Feedforward Sequential Memory Networks and its Applications

张仕良 (谵良)

阿里巴巴达摩院-机器智能技术实验室

Outline

Background

Feedforward Sequential Memory Networks (FSMN)

- Evolution of FSMM: sFSMN->vFSMN->cFSMN->DFSMN->LFR-DFSMN->DFSMN-CTC
- FSMN for Acoustic Modeling

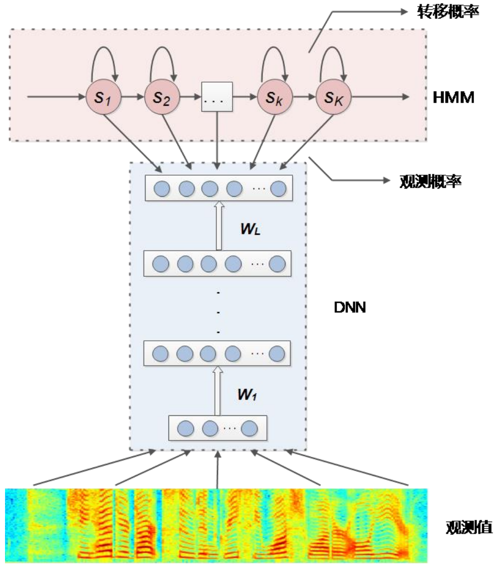
Promotion application of FSMN

- Language Modeling； Keyword Spotting (KWS)； TTS;
- Open source： FSMN in Kaldi-Nnet1

Background

◆ Acoustic Model ： GMM-HMM =>NN-HMM

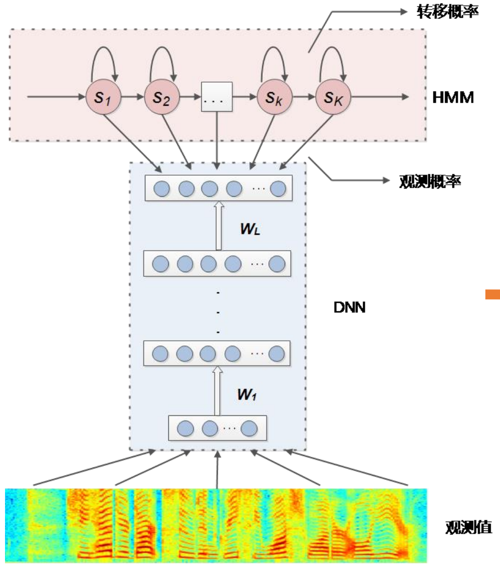
- Feedforward Fully-connected Deep Neural Networks (DNN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Network (RNN)
- Long Short-Term Memory (LSTM) - > BLSTM



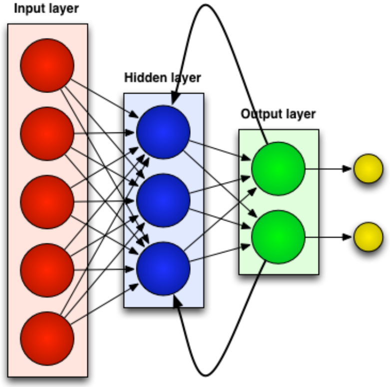
Background

◆ Acoustic Model ： GMM-HMM =>NN-HMM

- Feedforward Fully-connected Deep Neural Networks (DNN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Network (RNN)
- Long Short-Term Memory (LSTM) - > BLSTM



Long-term dependency

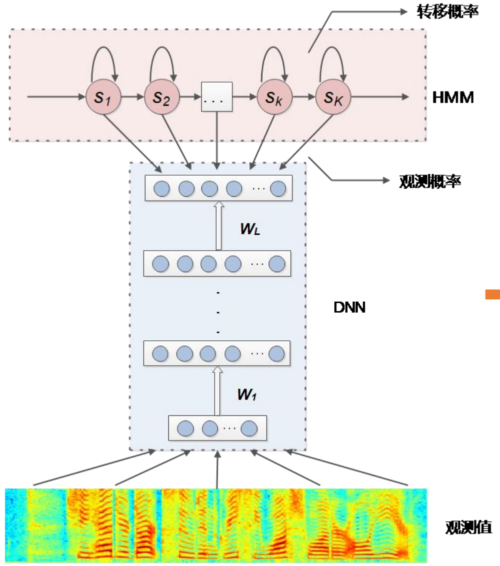


Simple RNN

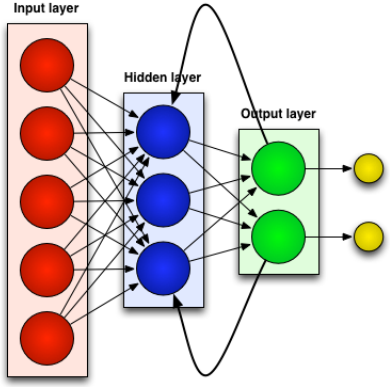
Background

◆ Acoustic Model ： GMM-HMM =>NN-HMM

- Feedforward Fully-connected Deep Neural Networks (DNN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Network (RNN)
- Long Short-Term Memory (LSTM) - > BLSTM

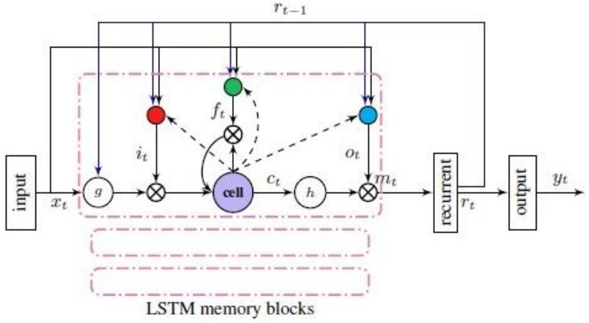


Long-term dependency



Simple RNN

Gradient vanishing/exploding

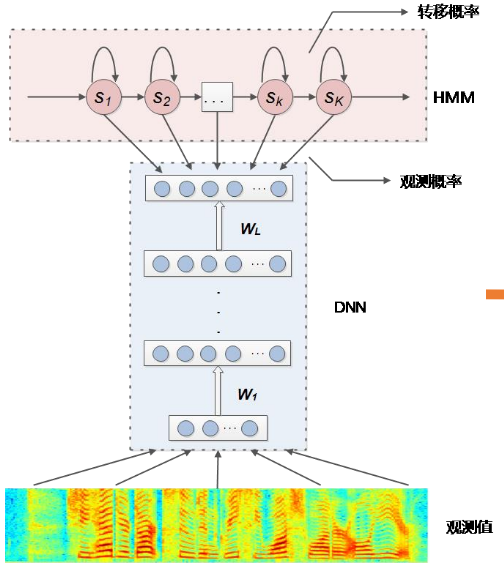


LSTM

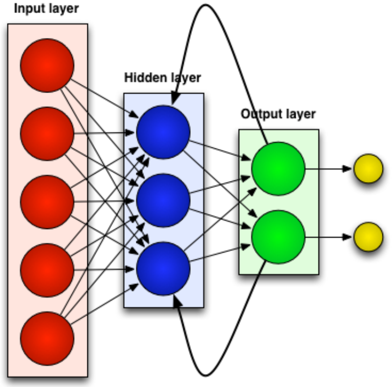
Background

◆ Acoustic Model ： GMM-HMM =>NN-HMM

- Feedforward Fully-connected Deep Neural Networks (DNN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Network (RNN)
- Long Short-Term Memory (LSTM) - > BLSTM

