Speaker Representations

From i-vectors to end to end systems

Deepu Vijayasenan

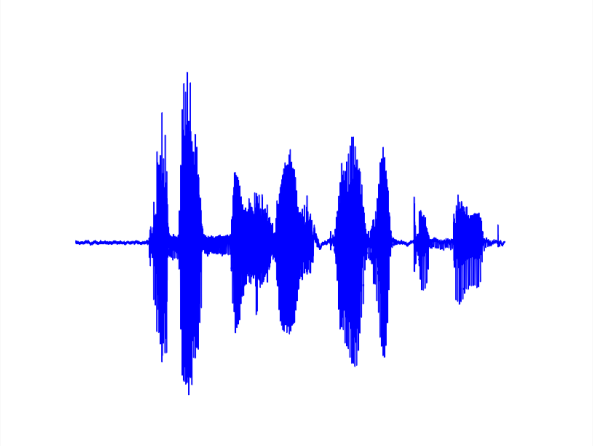
National Institute of Technology, Karnataka

Table of contents

Introduction

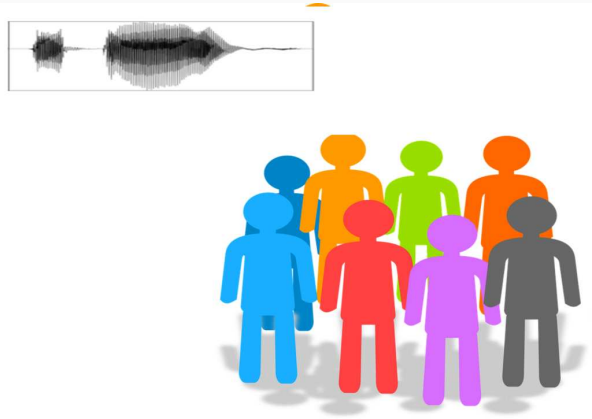
Speech Signal

- Text Information
- Speaker Identity
- Language Identity
- Emotion, articulation, ...

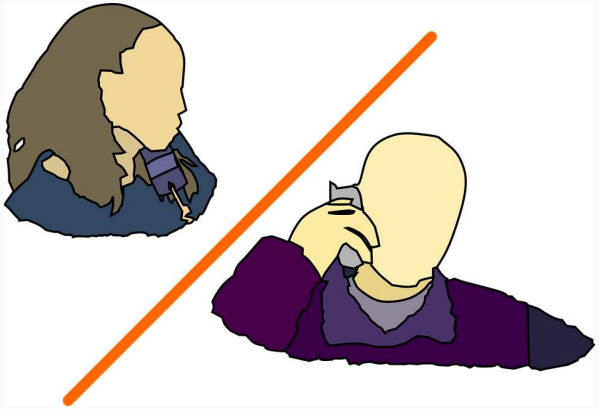


Speaker Identity

- Speaker Recognition
- Multiclass problem
- Reject Imposters



- Speaker Verification
- Verify claimed speaker identitiy
- Binary problem - genuine/imposter

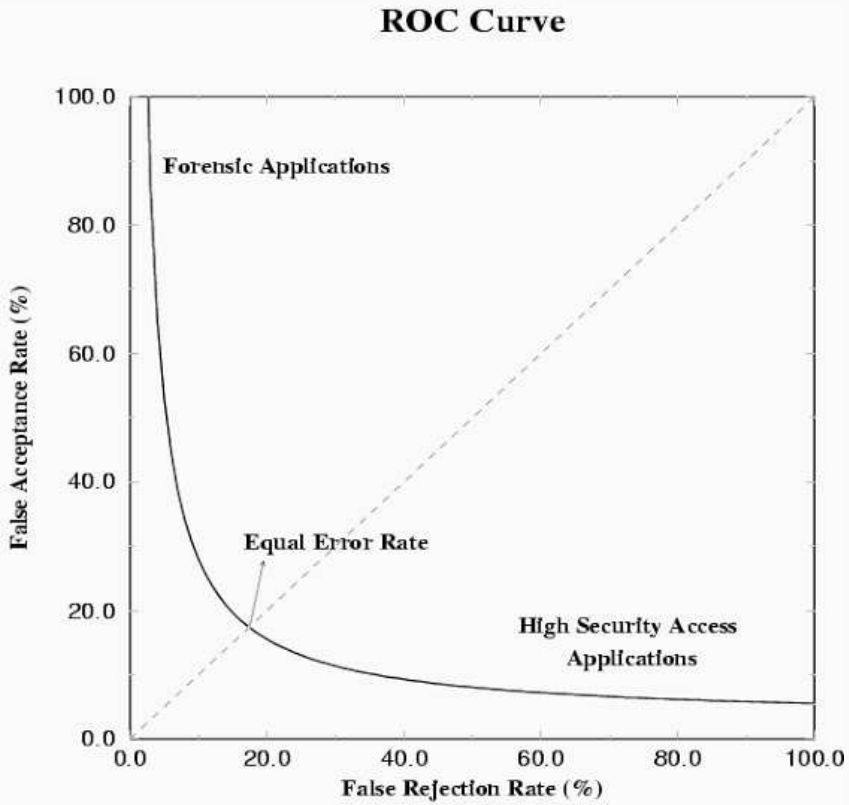


Speaker Identification

- Map the utterance to a known speaker or declare an imposter
- Limited data per speaker for enrollment
- Often new speakers needs to be enrolled dynamically
- Cannot be modeled as a classifier

Performance Measure

- Speaker systems needs to minimize FAR and FRR
- FAR/FRR trade off can be controlled by changing the score threshold



i Vectors

Pre-DNN State of the Art

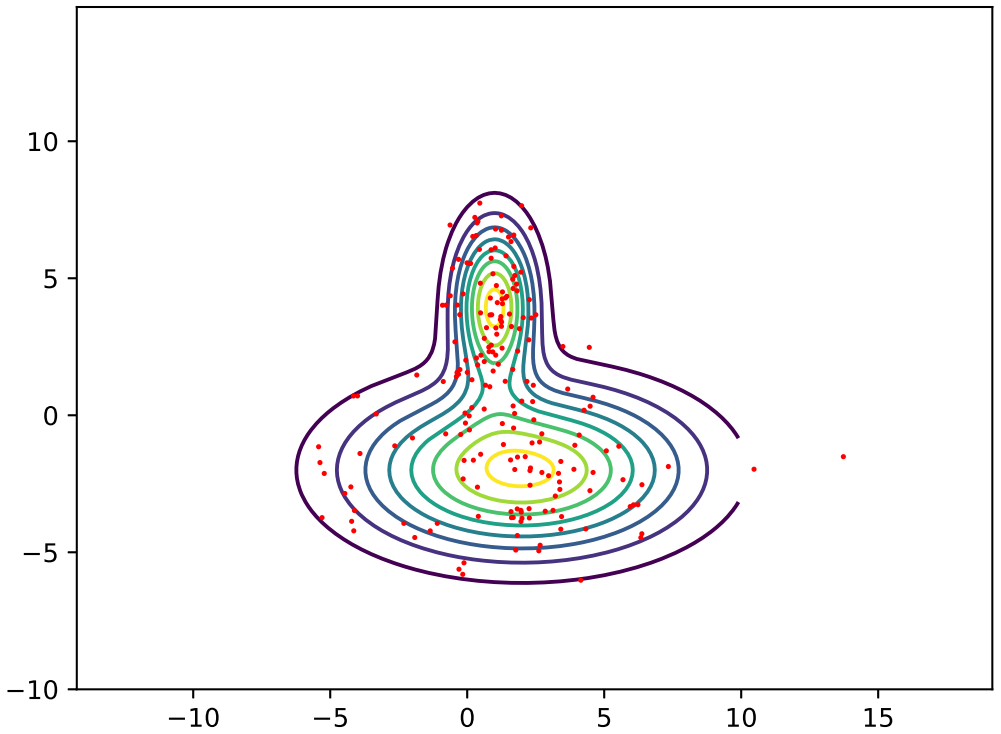
- Comparing a background model and the speaker model
- Universal Background Model
- i-Vectors
- Classification/Verification

Universal Background Model

- Gaussian Mixture Model $\lambda = \{\omega_i, \mu_\mathbf{i}, \mathbf{\Sigma_i}\}$

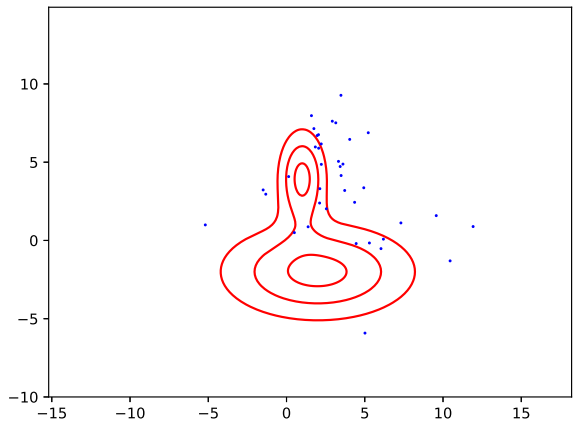
$$f_x(x) = \sum_{i=1}^{C} \omega_i [(2\pi)^d |\Sigma_i|]^{-\frac{1}{2}} \exp[-\frac{1}{2}(\mathbf{x} - \mu_i)' \Sigma_i^{-1} (x - \mu_i)] \qquad (1)$$

