An Unsupervised Autoregressive Model for Speech Representation Learning

Yu-An Chung  Wei-Ning Hsu  Hao Tang  James Glass

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA

Interspeech
Graz, Austria
September 16, 2019
Why representation learning?

• **Speech signals are complicated**
  – Contain rich acoustic and linguistic properties (e.g., lexical, speaker characteristics)

• **High-level properties are important but poorly captured by surface features**
  – E.g., wave signals, log Mel spectrums, MFCCs
  – Require a large model to learn feature transformation from surface features
  – Need large amounts of paired audio and text for supervised learning

• **Representation learning: a two-steps procedure**
  1) Learn a transformation function $T(x)$ that transforms a surface feature $x$
     into a higher-level and more accessible form
  2) Use $T(x)$ as input to downstream model instead of $x$

• **Linear separability as accessibility**

Autoregressive Predictive Coding, Interspeech 2019
Why unsupervised learning of $T(x)$?

- **Unlabeled data are (much) cheaper**
  - Vision: one-time collection of large-scale labeled data may be okay
  - Language: infeasible to collect labeled data for all languages

- **Less likely to learn specialized representations; sometimes the target task is unknown**

- **Our goal of $T(x)$: Retain as much information about $x$, while making them more accessible for (possibly unknown) downstream usage**
Learning $T(x)$ via Autoregressive Predictive Coding (APC)

- Basic idea: Given previous frames up to the current one $(x_1, x_2, \ldots, x_i)$, APC tries to predict a future frame $x_{i+n}$ that is $n$ steps ahead
  - Use an autoregressive RNN to summarize history and produce new output
  - $n \geq 1$ encourages encoder to infer more global structures rather than exploiting local smoothness

- **Training**
  \[
  \arg\min_{\{RNN, W\}} \sum_{i=1}^{N-n} |x_{i+n} - y_i|,
  \]
  \[
  y_i = RNN(x_i) \cdot W
  \]

- **Feature extraction**
  Take RNN output of each time step:
  \[
  T_{APC}(x_i) = RNN(x_i) \forall i = 1, 2, \ldots, N
  \]
Comparing with Contrastive Predictive Coding (CPC)

• Architecture
  – APC is almost a pure RNN
  – CPC consists of a CNN as frame encoder and an RNN as context encoder

• Training objective
  – APC predicts a future frame \( x_{i+n} \) directly
  – CPC distinguishes \( x_{i+n} \) and a set of randomly sampled negative frames \( \{\bar{x}\} \)

• Learned \( T(x) \)
  – \( T_{\text{CPC}}(x) \) encodes information most discriminative between \( x_{i+n} \) and \( \{\bar{x}\} \)
    * E.g., \( \{\bar{x}\} \) sampled from same vs. different utterance as \( x_{i+n} \)
    * Better to know what downstream task is when choosing sampling strategy
  – \( T_{\text{APC}}(x) \) encodes information sufficient for predicting future frames, more likely to retain information about original signals

* Representation Learning with Contrastive Predictive Coding, Oord et al., 2018
Experiments

• LibriSpeech 360-hour subset (921 speakers) for training all feature extractors (i.e., all APC and CPC variants)

• 80-dimensional log Mel spectrums as input (surface) features
  – Normalized to zero mean and unit variance per speaker

• Examine two important characteristics of speech: phone and speaker information contained in extracted features
  – Phone classification on WSJ
  – Speaker verification on TIMIT

• Test if they generalize to datasets of different domains
Model Hyperparameters

• **APC architecture**
  – $L$-layer LSTMs where $L \in \{1, 2, 3\}$
  – 512 hidden units each layer
  – Residual connections between two consecutive layers
  – Predict $x_{i+n}$ where $n \in \{1, 2, 3, 5, 10, 20\}$

• **CPC Architecture**
  – Mainly follow the original implementation
  – Change the frame encoder (to take log Mel spectrums as inputs)
    * Original: 5-layer strided CNN
    * New: 3-layer, 512-dim fully-connected NN w/ ReLU activations
Phone Classification on Wall Street Journal

- **Data split:**
  - Train set: 90% of si284
  - Dev set: 10% of si284
  - Test set: dev93

- **Task:** Predict phoneme class for each frame and report frame error rate (FER)

