Speech Transformers

1. Wav2Vec

Wav2Vec is a speech encoder model developed by Facebook AI. I was developed for the purpose of various speech related tasks like automatic speech recognition or emotion recognition. Since the release of its original version the model has been upgraded to use the transformers architecture similar to the BERT and GPT models and has also been retrained on 12 different languages in a multilingual version. In general speech recognition models require large amounts of transcribed audio data to attain good performance, but in recent developments pretraining of neural networks has emerged as an effective technique for settings where labelled data is scarce. The idea is to learn general representations where substantial amounts of labelled or unlabelled data is available and to leverage these learned representations to improve performance on a downstream task for which the amount of data is limited. In wav2vec unsupervised pretraining is used to improve supervised speech recognition. Learning these representations through unsupervised pretraining is also referred to as self-supervised learning of representations. Given an audio signal as input, wav2vec optimizes the model to predict future samples from a given signal context. The model takes raw audio signal as input and then applies two networks. Both the networks are convolutional networks.

- Encoder Network: Reduces the dimensionality of the speech data, by encoding 30 ms of audio into 512-dimensional feature vector zt at each timestep t, every 10ms.
- Context Network: Takes encoder output features as input, encoding 210 ms of raw audio into another 512-dimensional feature vector ct. The objective is to aggregate information over a longer timeframe to model higher order information. This network outputs contextual representations ct that are used to predict future audio samples.

The layers in both the encoder and context network consist of a causal convolution with 512 channels, a group normalization layer and ReLU non linearity. The training methodology behind wav2vec is called contrastive predictive coding. Instead of predicting the future samples of the audio signal directly wav2vec uses a contrastive loss that implicitly models the mutual information between the context and future audio samples. Wav2vec’s loss has been designed to implicitly maximize the mutual information between the context features and future audio samples. Doing so will pushes the model to learn contextual features which contains higher order information about the audio signal.

Model Training:

The model is trained to distinguish the true future audio samples from fake distractor samples by using the context vector. Here audio sample mean the encoder feature vector z. This is not applied to the waveform directly as the dimensionality would be too high. This prediction process is done k times at each timestep t.

At a given timestep t, for each step k=1….12, we do the following:

a. Extract the true audio sample (encoder representation) at future step k
b. Pick 10 random negative samples z 1….10 from the same audio sequence
c. Compute a step k specific transformation h(k) of the context vector at time t
d. Compute the similarity (dot product) of the transformed context vector with all z candidates
e. Compute the final probabilities of positive/negative through a sigmoid activation function
f. Compare the prediction with the ground truth and penalize the model for wrong predictions using binary cross entropy loss.

2. FAIRSEQ Speech2Text

FAIRSEQ is an extension for speech to text modelling tasks such as end to end speech recognition and speech to text translation. Fairseq S2T adds attention based RNN models, Transformer models as well as the latest Conformer models for Automatic Speech Recognition (ASR) and Speech Translation (ST). It also supports Connectionist Temporal Cost (CTC) criterion. It extracts Kaldi-compliant speech features like log mel-filter banks automatically from wav/flac audio files via PyKaldi or torchaudio. Unlike wav2vec which has been trained and evaluated on multiple languages Fairseq S2T is trained and evaluated on English ASR benchmark. For speech inputs it extracts 80 channel log filter bank features (25ms window size and 10ms shift). In this model training samples with more than 3000 frames have been removed during the data preparation stage for GPU memory efficiency. To alleviate overfitting, the speech translation model encoders are pretrained on English ASR and adopt SpecAugment. It uses the transformer based seq2seq (encoder-decoder) architecture for end to end Automatic Speech Recognition and Speech Translation. It uses a convolutional downsampler to reduce the length of speech inputs by 3/4th before they are fed into the encoder. The model is trained with standard autoregressive cross entropy loss and generates the transcripts and translations autoregressively.

Model Training:

Speech to Text modelling data consists of source speech features, target text and other optional information like source text, speaker id, etc. Fairseq S2T uses per dataset split to store these information. Unlike text token embeddings, speech features like log mel scale filter banks are usually fixed during model training and can be precomputed.

3. HuBERT

Self-supervised approaches for speech representation learning are challenged by three unique problems

- There are multiple sound units in each input utterance
- There is no lexicon of input sound units during the pre-training phase
- Sound units have variable length with no explicit segmentation

To deal with these problems HuBERT (Hidden-Unit BERT) approach for self-supervised speech representation learning has been proposes which utilizes an offline clustering step to provide aligned target labels for a BERT-like prediction loss. A key ingredient in this approach is to apply the prediction loss over masked regions only which forces the model to learn a combined acoustic and language model over the continuous inputs. HuBERT benefits from an offline clustering step to generate noisy labels for a BERT-like pretraining. BERT model consumes masked continuous speech features to predict pre-determined cluster assignments. The predictive loss is only applied over the masked regions, forcing the model to learn good high-level representations of unmasked inputs to infer the targets of masked ones correctly. Intuitively, the HuBERT model is forced to learn both acoustic and language models from continuous inputs. The HuBERT model first needs to model unmasked inputs into meaningful continuous latent representations which maps to the classical acoustic modelling problem. Then to reduce the prediction error the model needs to capture the long range temporal relations between learning representations. Consistency of targets is important for this model not just for the correctness but also to model the sequential structure of the input data. HuBERT draws its inspiration from the DeepCluster method for self-supervised visual learning however it benefits from the masked prediction loss over speech sequences to represent their sequential structure.

Model Training

When the HuBERT model is pretrained on either the standard Librispeech 960h or the Libri-Light 60k hours it either matches or improves upon the state of the art wav2vec 2.0 performance on all finetuning subsets of 10 mins, 1h, 10h, 100h, 960h. The results for the HuBERT model are presented systematically in three sizes.

- Base: 90M parameters
- Large: 300M parameters
- X-Large: 1B parameters

The X-Large model shows up to 19% and 13% relative WER (word error rate) improvement from LARGE models on dev-other and test-other evaluation subsets when pre-trained on the Libri-Light 60k hours.

4. SEW

SEQ (Squeezed and Efficient Wav2vec) is an efficient pretrained model architecture. SEW differs from wav2vec2 in the following aspects:

a) Using a squeezed context network
b) Replacing WFE-O with WFE-C
c) Reallocating computing across different components
d) Using MLP predictor heads with Batch Normalization

Conversational AI systems that use the SEW pre-trained models will better recognize what customers are saying, who is saying it, and how they feel and respond quicker. For various downstream models in automatic voice recognition, speaker identification, intent classification, emotion recognition, sentiment analysis, and named entity recognition, these pre-trained models open the door to cost reductions and/or performance benefits. A pre-trained model’s speed can be simply transmitted to downstream models. The fine-tuned downstream model is smaller and quicker because the pre-trained model is smaller and faster. These advances in efficiency lower the time spent training and fine-tuning and the actual delay noticed in products.

Model Training

Wav2Vec2, SEW, and SEW-D are pretrained on 960h LibriSpeech audios for 100K updates and fine-tuned on 100h labelled data. Without an LM, compared with the W2V2-tiny, SEW-tiny reduces the WER by 53.5% (22.8% to 10.6%) and 43.7% (41.1% to 23.7%) on test-clean and test-other, while being faster. With an LM, WER improves by 38.6% and 43.4% on test-clean and test-other. Compared with the W2V2-mid, SEW-mid reduces WER by 30.2% (9.6% to 6.7%) and 32.9% (22.2% to 14.9%) with similar inference times. SEW does incur slight increase in training time compared to W2V2 with similar inference times. However, SEW has lower WER even compared to a slower W2V2 which takes longer to train.