HMM 기반의 음성인식
Speech Recognition with Hidden Markov Models

서울대학교 전기정보공학부
교수 성원용

음성인식 응용

- 대용량 음성인식 (LVCSR)
  - 응용: voice search, texting 등
  - 보통 64K-word 이상 지원, 좋은 언어모델 필요
  - 대용량 메모리를 사용함, 전력소모 많음, 개인당 사용시간 짧음
- 소용량 인식
  - 응용: Key-word spotting (wake-up), speaker identification, voice activity detection (VAD)
  - 적용 단어 수 적음 (모델 사이즈 작음)
  - 대기시 항상 동작할 수 있음. 매우 낮은 전력의 높은 정밀도의 동작 필요.

음성인식 방법의 역사

- 1980's: DTW (dynamic time warping) 기반. 대부분 특정화자 인식, 소용량
- 1985: HMM(Hidden Markov Model) 기반 (음성인식을 확률적 순차처리 모델로 만듦)
- **1990's ~ 2010 GMM(Gaussian Mixture Model)+ HMM + n-gram 언어모델 + several different speech features, speaker adaptation**
- **2012 전후: DNN + HMM + n-gram 언어모델**
- 2013: speech recognition with CTC-RNN (end to end training), neural network LM
- 현재: fully neural 연구 (neural acoustic model, neural LM - no HMM, no n-gram LM)

음성인식의 역사

- 통계적 방법
  - Gaussian mixture model을 이용한 tri-state phone modeling
  - Hidden Markov model을 이용한 단어와 언어모델
- 인공신경망 방법
  - DNN (deep neural network)을 이용한 phone modeling (tri-phone, mono-phone)
  - CTC (Connectionist Temporal Classification) 기반의 end-to-end speech recognition

음성의 모델링

- 음성은 매우 짧은 시간 (약 20msec)에서는 스펙트럼이 변하지 않는다. (phoneme state)
- Phoneme – 우리가 언어학적으로 정의하는 어떤 발음 (발음기호), 그런데 이 발음은 아주 정해진 것이 아니라 옆의 발음에 따라 변한다.
- 단어 – 앞의 phoneme 의 sequence
- 문장 – 단어의 sequence

Acoustic modeling

- Phoneme: 언어학자가 정의한 기본 발음단위.
- 영어의 경우 40~50개의 phoneme

(There are total of 78 phonemes used in TIMIT database, out of which 46 phonemes are of English language (American), 1 phoneme for silence(sil), 1phoneme for short pause(sp) and the remaining 30 are stressed phonemes.)

- 하나의 phoneme 도 그 음을 처음 발음하는 frame, 중간 frame, 끝날 때 frame 에서의 발음이 다른데, 이 경우 이를 통째로 모델하는 것이 mono-phone, 세개의 state로 나누어서 model 하는 것이 tri-phone 이다.

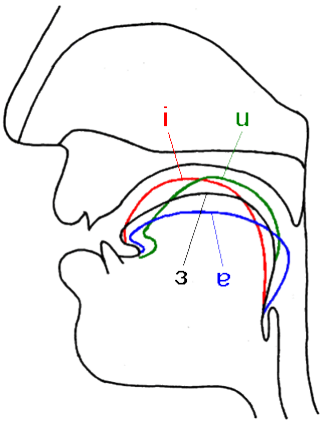
	Phone Label	Example		Phone Label	Example		Phone Label	Example
1	iy	b*ee*t	22	ch	*ch*oke	43	en	butt*on*
2	ih	b*i*t	23	b	*b*ee	44	eng	Wash*ing*ton
3	eh	b*e*t	24	d	*d*ay	45	l	*l*ay
4	ey	b*ai*t	25	g	*g*ay	46	r	*r*ay
5	ae	b*a*t	26	p	*p*ea	47	w	*w*ay
6	aa	b*o*b	27	t	*t*ea	48	y	*y*acht
7	aw	b*ou*t	28	k	*k*ey	49	hh	*h*ay
8	ay	b*i*te	29	dx	mu*dd*y	50	hv	a*h*ead
9	ah	b*u*t	30	s	*s*ea	51	el	bott*le*
10	ao	b*ough*t	31	sh	*sh*e	52	bcl	b closure
11	oy	b*oy*	32	z	*z*one	53	dcl	d closure
12	ow	b*oa*t	33	zh	a*z*ure	54	gcl	g closure
13	uh	b*oo*k	34	f	*f*in	55	pcl	p closure
14	uw	b*oo*t	35	th	*th*in	56	tcl	t closure
15	ux	t*oo*t	36	v	*v*an	57	kcl	k closure
16	er	b*ir*d	37	dh	*th*en	58	q	glotal stop
17	ax	*a*bout	38	m	*m*om	59	pau	pause
18	ix	deb*i*t	39	n	*n*oon	60	epi	epenthetic silence
19	axr	butt*er*	40	ng	si*ng*			
20	ax-h	s*u*spect	41	em	bott*om*	61	h#	begin/end marker
21	jh	*j*oke	42	nx	wi*nn*er			

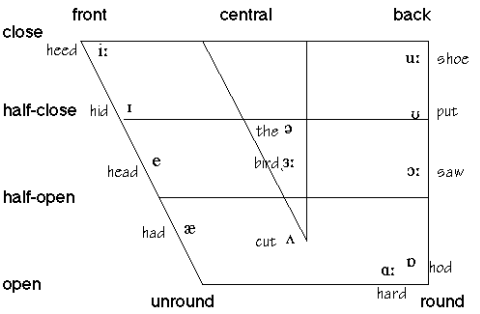
Table 2. 61 TIMIT original phone set.

aa, ao	aa
ah, ax, ax-h	ah
er, axr	er
hh, hv	hh
ih, ix	ih
l, el	l
m, em	m
n, en, nx	n
ng, eng	ng
sh, zh	sh
uw, ux	uw
pcl, tcl, kcl, bcl, dcl, gcl, h#, pau, epi	sil
q	-

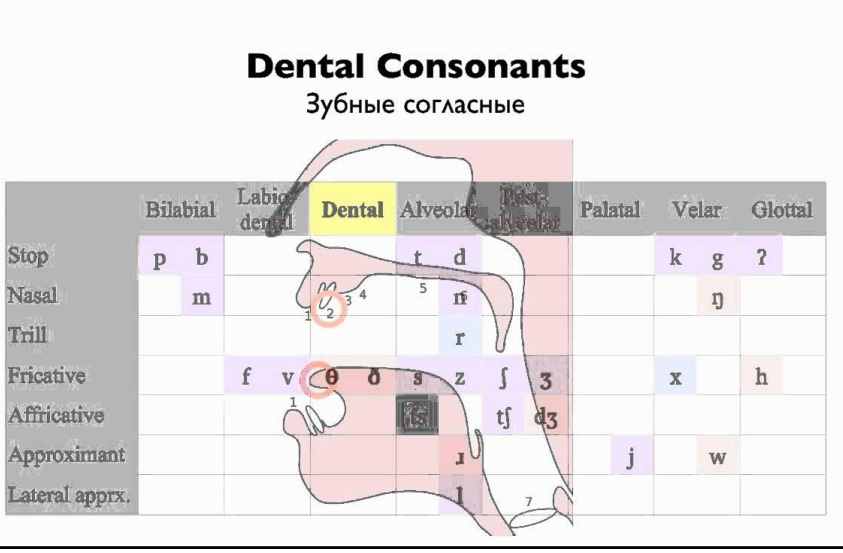
Table 3. Mapping from 61 classes to 39 classes, as proposed by Lee and Hon, (Lee & Hon, 1989). The phones in the left column are folded into the labels of the right column. The remaining phones are left intact. The phone 'q' is discarded.

모음


i
u
ɜ
ə


front
central
back
close
heed iː
uː shoe
half-close hid ɪ
the ɘ
ʊ put
head e
bird ɜː
ɔː saw
half-open
had æ
cut ʌ
ɑː ɒ hod
open
unround
hard
round

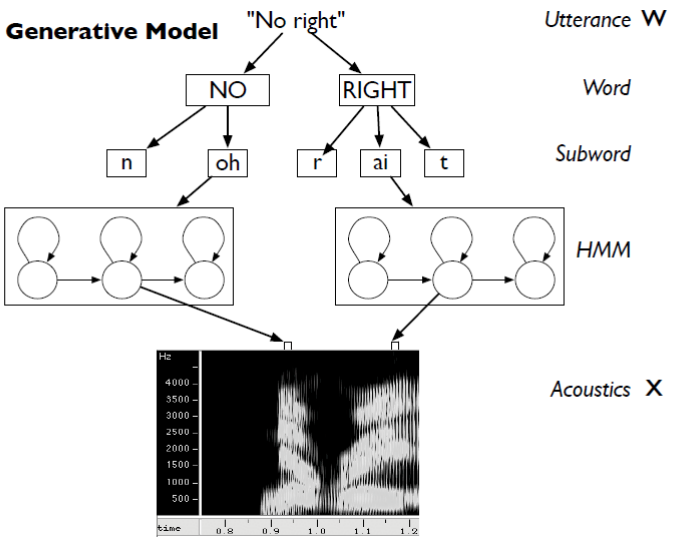
자음


Dental Consonants
Зубные согласные
Bilabial
Labio-dental
Dental
Alveolar
Post-alveolar
Palatal
Velar
Glottal
Stop
Nasal
Trill
Fricative
Affricative
Approximant
Lateral apprx.

