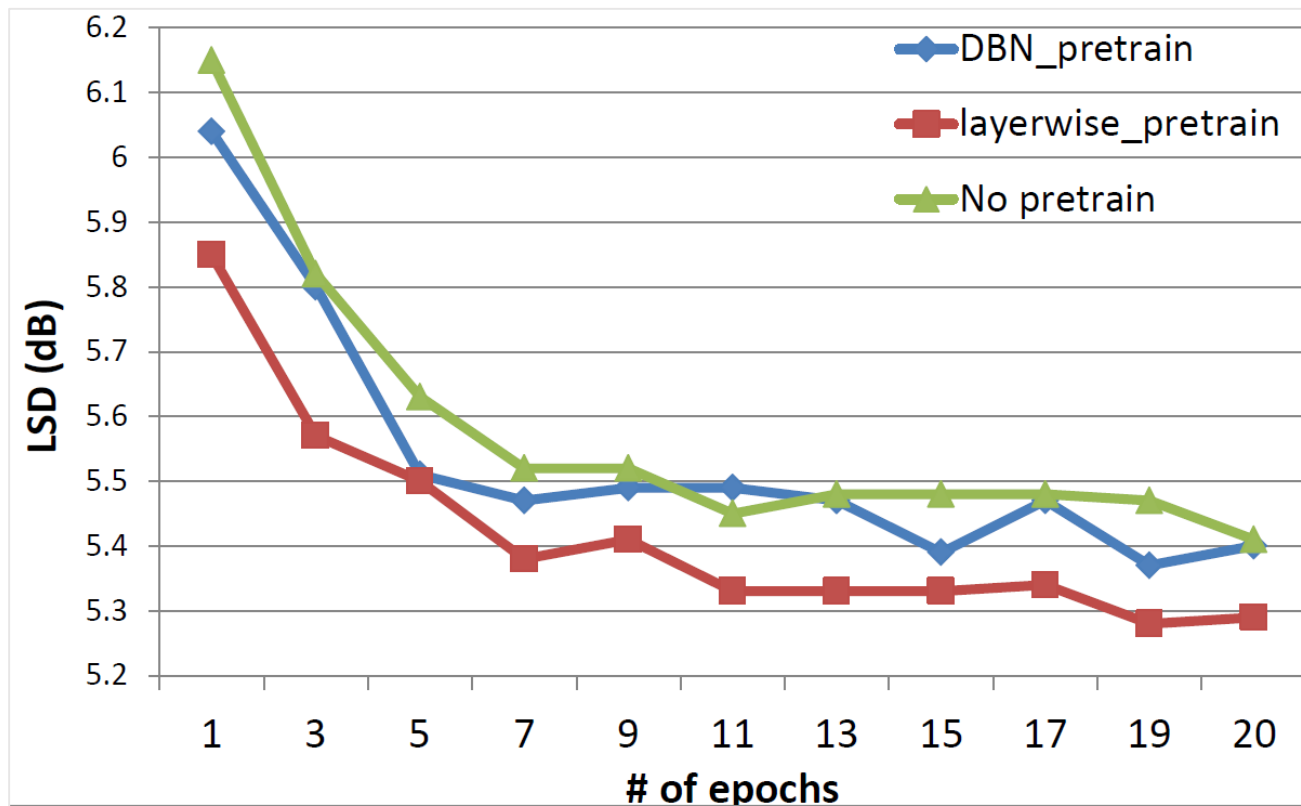
<sup>\*</sup>NN based VC can achieve a better or comparable performance, when compared with GMM [35-38]

# Evaluation Metrics

- Objective measures
  - Log Spectral Distance (LSD)
  - Root Mean Squared Error (RMSE) of F0
- Subjective measure
  - AB preference test for naturalness
  - ABX test for speaker similarity
  - Crowdsourcing (20 subjects, 50 pairs of sentences for each)