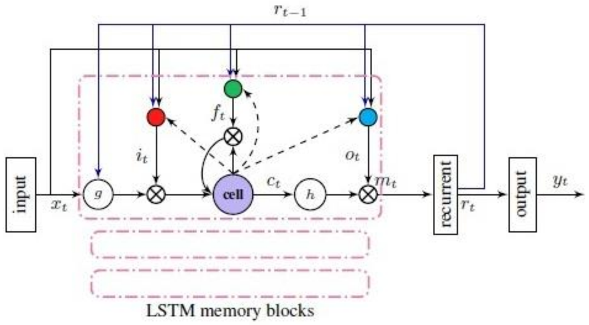


Long-term dependency



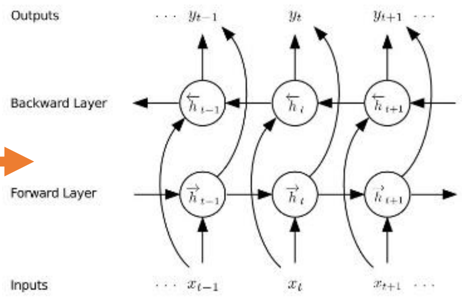
Simple RNN

Gradient vanishing/exploding



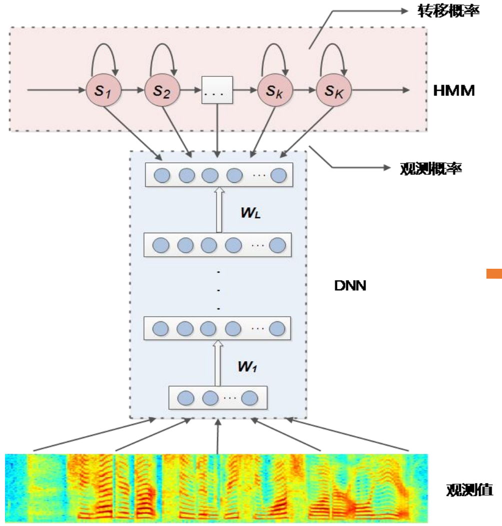
LSTM

Bidirectional context

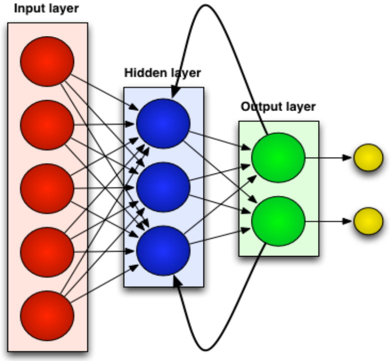


BLSTM

Background

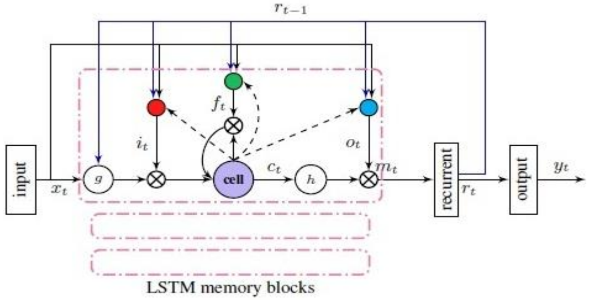


Long-term dependency



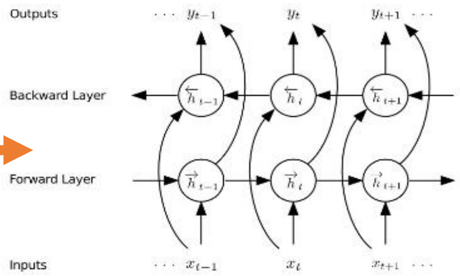
Simple RNN

Gradient vanishing/exploding



LSTM

Bidirectional context



BLSTM

- **From DNN to BLSTM**
  - Huge performance improvement: ＞20%
  - Suffer from the computation and latency problem
- **Non-recurrent architecture to model long-term dependency**
  - Unfolded RNN [G. Saon et al., 2014]
  - Time Delay Neural Networks (TDNN) [A. Waibel, 1989; D. Povey et al., 2015]
  - Feedforward Sequential Memory Networks (FSMN) [S. L. Zhang et al., 2015, 2016]

Feedforward Sequential Memory Networks

FSMN
(sFSMN/vFSMN)
2015
USTC, iflyTek

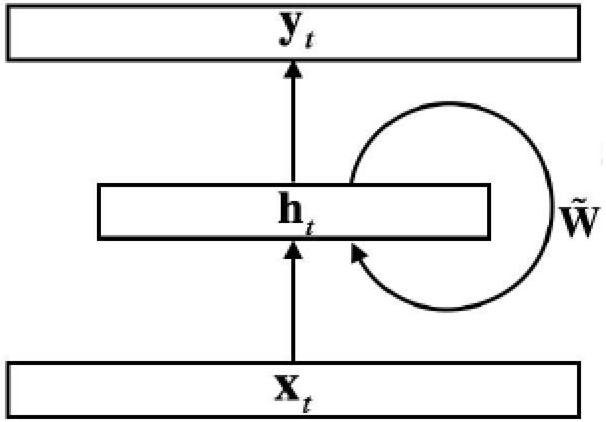
Compact FSMN
(cFSMN)
2016
USTC, iflyTek

DFSMN/
LFR-DFSMN
2017
USTC, Alibaba

DFSMN-CTC
2017-2018
Alibaba

Feedforward Sequential Memory Networks

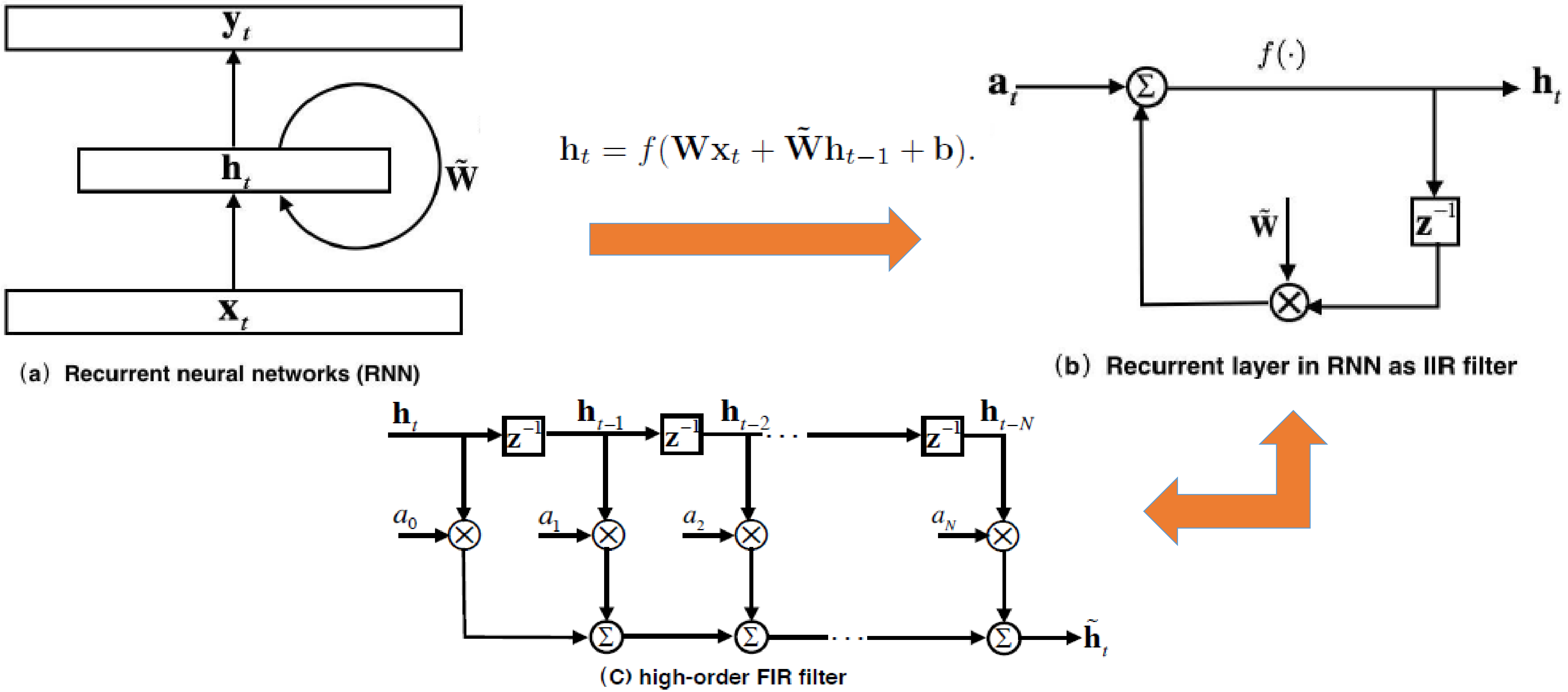
Motivation of FSMN



(a) Recurrent neural networks (RNN)