Tri-phone

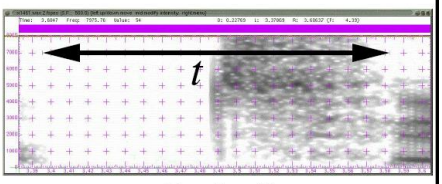
- 어떤 phoneme 을 발음하는 과정을 앞부분, 중간 부분, 뒷부분 이렇게 세부분 (tri)의 state 로 나누어서 모델링
- 이 경우 각 부분은 비교적 일정한 주파수 특성을 보인다.
- 앞부분과 뒷부분은 이어지는 발음에 따라 달라진다 (context dependent). 따라서 triphone 의 개수는 엄청 많아질 수 있는데 (수만) 이를 줄여서 수천개로 만들어 쓴다. 이를 CD-triphone states 라 한다. 이 CD-triphone states 가 보통의 hidden Markov model에 사용된다.

Hierarchical modeling of speech

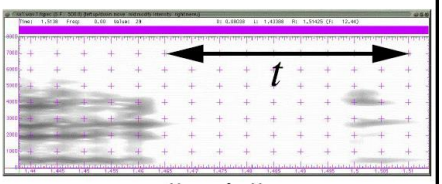


Phonetic context

- Context – 좌우의 발음에 의해서 가운데 발음이 영향을 받음 (coarticulation).
- /n/ in ten (dental) and tenth (alveolar)

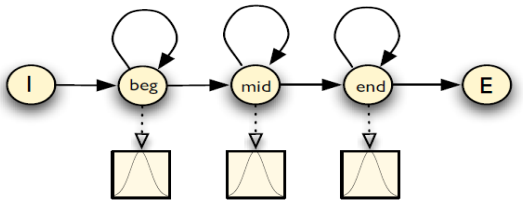


"tube"

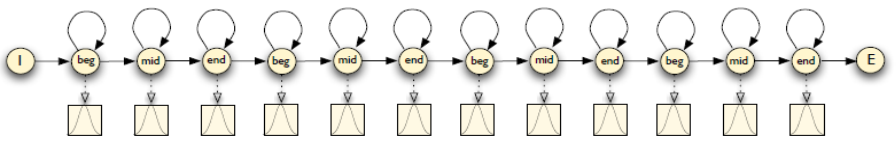


"suit"

Three state phone models (triphone)



/ih/



/s/ /ih/ /k/ /s/

"six"

Acoustic modeling

- 음성을 듣고서 발음으로 표시함. 이 때 표현하는 발음의 단위에 따라
  - Triphone states (매우 작은 시간 단위로 표현)
    - 매우 안정된 주파수 특징을 보이기 때문에 인식이 쉽다.
    - 이를 꾀어 맞추어서 단어와 문장으로 만들기가 복잡하다. 더 복잡한 hidden Markov model.
    - 전체적으로 정확도 좋다.
  - Monophone (40~70개의 phoneme 으로 표현)
  - Grapheme (알파벳 글자로 표현) – 수십개, 1초에 몇번 나옴.
  - Wordpiece – 수백 ~ 수만
  - Word – 수십만 단어, 1초 정도의 시간

10~20msec 정보

100msec~1초 정도의 정보

HMM 기반 음성인식 알고리즘의 기초

대용량 음성인식, 왜 어려운가?

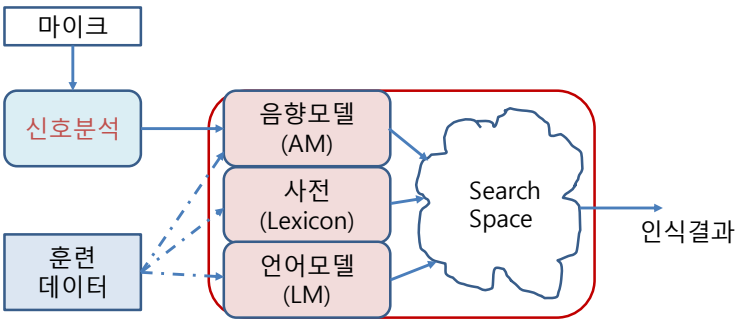
- Acoustic information만으로 음성인식 잘 안된다.
  - 불완전한 발음, phonetically (p <-> f), skipping, insertion
  - 각 phoneme의 duration이 일정치 않다.
  - 주변 환경, 마이크 등
- 인간은 어떻게 가능? Knowledge기반
  - 제한된 개수의 phoneme 사용
  - 제한된 개수의 단어 사용
  - 단어와 단어 사이의 transition 확률 (language model)
- 인간은 대화 context 상에서 나올 단어를 예상하고, 불완전한 발음에서도 어떤 단어인지를 예측함.
  - Statistical modeling in short-term and long-term

음성인식 에러율 (task dependent)

Ballpark numbers; exact numbers depend very much on the specific corpus

Task	Vocabulary	Word Error Rate %
Digits	11	0.5
WSJ read speech	5K	3
WSJ read speech	20K	3
Broadcast news	64,000+	5
Conversational Telephone	64,000+	10

음성인식 - knowledge integration



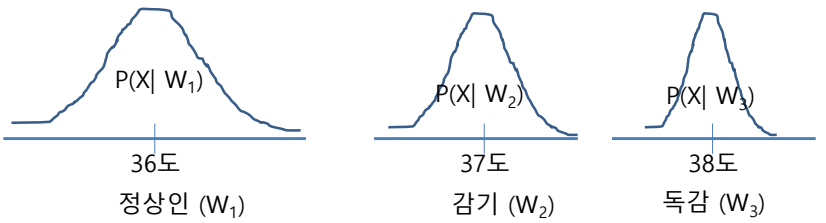
신호분석 – 음성을 주파수 성분으로 나눔 (time to frequency domain)
음향모델 – 음성을 음소 (phoneme) 단위로 분석
사전 – 어떤 단어의 발음 (음소의 연결)
언어모델 – 다음에 나올 단어를 예측 (perplexity를 줄임)
탐색 – 들어오는 발음을 세가지 모델에 맞추어서 가장 확률이 높은 연결을 찾음

통계적 분류 - Bayes' theorem

Bayes' theorem (Thomas Bayes, 영국 1701~1761)

$$P(W \mid X) = \{P(X \mid W)\, P(W)\}/P(X)$$
$$\sim= \text{Likelihood}(X|W)\, P(W)$$

X = 37.5도 이다. 이 사람은 정상, 감기, 독감?
The prior probability: 겨울이다. 독감이 돈다. 예방주사여부



Maximum likelihood (최대우도) and Maximum a Posteriori (최대 사후확율)

- Maximum likelihood classification: 소리를 듣고, 그 발음이 나왔을 가장 그럴 듯한 발음으로 판단하는 것 (generative model)
  - $P(X|C_j)$
  - (참고 X는 input feature, Cj 는 j 번째 class)
- Max a posteriori (MAP) 상황을 고려해서 판단하는 것
  - $P(X|C_j)\ P(C_j)$
- Speech의 경우 $P(C_j)$ 가 sequence의 형태로 나타내진다. (Sam?..) -> Hidden Markov Model (Lexicon: phoneme sequence model, Language model: word sequence model)

Statistical speech recognition (통계적 음성인식)

- 세가지 정보 (Acoustic, Lexicon, LM)를 확률적으로 곱합
- X: sequence of acoustic feature vector (observation), W: word sequence, the recognized word sequence W* is

$$W^* = \text{argmax}_{\text{for all W}}\ P(W \mid X)$$
$$= \text{argmax}_{\text{for all W}}\ P(X \mid W)\ P(W)$$

Bayes' theorem

$$P(W \mid X) = \{P(X \mid W)\ P(W)\}/P(X)$$

Acoustic and Lexicon LM

음성인식기 모델의 세가지 요소

(i) Feature extractor: 음성을 frame(10msec 정도 구간)으로 나누어서 주파수 도메인의 신호 (X)로 바꾼다

(ii) $P(X|C_i)$: 어떤 음소(phoneme)에서 어떤 feature 가 나올 확률을 구한다. GMM (Gaussian Mixture Model)

