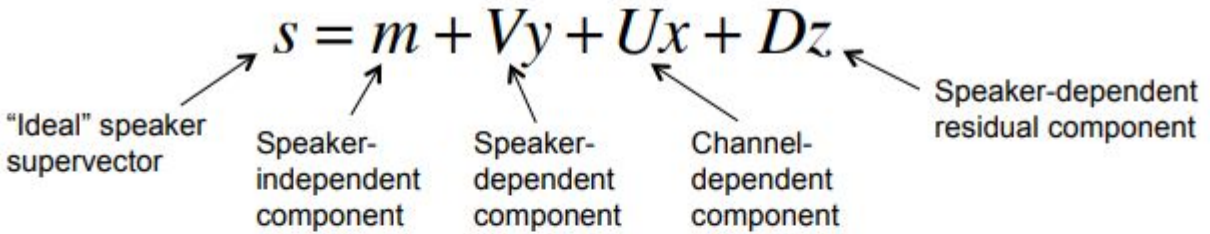
Speaker verification

Joint Factor Analysis (JFA)

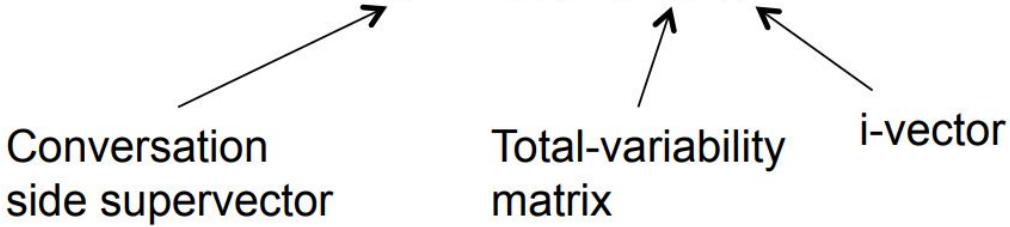


- where:
  - Vector m is a speaker-independent supervector (from UBM)
  - Matrix V is the eigenvoice matrix
  - Vector y is the speaker factors. Assumed to have N(0,1) prior distribution
  - Matrix U is the eigenchannel matrix
  - Vector x is the channel factors. Assumed to have N(0,1) prior distribution
  - Matrix D is the residual matrix, and is diagonal
  - Vector z is the speaker-specific residual factors. Assumed to have N(0,1) prior distribution

http://www1.icsi.berkeley.edu/Speech/presentations/AFRL_ICSI_visit2_JFA_tutorial_icsitalk.pdf

