A Review on Audible Sound Analysis based on State Clustering through Multiple Deep Neural Network Modeling

Ms. Sneha Thune[1], Dr. N.K. Choudhari[2], Ms. A.P. Thakare[3]

[1]PG Student, MTech, Department of Electronics &Communication Engineering, Priyadarshini Bagwati College of Engineering
[2]Professor, Department of Electronics &Communication Engineering, Priyadarshini Bagwati College of Engineering
[3]Asst.Professor, Department of Electronics &Communication Engineering, Priyadarshini Bagwati College of Engineering

---

Abstract:- *Statistical parametric speech synthesis (SPSS) combines an acoustic model and a vectored to render speech given a text. Typically decision tree-clustered context-dependent hidden Markov models (HMMs) are employed as the acoustic model, which represent a relationship between linguistic and acoustic features. Recently, artificial neural network-based acoustic models, such as deep neural networks, mixture density networks, and long short-term memory recurrent neural networks (LSTM-RNNs), showed significant improvements over the HMM-based approach. This project reviews the progress of acoustic modeling in SPSS from the HMM to the LSTM-RNN. Understanding sound is one of the basic tasks that our brain performs. This can be broadly classified into Speech and Non-Speech sounds. We have noise robust speech recognition systems in place but there is still no general purpose acoustic scene classifier which can enable a computer to listen and interpret everyday sounds and take actions based on those like humans do, like moving out of the way when we listen to a horn or hear a dog barking behind us*

Keywords: *SPSS, HMM, LSTM-RNNs*

1. INTRODUCTION

The goal of text-to-speech (TTS) synthesis is to render a naturally sounding speech waveform given a text to be synthesized. Figure 1 outlines a human speech production process. A text (or concept) is first translated into movements of articulators and organs. Using air-flow from a lung, vocal source excitation signals containing periodic (by vocal cord vibration) and aperiodic (by turbulent noise) components are generated.

By filtering the source signals by time varying vocal tract transfer functions controlled by the articulators, their frequency characteristics are modulated. Finally, the filtered source signals are emitted. The aim of TTS is to mimic this process by computers in some way. Text-to-speech can be viewed as a sequence-to-sequence mapping problem; from a sequence of discrete symbols (text) to a real valued time series (waveform).

Typical TTS systems consist of text analysis and speech synthesis parts. The text analysis part includes a number of natural language processing (NLP) steps, such as word segmentation, text normalization, part-of-speech (POS) tagging, and grapheme-to-phoneme (G2P) conversion. This part performs a mapping from a sequence of discrete symbols to another sequence of discrete symbols (e.g., sequence of characters to sequence of words). The speech synthesis part performs mapping from a sequence of discrete symbols to real-valued time series.

It includes prosody prediction and speech waveform generation. The former and latter parts are often called “front-end” and “back-end” in TTS, respectively. Although both of them are important to achieve high-quality TTS systems, this paper focuses on the latter one. Statistical parametric speech synthesis (SPSS) is one of the major approaches in the back-end part. This approach uses an acoustic model to represent the relationship between linguistic and acoustic features and a vocoder to render a speech waveform given acoustic features. This approach offers various advantages over concatenative speech synthesis , which is another major approach in the text (concept) frequency transfer characteristics magnitude start--end fundamental frequency air flow Sound source voiced: pulse unvoiced: noise speech

Outline of speech production process. back-end part of TTS systems, such as small footprint and flexibility to change its voice characteristics However, the naturalness of the synthesized speech from SPSS is not as good as that of the best samples from concatenative speech synthesizers. Zen et al. reported three major factors that can degrade the naturalness quality of vocoder, accuracy of acoustic model, and effect of over smoothing. This paper addresses the accuracy of acoustic model. Although there have been many attempts to develop a more accurate acoustic model for SPSS], the hidden Markov model (HMM) is the most popular one. Statistical parametric speech synthesis with HMMs is known as HMM-based speech synthesis Inspired from the success in machine learning and automatic speech recognition, 5 different types of artificial neural network based acoustic models were proposed in 2013.

Highly accurate audio classifiers, if exist, have many practical applications in all walks of our life, from medicine to industry. Developing such accurate classifiers, however, is arduous. Unlike in computer vision, advancements in computer listening are in early stages. Audio classifiers, unlike image classifiers, typically have lower accuracies. However, with the ready availability of curated, public audio datasets and ML classification algorithms, it is easier than ever to build accurate classifiers.Several research papers recently published classifiers on the UrbanSound8k dataset. However, these classifiers only have 50-79% accuracy range. In this project, by employing various ML strategies, I aim to significantly improve this accuracy from its current high of 79%. To obtain generalizable and reliable results, for ML model training, I will use industry gold-standard, k-fold-cross-validation on the training set, which is 80% of the source data. The trained model will be tested on the test set, which is the remaining 20% unseen source data. Experiments will be repeated and even run on different operating systems to measure for variability.