- **Linear separability among phoneme classes as accessibility by downstream models**
  - Comparing $x + \{\text{linear classifier, MLP}\}$, $T_{CPC}(x)$ + linear classifier, and $T_{APC}(x)$ + linear classifier
    - $x$: log Mel features
    - $T_{CPC}(x)$: representations extracted by CPC
    - $T_{APC}(x)$: representations extracted by APC
### Phone Classification Results

<table>
<thead>
<tr>
<th>Method</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>(a) $x$ + linear</td>
<td></td>
</tr>
<tr>
<td>(b) $x$ + 1-layer MLP</td>
<td></td>
</tr>
<tr>
<td>(c) $x$ + 3-layer MLP</td>
<td></td>
</tr>
<tr>
<td>(d) Best $T_{CPC}(x)$ + linear</td>
<td></td>
</tr>
<tr>
<td>(e) $T_{APC_1}(x)$ + linear</td>
<td>39.4</td>
</tr>
<tr>
<td>(f) $T_{APC_2}(x)$ + linear</td>
<td>38.5</td>
</tr>
<tr>
<td>(g) $T_{APC_3}(x)$ + linear</td>
<td>37.2</td>
</tr>
</tbody>
</table>

- $T_{APC_d}(x)$: $d$ is the number of RNN layers
- $n$ is not relevant for (a) ~ (d)

### Discussions

- **Best $T_{CPC}(x)$:**
  1. Training - Sample negatives from same utterance as target frame
  2. Feature extraction - Take context encoder output instead of frame encoder output

- **Surface features $x$ with linear / non-linear classifier (a) ~ (c):**
  1. Incorporating non-linearity improves FER
  2. $x$ + 3-layer MLP outperforms the best $T_{CPC}(x)$

- **Comparison of $T_{APC_d}(x)$ (e) ~ (g):**
  1. Sweep spot exists when we vary $n$
  2. Significantly outperform (a) ~ (d)
Speaker Verification on TIMIT

• Comparing APC with $i$-vector and CPC
  – Obtaining $i$-vector representations
    * Train a universal background model (GMM w/ 256 components), $i$-vector extractor, and LDA model on TIMIT train set
    * Extract 100-dim $i$-vectors, project them to 24-dim with LDA

• Utterance representation = simple average of frame representations

• Report equal error rates (EER) on dev set; only consider female-female & male-male pairs
## Speaker Verification Results

<table>
<thead>
<tr>
<th>Method</th>
<th>( n )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>(a) \textit{i}-vector</td>
<td></td>
</tr>
<tr>
<td>(b) Best ( T_{CPC}(x) )</td>
<td></td>
</tr>
<tr>
<td>(c) ( T_{APC.1}(x) )</td>
<td>4.71</td>
</tr>
<tr>
<td>(d) ( T_{APC.2}(x) )</td>
<td>4.71</td>
</tr>
<tr>
<td>(e) ( T_{APC.3}(x) )</td>
<td>5.21</td>
</tr>
<tr>
<td>(f) ( T_{APC.3-1}(x) )</td>
<td>3.79</td>
</tr>
<tr>
<td>(g) ( T_{APC.3-2}(x) )</td>
<td><strong>3.43</strong></td>
</tr>
</tbody>
</table>

### Discussions

- \( T_{APC} > \text{best} T_{CPC} > \text{i-vector} \)
- In general, smaller \( n \) captures more speaker information
- Unlike phone classification, deeper APC tends to perform worse on speaker verification (c) ~ (e)
- Shallow layers contain more speaker information (e) ~ (g)

- \( T_{APC.3-l}(x) \): output of the \( l \)-th layer of \( T_{APC.3}(x) \)
- \( n \) is not relevant for (a) and (b)
Conclusions

- **Autoregressive Predictive Coding for speech representation learning**
  - Unsupervised - no labeled data required for training
  - Transforms surface features (e.g., log Mel) into a more accessible form
    * **Accessibility is defined as linear separability**
  - Extracted representations contain both phone and speaker information
    * **Outperform surface features, CPC, i-vector**
  - In a deep APC, lower layers tend to be more discriminative for speakers while upper layers provide more phonetic content

- **Code:** [https://github.com/iamyuanchung/Autoregressive-Predictive-Coding](https://github.com/iamyuanchung/Autoregressive-Predictive-Coding)
Thank you!

Questions?