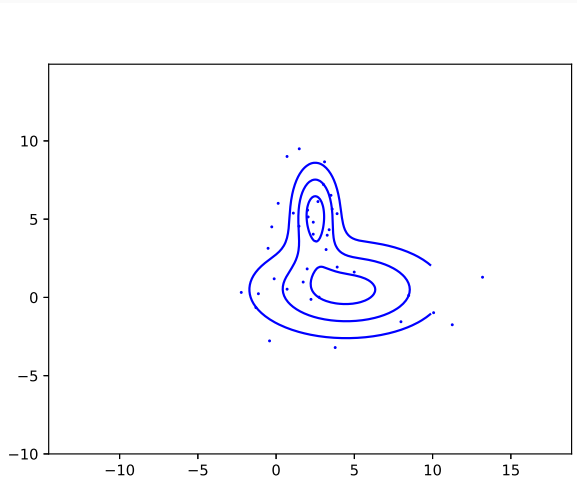
- Trained over large amount of data
- Could represent (sub) phoneme classes



GMM Adaptation

- Adapt the UBM with speaker data
- Typically Means, weights are adopted
- New model is $\lambda = \{\omega_i(s), \mu_\mathbf{i}(\mathbf{s}), \mathbf{\Sigma_i}\}$





i-Vector

- Supervectors are concatenated means

$$\mathcal{M}(s) = \begin{bmatrix} \mu_1(s) \\ \vdots \\ \mu_C(s) \end{bmatrix}$$

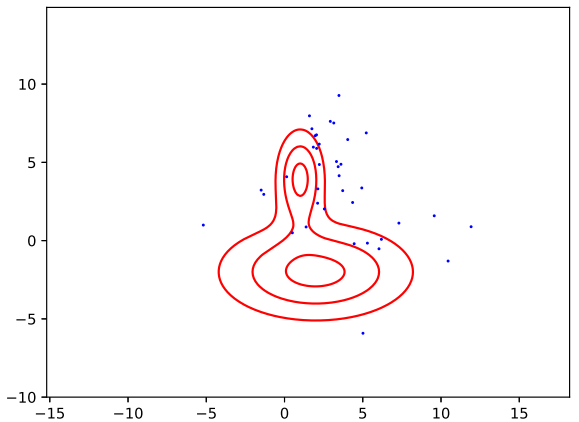
- The i-vector model

$$\mathcal{M}(s) = \mathcal{M}_0 + \mathbf{V}\mathbf{y}(\mathbf{s})$$

- $\mathbf{y}(\mathbf{s})$ i-Vector – $\mathcal{N}(\mathbf{0}, I)$
- $\mathbf{V}$ is the total variability matrix

Baum Welch Statistics

$$N_i(s) = \sum_t p_\lambda(i|x_t^s)$$
$$S_{X,c}(s) = \sum_t p_\lambda(i|x_t)(x_t^s - \mu_c)$$

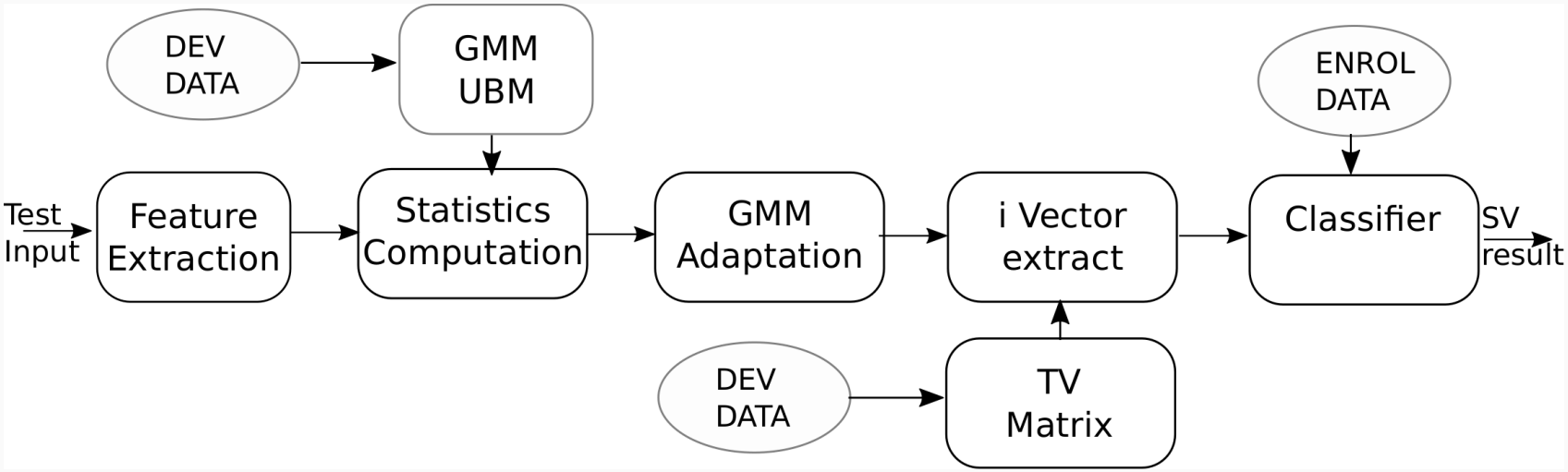


i-Vectors and the total variability matrix are computed using an EM algorithm [1]

[1] Kenny, Patrick, Gilles Boulianne, and Pierre Dumouchel. "Eigenvoice modeling with sparse training data." IEEE transactions on speech and audio processing 13.3 (2005): 345-354

Classification

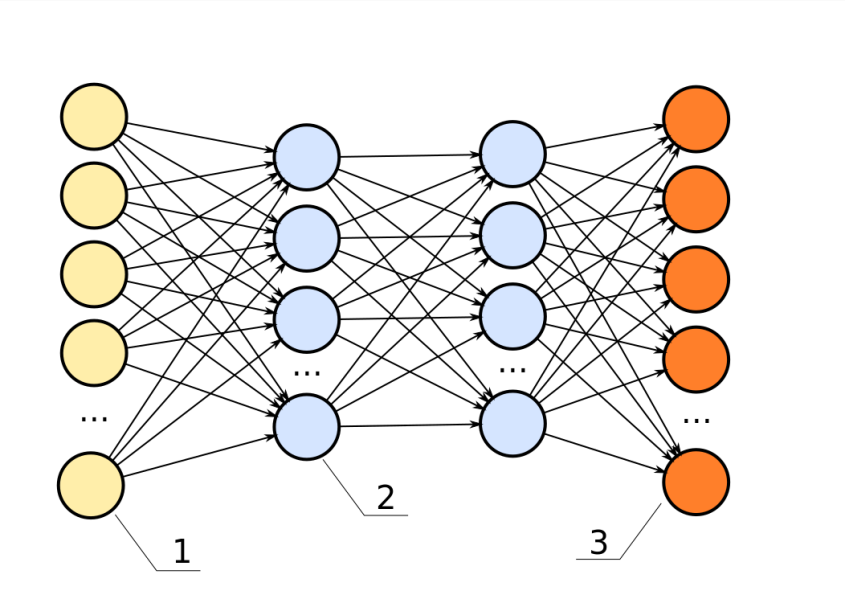
- Compare the i-vector of the test sample with speaker i-vector
- Cosine Distance/PLDA
- Score normalization
- Channel normalization



Deep Neural Networks

Neural Networks

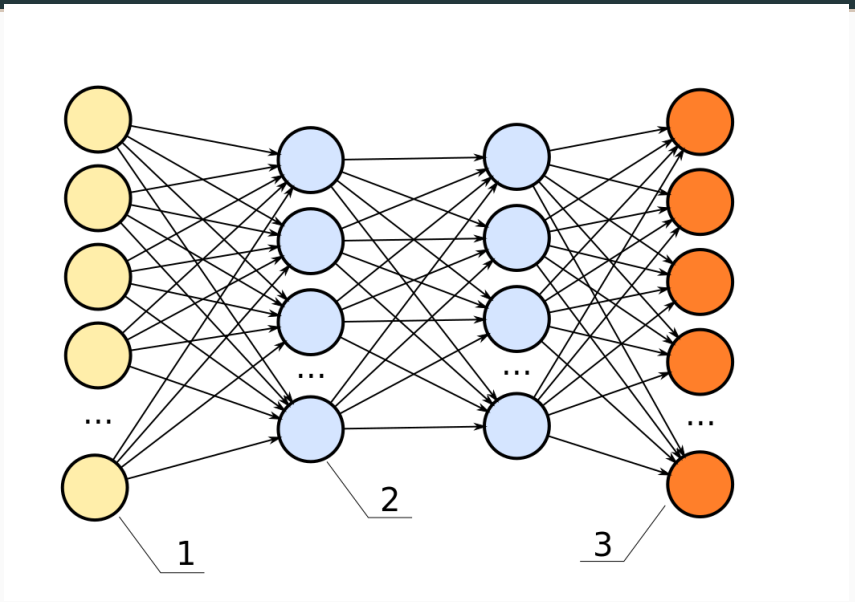
- Supervised training algorithms
- Learning from known input output samples
- Minimizes an objective function
  - Cross Entropy
  - Mean Square Error
- Training using back propagation algorithm