A typical convolutional neural network consists of a number of different layers stacked together in a deep architecture: an input layer, a group of convolutional and pooling layers (which can be combined in various ways), a limited number of fully connected hidden layers, and an output (loss) layer. The actual difference, when compared to the multilayer perceptron, lies in the introduction of a combination of convolution and pooling operations

A convolutional layer introduces a special way of organizing hidden units which aims to take advantage of the local structure present in the two-dimensional input data (mostly, but not limited to, images). Each hidden unit, instead of being connected to all the inputs coming from the previous layer, is limited to processing only a tiny part of the whole input space (e.g. small 3 ⍰ 3 blocks of pixels), called its receptive field. The weights of such a hidden unit create a convolutional kernel (filter) which is applied to (tiled over) the whole input space, resulting in a feature map. This way, one set of weights can be reused for the whole input space. This is based on the premise that locally useful features will be also useful in other places of the input space - a mechanism which not only vastly reduces the number of parameters to estimate, but improves robustness to translational shifts of the data.

A typical convolutional layer will consist of numerous filters (feature maps). Further dimensionality reduction can be achieved through pooling layers, which merge adjacent cells of a feature map. The most common pooling operations performed are taking the max (winner takes all) or mean of the input cells. This downsampling further improves invariance to translation

II BLOCK DIAGRAM

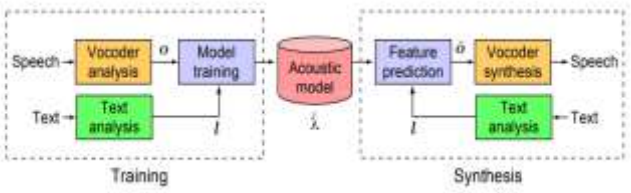

Fig 1 Block Diagram of Overall System

Neural Networks

Statistical parametric speech synthesis combines vocoder and acoustic models to render a speech waveform given a text. Although SPSS offers various advantages over concatenative speech synthesis, such as flexibility to change its voice characteristics and small footprint, the naturalness of synthesized speech from SPSS is still not as good as the best samples from concatenative one.

The accuracy of acoustic models is one of the factors that degrade the naturalness. This project reviewed the progress of acoustic models in SPSS from the acoustic trajectory and context modeling point of views. Although a number of different types of acoustic models have been applied to SPSS, the HMM has been the most popular one for the last two decades. However, recently proposed artificial neural network based acoustic models look promising and have started replacing HMMs in SPSS.

One major reason why the HMM has been a dominant acoustic model in SPSS is the existence of open-source software to build endto-end systems. As there are a number of open-source software for deep learning, we expects that artificial neural networks will be the next dominant acoustic model in the very near future

III Research Methodology/Planning of Work

Step 1. Extracting Features

Although deep learning eliminates the need for hand-engineered features, we have to choose a representation model for our data. Instead of directly using the sound file as an amplitude vs time signal we use a log-scaled mel-spectrogram with 128 components (bands) covering the audible frequency range (0-22050 Hz), using a window size of 23 ms (1024 samples at 44.1 kHz) and a hop size of the same duration.

This conversion takes into account the fact that human ear hears sound on log-scale, and closely scaled frequency are not well distinguished by the human Cochlea. The effect becomes stronger as frequency increases. Hence we only take into account power in different frequency bands. This sample code gives an insight into converting audio files into spectrogram images. We use glob and librosa library - this code is a standard one for conversion into spectrogram and you're free to make modifications to suit the needs.

Step 2. Choosing an Architecture

We use a convolutional Neural Network, to classify the spectrogram images. This is because CNNs work better in detecting local feature patterns (edges etc) in different parts of the image and are also good at capturing hierarchical features which become subsequently complex with every layer as illustrated in the image

Step 3. Transfer Learning

As the CNNs learn features hierarchically, we can observe that the initial few layers learn basic features like various edges which are common to many different types of images. Transfer learning is the concept of training the model on a dataset with large amounts of similar data and then modifying the network to perform well on the target task where we do not have a lot of data. This is also called ***fine-tuning*** - explains transfer learning very well.

Step 4. Data Augmentation

While dealing with small datasets, learning complex representations of the data is very prone to overfitting as the model just memorises the dataset and fails to generalize. One way to beat this is to augment the audio files into producing many files each with a slight variation.

We proposed a five-layer stacked CNN network for sound event recognition based on a special convolutional filter configuration with decreasing filter sizes and static and delta log-mel input features. The test results from three datasets, ESC-10, ESC-50, and Urbansound8k, indicated that the recognition performance of our model is higher than those of previous logmel-CNN models including Picazk Salamon and Bello and EnvNet We designed an end-to-end stacked CNN model for sound event recognition from raw waveforms without feature engineering. It has a special two-layer feature extraction convolution layer and convolutional filter configuration to directly learn features from raw waveforms.

Our models achieve a 2% and 16% improvement in recognition accuracy on the datasets of ESC-50 and Urbansound8k respectively, compared to the existing top end-to-end model EnvNet and the 18-layer convolutional neural network

We developed a novel ensemble environmental event sound recognition model, DS-CNN, by fusing logmel-CNN and end-to-end raw-CNN models using DS evidence theory to exploit raw waveform features as well as the log-mel features.