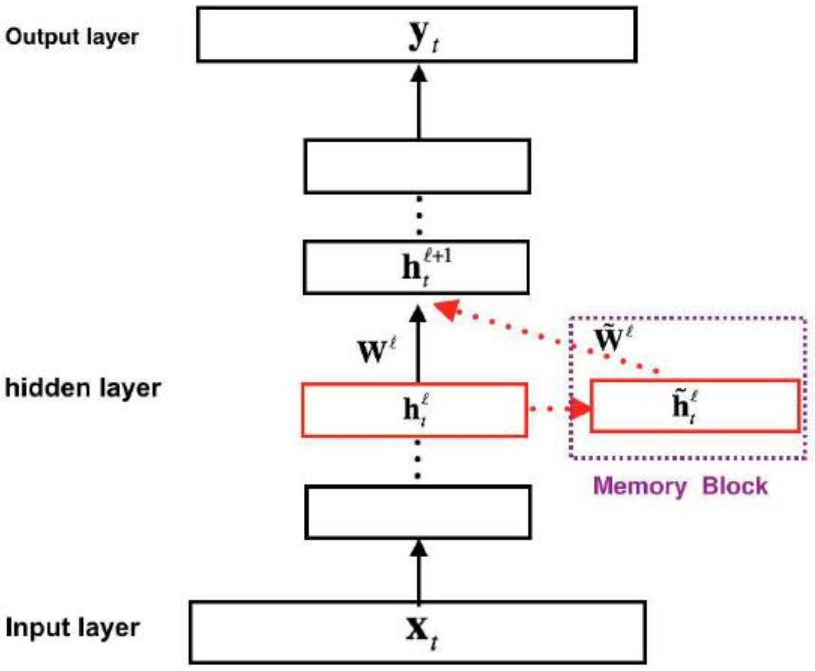
Feedforward Sequential Memory Networks

Motivation of FSMN

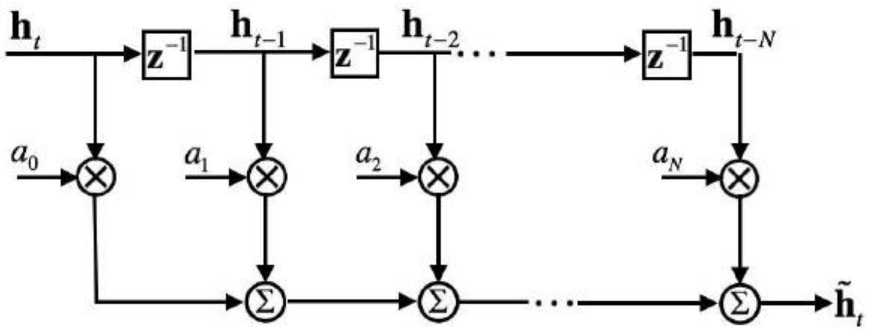


Any infinite impulse response (IIR) filter can be well approximated using a high-order finite impulse response (FIR) filter

Feedforward Sequential Memory Networks



(a) Feedforward sequential memory neural network (FSMN)



(b) Memory block in unidirectional FSMN as FIR filter

I. scalar FSMN (sFSMN):

$$\tilde{\mathbf{h}}_t^\ell = \sum_{i=0}^{N} a_i^\ell \cdot \mathbf{h}_{t-i}^\ell$$

$$\tilde{\mathbf{h}}_t^\ell = \sum_{i=0}^{N_1} a_i^\ell \cdot \mathbf{h}_{t-i}^\ell + \sum_{j=1}^{N_2} {c_j}^\ell \cdot \mathbf{h}_{t+j}^\ell$$

II. vectorized FSMN (vFSMN):

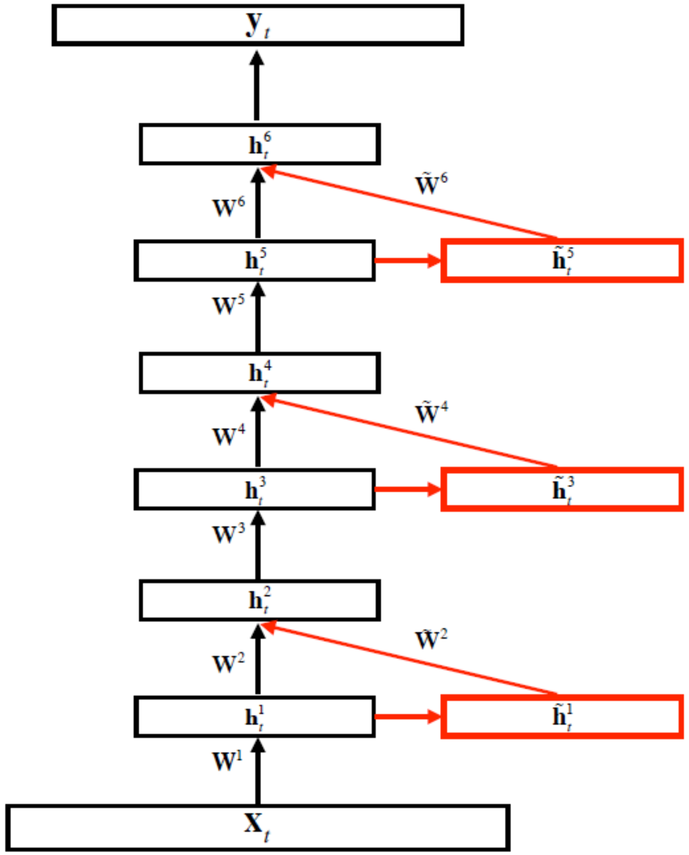
$$\tilde{\mathbf{h}}_t^\ell = \sum_{i=0}^{N} \mathbf{a}_i^\ell \odot \mathbf{h}_{t-i}^\ell$$

$$\tilde{\mathbf{h}}_t^\ell = \sum_{i=0}^{N_1} \mathbf{a}_i^\ell \odot \mathbf{h}_{t-i}^\ell + \sum_{j=1}^{N_2} \mathbf{c}_j^\ell \odot \mathbf{h}_{t+j}^\ell$$

Feedforward Sequential Memory Networks

From vFSMN to cFSMN

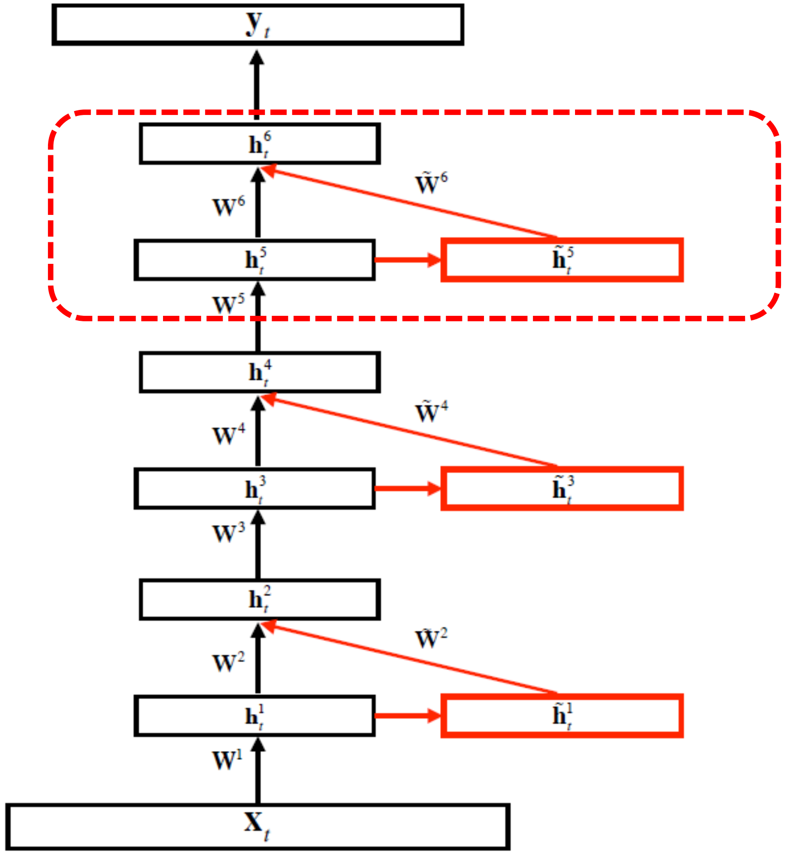
- vFSMN with multiple memory block: additional parameters & computation cost