Neural Networks in Speech

- Deep Neural Networks to predict senone (Context dependent Phoneme) posteriors
- Senone posteriors are derived from a force alignment from HMM-GMM
- LVCSR systems contains thousands of senons
- Different Deep NN architectures are employed
  - Feed forward architecture
  - Convolutional Network
  - Recurrent Networks

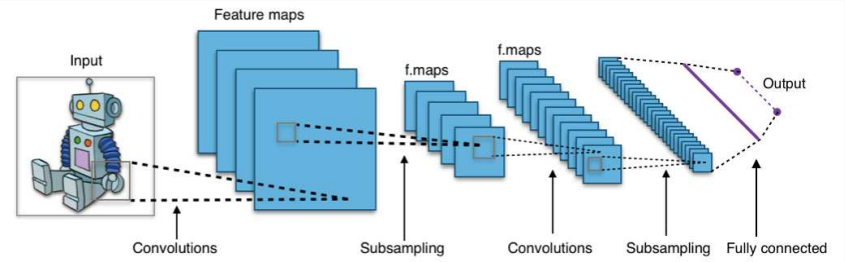
Feed forward Neural Network



- Deep NN example [2]
- 4-5 layer networks upto 2048 hidden neurons per layer
- 11 frames of 39 MFCC coefficients input
- 183 target classes
- Phone Error Rate of 23%

[2] Mohamed, Abdel-rahman, George Dahl, and Geoffrey Hinton. "Deep belief networks for phone recognition." Nips workshop on deep learning for speech recognition and related applications. Vol. 1. No. 9. 2009.

Convolutional Neural Network

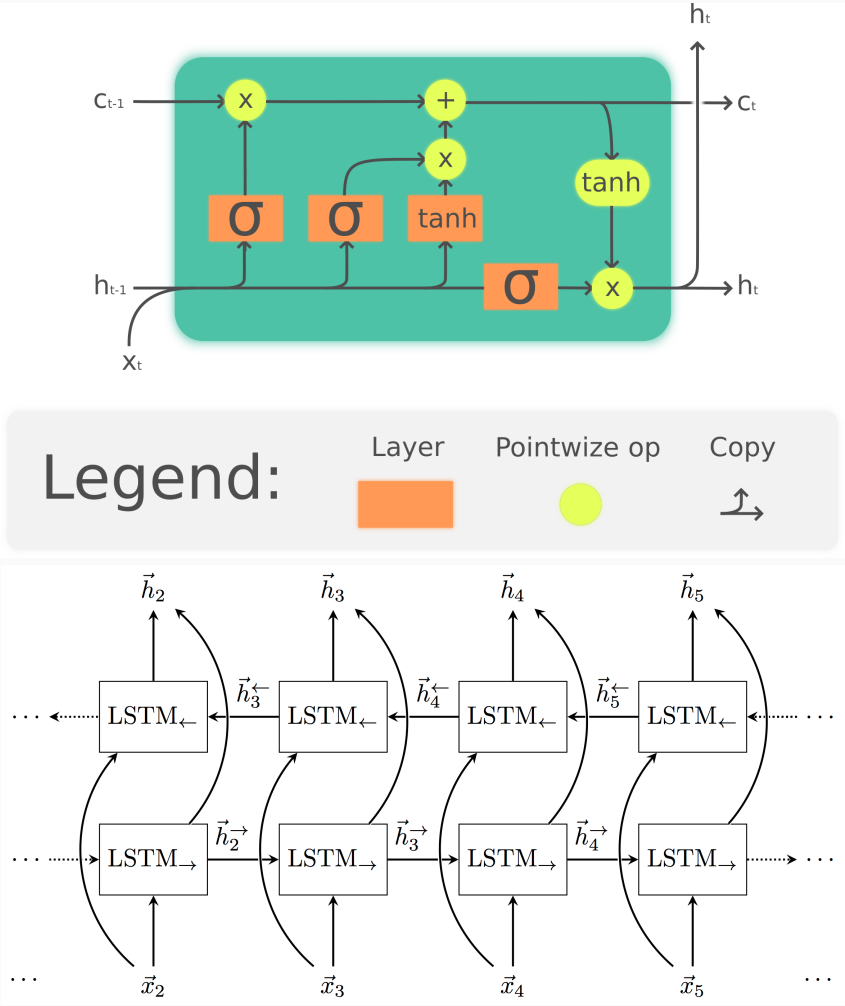


- Convolutional Network for LVCSR [3]
- 6 layer network - 2 convolutional layers - 128/256 filters , 4 fully connected layers
- 9 MFCC input
- 10 - 12 % improvement over the DNN

[3] Sainath, Tara N., et al. "Deep convolutional neural networks for LVCSR." Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on. IEEE, 2013

Recurrent Neural Networks

- Sequence nature of the speech is addressed
- Bidirectional LSTM example[4]



- single MFCC input, 61 phoneme posterior output, 250 LSTM cells
- Gates remember an appropriate context

[4] Graves, Alex, Navdeep Jaitly, and Abdel-rahman Mohamed. "Hybrid speech recognition with deep bidirectional LSTM." Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, 2013.

Deep NN in Speaker Identification

Followed the trend in ASR

- Used to derive for BW statistics
- Intermediate output as feature inputs
- Used in end to end systems

DNNs replaces GMM

BW Statistics

- BW statistics is computed from the senon posteriors from GMM
- Use Neural network senon posteriors in place of GMM posteriors
- Same i-vector computation, classification follows
- Deep NN replacing the UBM
- Trained on auxiliary data
- Could be even trained on an alternate feature
- DNN posteriors are often better than GMM posteriors

BW statistics from NN

- Deep NN to derive senone posteriors[5]. [6]
- Fully connected Deep NN

$$N_i(s) = \sum_t p_\lambda(i|x_t^s)$$
$$S_{X,c}(s) = \sum_t p_\lambda(i|x_t)(x_t^s - \mu_c)$$

- 7 consecutive frames of input features
- 7 layered DNN
- PLDA/Cosine/SVM classification
- Close to 30% relative reduction in EER

[5] Kenny, Patrick, et al. "Deep neural networks for extracting baum-welch statistics for speaker recognition." Proc. Odyssey. 2014

[6] Lei, Yun, et al. "A novel scheme for speaker recognition using a phonetically-aware deep neural network." Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014

Language Identification

- Same concept could be used for Language Identification [7]
- Convolutional Neural Networks with 40 dimensional filter banks
- One convolutional layer (200 Filters) followed by 5-7 layer fully connected layers
- Close to 15% relative improvement in EER - RATS LID task short sentences
- Bigram conditional probabilities are also used as LID features for longer utterances - Error rates halves for 2 min sentences
- Combination gives a 20% relative improvement over UBM/i-vector

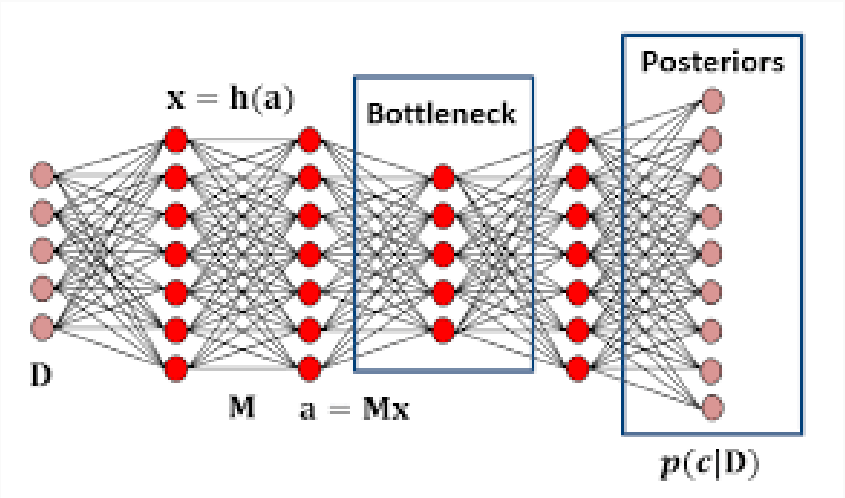
[7] Lei, Yun, et al. "Application of convolutional neural networks to language identification in noisy conditions." Proc. Odyssey-14, Joensuu, Finland 41 (2014).