i-vector

$$s = m + Tw$$

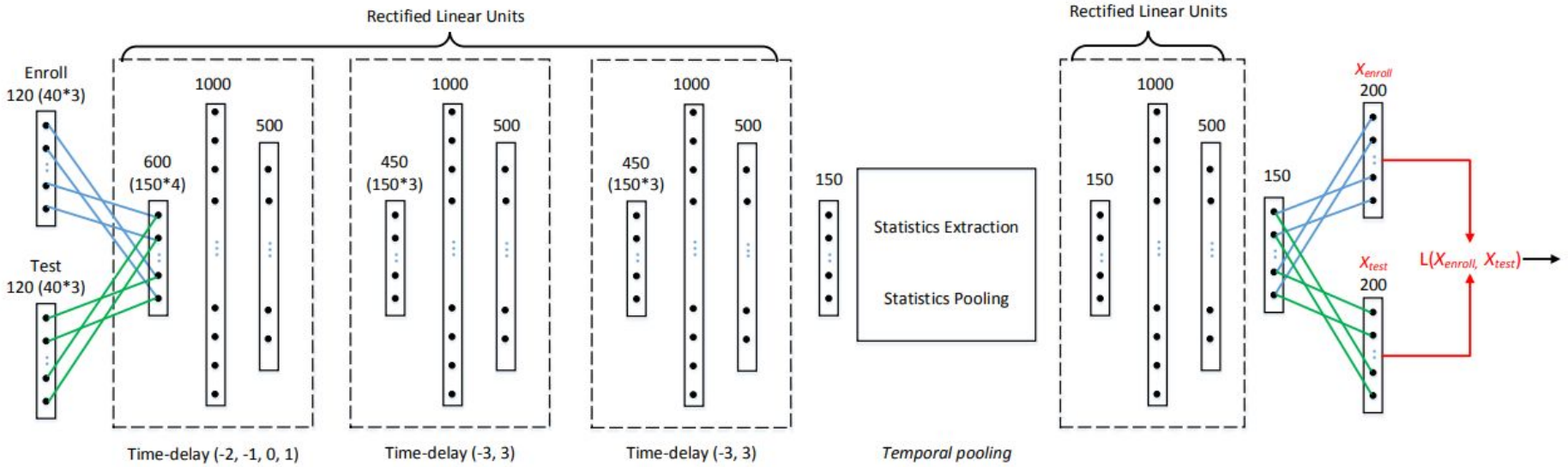


E2E vs Feature extraction

- E2E - metric learning
- feature extraction - классификация => embedding
- E2E - pairwise
- E2E - можно применить только для SV

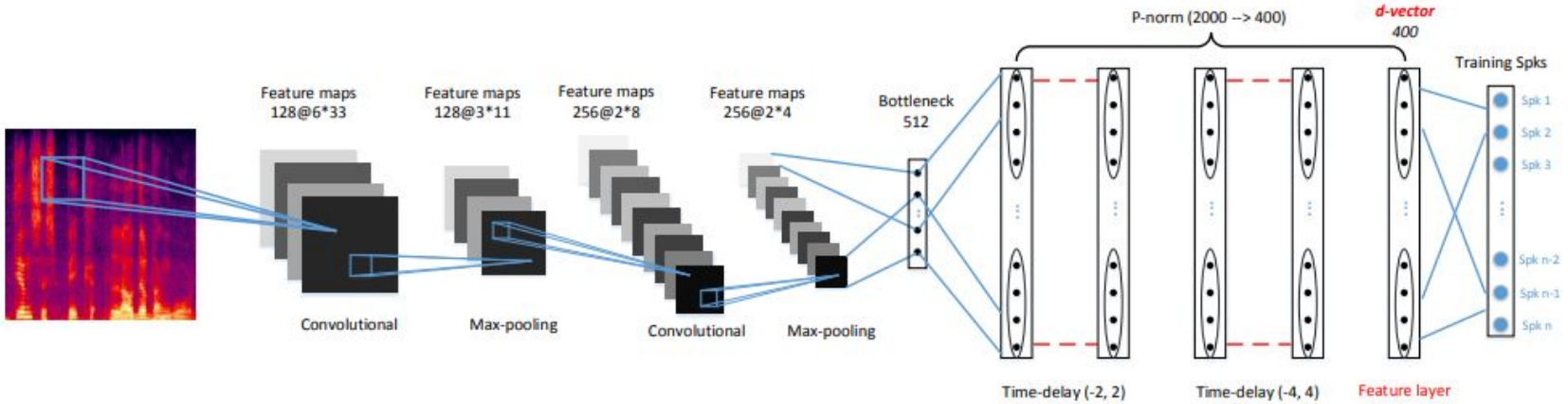
https://arxiv.org/pdf/1710.02369.pdf

E2E vs Feature extraction



$$L(x, y) = x^T y - x^T S x - y^T S y + b.$$

E2E vs Feature extraction



E2E vs Feature extraction

EER(%) RESULTS OF THE THREE SV SYSTEMS.

		EER%	
Systems	Scoring	C(4-4)	C(40-4)
i-vector	Cosine	16.96	4.81
	LDA	10.95	3.30
	PLDA	8.84	3.39
Deep feature	Cosine	10.31	4.01
	LDA	**7.86**	**2.39**
	PLDA	13.01	5.24
End-to-end	-	9.85	4.59

