# 9.0 Speech Recognition Updates

# Minimum-Classification-Error (MCE) and Discriminative Training

- **A Primary Problem with the Conventional Training Criterion : Confusing sets**

  find $\lambda^{(i)}$ such that $P(X|\lambda^{(i)})$ is maximum (Maximum Likelihood) if $X \in \boldsymbol{C_i}$

  - This does not always lead to minimum classification error, since it doesn't consider the mutual relationship among competing classes
  - The competing classes may give higher likelihood function for the test data

- **General Objective : find an optimal set of parameters (e.g. for recognition models) to *minimize the expected error of classification***
  - the statistics of test data may be quite different from that of the training data
  - training data is never enough

- **Assume the recognizer is operated with the following classification principles :**

  $\{C_i, i=1,2,...M\}$, M classes

  $\lambda^{(i)}$: statistical model for $C_i$

  $\Lambda=\{\lambda^{(i)}\}_{i=1......M}$ , the set of all models for all classes

  X : observations

  $g_i(X,\Lambda)$: class conditioned likelihood function, for example,

  $$g_i(X,\Lambda) = P(X|\lambda^{(i)})$$

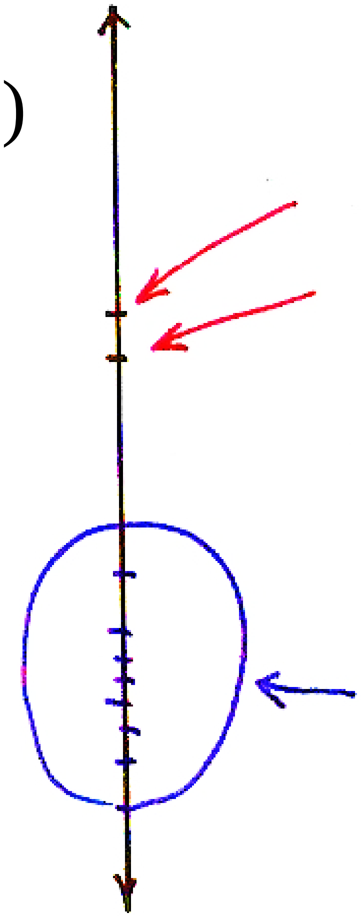
  - $\boldsymbol{C}(X) = \boldsymbol{C_i}$ if $g_i(X,\Lambda) = \max_j g_j(X,\Lambda)$ : classification principles

    an error happens when $P(X|\lambda^{(i)}) = \max$ but $X \notin \boldsymbol{C_i}$

# Minimum-Classification-Error (MCE)

$\lambda^{(0)}, \lambda^{(1)}, \lambda^{(2)}, \ldots \lambda^{(9)}$

$P\,(O|\lambda^{(k)})$



$P\,(O|\lambda^{(7)})$ : correct

$P\,(O|\lambda^{(1)})$ : competing

wrong

# Minimum-Classification-Error (MCE) Training

- **One form of the misclassification measure**

$$d_i(X,\Lambda) = -g_i(X,\Lambda) + \left[\frac{1}{M-1}\sum_{j \neq i} g_j(X,\Lambda)^{\alpha}\right]^{\frac{1}{\alpha}} \qquad X \in C_i$$

  – Comparison between the likelihood functions for the correct class and the competing classes

    $\alpha = 1$ all other classes included and averaged with equal weights
    $\alpha \to \infty$ only the most competing one considered
    $d_i(X) \geq 0$ implies a classification error
    $d_i(X) < 0$ implies a correct classification

- **A continuous loss function is defined**

$$l_i(X,\Lambda) = l(d_i(X,\Lambda)), \ \mathrm{X} \in \mathrm{C_i}$$

$$l(d) = \frac{1}{1+\exp[-\gamma(d-\theta)]}, \ sigmoid \ \ function$$

  – l(d) →0 when d →-∞
    l(d) →1 when d →∞
    θ: switching from 0 to 1 near θ
    γ: determining the slope at switching point

- **Overall Classification Performance Measure :**

$$L(\Lambda) = E_X\left[L(X,\Lambda)\right] = \sum_X \left[L(X,\Lambda)\right] = \sum_{i=1}^{M}\left[\sum_{X \in C_i} l_i(X,\Lambda)\right]$$

# Sigmoid Function

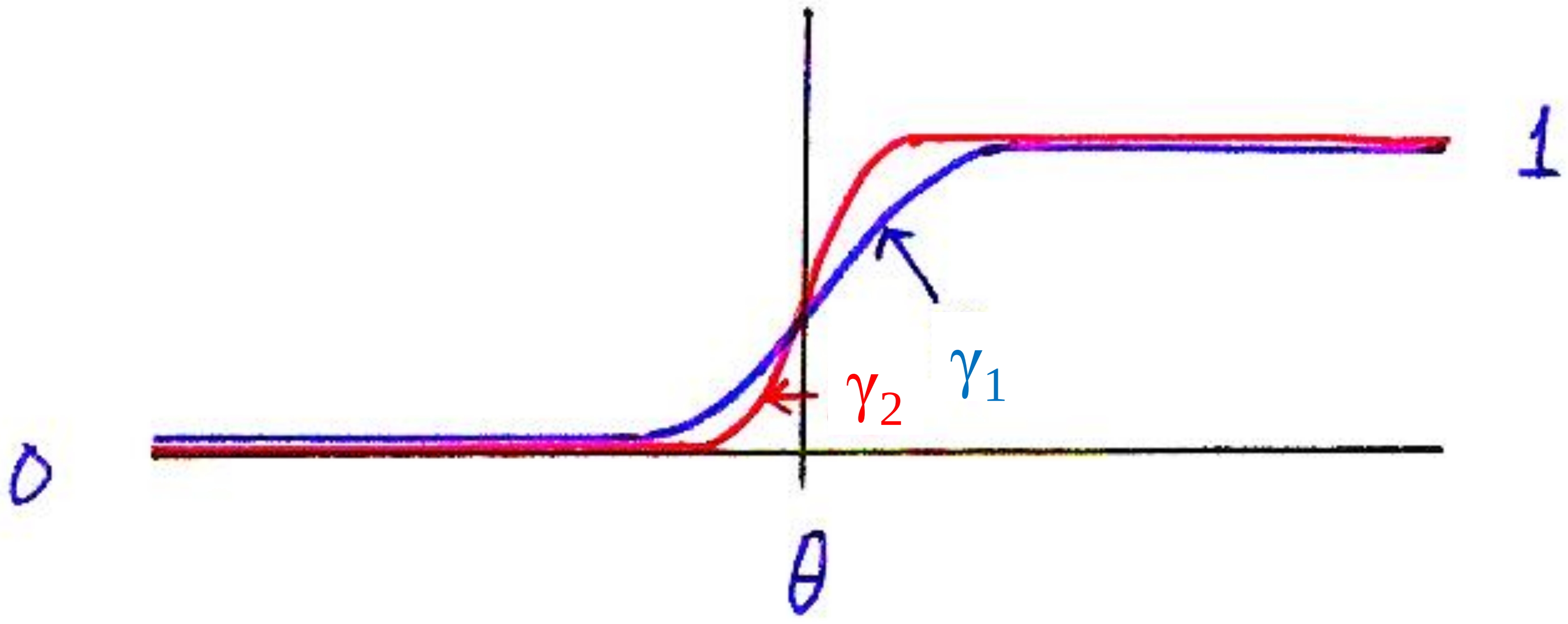
$$1(d) = \frac{1}{1 + exp[-\gamma(d - \theta)]}$$

l(d) →0 when d →-∞

l(d) →1 when d →∞

θ: switching from 0 to 1 near θ

γ: determining the slope at switching point



# Minimum-Classification-Error (MCE) Training

- **Find $\hat{\Lambda}$ such that**

$$\hat{\Lambda} = \arg\min_{\Lambda} L(\Lambda) = \arg\min_{\Lambda} E_X \left[ L(X, \Lambda) \right]$$

  - the above objective function in general is difficult to minimize directly
  - local minimum can be obtained iteratively using gradient (steepest) descent algorithm

$$\Lambda_{t+1} = \Lambda_t - \varepsilon_t \nabla L(\Lambda_t)$$

$\nabla$ : partial differentiation with respect to all different parameters individually

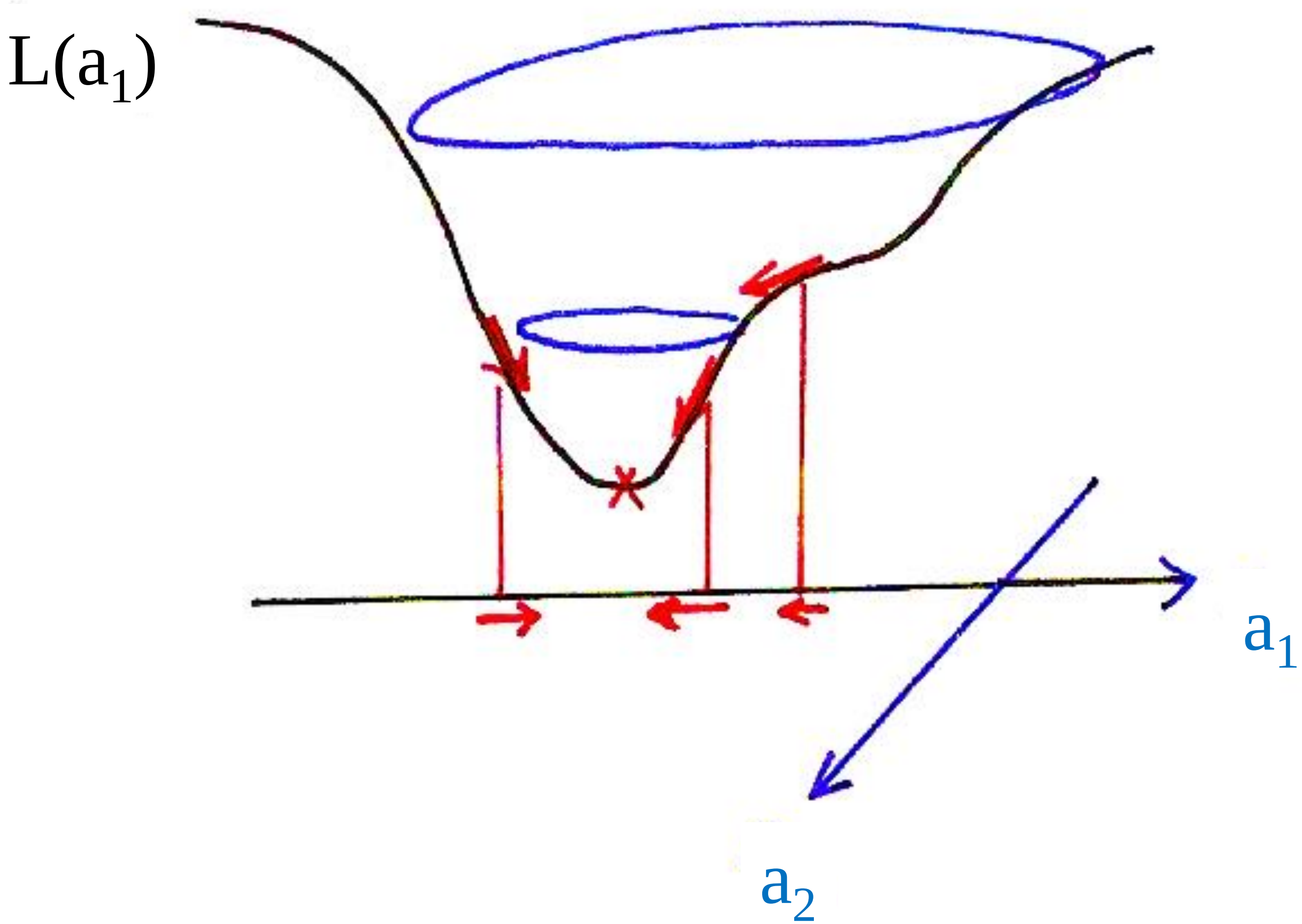
t : the t-th iteration

$\varepsilon$ : adjustment step size, should be carefully chosen

$$a_{t+1} = a_t - \varepsilon_t \frac{\partial L(\Lambda)}{\partial a}, \ a : an\ arbitrary\ parameter\ of\ \Lambda$$

  - every training observation may change the parameters of ALL models, not the model for its class only

# Gradient Descent Algorithm

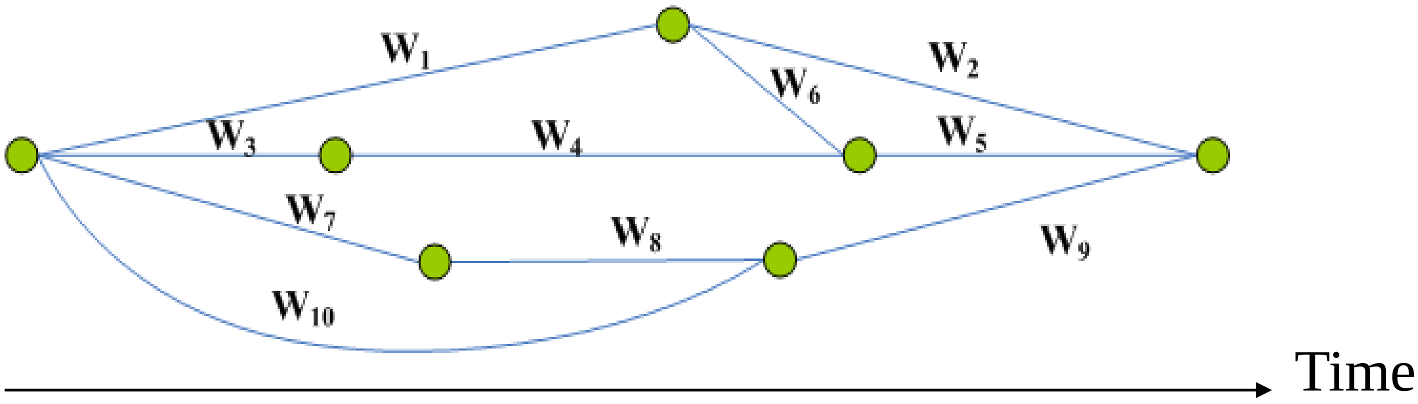


# Discriminative Training and Minimum Phone Error Rate (MPE) Training For Large Vocabulary Speech Recognition

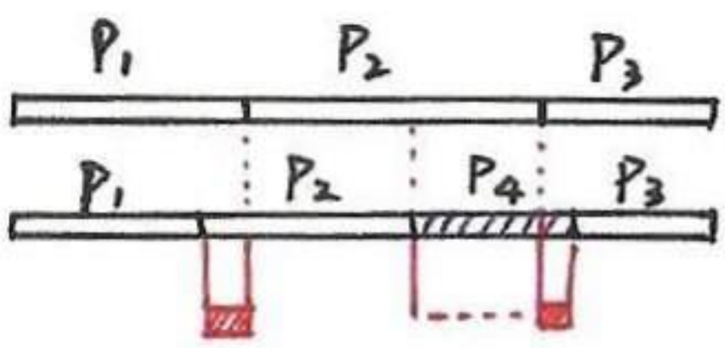
- **Minimum Bayesian Risk (MBR)**
  - $(\Lambda,\Gamma) = \arg\min_{\Lambda',\Gamma'} \sum_r R(s_r \mid O_r)$ adjusting all model parameters to minimize the Bayesian Risk
    - $\Lambda$: $\{\lambda_i, i=1,2,\ldots\ldots N\}$ acoustic models
    - $\Gamma$: Language model parameters
    - $O_r$ : r-th training utterance
    - $s_r$: correct transcription of $O_r$
  - $R(s_r \mid O_r) = \sum_u P_{\Lambda,\Gamma}(u \mid O_r) L(u, s_r)$ Bayesian Risk
    - u: a possible recognition output found in the lattice
    - $L(u,s_r)$ : Loss function
    - $P_{\Lambda,\Gamma}(u|O_r)$ : posteriori probability of $u$ given $O_r$ based on $\Lambda,\Gamma$
  - $L(u, s_r) = \begin{cases} 0, u = s_r \\ 1, u \neq s_r \end{cases} \rightarrow MAP$ principle
  - Other definitions of $L(u,s_r)$ possible
- **Minimum Phone Error Rate (MPE) Training**
  - $(\Lambda,\Gamma) = \arg\max_{\Lambda',\Gamma'} \sum_r \sum_u P(u \mid O_r) Acc(u, s_r)$
    - $Acc(u,s_r)$ : phone accuracy
  - Better features obtainable in the same way
    - e.g. $y_t = x_t + Mh_t$ feature-space MPE

# Minimum Phone Error (MPE) Rate Training

- **Lattice**




- **Phone Accuracy**




← Reference phone sequence

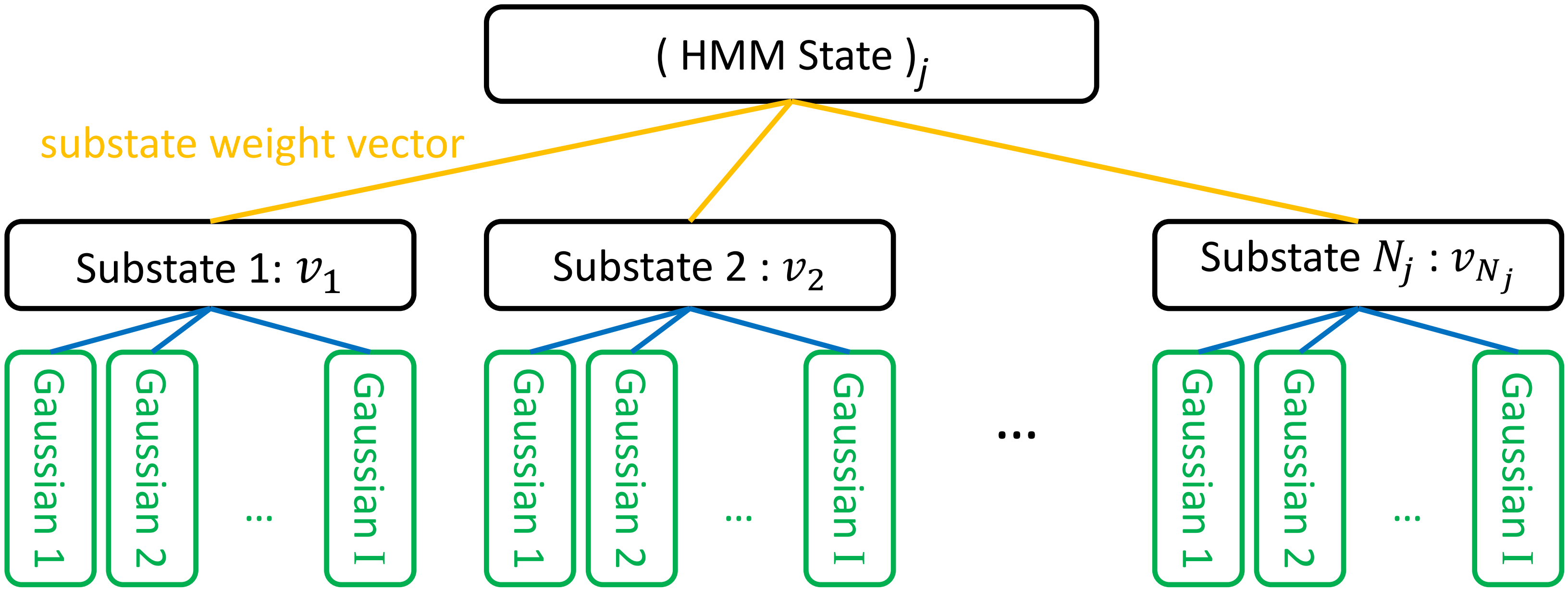
← Decoded phone sequence for a path in the lattice

# References for MCE, MPE and Discriminative Training


- “ Minimum Classification Error Rate Methods for Speech Recognition”, IEEE Trans. Speech and Audio Processing, May 1997
- “Segmental Minimum Bayes-Rick Decoding for Automatic Speech Recognition”, IEEE Trans. Speech and Audio Processing, 2004
- “Minimum Phone Error and I-smoothing for Improved Discriminative Training”, International Conference on Acoustics, Speech and Signal Processing, 2002
- “Discriminative Training for Automatic Speech Recognition”, IEEE Signal Processing Magazine, Nov 2012

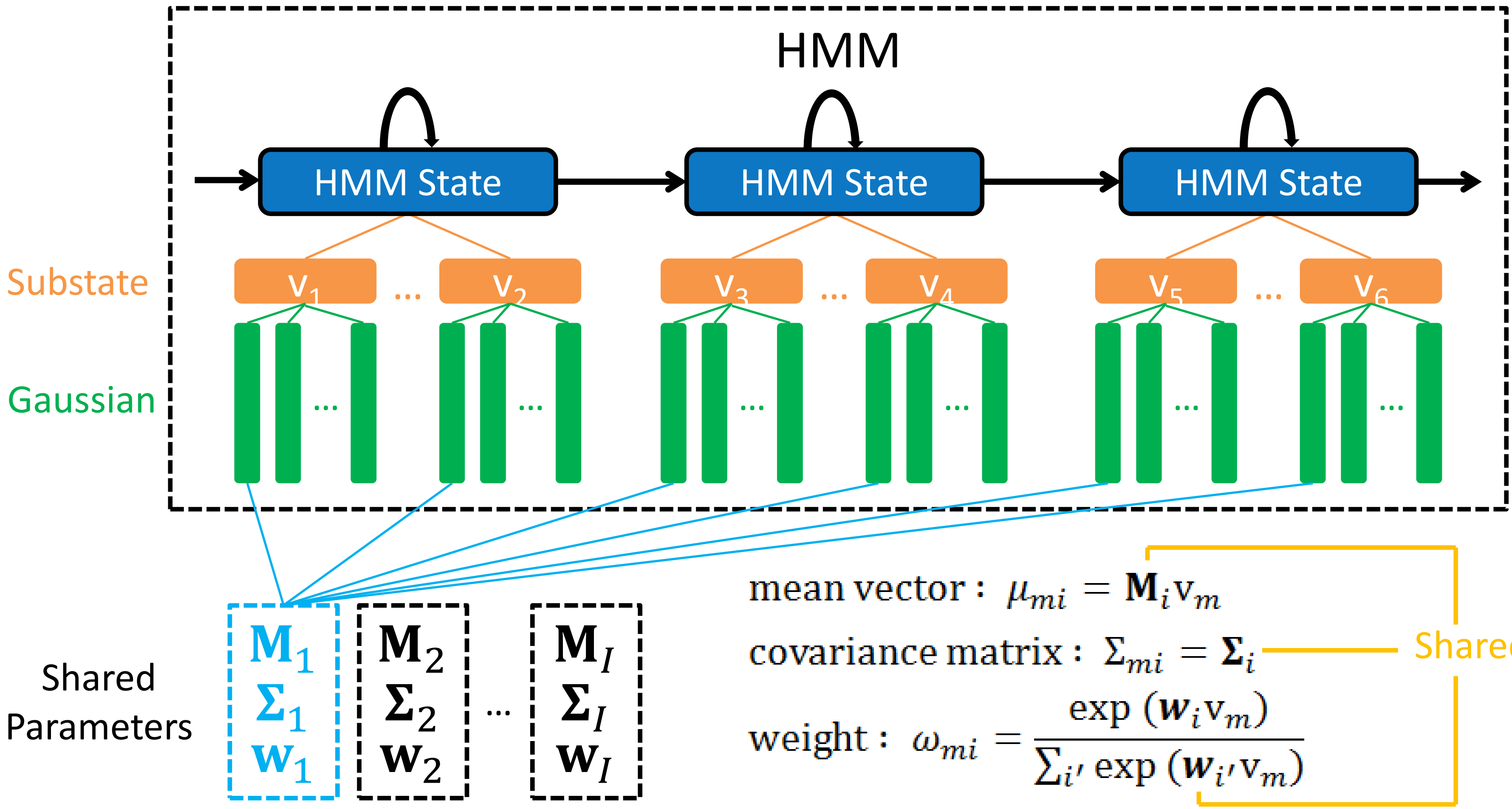
# Subspace Gaussian Mixture Model

- **To increase the modeling flexibility while reducing the required free parameters**
  - In a triphone HMM, different states can have different number of substates
  - Fixed number of I Gaussians in each substate, I ≈ 400
  - Similar to many and varying number of Gaussian mixtures in each state in conventional HMM-GMM
  - Each substate specified by a vector $v_m$ of S dimensions only, S ≈ 40, while the parameters of all Gaussians under all different triphones are determined based on a set of shared parameters $\{(M_i,\ \Sigma_i,\ w_i),\ i = 1, 2, \cdots, I\}$



# Subspace Gaussian Mixture Model

- **A triphone HMM in Subspace GMM**




$$\text{mean vector}: \ \mu_{mi} = \mathbf{M}_i \mathrm{v}_m$$

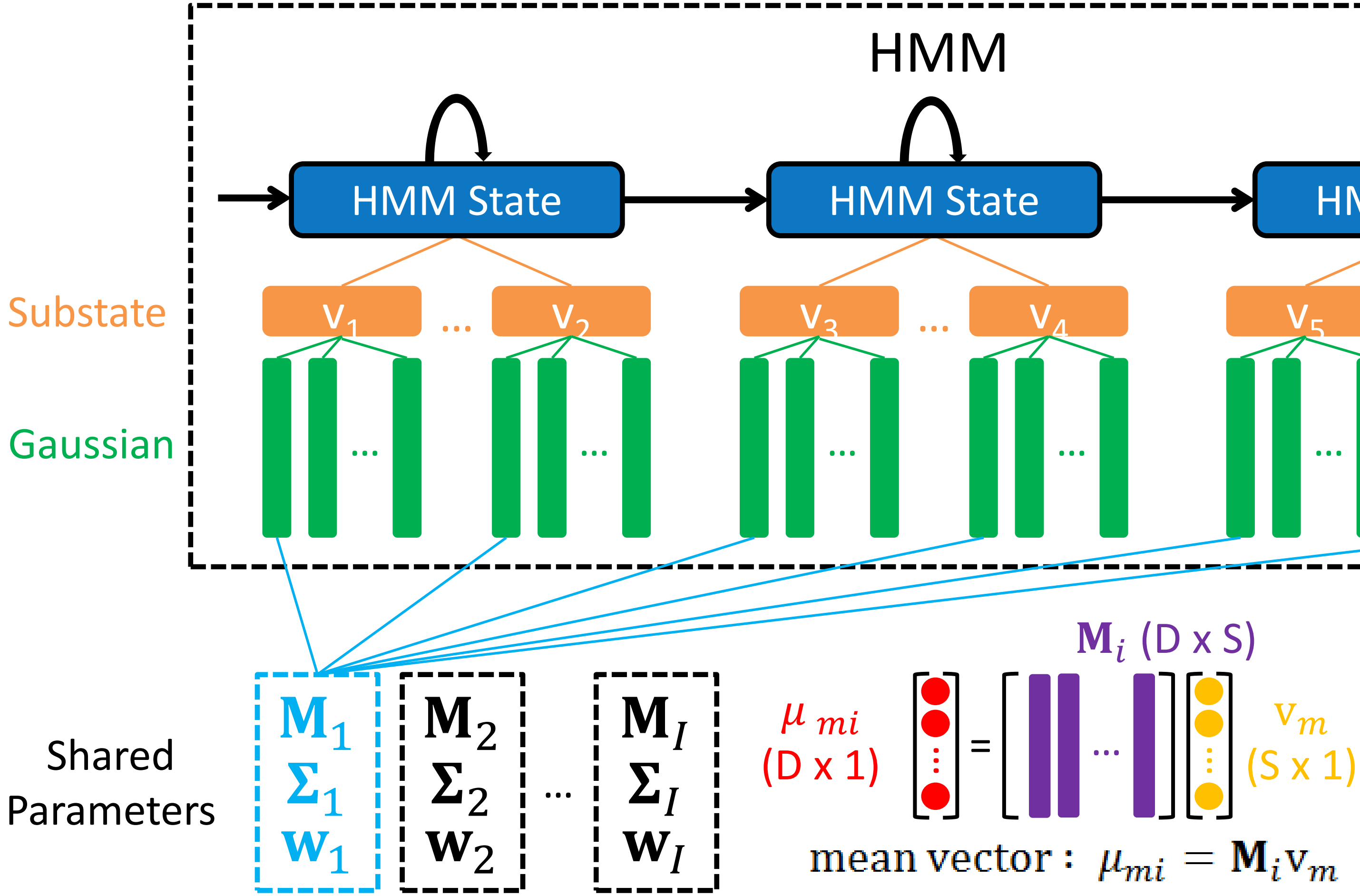
$$\text{covariance matrix}: \ \Sigma_{mi} = \mathbf{\Sigma}_i$$

$$\text{weight}: \ \omega_{mi} = \frac{\exp(\boldsymbol{w}_i \mathrm{v}_m)}{\sum_{i'} \exp(\boldsymbol{w}_{i'} \mathrm{v}_m)}$$