Bottlenecks

Bottleneck Features(BNF)

Intermediate activations from a neural network [8]

- Lower dimensional features that contains information for posterior prediction
- Linear Layer in the neural network
- Bottleneck layer is often towards the last layers
- Could be Fully connected or CNN network
- Used as input features in HMM/GMM ASR

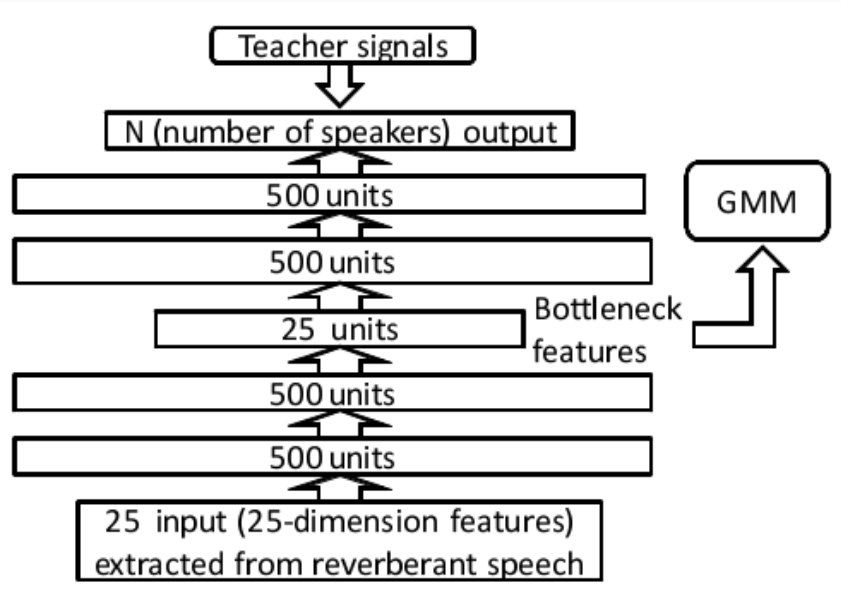


[8] Grzl, Frantisek, et al. "Probabilistic and bottle-neck features for LVCSR of meetings." Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. Vol. 4. IEEE, 2007.

BNF for speaker identification

Bottleneck Features used to train GMM for distance talking speaker id [9]

- Japanese Newspaper Article Sentences with reverberration
- Training set 50 speakers
- 25 dimensional MFCC features input, 25 dimensional BNF
- GMM Modeling of BNF
- BNF resulted in an average error reduction from 15.3% - 7.7% (40% relative)



[9] T Yamada et.al., Improvement of distant talking speaker identification using bottleneck features of DNN, Interspeech 2013, Lyon, France

BNF for Language Identification

Bottleneck features as input to i-Vector system for NIST 2009 language identification evaluation dataset [10]

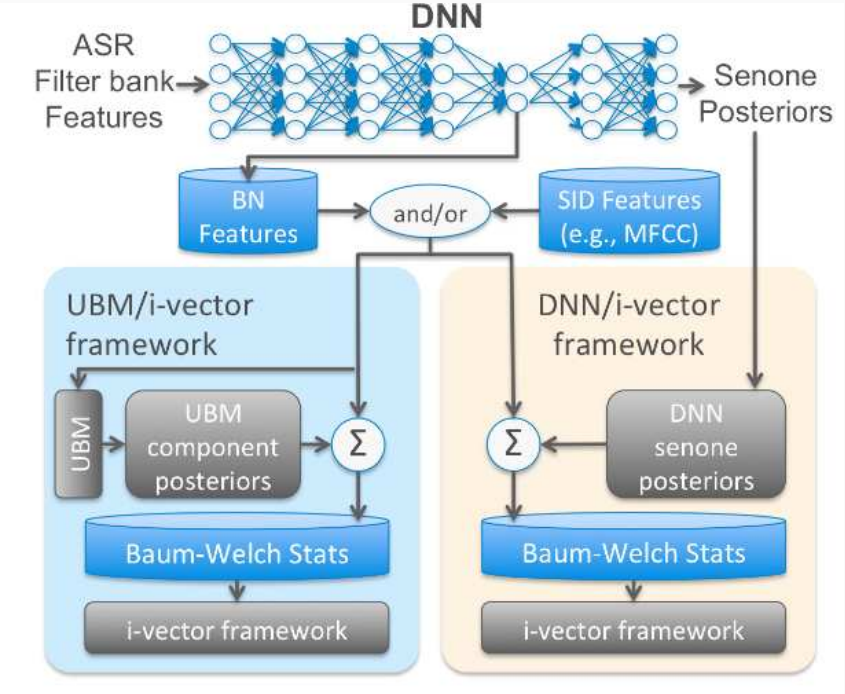
- 5 Layer DNN with 10 context-MFCC
- 80 hidden neurons
- i-Vector framework for BNF
- BNFs consistently outperform the PRLM and i-Vectors (eg: 3s duration error improvement from 14.2 to 9.7%)

[10] Yan Song et.al., i-Vector Representation based on Bottleneck Features for Language Identification

BNF/BW combination

Neural Network for BW statistics and BNF [11]

- 5 Layer DNN with 10 context-MFCC
- 80 hidden neurons
- i-Vector framework for BNF and BW statistics
- Combination found to be superior in 4/5 extended conditions in NIST SRE 12 evaluation



[11] McLaren et.al., "Advances in Deep Neural Network Approaches to Speaker Recognition, ICASSP 2015

BNF/BW for speaker and language ID

Bottleneck Features to speaker and language recognition [12]

- DNN for posteriors and Bottleneck Features, 21 frames of PLP-13 input
- 7 layers, 1024 hidden neurons, bottleneck layer 64 nodes, trained on SWB 100hr data
- Language recognition (NIST 2011 LRE) i-vector systems SDC or BNF features, BW statistics from GMM or DNN
- Speaker Recognition (DAC 13 challenge) i-vectort systems MFCC BNF features, BW statistics from GMM or DNN

[12] F. Richardson, D. Reynolds, N. Dehak, "Deep Speaker Network Approaches to Speaker and Language Recognition", IEEE Signal Processing Letters, 22 (10), 2015, pp 1671-1675.

BNF/BW for speaker and language ID

Out of Domain DAC13 Results

Feat	BW	EER
MFCC	GMM	6.18
MFCC	DNN	3.27
BNF	GMM	2.79
BNF	DNN	3.97

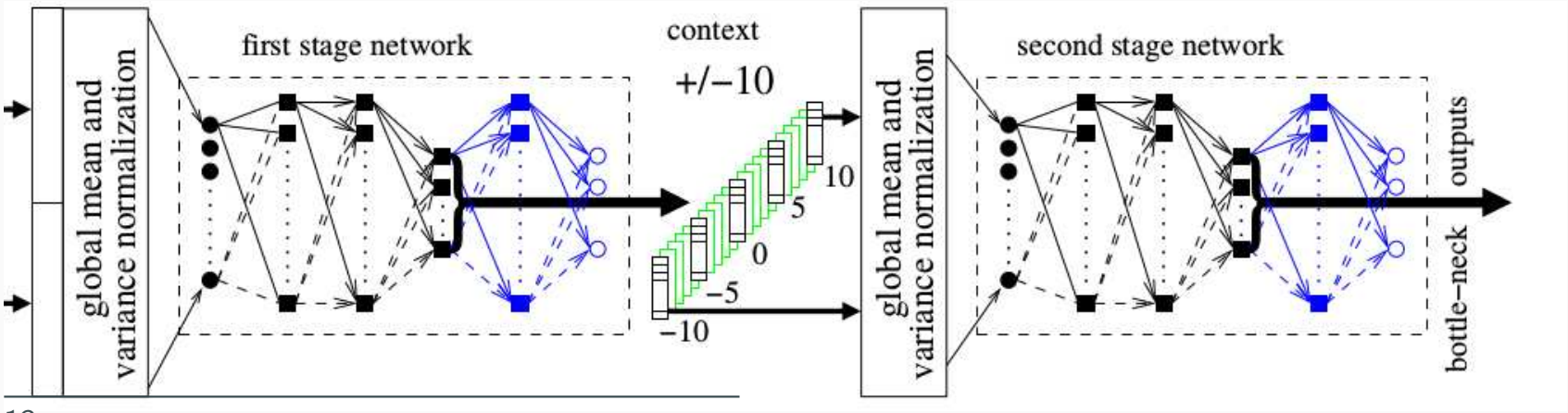
2011 LRE Results

Feat	BW	30s	3s
SDC	GMM	5.26	20.9
SDC	DNN	4.00	19.5
BNF	GMM	2.76	15.9
BNF	DNN	3.79	18.2

Stacked Bottleneck Features

Cascade of two neural netowrks (RATS LID) [13]

- FDLP input features - sub band temporal envelops
- 476 FDLP features + 11 pitch features
- First NN 5 layer DNN 1500 hidden nodes, 80 in Bottleneck
- 5 BNF features are sampled from first NN and is used to train a second NN
- Second DNN 5 layer 1500 hidden nodes, 80 in Bottleneck



[13] P Matejka et.al., "Neural Network Bottleneck Features for Language Identification", Odyssey 2014, The Speaker and Language Recognition Workshop, 2014, Joensuu, Finland

Stacked Bottleneck Features

- 400 dimensional i-vectors extracted with a 1024 component GMM
- NN Classifiers with one hidden layer
- 5 target languages and 10 non-target languages

