Zipformer Inspired Architectures, Parallel Downsampling and Ensemble Models for ASR

Lasbordes Maxence

FBK - September 2024

Abstract

This document presents the results from my 3 months summer internship at FBK (Fondazione Bruno Kessler) in the Automatic Speech Recognition (ASR) department. My work focused on the Early Conformer model from the Early Exit Transformer Model for ASR/SLU by FBK. This research work involves implementing, testing, and comparing new architectures for the Early Conformer model, including various downsampling configurations and factors, with a particular focus on the Zipformer architecture from the recently published Zipformer Paper [1] *A faster and better encoder for Automatic Speech Recognition, 10 Apr 2024*. Zipformer has achieved state-of-the-art results for ASR models ; in fact, its results are comparable to those reported in the Conformer paper on the LibriSpeech dataset, while being faster during training and accelerating inference. The goal is to enhance the existing Early Exit architecture from FBK by incorporating features suggested by the Zipformer architecture.

1 Introduction

This work is divided into two main configurations : the Early Conformer model with a single-exit and with multiple-exits. A small section will be dedicated to the experiments and research conducted on ensemble models for ASR, aiming to enhance model performance by combining them through ensemble techniques. The models are trained and compared using the open-source dataset LibriSpeech and its 1000 hours of transcribed audio files. Inference is done on test-clean and test-other from LibriSpeech. If not specified, the hyperparameters are set to default. The code and my contributions are available on GitHub : Early-Exit Transformer Model by FBK.

2 Models

2.1 Configuration with a Single-Exit

2.1.1 Early Conformer Baseline

The baseline for this configuration is the Early Conformer model with one exit. Since it uses a single exit it doesn't benefit from the Early Exit architecture. Each Conformer block consists of a single Conformer layer from PyTorch, and the Conv-1D block comprises two one-dimensional convolutional layers that downsample the sequence to 25 Hz. Some components, such as positional encoding and log-softmax, are not included in the diagrams I created, as our primary focus is on the arrangement of the various Conformer blocks.

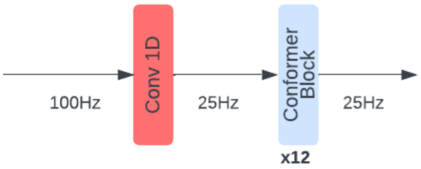

Figure 1
Early Conformer Model with 12 layers and 1 exit (Baseline)

2.1.2 Zipformer Architecture

The Zipformer Architecture revolves around using multiple downsampling stacks to lower the sequence to various frames rates. It uses more aggressive downsampling ratios in the middle encoder stacks which have more encoders. The sequence is also processed at 50Hz instead of 25Hz (for the Baseline) between the stacks which allow for more detailed feature extraction across different stages of the network.

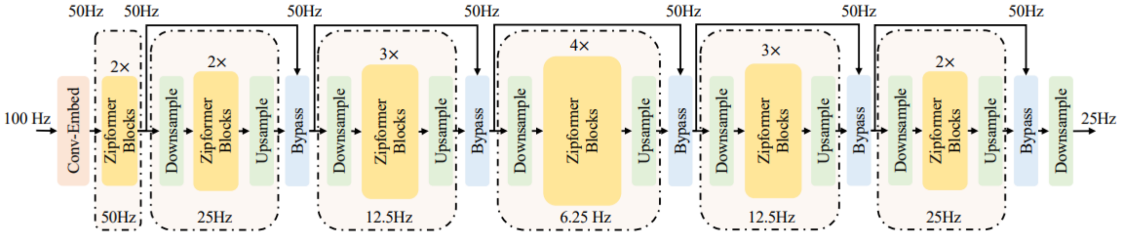

Figure 2
Zipformer-M Architecture from [1]

In the downsampled stacks, the pairwise Downsample and Upsample modules perform symmetric scaling down and scaling up in sequence length. We will experiment with various downsampling methods, such as selecting every other element or using a linear projection, but also with various types of Residual connections (Bypass). Zipformer has three different scales, each with a different number of layers and stack sizes : S, M, L.

Table 1 : Configuration of Zipformers.

Scale	Stack
S	{2,2,2,2,2,2}
M	{2,2,3,4,3,2}
L	{2,2,4,5,4,2}

In our case, we will implement and experiment with the Zipformer architecture on different scales using Conformer blocks instead of Zipformer blocks. Zipformer blocks are multiple Zipformer layers which have several differences compared to Conformer layers, such as having twice the number of parameters, different activation functions, layer normalization, and embedding dimensions.

