Computational Social Science with Images and Audio

Elliott Ash, **Philine Widmer**

24 November 2023

Recall that feature extraction is used for many audio (classification) tasks.

In classical machine learning – and often beyond!

- Recall the typical pipeline for classical machine learning:

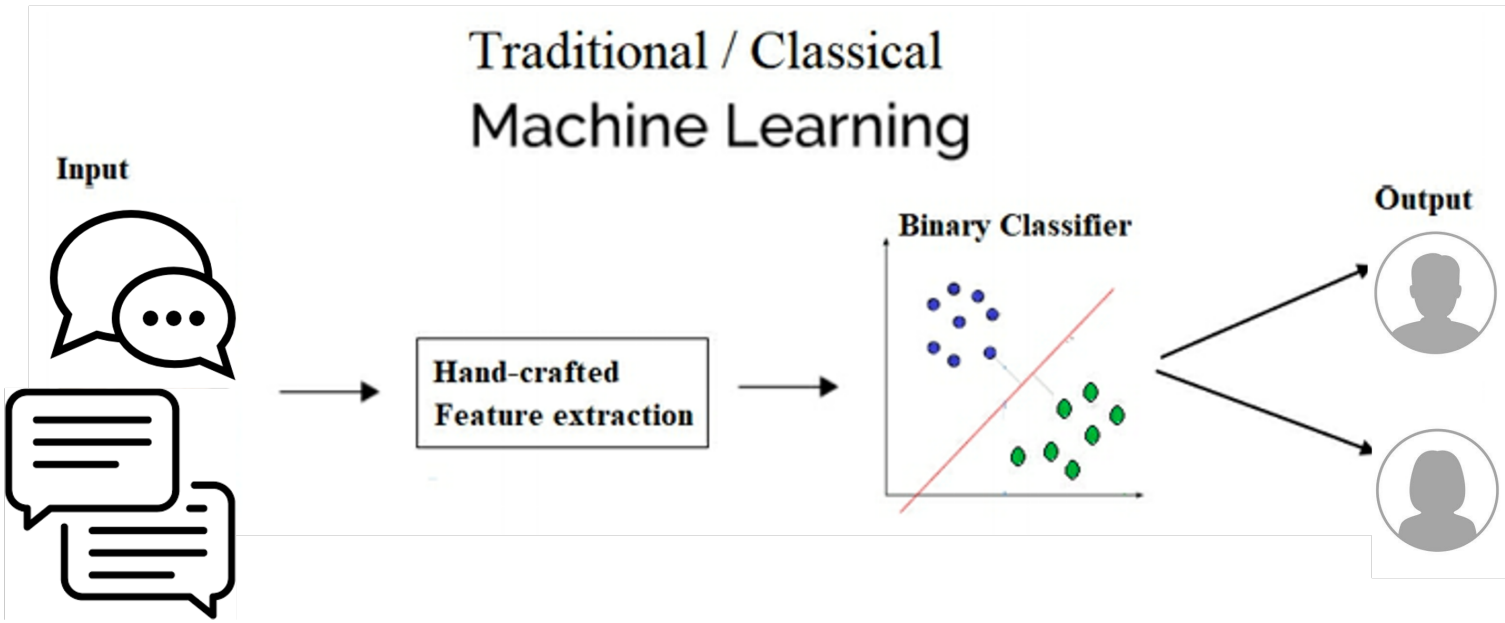

Figure: Own Adaptation of Dey (2018)[1]

[1] Dey, S. (2018). Hands-On Image Processing with Python: Expert Techniques for Advanced Image Analysis and Effective Interpretation of Image Data. Packt Publishing Ltd.

What models are used in audio analysis with deep learning?

The audio analysis landscape is diverse.

- CNN are used for classification with audio data
  - But less prominently than in CV!
  - Audio data is often transformed into spectrograms (time-frequency representations) to use CNN
  - Examples: Emotions, speaker identity, accents, ...
- Otherwise, varied, possibly hybrid approaches (RNN, transformers)
- In any case, audio data often requires pre-processing
- Other methods in audio analysis:
  - Classical ML techniques: Recall last week's features (e.g., MFCC)
  - Unsupervised and semi-supervised approaches: Useful with limited labeled data

Audio data is sometimes used with CNN or RNN.

Tailored for tasks with temporal dependencies

- For CNN, the conceptual foundation discussed in the computer vision part also applies here
- But what are Recurrent Neural Networks (RNN)?
  - Recall that CNN are "specialists" for spatial features
  - RNN are also "specialists", but for sequential data
  - Sequential data examples are audio or text
- RNN have a memory mechanism to remember previous inputs
  - They can, thus, "remember" context in sequences
  - For sequences, context is important to understand the overall pattern

Conceptually, how does an RNN proceed?

- The data is typically pre-processed
  - Extract features: for example, MFCC by time t (often relatively short snippets)
  - Each feature at time t becomes input x_t to the RNN
- $\forall t \in \{0, 1, 2, \ldots, T\}, x_t$ are processed by the RNN sequentially
- RNN computes new hidden state h_t for each time step t
- Typically: $h_t = \text{Activation}(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1} + b)$
- W_{xh}, W_{hh} are weight matrices; b is a bias vector
- RNN learns W_{xh}, W_{hh}, and b
- Hidden state h_t used for output (e.g., in classification)

Why are RNN suitable for sequential data?

- Quiz questions for better intuition:
  - Can you explain the dimensions of the weight matrices?
  - What is the equivalent of the bias in linear regression?
  - Why are RNN suitable for sequential data?

While useful for many tasks, RNN face some challenges.

- Vanishing and exploding gradient problems during training, especially with long sequences
- Difficulty with very long-range dependencies in sequences
- Variants like LSTM (Long Short-Term Memory) may mitigate some challenges
- RNNs and their variants remain important in some sequential data applications