Feat	3 s	10s	30s
PLP	18.25	12.95	10.32
SBN	13.72	6.84	4.65

Speaker Embeddings

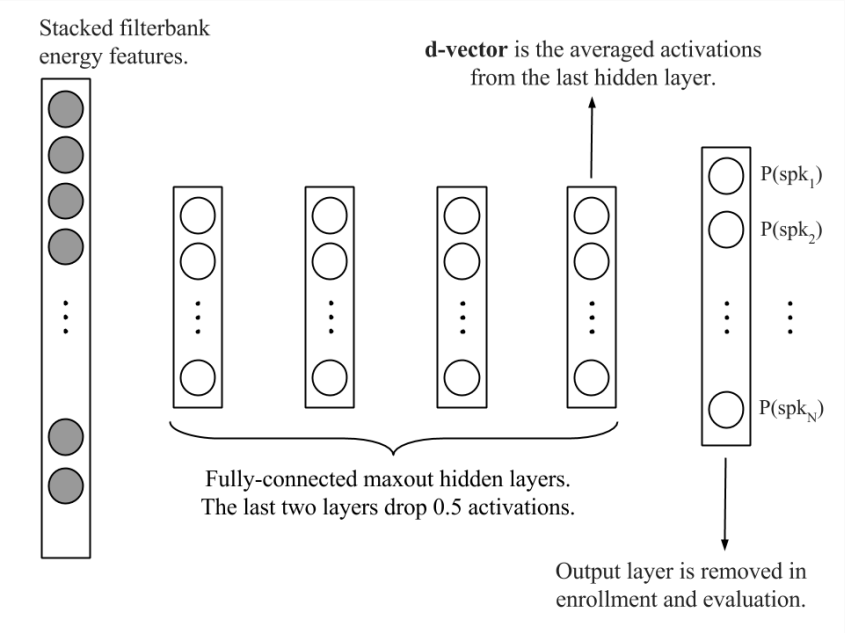
From Bottleneck to Speaker Embeddings

- Trained with senone posteriors
- No task related information is used in feature representation
- Speaker discriminability
  - Train the Neural network with speaker targets?
  - Generalize to unseen speakers in development data
- Framewise representation, need to do a ivector+PLDA
- Fixed length feature representations
  - Some kind of temporal pooling

d-vectors

Low Resource SR system [14]

- Train a neural network with 496 speaker targets
- 41 Frames of FBANK-40 input
- 4 hidden layers with 256 hidden neurons - drop out, maxout
- Enrollment – d-vector
  output of the last layer is L2 normalized and averaged over the utterance



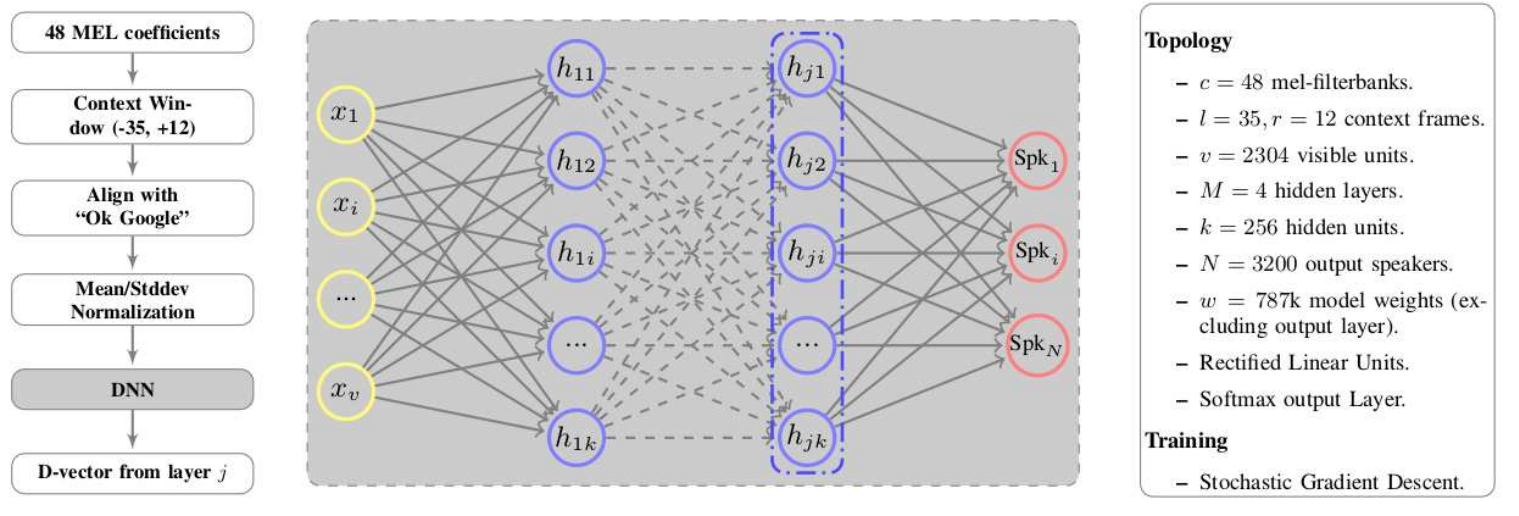
[14] Variani Ehsan et al. "Deep Neural Networks for small footprint text-dependent speaker verification". ICASSP 2014

d-vectors

- Text Dependent - "Ok Google"
- 496 speakers to train the DNN
- 150 speakers for enrollment and evaluation
- Cosine distance is used for scoring the d-vectors
- iVector+PLDA with similar number of parameters used as baseline

system	4 utter	20 utter
i-vec	2.8	1.2
d-vec	4.5	2.0

Modifying the DNN



- CNNs for dvector extraction [15]
- Initial layers are locally connected or convolutional

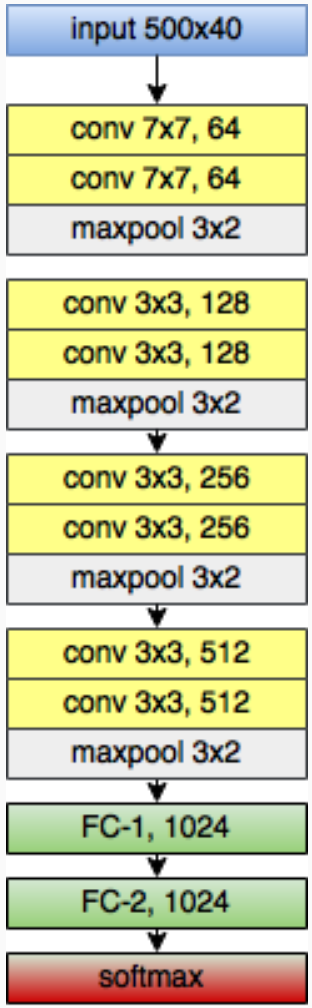
[15] Chen Y.H. et.al., "Locally Connected and convolutional neural networks for small footprint speaker recognition" INTERSPEECH 2015

Modifying DNNs

- Text Dependent - "Ok Google"
- 48x48 input blocks, FC ReLU baseline
- 2 Conv layers, 2 FC layers, 24x24 filters
- 2 locally connected layers, 12x12 blocks
- Matched for same number of parameters
- 3000 speakers - 7 utterences
- EER 3.88% $\rightarrow$ 3.6% $\rightarrow$ 3.5%

CNN embeddings

Fully Convolutional Network for short duration speech segments (5s) [16]



- VGGNet like CNN trained with cross entropy
- More aggressive temporal pooling
- Fixed 500 frames speech input, no temporal averaging
- Last hidden layer as speaker embedding
- NIST SRE 2004-08+SWB for development, 5s segments 10 SRE female data for evaluation

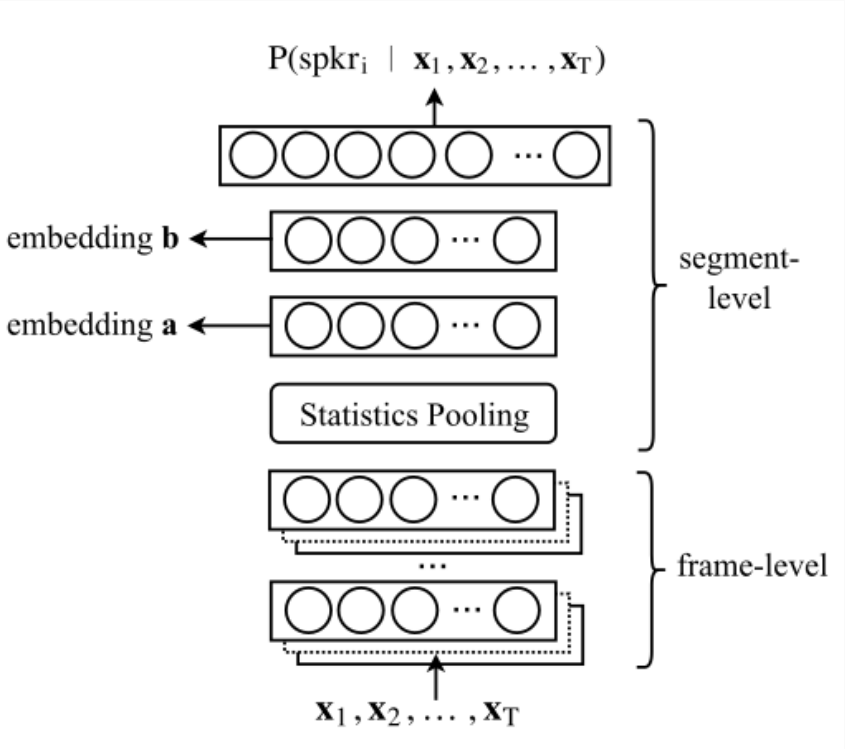
system	CD	PLDA
i-vec	31.1	24.8
convnet	23.7	23.2

[16] Bhattacharya et.al., "Deep speaker embeddings for short-duration speaker verification." Proc. Interspeech. 2017

TDNN-Statistics pooling

Towards TI longer variable dimensional input [17]

- 5 TDNN layers at the frame level
- Statistics pooling layer – mean and std.dev of each feature
- Embeddings - Two last hidden layers (512 and 300)



[17] Snyder, David, et al. "Deep neural network embeddings for text-independent speaker verification." Proc. Interspeech. 2017.