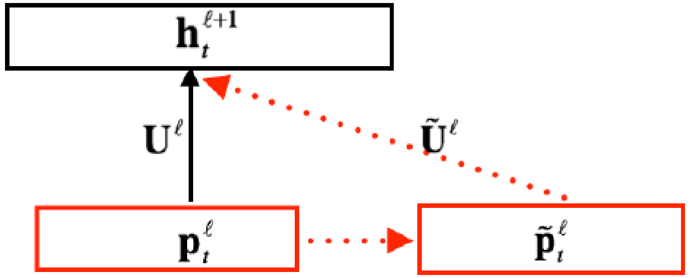


Feedforward Sequential Memory Networks

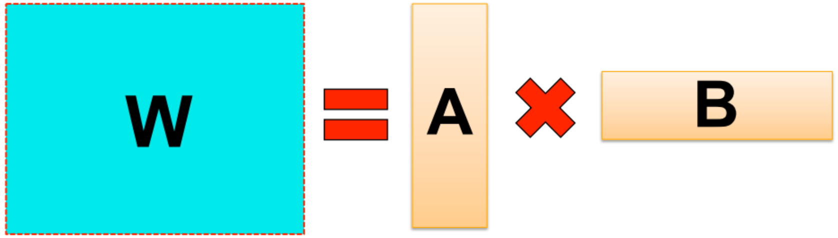
From vFSMN to cFSMN

- vFSMN with multiple memory block: additional parameters & computation cost



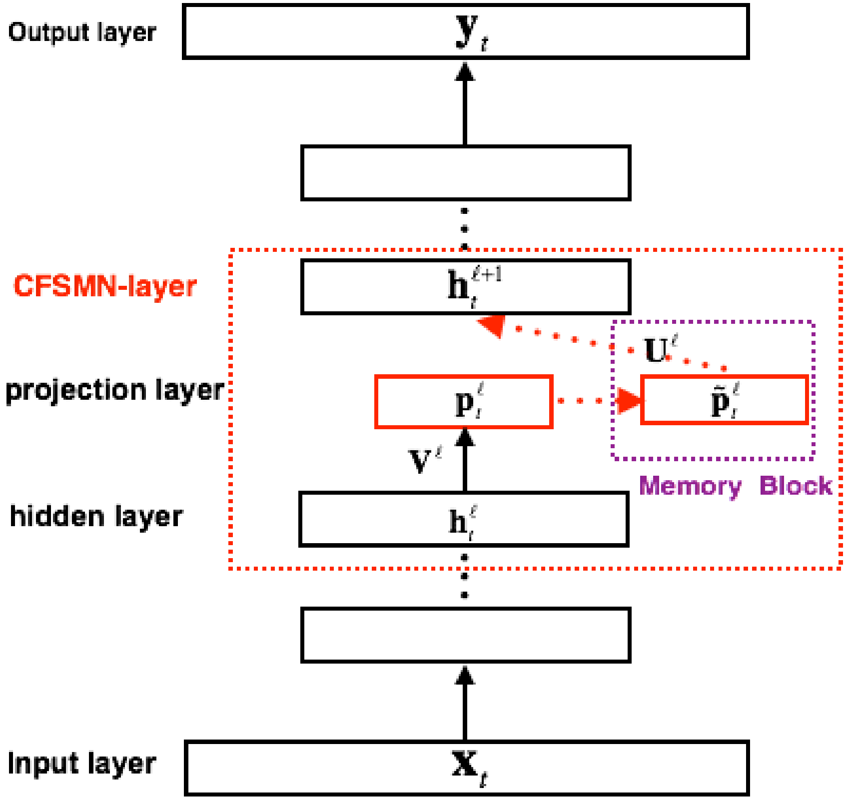


- Introduce a lot of additional parameters compared to FNN with the same architecture
- Reduce the hidden size->reduce the memory block size
- **cFSMN: FSMN + Low Rank Matrix Factorization**



Feedforward Sequential Memory Networks

Compact FSMN (cFSMN)



Unidirectional cFSMN :

$$\tilde{\mathbf{p}}_t^\ell = \boxed{\mathbf{p}_t^\ell} + \sum_{i=0}^{N} \mathbf{a}_i^\ell \odot \mathbf{p}_{t-i}^\ell$$

Bidirectional cFSMN :

$$\tilde{\mathbf{p}}_t^\ell = \boxed{\mathbf{p}_t^\ell} + \sum_{i=0}^{N_1} \mathbf{a}_i^\ell \odot \mathbf{p}_{t-i}^\ell + \sum_{j=1}^{N_2} \mathbf{c}_j^\ell \odot \mathbf{p}_{t+j}^\ell$$

Hidden output:

$$\mathbf{h}_t^{\ell+1} = f(\mathbf{U}^\ell \tilde{\mathbf{p}}_t^\ell + \mathbf{b}^{\ell+1})$$

Compared to the total parameters, the additional parameters introduced by the memory blocks in cFSMN can be ignored!

Feedforward Sequential Memory Networks

Experimental Results – 300 hours English SWB Task

- Feature: 123-dimensional FBK ; Label: GMM-HMM alignment
- Objective function : cross entropy (CE); LM: tri-gram
- Networks:
  - Sigmoid/ReLU-DNN: 123* 11 – 6 * 2048 – 8991
  - LSTM: 123 – 3* [2048-P512] – 8991
  - BLSTM: 123 – 3* [2*{1024-P512}] – 8991
  - sFSMN/vFSMN: 123*3 – 2048(M) – 2048 -2048(M) – 2048 – 2048(M) - 2048 – 8991
  - cFSMN: 123*3 – 4* [2048-P512(M)]-2048-2048 – 512 - 8991

model	model size (MB)	time (hr)	WER (in %)
Sigmoid-DNN	160	5.0	15.6
ReLU-DNN	160	4.8	14.6
LSTM	110	9.4	14.2
Kaldi-LSTM	110	10.6	14.4
BLSTM	180	22.6	13.5
sFSMN	202	6.7	14.2
vFSMN	**203**	**6.9**	**13.2**
cFSMN	**73**	**3.1**	**12.8**

Feedforward Sequential Memory Networks

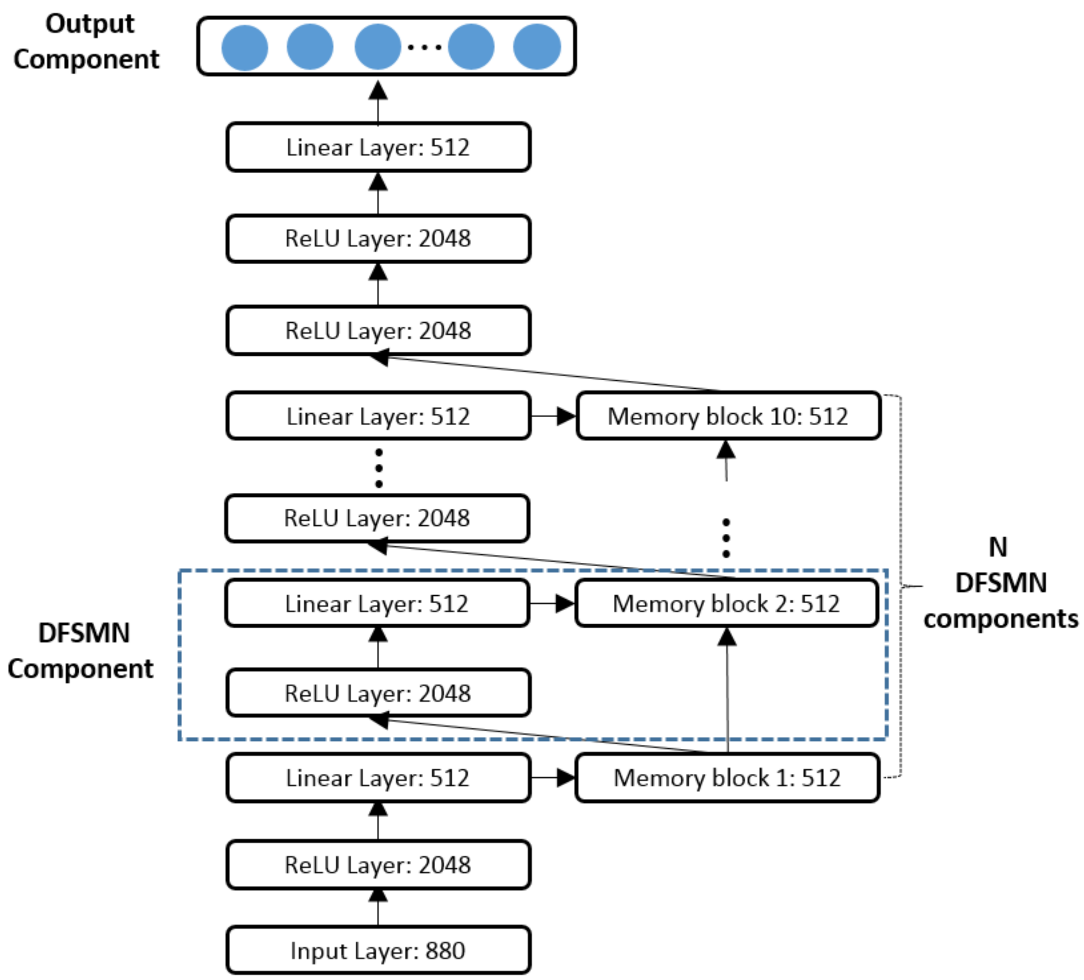
For cFSMN to DFSMN

- Industrial application: 300 hours Vs. 20000 hours
- Advanced FSMN: Deep-FSMN
  - Deep: dozens of DFSMN components
  - Skip connections
  - Stride factor
- **DFSMN components: ReLU layer, linear layer, memory block**

$$h_t^l = max(\mathbf{W}^l h_t^{l-1} + b^l, 0);\ \ p_t^l = \mathbf{V}^l h_t^l + v^l$$

$$m_t^l = \mathcal{H}(m_t^{l-1}) + p_t^l + \sum_{i=0}^{N_1^l} a_i^l \odot p_{t-s_1 \cdot i}^l + \sum_{j=1}^{N_2^l} c_j^l \odot p_{t+s_2 \cdot j}^l$$

i.e. $\mathcal{H}(m_t^{l-1}) = m_t^{l-1}$



Feedforward Sequential Memory Networks

Experimental Results – 2000 hours English Fisher Task

- Feature: 72-dimensional FBK
- Label: GMM-HMM alignment
- Objective function : cross entropy (CE)
- LM: tri-gram