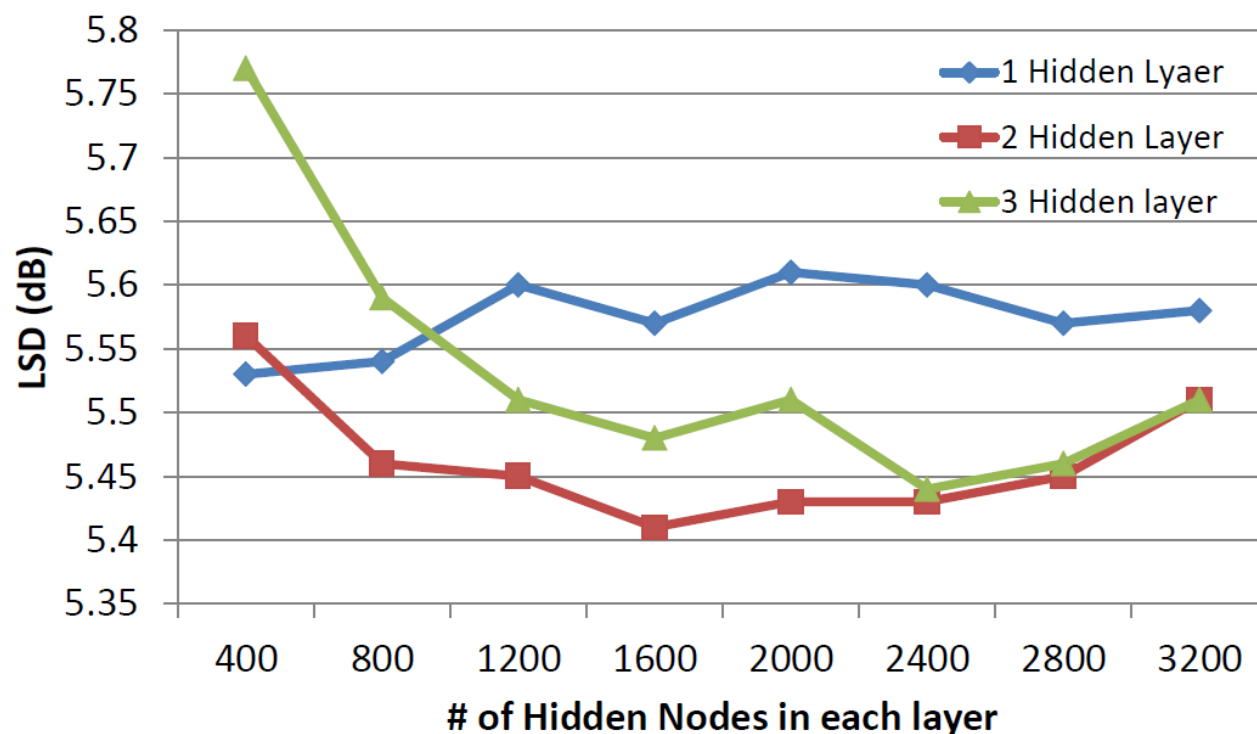
# Experimental Results

## ■ Pre-training



# Experimental Results

## ■ The Number of Hidden Layers



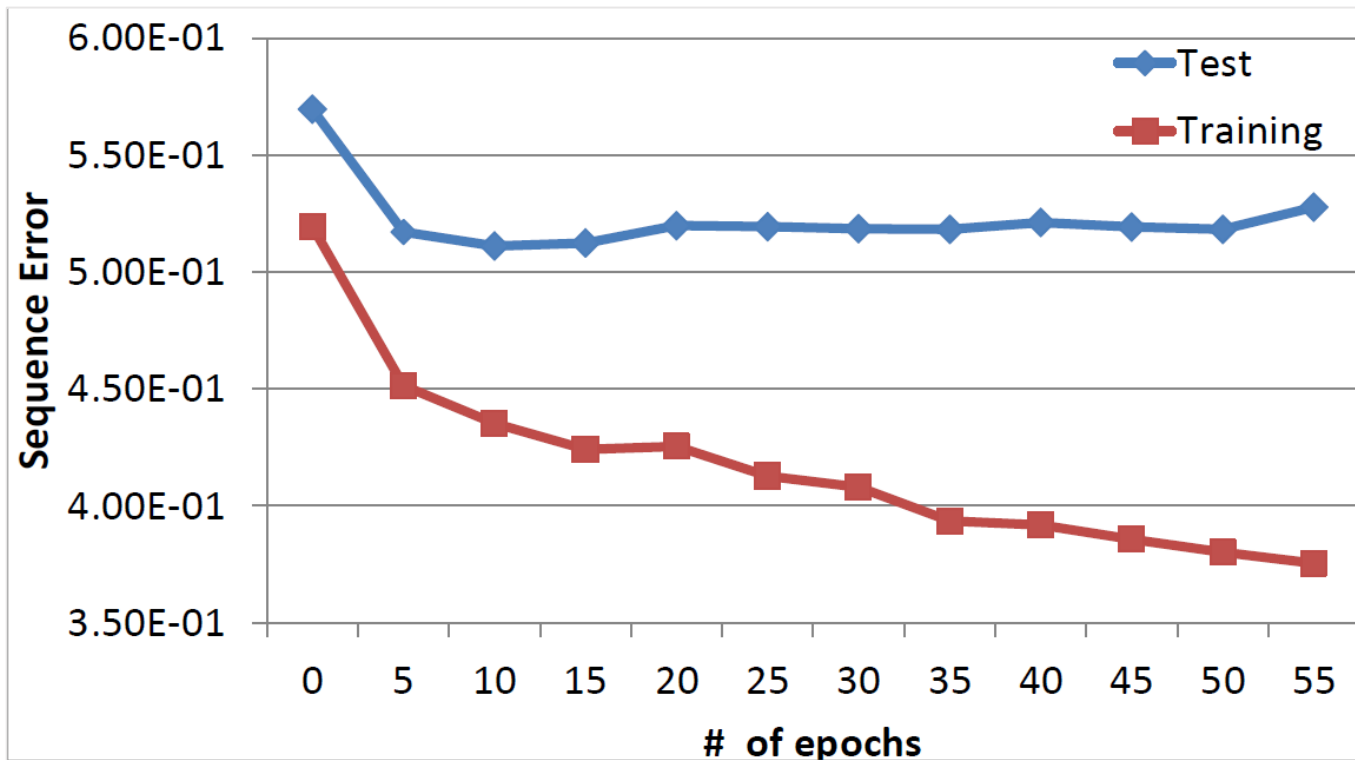
# Experimental Results

- F0 conversion

	RMSE of F0 (Hz)
Conventional method: normalized to same distribution	18.0
Embedded F0 modeling in NN (without context)	16.2
Embedded F0 modeling in NN (with 3 context)	14.8

























# Experimental Results

- Convergence of Minimum SE based NN Training



# Experimental Results

## ■ Minimum FE vs. Minimum SE

source to target	LSD (dB)		RMSE of F0 (Hz)	
	FE	SE	FE	SE
SLT to BDL  	5.41 	5.30 	16.27	15.47
BDL to SLT  	4.94 	4.81 	18.36	18.2
CLB to SLT  	4.83 	4.76 	15.04	14.81
SLT to CLB  	4.84 	4.73 	16.64	16.27
BDL to RMS  	4.92 	4.81 	15.25	14.78
RMS to BDL  	5.60 	5.37 	14.12	14.11

	FE	SE	N/P	P
Naturalness	23	60	17	<0.001
Similarity	35	65	-	<0.001

More Samples: <http://research.microsoft.com/en-us/projects/vcnn/default.aspx>



# Outline

- Statistical parametric speech generation and synthesis
  - HMM-based speech synthesis
  - GMM-based voice conversion
- Deep learning
  - RBM, DBN, DNN, MDN and RNN
- Deep learning for speech generation and synthesis
  - Approaches to speech synthesis
  - Approaches to voice conversion
- **Conclusions and future work**