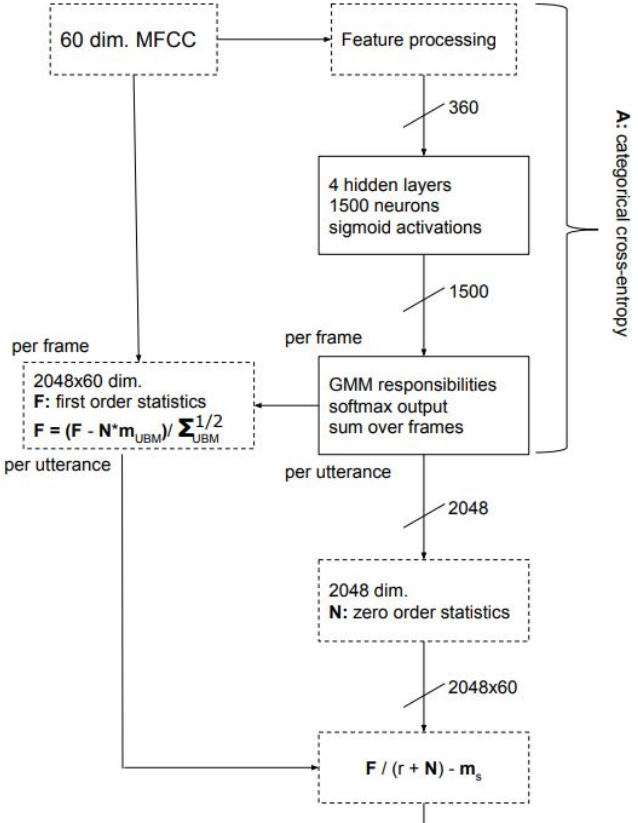
E2E vs Feature extraction

EER(%) RESULTS OF THE THREE SV SYSTEMS.

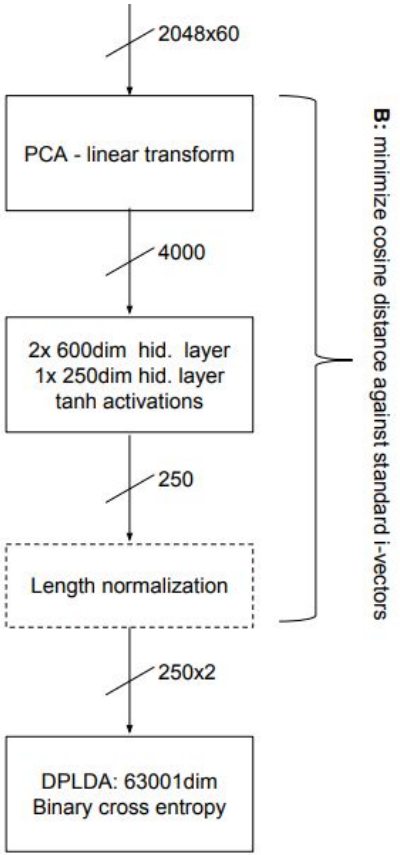
		EER%	
Systems	Scoring	C(4-4)	C(40-4)
i-vector	Cosine	16.96	4.81
	LDA	10.95	3.30
	PLDA	8.84	3.39
Deep feature	Cosine	10.31	4.01
	LDA	**7.86**	**2.39**
	PLDA	13.01	5.24
End-to-end	-	9.85	4.59

- LDA уменьшает вариацию внутри классов
- PLDA не взлетело потому что эмбединги не распределены нормально

END-TO-END DNN BASED SPEAKER RECOGNITION INSPIRED BY I-VECTOR AND PLDA (UBM->i-vector->(D)PLDA)



END-TO-END DNN BASED SPEAKER RECOGNITION INSPIRED BY I-VECTOR AND PLDA



https://arxiv.org/pdf/1710.02369.pdf

END-TO-END DNN BASED SPEAKER RECOGNITION INSPIRED BY I-VECTOR AND PLDA

	System Name	stats	i-vector	PLDA	SRE16		short lang		PRISM lang	
					$C_{\text{min}}^{\text{Prm}}$	EER	$C_{\text{min}}^{\text{Prm}}$	EER	$C_{\text{min}}^{\text{Prm}}$	EER
1	Baseline	UBM	i-extractor	Gen.	0.988	17.645	0.699	10.303	0.411	3.902
2	Baseline DPLDA	UBM	i-extractor	Discr.	0.975	16.902	0.616	9.462	0.360	3.461
3	f2s	NN	i-extractor	Gen.	0.980	16.809	0.687	9.866	0.394	3.713
4	s2i	UBM	NN	Gen.	0.988	16.686	0.788	11.141	0.430	4.584
5	f2s-s2i	NN	NN	Gen.	0.982	16.226	0.780	11.523	0.432	4.616
6	f2s-s2i-DPLDA	NN	NN	Discr	0.953	15.091	0.597	9.328	0.300	3.426
7	s2i-DPLDA_joint	NN	NN*	Discr.*	0.936	15.166	0.586	8.599	0.287	3.123
8	f2s-s2i-DPLDA_joint	NN*	NN*	Discr.*	0.936	15.170	0.587	8.661	0.287	3.125