I ≈ 400, $v_m$: S dimensional, S ≈ 40

# Subspace Gaussian Mixture Model

- **A triphone HMM in Subspace GMM**




mean vector : $\mu_{mi} = \mathbf{M}_i \mathrm{v}_m$

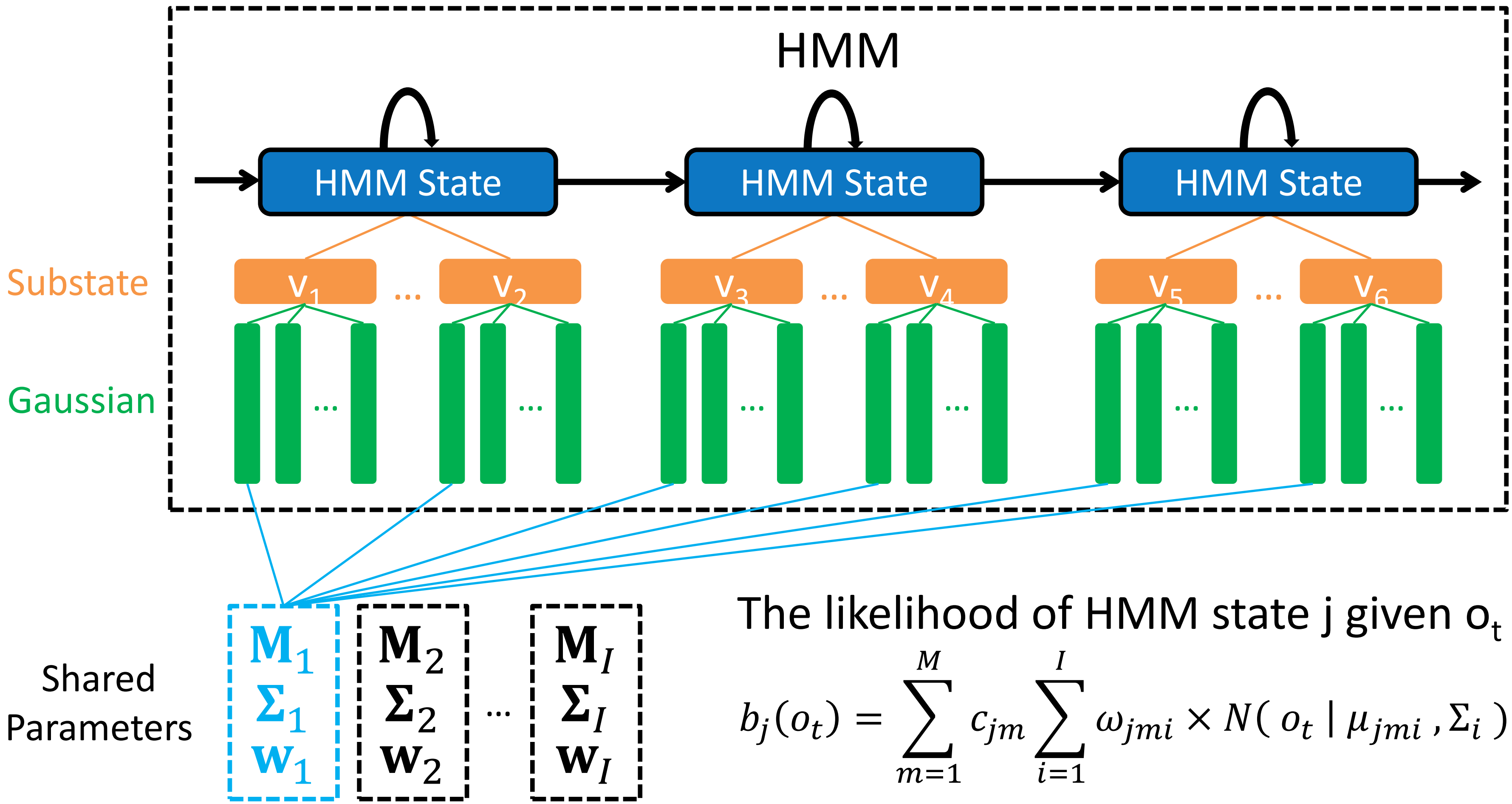
$I \approx 400$

$D \approx 39 \quad S \approx 40$

$\mathbf{M}_i$ is the basis set spanning a subspace of mean (columns of $\mathbf{M}_i$ not necessarily orthogonal)

# Subspace Gaussian Mixture Model

- **A triphone HMM in Subspace GMM**




The likelihood of HMM state j given $o_t$

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} \sum_{i=1}^{I} \omega_{jmi} \times N(\, o_t \mid \mu_{jmi}\, , \Sigma_i\, )$$

j: state, m:substate, i: Gaussian

# References for Subspace Gaussian Mixture Model


- **"The Subspace Gaussian Mixture Model– a Structured Model for Speech Recognition"**, D. Povey, Lukas Burget et. al Computer Speech and Language, 2011

- **"A Symmetrization of the Subspace Gaussian Mixture Model"**, Daniel Povey, Martin Karafiat, Arnab Ghoshal, Petr Schwarz, ICASSP 2011

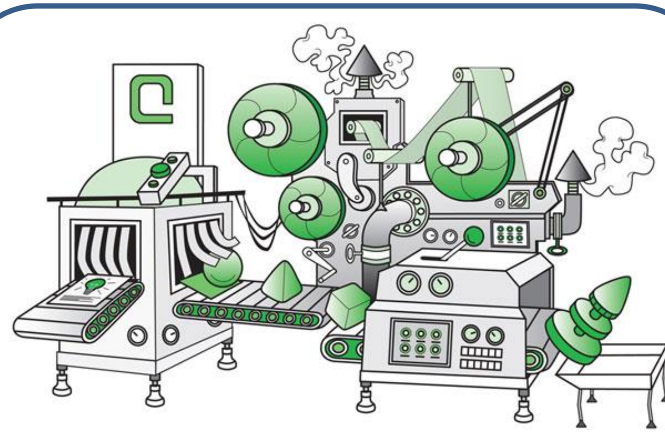
- **"Subspace Gaussian Mixture Models for Speech Recognition"**, D. Povey, Lukas Burget et al., ICASSP 2010

- **"A Tutorial-Style Introduction To Subspace Gaussian Mixture Models For Speech Recognition"**, Microsoft Research technical report MSR-TR-2009-111

# Neural Network — Classification Task

**Features** | **Classifier** | **Classes**

•Hair Length
•Make-up
.
.
.



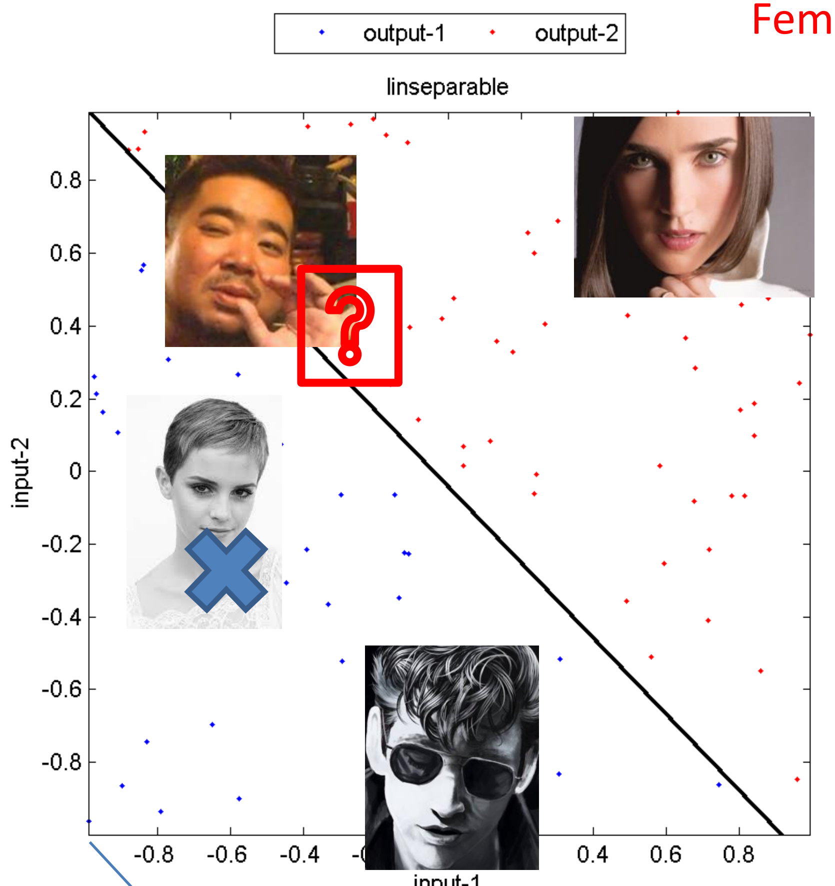
Male

Female

Others

# Neural Network — 2D Feature Space

Female



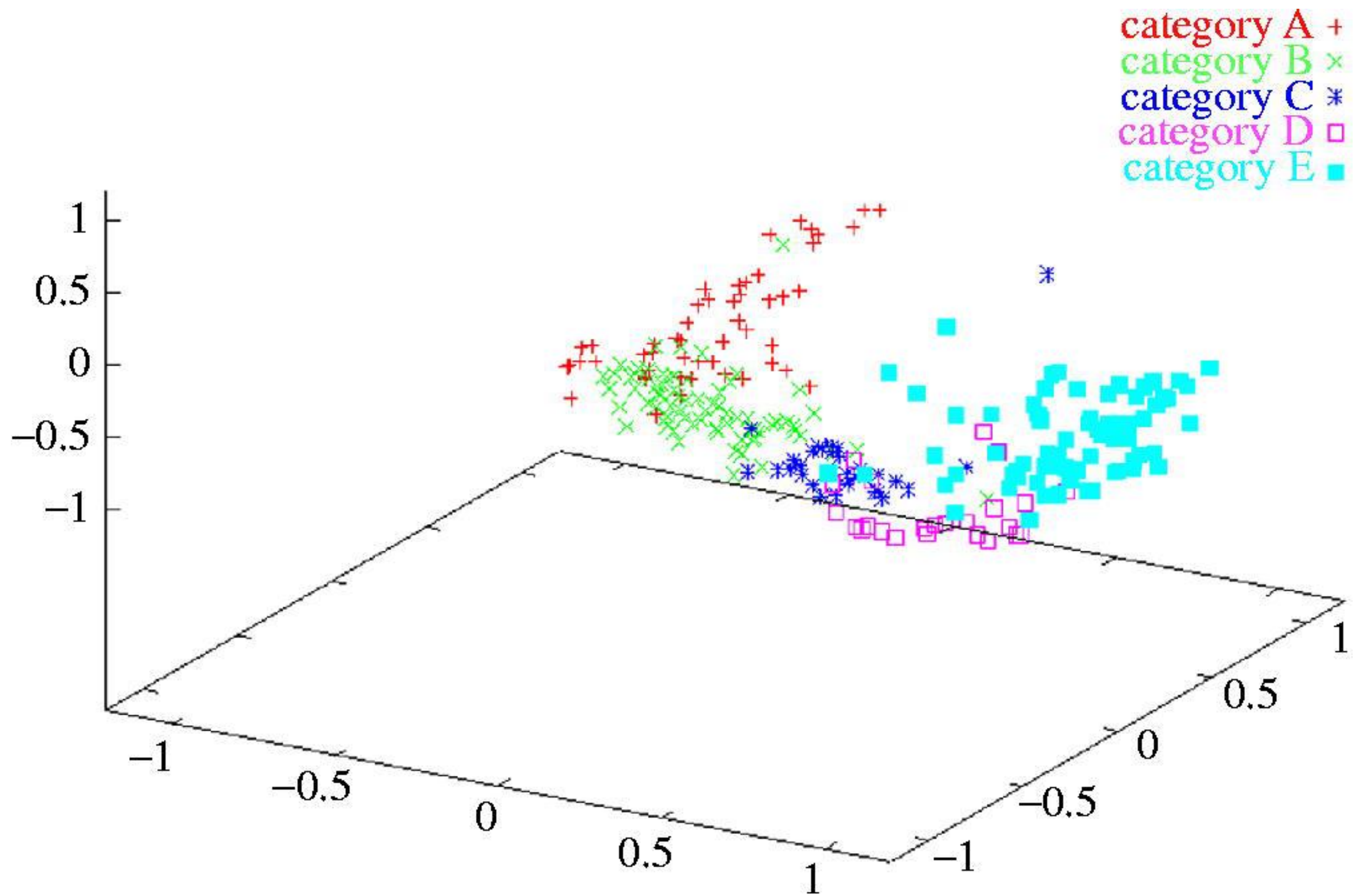

Make-Up

Male

Hair Length

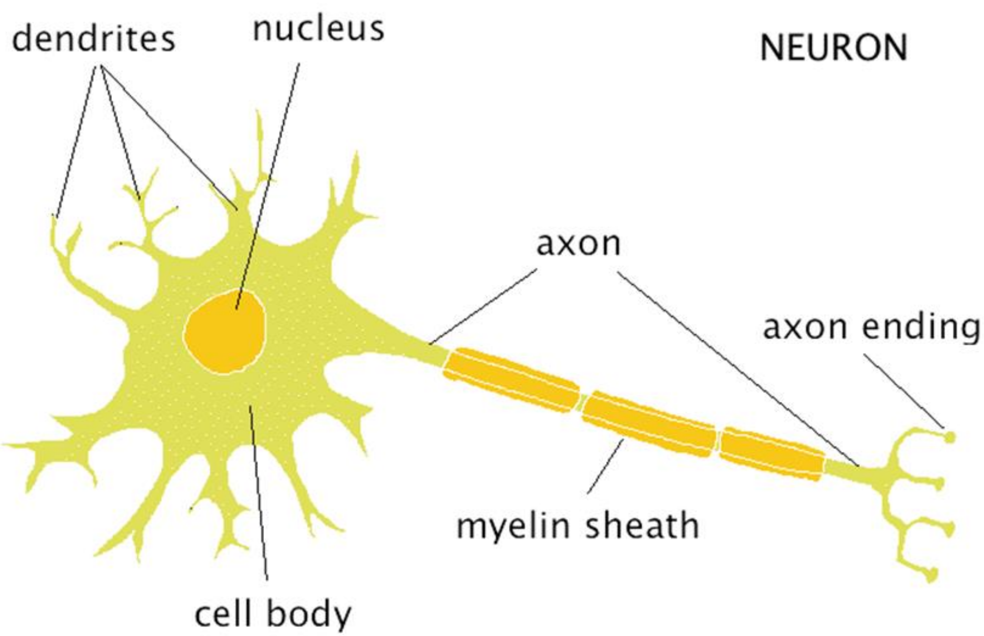
Voice pitch

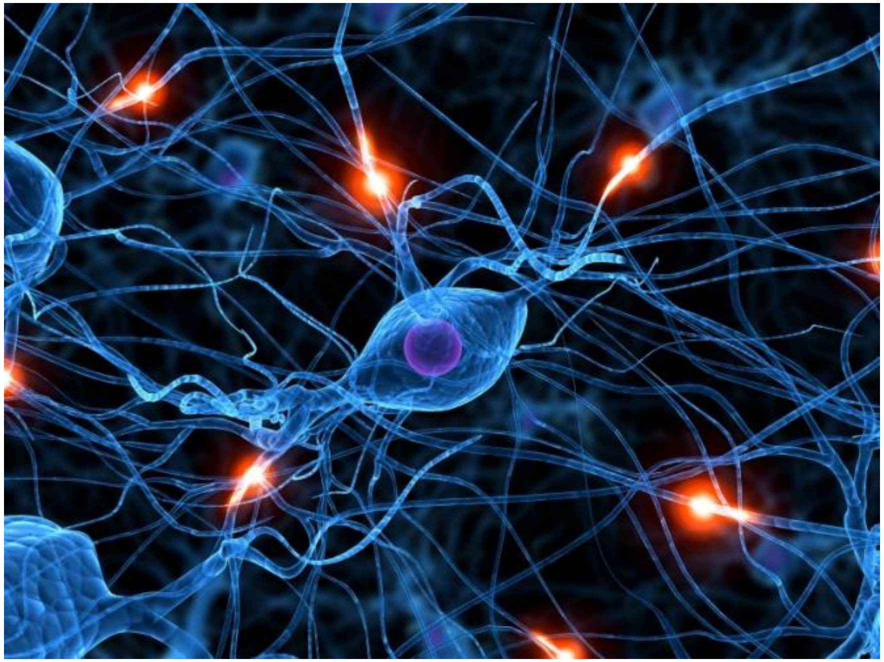
# Neural Network – Multi-Dimensional Feature Space




- **We need some type of non-linear function!**

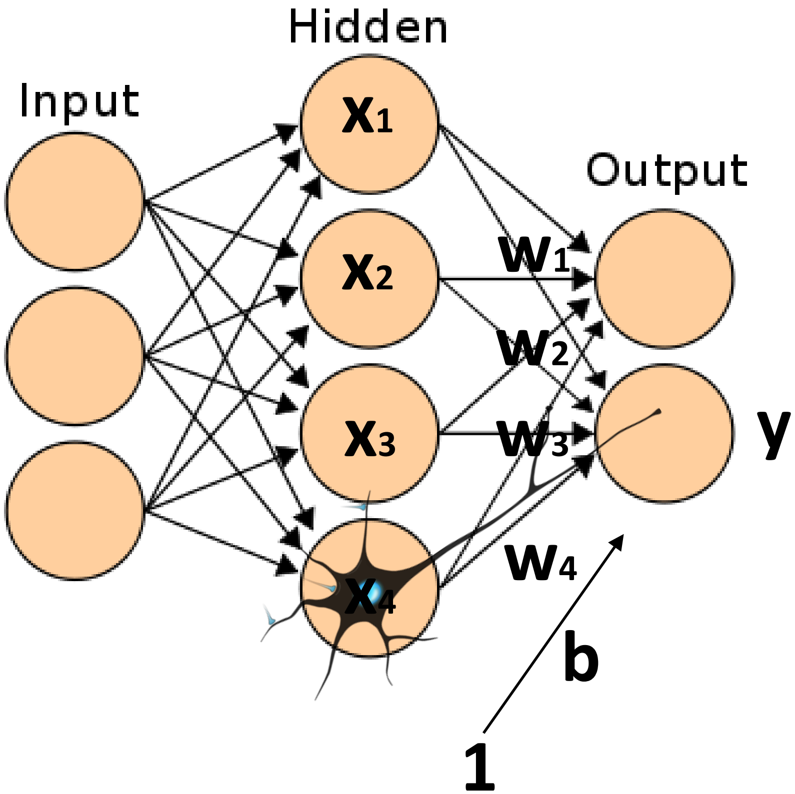
# Neural Network — Neurons



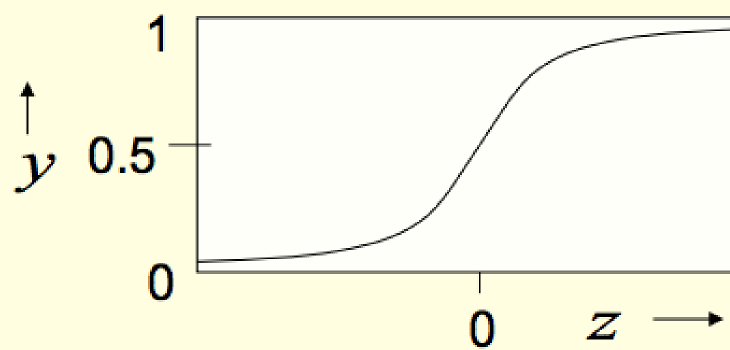
- **Each neuron receives inputs from other neurons**
- **The effect of each input on the neuron is adjustable (weighted)**
- **The weights adapt so that the whole network learns to perform useful tasks**

# Neural Network




$$y_k = f(b_k + \sum_i x_i w_{ik})$$

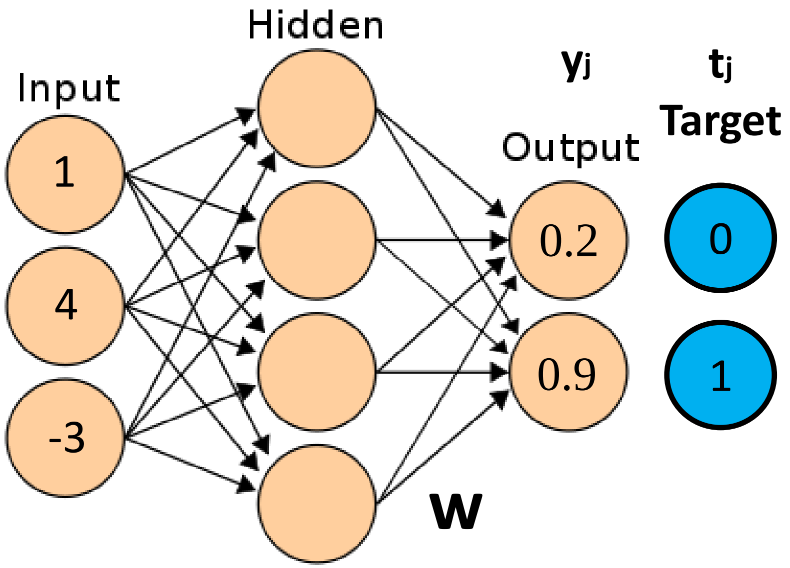
$$z = b + \sum_i x_i w_i \qquad y = \frac{1}{1+e^{-z}}$$




- A lot of simple non-linearity → complex non-linearity

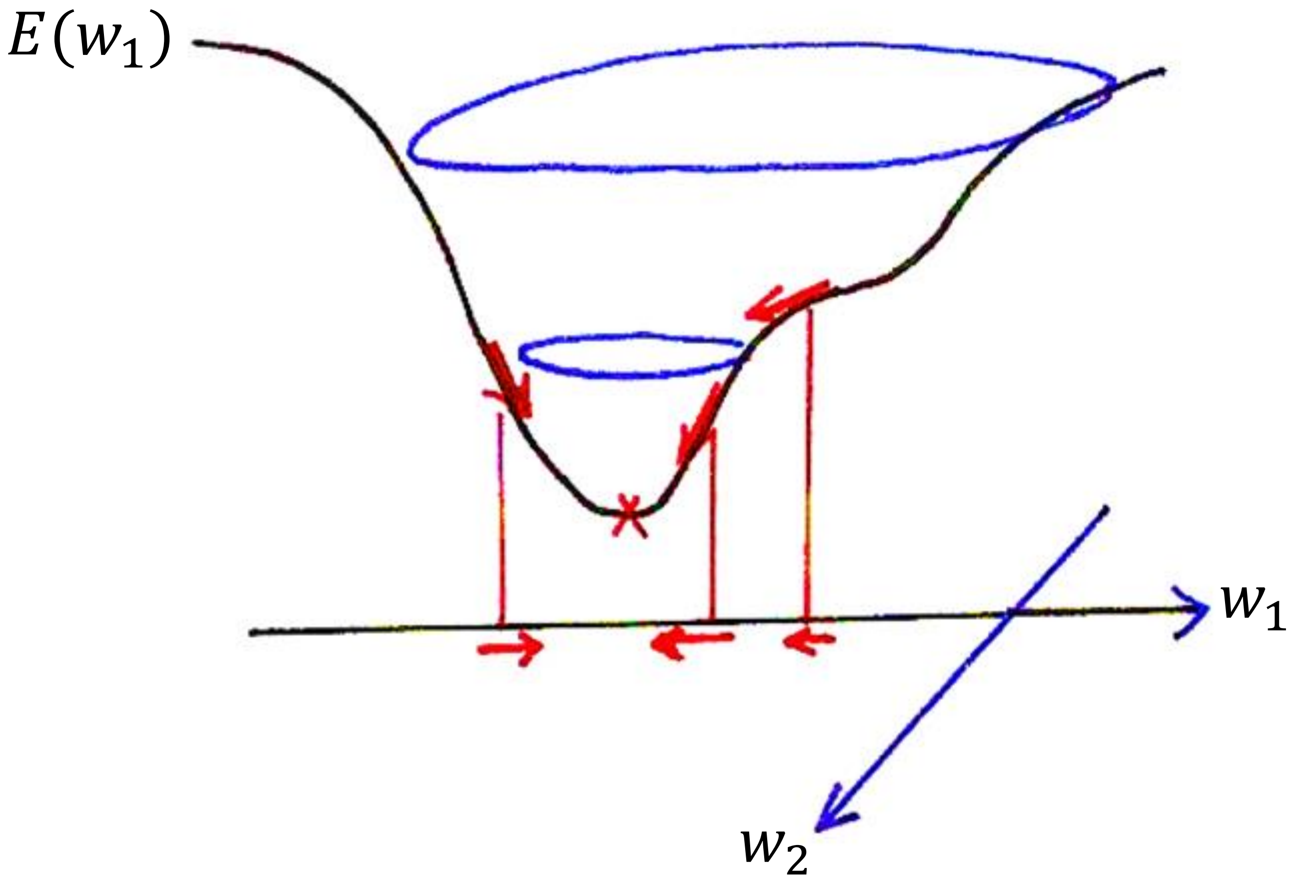
# Neural Network Training – Back Propagation

