STRUCTURAL SPARSIFICATION FOR FAR-FIELD SPEAKER RECOGNITION WITH INTEL® GNA

Jingchi zhang* , Jonathan Huang△, Michael Deisher △, Hai Li *, Yiran Chen*

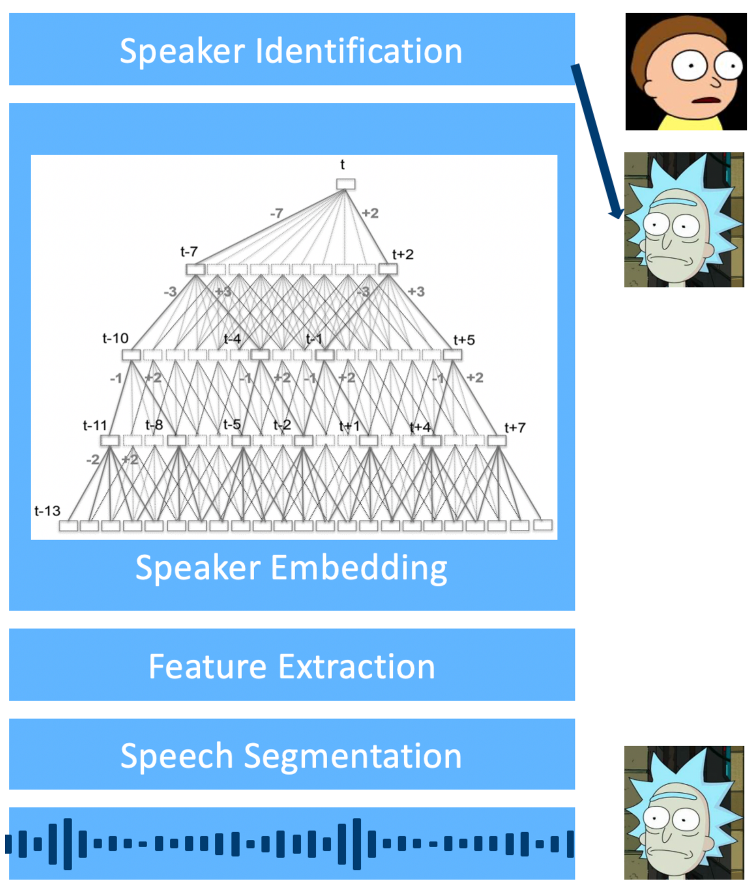
* Duke University

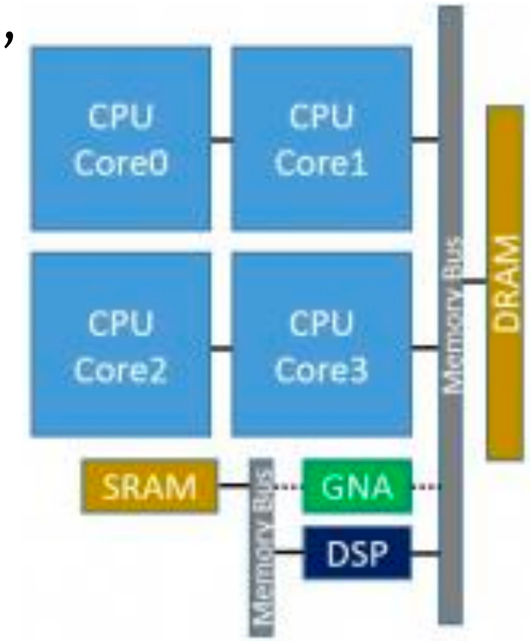
△ Intel Corporation

Background: Speaker Recognition and GNA

Answering one question: "Who is the speaker?"

- Log into your Netflix account on a family laptop.
- Personalization: play my favorite music.





INTEL® GNA takes 8 16bit integers or 16 8bit integers per DMA transaction

<u>Motivation: accelerate the Speaker Recognition models on GNA.</u>

Methodology: Structural sparsity

- Learning structural sparsity during training:
  1. Split the weight into groups $w_{(1,\dots,K)}$:

     e.g. a matrix --> K vectors
  2. Apply L2 regularization on each w_k:

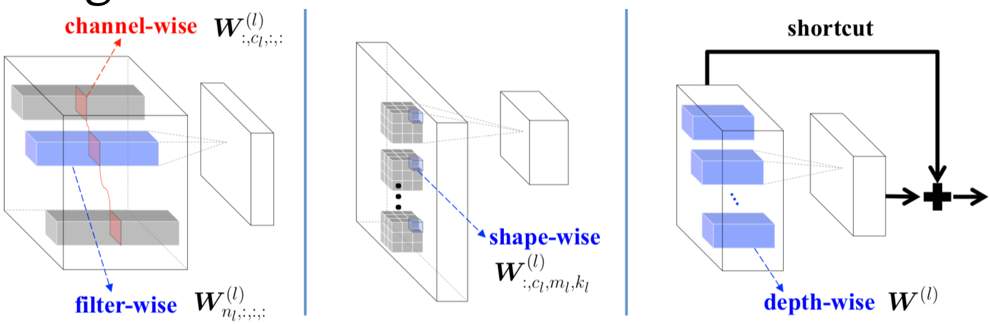
     $$\|w_k\|_2 = \sqrt{\sum_{i=1}^{|w_k|} (w_{k_i})^2} \quad (\mathrm{L_2\ norm})$$
  3. Sum L2 over all groups as a regularization (Group Lasso regularization) and add it to loss function:

