How Redundant is the Transformer Stack in Speech Representation Models?

Teresa Dorszewski*, Albert Kjøller Jacobsen*, Lenka Tětková, Lars Kai Hansen
Technical University of Denmark, Section for Cognitive Systems

Motivation

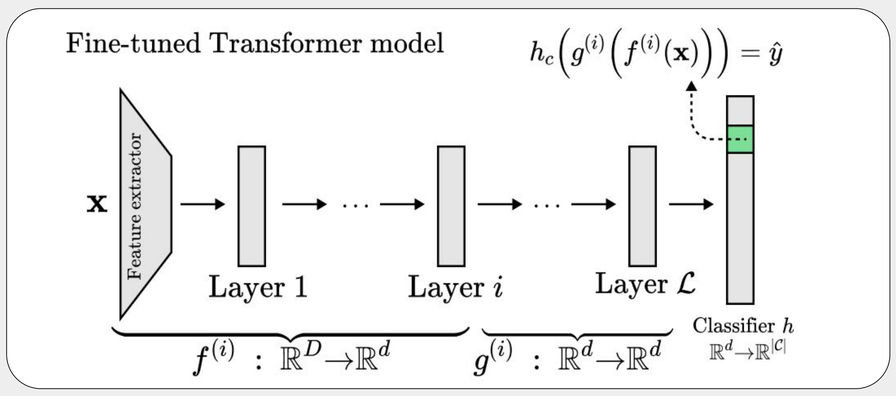
- Transformer-based **speech representation models perform well** but are **computationally demanding,** limiting their on-device applications.
- Recent studies on LLMs and speech models [1, 2, 3, 4, 5] reveal **redundancy of transformer layers**.
- **Reducing computational requirements** allows for **efficient inference** in resource-constrained environments.

Our contribution

Showing **redundancy** in **speech representation models** through **similarity analyses, pruning and mimicking selected layers.**

Notation

The general architecture of a transformer-based speech representation models with a word classification output layer:



Representations of the speech transformer at layers i and j are denoted:

$$A = f^{(i)}(X) \in \mathbb{R}^{n \times d} \qquad B = f^{(j)}(X) \in \mathbb{R}^{n \times d}$$

Layerwise similarity analysis

Similarity analysis of transformer-stack representations using three metrics:

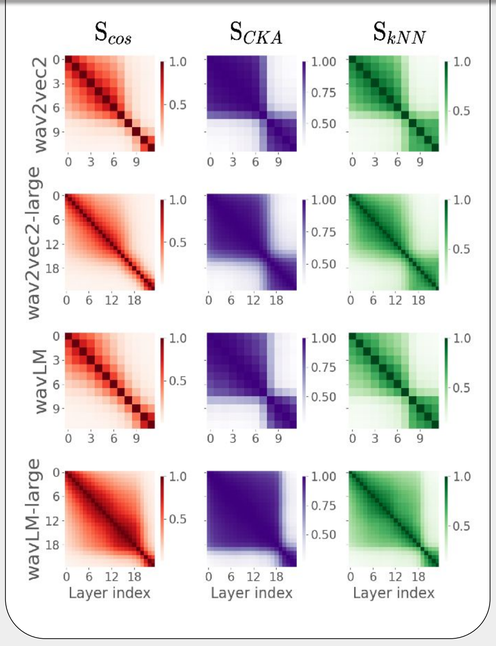
$$\mathrm{S}_{cos}(i,j) = \frac{1}{n}\sum_{l=1}^{n}\frac{A_{l,\cdot}^T B_{l,\cdot}}{||A_{l,\cdot}|| \cdot ||B_{l,\cdot}||}$$