- **Start with random weights**
- **Compare the outputs of the net to the targets**
- **Try to adjust the weights to minimize the error**

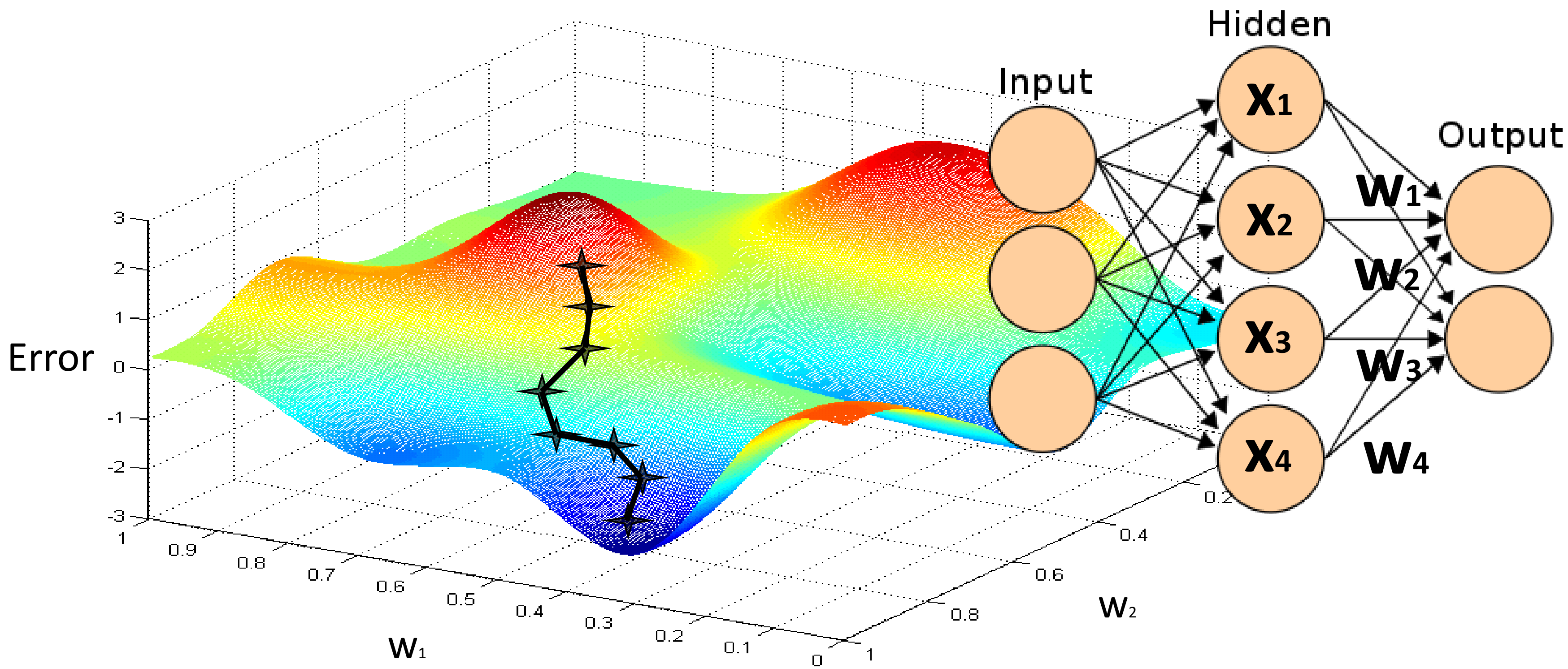



$$E = \frac{1}{2} \sum_{j \in output} (t_j - y_j)^2$$

# Gradient Descent Algorithm



# Gradient Descent Algorithm




$$w_{t+1} = w_t - \alpha \frac{\partial E}{\partial w}$$

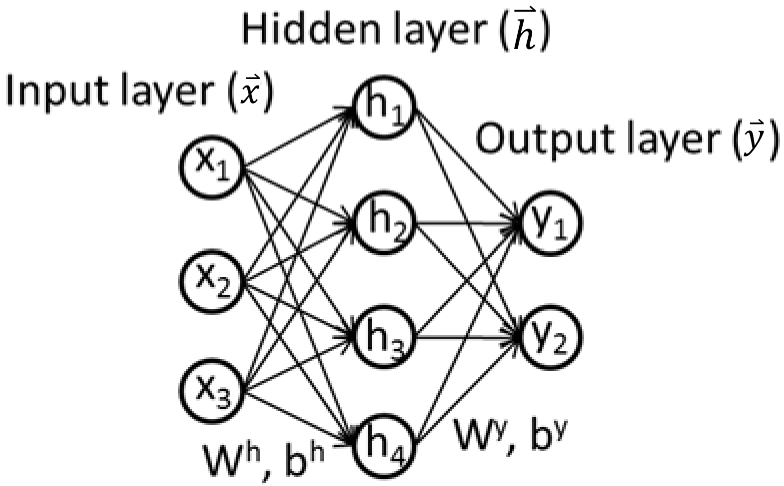
Updated weights

Weight at t-th iteration

Learning rate

# Neural Network — Formal Formulation

- **Neural Network (Multi-Layer Perceptron):**
  - a non-linear statistical modeling tool
  - architecture: input layer $\vec{x}$, hidden layer $\vec{h}$, and output layer $\vec{y}$




$$\vec{h} = f(W^h\vec{x} + b^h)$$
$$\vec{y} = g(W^y\vec{h} + b^y)$$

f,g: non-linear functions

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

$\mathrm{g}(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$ (softmax)

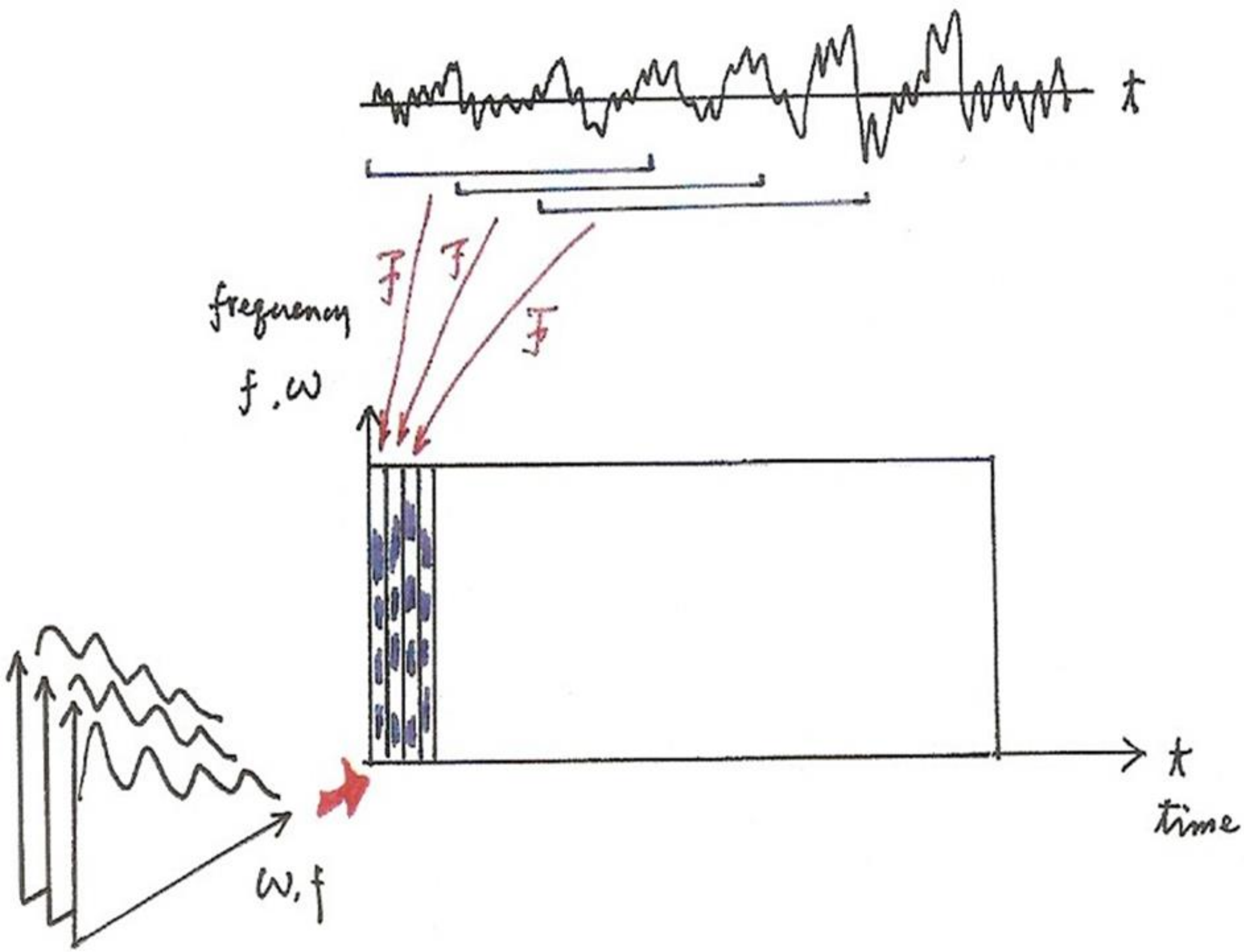
  - $W^h$, $W^y$: weight matrix; $b^h$, $b^y$: bias vector

- **Neural Network Training:**
  - with training examples $(\vec{x}^{(i)}, l^{(i)})$ ($l^{(i)}$: labels)
  - minimize the error function: $E(W^h, W^y, b^h, b^y) = \sum_i ||y^{(i)} - l^{(i)}||^2$
  - back propagation: minimizing the error function by adjusting the parameters applied beforehand

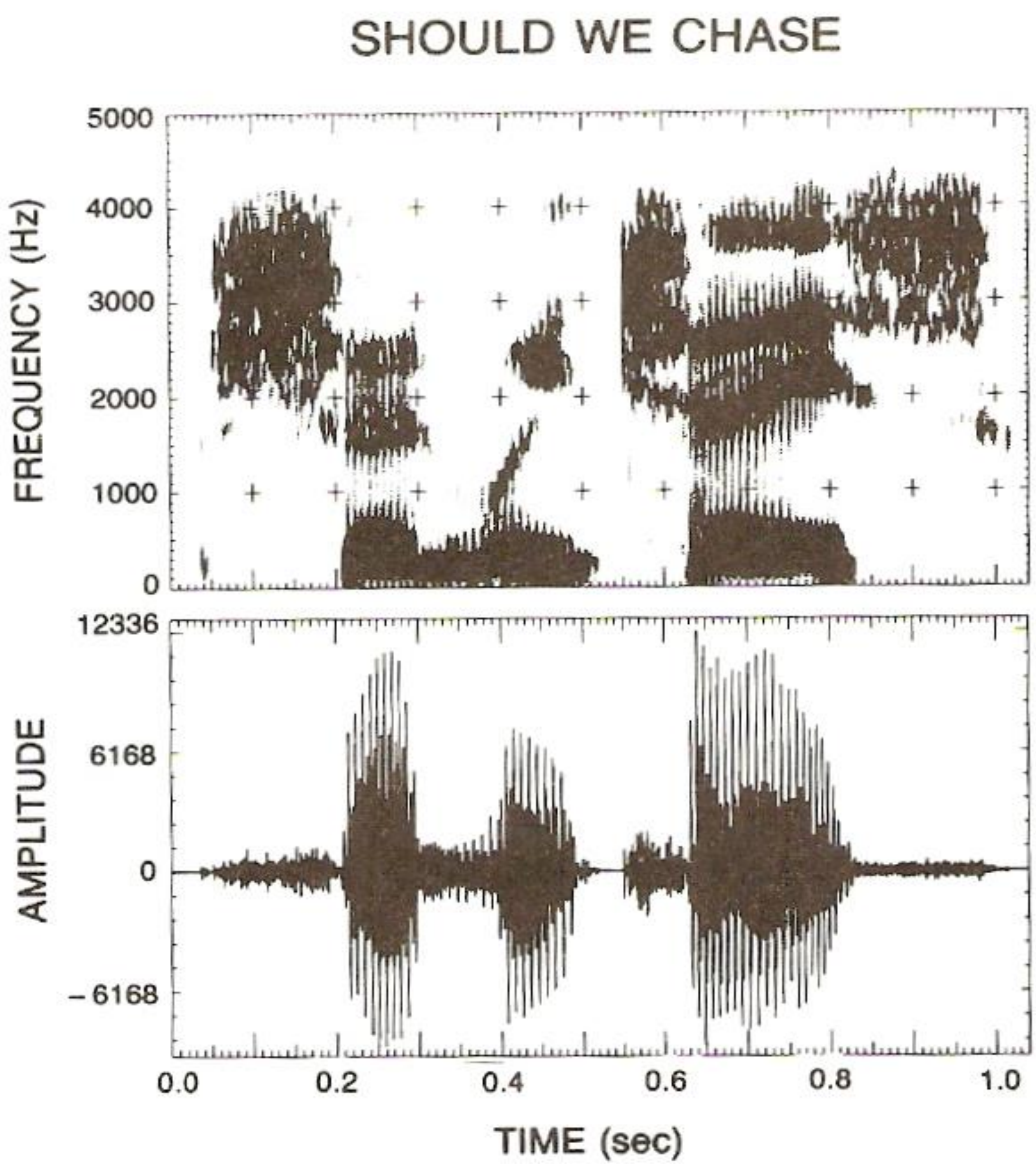
# References for Neural Network


- **Rumelhart, David E.; Hinton, Geoffrey E., Williams, Ronald J. "Learning representations by back-propagating errors". Nature, 1986.**
- **Alpaydın, Ethem. Introduction to machine learning (2nd ed.), MIT Press, 2010.**
- **Albert Nigrin, Neural Networks for Pattern Recognition(1st ed.). A Bradford Book, 1993.**
- **Reference: Neural Networks for Machine Learning course by Geoffrey Hinton, Coursera**

# Spectrogram



# Spectrogram

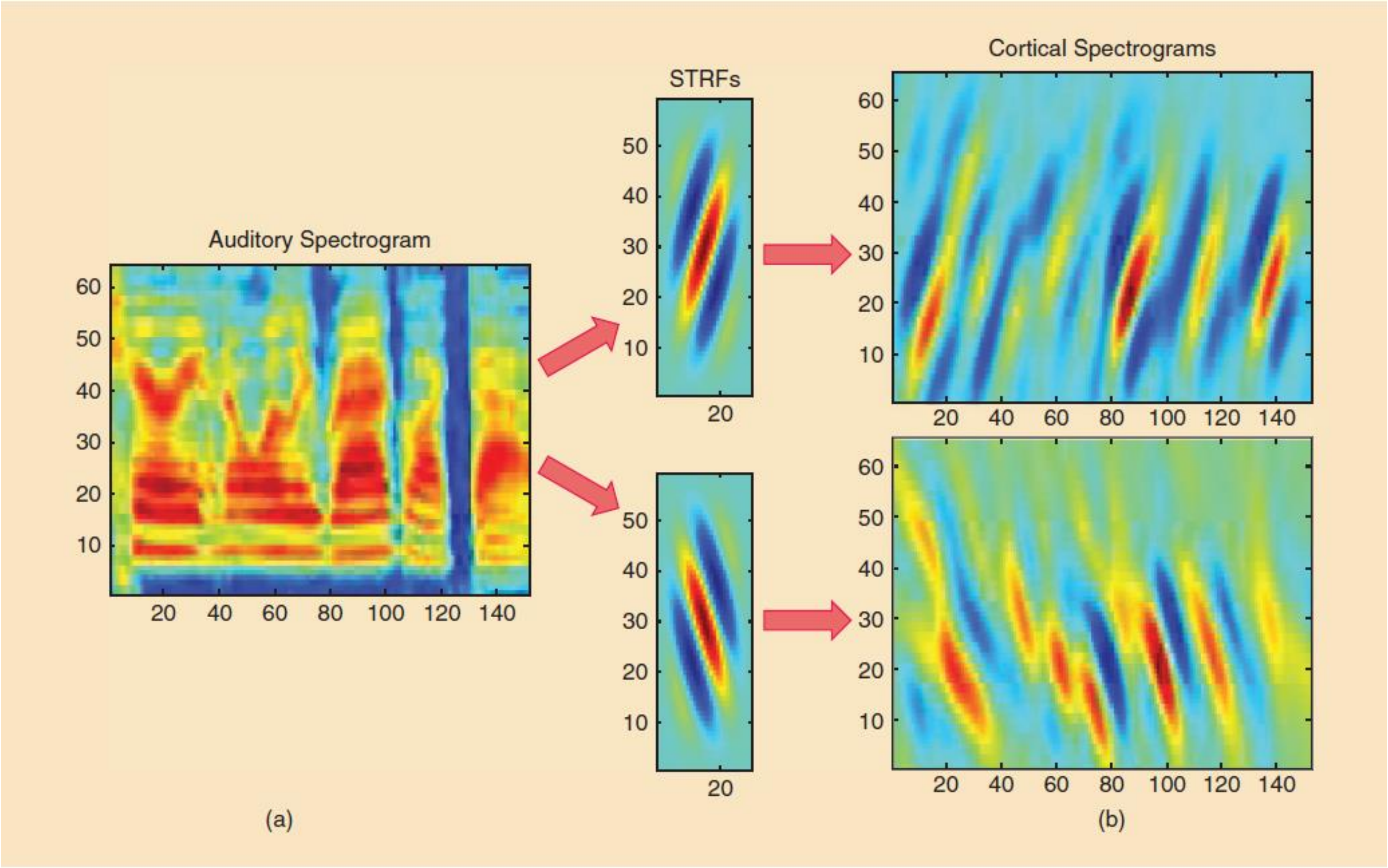


# Gabor Features (1/2)

$$G(t,f)=\frac{1}{2\pi\sigma_f\sigma_t}exp\left[\frac{-(f-f_0)^2}{2\sigma_f^2}+\frac{-(t-t_0)^2}{2\sigma_t^2}\right]exp\left[iw_f(f-f_0)+iw_t(t-t_0)\right]$$

- **2-dim Gabor filters**
  - 2-dim Gaussian multiplied by 2-dim sine waves
  - 2-dim convolution with the 2-dim (mel-) spectrogram
- **Gabor Features**
  - a whole set of features defined by $(f_0, t_0, \sigma_f^2, \sigma_t^2, w_f, w_t)$
  - some of them simulating human perception to some degree
  - spectrogram can be read by human expert in the past
  - how these features are related to sounds represented by speech signals can be learned by machine

# Gabor Features (2/2)

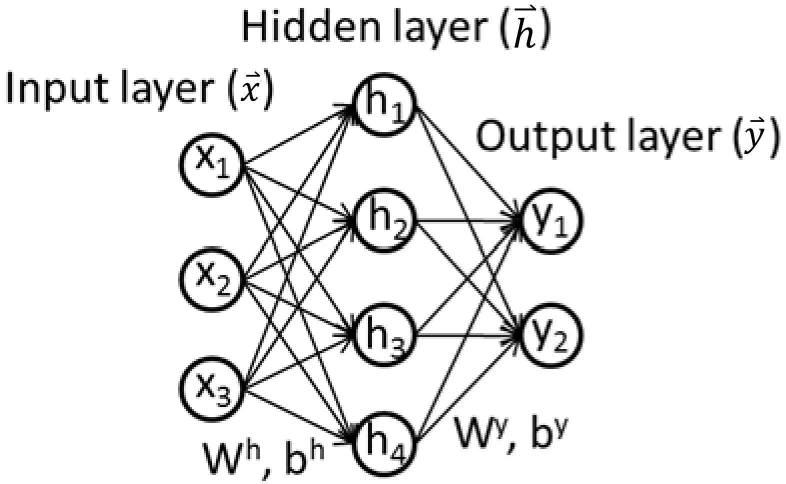


# Integrating HMM with Neural Networks

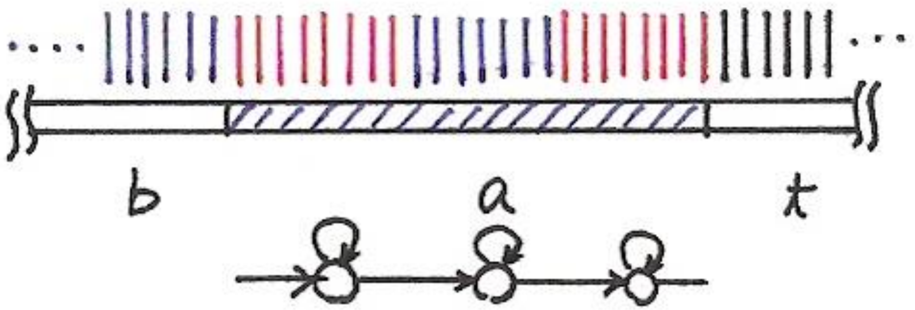
- **Tandem System**
  - Multi-layer Perceptron (MLP, or Neural Network) offers phoneme posterior vectors (posterior probability for each phoneme)
  - MLP trained with known phonemes for MFCC (or plus Gabor) vectors for one or several consecutive frames as target
  - phoneme posteriors concatenated with MFCC as a new set of features for HMM
  - phoneme posterior probabilities may need further processing to be better modeled by Gaussians
- **Hybrid System**
  - Gaussian probabilities in each triphone HMM state replaced by state posteriors for phonemes from MLP trained by feature vectors with known state segmentation

# Phoneme Posteriors and State Posteriors

- **Neural Network Training**

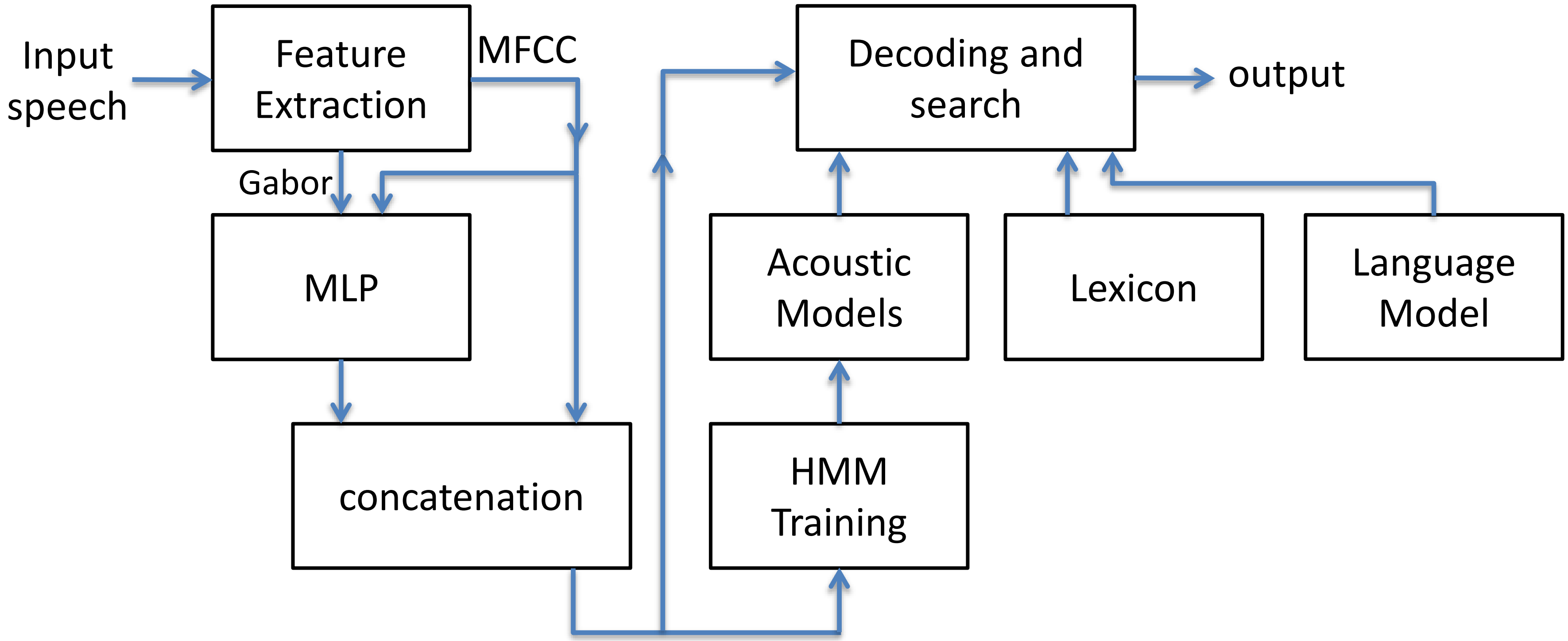



| Phone Posterior | State Posterior |
|---|---|
| $P(a\|x)$ | $P(b\text{–}a(1)\text{–}t\|x)$ |
| $P(b\|x)$ | $P(b\text{–}a(2)\text{–}t\|x)$ |
| ⋮ | ⋮ |



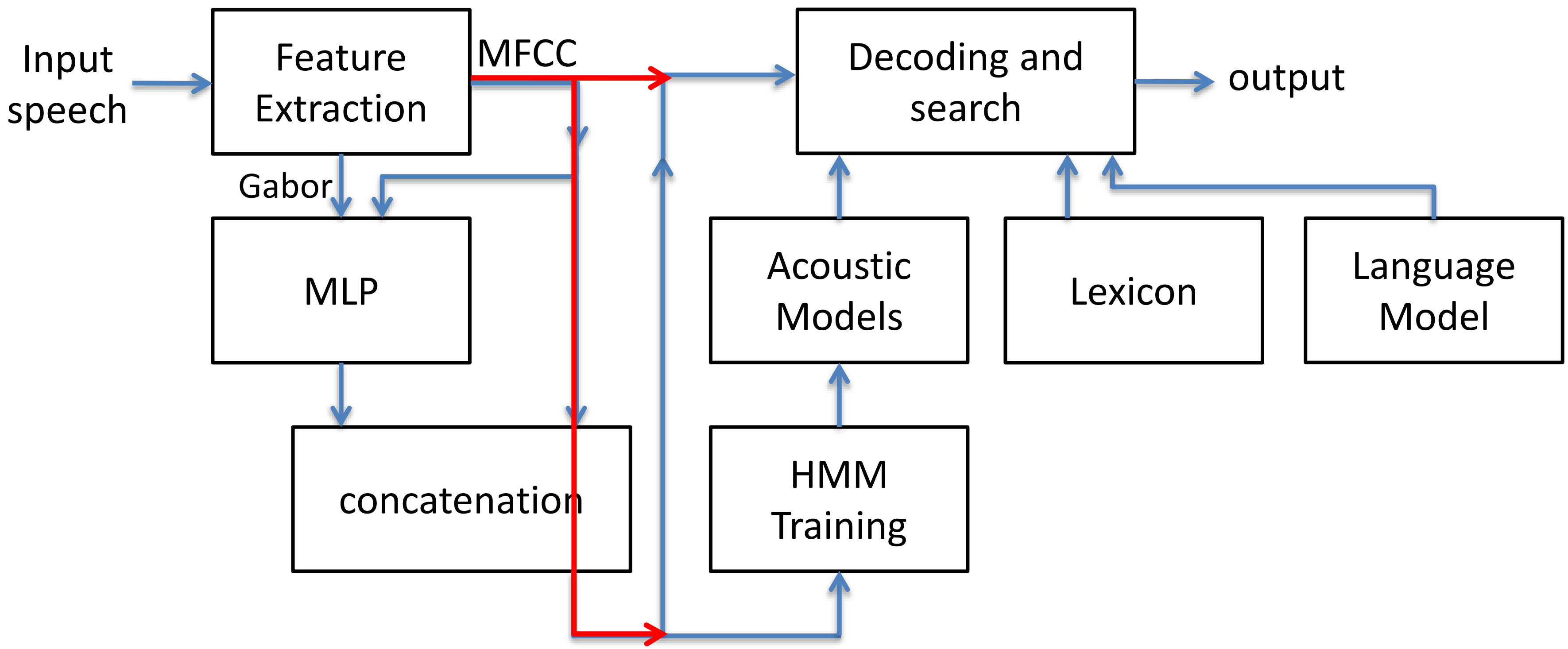
# Integrating HMM with Neural Networks

- **Tandem System**
  - phoneme posterior vectors from MLP concatenated with MFCC as a new set of features for HMM



# Integrating HMM with Neural Networks

- **Tandem System**
  - phoneme posterior vectors from MLP concatenated with MFCC as a new set of features for HMM



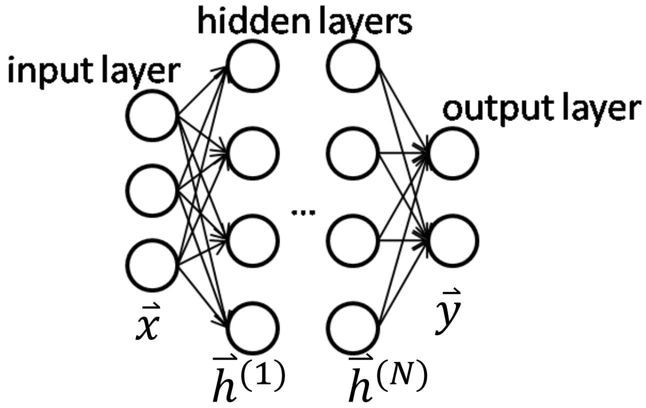
# References

- **References for Gabor Features and Tandem System**


  - Richard M. Stern & Nelson Morgan, “Hearing Is Believing”, IEEE SIGNAL PROCESSING MAGAZINE, NOVEMBER 2012
  - Hermansky, H., Ellis, D.P.W., Sharma, S., “Tandem Connectionist Feature Extraction For Conventional Hmm Systems”, in Proc. ICASSP 2000.
  - Ellis, D.P.W. and Singh, R. and Sivadas, S., “Tandem acoustic modeling in large-vocabulary recognition”, in Proc. ICASSP 2001.
  - “Improved Tonal Language Speech Recognition by Integrating Spectro-Temporal Evidence and Pitch Information with Properly Chosen Tonal Acoustic Units”, Interspeech, Florence, Italy, Aug 2011, pp. 2293-2296.

