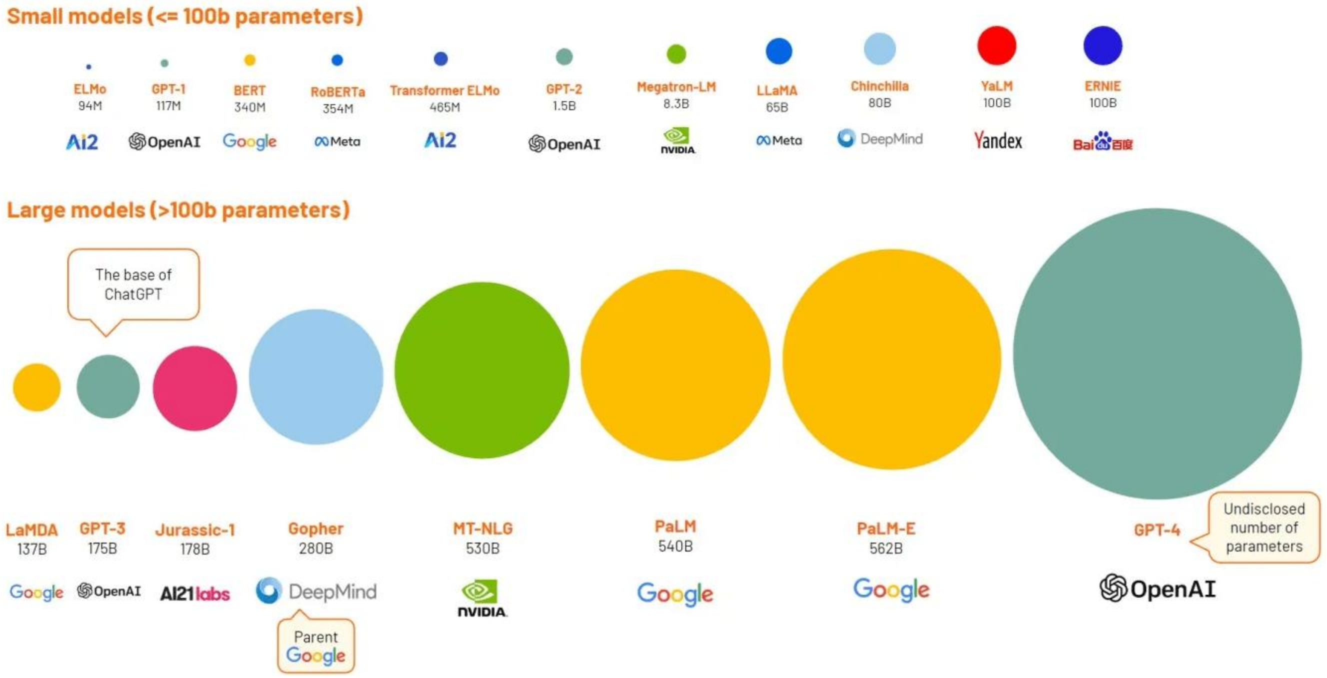
BEEM: Boosting Performance of Early Exit DNNs Using Multi-Exit Classifiers as Experts

AUTHORS: DIVYA JYOTI BAJPAI AND MANJESH KUMAR HANAWAL

PUBLISHED AT INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS (ICLR) 2025

Introduction

- To achieve better accuracy, DNNs have grown significantly.
  - ➢ LLAMA - **7 Billion** Parameters
  - ➢ BLIP-2 – **4.1 Billion** Parameters
  - ➢ GPT-3 – **175 Billion** Parameters