END-TO-END DNN BASED SPEAKER RECOGNITION INSPIRED BY I-VECTOR AND PLDA

	System Name	stats	i-vector	PLDA	SRE16		short lang		PRISM lang	
					$C_{\text{min}}^{\text{Prm}}$	EER	$C_{\text{min}}^{\text{Prm}}$	EER	$C_{\text{min}}^{\text{Prm}}$	EER
1	Baseline	UBM	i-extractor	Gen.	0.988	17.645	0.699	10.303	0.411	3.902
2	Baseline DPLDA	UBM	i-extractor	Discr.	0.975	16.902	0.616	9.462	0.360	3.461
3	f2s	NN	i-extractor	Gen.	0.980	16.809	0.687	9.866	0.394	3.713
4	s2i	UBM	NN	Gen.	0.988	16.686	0.788	11.141	0.430	4.584
5	f2s-s2i	NN	NN	Gen.	0.982	16.226	0.780	11.523	0.432	4.616
6	f2s-s2i-DPLDA	NN	NN	Discr	0.953	15.091	0.597	9.328	0.300	3.426
7	s2i-DPLDA_joint	NN	NN*	Discr.*	0.936	15.166	0.586	8.599	0.287	3.123
8	f2s-s2i-DPLDA_joint	NN*	NN*	Discr.*	0.936	15.170	0.587	8.661	0.287	3.125

- Проблема E2E -- маленький батч для PLDA

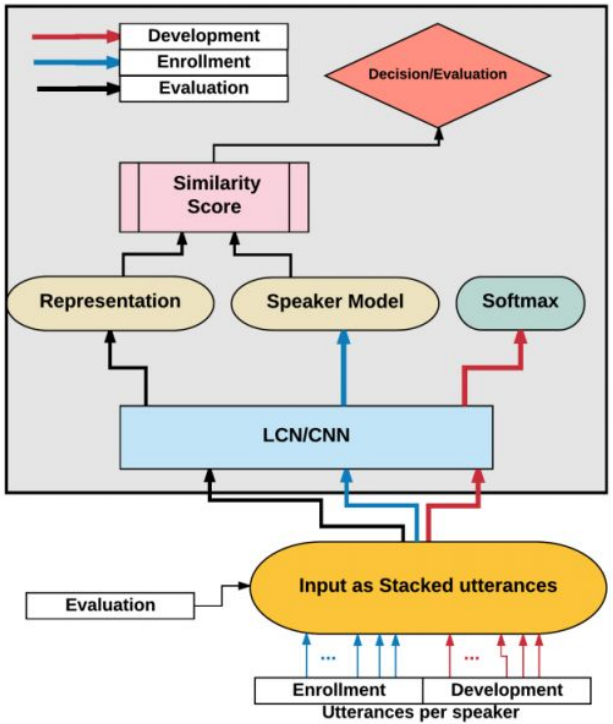
SV using 3D CNN (2017)

Минусы обычного d-vector подхода:

- frame-level не забирает всю относящуюся к спикеру контекстную информацию
- utterance level извлекает много инфы относящийся к тексту а не к спикеру
- Softmax + cross-entropy требует много примеров для каждого спикера чтобы обучиться

https://arxiv.org/pdf/1705.09422.pdf

SV using 3D CNN (2017)