# Deep Neural Network (DNN)

- **Deep Neural Network (DNN):**
  - Neural network with multiple hidden layers
  - architecture: with input $\vec{x}$, N hidden layers and output $\vec{y}$



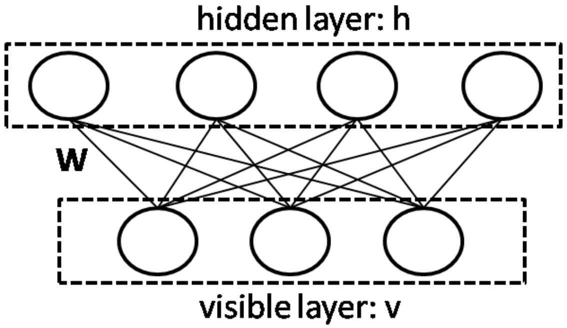

$$\vec{h}^{(1)} = f(W_{0,1}\vec{x} + b_{0,1})$$
$$\vec{h}^{(n)} = f(W_{n-1,n}\vec{h}^{(n-1)} + b_{n-1,n})$$
$$\vec{y} = g(W_{N,N+1}\vec{h}^{(N)} + b_{N,N+1})$$

- **Property:**
  - able to deal with huge and complicated structure of data

- **Difficulties:**
  - large quantities of labelled data needed for training
  - very long training time needed
  - solution: Restricted Boltzmann Machine for initialization

# Restricted Boltzmann Machine

- **Restricted Boltzmann Machine (RBM):**
  - a generative model for probability of visible examples (p(v))
  - with a hidden layer of random variables (h)
  - topology: undirected bipartite graph




$$p(v,h) = \frac{1}{Z} e^{-E(v,h)}$$

$$E(v,h) = -a^T v - b^T h - v^T W h$$

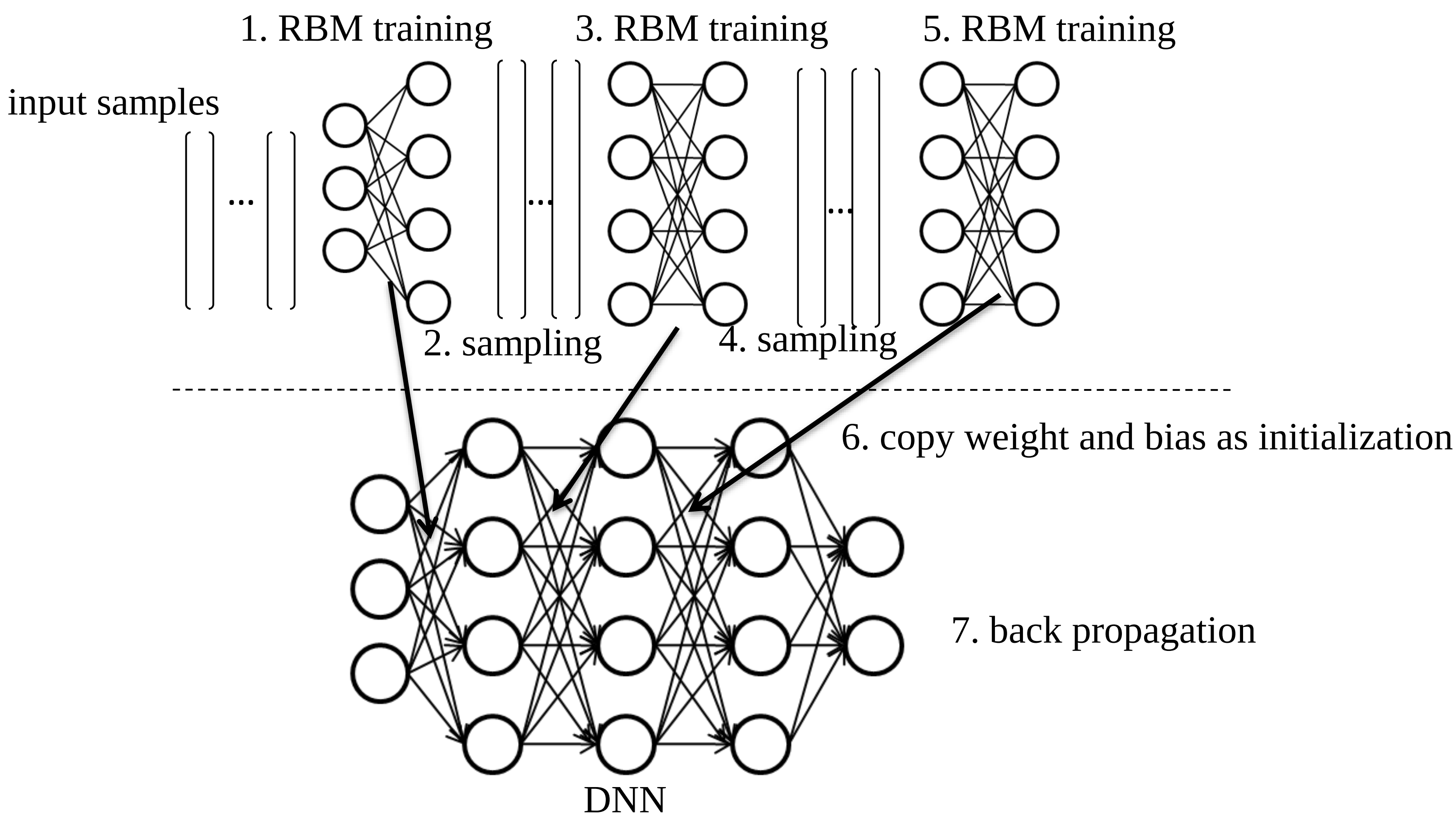
$$p(v) = \frac{1}{Z} \sum_h e^{-E(v,h)}$$

  - W: weight matrix, describing correlation between visible and hidden layers
  - a, b: bias vectors for visible and hidden layers
  - E: energy function for a (v,h) pair
  - RBM training: adjusting W, a, and b to maximize p(v)

- **Property:**
  - finding a good representation (h) for v in unsupervised manner
  - Using large quantities of unlabelled data

# RBM Initialization for DNN Training

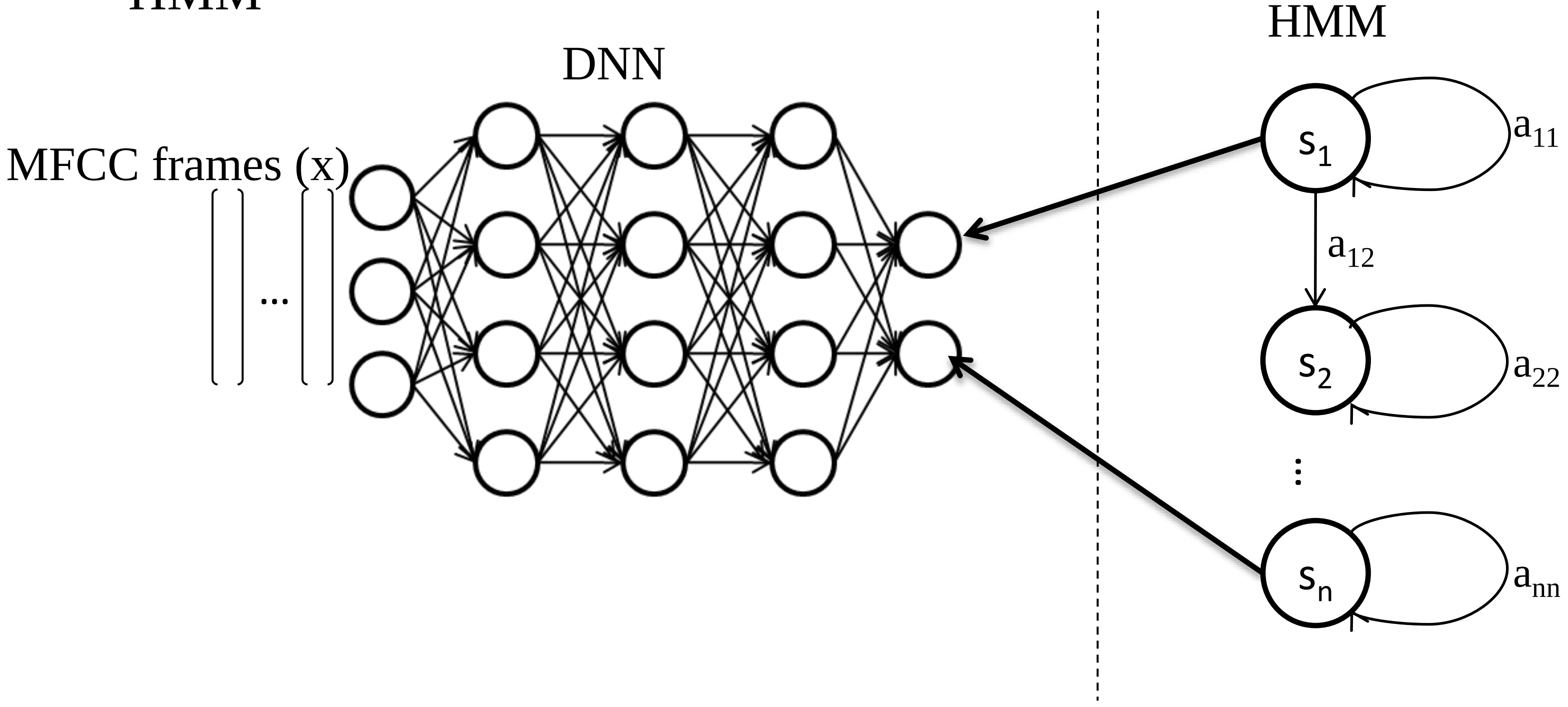
- **RBM Initialization**
  - weight matrices of DNN initialized by weight matrixes of RBMs
  - after training an RBM, generate samples in hidden layer used for next layer of RBM
  - steps of initialization (e.g. 3 hidden layers)



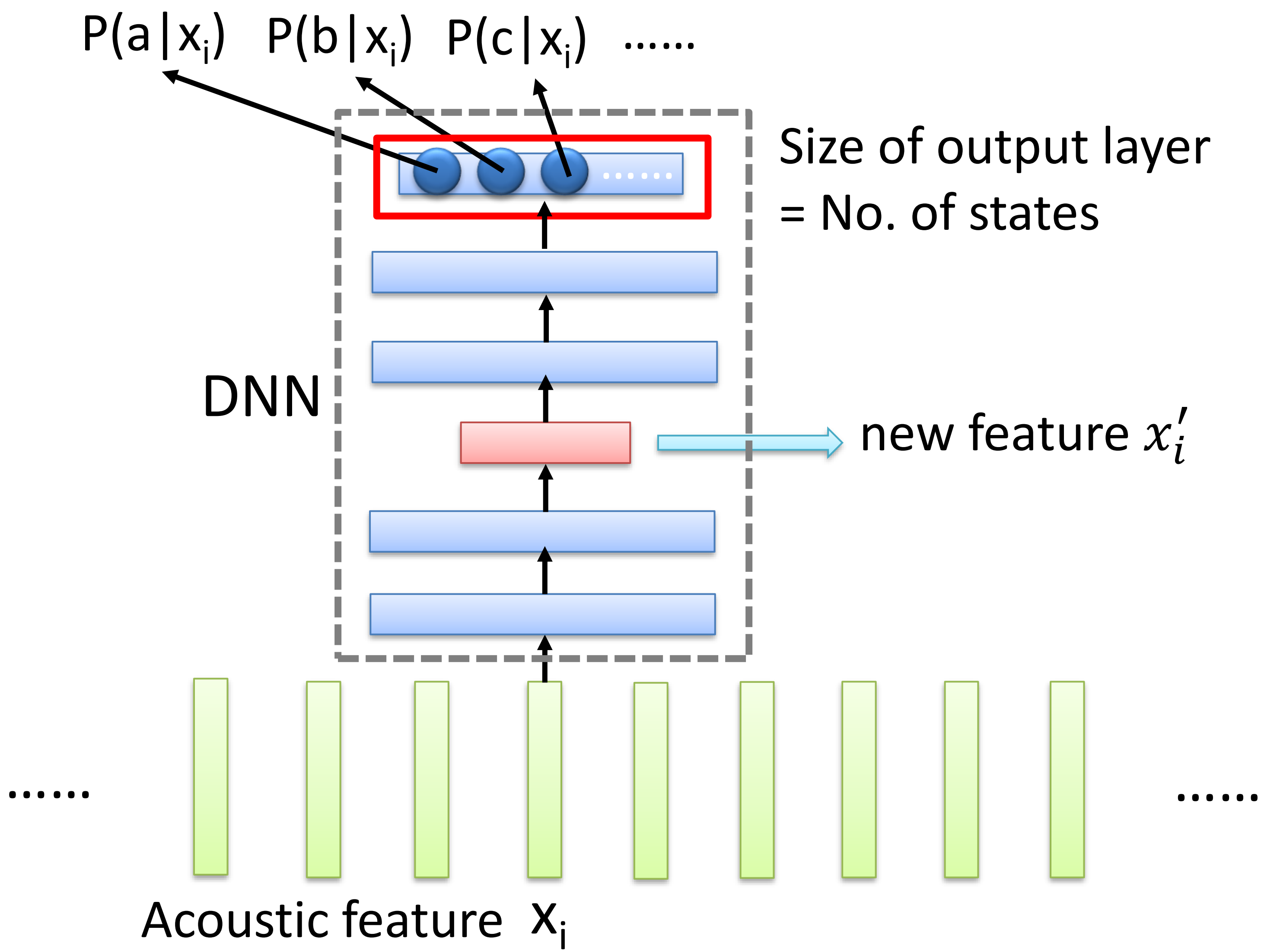
# Deep Neural Network for Acoustic Modeling

- **DNN as triphone state classifier**
  - input: acoustic features, e.g. MFCC
  - output layer of DNN representing triphone states
  - fine tuning the DNN by back propagation using labelled data

- **Hybrid System**
  - normalized output of DNN as posterior of states p(s|x)
  - state transition remaining unchanged, modeled by transition probabilities of HMM



# Bottleneck Features from DNN

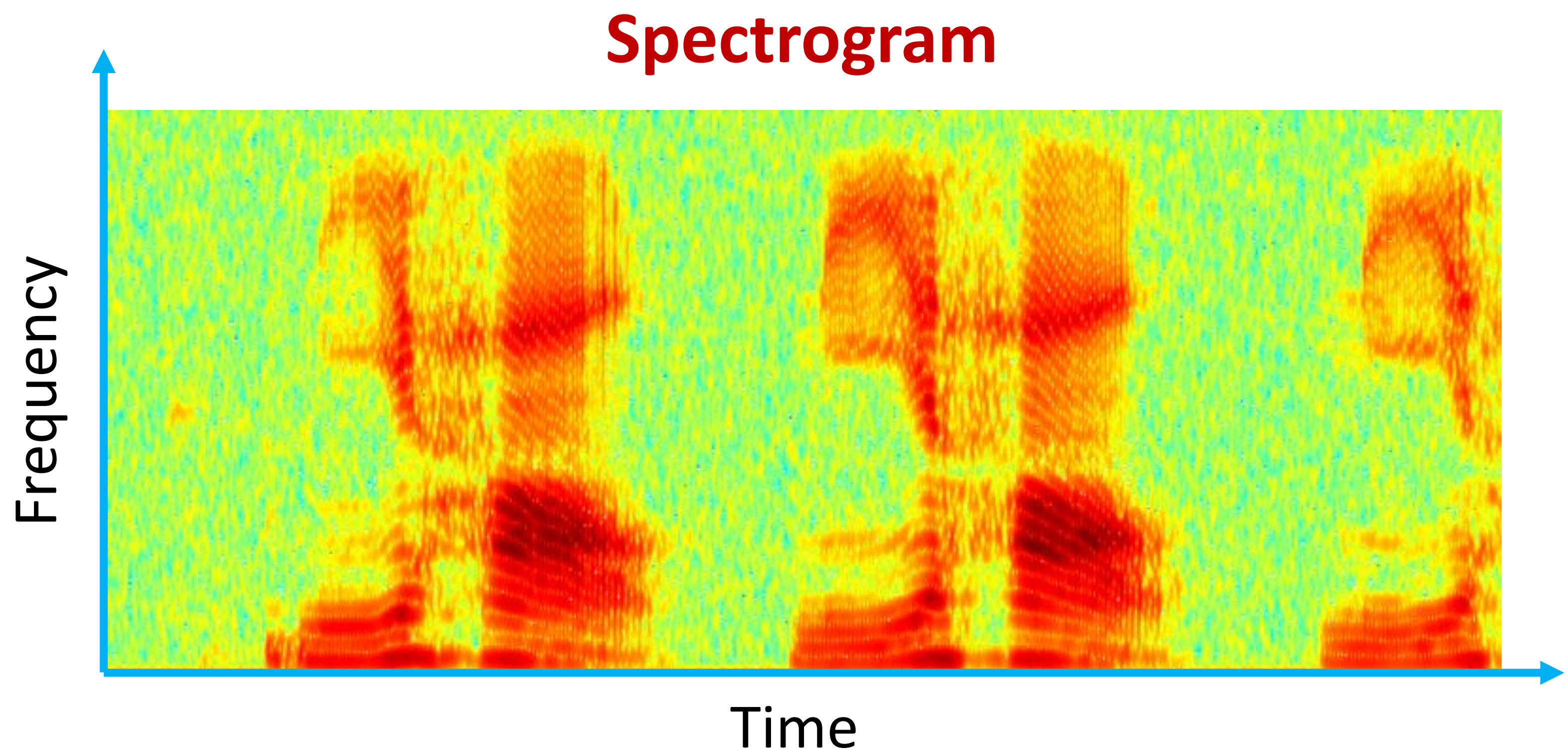


# References for DNN

- **Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition**
  - George E. Dahl, Dong Yu, Deng Li, and Alex Acero
  - IEEE Trans. on Audio, Speech and Language Processing, Jan, 2012
- **A fast learning algorithm for deep belief**
  - Hinton, G. E., Osindero, S. and Teh, Y
  - Neural Computation, 18, pp 1527-1554, 2006
- **Deep Neural Networks for Acoustic Modeling in Speech Recognition**
  - G. Hinton, L. Deng, D. Yu, G. Dahl, A.Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury
  - IEEE Signal Processing Magazine, 29, November 2012
- **Deep Learning and Its Applications to Signal and Information Processing**
  - IEEE Signal Processing Magazine, Jan 2011
- **Improved Bottleneck Features Using Pretrained Deep Neural Networks**
  - Yu, Dong, and Michael L. Seltzer
  - Interspeech 2011
- **Extracting deep bottleneck features using stacked auto-encoders**
  - Gehring, Jonas, et al.
  - ICASSP 2013

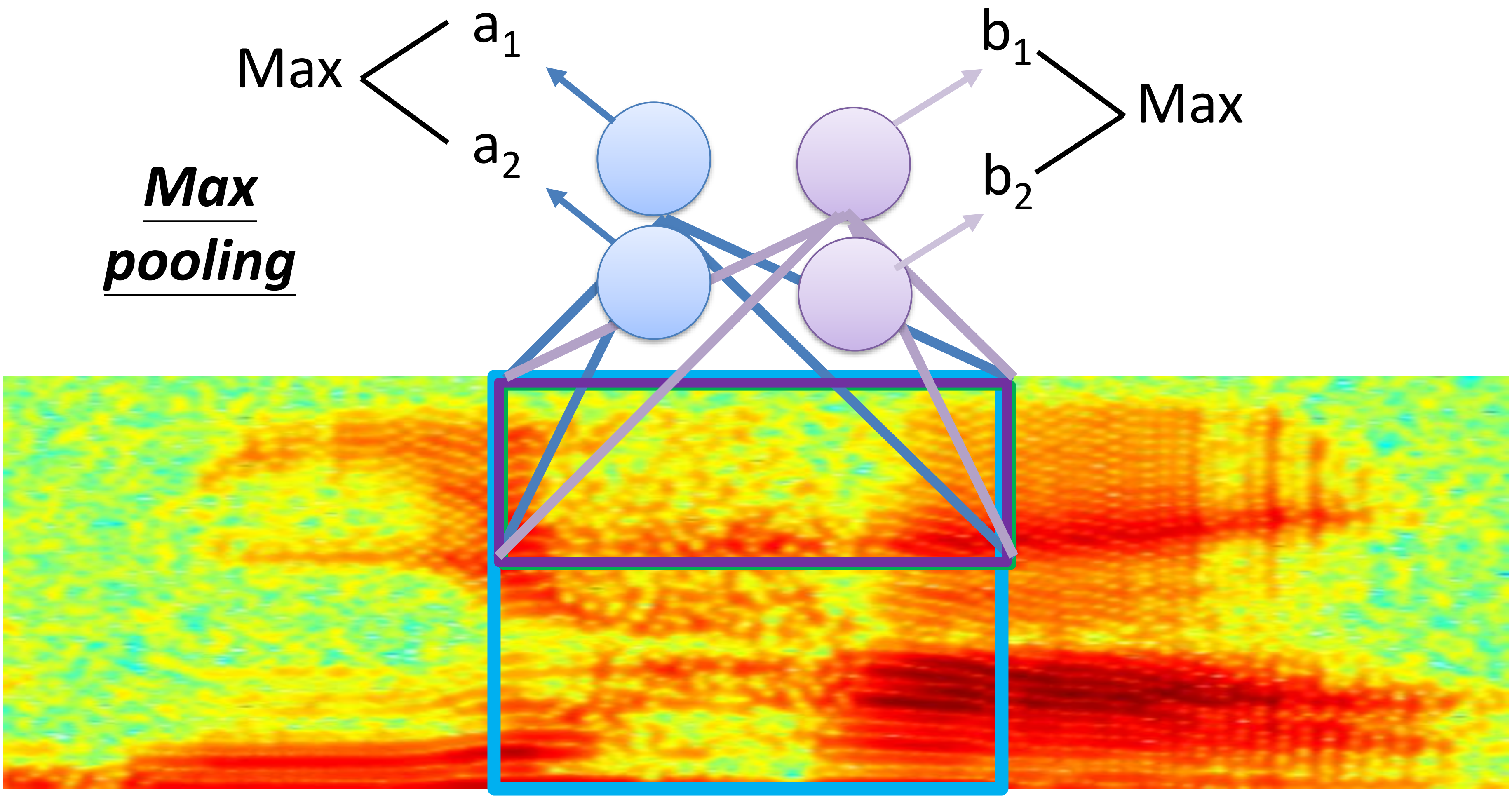
# Convolutional Neural Network (CNN)