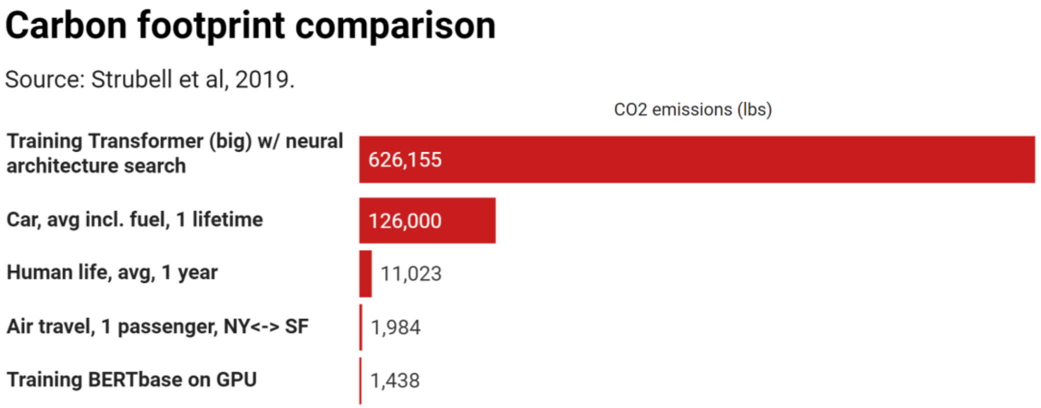


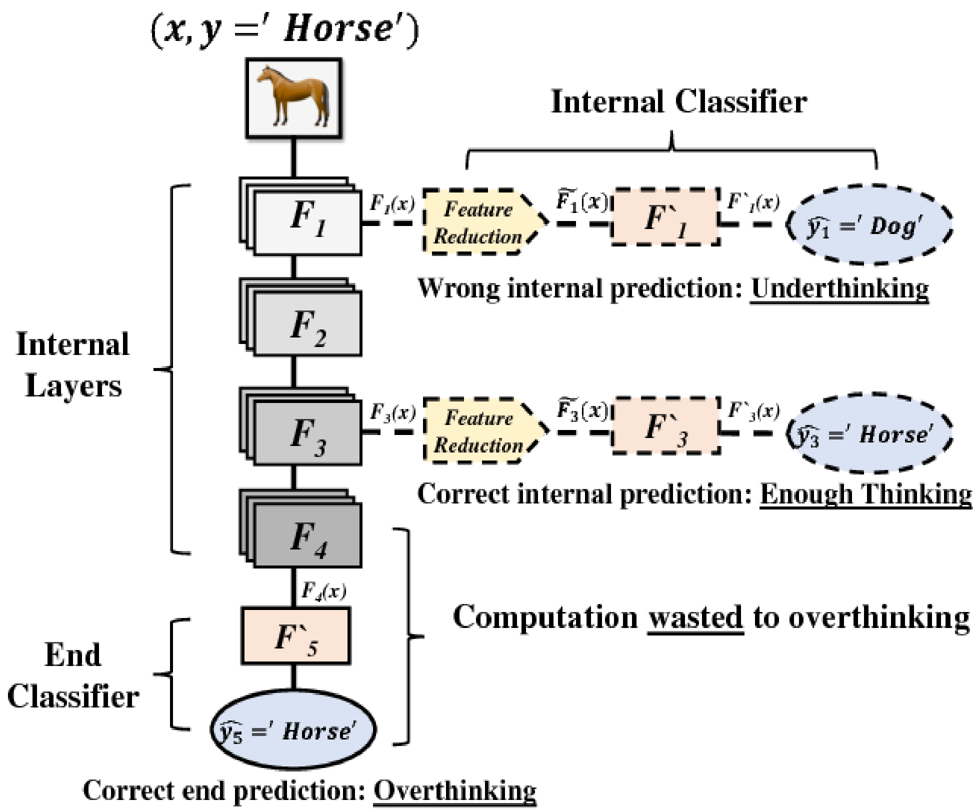
1 Xin, Ji, et al. "DeeBERT: Dynamic early exiting for accelerating BERT inference." *arXiv preprint arXiv:2004.12993* (2020)