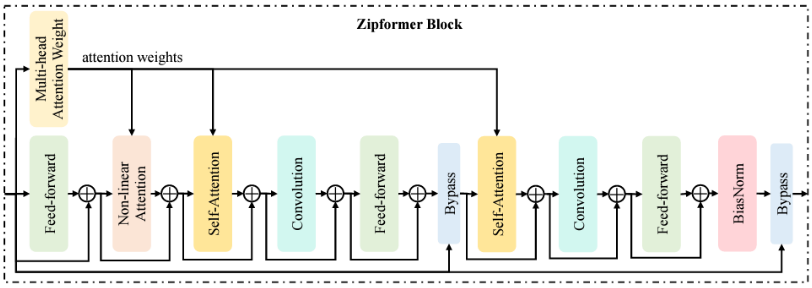

FIGURE 3
Zipformer Block

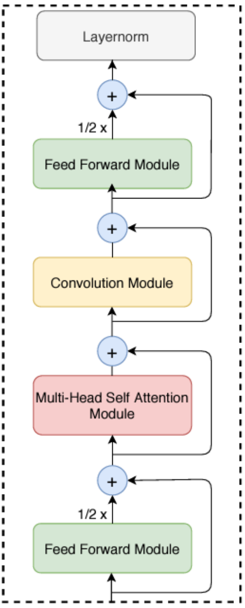

FIGURE 4
Conformer Block

2.2 Configuration with Multiple-Exit

2.2.1 Early Conformer Baseline

The baseline for this configuration is the Early Conformer model with six exits. The blocks are the same as previously mentioned. In this case, the CTC loss is the sum of the losses from each exit, so we optimize all six exits simultaneously, which is a key feature of the Early Exit architecture.

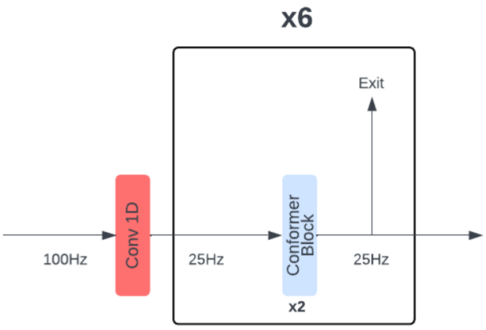

FIGURE 5
Early Conformer Model with 12 layers and 6 exits (Baseline)

2.2.2 Parallel Downsampling

We will experiment with various Parallel Downsampling architectures on the Early Conformer with 6 exits to improve the model. The concept of incorporating parallel downsampled layers is not new. Downsampled layers help the model focus on key features while reducing redundancy. By capturing information at varying levels of detail, the model gains a deeper understanding of the input sequence and can make more accurate decisions. These layers act as feature extractors, capturing different temporal information at multiple scales. Additionally, this approach reduces computational overhead since the sequence is downsampled. Here are some examples of architectures I considered and chose to experiment with.

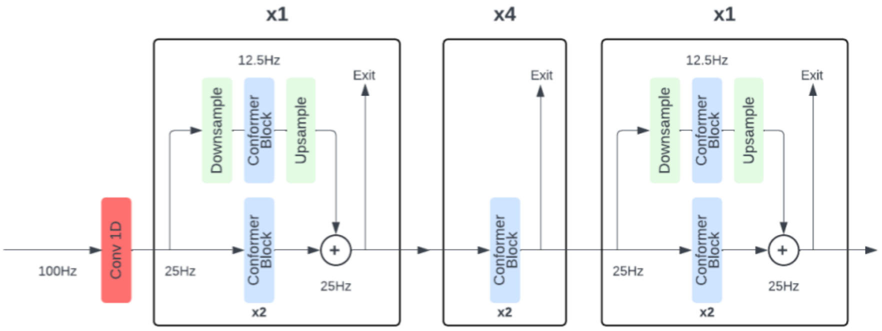

FIGURE 6
Early Conformer Model with 2 Parallel Downsampled Layers by a Factor of 2 (PD12-2-D2)

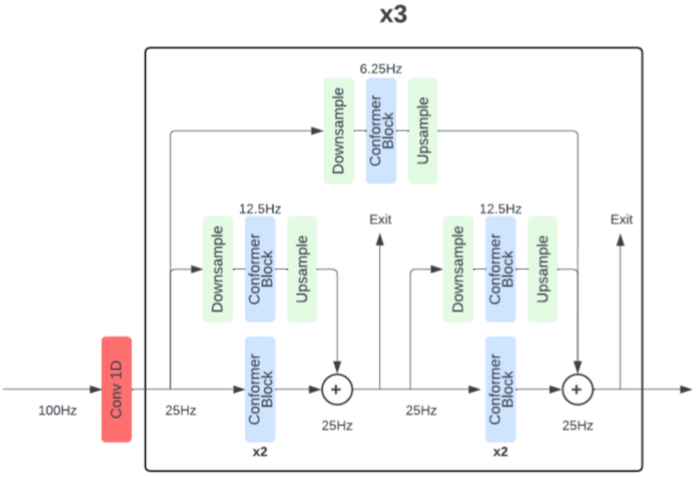

FIGURE 7
Early Conformer Model with 6 PD Layers by a Factor of 2 and 3 by a Factor of 4 (PD12-6-3-D2/4)