- Successful in processing images
- Speech can be treated as images



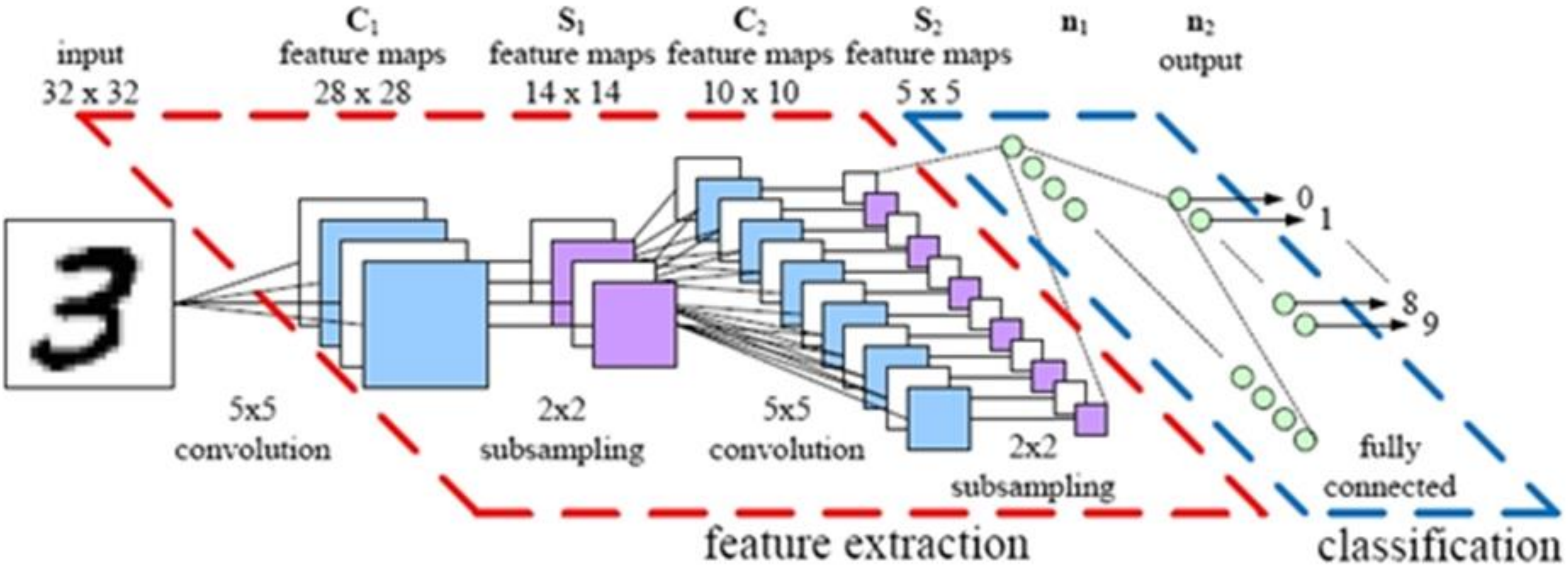
# Convolutional Neural Network (CNN)

- An example



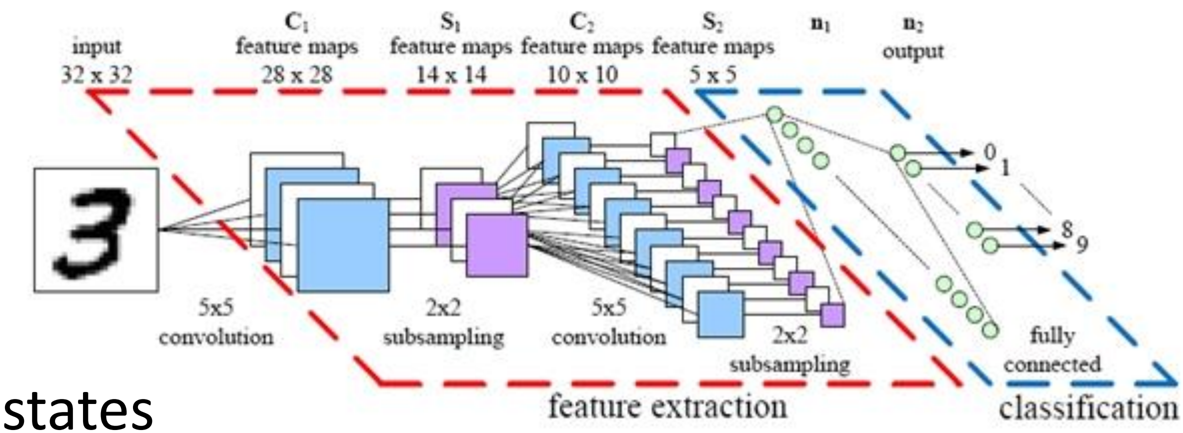
# Convolutional Neural Network (CNN)

- An example



# Convolutional Neural Network (CNN)

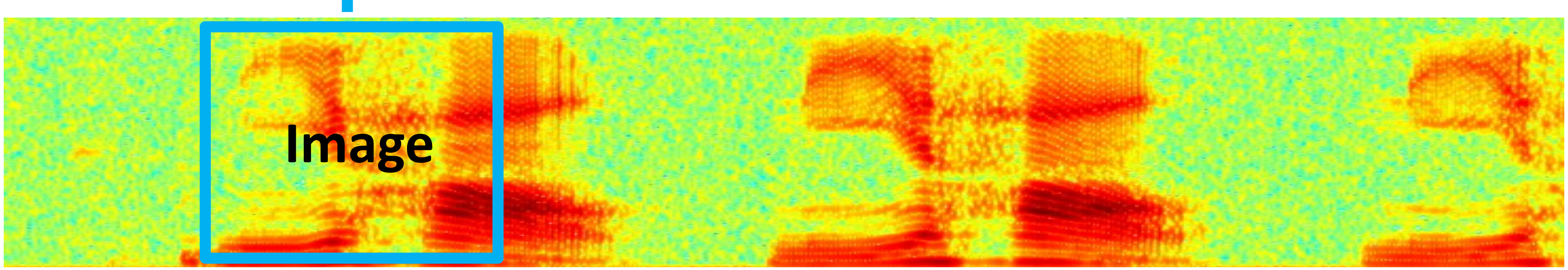
- An example



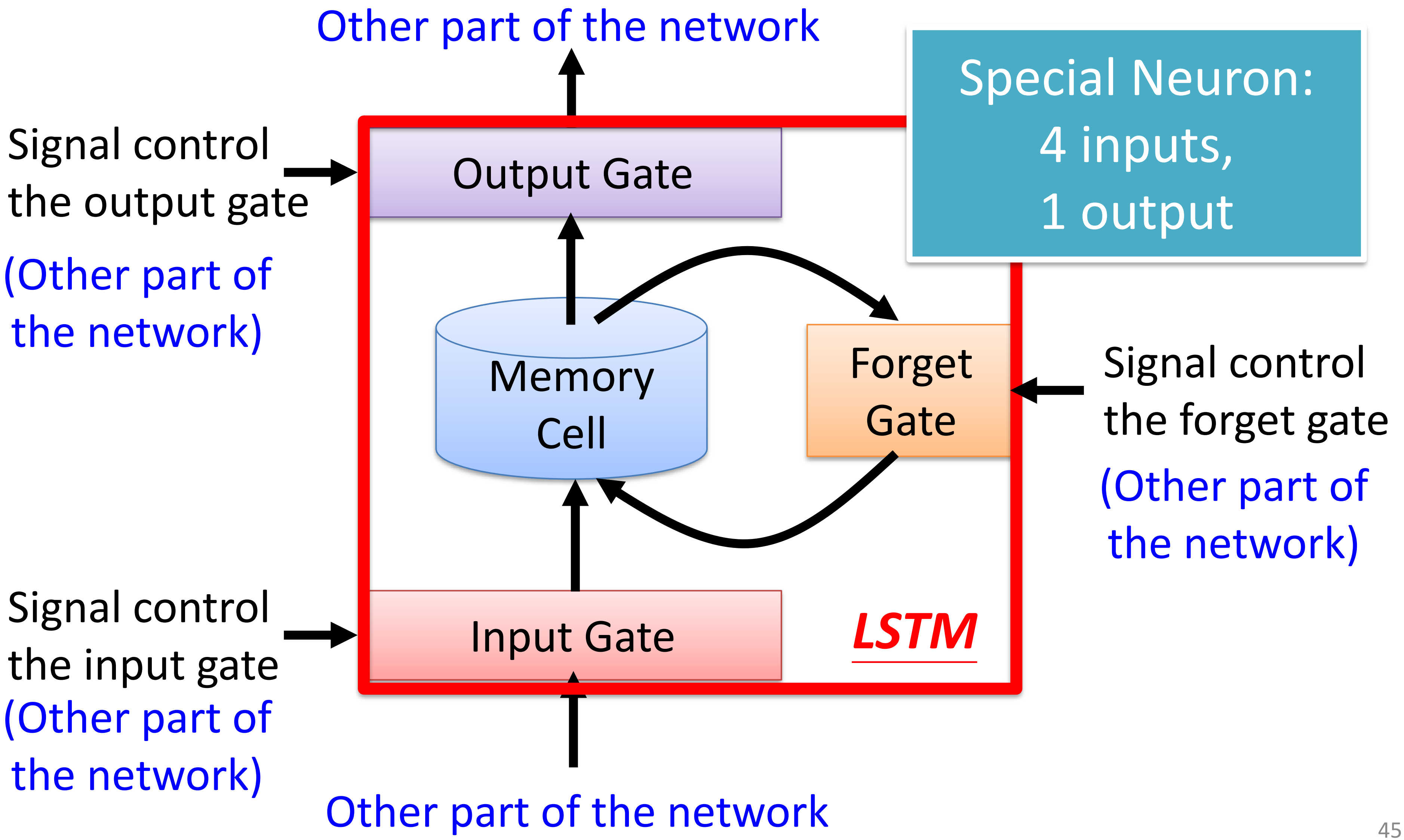

Probabilities of states

CNN

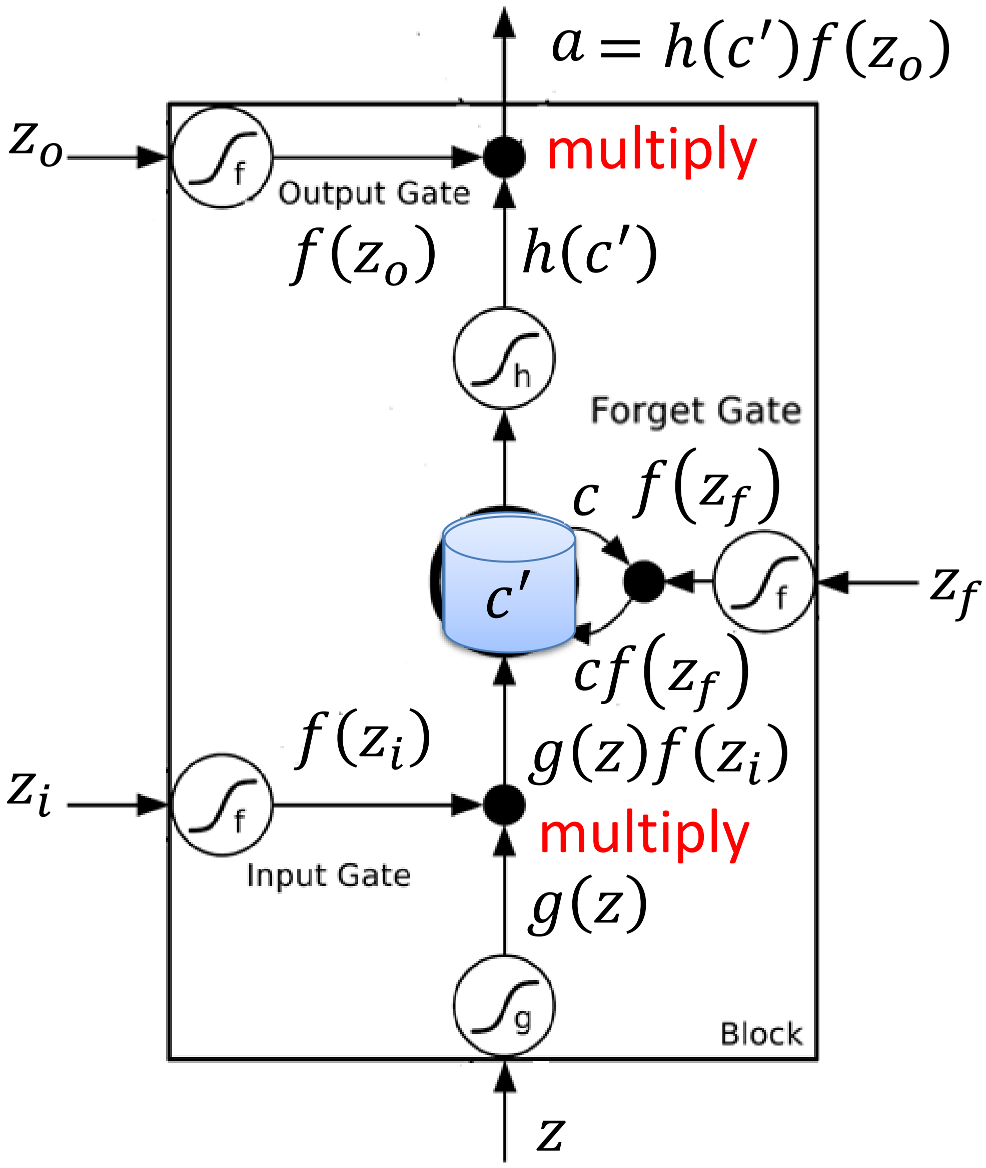
Replace DNN by CNN



# Long Short-term Memory (LSTM)



# Long Short-term Memory (LSTM)




Activation function f(·) is usually a sigmoid function

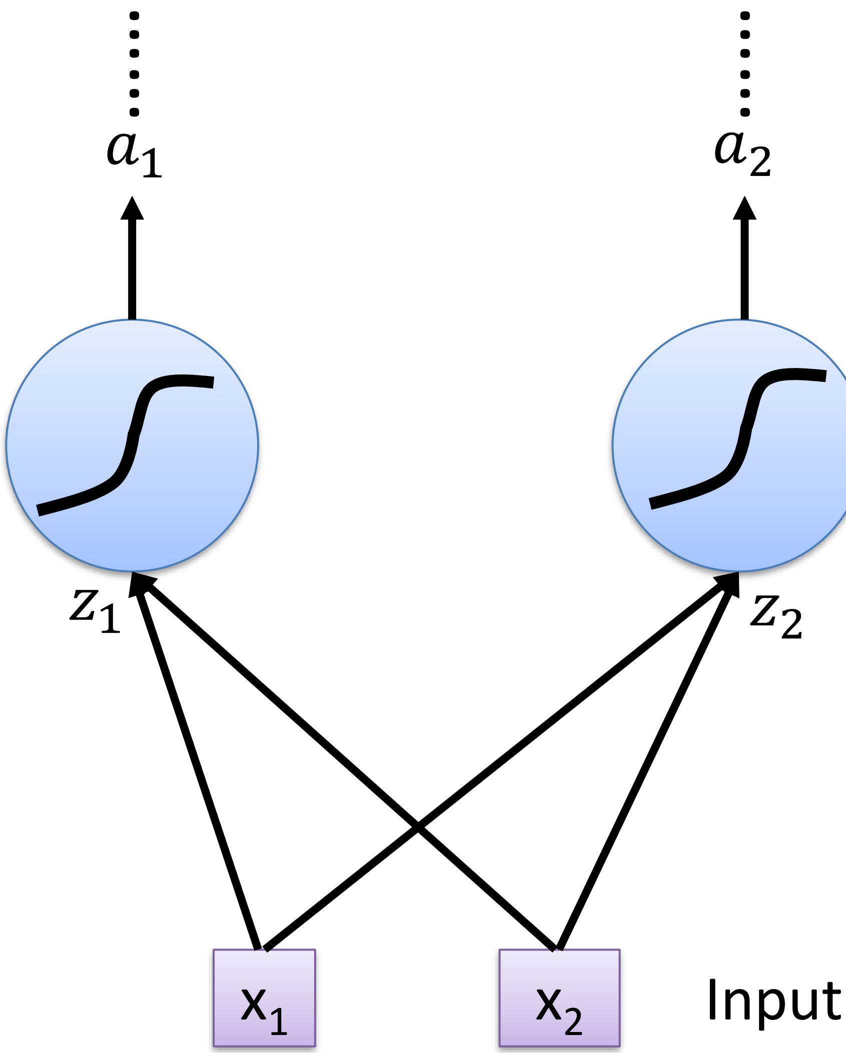
between 0 and 1 for

opening and closing the gate

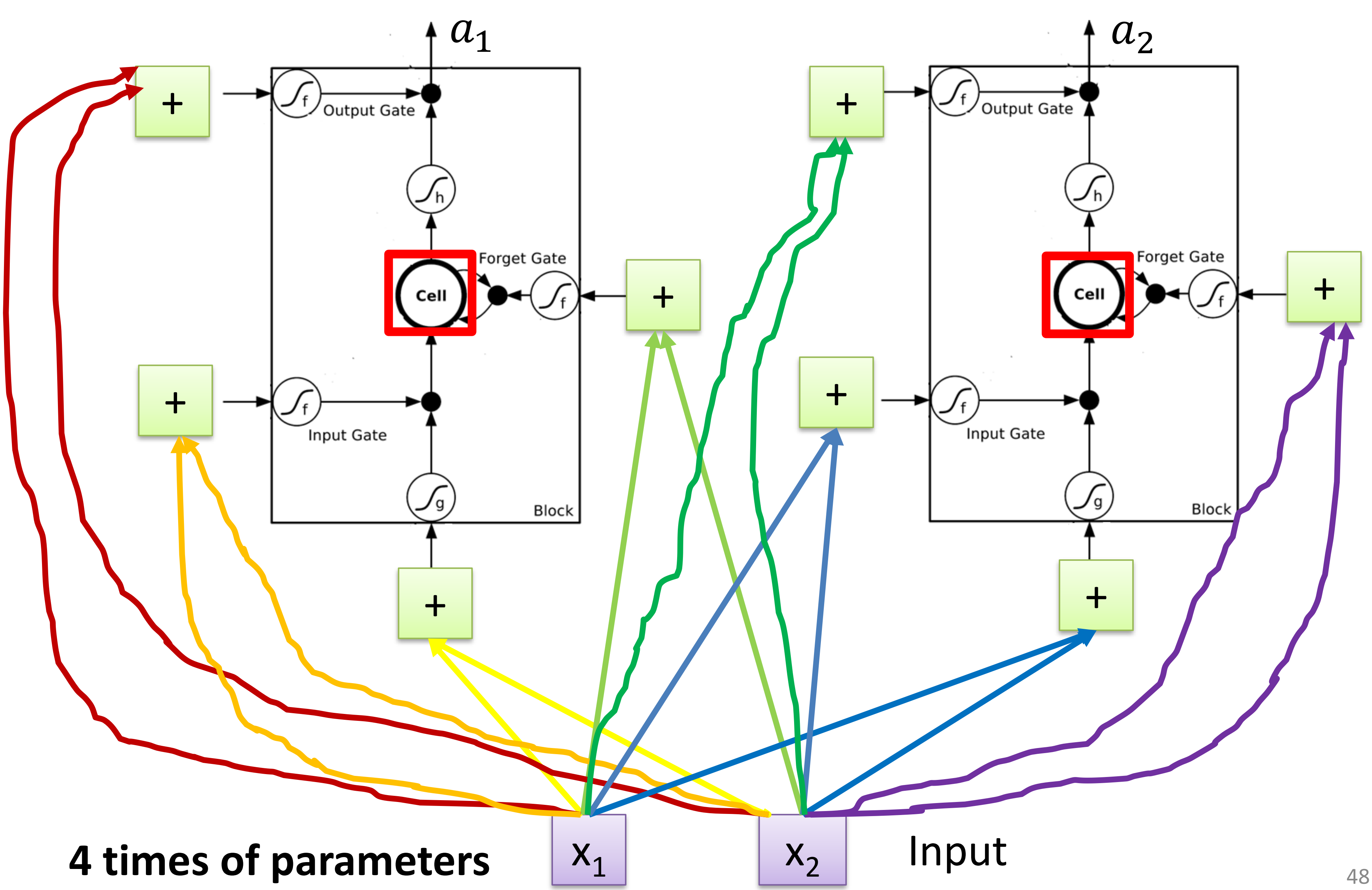
$$c' = g(z)f(z_i) + cf(z_f)$$

# Long Short-term Memory (LSTM)

- **Simply replacing the neurons with LSTM**
  - original network



# Long Short-term Memory (LSTM)




**4 times of parameters**

# References

## Convolutional Neural Network (CNN)


- Convolutional Neural Network for Image processing
  - Zeiler, M. D., & Fergus, R. (2014). “Visualizing and understanding convolutional networks.” In Computer Vision–ECCV 2014
- Convolutional Neural Network for speech processing
  - Tóth, László. "Convolutional deep maxout networks for phone recognition." Proc. Interspeech. 2014.
- Convolutional Neural Network for text processing
  - Shen, Yelong, et al. "A latent semantic model with convolutional-pooling structure for information retrieval." Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, 2014.


## Long Short-term Memory (LSTM)


- Graves, N. Jaitly, A. Mohamed. “Hybrid Speech Recognition with Deep Bidirectional LSTM”, ASRU 2013.
- Graves, Alex, and Navdeep Jaitly. "Towards end-to-end speech recognition with recurrent neural networks." Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014.

# Neural Network Language Modeling

- **Input words represented by 1-of-N encoding**

$$\underbrace{[\ 0\ 0\ 0 \cdots 0\ 1\ 0\ 0 \cdots 0\ ]}_{\text{vocabulary size}}$$

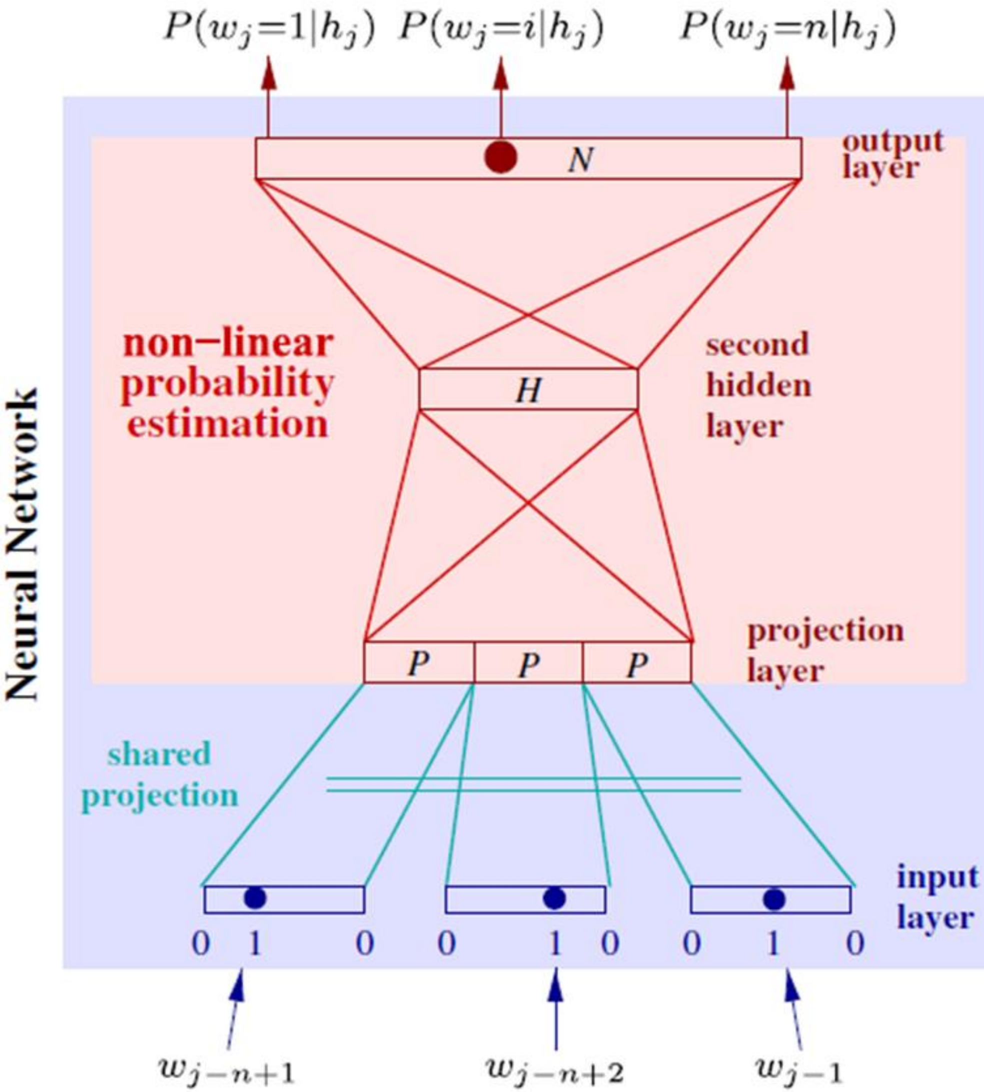
- **Output layer gives the probabilities of words given the history**

$$\text{Prob}\left[\, w_j = i \,\middle|\, h_j \,\right]$$

- **Example:**

P=120, H=800

- **Continuous space language modeling**



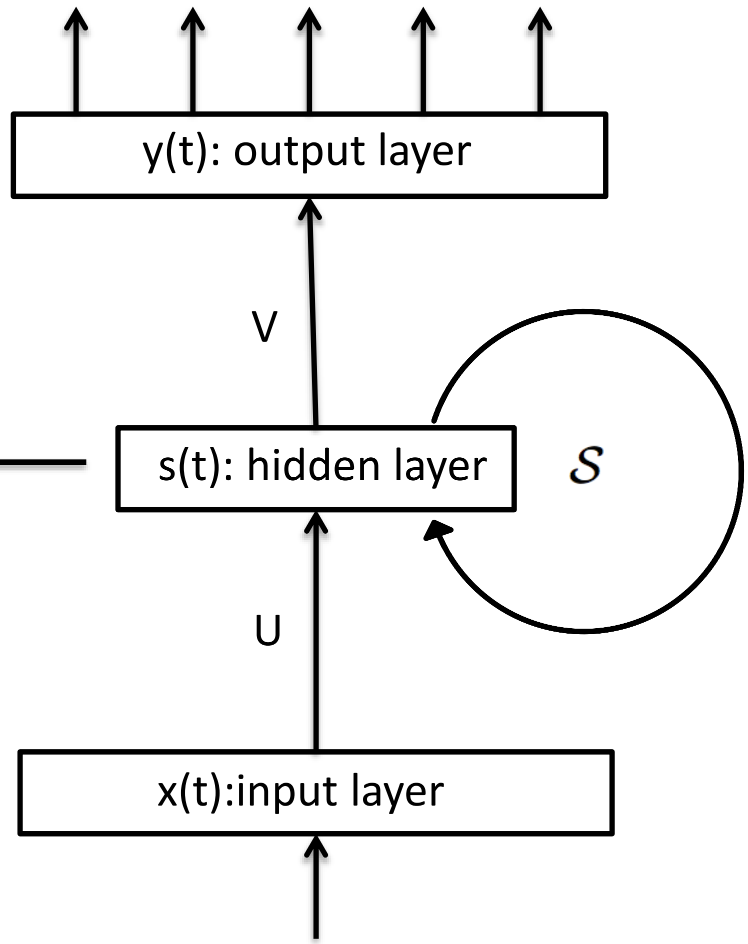
# Recurrent Neural Network Language Modeling(RNNLM)

