Introduction of Wav2Vec

General Introduction

Wav2Vec2 is a pretrained model for Automatic Speech Recognition (ASR) and was released in September 2020 by Hugging Face.

Wav2Vec2, is a convolutional neural network (CNN) that takes raw audio as input and computes a general representation that can be input to a speech recognition system. The objective is a contrastive loss that requires distinguishing a true future audio sample from negatives.

Principle

The core idea of Wav2Vec2 is to convert the speech signal into a series of fixed-length feature vectors, and then use these vectors to train a classifier to predict the text content represented by the speech signal.

Using a novel contrastive pretraining objective, Wav2Vec2 learns powerful speech representations from more than 50,000 hours of unlabeled speech. Similar to BERT's masked language modeling, the model learns contextualized speech representations by randomly masking feature vectors before passing them to a transformer network.

Features

- **End-to-end speech processing**: Wav2Vec is an end-to-end speech processing model that takes raw audio data as input without manually extracting acoustic features, thus simplifying the audio processing process .

- **Self-supervised learning**: Wav2Vec is trained using self-supervised learning methods. It trains itself on large amounts of unlabeled audio data, with the goal of learning to encode audio data into useful representations without explicit labels. This makes Wav2Vec perform well in unsupervised learning tasks.

- **Pre-training and fine-tuning**: Wav2Vec adopts pre-training and fine-tuning methods. First, pre-train on large-scale data, and then fine-tune the model to fit the specific application needs for a specific task, such as speech recognition or speech classification.

- **Multi-language support**: Wav2Vec has the capability of multi-language support and can be used to process audio data in multiple languages. This makes it very useful in cross-language applications and multilingual environments.

- **High performance**: Wav2Vec achieves excellent performance on multiple speech processing tasks, including speech recognition, speech classification, and speech sentiment analysis. Its representation learning capabilities enable it to capture important features in audio data.

- **Open Source Availability**: Wav2Vec is an open source project that provides pre-trained models and training code that can be easily used for various audio processing tasks. This makes it easy for researchers and developers to leverage the model.

Applications

Automatic speech recognition

Speech translation

Music information retrieval

Speech generation

Discussion 1

If wav2vec makes slight changes to the input audio, will the resulting vector change significantly?

During the training process of wav2vec, the input audio signal will be subject to small random perturbations, but these perturbations are limited and will not change the basic characteristics of the audio, such as pitch, rhythm, intonation, etc. These perturbations help the training model learn more robust feature representations and improve the model's robustness in noisy environments.

During the inference process, if the input audio signal is not severely disturbed by noise or deformed, then the vector representation obtained through wav2vec should be relatively stable and not change much. Of course, the resulting vector representation may also change when there is noise or distortion in the input signal. But in general, the vector representation obtained by wav2vec should be relatively robust and can effectively capture the important features of the audio signal.

Discussion 2

Why "the vector obtained by wav2vec is relatively robust”

First, the wav2vec model is trained based on a self-supervised learning method and does not require manual annotation of labels. During the training process, the wav2vec model not only learned the basic features of the audio signal, but also learned some higher-level features, such as the boundaries of speech units and the representation of phonemes. These feature representations are robust to audio signal changes and perturbations.

Secondly, wav2vec uses many techniques during the training process to improve the robustness of the model. For example, the model will make small random perturbations to the input audio signal, which allows the model to learn the characteristics of the signal more robustly. In addition, the wav2vec model also uses a contrastive learning method, which requires the model to distinguish the feature representations of positive samples and negative samples. This method also helps to improve the robustness of the model.

Finally, the wav2vec model is based on the Transformer structure, which has been proven to be very effective in many tasks in the field of natural language processing, such as language modeling, translation, and question answering. Therefore, using this structure can make the model more adaptable to the feature representation of audio signals and improve the robustness of the model.

Discussion 3

How to choose the duration of the voice signal?

The input length of the wav2vec model is usually fixed, so the input speech signal needs to be truncated or padded to make its length meet the requirements of the model. Generally speaking, shorter speech signals can be directly padded to the specified length, while longer speech signals need to be truncated.

On the other hand, the amount of information contained in the speech signal will also affect the selection of the duration of the speech signal. If the duration of the speech signal is too short, it may not contain enough information for identifying keywords, while if the duration is too long, it may contain too much irrelevant information, thus affecting the detection results. Therefore, it is necessary to select an appropriate speech signal duration while ensuring that the model input length is satisfied.

Discussion 4

Now there are two pieces of speech. The first piece of voice is a keyword. The second piece of voice contains not only all the content of the first piece of voice but also other content. Using wav2vec, can we detect that the second piece of voice contains the first piece of voice?

You can use the wav2vec model to detect speech keywords and detect whether the second speech contains the first speech. First, it is necessary to extract and encode features of the first speech to obtain its speech embedding vector (speech embedding). Then, feature extraction and encoding are performed on the second speech segment to obtain its speech embedding vector. Then, vector similarity measurement methods, such as cosine similarity, can be used to measure the similarity between the two speech embedding vectors. If the similarity is higher than a set threshold, the second speech segment can be considered to contain The first speech.

Part of the embedding vector of the second speech is similar to the embedding vector of the first speech. However, since the second speech also contains other content, another part of its embedding vector may not be similar to the embedding vector of the first speech. Therefore, when performing keyword detection, it is necessary to select an appropriate similarity measurement method and threshold according to the specific situation to ensure the accuracy of detection.

Feasibility of application in our project

Using wav2vec for speech keyword detection can be divided into the following steps:

1. Data preprocessing: For each speech signal, sampling rate, duration, etc. need to be processed so that it can be accepted by the wav2vec model.

2. Feature extraction: Use the wav2vec model to extract the feature vector of the speech signal. This process can be achieved through a pre-trained model. The wav2vec model can convert speech signals into a series of feature vectors. We can use wav2vec CTC model to collect word starting and ending time stamps.

3. Keyword detection model training: The extracted feature vectors and corresponding keyword labels are used to train a classification model, such as support vector machines, random forests, neural networks, etc. This classifier can predict whether the feature vector contains keywords.

4. Keyword detection application: In practical applications, each speech signal is input into the wav2vec model to extract feature vectors, and then these vectors are input into the keyword classifier for classification. The result output by the classifier indicates whether the speech signal contains keywords.

It should be noted that when training a keyword detection model, it is necessary to use enough annotated data for training to improve the accuracy of the model. So in our project, we use transfer learning method to initialize the keyword detection model using the pre-trained wav2vec model, thereby improving the training efficiency and accuracy of the model.