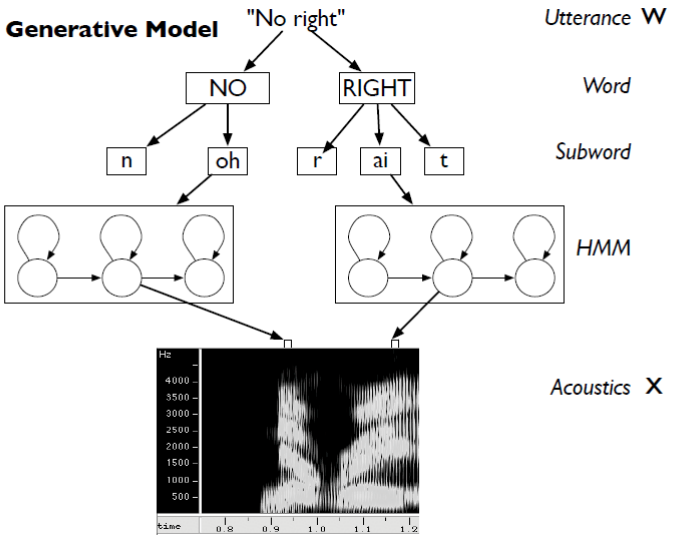
(iii) $P(C_i)$: 어떤 음소상태에 있을 a priori 정보를 구한다. (Hidden Markov Model)

GMM-HMM model: 2010년정도까지 주류

DNN-HMM model: 2010~

Fully neural: 현재 연구 중

Hierarchical modeling of speech



Feature vector after MFCC

- 39-dimensional Features per 10 ms frame:
  - 12 MFCC features
  - 12 Delta MFCC features
  - 12 Delta-Delta MFCC features
  - 1 (log) frame energy
  - 1 Delta (log) frame energy
  - 1 Delta-Delta (log frame energy)
- So each frame represented by a 39D vector
- Why delta? – 거의 움직이지 않는 성분을 제거 (주변환경의 impulse response 등)

Acoustic modeling with phoneme

- Phoneme: 언어학자가 정의한 기본 발음단위.
- 영어의 경우 40~50개의 phoneme

(There are total of 78 phonemes used in TIMIT database, out of which 46 phonemes are of English language (American), 1 phoneme for silence(sil), 1phoneme for short pause(sp) and the remaining 30 are stressed phonemes.)

- 하나의 phoneme 도 그 음을 처음 발음하는 frame, 중간 frame, 끝날 때 frame 에서의 발음이 다른데, 이 경우 이를 통째로 모델하는 것이 mono-phone, 세개의 state로 나누어서 model 하는 것이 tri-phone 이다.

	Phone Label	Example		Phone Label	Example		Phone Label	Example
1	iy	b*ee*t	22	ch	*ch*oke	43	en	butt*on*
2	ih	b*i*t	23	b	*b*ee	44	eng	Wash*ing*ton
3	eh	b*e*t	24	d	*d*ay	45	l	*l*ay
4	ey	b*ai*t	25	g	*g*ay	46	r	*r*ay
5	ae	b*a*t	26	p	*p*ea	47	w	*w*ay
6	aa	b*o*b	27	t	*t*ea	48	y	*y*acht
7	aw	b*ou*t	28	k	*k*ey	49	hh	*h*ay
8	ay	b*i*te	29	dx	mu*dd*y	50	hv	a*h*ead
9	ah	b*u*t	30	s	*s*ea	51	el	bott*le*
10	ao	b*ough*t	31	sh	*sh*e	52	bcl	b closure
11	oy	b*oy*	32	z	*z*one	53	dcl	d closure
12	ow	b*oa*t	33	zh	a*z*ure	54	gcl	g closure
13	uh	b*oo*k	34	f	*f*in	55	pcl	p closure
14	uw	b*oo*t	35	th	*th*in	56	tcl	t closure
15	ux	t*oo*t	36	v	*v*an	57	kcl	k closure
16	er	b*ir*d	37	dh	*th*en	58	q	glotal stop
17	ax	*a*bout	38	m	*m*om	59	pau	pause
18	ix	deb*i*t	39	n	*n*oon	60	epi	epenthetic silence
19	axr	butt*er*	40	ng	si*ng*			
20	ax-h	s*u*spect	41	em	bott*om*	61	h#	begin/end marker
21	jh	*j*oke	42	nx	wi*nn*er			

Table 2. 61 TIMIT original phone set.

aa, ao	aa
ah, ax, ax-h	ah
er, axr	er
hh, hv	hh
ih, ix	ih
l, el	l
m, em	m
n, en, nx	n
ng, eng	ng
sh, zh	sh
uw, ux	uw
pcl, tcl, kcl, bcl, dcl, gcl, h#, pau, epi	sil
q	-

Table 3. Mapping from 61 classes to 39 classes, as proposed by Lee and Hon, (Lee & Hon, 1989). The phones in the left column are folded into the labels of the right column. The remaining phones are left intact. The phone 'q' is discarded.

각 phoneme의 input feature vector를 이용한 모델링

- Input parameters can be represented in multi-dimensional Gaussian (16~64 dimension)
  - The d-dimensional vector x is multivariate Gaussian if it has a probability density function of the following form:

$$p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

    The pdf is parameterized by the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$.
  - The 1-dimensional Gaussian is a special case of this pdf
  - The argument to the exponential $0.5(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$ is referred to as a *quadratic form*.

Multivariate Gaussians

- Instead of a single mean μ and variance σ:

$$f(x \mid m, s) = \frac{1}{s\sqrt{2p}} \exp(-\frac{(x - m)^2}{2s^2})$$

- Vector of observations x modeled by vector of means μ and covariance matrix Σ

$$f(x \mid m, S) = \frac{1}{(2p)^{D/2} \mid S \mid^{1/2}} \exp\left(-\frac{1}{2}(x - m)^T S^{-1}(x - m)\right)$$

Multivariate Gaussians

- Defining μ and Σ

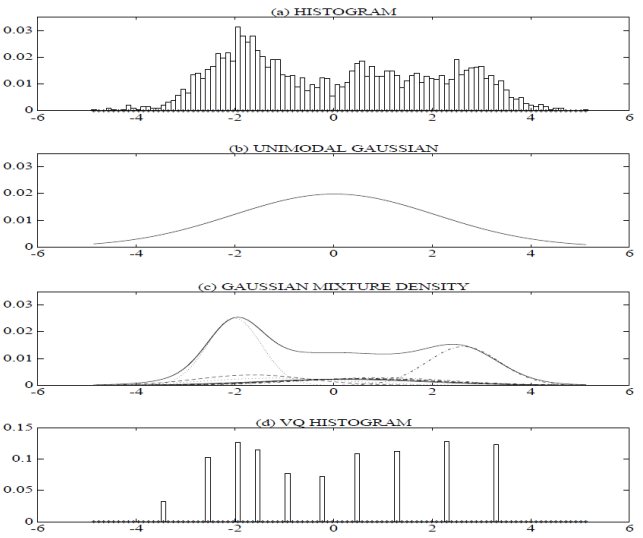
$$m = E(x)$$

$$S = E\left[(x - m)(x - m)^T\right]$$

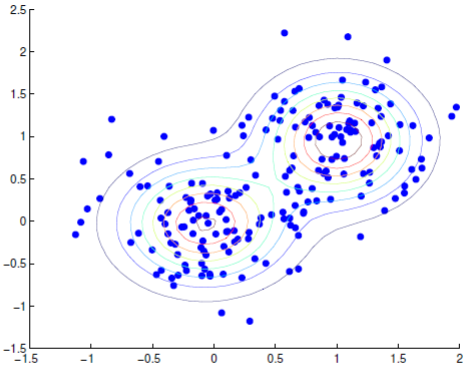
- So the i-jth element of Σ is:

$$s_{ij}^2 = E\left[(x_i - m_i)(x_j - m_j)\right]$$

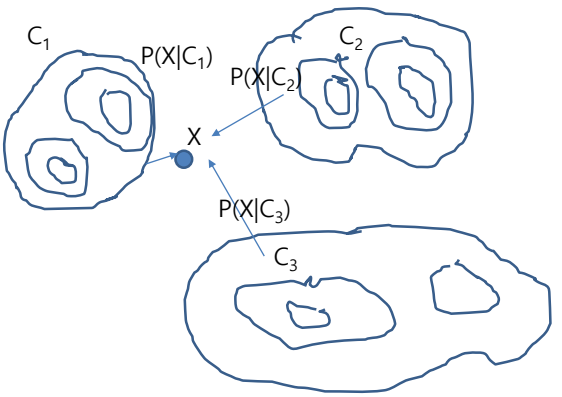
Gaussian Mixture Modeling



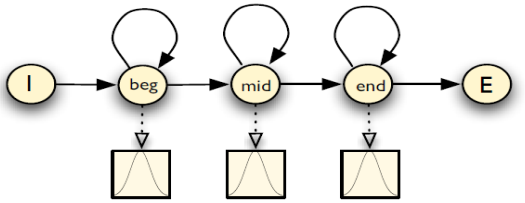
Gaussian mixture model of phones



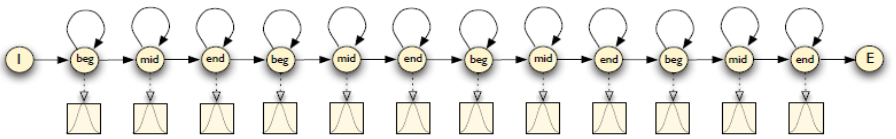
Fitted with a two component GMM using EM