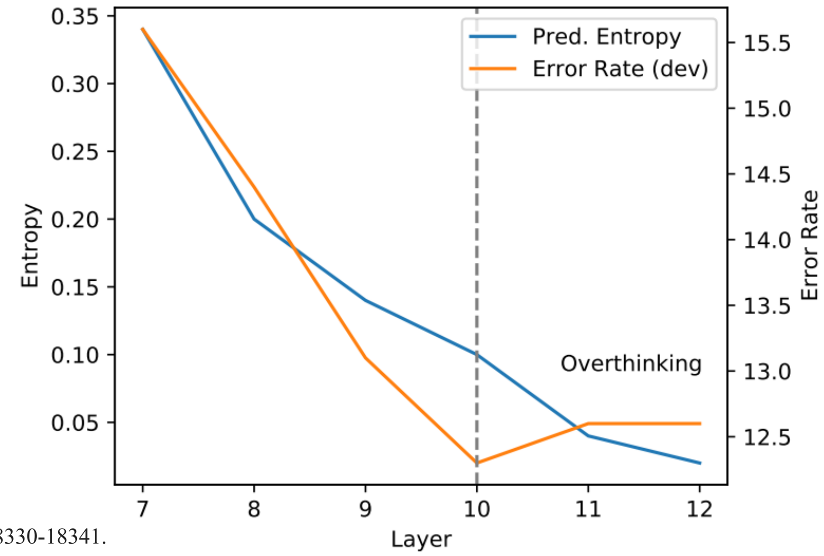
[2] Zhou, Wangchunshu, et al. "Bert loses patience: Fast and robust inference with early exit." *Advances in Neural Information Processing Systems* 33 (2020): 18330-18341.

Introduction

- To achieve better accuracy, DNNs have grown significantly.
  - LLAMA - **7 Billion** Parameters
  - BLIP-2 – **4.1 Billion** Parameters
  - GPT-3 – **175 Billion** Parameters
- Large size improves accuracy but:
  - Requires more resources [1].
  - Environmental factors.
  - Inference latency.
- Overthinking [2]:
  - Datasets consist of a mixture of easy and hard samples.
  - DNNs are highly overparameterized for easier inputs.







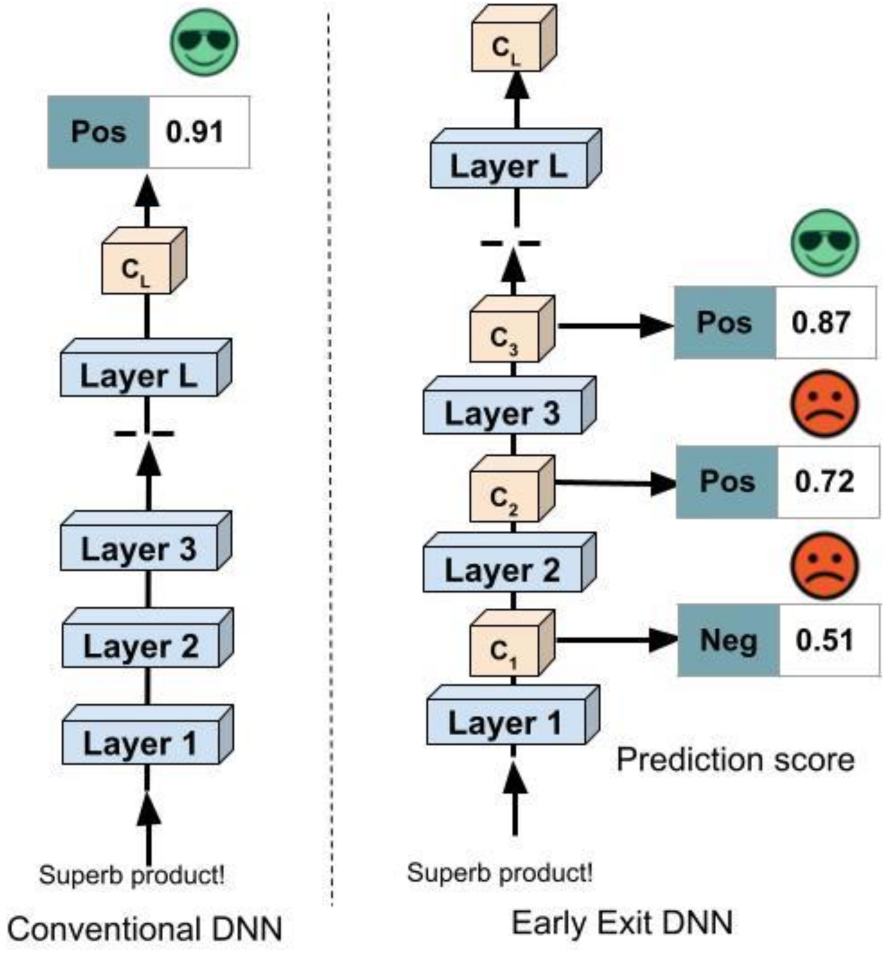
1 Xin, Ji, et al. "DeeBERT: Dynamic early exiting for accelerating BERT inference." *arXiv preprint arXiv:2004.12993* (2020)

2 Zhou, Wangchunshu, et al. "Bert loses patience: Fast and robust inference with early exit." *Advances in Neural Information Processing Systems* 33 (2020): 18330-18341.