layer	input-size	output-size	kernel	stride
Conv1-1	$\zeta \times 80 \times 40$	$80 \times 36 \times 16$	$3 \times 1 \times 5$	$1 \times 1 \times 1$
Conv1-2	$80 \times 36 \times 16$	$36 \times 36 \times 16$	$3 \times 9 \times 1$	$1 \times 2 \times 1$
Pool1	$36 \times 36 \times 16$	$36 \times 18 \times 16$	$1 \times 1 \times 2$	$1 \times 1 \times 2$
Conv2-1	$36 \times 18 \times 16$	$36 \times 15 \times 32$	$3 \times 1 \times 4$	$1 \times 1 \times 1$
Conv2-2	$36 \times 15 \times 32$	$15 \times 15 \times 32$	$3 \times 8 \times 1$	$1 \times 2 \times 1$
Pool2	$15 \times 15 \times 32$	$15 \times 7 \times 32$	$1 \times 1 \times 2$	$1 \times 1 \times 2$
Conv3-1	$15 \times 7 \times 32$	$15 \times 5 \times 64$	$3 \times 1 \times 3$	$1 \times 1 \times 1$
Conv3-2	$15 \times 5 \times 64$	$9 \times 5 \times 64$	$3 \times 7 \times 1$	$1 \times 1 \times 1$
Conv4-1	$9 \times 5 \times 64$	$9 \times 3 \times 128$	$3 \times 1 \times 3$	$1 \times 1 \times 1$
Conv4-2	$9 \times 3 \times 128$	$3 \times 3 \times 128$	$3 \times 7 \times 1$	$1 \times 1 \times 1$
FC5	$4 \times 3 \times 3 \times 128$	128	-	-

SV using 3D CNN (2017)

# utterances(ζ)	EER	AUC
5	24.5% $\pm$ 0.96	83.5% $\pm$ 1.06
10	22.9% $\pm$ 0.84	85.6% $\pm$ 1.12
20	**21.1% $\pm$ 0.73**	**87.3% $\pm$ 1.33**
40	21.7% $\pm$ 0.82	86.1% $\pm$ 1.17

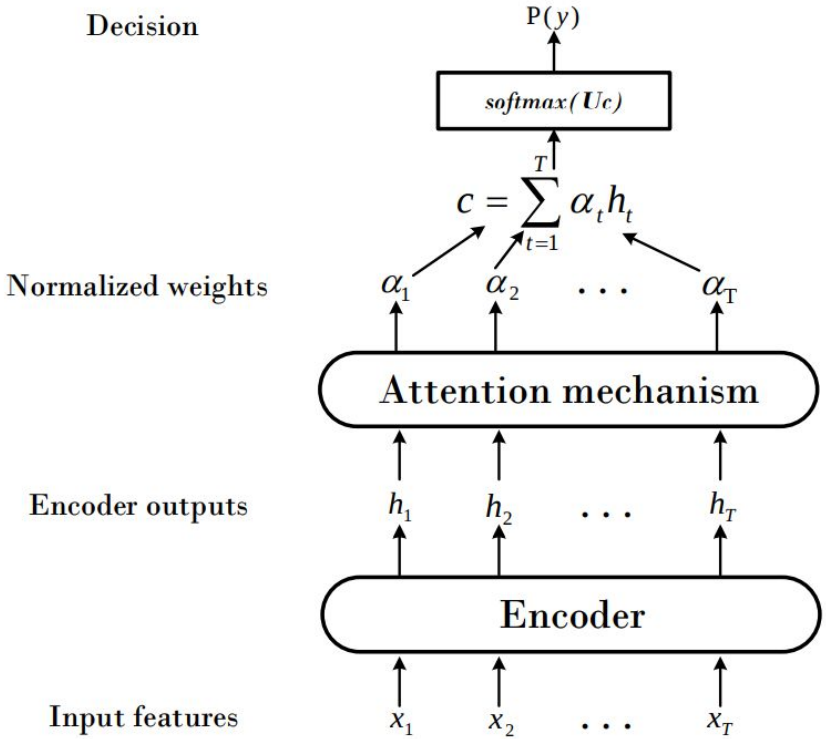
representation-level	model	system	EER	AUC
frame [21]	i-vector	-	25.3%	80.5%
frame [13]	LCN	d-vector	24.9%	81.2%
utterance [14]	LCN	d-vector	24.2%	82.6%
utterance [13]	CNN	d-vector	23.9%	83.1%
utterance [14]	LSTM	End-to-End	22.4%	86.0%
utterance [ours]	**3D-CNN**	proposed	**21.1%**	**87.3%**

Keyword spotting(KWS)