     $$R(w) = \sum_{k=1}^{K} \|w_k\|_2$$
  4. Optimize new loss function:

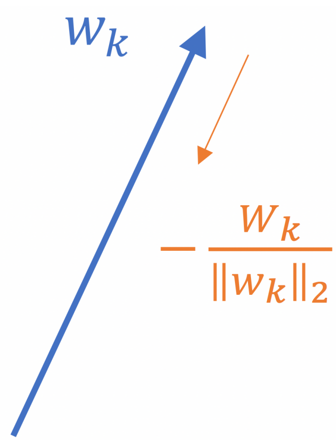
     $$\arg\min_w \{E(w)\} = \arg\min_w \{E_D(w) + \lambda \cdot R(w)\}$$

     In actual update in SGD:

     $$w_k \leftarrow w_k - \eta \cdot \left(\frac{\partial E_D(w)}{\partial w_k} + \lambda \cdot \frac{w_k}{\|w_k\|_2}\right)$$



Wen, W., Wu, C., Wang, Y., Chen, Y. and Li, H., 2016. Learning structured sparsity in deep neural networks. In Advances in neural information processing systems (pp. 2074-2082).



- Fixed the sparse structures and retrain the model:

  $$w_k \leftarrow w_k - \eta \cdot \left(\frac{\partial E_D(w)}{\partial w_k} \cdot \theta(w_k)\right), \text{ where } \theta(\xi) = \begin{cases} 0, \xi = 0 \\ 1, \xi \neq 0 \end{cases}$$

Experiment: Setup

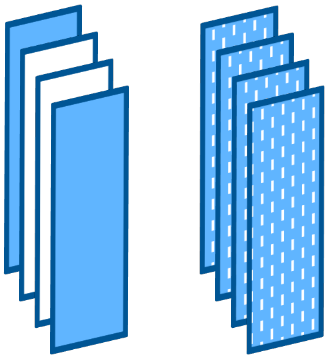
- Model topology:
  - Based on x-vector model structure and write TDNN as one-dimension CNN layer.
- Loss function:
  - Additive Margin Softmax (AM-softmax)
  - Eliminate the PLDA and easy to deploy on hardware
- Training detail:
  - Initialize with a pretrained dense model
  - Train with sparse regularization
  - Finetune without sparsity
- Dataset:
  - Training: VoxCeleb 1 and 2
  - Testing: VOiCES far-field
  - Data augmentation: Pyroomacoustic, MUSAN and AudioSet

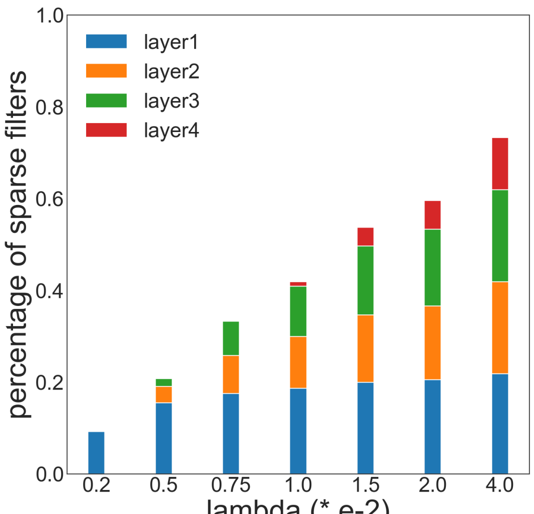
	layer context	Affine	Convolution
Layer1	[t-2,t+2]	200×512	512 40×5
Layer2	{t-2,t,t+2}	1536×512	512 512×3
Layer3	{t-2,t,t+2}	1536×512	512 512×3
Layer4	{t}	512×512	512 512×1
Layer5	{t}	512×512	512 512×1
Stats pooling	[0,T)	512T×1024	N/A
Segment6	{0}	1024×256	N/A
Softmax	{0}	256×N	N/A

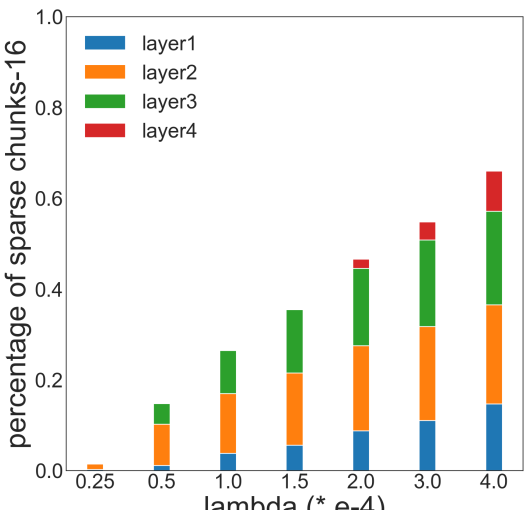
denotes the number of training speakers.