ID	model architecture	stride	WER (%)
exp1	216-6x[2048-512(20,20)]-3x2048-512-9004	1	10.7
exp2	216-6x[2048-512(20,20)]-3x2048-512-9004	2	10.3
exp3	216-8x[2048-512(20,20)]-3x2048-512-9004	2	9.6
exp4	216-10x[2048-512(20,20)]-3x2048-512-9004	2	9.5
exp5	216-10x[2048-512(10,10)]-3x2048-512-9004	2	9.7
exp6	**216-12x[2048-512(20,20)]-3x2048-512-9004**	**2**	**9.4**

Feedforward Sequential Memory Networks

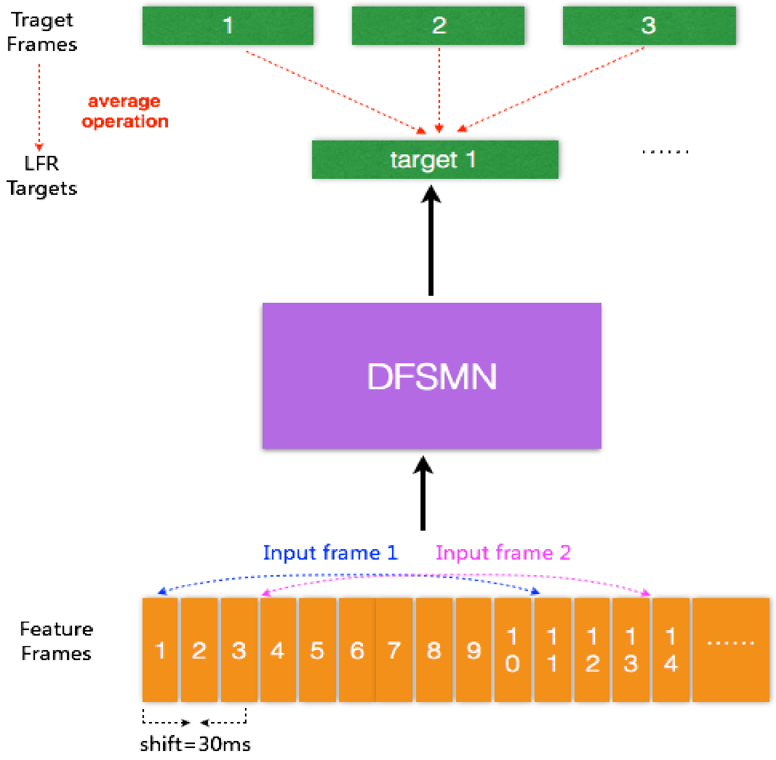
Experimental Results – 2000 hours English Fisher Task

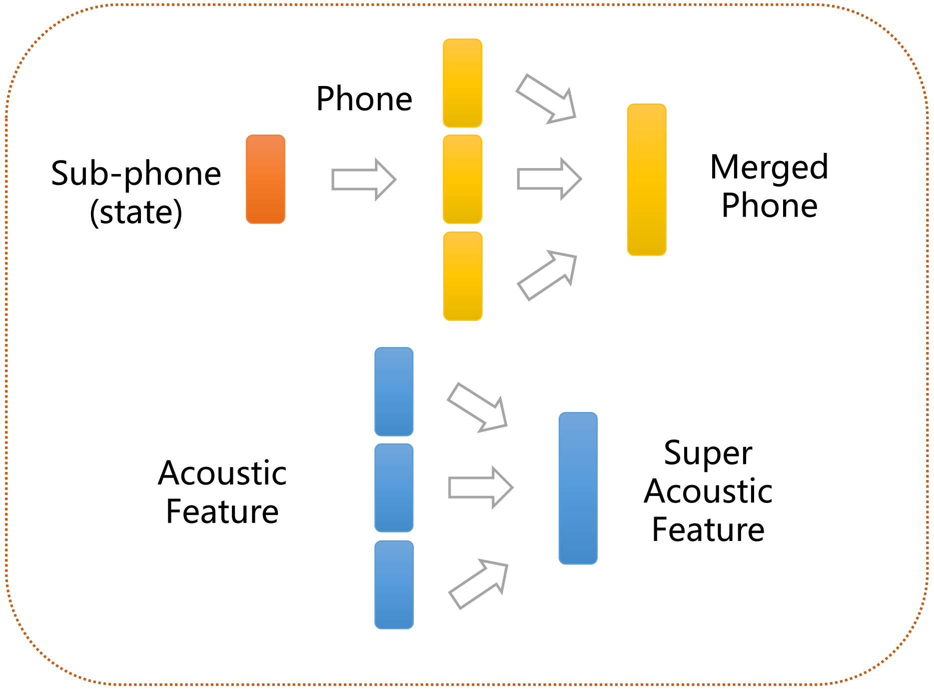
- Feature: 72-dimensional FBK
- Label: GMM-HMM alignment
- Objective function : cross entropy (CE)
- LM: tri-gram

Model	Model Size (MB)	WER%
DNN	159	14.3
BLSTM-ours	**180**	**10.9**
BLSTM-微软	**166***	**10.3**
FSMN	104	10.8
DFSMN(12)	**152**	**9.4**

DFSMN with Lower Frame Rate

- Lower Frame Rate [Tara et al., 2016]
- LFR-DFSMN: efficient training and decoding





Feedforward Sequential Memory Networks

Experimental Results – 5000 hours Mandarin Task

- **CD-state**: 14359; **CD-Phone**: 9841
- **Lower Frame Rate** [Tara et al. 2016]: 30ms-frame-shift
- **LCBLSTM**: 80-3*[500,500]-2*2048-14359, N_c=80, N_r=40
- **LFR-LCBLSTM**: 80*17- 3*[500,500]-2*2048-9841, N_c=27, N_r=13
- **LFR-DFSMN**: 80*11 – N* [2048-P512(M)]-2048-2048 – 512 - 8991

Model	Target	Size (MB)	CER %	Gain
LCBLSTM	CD-State	196	18.78	-
cFSMN(6)		102	17.72	+5.32%
LFR-LCBLSTM	CD-Phone	**220**	**18.92**	-
LFR-cFSMN(6)		108	16.85	+11.00%
LFR-cFSMN(8)	CD-Phone	124	15.80	+16.50%
LFR-cFSMN(10)		140	15.91	+15.86%
LFR-DFSMN(8)		124	15.45	+18.34%
LFR-DFSMN(10)	CD-Phone	**140**	**15.00**	**+20.72%**

Feedforward Sequential Memory Networks

Experimental Results – 5000 hours Mandarin Task

- **Lower Frame Rate** [Tara et al. 2016]: 30ms-frame-shift
- **LFR-LCBLSTM**: 80*17- 3*[500,500]-2*2048-9841, N_c=27, N_r=13
- **LFR-DFSMN**: 80*11 – 10*[2048-P512]-2*2048-P512-9841

Model	Traing Time （hr/epoch）	Model Size (MB)	WER%	RTF
LFR-LCBLSTM	21.62	220	18.92	0.4289
LFR-DFSMN	6.85	140	15.00	0.1486
Gain	**x 3.15**	**-36%**	**+ 20.72**	**x 2.88**

Feedforward Sequential Memory Networks

Experimental Results – 20000 hours Mandarin Task

- Toward lower latency

Model	N_2	Delay Frame	CER%	Gain
LFR-LCBLSTM	-	40	16.05	-
LFR-DFSMN(10)	2	20	12.67	+21.06%
	1	10	12.94	+19.38%
	1 and 0	5	13.38	+16.64%

"1 and 0" denotes the lookahead order of the odd layer and even layer is 1 and 0 respectively.