Early Exits

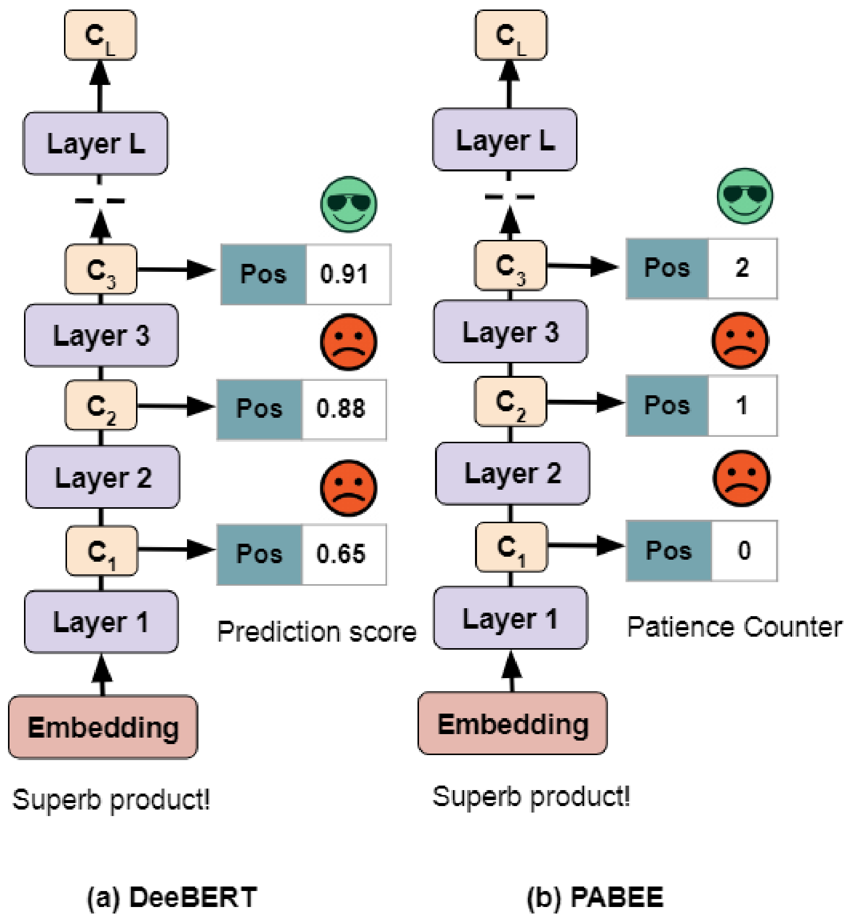
Early Exits are input adaptive inference methods where the inference is performed based on the sample complexity.

Advantages:	Disadvantages:
• Reduction in inference latency. • Reduces overthinking. • Input-adaptive inference. • Generalizable.	• Requires to train exits (Dataset required). • Performance degradation where deeper layer knowledge is required.



