ADDITIVE LEARNING FRAMEWORK FOR SELF EVOLVING AI

Collaborative learning
OBJECTIVES

Demonstrate additive learning for modules estimating User Identity

- What is additive learning?
  - AI modules helping each other for improved confidence
- Why additive learning?
  - Because networks need to evolve and stay up-to-date
- Where can these be employed?
  - In homes, cars, where user interacts frequently
OBJECTIVES

Demonstrate additive learning for modules estimating User Identity

• How are these useful?
  • correct functionality with changing subject features

• When to use it?
  • Low risk local system updates fine tuned for user experience
HOW IT WORKS

Strategies

• **Enrollment phase:**
  • Enroll subject into the system

• **Selection phase:**
  • Select entries from stored database

• **Evolution phase:**
  • Retrain the classifier with selected entries from above phase
ENROLLMENT PHASE
ENROLLMENT STRATEGY

- Collect data (Face and Voice) for enrolling the subject, using camera and mic
- Generate embeddings for enrolled person
- Train classifier using generated embeddings
SELECTION PHASE
FaceID: “FACE”  
FaceSure: “Yes/No”  

SpeakerID: “SPEAKER”  
SpeakerSure: “Yes/No”  

SpeakerID == FaceID  

True  

Face Sure?  

Yes  

Speaker Sure?  

No  

Save Face Image data  

Yes  

Do Nothing

No  

Save Voice data

No  

Save Face Image data
<table>
<thead>
<tr>
<th>Case</th>
<th>FACE-ID SURE</th>
<th>SPEAKER-ID SURE</th>
<th>SAVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case I</td>
<td>No</td>
<td>No</td>
<td>None</td>
</tr>
<tr>
<td>Case II</td>
<td>Yes</td>
<td>No</td>
<td>Voice data</td>
</tr>
<tr>
<td>Case III</td>
<td>No</td>
<td>Yes</td>
<td>Image data</td>
</tr>
<tr>
<td>Case IV</td>
<td>Yes</td>
<td>Yes</td>
<td>None</td>
</tr>
</tbody>
</table>
EVOLUTION PHASE
EVOLUTION STRATEGY

- Retrieve all (enrolled + saved) dataset.
- Generate embeddings for each data point.
- Select diverse Images/Audio to be used to retained using Euclidean and Cosine filters.
- Retrain if new information is added.
EVOLUTION STRATEGY

Saved Data + Enrolled Data

Generate Embedding

Retain entries with max Euclidean distance

Retain the ones with minimum Cosine similarity from remaining entries.

Retrain Classifier
IMPLEMENTATION DETAILS
ARCHITECTURE

FaceID®

SpeakerID

Arbiter

Inference

Retrain

Result

Retrain

Result

® https://cmusatyalab.github.io/openface/
ENROLLMENT

- OpenFace for generating Face embeddings
- SpeakerNet for generating Voice embeddings
- RadialSVM classifier for classification Training
FACE RECOGNITION

• OpenFace generates embedding to train classifier.
• FaceID module uses OpenFace’s confidence to generate surety.
• Surety is calculated based on classifier confidence, Euclidean distance and cosine similarity.
FACE RECOGNITION

Input Image ➔ Preprocessing ➔ FaceNet ➔ Classifier ➔ Output

- Preprocessing:
  - Face Detection
  - Alignment

- FaceNet:
  - 128 Embedding Vector

- Classifier:
  - Label 1
  - Label 2
  - Label N

Reference: https://cmusatyalab.github.io/openface/
SPEAKER RECOGNITION

• SpeakerNet Deep Neural Network generates embedding.
• RadialSVM classifier uses embedding to generate labels and confidence.
• Surety is calculated based on classifier confidence, Euclidean distance and cosine similarity.
SPEAKERNET

Input Audio

Preprocessing
- Noise Removal
- Silence Removal

Feature Extraction
- 64 MFCC Feature
  - 1st Δ
  - 2nd Δ

Convolutional Deep Neural Network

Triplet Loss

512 embedding Vector
## SPEAKERNET TRAINING DETAILS

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No of Speaker</th>
<th>No. Of Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMIT</td>
<td>630</td>
<td>6300</td>
</tr>
<tr>
<td>VoxForge</td>
<td>6122</td>
<td>86504</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>6860</strong></td>
<td><strong>137061</strong></td>
</tr>
</tbody>
</table>