Architecture Names Description :

Early-Conformer : 12 normal layers (Baseline).
PD12-12-D2 : Parallel Downsampling Model with 12 normal layers and 12 parallel layers downsampled by 2.
PD12-6-D2 : Parallel Downsampling Model with 12 normal layers and 6 parallel layers downsampled by 2.
PD12-6-3-D2/4 : Parallel Downsampling Model with 12 normal layers, 6 parallel layers downsampled by 2 and 3 parallel layers downsampled by 4.
PD12-3-D2 : Parallel Downsampling Model with 12 normal layers, 3 parallel layers downsampled by 2 located every.

PD12-2-D2 : Parallel Downsampling Model with 12 normal layers, 2 parallel layers downsampled by 2 (located in parallel with the first 2 and last 2 normal layers).
PD12-12-Stack : 12 normal layers in stacks {1,2,3,3,2,1}, 12 parallel layers with factors {2,4,8,8,4,2}.

3 Experiments with Single-Exit Architectures

3.1 Experiments on LibriSpeech 100h

In this configuration, the models were trained on the same number of epochs, but some architectures, like the Zipformer, train much faster thanks to the downsampling stacks, compared to the classic Early-Conformer or the parallel downsampling models. This should be taken into consideration, as they could achieve even better performance with the same amount of training time. To simplify, Zipformer-S/M/L correspond to the Early Conformer with the Zipformer architecture, which uses Conformer blocks.

Table 2 : WERs of the models on Librispeech train-clean-100 with CTC loss. All models have been trained with a single exit. "Layer" : Total number of layers in each model (1 layer = 1 conformer block). "EOE" : Every Other Element as the downsample method. "LP" : Linear Projection as the downsample method.

	Type	test-clean (%)	test-other (%)	Layers	Residual Connection	Downsample
Early-Conformer	CTC	16.1	41.6	12	-	-
PD16-8-D2	CTC	17.1	42.7	24	Sum	EOE
Zipformer-S	CTC	16.2	41.9	12	Sum	EOE
Zipformer-M	CTC	**15.0**	40.7	16	Sum	EOE
Zipformer-M	CTC	16.4	41.4	16	Linear	EOE
Zipformer-M	CTC	17.0	43.4	16	Sum	LP
Zipformer-L	CTC	15.9	41.8	19	Sum	EOE

3.2 Experiments on LibriSpeech 1000h

Table 3 : WERs (%) of the models on Librispeech train-clean-100/360/500 with CTC loss. All models were trained on 50 Epochs.

	Type	test-clean (%)	test-other (%)	Layers	Residual Connection	Downsample
Early-Conformer	CTC	6.0	17.3	12	-	-
Zipformer-S	CTC	6.5	18.3	12	Sum	EOE
Zipformer-M	CTC	5.4	16.1	16	Sum	EOE
Zipformer-L	CTC	**5.0**	**14.7**	19	Sum	EOE

Zipformer-L demonstrates a significant advantage in inference time, being more than twice as fast as the Early-Conformer model. In addition to this impressive speed increase, Zipformer-L also achieves better performance metrics, despite having only slightly more layers and parameters. This combination of speed and performance highlights the effectiveness of the Zipformer architecture.

Table 4 : Table representing the **Number of Parameters** and **Inference time** of different models on the LibriSpeech (test-clean & test-other) benchmarks. The GPU used is an NVIDIA A40.

	Early-Conformer	Zipformer-S	Zipformer-M	Zipformer-L
Parameters (M)	31.2	31.0	41.3	49.0
Time (seconds)	1394.50	590.48	620.45	652.83
Layers	12	12	16	19

4 Experiments with Multiple-Exit Architectures

4.1 Results on LibriSpeech 100h

In this configuration, the models have 6 exits and the CTC loss is the sum of the losses from each exit, so we optimize the models using all exits, not just the last one.

Table 5 : WERs (%) of the models on Librispeech train-clean-100 with CTC loss. All models have been trained with 6 exits over 60 Epochs. "Layer" : Total number of layers in each model. "LP" : Linear Projection as the downsample method.