Three state phone models (triphone)



/ih/



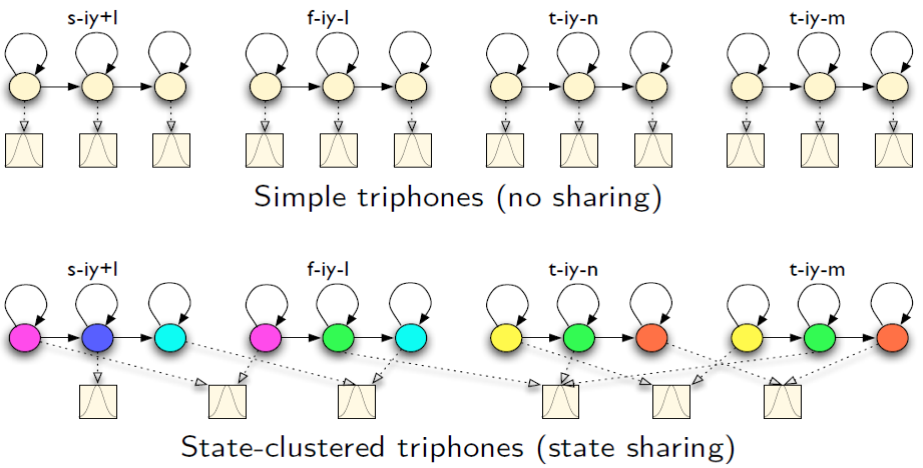
/s/ /ih/ /k/ /s/

"six"

Context dependent phone models

- Triphones Each phone has a unique model for each left and right context. Represent a phone `x` with left context `l` and right context `r` as `l-x+r`
- Word-internal triphones Only take account of context within words, so "don't ask" is represented by:
  `sil d+oh d-oh+n oh-n+t n-t ah+s ah-s+k s-k sil`
  Word internal triphones result in far fewer models than cross-word models, and enable the subword sequence for a word to be known independent of the neighbouring words. But: context is not well-modelled at word boundaries.
- Cross-word triphones "don't ask" is represented by:
  `sil sil-d+oh d-oh+n oh-n+t n-t+ah t-ah+s ah-s+k s-k+sil sil`
  Note that triphone context extends across words (eg unit `n-t+ah`)

State clustering



CD-triphone models (GMM)

- How many triphones?
  - Consider a 40 phone system. The number of CD-triphones can be 40^3 = 64,000. We usually reduce it to 1000~10,000 triphone states.
- Number of Gaussian parameters?
  - 16 mixture의 경우, 16개의 Gaussian 분포를 조합 (16개의 parameter 필요)
  - 한 개의 Gaussian에는 39차의 μ, σ 가 필요. (참고, feature vector가 39차였음)
  - 즉, 하나의 Gaussian에 2*39 = 78
  - 하나의 triphone state 당 필요한 parameter 의 수 (16 mixture): 16*(1+2*39) = 1,264
  - Monophone (50개): 63,200
  - Triphone (1,000~10,000개): 10million 수준
- We need a very large amount of training data to train such a system

음성인식기 모델의 세가지 요소

(i) Feature extractor: 음성을 frame(10msec 정도 구간) 으로 나누어서 주파수 도메인의 신호 (X)로 바꾼다

(ii) $P(X|C_j)$: 어떤 음소(phoneme)에서 어떤 featur가 나올 확률을 구한다. GMM (Gaussian Mixture Model)

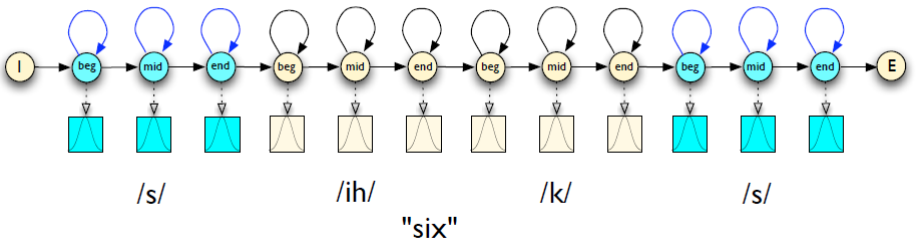
(iii) $P(C_j)$: 어떤 음소상태에 있을 a priori 정보를 구한다. (Hidden Markov Model)

Maximum likelihood (최대우도) and Maximum a Posteriori (최대 사후확율)

- Maximum likelihood classification: 소리를 듣고, 그 발음이 나왔을 가장 그럴 듯한 발음으로 판단하는 것
  - $P(X|C_j)$
  - (참고 X는 input feature, Cj 는 j 번째 class)
- Max a posteriori (MAP) 상황을 고려해서 판단하는 것
  - $P(X|C_j)\ P(C_j)$
- Speech의 경우 $P(C_j)$ 가 sequence의 형태로 나타내진다. (Sam?..) -> Hidden Markov Model (Lexicon: phoneme sequence model, Language model: word sequence model)

(iii) Hidden Markov Model

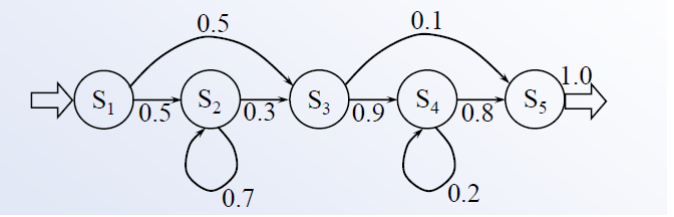
- 각각의 phone sequence 의 연결로 transition probability를 나타냄



https://www.inf.ed.ac.uk/teaching/courses/asr/2016-17/asr04-cdhmm.pdf

Markov model

- Probabilities of transitioning from one state to another (state: 지속이 될 수 있는 어떤 상태), 현재의 state에 의해 미래 state 가 결정이 됨 (매우 간단한 sequence 모델)
  - Self loop: 그 상태에 계속 있는 경우
  - 어떤 상태에서 생성되는 확률의 합은 100%
  - 단어의 발음모델, 언어모델(bigram)

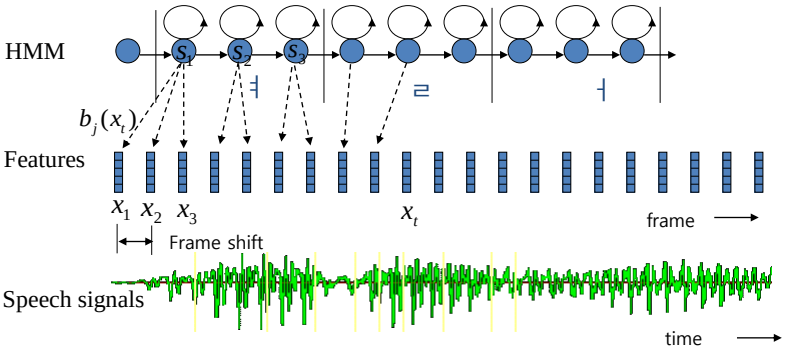


Hidden Markov model

- Hidden: 현재 어떤 state인지 불확실한 것
- 그 state에서 나오는 소리의 확률분포를 이용하여 guess를 할 수 있다.
- 이 확률분포를 GMM으로 나타낸다.