Probability distribution of next word, vocabulary size.
using softmax: $g(z_k) = \frac{e^{z_k}}{\sum_k e^{z_k}}$

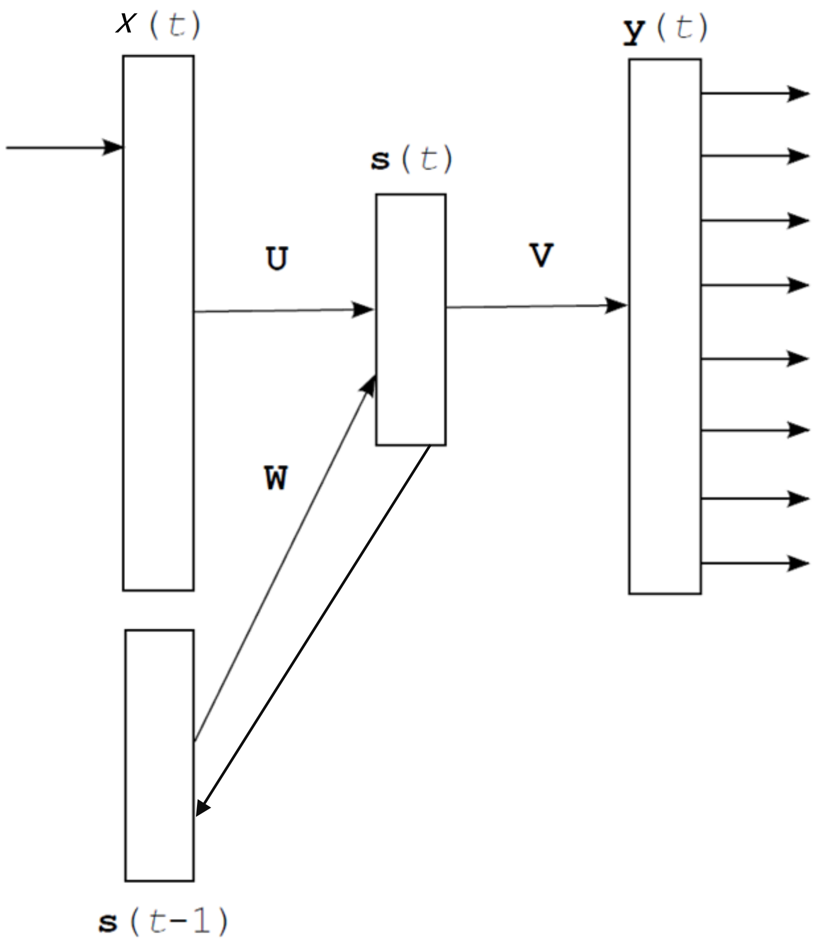
Recursive structure preserves long-term historical context.
using logic unit: $f(z) = \frac{1}{1+e^{-z}}$

Previous word, using 1-of-N encoding

$$\underbrace{[\ 0\ 0\ 0\ \ldots\ldots\ 0\ 0\ 1\ 0\ 0\ 0\ \ldots\ ]}_{\text{Vocab. size}}$$



# RNNLM Structure



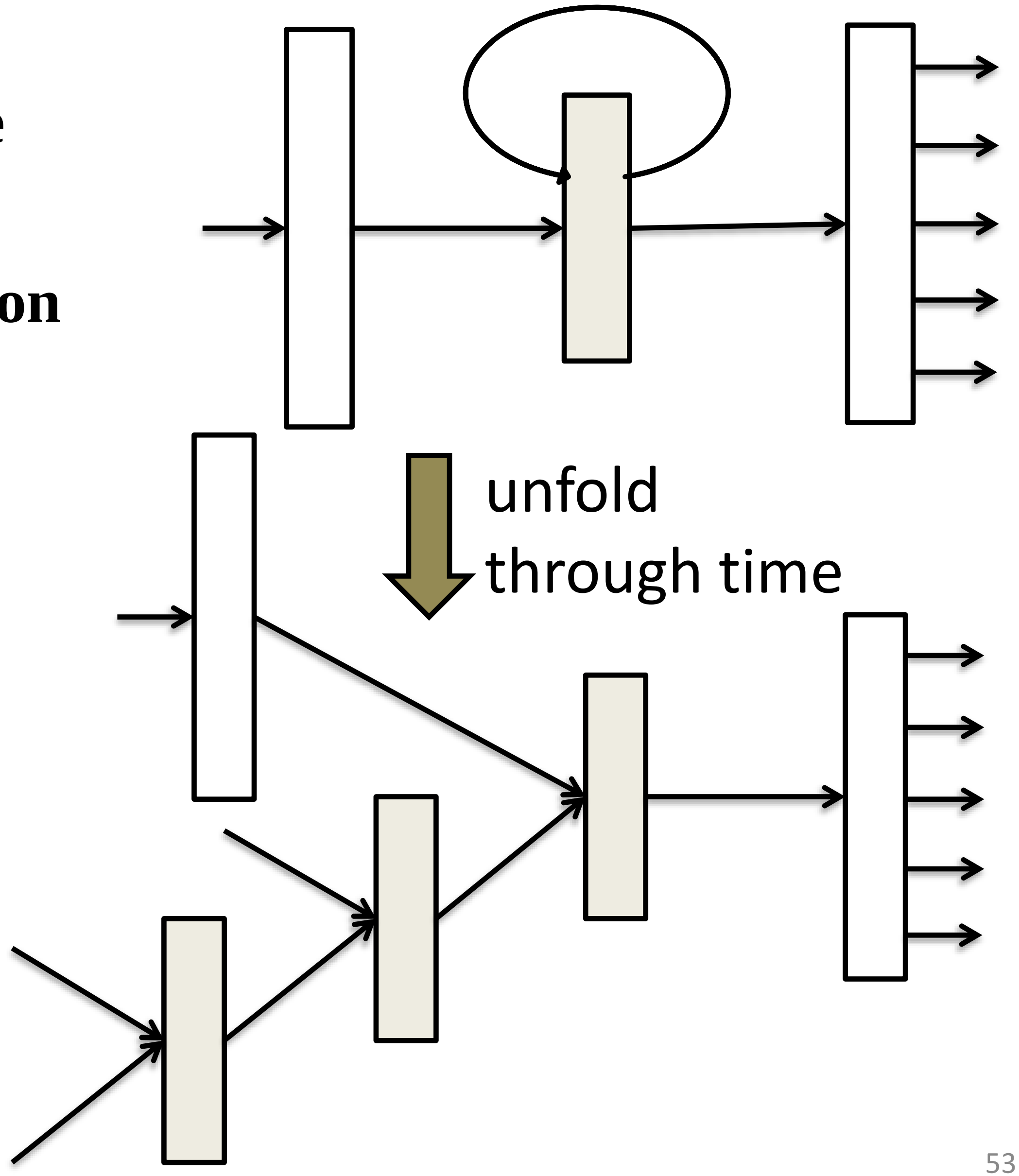

$$s_j(t) = f\left(\sum_i x_i(t)\, u_{ji} + \sum_l s_l(t-1)\, w_{jl}\right)$$

$$y_k(t) = g\left(\sum_j s_j(t)\, v_{kj}\right)$$

$$f(z) = \frac{1}{1+e^{-z}}, \quad g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

# Back propagation for RNNLM

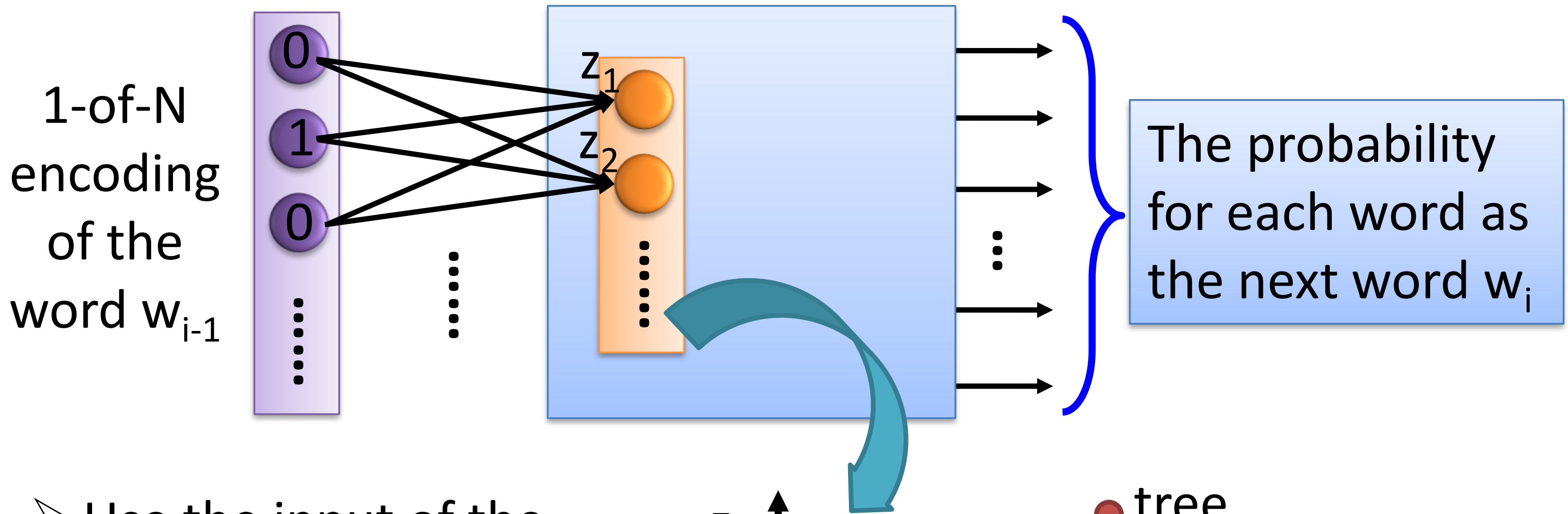
1. Unfold recurrent structure
2. Input one word at a time
3. Do normal back propagation



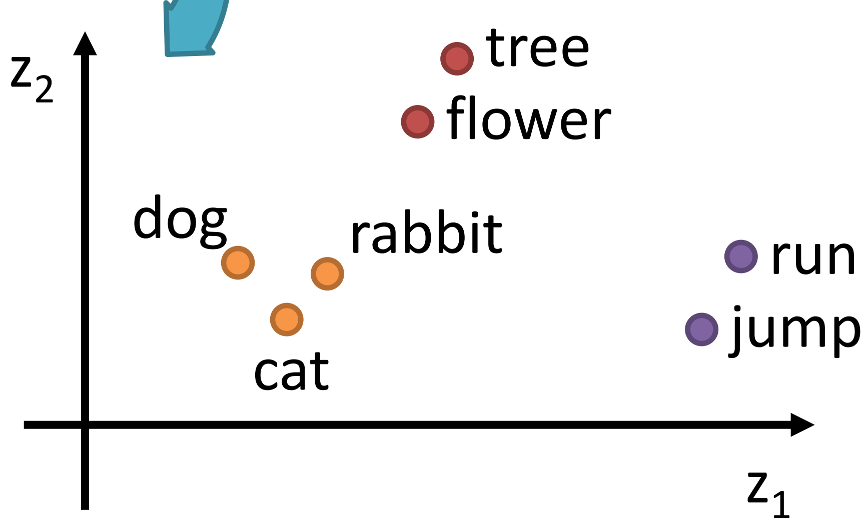
# References for RNNLM


- Yoshua Bengio, Rejean Ducharme and Pascal Vincent. "**A neural probabilistic language model**," *Journal of Machine Learning Research*, 3:1137–1155, 2003
- Holger Schwenk. "**Continuous space language models**," *Computer Speech and Language*, vol. 21, pp. 492–518, 2007
- Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký and Sanjeev Khudanpur. "**Recurrent neural network based language model**," in *Interspeech 2010*
- Mikolov Tomáš et al, "Extensions of Recurrent Neural Network Language Model", ICASSP 2011.
- Mikolov Tomáš et al, "Context Dependent Recurrent Neural Network Language Model", IEEE SLT 2012.

# Word Vector Representations (Word Embedding)

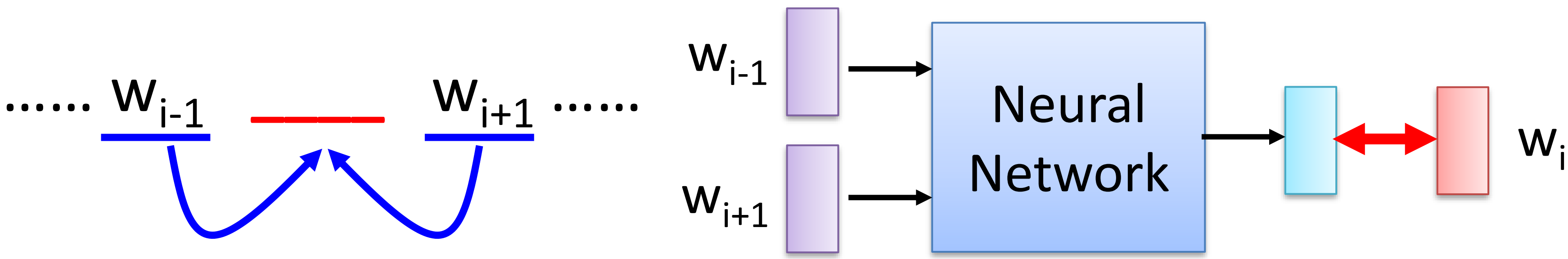



- Use the input of the neurons in the first layer to represent a word w
- Word vector, word embedding feature: V(w)
- Word analogy task: (king)-(man)+(woman)→(queen)



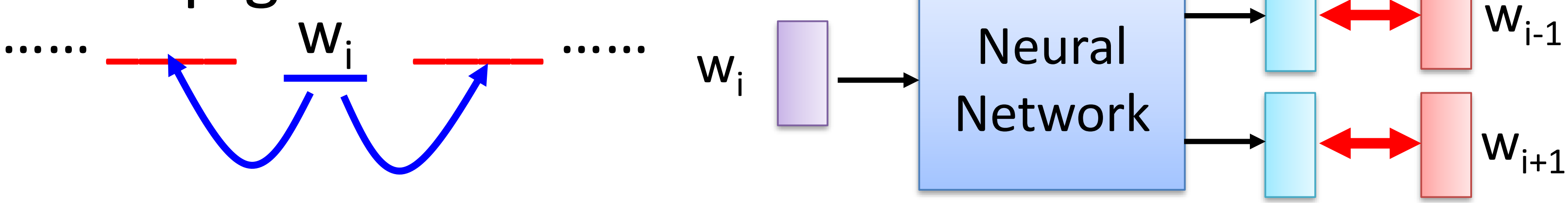
# Word Vector Representations – Various Architectures

- Continuous bag of word (CBOW) model




***predicting the word given its context***

- Skip-gram



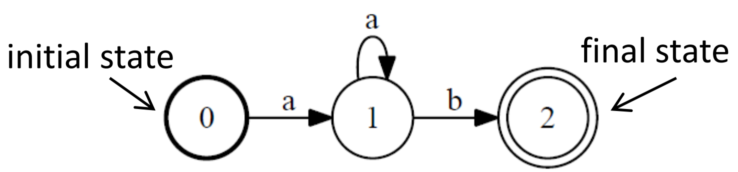

***predicting the context given a word***

# References for Word Vector Representations

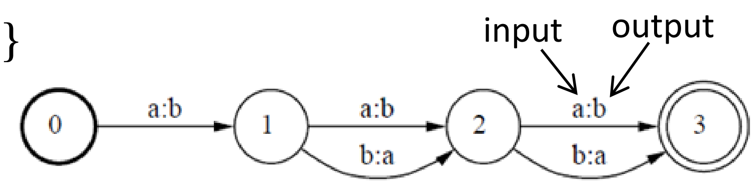

- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. ”Efficient Estimation of Word Representations in Vector Space.” In Proceedings of Workshop at ICLR, 2013.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. ”Distributed Representations of Words and Phrases and their Compositionality.” In Proceedings of NIPS, 2013.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. ”Linguistic Regularities in Continuous Space Word Representations.” In Proceedings of NAACL HLT, 2013.

# Weighted Finite State Transducer(WFST)

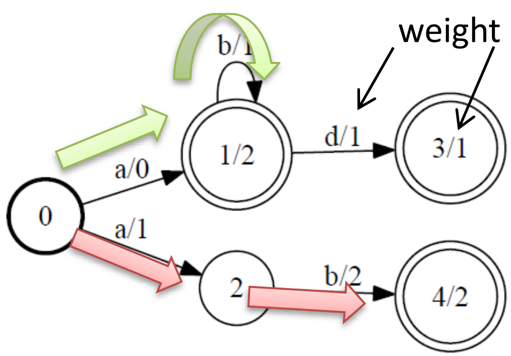
- **Finite State Machine**
  - A mathematical model with theories and algorithms used to design computer programs and digital logic circuits, which is also called “Finite Automaton”.
  - The common automata are used as acceptors, which can recognize its legal input strings.
- **Acceptor**
  - Accept any legal string, or reject it
  - EX: {ab, aab, aaab, . . .} = aa*b




- **Transducer**
  - A finite state transducer (FST) is an extension to an acceptor
  - Transduce any legal input string to another output string, or reject it
  - EX: {aaa, aab, aba, abb} -> {bbb, bba, bab, baa}



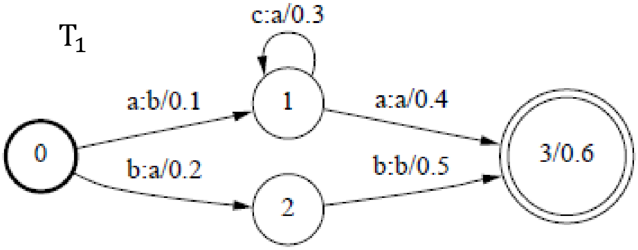

- **Weighted Finite State Machine**
  - FSM with weighted transition
  - Two paths for “ab”
    - Through states (0, 1, 1); cost is (0+1+2) = 3
    - Through states (0, 2, 4); cost is (1+2+2) = 5



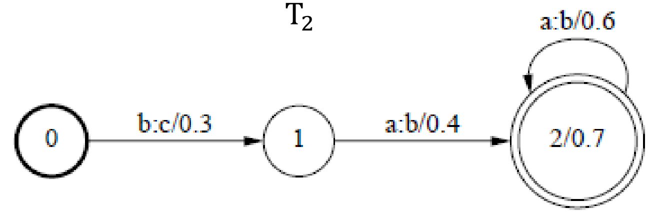
# WFST Operations (1/2)

- **Composition**
  - Combining different levels of representation
  - T is the composition of $T_1$ and $T_2$ $\Rightarrow T \equiv T_1 \circ T_2$
  - The fact that T mapping u to w, implying $T_1$ mapping u to v, and $T_2$ mapping v to w.

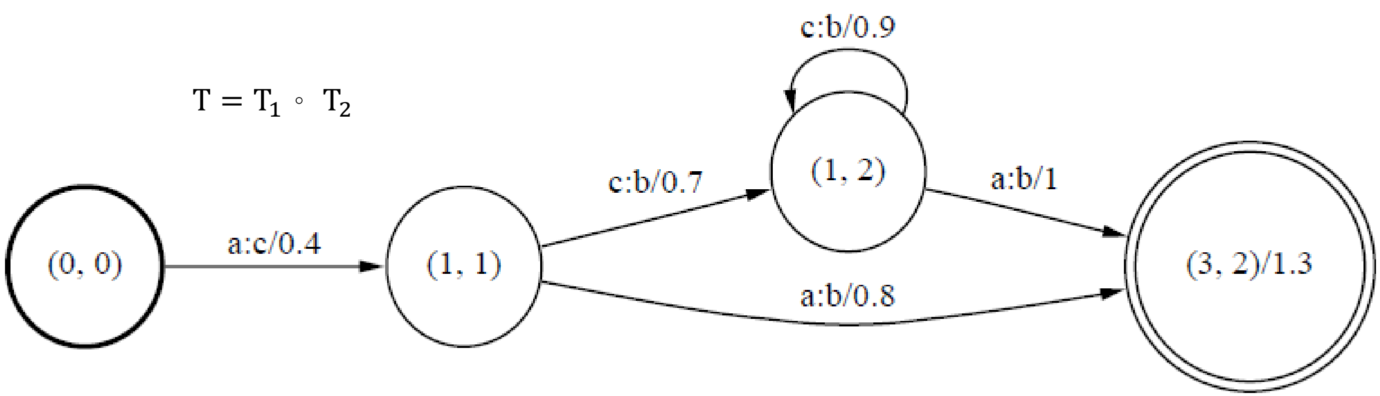
$T_1$




$T_2$



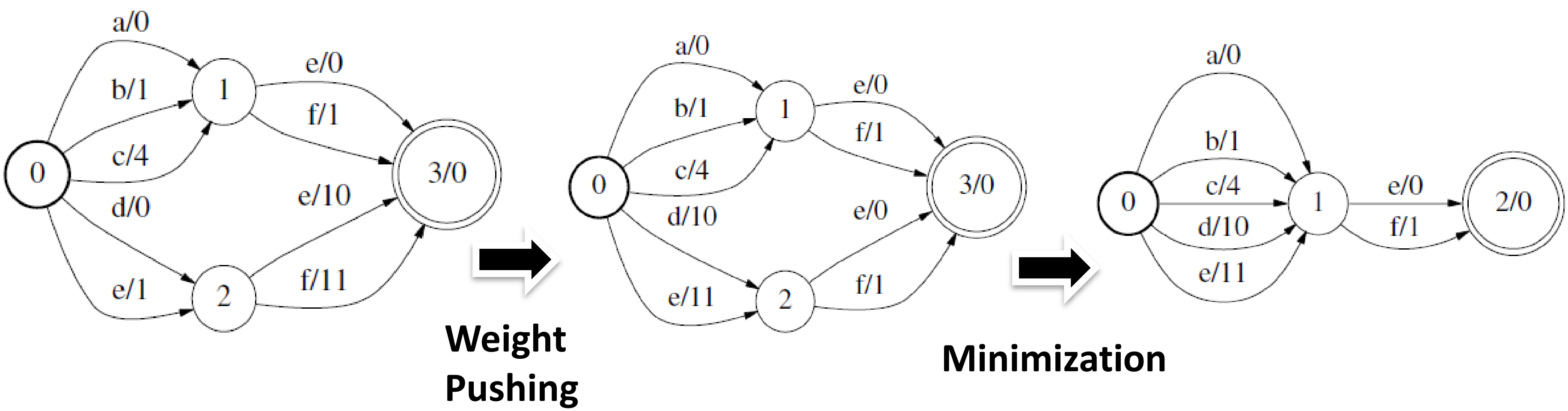

$T = T_1 \circ T_2$




$\{aa\} \rightarrow \{ba\} : 1.1$
$\{ba\} \rightarrow \{cb\} : 1.4$
$\Rightarrow$ $\{aa\} \rightarrow \{cb\} : 2.5$

# WFST Operations (2/2)

- **Minimization**
  - The equivalent automaton with least number of states and  least transitions
- **Weight pushing**
  - Re-distributing weight among transitions while kept equivalent to improve search(future developments known earlier, *etc*.), especially pruned search



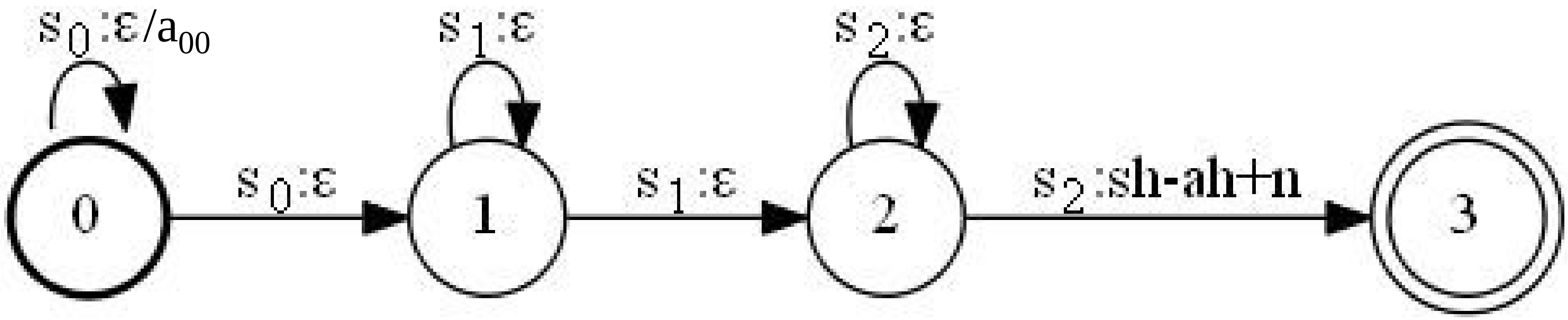
# WFST for ASR (1/6)

- **HCLG ≡ H ∘ C ∘ L ∘ G is the recognition graph**
  - G is the grammar or LM (an acceptor)
  - L is the lexicon
  - C adds phonetic context-dependency
  - H specifies the HMM structure of context-dependent phones

| | Input | Output |
|---|---|---|
| $H$ | HMM state sequence | triphone |
| $C$ | triphone | phoneme |
| $L$ | Phoneme sequence | word |
| $G$ | word | word |

