Ruchira Ray EID:rr52486

Paper Response: HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units

There are typically three obstacles to overcome when attempting to apply BERT or other NLP models to speech: there are many different acoustic units in the input speech expression, no lexicon of individual phonemes, and the length and division between sound units vary widely. Hidden-Unit BERT (HuBERT) is a novel method for self-supervised learning of speech representations, which utilizes an offline unsupervised clustering step to provide aligned target labels for a BERT-like prediction loss and thus, can be used to model the abundance of lexical and non-lexical information in audio. The main features of this method are the use of an offline unsupervised clustering step to generate aligned target labels and the application of the prediction loss only to the masked regions, which forces the model to learn a combined acoustic and language model over the continuous inputs. The HuBERT model, which begins with a simple k-means teacher of 100 clusters and employs two iterations of clustering, achieves results on Librispeech that are at par with or better than those achieved by the current SOTA. This paper proposes employing acoustic unit discovery models to produce frame-level targets. During training, the process alternates between a clustering step, in which pseudo-targets are generated, and a prediction step, in which the model attempts to guess these targets at masked positions. The first training step is to find the hidden units, which is accomplished by extracting MFCCs from the audio waveform. The K-means clustering algorithm is used to classify these audio feature vectors into one of the K clusters. Once the clusters are identified, the audio frames are tagged with the corresponding cluster and each identified cluster then becomes a hidden unit. Each hidden unit is then mapped to its corresponding embedding vector which can be used during the second step to make predictions. Representations from an intermediate layer of the HUBERT transformer encoder (from the previous iteration) are reused in subsequent clustering steps. The next phase, which employs masked language modelling, is very similar training of the original BERT model. At first, the convolutional waveform encoder takes the raw audio and generates features, which are then masked randomly before being fed into the BERT encoder. A feature sequence is generated by the BERT encoder (Context Net), which fills in the unmasked tokens. This output is then projected into a lower dimension to match the labels. The cosine similarity between these outputs and hidden unit embeddings generated in the first clustering step is then calculated. The logits are then penalized for inaccurate predictions using the cross-entropy loss. We use the connectionist temporal classification (CTC) loss for ASR fine-tuning following HuBERT pre-training, with the exception of the convolutional audio encoder, whose weights are not adjusted.

The primary question the authors seek to address is whether it is possible to substitute a speech sequence for the text input in BERT, mask a portion of the sequence, and then train the model to recover the masked portion for better speech understanding. The paper introduces HuBERT, a method for learning speech representations that makes use of K-means clustering to classify segments of continuous input that have been masked. The core idea behind HuBERT is to find hidden units that can be used to format speech data into a more language-like format. These unseen components are of equal importance as tokens or words in a string of text. By modelling speech as a series of tokens, we can attempt to use the same robust models for speech recognition. Iteratively refining K Means cluster assignments with previously learned latent representations leads to a significant increase in the quality of the learned representation. Finally, on the test-other subset, HuBERT shows a relative reduction in WER of up to 13% when scaling to a 1B-transformer model. We notice that wav2vec 2.0 and HuBERT seem to share a lot of similarities at first glance. HuBERT's performance, when tuned for automatic speech recognition, either matches or improves upon wav2vec 2.0, even though their training processes are very different. HuBERT uses a simpler cross-entropy loss than a complicated combination of contrastive loss & diversity loss. HuBERT also builds targets through a separate clustering process, while wav2vec 2.0 learns its targets simultaneously through a quantization process using Gumbel-softmax. HuBERT re-uses embeddings from the BERT encoder to improve targets, while wav2vec 2.0 only uses the output of the convolutional network for quantization. The paper answers their primary with enough evidence. The paper provides extensive experimentation with low- and high-resource setups. The authors also play around with different numbers of clusters to capture targets of varying granularity. They prove the marginal gains that can be made by utilizing cluster ensembles. Further, it would be interesting to see results on the model's speed, complexity and environmental impact. An interesting experiment would be to use features other than MFCC for the clustering step like those in wave2vec. The authors spoke about using features from an intermediate layer of the encoder from the previous iteration for the clustering steps, however, which layer was being used wasn't very clear and wasn't justified. This paper opens up more opportunities for self-supervised learning of speech. The learned representations in this paper can also be very useful for generation tasks in addition to their use in automatic speech recognition.