Используемые модели

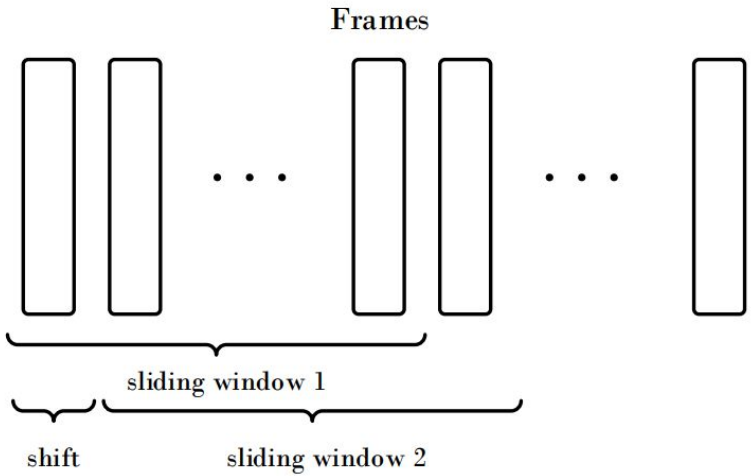
- large vocabulary continuous speech recognition
- Использовать HMM с GMM или NN - state of the art до недавнего времени
- CNN + RNN + CTC
- Модели с механизмом внимания

Attention-based End-to-End Model (Xiaomi 2018)



https://arxiv.org/pdf/1803.10916.pdf

Attention-based End-to-End Model (Xiaomi 2018)



https://arxiv.org/pdf/1803.10916.pdf

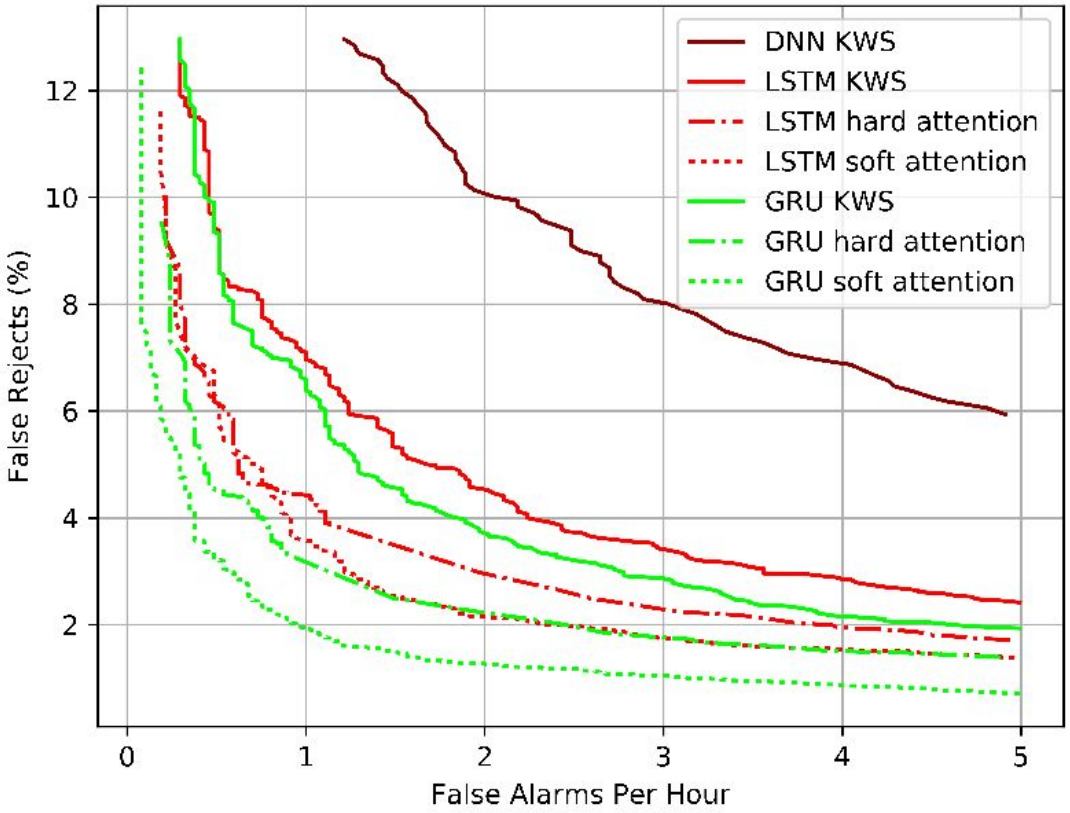
Attention-based End-to-End Model (Xiaomi 2018)