# WFST for ASR (2/6)

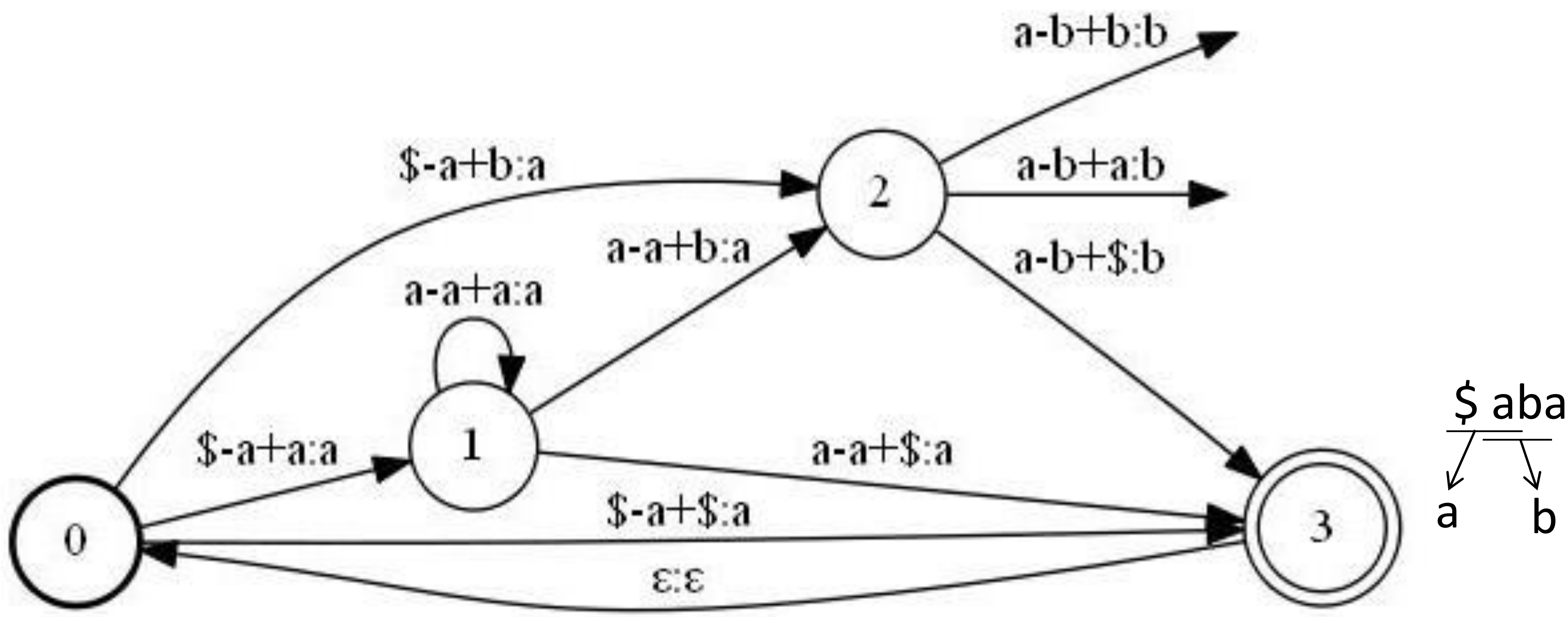
- **Transducer H: HMM topology**
  - Input: HMM state sequence
  - Output: context-dependent phoneme (e.g., triphone)
  - Weight: HMM transition probability




$$\{s_0\ s_0\ s_0\ s_1\ s_1\ s_2\ s_2\ s_2\} \rightarrow \{sh - ah + n\} : a_{00}a_{00}\ a_{01} \cdots$$

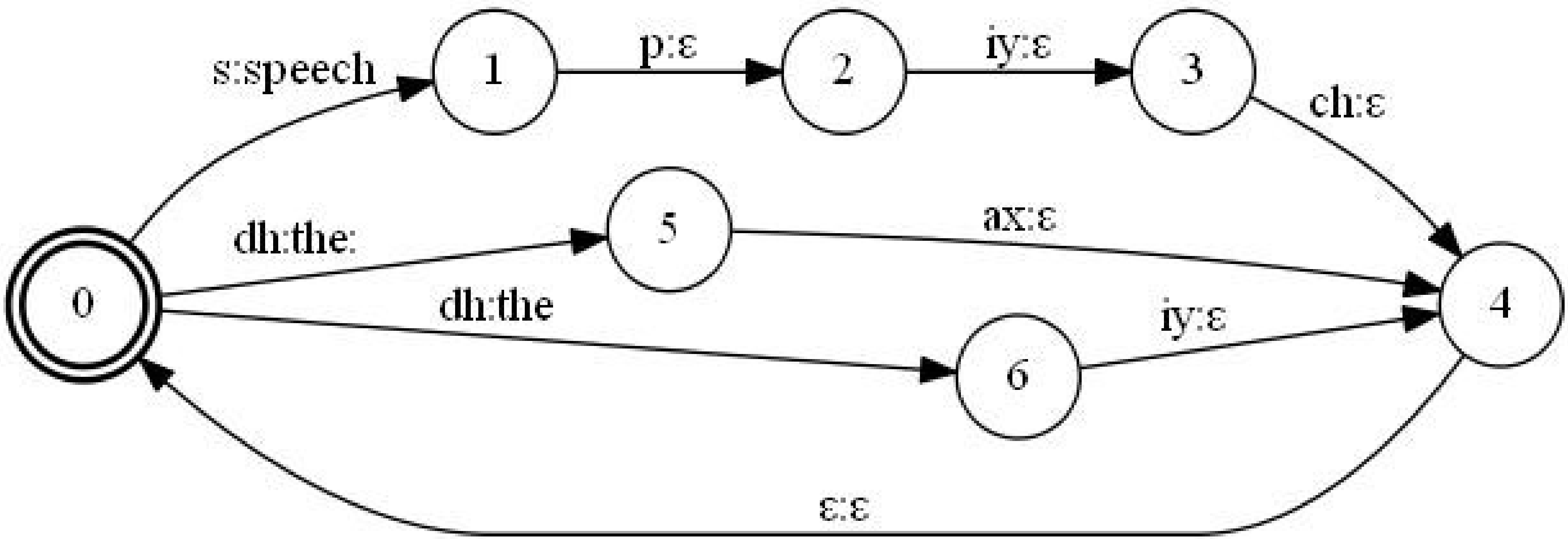
# WFST for ASR (3/6)

- **Transducer C: context-dependency**
  - Input: context-dependent phoneme (triphone)
  - Output: context-independent phoneme (phoneme)



# WFST for ASR (4/6)

- **Transducer L: lexicon**
  - Input: context-independent phoneme (phoneme) sequence
  - Output: word
  - Weight: pronunciation probability

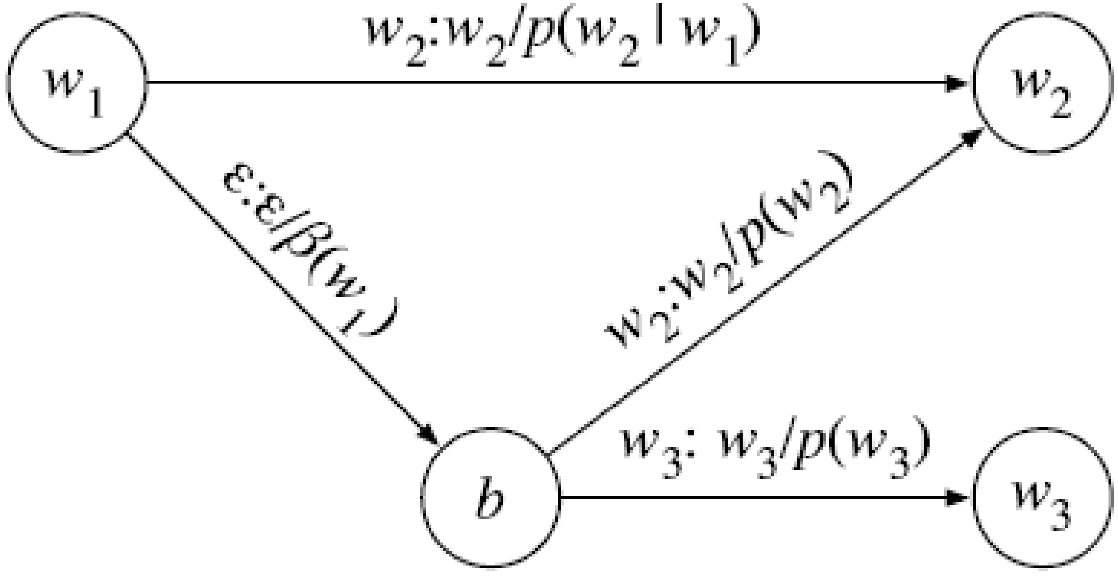



$\{s, p, iy, ch\} \rightarrow$ speech

$\{dh, ax\} \rightarrow$ the

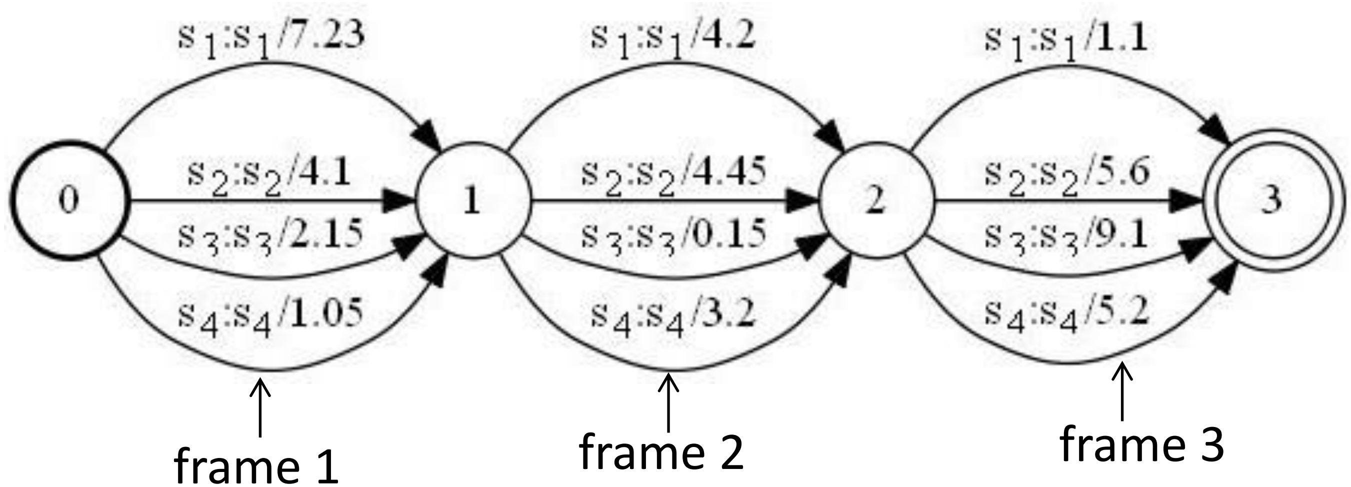
# WFST for ASR (5/6)

- **Acceptor G: N-gram models**
- **Bigram**
  - Each word has a state
  - Each bigram w1w2 has a transition w1 to w2
  - Introducing back-off state b for back-off estimation.
  - An unseen w1w3 bigram is represented as two transitions: an ε-transition from w1 to b and a transition from b to w3.



# WFST for ASR (6/6)

- **Acceptor U: utterance**
  - Transition between the state labeled t-1 and the state labeled t giving the posterior probabilities for all HMM states given frame t



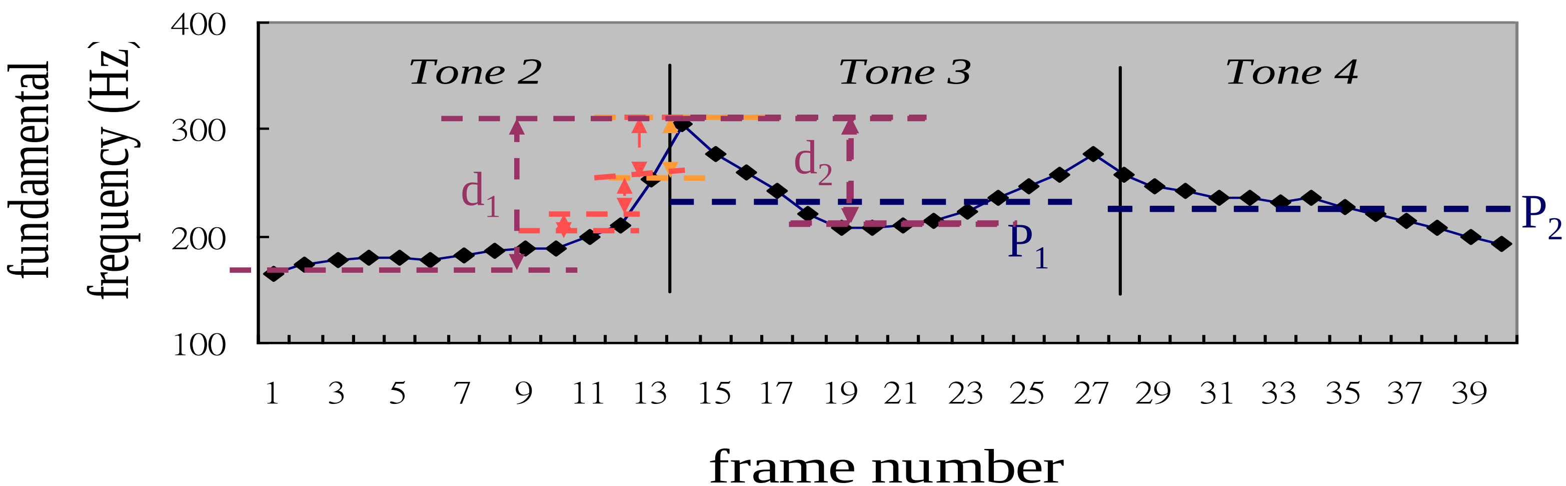

- **Decoding**
  - $w' = argmax_w \, U \circ (H \circ C \circ L \circ G)$
  - $(H \circ C \circ L \circ G)$ replacing the conventional tree structure expanded by lexicon trees, built off-line
  - $U \circ (H \circ C \circ L \circ G)$ constructing a graph given U, over which all constraints or criteria for search can be applied

# References

- **WFST**
  – Mehryar Mohri, “Finite-state transducers in language and speech processing,”Comput. Linguist., vol. 23, no. 2, pp. 269–311, 1997.
- **WFST for LVCSR**
  – Mehryar Mohri, Fernando Pereira, and Michael Riley, “Weighted automata in text and speech processing,” in European Conference on Artificial Intelligence. 1996, pp. 46–50, John Wiley and Sons.
  – Mehryar Mohri, Fernando C. Pereira, and Michael Riley, “Speech Recognition with Weighted Finite-State Transducers,” in Springer Handbook of Speech Processing, Jacob Benesty, Mohan M. Sondhi, and Yiteng A. Huang, Eds., pp. 559–584. Springer Berlin Heidelberg, Secaucus, NJ, USA, 2008.

# Prosodic Features (I)

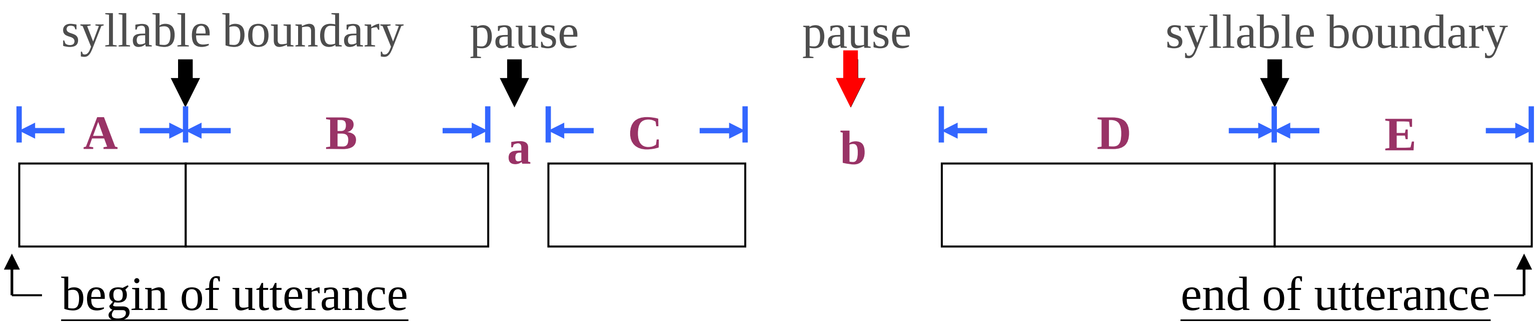
- **Pitch-related Features (examples in Mandarin Chinese)**
  - The average pitch value within the syllable
  - The maximum difference of pitch value within the syllable
  - The average of absolute values of pitch variations within the syllable
  - The magnitude of pitch reset for boundaries
  - The difference of such feature values of adjacent syllable boundaries ( $P_1$-$P_2$ , $d_1$-$d_2$ , etc.)




  - at least 50 pitch-related features

# Prosodic Features (II)

- **Duration-related Features (examples in Mandarin Chinese)**



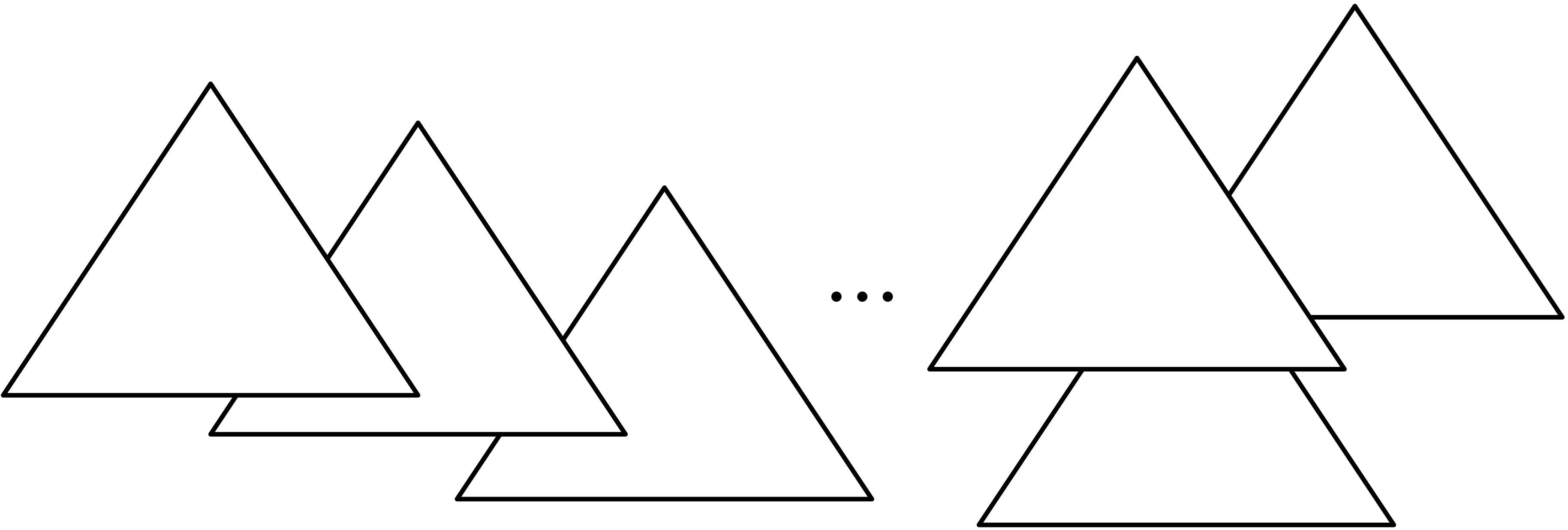

- Pause duration b
- Average syllable duration
  (B+C+D+E)/4 *or* ( (D+E)/2 + C )/2
- Average syllable duration ratio
  (D+E)/(B+C) *or* (D+E)/2 /C
- Combination of pause & syllable features (ratio or product)
  C*b , D*b, C/b, D/b
- Lengthening C / ( (A+B)/2 )
- Standard deviation of feature values

  – at least 40 duration-related features

- **Energy-related Features**
  – similarly obtained

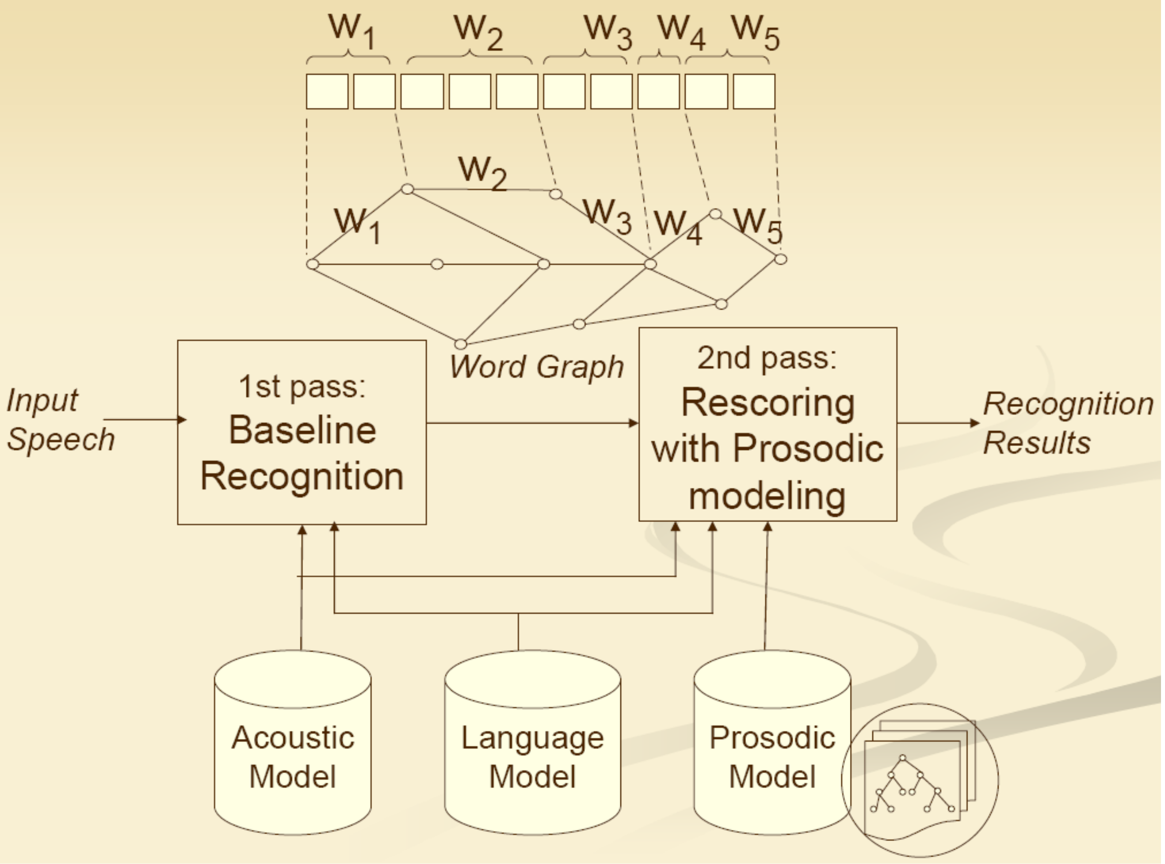
# Random Forest for Tone Recognition for Mandarin

- **Random Forest**
  - a large number of decision trees
  - each trained with a randomly selected subset of training data and/or a randomly selected subset of features
  - decision for test data by voting of all trees



# Recognition Framework with Prosodic Modeling

- **An example approach: Two-pass Recognition**




- **Rescoring Formula:**

$$S(W) = \log P\left(X \middle| W\right) + \lambda_l \log P\left(W\right) + \lambda_p \log P\left(F \middle| W\right)$$

Prosodic model

$\lambda_l, \lambda_p$: weighting coefficients

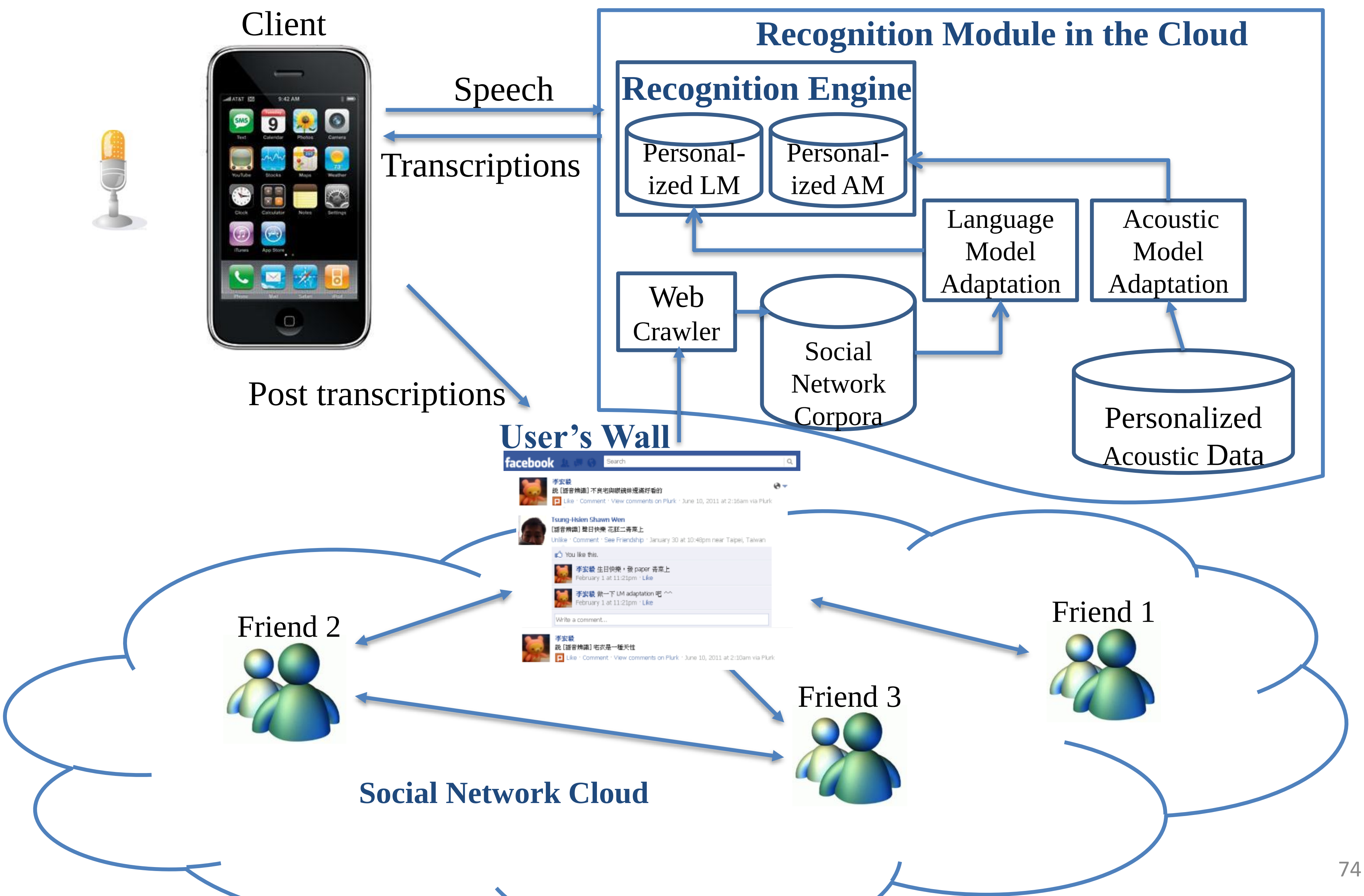
# References

- **Prosody**
  - “Improved Large Vocabulary Mandarin Speech Recognition by Selectively Using Tone Information with a Two-stage Prosodic Model”, Interspeech, Brisbane, Australia, Sep 2008, pp. 1137-1140
  - “Latent Prosodic Modeling (LPM) for Speech with Applications in Recognizing Spontaneous Mandarin Speech with Disfluencies”, International Conference on Spoken Language Processing, Pittsburgh, U.S.A., Sep 2006.
  - “Improved Features and Models for Detecting Edit Disfluencies in Transcribing Spontaneous Mandarin Speech”, IEEE Transactions on Audio, Speech and Language Processing, Vol. 17, No. 7, Sep 2009, pp. 1263-1278.
- **Random Forest**
  - http://stat-www.berkeley.edu/users/breiman/RandomForests/cc_home.htm
  - http://stat-www.berkeley.edu/users/breiman/RandomForests/cc_papers.htm