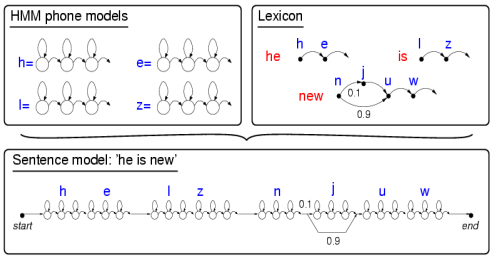
HMM과 음성신호의 대응관계

- Assume speech signals are generated by HMM
- Find model parameters from training data
- Compute probability of test speech using the HMM and select the model having maximum likelihood
- Left-to-right model: easy to model signal whose properties change over time in a successive manner – e.g., speech



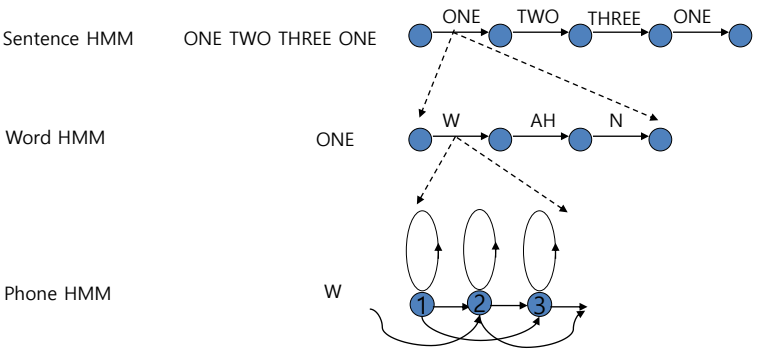
Hidden Markov model for speech recognition

- Hidden Markov model contains states (tri-phone states) and state transitions according to the speech input and the network connections. It usually combines three knowledge sources.
  - Acoustic model – Phoneme representation, (Gaussian mixture model for emission probability computation)
  - Pronunciation model – vocabulary (lexicon)
  - Language model – n-gram



음향모델 훈련용 네트워크 구성

- Sentence model = ($word_1$ $word_2$ ... $word_N$)
- Word model = ($phone_1$ $phone_2$ ... $phone_M$)
- Phone model = ($state_1$ $state_2$ $state_3$)



HMM 기반 연속음성인식 이론

- Continuous speech recognition can be formulated as:
  음성 특징벡터열 X가 주어져 있을 때, 아래의 사후확률을 최대화 시키는 단어열 W를 찾는 것.

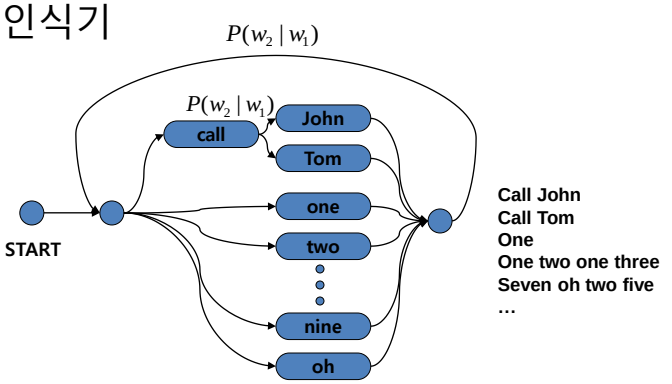
$$W' = \text{argmax}_W P(W|X)$$
$$= \text{argmax}_W P(X|W) P(W)$$

  - P(X|W): Acoustic model (ML probability)
    P(W): Lexicon and language model (a priori probability)
- 결국, 어떤 검색 알고리즘을 이용하여 모든 가능한 인식가능 공간에서 최고의 확률을 가지는 단어열을 찾는 것.

47

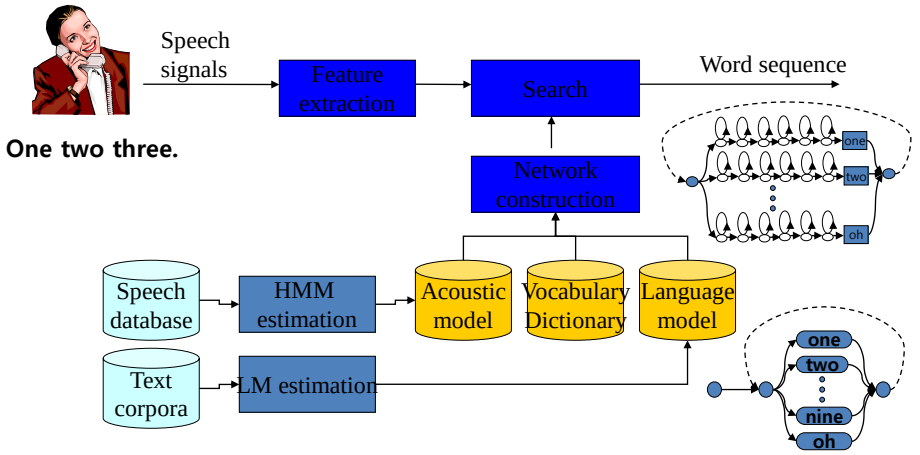
언어모델(language model)

- 지금까지의 내용으로 다음의 단어를 예측
- 이 예측확률을 이용하면 인식률 향상 됨
- 6만단어 인식 –(LM, PPL = 100)→ 실질적 100단어 인식기



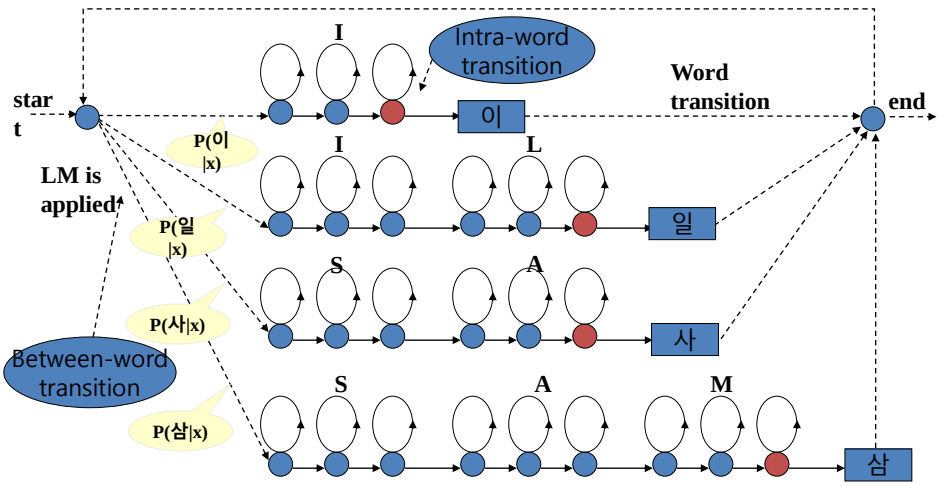
48

HMM 기반 연속음성인식 과정



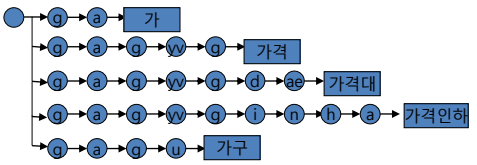
49

인식용 전체 네트워크 상세구조

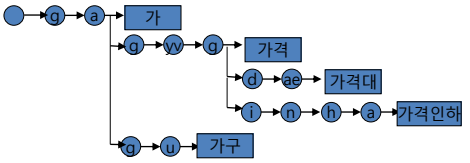


50

대어휘 처리를 위한 검색공간 최적화



Flat lexicon



Tree lexicon

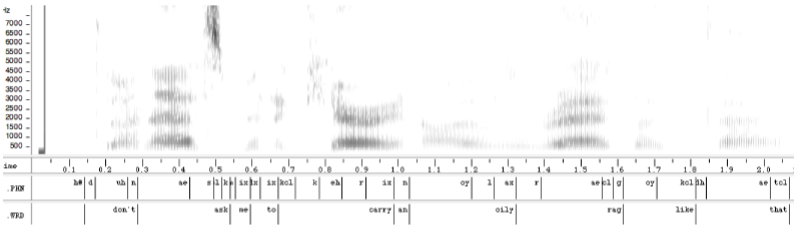
51

Training: TIMIT Corpus

About 5 hours

- TIMIT corpus (1986)—first widely used corpus, still in use
  - Utterances from 630 North American speakers
  - Phonetically transcribed, time-aligned
  - Standard training and test sets, agreed evaluation metric (phone error rate)
- TIMIT phone recognition - label the audio of a recorded utterance using a sequence of phone symbols
  - Frame classification – attach a phone label to each frame data
  - Phone classification – given a segmentation of the audio, attach a phone label to each (multi-frame) segment
  - Phone recognition – supply the sequence of labels corresponding to the recorded utterance

Labeling speech (W)



