CS11-737 Multilingual NLP

Speech Pre-training

Lei Li

https://lileicc.github.io/course/11737mnlp23fa/
Feature Extraction for Speech Recognition

“Pittsburgh is a city of bridge”

Neural Network

MFCC

- need 1,000+ hours of transcribed data to train a good ASR system
- how to generalize to many languages/dialects?
Self-supervised Speech Representation Learning

- Self-supervised Training on unlabeled audio data
- generalize to many tasks (ASR, ST)
- generalize to language/dialect/domain

Pre-trained Neural Network
Transfer to Downstream Tasks

Fine-tuning

Task-specific network

Pre-trained Neural Network

ASR
Speech Translation

Fine-tuning

Task-specific network

Pre-trained Neural Network

Speech Translation

Fine-tuning

Task-specific network

Pre-trained Neural Network

Speech Translation
Wav2Vec / Wav2Vec 2.0

- **Architecture:**
  - CNN+Transformer

- **Training**
  - Masked prediction of quantized vector
  - Contrast true quantized latent with distractor latent embeddings
Wav2Vec2

Context C

Mask during training

Quantized Rep Q

latent rep Z

Raw wav X, each frame ~ 25ms, stride 20ms

How many layers of Convolution?

How to design each kernel size/stride?


Context C

Mask during training

Quantized Rep Q

latent rep Z

Raw wav X, each frame ~ 25ms, stride 20ms

waveform $x$ at 16kHz

Wav2Vec2

Context C

Mask during training

Quantized Rep Q

latent rep Z

Raw wav X, each frame ~ 25ms, stride 20ms

frame size=399 (25ms)
sampling rate=50Hz
(sliding 320=20ms)

waveform x at 16kHz

Discrete Quantization with Codebook

G groups of probability vector of size $V$

$G \times V$

one frame vector from CNN

(Gumbel) Softmax

Linear

Concat

pick max prob

codebook1

codebook2

codebook3

codebookG
How to obtain codebook — Product Quantization

Splitting a vector into equally sized chunks — subvectors, Assigning each of these subvectors to its nearest centroid
Contextual Encoder

- Transformer Encoder x12

- CNN+GELU replacing pos emb.
- Vector from CNN (before quantization)
Wav2Vec2.0: Contrastive on quantized acoustic state

Training data: (audio only)
LibriSpeech 960 hrs
LibriVox 53k hrs

Minimize contrastive loss
\[ L = - \sum \log \frac{\exp \text{Sim}(c_t, q_t)}{\sum \exp \text{Sim}(c_t, q_-)} + \text{penalty} \]

Bring closer masked context and quantized acoustic state

Training Loss

\[ \mathcal{L}_m = - \log \frac{\exp\left(\frac{\text{sim}(c_t, q_t)}{\kappa}\right)}{\sum_{\tilde{q} \sim Q_t} \exp\left(\frac{\text{sim}(c_t, \tilde{q})}{\kappa}\right)} \]

Codebook diversity penalty to encourage more codes to be used
- Sample starting points for masks without replacement, then expand to 10 frames
  - span can overlap
  - for a 15s sample, ~49% of frames masked with an avg span of 300ms
Model Setup

• Wav2vec2 base:
  ○ 12 Transformer layers, $d=768$, $d_{ffn}=3072$, $\#\text{heads}=8$
  ○ 16 groups
  ○ rel pos emb cnn kernel size 128

• Wav2vec2 large:
  ○ 24 Transformer layers, $d=1024$, $d_{ffn}=4096$, $\#\text{heads}=16$


Training

• LibriSpeech: 960 hours of English speech (just audio)

• LibriVox (LV-60k): about 53k hours of audio for book reading

• Wav2Vec2 base:
  ○ each sample is cropped with length 250k (=15.6s)
  ○ total batch size: 1.6 hours on 64 V100 GPUs

• Wav2Vec2 Large:
  ○ each sample is cropped with length 320k (=20s)
  ○ total batch size: 2.7 hours on 128 V100 GPUs.
Fine-tuning

• Add a single linear projection on top into target vocab and train with CTC loss with a low learning rate (CNN encoder is not trained).

• Use modified SpecAugment in latent space to prevent early overfitting

• Uses wav to letter generation with the official 4gram LM and Transformer LM
Wav2Vec2 Results

High resource
(Librispeech 960h labeled)

Word error rate
0.0 1.5 3.0 4.5

ContextNet (supervised)
Noisy Student (60k-h unlabeled)
wav2vec (60k-h unlabeled)

Low resource setup
(Librispeech 10min - 100h labeled)

Word error rate
0.0 2.8 5.5 8.3 11.0

Noisy Student 100h
wav2vec 100h
wav2vec 1h
wav2vec 10m
wav2vec 10m + (60k-h unlabeled)

10 min labeled data + 960h unlabeled
Effects of Model size and raw data

Effects of model size and amount of unlabeled data

Word error rate on test-other

Labeled data

Base (100m)
Large (300m)
+ 60k-h

10m 1h 10h 100h 960h
Overall ASR results

![Graph showing Word Error Rate for various ASR systems. The graph uses a bar chart to compare different models and their performance. The y-axis represents the Word Error Rate (WER) on test-other, with values ranging from 0 to 14. The x-axis lists different models, including Deep Speech 2 (Baidu 15), Fully Conv ASR (FB 19), tdnn / Kaldi (18), SpecAugment (Google, 19), RWTH Hybrid (19), Pseudo-labeling (FB 20), Conformer (Google 20), Noisy Student (Google 20), wav2vec 2.0 (FB 20), wav2vec 2.0 + Conf. + NST (Google, 2020), wav2vec 2.0 (FB, 20) + SelfTrain (FB, 20). The x-axis shows the amount of labeled data used, with 960h labeled and 10min labeled. The graph indicates that wav2vec 2.0 and wav2vec 2.0 + Conf. + NST have the lowest WER, while Deep Speech 2 has the highest. Data based on Papers with Code (25 Oct 2020).]
XLSR: Multilingual Wav2Vec2

Cross-lingual transfer

Multilingual fine-tuning

CommonVoice results:

Phoneme Error Rate (PER)

High-resource languages
Low-resource languages

XLSR-Mono
XLSR-10 (Base)
XLSR-10 (Large)

CommonVoice results:

Phoneme Error Rate (PER)

XLSR-Mono
XLSR-10
XLSR-10

Monolingual finetuning
Multilingual finetuning
Summary

- Self-supervised pre-training with audio data only
- Wav2Vec2 Model: CNN+Transformer
- construct the frames with reasonable size (25ms) and sliding (20ms)
  ○ proper design of CNNs
- Masked training with contrastive loss on quantized representation
Language in 10
TTS Code in Notebook

- [https://github.com/lileicc/FastSpeech2](https://github.com/lileicc/FastSpeech2)

- [https://www.cs.cmu.edu/~leili/course/11737mnlp23fa/code/tts/run_tactron2.ipynb](https://www.cs.cmu.edu/~leili/course/11737mnlp23fa/code/tts/run_tactron2.ipynb)