CS 182 Final Project Commentary

Jerry Li, Kanu Grover, Samarth Goel, Aniruddh Chennapragada

Introduction

In this project, we designed an assignment based on the paper "Conformer: Convolution-augmented Transformer for Speech Recognition" to facilitate students' understanding of the Conformer neural network model in the context of Automatic Speech Recognition (ASR). Students are tasked with implementing every major component of the Conformer, constructing the architecture from scratch and applying it to both the LibriSpeech Benchmark Dataset and the AN4 Dataset. By comparing the model's performance and design choices to alternative solutions, students gain a deeper comprehension of the employed concepts and their practical applications.

The Conformer

The Conformer is a hybrid neural network model that combines the strengths of Transformer-style and Convolution-based architectures, excelling in capturing both long-range interactions and local properties within input sequences. Transformers employ a distinct self-attention mechanism that effectively captures crucial global interactions, while Convolutional Layers use kernels and convolutions to incorporate meaningful local context. By integrating features of both architectures, the Conformer achieves a remarkable performance improvement, surpassing earlier models such as RNNs and ContextNet. Performance is assessed using the WER (word error rate) metric on the LibriSpeech benchmark, a widely adopted speech data corpus. Notably, Conformer models with a fewer number of parameters than earlier, state-of-the-art models consistently outperform their more complex competitors. The Conformer architecture's core lies in the Conformer Block, which consists of a Multi-Headed Self-Attention Module, Convolution Module, and Feed Forward layers.

Pedagogy

Overall, we set up our approach in the assignment to mirror the introduction of concepts done by the authors of the original paper. We believe this approach is ideal for introducing the fundamental concepts that are most unique and important to the model first, then adding on blocks that the student may be more familiar with or that may not be as consequential for the Conformer Model specifically.

The assignment is intentionally designed to enable students to implement the modules of the Conformer architecture independently, fostering a comprehensive understanding of each element in the model. We deconstruct the Conformer into its distinct components, establishing separate classes for embedding, encoding, activation functions, attention, convolution, and the model as a whole. This modular structure allows students to visualize the Conformer's building blocks and their interconnections. As many of these modules cover concepts previously addressed in class, we provide substantial starter code to prevent students from being overwhelmed by implementation details.

To supplement code implementation, we offer guidance through hints and questions that encourage students to delve into the architecture and comprehend the authors' choices. For instance, the Feed Forward Layer utilizes the Swish activation function. Students first derive its gradient, analyze its properties, and then implement it. Subsequently, they explore the usage of various activation functions and compare their performance to standard activations like GeLU and ReLU. Another key area is encodings, specifically Relative Multi-Headed Self Attention, which allows the Conformer to model long-range sequences. This capability is partially enabled by positional encodings, which crucially introduce a time component to Transformers, lacking an intrinsic "state" like RNNs. To grasp this concept, students implement sinusoidal positional encoding from scratch, gaining insight into the effectiveness of this encoding scheme. They then proceed to implement the high-level Transformer architecture, including the interface between Dropout, Layer Normalization, and a pre-completed Multi-Headed Attention module, while omitting the self-attention components they had previously implemented in another assignment.

Subsequently, students implement the Convolution Layer, the Conformer Block's final component. We emphasize the connection between Pointwise and Depthwise convolution by illustrating how one is a particular case of the other, and relate these to the Depthwise-separable convolution, which students have encountered earlier in the course. We also highlight the significance of understanding the relatively low computational requirements of the Conformer convolution operation by having students tackle a parameter calculation problem.

In the final stage, students train the model and assess its performance. To showcase the Conformer's capabilities, they train various models on the LibriSpeech Benchmark dataset, comparing their Conformer Model against an imported Transformer model. Due to dataset size and model complexity, we have chosen training to end after only a few epochs. Although the Transformer model outperforms the Conformer model in this limited example, this outcome results from resource constraints rather than model efficacy. Nevertheless, students test their model on this dataset to maintain the assignment's focus on understanding the original paper.

After the LibriSpeech dataset experiment, we deviate from the paper by introducing the AN4 dataset, aiming to demonstrate the Conformer model's superiority over other models. We replicate the LibriSpeech dataset experiment, comparing the Conformer model against a Convolutional model. With the smaller dataset, the Conformer trains more rapidly, enabling students to observe its superior performance. Lastly, we provide an optional section where students retrain the models with data augmentations, illustrating a real-world training approach. This method equips students with the knowledge to apply their learnings to practical problems and appreciate the Conformer's utility.

Conclusion

Successfully completing this assignment requires students to grasp the intricacies of the Conformer model and the rationale behind the authors' design choices. By constructing the model primarily from scratch and training it on multiple datasets, students gain a comprehensive understanding of how each concept collaborates to enhance performance on a widely recognized AI benchmark. This hands-on experience empowers them to tackle new innovations in the field with a broader perspective, surpassing the insights gained by merely reading the paper.