TDNN-Statistics Pooling

- Trained on SRE 04-08 and SWB,Total 6500 speakers
- Evaluated on SRE10, SRE16
- PLDA back ends, scores averaged for both embeddings
- SRE10 evaluation shows robustness to short utterances

SRE 10

system	5s	10s	Full
ivec	9.1	6.0	1.9
emb	7.6	5.0	2.6

SRE 16

system	Cant.	Taglog	Pool
ivec	8.3	17.6	13.6
emb	6.5	16.3	11.9

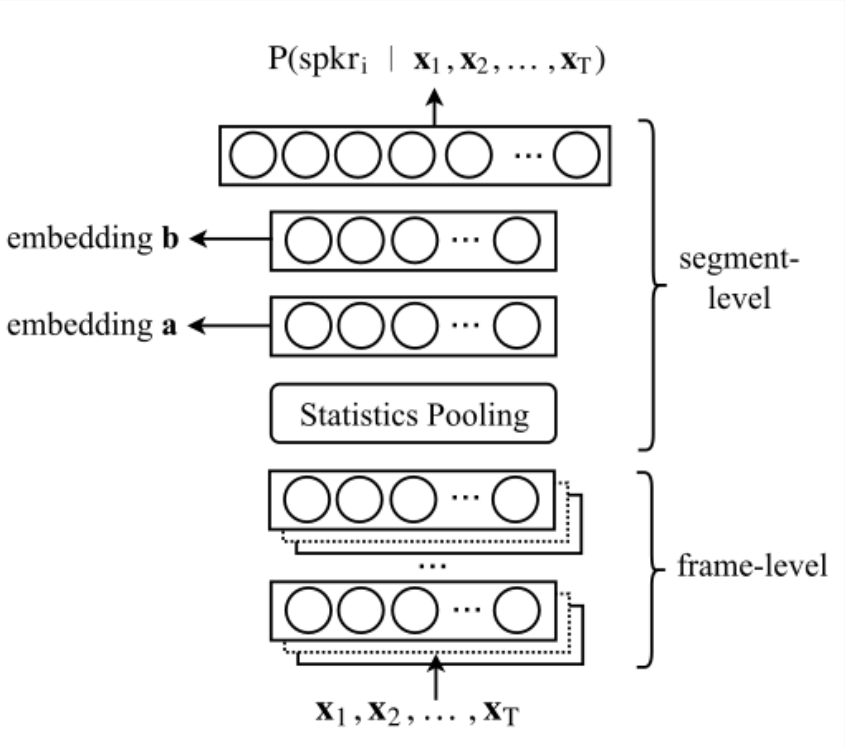
x vectors

Large scale system [18]

- Large training data - 8k+ speakers 100k+ recordings
- Data augmentation by introducing reverberration (RIR) and noise (MUSAN)
- PLDA backend

SRE 16 Cantonese

i-vector	BNF	x-vector
9.23	8.12	5.71

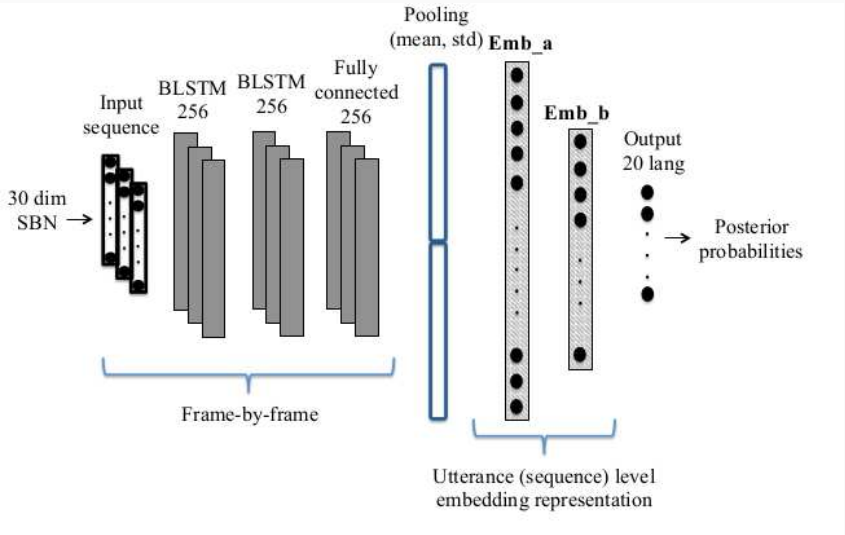


[18] Snyder, David, et al. "X-vectors: Robust DNN embeddings for speaker recognition." ICASSP (2018).

Language Identification

Similar principles applied to language identification [19]

- BLSTM layers + Mean and Std. dev pooling + Cross entropy
- 256 and 150 dimensional embeddings
- NIST LRE 2015, 300+ hours for DNN training
- Backend PCA + Gaussian Classifier



[19] Lozano-Diez, Alicia, et al. "DNN based embeddings for language recognition." ICASSP 2018.

Language Identification

- spkr embeddings + backend outperforms the i-vectors
- Posteriors do not outperform i-vectors

i-vec	postr	embedd
16.93	19.68	16.05

Self Attentive Embeddings

- Statistical Pooling averages with equal weight

$$\mathbf{h} = \frac{1}{T}\sum_{t=0}^{T-1}\mathbf{h_t}$$

  - FA from VAD
  - Noisy frames
- A relative weight of each frame needs to be learned

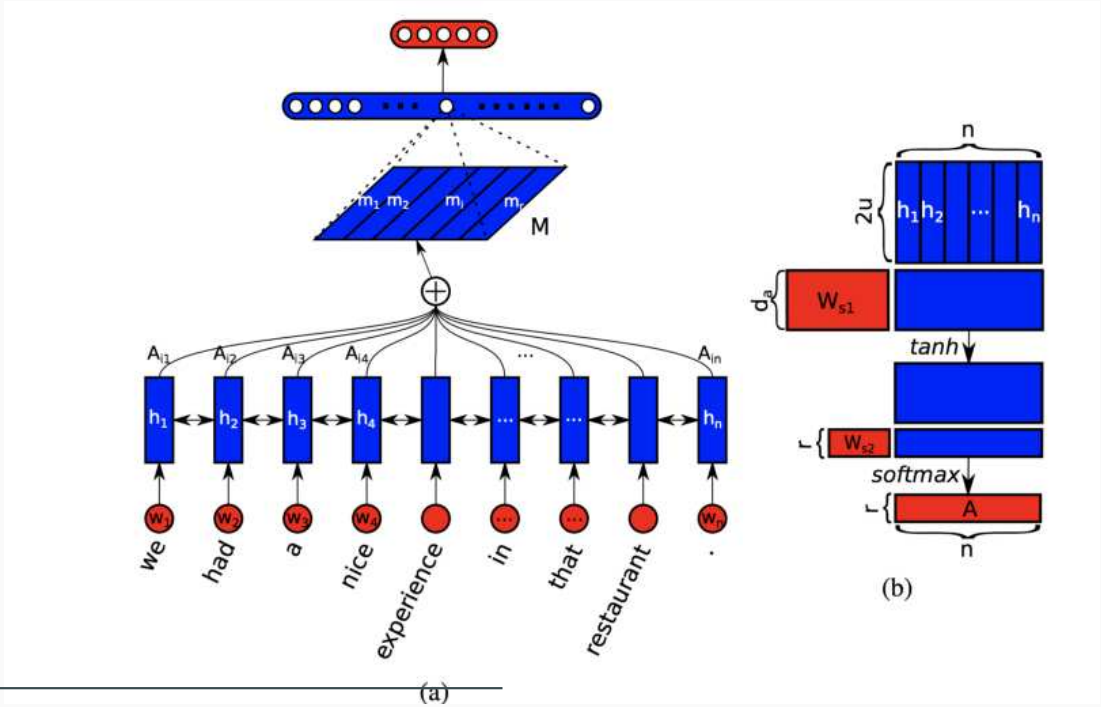
$$\mathbf{h} = \frac{1}{T}\sum_{t=0}^{T-1}a_t\mathbf{h_t}$$