$$\mathrm{S}_{CKA}(i,j) = \frac{||B^T A||_F^2}{||A^T A||_F ||B^T B||_F}$$

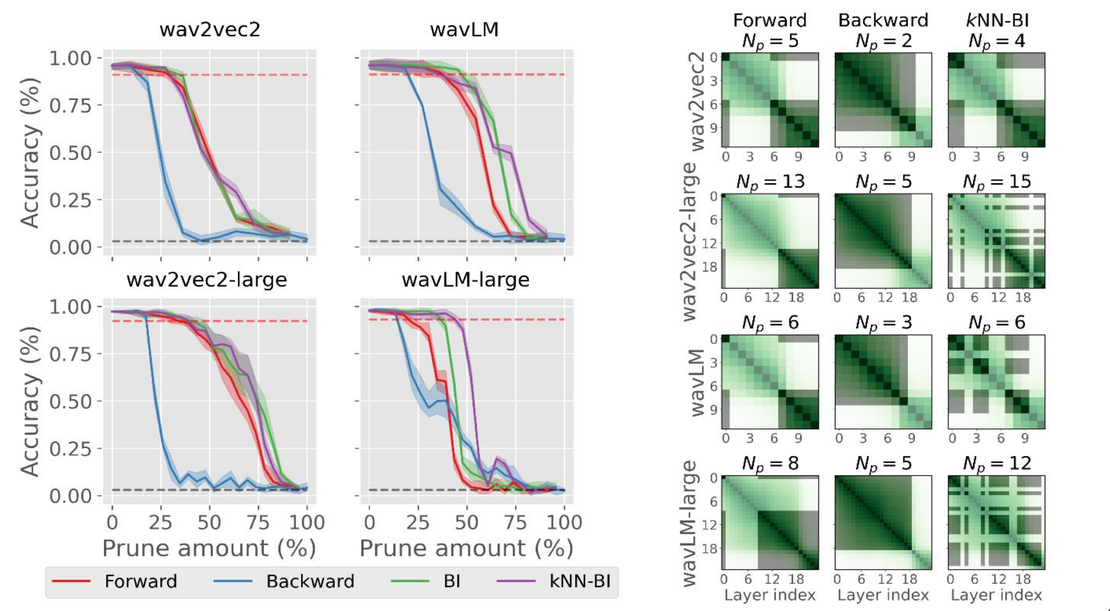
$$\mathrm{S}_{kNN}(i,j) = \frac{1}{n}\sum_{l=1}^{n}\left(\frac{1}{k}|\mathcal{N}_k(A_{l,\cdot}) \cap \mathcal{N}_k(B_{l,\cdot})|\right)$$

Cosine and kNN similarities are used for block-influence pruning heuristic [1, 6]:

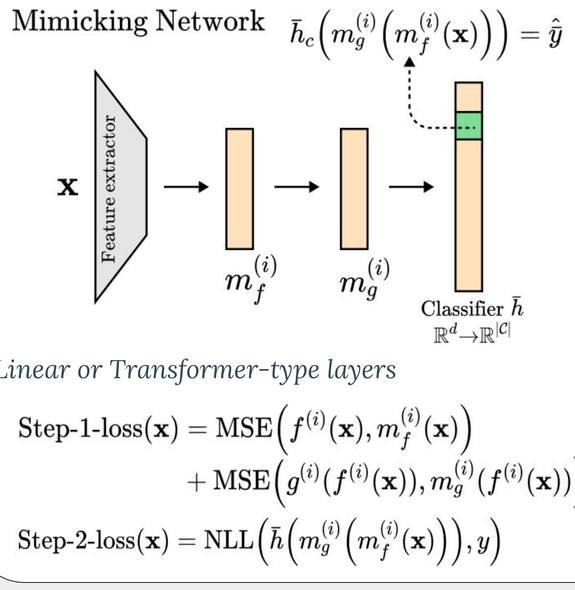
$$\mathrm{BI}(i) = 1 - \mathrm{S}_{cos}(i-1, i)$$



