Mobile and Sensor Systems

Lecture 5: Modeling and Inference

Sourav Bhattacharya
Lecture Overview

• Part - 1
  • Introduction to mobile and wearable sensing
  • Mobile sensing applications
  • Understanding the key tasks in mobile sensing
  • Challenges in mobile sensing

• Part - 2
  • Modeling audio using Deep Neural Networks
  • Multi-task learning through shared architecture
  • Open research questions
Part - 1
Mobile and Wearable Sensing

The mobile phone and wearable sensing domain is filled with hacks, and imaginative techniques that are used to circumvent the limitations of a platform that was designed for a different purpose.
## Mobile / Wearable Sensing Vs. Sensor Networks

<table>
<thead>
<tr>
<th>Mobile Sensing</th>
<th>Sensor Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Well suited for human activities</td>
<td>• Well suited for sensing the environment</td>
</tr>
<tr>
<td>• General purpose sensors, often not well suited for accurate sensing of the target phenomena</td>
<td>• Specialized sensors, designed to accurately monitor specific phenomena</td>
</tr>
<tr>
<td>• Multi-tasking OS. Main purpose is to support various applications</td>
<td>• All resources dedicated to sensing</td>
</tr>
<tr>
<td>• Low cost of deployment and maintenance (millions of users charge their devices)</td>
<td>• High cost deployment and maintenance (regular charging thousands of sensor nodes)</td>
</tr>
</tbody>
</table>
Mobile Sensing Applications

**Individual sensing:**
- fitness applications
- behaviour intervention applications

**Group/community sensing:**
- sense common activities and help achieving group goals
- examples: assessment of neighbourhood safety, environmental sensing, collective recycling efforts

**Urban-scale sensing:**
- large scale sensing, where large number of people have the same application installed
- examples: tracking speed of disease across a city, congestion and pollution in a city
Human Activity Recognition

Sensor used:
• Accelerometer or Gyroscope

Example inference:
• Walking, running, biking, up/down stairs etc.

Applications:
• Health / behaviour intervention
• Fitness monitoring
• Sharing within a community
Transportation-mode Detection

Sensor used:
- Accelerometer or Gyroscope
- GPS, WiFi localization

Example inference:
- Bus, bike, tram, train, car etc.

Applications:
- Intelligent transportation
- Smart commuting
Emotion Detection

Sensor used:
- Microphone, bluetooth
- GPS, WiFi localization
- Map speaking features to emotional state

Example inference:
- Emotional state, location and co-location with others

Applications:
- Behaviour intervention
- Computational social science
  - Using mobile sensing for quantifying theories in social science
Context and Environment

Sensor used:
• Microphone
• Camera

Example inference:
• Conversation, music, party, activity-related sound etc.

Applications:
• Automated diary
• Health and wellness
Challenges in Mobile Sensing

• Complex natural environment
• Heterogeneity of sensors
  • Vary in sampling frequency, sensitivity
• Noisy measurements
• Different sensor position and orientation
• Diverse population
• Privacy
• Limited processing and battery power
Challenges in Mobile Sensing

- Sensing is resource intensive

  - Battery
  - Memory
  - CPU
  - GPU
  - Storage

- The purpose of the embedded platform is to support multiple applications

- A sensing application needs to maintain a balance between
  - The amount of resource needed to operate
  - The accuracy of the detection that is achieved
Context Recognition: Machine Learning

**Supervised Learning:**
- Labeled data (training data)
- Objective: Learn a function from training data

\[ \mathcal{F}: \mathbf{X} \rightarrow \mathbf{Y} \quad \mathbf{x}_i \in \mathbb{R}^d \]

**Classification**
- Label is discrete / categorical variable

**Regression**
- Label is real-valued / continuous variable

In mobile sensing we have a large number of sensors

\[ x^T_1, \ldots, x^T_n \]

[Diagram of sensors and learner processing feature vectors to context]
Context Recognition: Machine Learning

Supervised Learning:
• Labeled data (training data)
• Objective: Learn a function from training data

\[ \mathcal{F} : X \rightarrow Y \quad x_i \in \mathbb{R}^d \]

Classification
• Label is discrete / categorical variable

Regression
• Label is real-valued / continuous variable

In mobile sensing we have a large number of sensors

\[ x_1 \in \mathbb{R}^{d_1} \rightarrow \text{Learner 1} \]
\[ \vdots \]
\[ x_n \in \mathbb{R}^{d_n} \rightarrow \text{Learner N} \]

\[ y_i \]

Feature vector | Label
---|---
\( x_1 \) | \( y_1 \)
\( x_2 \) | \( y_2 \)
\( \vdots \) | \( \vdots \)
\( x_n \) | \( y_n \)
Part - 2
Hot-keyword Detection: Problem Definition

Audio

\[ h_\theta : X \to \{C_1, \ldots, C_{12}\} \text{, where } X \in \mathcal{R}^d \]
HotKeyword Dataset