DFSMN-CTC & Joint CTC-CE Learning

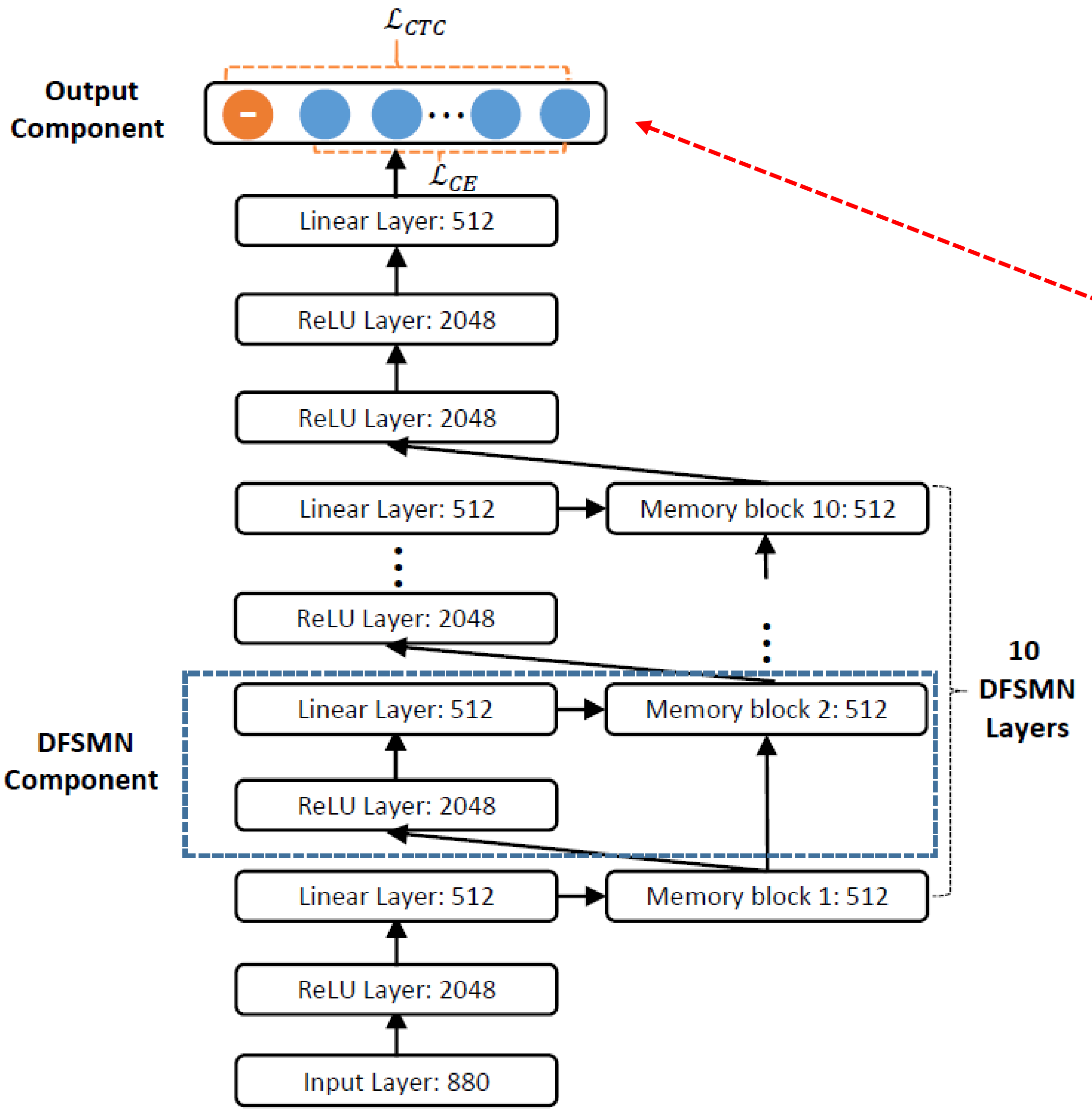
Acoustic Modeling with DFSMN-CTC and Joint CTC-CE Learning

Shiliang Zhang, Ming Lei

Machine Intelligence Technology, Alibaba Group
{sly.zsl, lm86501}@alibaba-inc.com

- **Acoustic modeling with Connectionist Temporal Classification (CTC)**
  - Advantage: better performance; Faster decoding speed
  - Drawback:
    a. LSTM-type networks: BLSTM-CTC is not suitable for online ASR
    b. Latency problem: output target can be arbitrarily delayed after its corresponding input event
    c. Stability problem: sometime training will fail to converge
- **In this work**
  - DFSMN-CTC: explore how this type of non-recurrent models behave when trained with CTC loss
  - A novel joint CTC-CE learning method to handle the “latency problem” & “stability problem”

DFSMN-CTC & Joint CTC-CE Learning



- **A novel Joint CTC-CE Learning method**

$$\mathcal{L}_{ctcce}(\mathbf{x}) = \mathcal{L}_{ctc}(\mathbf{x}) + \alpha \cdot \mathcal{L}_{ce}(\mathbf{x})$$

Pre-set constant

$$\mathcal{L}_{ce}(\mathbf{x}) = -\sum_{k=1}^{T} \boxed{(1 - p(y_1|\boldsymbol{x}_k))} \sum_{i=2}^{K} t_i \log p(y_i|\boldsymbol{x}_k)$$

Context-dependent regularization term

During training, the CTC loss tend to generate the shape spike distribution that only a few spikes for each output target while predicting blank label with high probability the rest of time. Thereby, the regularized CE loss will help to produce the accurate alignment for the output target while won't effect the distribution of blank label

DFSMN-CTC & Joint CTC-CE Learning

Mandarin Speech Recognition: EXP1.CTC Vs. CE

- **Training set**: 1000-hours (1k), 4000-hours (4k), 20000-hours(20k)
- **Test set**: 1) normal test set : 30 hours; 2) fast speed test set: 1 hour
- **Input**: FBK (80) * context window (5-1-5)
- **Lower frame rate**: subsample the input frames with 3.

Method	Label	Data (Hours)	Test set (WER %)	
			Normal	Fast
BLSTM-CE	CD-Phone	1k	19.77	47.56
		4k	16.53	37.17
		20k	13.97	31.71
DFSMN-CE	CD-Phone	1k	18.19	44.25
		4k	14.24	33.92
		20k	12.10	29.79
DFSMN-CTC	CI-Phone	1k	17.82	43.22
		4k	13.82	32.15
		20k	11.46	26.84
DFSMN-CTC	CD-Phone	1k	16.95	40.27
		4k	13.13	26.70
		20k	11.71	24.04

DFSMN-CTC & Joint CTC-CE Learning

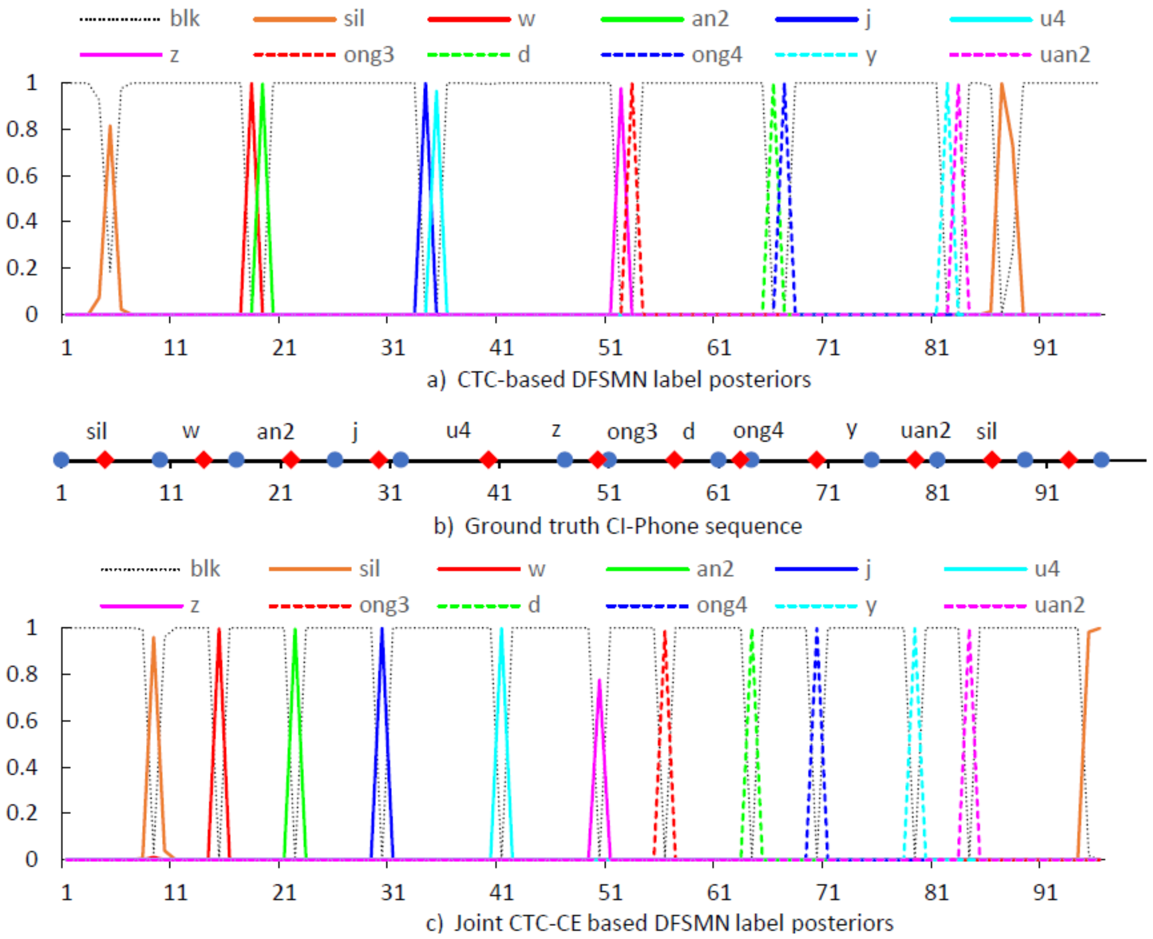
Mandarin Speech Recognition: EXP2. Joint CTC-CE

- **Training set**: 20000-hours(20k)
- **Test set**: 1) normal test set : 30 hours; 2) fast speed test set: 1 hour
- **Input**: FBK (80) * context window (5-1-5)
- **Lower frame rate**: subsample the input frames with 3.