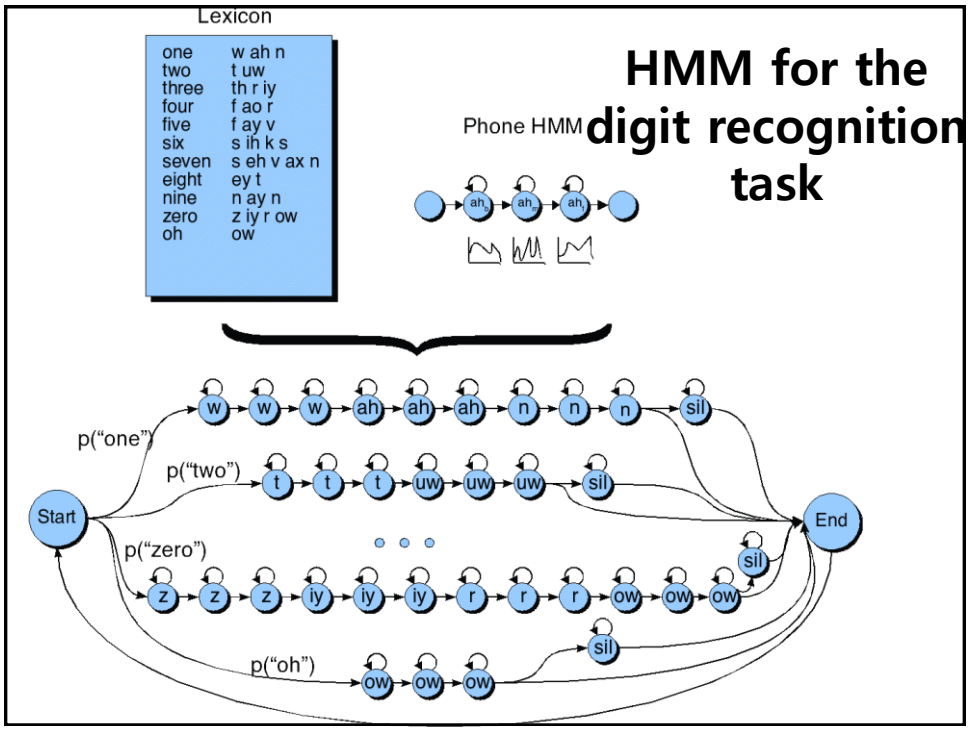
Labels may be at different levels: words, phones, etc.
Labels may be *time-aligned* – i.e. the start and end times of an acoustic segment corresponding to a label are known

Reading: Jurafsky & Martin chapter 7 (especially sections 7.4, 7.5)

Speech recognition on TIMIT

- Train a classifier of some sort to associate each feature vector with its corresponding label. Classifier could be
  - Neural network
  - Gaussian mixture model
  - ...

  The at test time, a label is assigned to each frame
- Questions
  - What's good about this approach?
  - What the limitations? How might we address them?

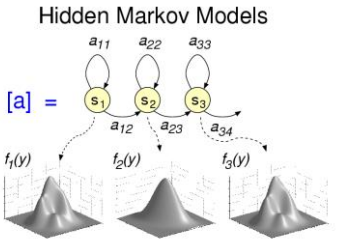


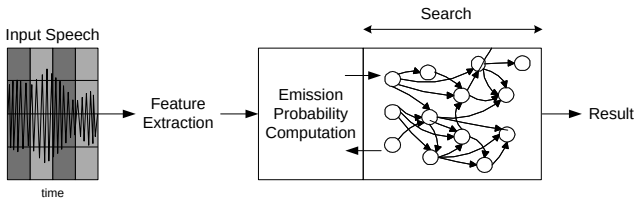
HMM network search

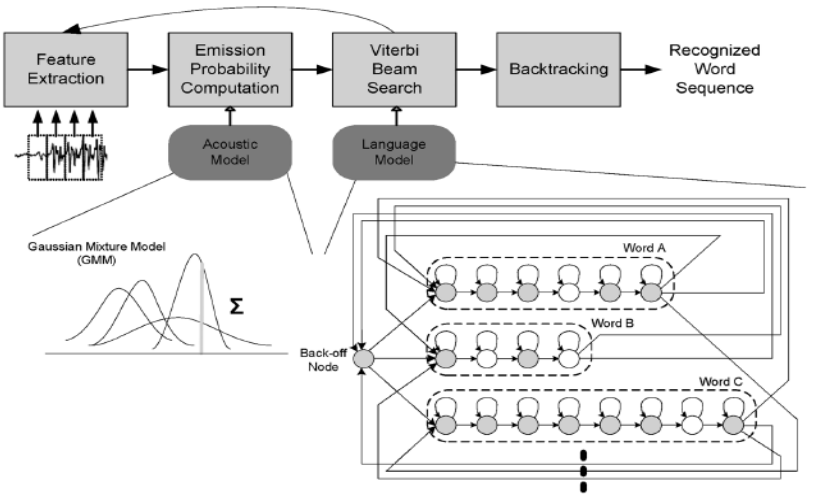
- 앞의 네트워크 모델에서 t=0에서 시작해서 t=10 frame에서 지금 'ㅅ'라는 발음을 하고 있다고 치자. 그러면 4개의 단어(일,이,삼, 사)의 첫번째 phone state와 발음(X)를 비교.
- 그러면 'ㅅ'에 해당하는 state에서 Emission probability가 크다.
- (이 때 위의 두 path는 prune이 될 수 있다)다음 순간 'ㅏ'라는 발음이 나오면 ..
- 다음 순간 'ㅁ'이라는 발음이 나오면 ...

HMM base speech recognition implementation

1. Feature extraction: speech to acoustic parameter
   - MFCC (Mel-Frequency Cepstrum Coefficient)
2. Emission probability computation
   - Generate the log-likelihood of each hypo
   - Higher dimension needed for large vocab
3. Viterbi beam search
   - Dynamic programming through the netw
   - High complex 'compare and select' opera







IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 57, NO. 8, AUGUST 2010 21

A Real-Time FPGA-Based 20 000-Word Speech Recognizer With Optimized DRAM Access

Young-kyu Choi, Kisun You, *Student Member, IEEE*, Jungwook Choi, *Student Member, IEEE*, and Wonyong Sung, *Senior Member, IEEE*

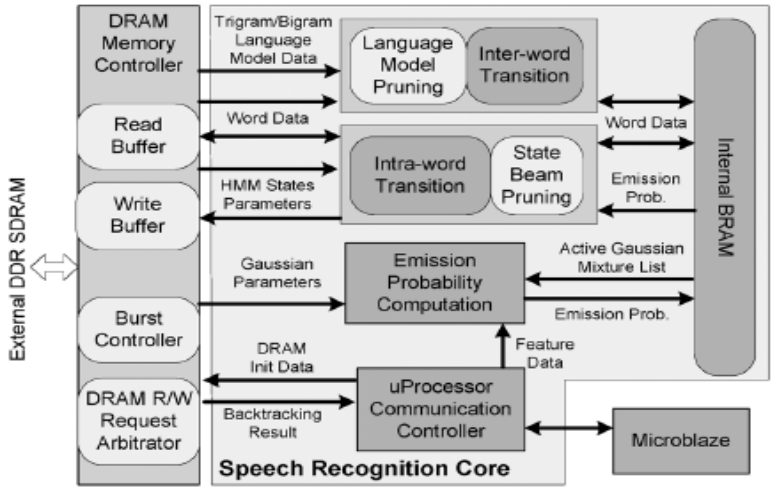


Fig. 3. Speech recognition core architecture.

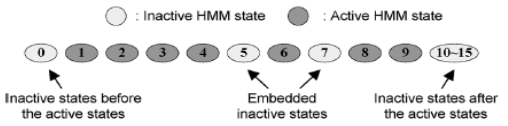
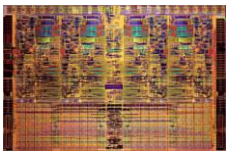


Fig. 8. Various types of fetched HMM states.

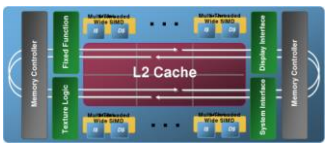
Fig. 9. Detecting start/end position of active states.

Computer architecture for speech recognition

- **Increasing SIMD width:** 4~16 arithmetic with one instruction (vector arithmetic)
- **Multicore:** Two ~ 10's CPU cores on each chip. Good single thread performance
- **GPU (manycore):** 100's of processing cores, maximizing computation throughput at the expense of single thread performance



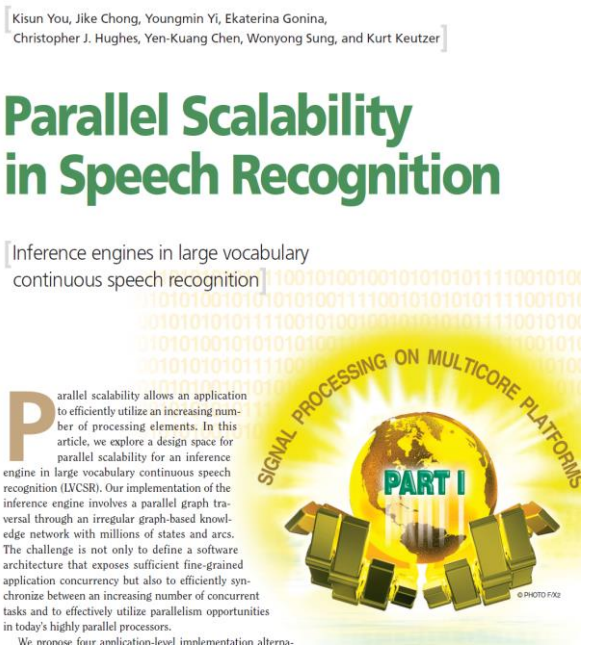
Intel Core i7 (45nm)
4 cores



Intel Xeon Phi
96 cores

NVIDIA GTX285 (55nm)
30 cores

- IEEE Signal Processing Magazine, 2009
- Kisun You



Kisun You, Jike Chong, Youngmin Yi, Ekaterina Gonina, Christopher J. Hughes, Yen-Kuang Chen, Wonyong Sung, and Kurt Keutzer

