Theophilus Pedapolu
Kevin Wang
Timber Lin

Conformer: Convolution-augmented Transformer for Speech Recognition
Homework Commentary

Convolutional Layers

Convolutions are an important component of ASR (automated speech recognition) and have been used conventionally for deep learning tasks involving audio data, so it is important that students understand how the convolutional layers work in the specifics of the Conformer architecture. The paper uses two non-standard types of convolution in addition to the regular convolution operation in the Conformer encoder. The 1D depthwise and pointwise convolutions are particularly important because they form the backbone of the "convolution module", which is the novel idea the paper introduces. The reason the authors use these depthwise-separable convolutions is that they have been shown to reduce the number of parameters and computational complexity while increasing the representational efficiency of neural networks, which is the key concept we wanted students to understand.

Question 1 begins by defining depthwise and pointwise convolutions and how they differ from the standard convolution procedure. Here we emphasize that depthwise convolution kernels act on individual channels and pointwise convolution has 1x1 kernels and, together, they essentially split the standard convolution operation into two parts. In the problems, we have students mechanically perform depthwise and pointwise convolution on a 3x3x3 input tensor to get a feel for how they work. Then, they are asked to compute the number of parameters and multiplications performed in depthwise-separable convolution vs. standard convolution. This engages students with our key concept because they get to see the space and computational efficiency of depthwise-separable convolutions, hence understanding why they are used in place of standard convolutions.

The coding portion of the homework (Question 3) also contains subparts where students have to implement depthwise and pointwise convolutions and combine them structurally by writing the convolution model pipeline. In particular they are asked to consider the practical implementations of these two convolutional layers in the context of the input data and the task the model is training on. While students should be well familiar with 2D convolutions, audio data should be processed by the 1D variant, since instead of width and height, the data only has a time dimension.

Theophilus Pedapolu
Kevin Wang
Timber Lin

Conformer Architecture

The key innovation of the paper is its architecture, which combines two key types of neural networks: convolutional and self-attention. They are both very powerful but take wildly different approaches that could both be used for the task of speech recognition. CNNs attempt to find patterns among spatially-similar values, in this case audio data within a time range, in order to extract higher level features from the data as a whole. Transformers on the other hand, are a powerful means of finding relationships between values across an entire piece of data.

Through the coding portion in question 3, the assignment attempts to teach students how the paper combines these two models to take advantage of their respective capabilities. Students will implement parts of the architecture, as well as put them together into higher-level modules which they then use to create the Conformer block. Through this, students will learn how the convolution and self-attention modules interact with each other. Furthermore, in question 4, we train the Conformer model on a small dataset containing speech commands to let students interact with the model and examine how different parameters influence training. This includes formatting the dataset and testing various ablation studies to see the contribution of the convolution and self-attention modules.

An additional goal of the coding portion is to give students practice on using some of the deep learning building blocks they learned throughout the course (ex: ResNets, activation functions, linear layers, dropout, normalization) as abstractions in PyTorch to create novel modules and architectures.

Activation Functions

The paper emphasizes the use of the swish activation function, citing higher performance and faster convergence in the conformer model. The swish activation function is presented as a blend of the ReLU activation and sigmoid activations, as both have desirable properties. The conceptual subproblems on Question 2 involve the students' understanding of the benefits and drawbacks of these various activation functions. The mechanical subproblem requires the students to take the derivative of the swish activation function, and utilize their understanding of the vanishing gradient problem to analyze how the swish activation is preferable to the traditional sigmoid activation in avoiding this issue. Students are also guided towards drawing similarities between the self-gating behavior in the swish activation and the LSTM cell's input modulation gate.

Furthermore, the homework asks students to explore the optional tunable beta parameter of the swish activation. Using the derivative of this more general form of swish, students will analyze the asymptotic behavior of this activation function and comment on how certain values of beta cause swish to resemble a ReLU activation or a linear layer.

Finally, students are asked to implement the Swish activation function in code, as well as GLU (Gated Linear Unit) activation, which they are given a conceptual explanation of in the Colab notebook. Implementing each of these functions gives them practice converting one concept they should be very familiar with, plus another concept they were just introduced to, both into working code.

Analysis/Ablation Studies

The last section of the paper describes the ablation studies the authors ran to study the effects of various layers and parameters in the Conformer architecture and how each contributes to the accuracy improvement. The key takeaway we wanted students to understand here is that the convolution and self-attention modules work in tandem to give the Conformer its representational power and that various parameters can be modified to increase the accuracy of the model.

The second part of question 4 asks students to run the code for two ablation studies: varying the number of attention heads and exchanging the order of the convolution and self-attention modules. The goal is for students to understand that increasing the number of heads generally increases accuracy since the model captures more dependencies and having the convolutional model first leads to slightly worse performance. These “mini” ablation studies give students intuition as to how the Conformer model works because they see the effects of tampering with parts of the Conformer Block pipeline. The conceptual questions guide their reasoning toward the key takeaway, most importantly how the convolution and attention modules work together.