Speech commands

Machine Learning

Claudio Cusano

A.A. 2021/2022

Speech recognition is a task in which the goal is to infer the words expressed by a speaker from a voice recording. The input is usually just the audio signal as recorded by a microphone (mono or stereo). The output is the corresponding sequence of words.

Speech recognition systems are usually required to be able to transcribe any sequence of meaningful words. However, there are applications in which a single word at a time is expected, and only a small set of words are of interest. As an extreme example, in *trigger word detection* only a specific word, or a short sentence need to be recognized ("Alexa!", or "OK Google!").

Here we will focus on a particular case of of speech recognition, in which the audio represents the pronunciation of a single word from the following list:

0. `backward`;
1. `bed`;
2. `bird`;
3. `cat`;
4. `dog`;
5. `down`;
6. `eight`;
7. `five`;
8. `follow`;
9. `forward`;
10. `four`;
11. `go`;
12. `happy`;
13. `house`;
14. `learn`;
15. `left`;
16. `marvin`;
17. `nine`;
18. `no`;
19. `off`;
20. `on`;
21. `one`;
22. `right`;
23. `seven`;
24. `sheila`;
25. `six`;
26. `stop`;
27. `three`;
28. `tree`;
29. `two`;
30. `up`;
31. `visual`;
32. `wow`;
33. `yes`;
34. `zero`.

It is a 35-class classification problem.

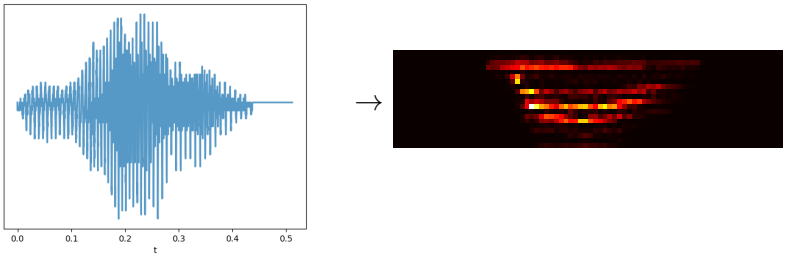
1 Lab activity

In this lab exercise we will use the *Speech Commands Data Set*[1]. Before starting the exercise be sure to read the "Preliminaries" section here below, download the data set and review the relevant classification models.

1.0 Preliminaries

The data set includes 105 829 recordings of the 35 words uttered by many different speakers. It has been divided into a training set of 84 291 audio clips, a validation set of 12 162 and a test set including 9376 clips.

Feature extraction has already been performed. The features are spectrograms extracted from the recorded waveforms (a spectrogram encodes the power distribution of the signal in a given time period, and over a set of frequencies). The figure here below shows one example of waveform with the corresponding spectrogram.



Spectrograms have been made uniform in size by padding or truncating the original signals. In the end, each spectrogram is an array of 20 frequencies times 80 time periods, reshaped as a a vector of 1600 components.

The original clips can be freely downloaded from the address `http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz`, features are stored in the files `train.npz`, `validation.npz` and `test.npz`, while the files `train-names.txt`, `validation-names.txt` and `test-names.txt` list the content of the training, validation and test sets, in the correct order. The feature extraction script `extract_features.py` is provided as reference (you need the `scipy` library if you want to use it).

To classify the data we will build multilayer perceptrons. If you are not familiar with this model, please review it before starting the lab activity. It is also a good idea to quickly review the notes about feature normalization.

1.1 Visualize the data

Having an intuition of the data to be processed may help in designing a good model. Write a script for the visualization of some of the spectrograms. Examine the range of values of the features.

[1] `https://www.tensorflow.org/datasets/catalog/speech_commands`

1.2 Feature normalization

Guess which normalization technique could be used for this problem. You can repeat the experiments with different techniques.

1.3 Train a neural network

Train a neural network for spoken digit recognition. Define and train a multilayer perceptron without hidden layers. Train and evaluate it. Try with batch gradient descent and with stochastic gradient descent with minibatches of different size.

1.4 Network architecture

Add one or more hidden layers. Try with layers of different width. What is the best architecture for this problem?

1.5 Analysis

Build a *confusion matrix* which summarizes the behavior of the network. This will be a 35×35 matrix where the element C_{ij} counts how many times the class i has been recognized as a sample of class j. Also normalize each row to show percentages instead of raw counters. Which classes are more likely to be confused?

Identify some of the classification errors, visualize their spectrogram and listen to the original audio clips. What kind of samples are easily misclassified?

2 Assignment

As homework, review and refine the scripts programmed in the lab activity. Repeat the experiments with different network architectures. In addition, perform the following exercises.

2.1 Feature normalization

Compare the effectiveness of different feature normalization techniques.

2.2 Visualization

Show as images the set of weights of the MLP without hidden layers. Can you identify which parts of the spectrograms are used by the network to choose the output class?

2.3 Report

Prepare a report of one or two pages with the answers to document all the experiments and their results. The report must be in the PDF format. Include your name in the report and conclude the document with the following statement: "I affirm that this

report is the result of my own work and that I did not share any part of it with anyone else except the teacher."

Make a ZIP archive with the report and the python scripts you used and and upload it on the course website.