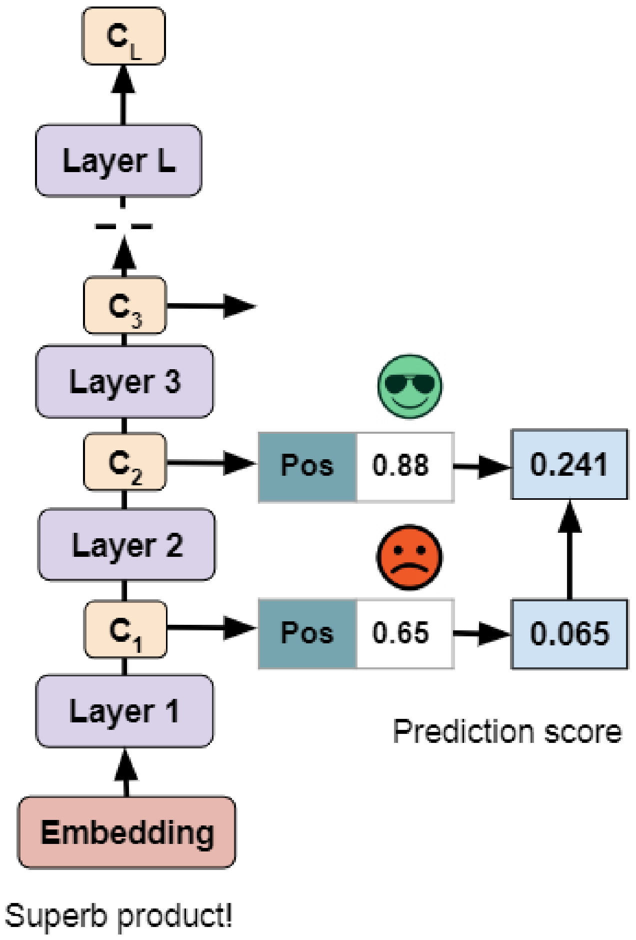
Related Works and Problem setup

- Confidence-based exiting: Confidence (entropy or highest probability) above a threshold.
- Patience-based: Consistent predictions t times.
- Max of All: Consistency on a class for t times.
- Confidence based methods are overly confident towards a single class and classifier.
- Patience-based and Max of all methods treat each classifier equally.
- It also affects the adaptability of the model.
- There is a need of a confidence score that can utilize all the classifiers effectively.



BEEM

- We propose an approach that works on ensemble learning principles.
- We treat each exit as an expert that provides predictions.
- Our method utilizes the strengths of individual exits.
- It proposes a confidence score that aggregates the weighted confidence of previous classifiers.
- The weights are provided based on accuracy of validation dataset or the cost associated with expert.



(c) Ours

BEEM-Training & inference

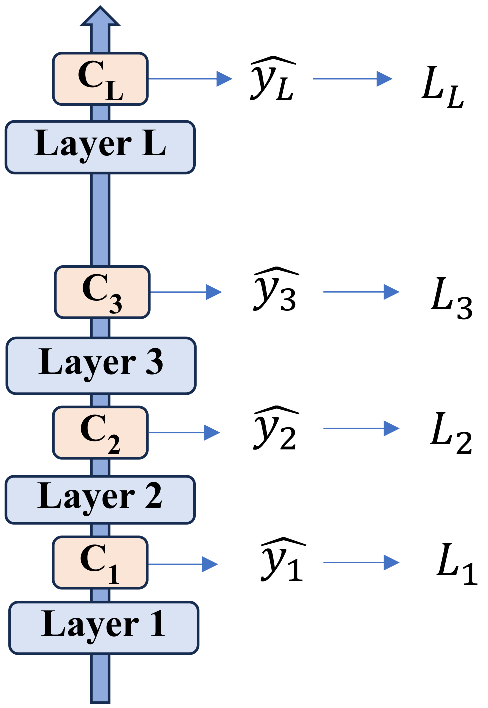
- We train the backbone using joint optimization.
- The exits are trained using CE and KD loss.
- After getting the backbone ready for inference, we define the score as:

$$S_i = \begin{cases} S_{i-1} + w_i C_i & \textit{if} \quad \hat{y}_{i-1} = \hat{y}_i \\ w_i C_i & \textit{if} \quad \hat{y}_{i-1} \neq \hat{y}_i \end{cases}$$

- Sample will exit if $S_i \geq \alpha$.
- Weights w_i can be set using accuracy on validation set.
- Or we can assign $w_i = \lambda i$ where λ is the cost.

Setting the thresholds:

- Note that we can change the error rates of internal classifiers using thresholds.
- We can perform better than final classifier if error rates of ICs q_{α_t} are smaller than FC.



Joint Optimization of Backbone

$L_i = CE(y, \hat{y}_i) + KD(y_L, y_i)$

Total loss:

$$L = \frac{\sum_{i \in [L]} i L_i}{\sum_{i \in [L]} i}$$

Threshold choice

- The threshold models the accuracy-efficiency trade-off.
- The error rates of the classifier depend on the threshold.