# Conclusions and Future Work

- Deep learning for speech generation and synthesis is working, it can
  - Replace decision tree with full-context, distributed, easy-to-interpolate representation
  - Replace GMM with more fitted distributions in correlated, high dimensional space
  - Replace ML training with criteria closer to the final objectives, e.g., sequence error training
  - Replace rich context with wider window in time, e.g., the whole sentence
- Many possible future directions with deep learning, e.g.
  - Improving the front-end of TTS, i.e., text analysis
  - More efficient speaker adaptation with less data
  - Cross-lingual TTS

# References

- [1] K. Tokuda, H. Zen, A.W. Black, *An HMM-based speech synthesis system applied to English*, Proc. of 2002 IEEE SSW, 2002.
- [2] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, T. Kitamura, *Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis*, Proc. of Eurospeech, pp.2347-2350, 1999.
- [3] K. Tokuda, T. Mausk, N. Miyazaki, T. Kobayashi, *Multi-space probability distribution HMM*, IEICE Trans. Inf. & Syst., vol.E85-D, no.3, pp.455-464, 2002.
- [4] J.J. Odell, The use of context in large vocabulary speech recognition, Ph.D dissertation, Cambridge University, 1995.
- [5] K. Shinoda and T. Watanabe, MDL-based context-dependent sub-word modeling for speech recognition, J. Acoust. Soc. Jpn (E), Vol. 21, No.2, pp. 79-86, 2000.
- [6] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, T. Kitamura, *Speech parameter generation algorithms for HMM-based speech synthesis*, Proc. of ICASSP, pp.1315-1318, 2000.
- [7] K. Tokuda, An HMM-based Approach to Flexible Speech Synthesis, slides for Tutorial in ISCSLP 2006.
- [8] A. V. McCree and T. P. Barnwell III, "A mixed excitation LPC vocoder model for low bit rate speech coding," IEEE Transactions on Speech and Audio Processing, Vol.3, No.4, 1995.
- [9] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, T. Kitamura, "Mixed excitation for HMM-based speech synthesis", in Proc. EuroSpeech, 2001.

# References

- [10] T. Toda and K. Tokuda, Speech Parameter Generation Algorithm Considering Global Variance for HMM-Based Speech Synthesis, in Proc. Eurospeech, 2005.
- [11] Y.-J. Wu, R.-H. Wang, *Minimum generation error training for HMM-based speech synthesis*, Proc. of ICASSP, pp.89-92, 2006.
- [12] Y. Stylianou, O. Cappe and E. Moulines, “Continuous probabilistic transform for voice conversion,” IEEE Trans. Speech and Audio Processing, Vol 6, No. 2, pp. 131-142, 2002.
- [13] T. Toda, A.W. Black and K. Tokuda, “Voice conversion based on maximum likelihood estimation of spectral parameter trajectory,” IEEE Trans. Audio, Speech and Language Processing, Vol. 15, No. 8, pp. 2222-2235, 2007.
- [14] T. Toda, A.W. Black, K. Tokuda, Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter. *Proc. ICASSP*, Vol. 1, pp. 9-12, Philadelphia, USA, Mar 2005.
- [15] T. Toda, A Statistical Approach to Voice Conversion and Its Applications for Augmented Human Communication, slides for Tutorial in ISCSLP 2012.
- [16] Y. Bengio., Learning deep architectures for AI, Foundations and Trends in Machine Learning, 2(1):1-127, 2009.