Self Attentive Embedding - Training

Learn the relative weights from the frame based outputs [20]

$$\mathbf{a} = softmax(g(H^T W_1)W_2)$$

- g(.) - Nonlinearity in hidden unit (ReLU)
- Softmax over time



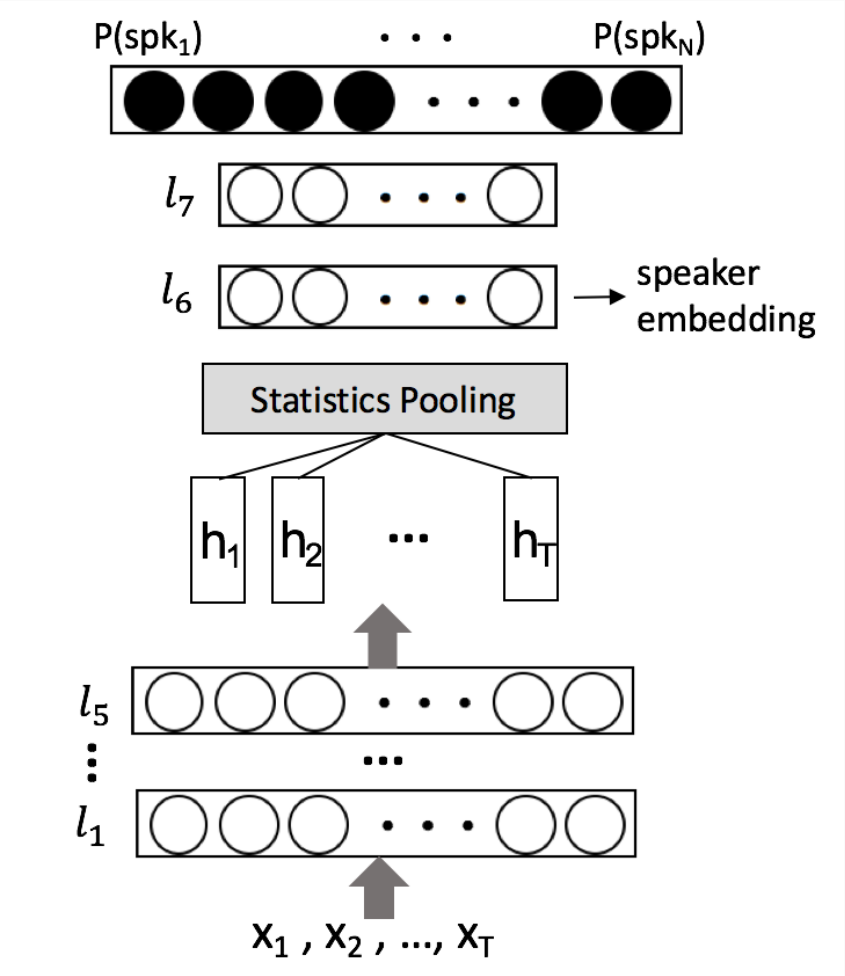
[20] Z. Lin, et.al., "A structured self-attentive sentence embedding", ICLR, 2017.

Self Attentive Speaker Embedding

Mean and Std are computed in the pooling layer – SRE 16 evaluation [21]

- Pooling layer has attention weighting
- Multiple weighting schemes are computed

system	cant	taglog	pool
ivec	8.3	17.6	13.6
xvec	5.4	15.2	11.2
att-1	5.2	14.5	10.7
att-5	4.6	14.2	10.2

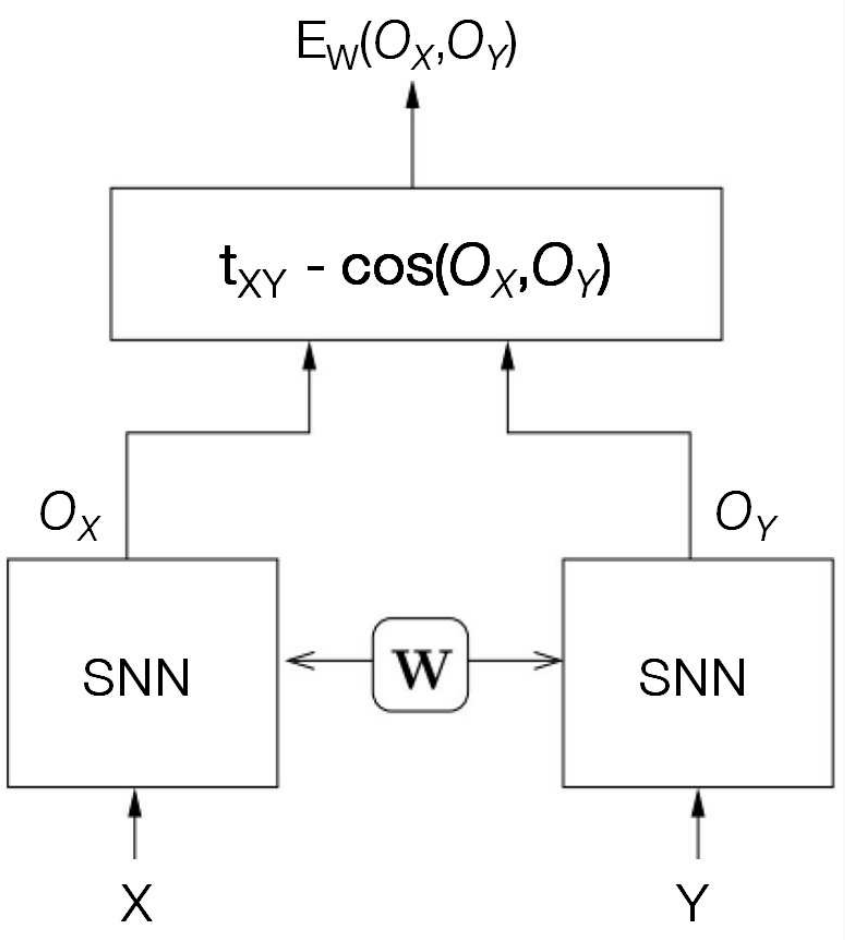


[21] Zhu, Yingke, et al. "Self-attentive speaker embeddings for text-independent speaker verification." Proc. Interspeech. 2018

End to End Systems

End to End (E2E) systems

- NNs are not flexible with enrolling new users
- Learn an NN to output a similarity score
- Shared weights on two branches- Feature embedding output
- Merged part perform Similarity computation and prediction



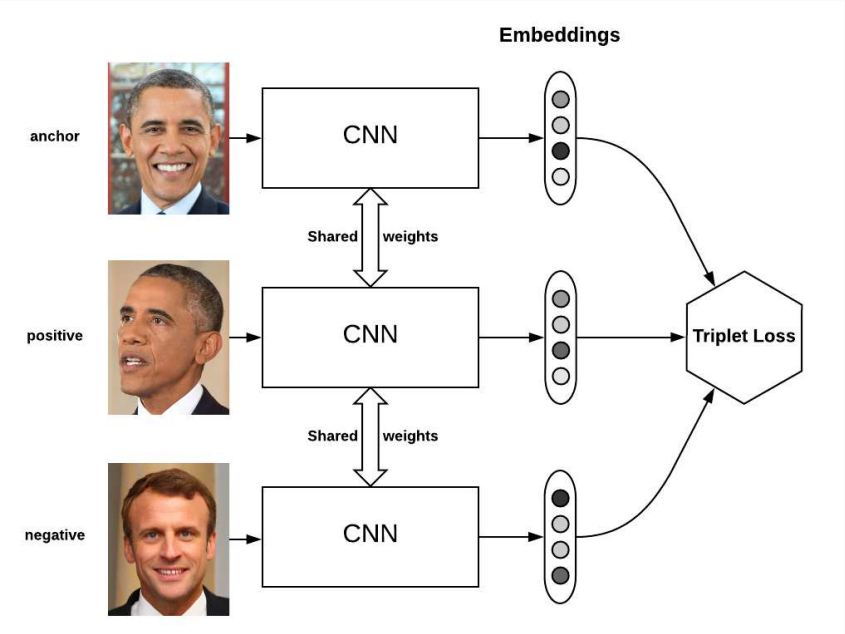
[22]

[22] Taigman, Yaniv et.al., "Deepface: Closing the gap to human level performance in face verification", CVPR 2014

Objective Function

- Objective function from triplets (Anchor, Positive, Negative) – $(a, p, n)_i$
- Difference class similarity $<$ same class similarity
- $S_i^{a,n} + \alpha < S_i^{a,p}$
- Cost Function