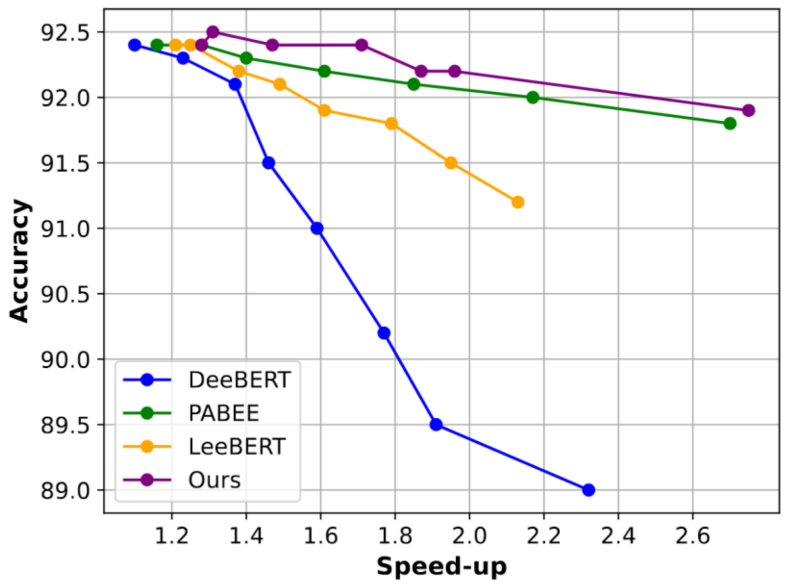
$$\begin{aligned} \underset{\alpha_t \in S}{\text{minimize}} \quad & \alpha_t \\ \text{subject to} \quad & q_{\alpha_t} \leq p, \end{aligned}$$

- We find the optimal α_t as the minimum α such that it is better than the final layer.
- This guarantees that our method is always better than the final layer.

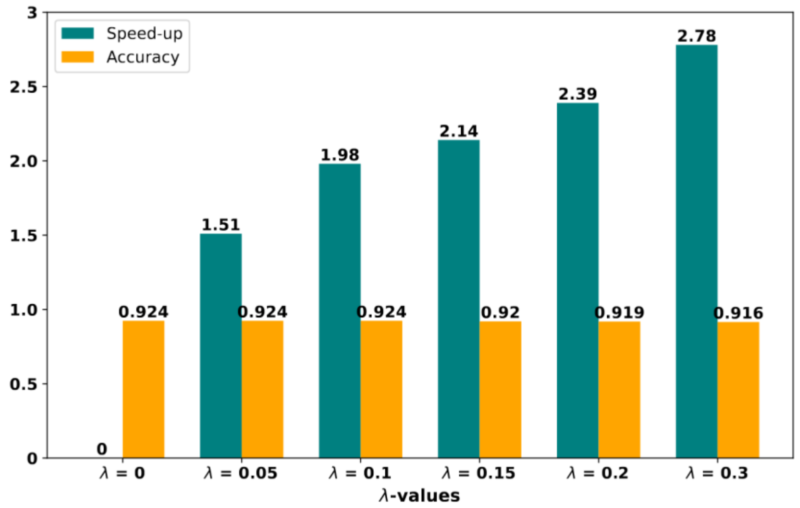
Consider an early exit PLM with L layers. Let p denote the error rate of the final classifier and the error probability of the ith exit classifiers be q_i such that $q_i < \frac{a_i}{a_i + \left(\left(\frac{1}{p} - 1\right) b_i^{i-1}\right)}$ for all exit layers $i = 1, 2, \ldots L-1$ where a_i and b_i are constants for a given exit i then BEEM performs better than the final layer.

Experiments

- We perform experiments on GLUE datasets, comprising of CoLA, RTE, QQP, SST-2, MNLI, QNLI datasets.
- We use BERT and ALBERT models and their base and large variants.
- We set the cost to be 0.1.
- All the metrics are same as used in CeeBERT.



Speedup vs Accuracy curve



Rate of change with λ.