	Type	test-clean (%)	test-other (%)	Layers	Residual Connection	Downsample
Early-Conformer	CTC	16.6	42.8	12	-	-
PD12-2-D2	CTC	**13.6**	38.6	14	Sum	EOE
PD12-2-D2	CTC	13.7	**38.4**	14	Linear	LP
PD12-3-D2	CTC	13.8	38.8	15	Sum	LP
PD12-6-D2	CTC	14.1	39.1	18	Sum	LP
PD12-6-D2	CTC	14.8	40.3	18	Linear	LP
PD12-6-3-D2/4	CTC	21.5	49.0	21	Sum	LP
PD12-12-Stack	CTC	13.8	38.7	24	Sum	LP
PD12-12-D2	CTC	14.7	40.0	24	Sum	LP

4.2 Results on LibriSpeech 1000h

Table 6 : WERs (%) of the models on Librispeech train-clean-100/360/500 with CTC loss. All models were trained on 70 Epochs.

	Type	test-clean (%)	test-other (%)	Layers	Residual Connection	Downsample
Early-Conformer	CTC	5.6	16.1	12	-	-
PD12-2-D2	CTC	5.1	**15.0**	14	Linear	LP
PD12-2-D2	CTC	**4.9**	15.1	14	Sum	EOE

These results show that adding just two downsampled parallel layers—one before the first exit and the other before the last exit—reduces the model's WER with minimal added parameters and computational time. P12-2-D2 achieves a 0.7% lower WER compared to the baseline after 70 epochs on 1,000 hours of training, with only a minimal increase in parameters. Additionally, using EOE and a sum for the residual connection (or bypass) seem to further reduce the WER.

5 Ensemble Models

In this section, we aim to enhance model performance by combining them through ensemble techniques. Our goal is to leverage the individual strengths of each model to achieve improved accuracy and robustness in transcription. We employ a linear layer to integrate the outputs of two ASR models (only two for now), which should allow for optimal weighting and combination of their predictions to boost overall recognition performance. The linear layer utilizes distinct weights for each token.

5.1 Proof of concept

We aim to determine whether this approach is effective by experimenting with models trained on a small dataset. Specifically, we will use two Early Conformer models, each with 6 exits and 12 total layers, trained on the Tedlium and Voxpopuli datasets. Our ensemble model employs a single linear layer with 131,328 trainable parameters, using distinct weights for each token. We only train this linear layer while keeping the pre-trained models' weights frozen during training.

Table 7 : WERs (%) of the models of the ensemble model compared to the pre-trained models used in it.

	Dataset	test-clean (%)	test-other (%)
Ensemble Model	Dev-clean	18.98	38.44
Model 1 - Early-Conformer	Tedlium	16.19	33.68
Model 2 - Early-Conformer	Voxpopuli	29.27	48.74

Unfortunately, it seems that the ensemble model does not perform as expected and the results are not even as good as those of the best of the two individual models. This might be explained by the fact that one of the models is significantly worse than the other. To investigate further, we could try using models with more similar performance. Additionally, the results could be influenced by the fact that the models used are the same, with only the dataset changing ; using two very different models might yield better performance.

6 Conclusions

In this document, we have seen how the Zipformer architecture outperforms the Early Conformer structure with standard successive conformers for a single exit. It presents a very promising architecture that offers significant advantages, such as increased speed in both training and inference, as stated in the paper. Additionally, adding parallel downsampled layers before the first and last exits has shown to slightly improve the performance of the Early Exit configuration with multiple exits, with minimal additional parameters. This is because the residual connection and downsampling use the simplest methods and do not require trained weights to be effective. Indeed, no matter the number of exits, using downsampled layers with EOE as the downsampling method and simple sum for the residual connection consistently achieves comparable or slightly better results than using learnable weights.

6.1 Going further

Using features like the SwooshL/SwooshR activation functions, higher embedding dimensions in the middle stacks (for the Zipformers), different convolution layers or an other optimizer like ScaledAdam instead of Adam, could improve training efficiency and performance, potentially enhancing our model as suggested in the Zipformer paper.

7 Appendix

7.1 References

[1] Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, Daniel Povey, Zipformer : A faster and better encoder for Automatic Speech Recognition, 10 Apr 2024.

[2] George August Wright, Umberto Cappellazzo, Salah Zaiem, Desh Raj, Lucas Ondel Yang, Daniele Falavigna, Mohamed Nabih Ali, Alessio Brutti, Training Early-Exit Architectures for Automatic Speech Recognition : Fine-Tuning Pre-Trained Models or Training from Scratch, 22 Feb 2024.

[3] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, Ruoming Pang, Conformer : Convolution-augmented Transformer for Speech Recognition, 16 May 2020.

[4] Kiran Praveen, Abhishek Pandey, Deepak Kumar, Shakti Prasad Rath, Sandip Shriram Bapat, Dynamically Weighted Ensemble Models For Automatic Speech Recognition, 25 March, 2021.