### Input Specs: MFCC + DELTA + DELTA-DELTA Features

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Window Length</strong></td>
<td>25ms</td>
</tr>
<tr>
<td><strong>filters</strong></td>
<td>64</td>
</tr>
<tr>
<td><strong>sample Freq</strong></td>
<td>16 KHz</td>
</tr>
<tr>
<td><strong>Delta(acceleration)</strong></td>
<td>Used</td>
</tr>
<tr>
<td><strong>Delta-Delta(velocity)</strong></td>
<td>Used</td>
</tr>
<tr>
<td><strong>Feature length</strong></td>
<td>64</td>
</tr>
<tr>
<td><strong>Input sample duration</strong></td>
<td>~655ms</td>
</tr>
</tbody>
</table>
SPEAKERNET TRAINING DETAILS

- Convolution Neural Network is inspired by DeepSpeaker’s conv+res neural network.
- Used ADAGRAD optimizer with exponential decay learning rate.
- Alpha margin set to 0.5 (Euclidean distance) to choose triplets.
- TIMIT test data used to validate the SpeakerNet performance.
Different classifier tested for classification.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy (Test Data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinearSvm</td>
<td>76.8</td>
</tr>
<tr>
<td>GridSearchSvm</td>
<td>79.8</td>
</tr>
<tr>
<td>RadialSvm</td>
<td>86.7</td>
</tr>
<tr>
<td>DecisionTree</td>
<td>69.5</td>
</tr>
<tr>
<td>GaussianNB</td>
<td>65.1</td>
</tr>
<tr>
<td>MLPClassifier</td>
<td>70</td>
</tr>
<tr>
<td>Nearest_Neighbors</td>
<td>84.4</td>
</tr>
<tr>
<td>Gaussian_Process</td>
<td>82.3</td>
</tr>
<tr>
<td>Random_Forest</td>
<td>75.4</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>81.3</td>
</tr>
<tr>
<td>QDA</td>
<td>79.7</td>
</tr>
</tbody>
</table>

Results mentioned above are based on local test dataset generated for classifier training; RadialSVM classifier performs best with 86.7% accuracy.
• Send SpeakerSure as True to arbiter if SpeakerID satisfies below condition:
  • Classification confidence $\geq 0.85$
  • Euclidean Distance with same person enrolled data $\leq 1.0$
  • Cosine similarity with same person enrolled data $\geq 0.85$
• Send FaceSure as True to arbiter if FaceID satisfies below condition:
  
  • Classification confidence $\geq 0.75$
  
  • Euclidean Distance with same person enrolled data $\leq 1.0$
  
  • Cosine similarity with same person enrolled data $\geq 0.85$
• Compute Euclidean Distance and cosine similarity to decide which sample to retain. (both FaceID and SpeakerID)

• Retain 10 distinct images of each label for FaceID.

• Retain 40 Distinct Speaker feature of each label for SpeakerID.

• Trigger retraining if new samples were added.
EVOLUTION RESULTS

• With Additive learning framework, confidence increases with time and diverse training data.

• Face and Voice recognition confidence for a person increases by ~20% with time, as arbiter module selects diverse data and re trains the classifier.

• With Additive learning framework, we are also able to achieve more accuracy on our test dataset from 86.7% to ~90%.
ARBITER PERFORMANCE

• Time taken by arbiter module to generate embeddings and train the classifier:
  • **Speaker**: ~40 Secs [15 subjects, ~25 sec audio/subject]
  • **Face**: ~30 Secs [15 subjects, ~5 images/subject]

• Hardware configuration used:
  • X86 Ubuntu 16.04
  • 1 TitanX GPU
REFERENCES

• [1] OpenFace: https://cmusatyalab.github.io/openface/
• [6] Python Modules: scipy, scikit-learn, opencv-python, h5py, matplotlib, Pillow, requests, psutil, pypaudio, numpy, xgboost, scikits.talkbox, sklearn, python_speech_features, pandas
• [8] FFmpeg: https://www.ffmpeg.org/
• [9] TIMIT: https://catalog.ldc.upenn.edu/ldc93s1
• [10] VoxForge: https://old.datahub.io/dataset/voxforge
“Evolution, of course, is not something that simply applies to life here on earth; it applies to the whole universe.”

John Polkinghorne
“It is not the strongest species that survive, nor the most intelligent, but the ones most responsive to change.”

Charles Darwin
Thank you!
Inserting video: Insert/Video/Video from File.
Insert video by browsing your directory and selecting OK.

File type that works best in PowerPoint is: .wmv
DEMO: Placeholder (Insert picture behind gray bar)