# References

- [17] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.
- [18] G.E. Hinton, S. Osindero and Y. W. Teh, “A Fast Learning Algorithm for Deep Belief Nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 9, pp. 533–536, 1986.
- [20] F. Seide, G. Li, X. Chen, and D. Yu, “Feature engineering in context-dependent deep neural networks for conversational speech transcription,” in *IEEE ASRU*, 2011.
- [21] C. Bishop, Mixture density network, Tech. Rep. NCRG/94/004, Neural Computing Research Group, Aston University, 1994.
- [22] A. Graves. Supervised Sequence Labelling with Recurrent Neural Networks, Dissertation, Technische Universität München, München, July 2008.
- [23] H. Sepp, S. Jürgen, “Long short-term memory.” *Neural computation*, vol.9, no.8, pp. 1735-1780, 1997.
- [24] S. Mike, K. Paliwal. “Bidirectional recurrent neural networks.” *IEEE Transactions on Signal Processing*, vol.45, no.11, pp.2673-2681, 1997.

# References

- [25] S. Kang, X. Qian, and H. Meng, “Multi-distribution deep belief network for speech synthesis”, In Proc. ICASSP, pp. 7962-7966, 2013.
- [26] Z.-H. Ling, L. Deng, and D. Yu, “Modeling spectral envelopes using restricted Boltzmann machines for statistical parametric speech synthesis”, In Proc. ICASSP, pp. 7825-7829, 2013.
- [27] H. Zen, A. Senior and M. Senior, “Statistical Parametric Speech Synthesis Using Deep Neural Networks”, In Proc. ICASSP, pp. 8012-8016, 2013.
- [28] H. Lu, S. King, and O. Watts, “Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis”, In 8th ISCA Workshop on .Speech Synthesis, pp. 281-285, 2013.
- [29] Y. Qian, Y.-C. Fan, W.-P. Hu and F. K. Soong, “On the training aspects of deep neural network (DNN) for parametric TTS synthesis”, In Proc. ICASSP, 2014.
- [30] R. Fernandez, A. Rendel, B. Ramabhadran, R. Hoory, “F0 Contour Prediction with a Deep Belief Network-Gaussian Process Hybrid Model”, in Proc. ICASSP, 2013.
- [31] H. Zen and A. Senior, “Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis”, in Proc. ICASSP 2014.
- [32] Y.-C. Fan, Y. Qian, F.-L. Xie and F. K. Soong, “TTS Synthesis with Bidirectional LSTM based Recurrent Neural Networks”, In. Proc. Interspeech, 2014.

# References

- [33] R. Fernandez, A. Rendel, B. Ramabhadran, R. Hoory, “Prosody Contour Prediction with Long Short-Term Memory, Bi-Directional Deep Recurrent Neural Networks”, In. Proc. Interspeech 2014.
- [34] D. Yu, “Large Vocabulary Speech Recognition Using Deep Neural Networks: Insights, Theory and Practice”, slides for Tutorial in ISCSLP 2012.
- [35] S. Desai, A. W. Black, B. Yegnanarayana, “Spectrum Mapping Using Artificial Neural Networks for Voice Conversion,” IEEE Trans. Audio, Speech, Lang. Process, vol. 18, No.5, July 2010
- [36] F.-L. Xie, Y. Qian, Y.-C. Fan, F. K. Soong, H.-F. Li, “Sequence Error(SE) Minimization Training of Neural Network for Voice Conversion”, in Proc. Interspeech, 2014.
- [37] L.-H. Chen, Z.-H. Ling, Y. Song, L.-R. Dai, “Joint Spectral Distribution Modeling Using Restricted Boltzmann Machines for Voice Conversion,” in Proc. Interspeech, pp. 3052-3056, 2013.
- [38] Z.-Z. Wu, E.-S. Chng, H.-Z. Li, “Conditional Restricted Boltzmann machine for Voice Conversion”, in Proc. ChinaSip, pp.104-108, 2013.
- [39] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, “Voice Conversion in high-order eigen space using deep belief nets,” in Proc. Interspeech, pp.369-372, 2013.
- [40] <http://sourceforge.net/projects/currentnt/?source=navbar>

# References

[41] F.-L. Xie, Y. Qian, F. K. Soong and H.-F. Li, "Pitch Transformation in Neural Network Based Voice Conversion" in Proc. ISCSP, 2014.