Average attention: $\alpha_t = \frac{1}{T}$

Soft attention: $e_t = v^T tanh(\boldsymbol{W} h_t + \boldsymbol{b})$ $\alpha_t = \frac{exp(e_t)}{\sum_{j=1}^{T} exp(e_j)}$

https://arxiv.org/pdf/1803.10916.pdf

Attention-based End-to-End Model (Xiaomi 2018)



https://arxiv.org/pdf/1803.10916.pdf

Attention-based End-to-End Model (Xiaomi 2018)

Model	FRR (%)	Params (K)
DNN KWS	13.9	62.5
LSTM KWS	7.10	54.1
LSTM average attention	4.43	60.0
LSTM soft attention	3.58	64.3
GRU KWS	6.38	44.8
GRU average attention	3.22	49.2
GRU soft attention	**1.93**	53.4

https://arxiv.org/pdf/1803.10916.pdf

Attention-based End-to-End Model (Xiaomi 2018)

Adding conv layers

Channel	Layer	Node	FRR (%)	Params (K)
8	1	64	2.48	52.5
8	2	64	1.34	77.3
16	1	64	**1.02**	84.1
16	2	64	1.29	109

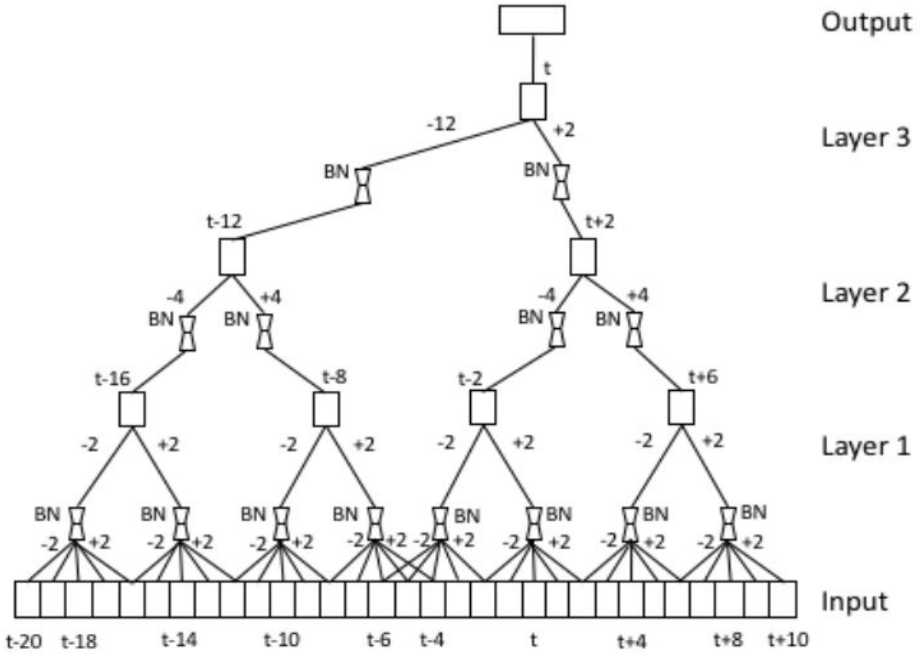
https://arxiv.org/pdf/1803.10916.pdf

Attention-based End-to-End Model (Xiaomi 2018)

- Простой пайплайн для продакшена
- В качестве энкодера пробовали LSTM, GRU и CRNN
- 84к параметров
- Пересчитывают только один фрейм во время инференса
- Задержка 100мс

https://arxiv.org/pdf/1803.10916.pdf

TDNN (Amazon 2017)