Parallel Scalability in Speech Recognition

Inference engines in large vocabulary continuous speech recognition

SIGNAL PROCESSING ON MULTICORE PLATFORMS

PART I

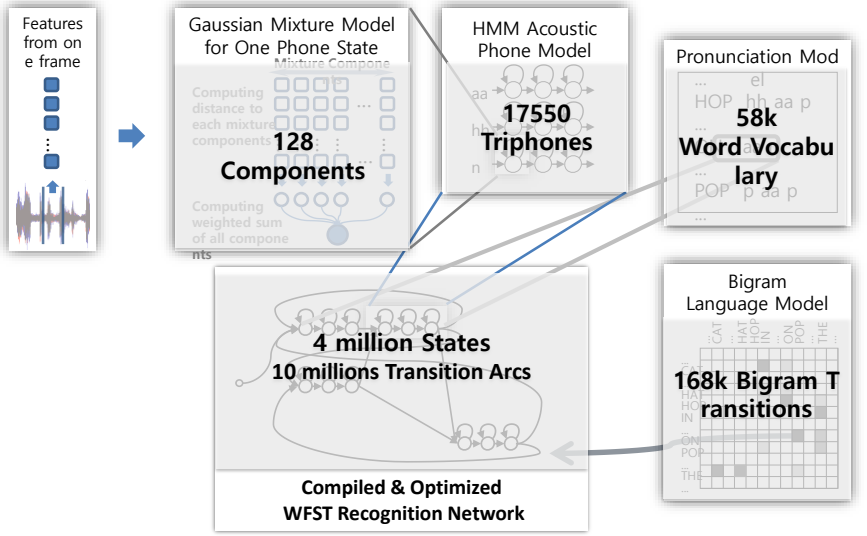
Parallel scalability allows an application to efficiently utilize an increasing number of processing elements. In this article, we explore a design space for parallel scalability for an inference engine in large vocabulary continuous speech recognition (LVCSR). Our implementation of the inference engine involves a parallel graph traversal through an irregular graph-based knowledge network with millions of states and arcs. The challenge is not only to define a software architecture that exposes sufficient fine-grained application concurrency but also to efficiently synchronize between an increasing number of concurrent tasks and to effectively utilize parallelism opportunities in today's highly parallel processors.

We propose four application-level implementation alterna-

Speech recognition for large vocabulary (>60K)

- More precise acoustic modeling with larger lexicons
  - High dimension (32~128) Gaussian mixture model
  - Thousands of tri-phone states
- ***The network complexity for HMM (WFST) grows very rapidly.***
  - We need to ***prune many states or arcs*** during the search
  - Resulting in ***very irregular computation***
- High complexity language model needed: 3-gram or higher is desired.
  - Large memory size

Recognition Network Example



Parallel scalability

- Can we achieve the speed-up of 'SIMD_width x #_of_cores'?
  - E.g. With 8-way SIMD, 8-core CPU, 64 times speed-up.
- Some parts of speech recognition algorithm is quite parallel scalable, but other parts are not.
  - Emission probability computation:
    - Computation flow is quite regular and has a good scalability.
  - Hidden Markov network search
    - Network search is quite irregular.
    - Packing overhead exists for SIMD .
    - Synchronization overhead exists.

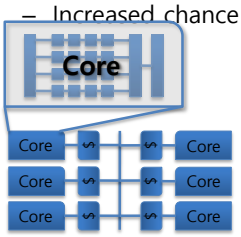
Irregular network search <-> parallel scalability

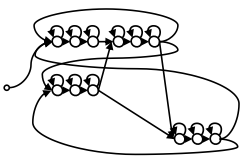
Parallel graph traversal through an irregular network with millions of arcs and states

Vector (SIMD) unit efficiency

- SIMD operation demands a packed data (packing overhead may be needed)
- Continuously *changing working set* guided by input

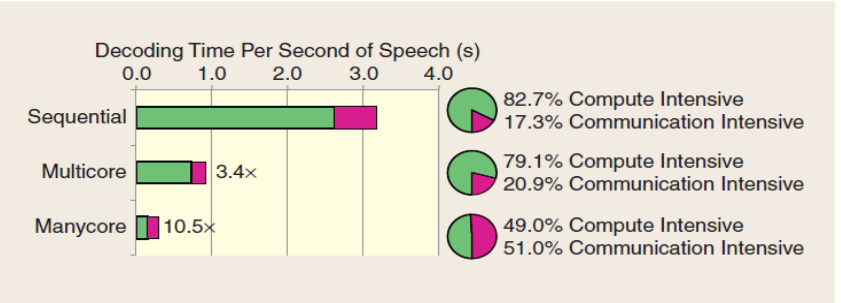
- Synchronization
  - Arc traversal induces *write conflicts* for destination state update
  - Increased chance of conflicts for large number of cores





HMM Network

Decoding time comparison



[FIG8] Ratio of computation-intensive phase of the algorithm versus communication intensive phase of the algorithm.

왜 순수인공신경망이 좋은가?

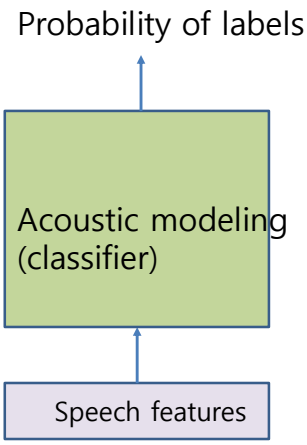
HMM

- 음성처리의 중간 정보로 tri-phone을 사용함
- HMM 중간결과를 저장하기 위해 매우 큰 DRAM space 필요
- 많은 irregular 메모리 access (전력소모, 병렬처리 부적당)
- 높은 정밀도 연산(32비트 부동소수점)
- 총 연산회수는 적을 수 있음, CPU구현시 유리

Fully neural

- 음성이 들어가면 그냥 글자가 나옴 (end-to-end)
- 중간결과는 인공신경망에 의해 매우 압축된 형태로 저장됨 (사이즈 작음, RNN)
- Parameter size가 많지만 inference 시 read-only임
- 매우 높은 병렬도(matrix-vector 연산)
- 낮은 정밀도 (4비트~8비트)
- 저전력 단일 칩 형태로 구현하기 쉬움 (?)

Acoustic modeling classification



- Context dependent tri-phone states - modeling one frame, about 10K labels
- Context independent mono-phones, about 40~50 labels
- Graphemes (characters)
- Word pieces
- Words - modeling about 1 sec, 100K labels

AM: CD tri-phone states vs word or characters

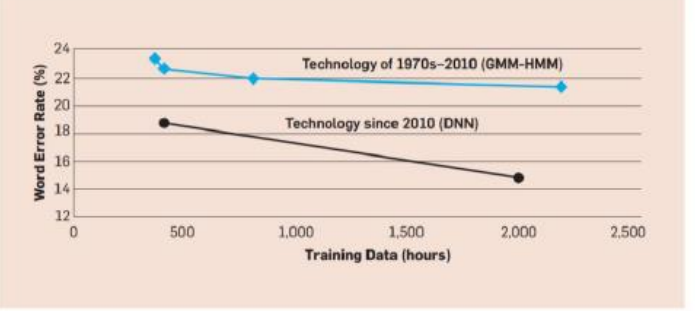
- (intra frame) CD tri-phone states: 10 msec (overlap 25 msec) of speech frame (stationary) is mapped to probability of CD tri-phone states (the number over 10K in many cases)
  - Since speech is stationary, this is similar to simple classification problem (MNIST)
  - Can be done using FCDNN, CNN
- (inter frame) Word or characters: many input speech frames (say 100 frames, non-stationary) are converted to one label
  - **Sequence recognition problem** (more difficult problem)
  - Needs RNN or CNN for sequence recognition
  - We call it end-to-end approach

Choice of AM labels

As AM does more, decoding is simpler
Simple AM can be more accurate

- CD tri-phone states:
  - AM is simple and can be accurate
  - The decoding should consider the search space of more than 1 million states (HMM model based) (very large). The decoding time-interval is 10msec or 20 msec (very frequent)
  - Usually employ HMM or WFST network
- Words and characters:
  - End to end approach (AM is more difficult to train)
  - The search space is much smaller
  - Usually RNN based LM and beam search