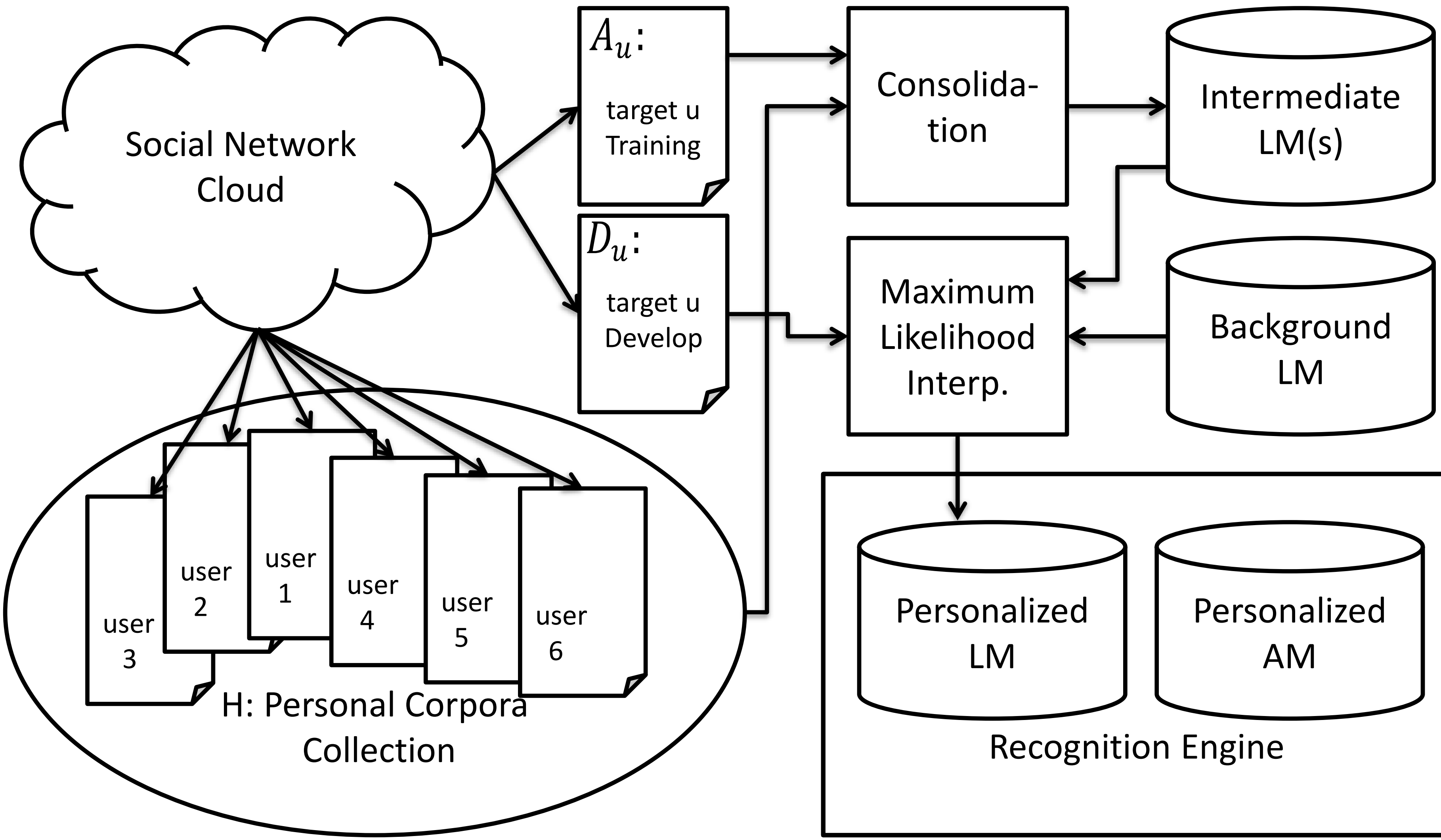
# Personalized Recognizer and Social Networks

- **Personalized recognizer is feasible today**
  - Smart phone user is personal
    - each smart phone used by a single user
    - user identification is known once the smart phone is turned on
  - Personal corpus is available
    - Audio data easily collected at server
    - Text data available on social networks

# Personalized Recognizer and Social Networks



# Language Model Adaptation Framework



# References for Personalized Recognizer

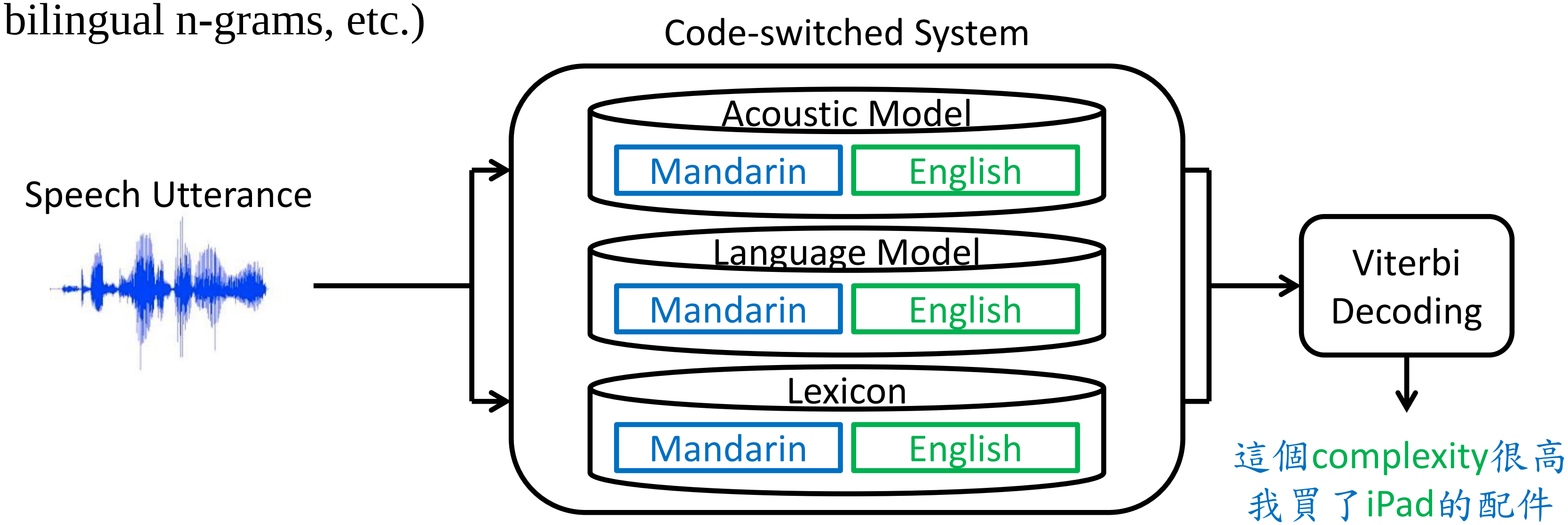

- "Recurrent Neural Network Based Language Model Personalization by Social Network Crowdsourcing", Interspeech 2013.
- "Personalizing A Universal Recurrent Neural Network Language Model with User Characteristic Features by Social Network Crowdsourcing", ASRU, 2015.
- "Personalized Speech Recognizer with Keyword-based Personalized Lexicon and Language Model using Word Vector Representations", Interspeech, 2015.

# Recognizing Code-switched Speech

- **Definition**
  - Code-switching occurs from word to word in an utterance
  - Example : 當我們要作 Fourier Transform 的時候

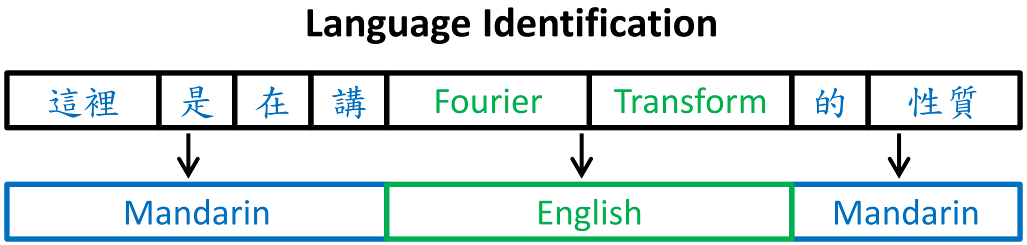
“Host” language “Guest” language

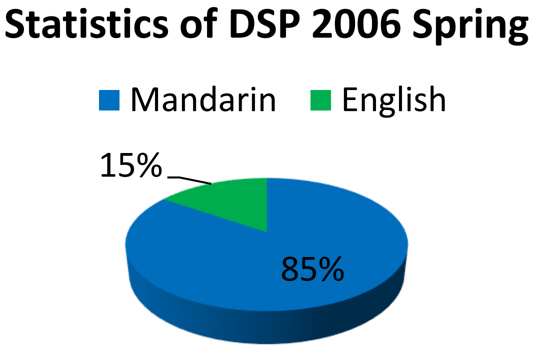
- **Speech Recognition**
  - Bilingual acoustic models, language model, and lexicon
  - A signal frame may belong to a Mandarin phoneme or an English phoneme, a Mandarin phoneme may be preceded or followed by an English phoneme and vice versa, a Chinese word may be preceded or followed by an English word and vice versa (bilingual triphones, bilingual n-grams, etc.)



# Recognizing Code-switched Speech

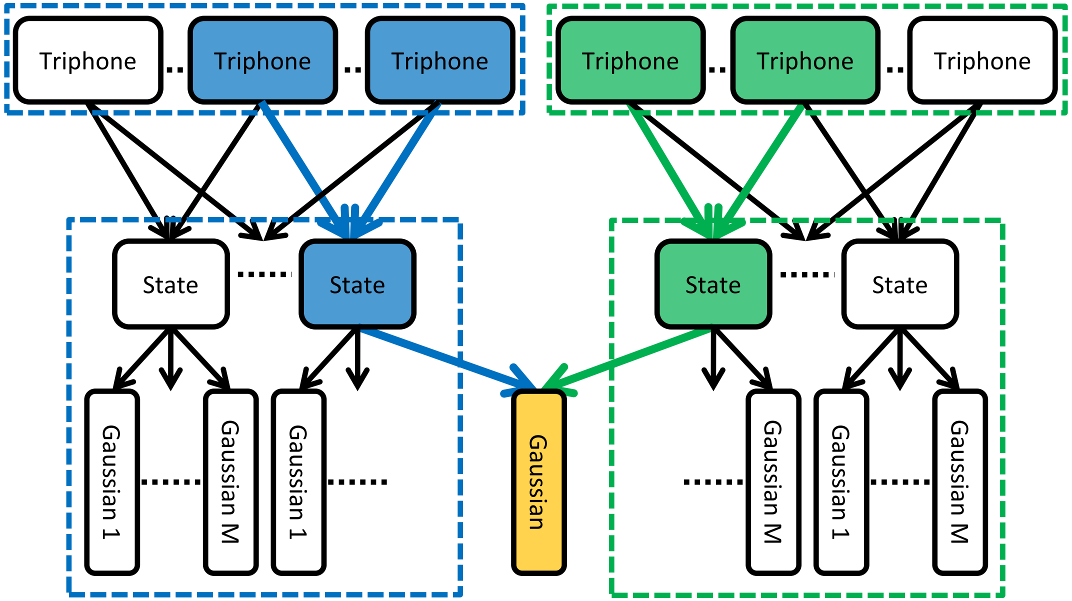
- **Code-switching issues**
  - Imbalanced data distribution
    - There are much more data for host language but only very limited for guest language
    - The models for guest language are usually weak, therefore accuracy is low
  - Inter-lingual ambiguity
    - Some phonemes for different languages are very similar but different (*e.g.* ㄅ vs. B ), but may be produced very closely by the same speaker
  - Language identification (LID)
    - Units for LID are smaller than an utterance
    - Very limited information is available

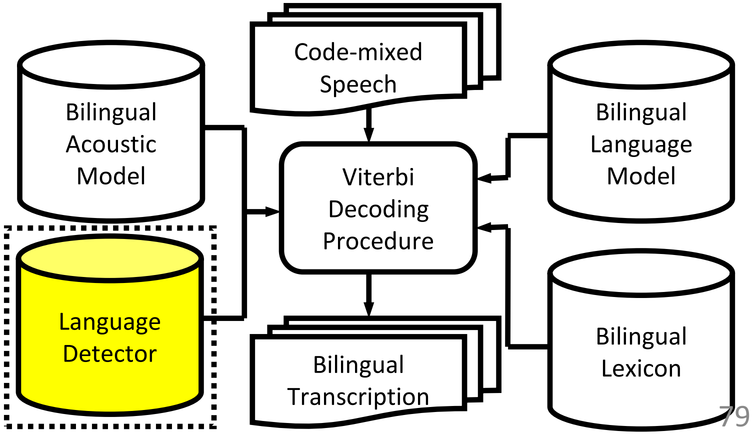


# Recognizing Code-switched Speech

- **Some approaches to handle the above problems**
  - Acoustic unit merging and recovery
    - Some acoustic units shared across languages: Gaussian, state, model
    - Shared training data
    - Models recovered with respective data to preserve the language identity
  - Frame-level language identification (LID)
    - LID for each frame
    - Integrated in recognition

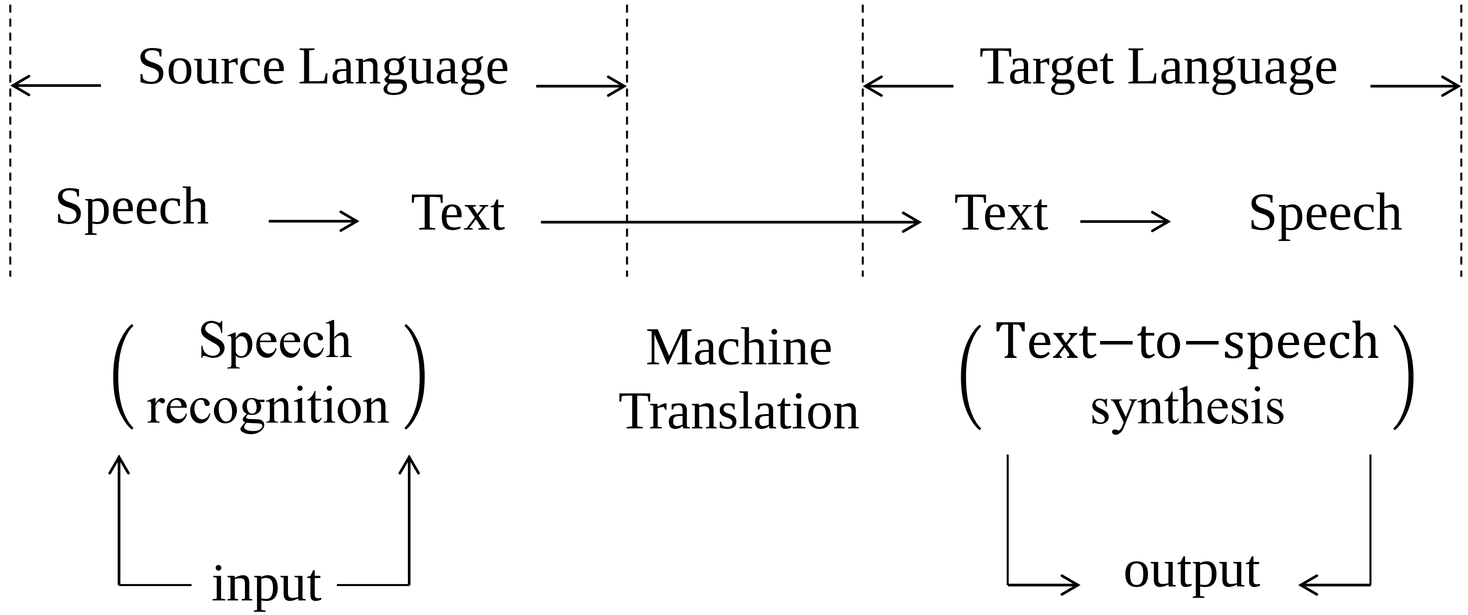


# References for Recognizing Code-switched Speech

1. **"An Improved Framework for Recognizing Highly Imbalanced Bilingual Code-Switched Lectures with Cross-Language Acoustic Modeling and Frame-Level Language Identification",** *IEEE/ACM Transactions on Audio, Speech and Language Processing*, *Vol. 23, No. 7, 2015.*
2. **"Recognition Of Highly Imbalanced Code-mixed Bilin-gual Speech With Frame-level Language Detection Based On Blurred Posteriorgram,"** *ICASSP, 2012.*
3. **"Language Independent And Language Adaptive Acoustic Modeling For Speech Recognition,"** Tanja Schultz and Alex Waibel, *Speech Communication, 2001.*
4. **"Learning Methods In Multilingual Speech Recognition,"** Hui Lin, Li Deng, Jasha Droppo, Dong Yu, and Alex Acero, *NIPS, 2008.*

# Speech-to-speech Translation




- **Language difference is a major problem in the globalized world**
- **For N languages considered, ~ $N^2$ pairs of languages for translation**
- **Human revision after machine translation feasible**

# Machine Translation — Simplified Formulation

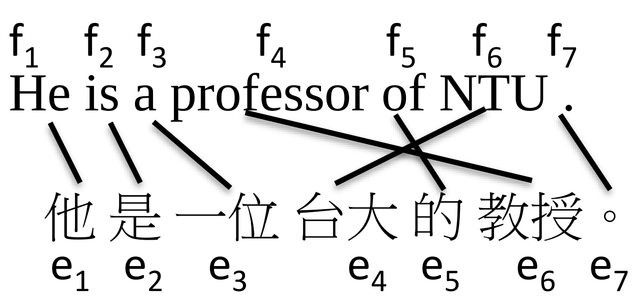
- **Source language (Foreign) f:**
  - word set (dictionary): F
  - a sentence: f = $f_1 f_2 \ldots f_j \ldots f_J$, $f_j \in F$, J: number of words
- **Target language (English) e:**
  - word set (dictionary): E
  - a sentence: e = $e_1 e_2 \ldots e_i \ldots e_I$, $e_i \in E$, I: number of words
- **Statistical Machine Translation (SMT) task:**
  - model $p(e|f)$
  - given a new source language sentence $f', e' = argmax_e\, p(e|f')$
  - $e' = argmax_{Y(f')}\, p(e|f')$

    $Y(f')$: a smaller set of $e$ considered
  - $p(e|f) = p(f|e)p(e)/p(f) \propto p(f|e)p(e)$ (Bayesian theorem)
  - $p(e)$: language model
  - $p(f|e)$: translation model

# Generative Models for SMT

- **Language model (p(e)):**
  - –conventional n-gram model
  - –recurrent neural network
  - –domain adaptation can be applied (corpus collection needed)

- **Translation model (p(f|e)):**
  - –$p(f|e) = \sum_a p(f|e,a)p(a)$, $a$ : alignment
  - –$p(f|e,a)$: unit (word/phrase) translation model
  - –$p(a)$: reordering model
  - –Example for an alignment:




For this example alignment a

$p(f|e,a)$= p(He|他)*p(is|是)…

$p(a)$= p(a: He<-->他, is<-->是,….)

All probabilities trained with parallel bilingual corpora aligned or not

# Generative Models for SMT

- **Unit translation model p(f|e,a):**
  - Based on unit translation table:
  - Examples:

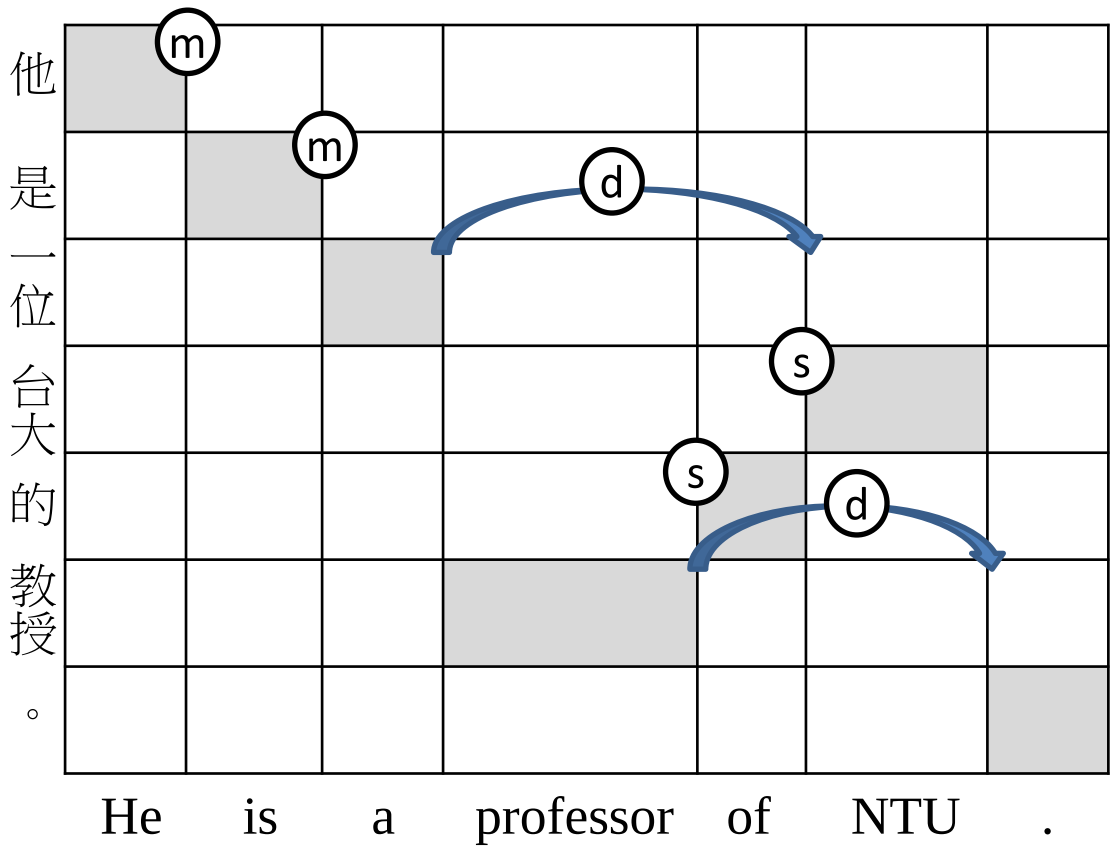
| | |
|---|---|
| p(book\|書) | 0.95 |
| p(write\|書) | 0.05 |

| | |
|---|---|
| p(walk\|走) | 0.8 |
| p(leave\|走) | 0.2 |

  - Tables can be accumulated from training data

# An Example of Reordering Model

- **Lexicalized reordering model:**
  - model the orientation
  - orientation types: monotone(m), swap(s), discontinuous(d)
  - Ex. p(他<-->He,是<-->is…)=p( {他,He,(m)}, {是,is,(m)}, {一位,a,(d)}, {台大,NTU,(s)}, {的,of,(s)}, {教授,professor,(d)} )

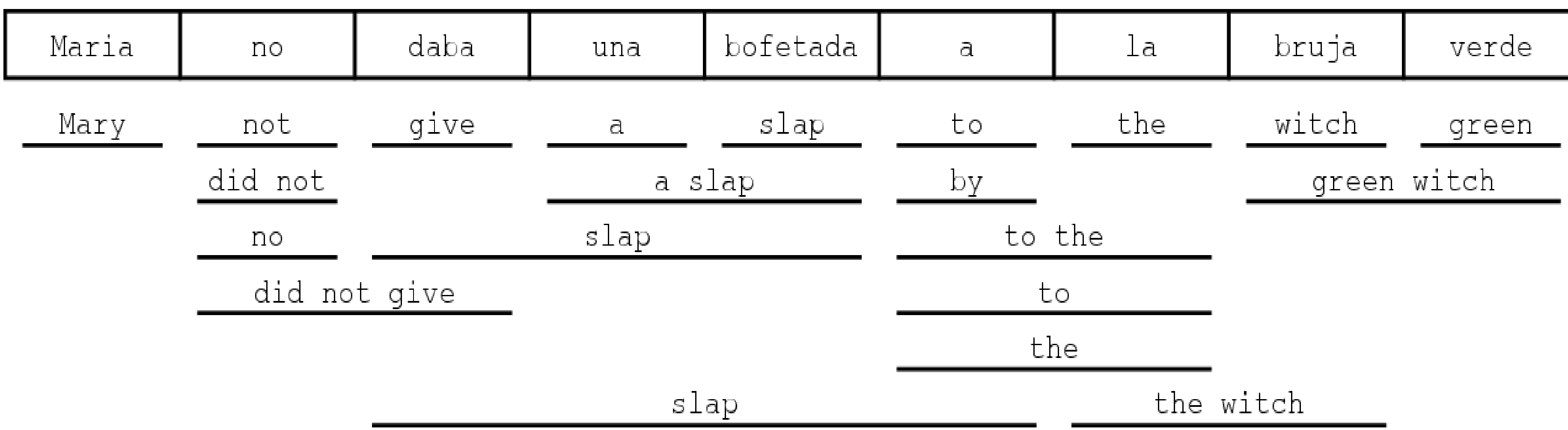



Probabilities trained with parallel bilingual corpora

# Modeling the Phrases

# Decoding Considering Phrases

- **Phrase-based Translation**
  - first source word covered
  - last source word covered
  - phrase translation considered
  - phrase translation probabilities trained



# References for Translation

- **A Survey of Statistical Machine Translation**
  - Adam Lopez
  - Tech. report of Univ. of Maryland
- **Statistical Machine Translation**
  - Philipp Koehn
  - Cambridge University Press
- **Building a Phrase-based Machine Translation System**
  - Kevin Duh and Graham Neubig
  - Lecture note of "Statistical Machine Translation," NAIST, 2012 spring
- **Speech Recognition, Machine Translation, and Speech Translation – A Unified Discriminative Learning Paradigm**
  - IEEE Signal Processing Magazine, Sept 2011
- **Moses: Open Source Toolkit for Statistical Machine Translation**
  - Annual Meeting of the Association for Computational Linguistics (ACL) demonstration session, Prague, Czech Republic, June 2007

# References for Speech Recognition Updates


- **"Structured Discriminative Models for Speech Recognition", IEEE Signal Processing Magazine, Nov 2012**
- **"Subword Modeling for Automatic Speech Recognition", IEEE Signal Processing Magazine, Nov 2012**
- **"Machine Learning Paradigms for Speech Recognition – An Overview", IEEE Transactions on Audio, Speech and Language Processing, May 2013**