Few-Shot Speaker Identification Using Masked Autoencoders and Meta-Learning

John Boccio
Department of Electrical Engineering
Stanford University
jboccio@stanford.edu

Abstract

Speaker identification refers to the task of identifying which speaker is talking from a given set of speakers. With the rise of deep learning and data driven approaches to audio processing problems, new approaches can be taken on this problem which traditionally was performed by trying to find separability between speakers using hand-engineered feature extractors. The speaker identification problem is a very appropriate problem for meta-learning. Meta learning is a machine learning technique in which a model is trained to learn how to learn, allowing it to adapt to new tasks quickly using relatively little data. By applying meta-learning to the speaker identification problem, a model can be trained to learn how to recognize speakers from a small amount of audio samples from each of the speakers. This could be useful in scenarios where the set of possible speakers is not known in advance, or where the speakers may vary over time.

All of the data that was used for this paper comes from the VoxCeleb dataset [2]. VoxCeleb provides labeled audio data spoken from over 1,000 celebrities. These audio files are then cut into 3 second segments and converted into a spectrogram which will then be fed into a convolutional neural network. The final result is a dataset with over 200,000 spectrograms, each labeled with an id that represents the celebrity that is talking.

The approach that this paper has taken towards applying meta-learning to the speaker identification problem is through the use of a prototypical network (protonet) [3]. The protonet generates an representation of a spectrogram from one speaker which, ideally, is a far distance from the representation of a spectrogram from a different speaker. These representations of each speaker are then used to perform classification. This paper also experiments with the affects of pretraining the network using a masked autoencoder [1]. When applied to speaker identification, a masked auto encoder can be trained to extract characteristics of a speaker's voice from audio samples and use these characteristics to identify the speaker. Combining meta learning with masked auto encoders offers the potential to train a speaker identification model that can adapt to new speakers quickly and accurately using a small amount of training data.

Given 5 examples of 5 different speakers, a model trained from scratch is able to identify who from these 5 speakers is talking with 94% accuracy. When the model is initialized with the weights learned from the masked autoencoder, the best accuracy achieved decreased to 91%.

C330: Deep Multi-Task and Meta-Learning, Fall 2022, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

1 Introduction

Explain the problem and why it is important. Discuss your motivation for pursuing this problem. Give some background if necessary. Clearly state what the input and output is. Be very explicit: "The input to our algorithm is an image, amplitude, patient age, rainfall measurements, grayscale video, etc.. We then use a SVM, neural network, linear regression, etc. to output a predicted age, stock price, cancer type, music genre, etc.." This is very important since different teams have different inputs/outputs spanning different application domains. Being explicit about this makes it easier for readers. If you are using your project for multiple classes, add a paragraph explaining which components of the project were used for each class.

2 Related work

You should find existing papers, group them into categories based on their approaches, and discuss their strengths and weaknesses, as well as how they are similar to and differ from your work. In your opinion, which approaches were clever/good? What is the stateof-the-art? Do most people perform the task by hand? You should aim to have at least 5 references in the related work. Include previous attempts by others at your problem, previous technical methods, or previous learning algorithms. Google Scholar is very useful for this: https://scholar.google.com/ (you can click "cite" and it generates MLA, APA, BibTeX, etc.)

3 Dataset and Features

Describe your dataset: how many training/validation/test examples do you have? Is there any preprocessing you did? What about normalization or data augmentation? What is the resolution of your images? How is your time-series data discretized? Include a citation on where you obtained your dataset from. Depending on available space, show some examples from your dataset. You should also talk about the features you used. If you extracted features using Fourier transforms, word2vec, PCA, ICA, etc. make sure to talk about it. Try to include examples of your data in the report (e.g. include an image, show a waveform, etc.).

4 Methods

Describe your learning algorithms, proposed algorithm(s), or theoretical proof(s). Make sure to include relevant mathematical notation. For example, you can include the loss function you are using. It is okay to use formulas from the lectures (online or in-class). For each algorithm, give a short description of how it works. Again, we are looking for your understanding of how these deep learning algorithms work. Although the teaching staff probably know the algorithms, future readers may not (reports will be posted on the class website). Additionally, if you are using a niche or cutting-edge algorithm (anything else not covered in the class), you may want to explain your algorithm using 1/2 paragraphs. Note: Theory/algorithms projects may have an appendix showing extended proofs (see Appendix section below).

5 Experiments/Results/Discussion

You should also give details about what (hyper)parameters you chose (e.g. why did you use X learning rate for gradient descent, what was your mini-batch size and why) and how you chose them. What your primary metrics are: accuracy, precision, AUC, etc. Provide equations for the metrics if necessary. For results, you want to have a mixture of tables and plots. If you are solving a classification problem, you should include a confusion matrix or AUC/AUPRC curves. Include performance metrics such as precision, recall, and accuracy. For regression problems, state the average error. You should have both quantitative and qualitative results. To reiterate, you must have both quantitative and qualitative results! If it applies: include visualizations of results, heatmaps, examples of where your algorithm failed and a discussion of why certain algorithms failed or succeeded. In addition, explain whether you think you have overfit to your training set and what, if anything, you did to mitigate that. Make sure to discuss the figures/tables in your main text throughout this section. Your plots should include legends, axis labels, and have font sizes that are legible when printed.

6 Conclusion and Future Work

Summarize your report and reiterate key points. Which algorithms were the highestperforming? Why do you think that some algorithms worked better than others? For future work, if you had more time, more team members, or more computational resources, what would you explore?

7 Contributions

John Boccio was the only member of the team and completed all of the work described in this report.

References

[1] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021.

[2] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *CoRR*, abs/1706.08612, 2017.

[3] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017.