Pruning the transformer by heuristics



Knowledge distillation by mimicking selected layers



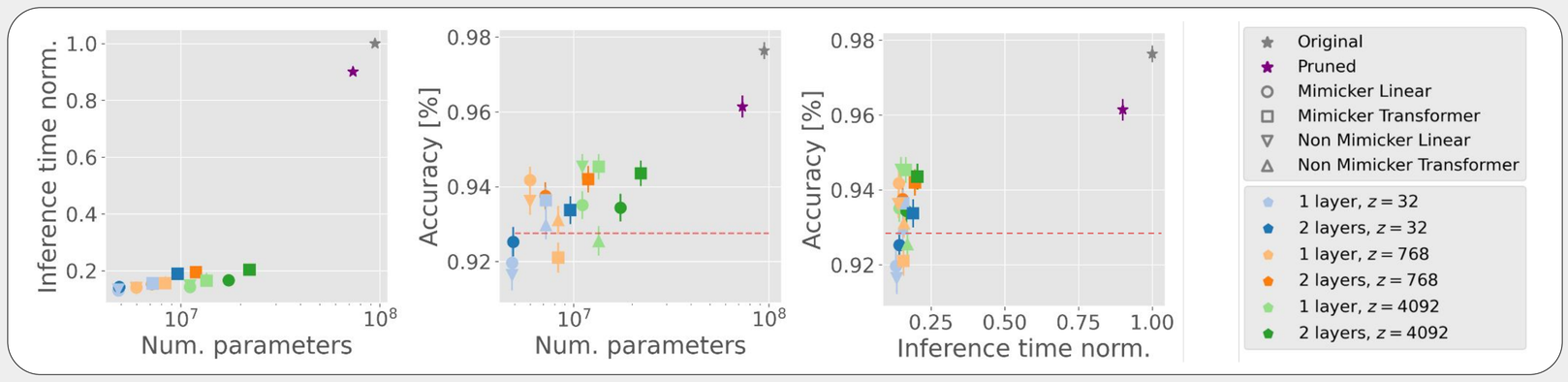
Linear or Transformer-type layers

$$\text{Step-1-loss}(\mathbf{x}) = \mathrm{MSE}\left(f^{(i)}(\mathbf{x}), m_f^{(i)}(\mathbf{x})\right) + \mathrm{MSE}\left(g^{(i)}(f^{(i)}(\mathbf{x})), m_g^{(i)}(f^{(i)}(\mathbf{x}))\right)$$

$$\text{Step-2-loss}(\mathbf{x}) = \mathrm{NLL}\left(\bar{h}\left(m_g^{(i)}\left(m_f^{(i)}(\mathbf{x})\right)\right), y\right)$$

Mimicking networks learn to **reproduce selected latent representations**.

- **Step 1: *mimicking phase*** *using MSE loss*
- **Step 2: *adaption phase*** *using fine-tuning (NLL) loss.* **Non-mimicker models** only learn with NLL loss.



Conclusions

We find:

1. **block-like similarity structure** suggesting two main processing steps.
2. that **pruning up to 45% of the layers** of transformer-based speech representation models can be done **with low performance drop**.
3. importance of **both similarity blocks** to maintain performance.
4. that **mimicking layers maintain 95% of original performance while reducing inference time by up to 87%.**

References & sponsors

1. Xin Men et al., "Shortgpt: Layers in large language models are more redundant than you expect", *arXiv preprint* 2024
2. Andrey Gromov et al., "The unreasonable ineffectiveness of the deeper layers", *arXiv preprint* 2024.
3. Anton Razzhigaev et al.,, "Your transformer is secretly linear", *arXiv preprint* 2024.
4. Ankita Pasad et al, "Comparative layer-wise analysis of self-supervised speech models", *ICASSP IEEE 2023*
5. Teresa Dorszewski et al., "Convexity-based pruning of speech representation models", *MLSP IEEE 2024*
6. Minyoung Huh et al., "The platonic representation hypothesis", *ICML 2024*

PIONEER CENTRE FOR ARTIFICIAL INTELLIGENCE
DIREC Digital Research Centre Denmark
novo nordisk fonden
IEEE Signal Processing Society