Method	Alpha	Test set (WER %)			
		Normal	Gain	Fast	Gain
CE	-	12.10	-	29.79	-
CTC	-	11.71	3.2%	24.04	19.3%
Joint CTC CE	0.1	10.92	9.8%	21.68	27.2%
	0.5	10.67	11.8%	21.98	26.2%
	1.0	10.77	11.0%	20.80	30.1%
	2.0	11.03	8.8%	22.86	23.3%

DFSMN-CTC & Joint CTC-CE Learning

Mandarin Speech Recognition: EXP2. Joint CTC-CE



a) The label posteriors distribution estimated by CTC is inconsistent with the ground-truth

b) For the proposed joint CTC-CE trained DFSMN, the constrained CE loss helps to estimate the accurate alignment.

Feedforward Sequential Memory Networks

Promotion application of FSMN

- FSMN for Language Modeling: S Zhang, C Liu, H Jiang, S Wei, L Dai, Y Hu, Feedforward sequential memory networks: A new structure to learn long-term dependency, arXiv preprint arXiv:1512.08301, 2015.
- FSMN for TTS: Mengxiao Bi, Heng Lu, Shiliang Zhang, Ming Lei, Zhijie Yan， *DEEP FEED-FORWARD SEQUENTIAL MEMORY NETWORKS FOR SPEECH SYNTHESIS*， ICASSP2018
- FSMN for KWS: Mengzhe Chen, Shiliang Zhang, Ming Lei, Yong Liu, Haitao Yao, Jie Gao, *Compact Feedforward Sequential Memory Networks for Small-footprint Keyword Spotting*, accepted by INTERSPEECH 2018
- DFSMN-CTC: Shiliang Zhang, Ming Lei, *Acoustic Modeling with DFSMN-CTC and Joint CTC-CE Learning*, accepted by INTERSPEECH 2018

FSMN for Language Modeling

Language Modeling

- Prediction: $p\,(w^t|w^1 \cdots w^{t-1})$
- Unidirectional FSMN: without the lookahead filters

Corpus	Train	Valid	Test	Vocabulary
PTB	930k	74k	82k	10k
LTCB	153M	8.9M	8.9M	80k

Table 4. Perplexities on the PTB database for various LMs.

Model	Test PPL
KN 5-gram (Mikolov et al., 2011)	141
3-gram FNN-LM (Zhang et al., 2015d)	131
RNN-LM (Mikolov et al., 2011)	123
LSTM-LM (Graves, 2013)	117
MemN2N-LM (Sukhbaatar et al., 2015)	111
FOFE-LM (Zhang et al., 2015d)	108
Deep RNN (Pascanu et al., 2013)	107.5
Sum-Prod Net (Cheng et al., 2014)	100
LSTM-LM (1-layer)	114
LSTM-LM (2-layer)	105
sFSMN-LM	**102**
vFSMN-LM	**101**

Table 5. Perplexities on the English wiki9 test set for various language models (M denotes a hidden layer with memory block).

Model	Architecture	PPL
KN 3-gram	-	156
KN 5-gram	-	132
FNN-LM	[2*200]-3*600-80k	155
RNN-LM	[1*600]-80k	112
FOFE-LM	[2*200]-3*600-80k	104
sFSMN-LM	[2*200]-600(M)-600-600-80k	95
	[2*200]-600-600(M)-600-80k	96
	[2*200]-600(M)-600(M)-600-80k	**92**
vFSMN-LM	[2*200]-600(M)-600-600-80k	95
	[2*200]-600(M)-600(M)-600-80k	**90**

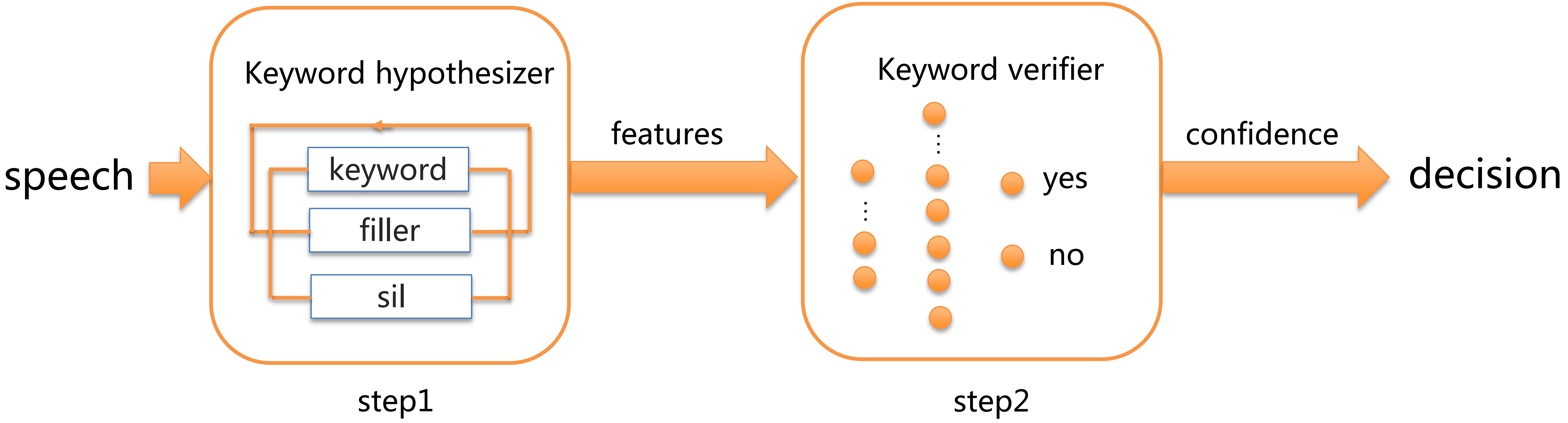
FSMN for Small-Footprint Keyword Spotting