- 16 KHz, 16-bit audio

\[ h_\theta : X \to \{ C_1, \ldots, C_{12} \} \]

\[ X \in \mathcal{R}^{16,000} \]

Training a CNN: Supervised Learning

\[ h_\theta, \text{ where } \theta \in \mathcal{R}^p \]

<table>
<thead>
<tr>
<th>Feature vector</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>( C_1 )</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>( C_1 )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( x_n )</td>
<td>( C_{10} )</td>
</tr>
</tbody>
</table>

Acknowledgement: Pete Warden, Google

https://www.tensorflow.org/versions/master/tutorials/audio_recognition
End-to-end CNN Architecture

- **Input**: Raw audio samples
- **Output**: Logits (dimension=12)
- **Normalization**:
  \[
  \sigma(x)_j = \frac{e^{x_j}}{\sum_{i=1}^{K} e^{x_i}}
  \]
- **Distance metric**: Cross-entropy, KLD
Loss Function

- In case of supervised learning

- Cross-entropy loss

\[
\mathcal{L}(h_\theta(x_i), y_i) \\
\mathcal{L}(\sigma(h_\theta(x_i)), \text{onehot}(y_i)) \\
- \sum_{i=1}^{K} p_i \log(q_i)
\]

\[
- \sum_{I=1}^{K} y_i \log \left( \frac{\exp(h_\theta(x)_i)}{\sum_{j=1}^{K} \exp(h_\theta(x)_j)} \right) = - \log \left( \frac{\exp(h_\theta(x)_y)}{\sum_{j=1}^{K} \exp(h_\theta(x)_j)} \right) \\
= \log \left( \sum_{j=1}^{K} \exp(h_\theta(x)_j) \right) - h_\theta(x)_y
\]
Training CNN: Loss Minimization

Average loss: \[
\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(h_{\theta}(x_i), y_i)
\]

Gradient descent: \[
\theta \leftarrow \theta - \frac{\alpha}{|\mathcal{B}|} \sum_{i=1}^{N} \nabla_{\theta} \mathcal{L}(h_{\theta}(x_i), y_i)
\]

Problems with gradient descent:
- No guarantee that it will find a global minimum
- Convergence to a local minimum can be slow
HotKeyword recognition: A Practical Guide

Step 1: Splitting dataset into training, validation and test sets

Step 2: Perform data normalization, e.g., 0 dbFS

Step 3: Model architecture selection and parameter initialization

Step 4: Fast mini-batch generation

Step 5: Data augmentation to make the trained model resilient to noise

Step 6: Perform model prediction on the augmented mini-batch

Step 7: Compute loss and perform gradient descent

Step 8: Stop if the model has converged, otherwise go to Step 4
Convolutional Neural Network Training for Hot Key-word Recognition

Divide audio files into Training, Validation and Testing sets

Get a random index

Select random background noise

Prepare Background Noise Dataset

Get Audio samples

Get random time-shift amount

Get random background audio volume

Audio Augmentation

×256 (Batch size)

Loss function: Cross entropy loss

Optimizer: Adam/SGD

Repeat steps: 500 time or Employ early stopping
Input: MFCC Features

Audio → Pre-emphasis → Framing → Windowing → Short-time Fourier Transform

MFCC ← Lifting ← Discrete Cosine Transformation ← Filter Banks ← Power Spectrum

Audio Waveform

MFCC Coefficients
Multi-task Audio Inferencing

- **Objective:** Infer multiple contexts from the same input audio
  - Who is the speaker? Is the person stressed? Male or female speaker?

- Save memory & bandwidth
- Improve latency
- Preserve privacy
Multi-task Training

- Audio pre-processing
  - Audio signal 8KHz → Filter bank coefficients → Summary → Normalization
  - Mini-batch for Output Layer

- Training shared architecture
  - Filter Bank Summary (Input Layer) → Hidden Layers → Mini-batch for Hidden Layers
  - Task 1 → Task 2 → Task 3 → Task 4
Open Research Questions

• How can we use unsupervised data to bootstrap the training procedure and reduce the amount of labeled data?

• How can we squeeze the resource requirements of large-scale neural networks for resource-constrained devices?

• Protecting privacy of the users.

• Multi-modal rich modeling of sensor data for accurate high-level context-recognition.
References


• S. Hemminki, P. Nurmi, S. Tarkoma, Accelerometer-based transportation mode detection on smartphones, SenSys 2011.

• S. Bhattacharya and N.D. Lane, Sparsification and Separation of Deep Learning Layers for Constrained Resource Inference on Wearables, SenSys 2016.

• P. Georgiev, S. Bhattacharya, N.D. Lane, C. Mascolo, Low-resource Multi-task Audio Sensing for Mobile and Embedded Devices via Shared Deep Neural Network Representations. IMWUT (UbiComp) September 2017.