Results

Model/Data	SST-2		MNLI		RTE		QNLI		QQP	
	Acc	Speed	Acc	Speed	Acc	Speed	Acc	Speed	Acc	Speed
Dev set										
ALBERT	92.4	1.00x	84.5	1.00x	77.9	1.00x	91.3	1.00x	90.6	1.00x
ALBERT-9L	-1.6	1.33x	-3.2	1.33x	-2.5	1.33x	-2.7	1.33x	-1.5	1.33x
DeeBERT	-2.3	1.72x	-2.9	1.65x	-3.1	1.78x	-1.9	1.57x	-2.5	1.81x
ElasticBERT	-2.1	1.75x	-2.3	1.71x	-2.7	1.81x	-1.7	1.66x	-2.1	1.78x
FastBERT	-1.1	1.85x	-0.3	1.61x	-0.2	1.79x	-0.8	1.71x	-0.3	1.88x
PABEE	-0.1	1.87x	-0.5	1.85x	-0.7	1.64x	-0.6	1.81x	-0.2	1.68x
ZTW	-0.2	1.64x	-0.3	1.67x	+0.2	1.63x	-0.3	1.75x	-0.1	1.71x
PCEEBERT	+0.1	1.24x	0.0	1.31x	+0.3	1.27x	-0.1	1.21x	+0.1	1.37x
LeeBERT	0.0	1.78x	-0.2	1.74x	-0.1	1.59x	+0.1	1.79x	-0.2	1.97x
PALBERT	-0.4	1.54x	-0.8	1.61x	+0.3	1.45x	-0.2	1.59x	-0.1	1.63x
JEI-DNN	-0.1	1.77x	+0.1	1.67x	0.0	1.35x	-0.1	1.43x	+0.2	1.57x
BEEM-C	0.0	1.71x	+0.1	**2.03x**	+0.4	1.79x	0.0	1.90x	0.0	1.93x
BEEM-A	**+0.4**	**1.98x**	**+0.3**	1.96x	**+0.7**	**1.89x**	**+0.2**	**1.92x**	**+0.5**	**2.09x**
Test set										
ALBERT	92.3	1.00x	84.2	1.00x	72.1	1.00x	90.9	1.00x	80.1	1.00x
ZTW	-0.4	1.61x	-0.5	1.52x	+0.1	1.64x	-0.1	1.59x	-0.5	1.81x
LeeBERT	-0.5	1.79x	-0.9	1.88x	0.0	1.68x	-0.4	1.72x	-0.3	1.86x
PALBERT	-0.3	1.49x	-1.1	1.72x	+0.2	1.27x	-0.4	1.51x	-0.3	1.50x
JEI-DNN	-0.1	1.35x	-0.7	1.59x	0.0	1.36x	-0.2	1.39x	0.0	1.47x
BEEM-C	-0.2	**1.98x**	-0.4	1.95x	+0.1	1.74x	+0.1	1.81x	+0.1	**1.97x**
BEEM-A	**+0.4**	1.91x	**-0.3**	**2.06x**	**+0.6**	**1.77x**	**+0.5**	**1.88x**	**+0.2**	1.95x

Table 1: Main results: This table compares BEEM against all the state-of-the-art early exiting baselines. We report the accuracy (Acc in %) and Speed-up (Speed).

Data	RTE		CoLA		QQP	
	Acc	Spd	Acc	Spd	Acc	Spd
AB-L	80.5	1.00x	60.9	1.00x	91.1	1.00x
Our-A	+1.8	2.04x	+1.3	2.85x	+0.1	3.33x
B-L	70.9	1.00x	64.3	1.00x	91.2	1.00x
Our-A	+0.5	1.81x	+0.9	1.71x	+0.3	2.51x

Table 4: This table provides results on the large variants of (AL)BERT models compared with BEEM-A. AB-L is ALBERT-Large and B-L is BERT-Large.

Conclusion & Future work

- Conclusion:
  - We proposed a method that gives better results than final classifier.
  - It leverages the ensemble learning principles.
  - It also provides a method to set the thresholds based on error rates.
  - Results on GLUE tasks demonstrate its effectives.
- Future works:
  - By the method proposed, it can be used for adapting across domains.
  - Also, we can have small separate model to learn the weights of the exits but it can make the model complex.