KWS

- IoT device:
- Model：performance，model size，latency

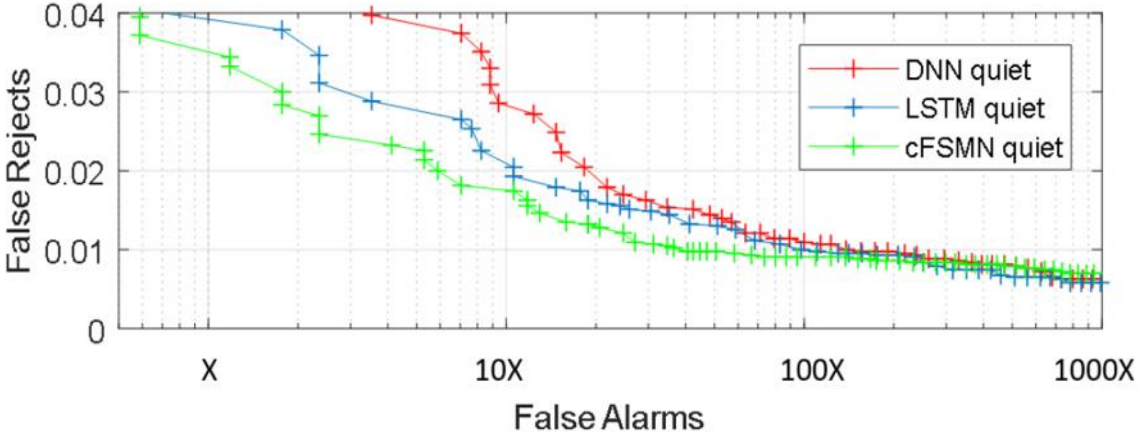
你好斑马

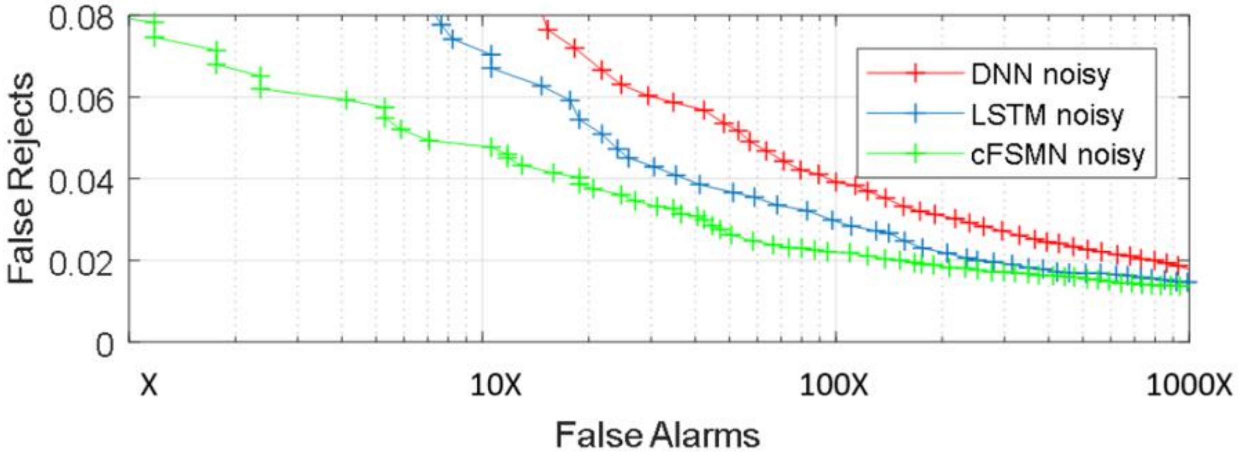
天猫精灵



FSMN for Small-Footprint Keyword Spotting

Experiments



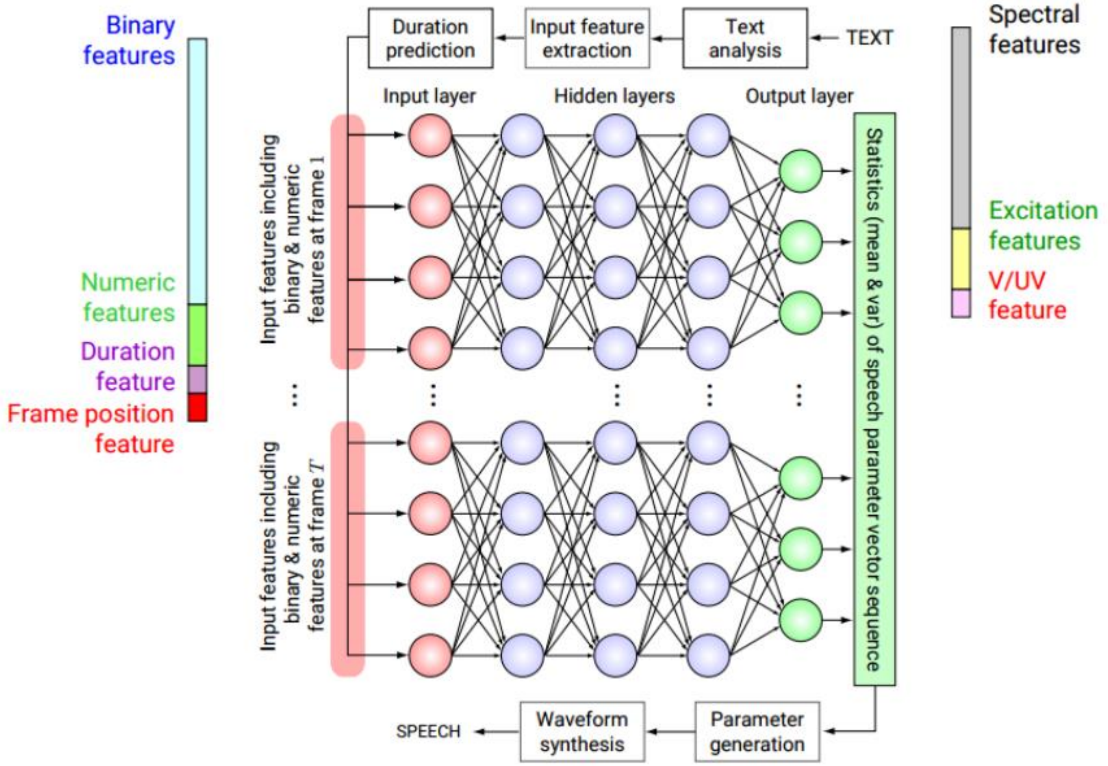


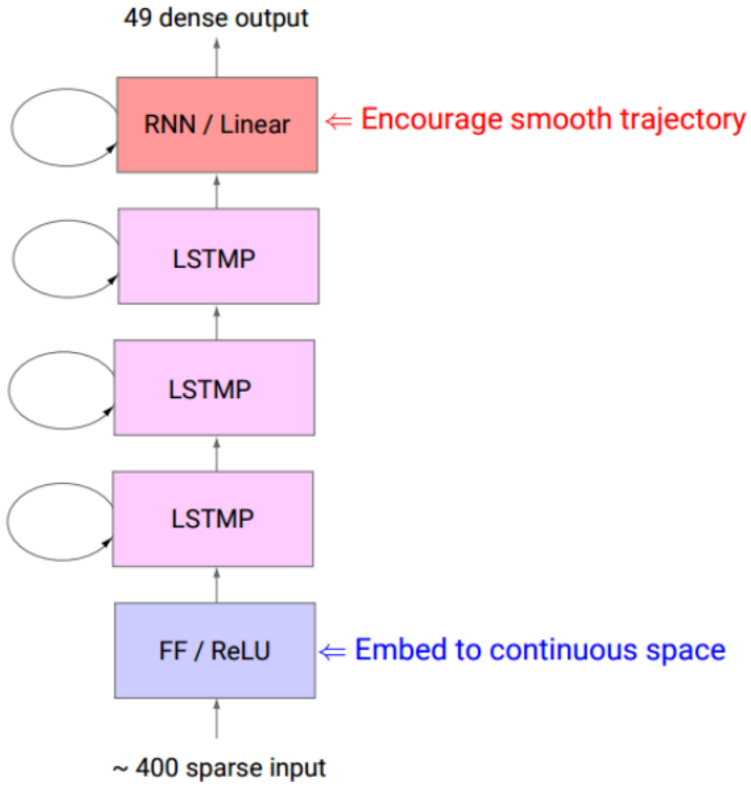
Model Architecture	Model Size （MB）	AM ACC	AUC Relative Reduction
DNN	2.092	49.0	-
LSTM	2.725	60.1	52.10%
FSMN	2.091	66.6	68.00%

FSMN for Text to Speech (TTS)

Text to Speech (TTS)<=>Speech synthesis

- **ASR:** Waveform->Spectral features->Text
- **TTS：** Text->Linguistic features->spectral features->waveform





FSMN for Text to Speech (TTS)

Text to Speech (TTS)<=>Speech synthesis

- **ASR:** Waveform->Spectral features->Text
- **TTS：** Text->Linguistic features->spectral features->waveform

	#Param	Valid FACC	Train Time per Epoch	Eval Speed
BLSTM	295M	0.528	330min	1x
FSMN(L10, O20, S1)	119M	0.529	250min	8.6x

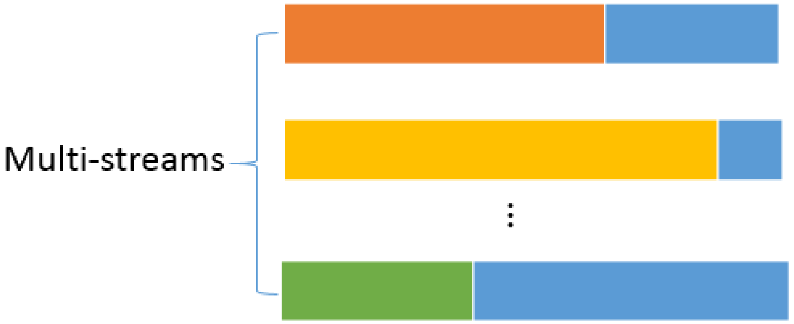
Open source

FSMN in Kaldi-Nnet1

- **Open source**: https://github.com/tramphero/kaldi or https://github.com/alibaba/Alibaba-MIT-Speech
- LibriSpeech recipe & reference results
- Two differences
  - Initialization method: Gaussian - > modified "xavier-glorot"

$$W \sim [-\beta \cdot \frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}, \beta \cdot \frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}]$$

  - Mini-batch based training instead of multi-streams
    - Stable & Faster
    - Basic CUDA Kernel Functions



Mini-batch: 2048, 4096 ..

Thank You！