$$\mathcal{L} = \sum_i [S_i^{a,n} + \alpha - S_i^{a,p}]_+$$

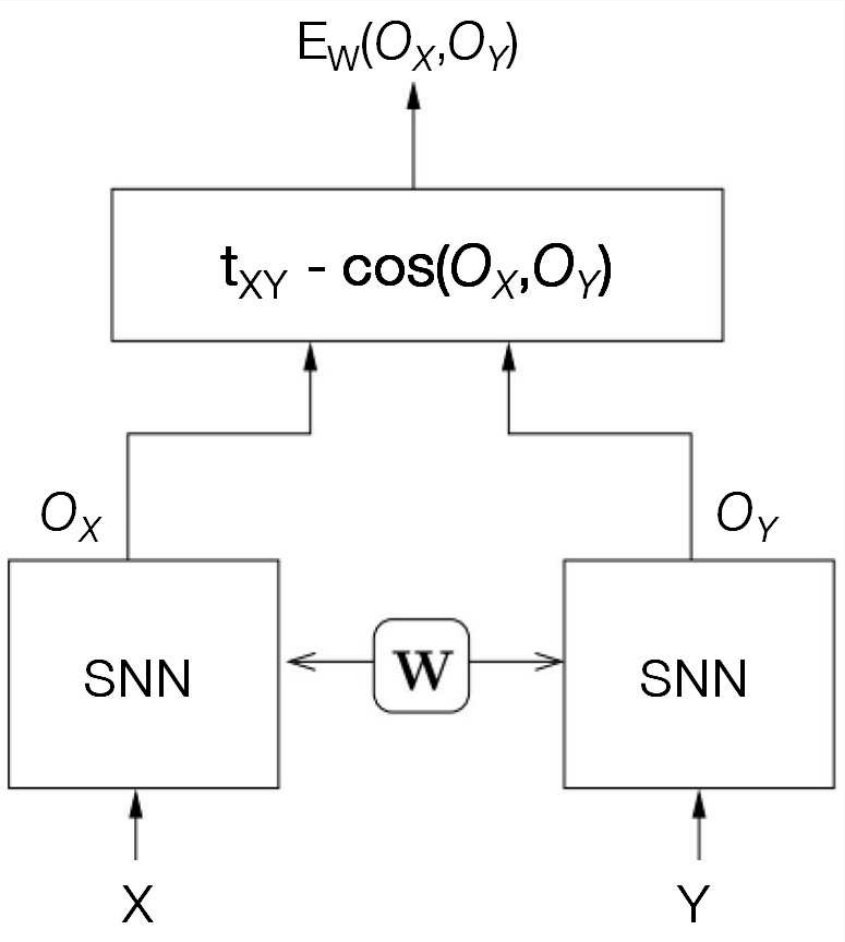


Training

- One possible cost Function

$$\mathcal{L} = \sum_i [S_i^{a,n} + \alpha - S_i^{a,p}]_+$$

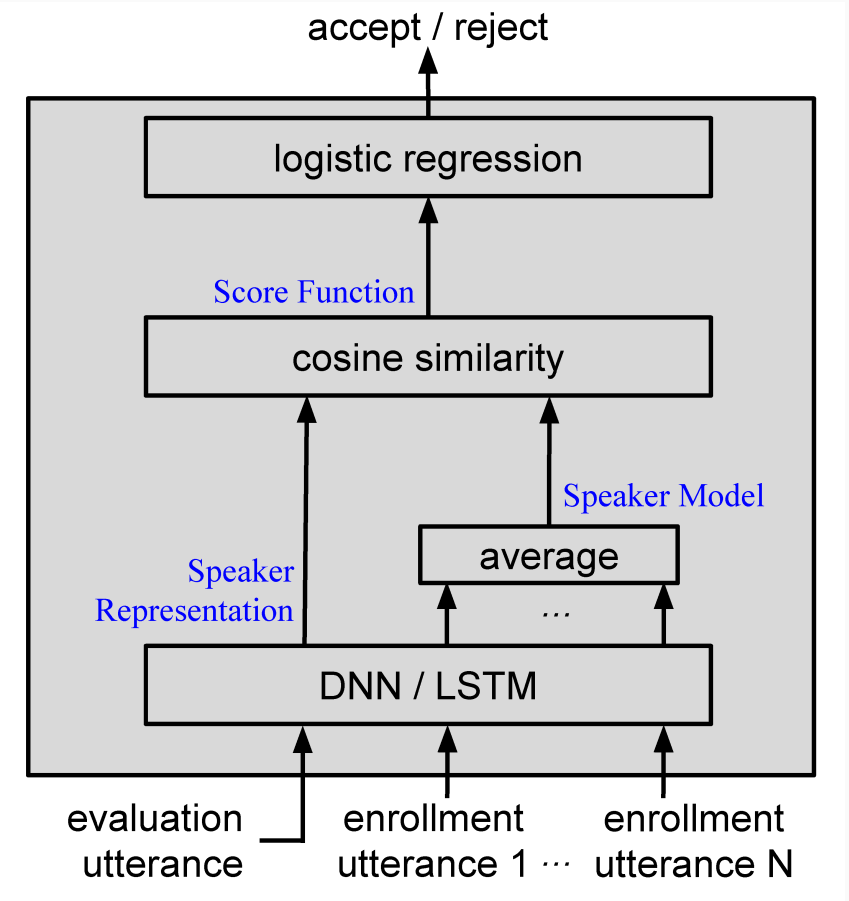
- Choose the triplets carefully
- Start with semi-hard examples
- Move to harder examples later
- Softmax pre training to initialize



E2E Text Dependent SV

Earlier attempt to "Okay Google" speaker verification [23]

- LSTMs/DNNs to extract the speaker embedding
- Embedding derived from N different training utterances averaged
- Cosine similarity followed by a logistic regression



[23] Heigold, George et. al., "End to end text-dependent speaker verification ", ICASSP 2016

E2E Text Dependent SV

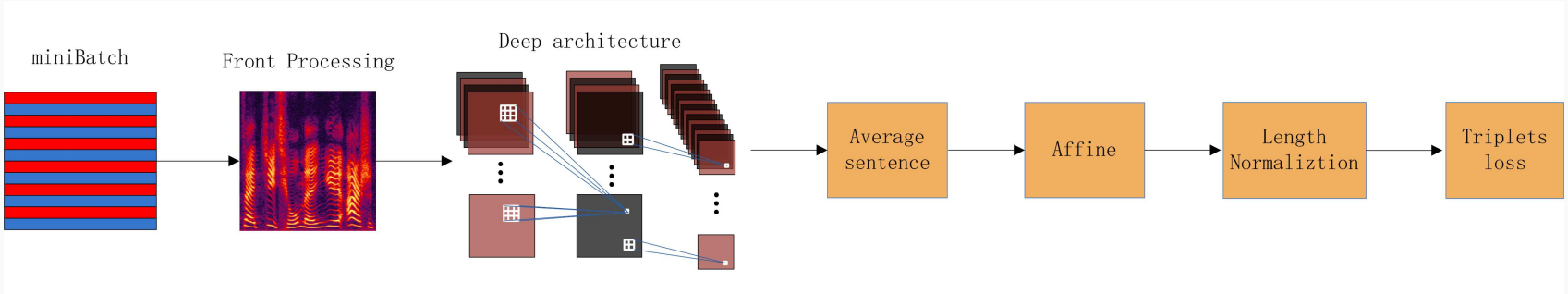
- Predicts $p(Pos) = \frac{1}{1+exp[-(wS(X,spk)+b)]}$
- Optimize binary cross entropy [No triplet loss]
- Last output of the LSTM is taken
- DNN framewise outputs are averaged

system	EER
ivector+PLDA	4.89
dvector	3.32
DNN embed	1.87
LSTM embed	1.36

Deep Speaker

Baidu's TI E2E SV system [24]

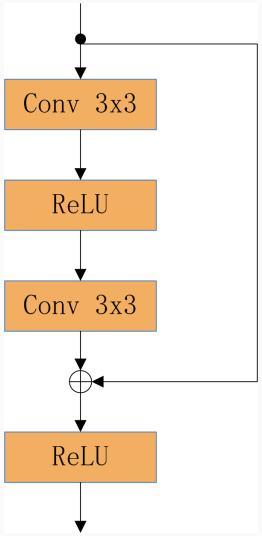
- CNN/GRUs followed by an affine and length normalization layers to extract speaker embedding
- Cosine similarity metric trained with triplet loss



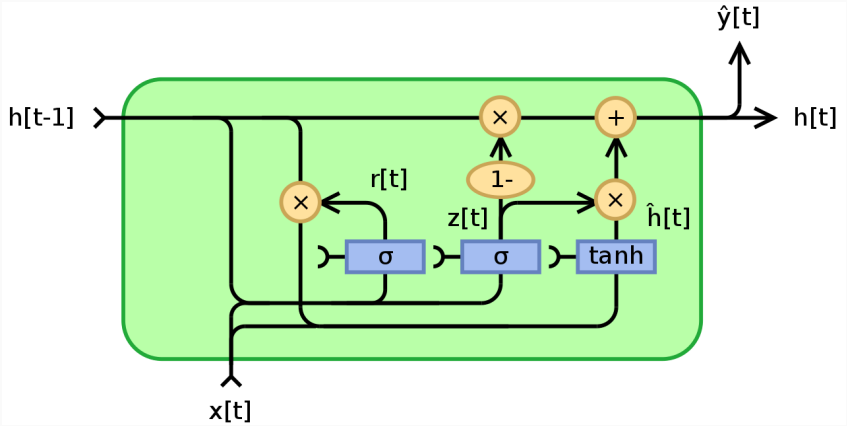
[24] Li, Chao, et al. "Deep speaker: an end-to-end neural speaker embedding system." arXiv preprint arXiv:1705.02304 (2017).

Deep Speaker

- [Conv2d + 3 Res blk] × 4
- 64,128,256,512 filters



- Conv64 followed by 3 GRU layers 1024 cells



- Statistics pooling (mean only)
- Normalize to unit length and cosine similarity
- Softmax initialization

Deep Spkr Evaluation

UID dataset (DNN training - 250k Speakers, Enrollment - 50K speakers, Evaluation - 200)

system	EER
BNF	13.7
ResNet 50k	2.23
GRU 50k	2.77
ResNet 250k	1.83
GRU 250k	2.35

Permutations Combinations?

- Attention Layers in E2E systems
- Other Loss functions - eg: Center Loss
- Better E2E systems for LID, DID
- New Network architectures ??

Summary

- Sophisticated Speaker Embeddings
- Eliminated Need for speech labels
- Working towards Language, channel independence

THANK YOU

QUESTIONS?