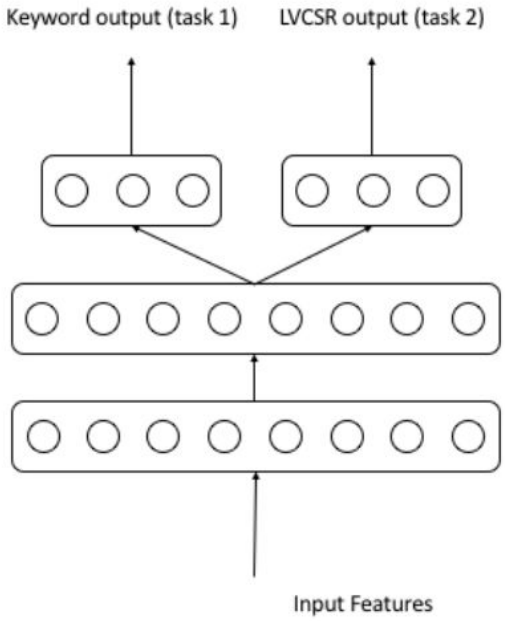


http://g-ecx.images-amazon.com/images/G/01/amazon.jobs/Interspeech_2017_4._CB503635227_.pdf

TDNN (Amazon 2017)

Multi-task training

$$\mathcal{L}_t^* = \lambda \mathcal{L}_t^1 + (1-\lambda)\mathcal{L}_t^2$$



TDNN (Amazon 2017)

Full training process

i. Train a full-rank LVCSR TDNN with the same architecture as the full-rank keyword TDNN. The hidden layers of the LVCSR TDNN are used for initialization.

ii. Add a separate hidden layer and an output layer for the main keyword spotting task. Other hidden layers are initialized by the LVCSR TDNN model.

iii. Train a TDNN jointly with the keyword spotting task and LVCSR task using multi-task learning setup.

iv. Add linear bottleneck layers to the full-rank TDNN and pre-train. These linear bottleneck layers are initialized by SVD.

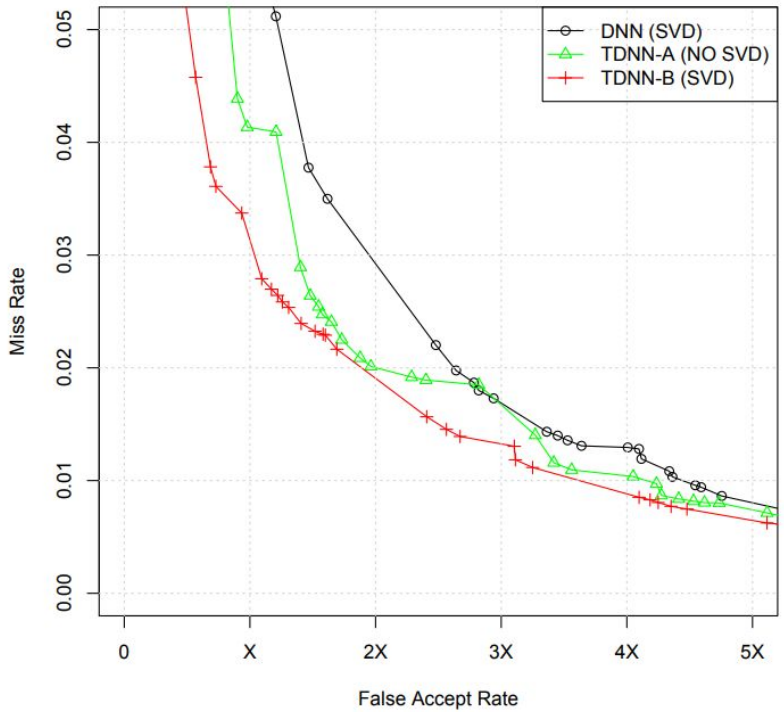
v. Run additional epochs of multi-task fine-tuning for the SVD compressed TDNN.

vi. Remove the last separate hidden layer and output layer for the LVCSR task. The remaining TDNN is used for keyword spotting.

TDNN (Amazon 2017)

Results

Model	DNN	TDNN-A	TDNN-B
AUC Relative Change	0%	−19.7%	−37.6%



Voice activity detection

VAD из G.729B

- На вход line spectral frequencies (LSF), full-band energy, low-band energy, zero-crossing rate
- Параметры на начальных фреймах считают параметрами фонового шума
- Результат сглаживается по фреймам

https://arxiv.org/pdf/1803.10916.pdf

End-of-utterance detection

EOQ

- Самый простой подход считать что высказывание закончено если VAD выдает нули некоторое время
- Это не учитывает звуковые “подсказки”

https://arxiv.org/pdf/1803.10916.pdf

VAD vs EOQ(CNN + 2 LSTM)