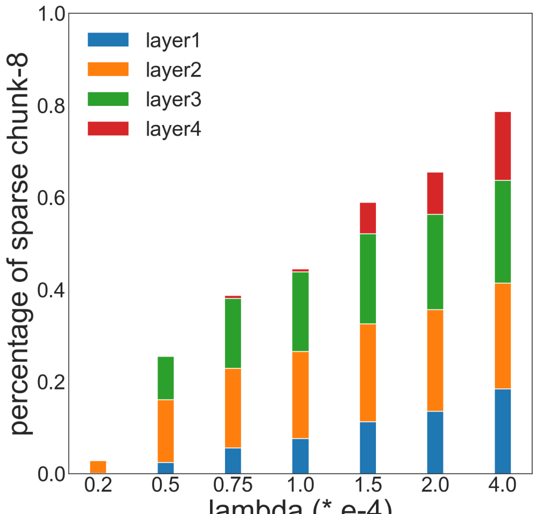
Result: Sparsity

- Apply sparsity on:
  - Filters: for all hardware, easy to deploy
  - Chunks: every 8 or 16 elements, for GNA
  - Only on first four layers
- When λ increases, the sparsity increases.
- The sparsity growth in each layer is different.
- In layer 4 the sparsity would result in higher penalty on the AM-softmax loss.



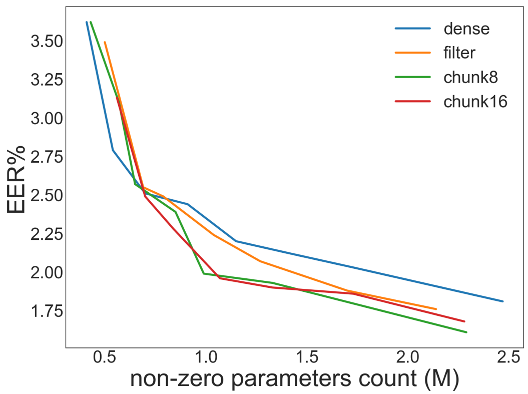


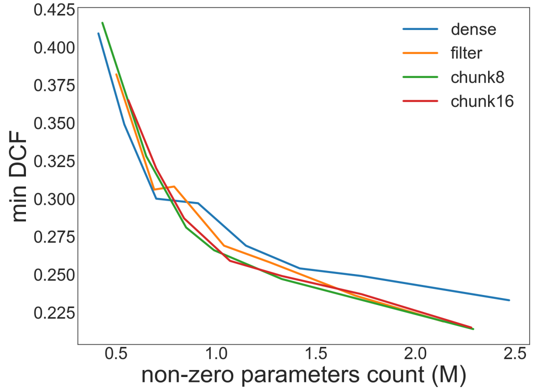


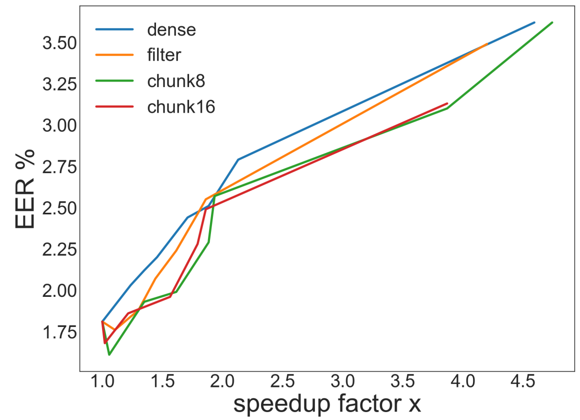


Result: Performance

- Compared with dense model as a baseline:
  - When the number of non-zero parameters is large, sparse models achieve lower EER. When the number of non-zero parameters is small, dense models have better performance.
  - When non-zero parameter count is larger than 1.5 million, there is a tendency that chunk-8 has the best performance.
- Actual speedup on GNA:
  - Under the same EER, structural sparse models are always faster than the dense models.
  - When speedup is around 1.2x, sparse models even have lower EER.







Conclusion

- In this paper, we applied structural sparsification for speaker recognition models.
- By using group Lasso regularization, we kept the good performance of the original model while reducing the number of parameters and accelerating the actual inference of the models.

- Feel free to contact: jingchi.zhang@duke.edu