GMM vs DNN performance according to the data size

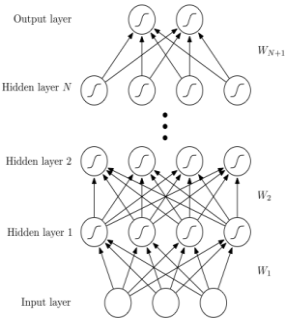


[X. Huang, et. al. 2014]

DNN based CD tri-phone states modeling

Hinton, Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." *IEEE Signal Processing Magazine* 29.6 (2012):

- Fully connected deep neural network (FCDNN)



FFDNN structure

Activation Function Bias vector

$$\mathbf{y}_{k+1} = \phi_{k+1}(\mathbf{W}_{k+1}\,\mathbf{y}_k + \mathbf{b}_{k+1})$$

Signal vector Weight matrix

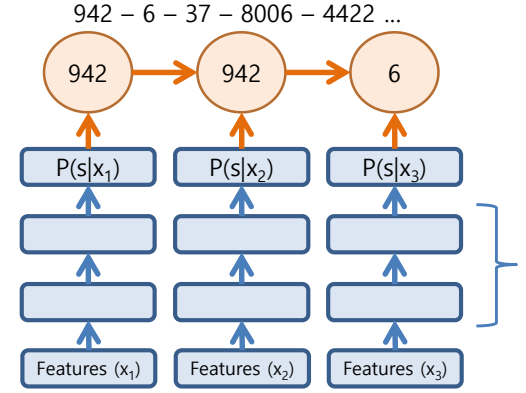
The size of weight matrix is about 1000x1000
Matrix-vector multiplication followed by non-linearity
About 3~10 layers

DNN Hybrid Acoustic Models

Transcription: Samson

Pronunciation : S – AE – M – S –AH – N

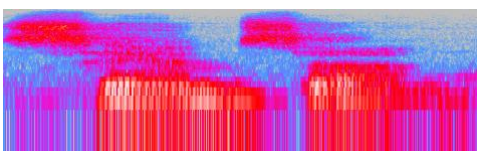
Sub-phones : 942 – 6 – 37 – 8006 – 4422 …



Hidden Markov Model (HMM):

Acoustic Model:

Audio Input:



From Stanford CS224S Andrew Maas

Input features

- DNN: multi-frame (+/- 10 or so) mel-frequency filter bank, fMLLR (speaker adaptation)
  - 39 MFCC * 21 frames -> 819 (good for DNN dimension)
  - DNN (and CNN) is supposed to have the ability of removing the correlation (thus mel-freq-filter-bank may work as good as (or better) than MFCC

Training procedure for DNN

We need frame wise reference labels

- Build a GMM-HMM system with CD-Triphone
- Forced alignment using the GMM-HMM system
  - For each frame, the CD-triphone labeling is generated
- Selecting the DNN architecture, and number of layers, number of units - FCDNN, CNN
- Training the DNN with the cross-entropy loss (reducing classification error for each frame!, not WER -> discriminative loss function as a second step)
- Optimized training
  - SGD, AdaGrad, Nesterov's Accelerated Gradient, Dropout,

Complexity in softmax function

- As the number of labels for CD-triphone states is very large (around 10K), the final output layer is considerably complex – demanding many arithmetic operations
  - FCDNN with layer size 1000, each layers consumers 1 M parameters
  - The final layer demands 1K*10K = 10M parameters

$$\hat{y}_j = \frac{\exp(W_j^{(L)T} h^{(L-1)} + b_j^{(L)})}{\sum_{k=1}^{N} \exp(W_k^{(L)T} h^{(L-1)} + b_k^{(L)})}.$$

Output layer:
51% of all parameters for a 36M DNN
6% of all parameters for a 200M DNN

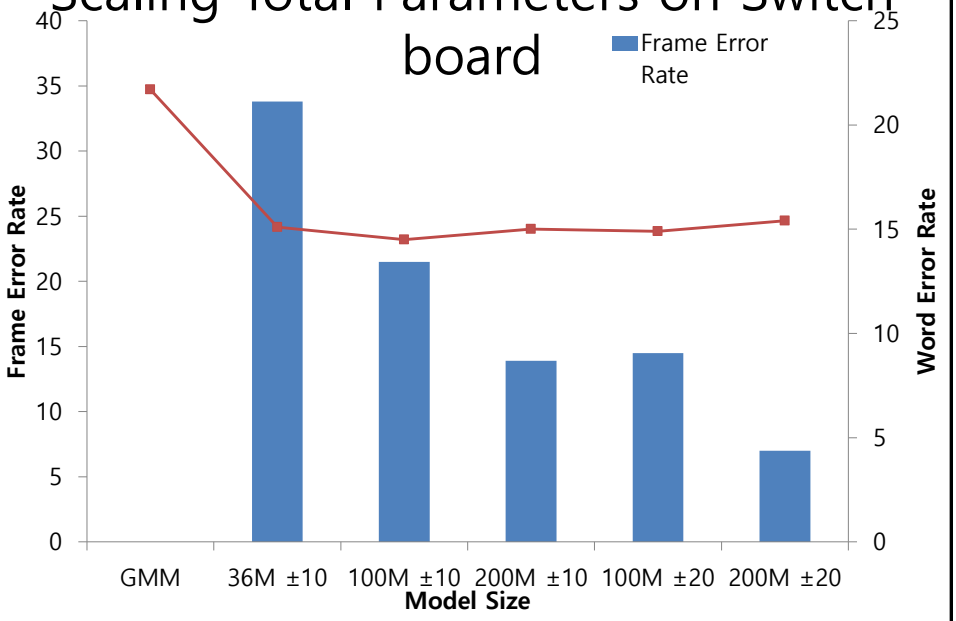
Hybrid Systems (DNN-HMM) no w Dominate ASR

[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	

Hinton et al. 2012.

Scaling Total Parameters on Switch board



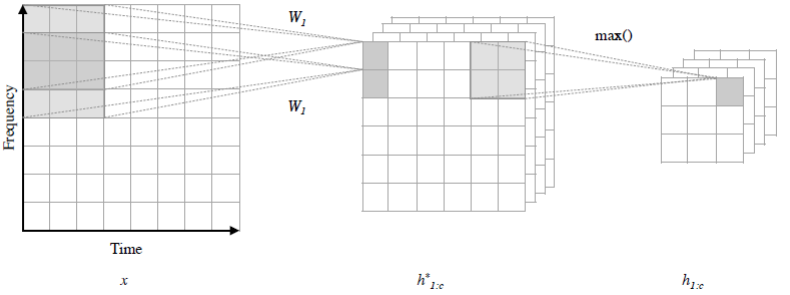
(Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. 2017)

Building A Strong DNN Acoustic Model

- Large
- At least 3 hidden layers
- Training data
- Less important:
  - Dropout regularization
  - Specific optimization algorithm settings
  - Initialization (we don't need pre-training)

(Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. 2017)

Time frequency domain CNN

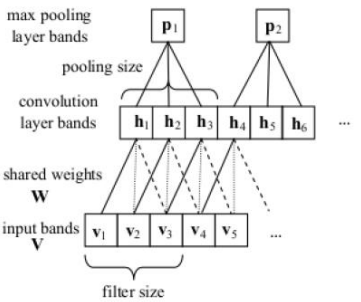


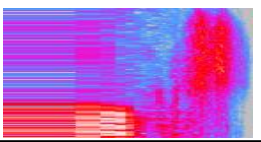
Pooling in time axis helps reducing the number of operations in the DNN laye

Convolutional Networks

- Slide your filters along the frequency a xis of filterbank features
- Great for spectral distortions (e.g. shor t wave r

# Conv/Fully Connected Layers	WER
No Conv, 6 full (DNN)	22.6
1 conv, 5 full	22.3
2 conv, 4 full	19.9
3 conv, 3 full	21.2





(Sainath, Mohamed, Kingsbury, & R amabhadran. 2013)

Comparing CNNs, DNNs, & GMMs

- GMM system: speaker adapted, discriminatively-trained, 9300 states and 150K Gaussians
- Baseline DNN trained with fMLLR features
- CNN trained with vtln-warped, log-mel+d+dd features
- All feature-based networks have 512 output targets, cross-entropy +sequence trained

Model	Hub5	rt03 FSH	Rt03 SWB
Baseline GMM/HMM	14.5	17.0	25.2
Hybrid DNN	12.2	14.9	23.5
Hybrid CNN	**11.5**	**14.5**	**22.1**

Slide from Tara Sainath

Conclusion

- Acoustic modeling is the most important part of speech recognition pipeline
- Frame-wise recognition using CD-triphone states can be done with FCDNN or CNN, but needs complex decoding using HMM or WFST
- Word or word-piece based modeling is the direction of AM, but needs a large corpus and complex DNN models (RNN).