The goal of this project was to evaluate whether convolutional neural networks can be successfully applied to environmental sound classification tasks, especially considering the limited nature of datasets available in this field. It seems that they are indeed a viable solution to this problem. Conducted experiments show that a convolutional model outperforms common approaches based on manuallyengineered features and achieves a similar level as other feature learning methods.

Although, taking into consideration much longer training times, the result is far from groundbreaking, it shows that convolutional neural networks can be effectively applied in environmental sound classification tasks even with limited datasets and simple data augmentation. What is more, it is quite likely that a considerable increase in the size of the available dataset would vastly improve the performance of trained models, as the gap to human accuracy is still profound.

IV Design and Implementation of Neural Networks

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be −1 and 1.

Unlike von Neumann model computations, artificial neural networks do not separate memory and processing and operate via the flow of signals through the net connections, somewhat akin to biological networks.

These artificial networks may be used for predictive modeling, adaptive control and applications where they can be trained via a dataset. Self-learning resulting from experience can occurwithin networks, which can derive conclusions from a complex and seemingly unrelated set of information

One of the main things with training deep neural architectures in a supervised manner is the amount of computational effort and labeled data required for efficient learning. While the former is in some part addressed on a universal basis by hardware advances and general-purpose GPU computing, the latter is very domain-dependent.

Unfortunately, publicly available datasets of environmental recordings are still very limited - both in number and in size1. This is quite understandable, considering the high cost of manual annotation. Although the situation gradually improves with the introduction of new collections of recordings, it is still one of the major hindrances to the development of new data-intensive approaches in this field.

This is especially important, since the performance of supervised deep models is strongly influenced by the size of the dataset available for learning

VI CONCLUSION

Statistical parametric speech synthesis combines vocoder and acoustic models to render a speech waveform given a text. Although SPSS offers various advantages over concatenative speech synthesis, such as flexibility to change its voice characteristics and small footprint.

REFERENCES

[1]. S.-R. Kuang and J.-P. Wang, “Design of power-efficient configurable booth multiplier,”IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp. 568–580, Mar. 2010.

[2]. W. Ling and Y. Savaria, “Variable-precision multiplier for equalizer with adaptive modulation,” in Proc. 47th Midwest Symp. Circuits Syst., vol. 1. Jul. 2004, pp. I 553–I556.

[3]. Xilinx Inc.: XAPP 290: Two flows for Partial Reconfiguration: Module Based or Difference Based.www.xilinx.com, Sept. (2004).

[4]. O. A. Pfander, R. Hacker, and H.-J. Pfleiderer, “A multiplexer-based concept for reconfigurable multiplier arrays,”in Proc. Int. Conf. Field Program. Logic Appl., vol. 3203. Sep. 2004, pp. 938–942.

[5]. H. Lee, “A power-aware scalable pipelined booth multiplier,” in Proc. IEEE Int. SOC Conf., Sep. 2004, pp. 123– 126.

[6]. Mesquita, D., Moraes, F., Palma, J., Moller, L., Calazanas, N.: Remote and Partial Reconfiguration of FPGAs: tools and trends. International Parallel and Distributed Processing Symposium,(2003).

[7]. Meyer-Baese, U.: Digital Signal Processing with Field Programmable Gate Arrays. Springer, (2001).

[8]. Xilinx Inc.: Development System Reference Guide. www.xilinx.com.

[9]. A. Bermak, D. Martinez, and J.-L. Noullet, “High density 16/8/4-bit configurable multiplier,” Proc. Inst. Electr. Eng. Circuits Devices Syst.,vol. 144, no. 5, pp. 272–276, Oct. 1997.

[10]. M. Hatamian and G. L. Cash, "A 70 MHz 8 bit x 8 bit parallel pipelined multiplier in 2.5 µm CMOS,"IEEE Journal of Solid-State Circuits, vol. 21, no. 4, pp. 505–513, 1986.

[11]. Yeong-Jae Oh, Hanho Lee, Chong-Ho Lee, "A Reconfigurable FIR Filter Design Using Dynamic Partial Reconfguration", IEEE,vol-06, pp. 4851–4854, ISCAS 2006.

[12]. S.Karthick, Dr. s. Valarmathy and E.Prabhu ," RECONFIGURABLE FIR FILTER WITH RADIX-4 ARRAY MULTIPLIER" , jatit, Vol. 57 No.3, pp.326-336 , Nov.2013.

[13]. K.Anandan and N.S.Yogaananth, "VLSI Implementation of Reconfigurable Low Power Fir Filter Architecture" , IJIRCCE, Vol.2, Special Issue 1, pp no 3514-. 3520, March 2014.

[14]. Martin Kumm, Konrad M¨oller and Peter Zipf "Dynamically Reconfigurable FIR Filter Architectures with Fast Reconfiguration", ieee journal of solid-state circuits, vol. 41, no. 4, april 2006.

[15]. Pramod Kumar Meher, Shrutisagar Chandrasekaran,and Abbes Amira , "FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic", ieee transactions on signal processing ,pp no-1-9.

[16]. Xiaoxiao Zhang, Farid Boussaid and Amine Bermak, "32 Bit×32 Bit Multiprecision Razor-Based Dynamic Voltage Scaling Multiplier With Operands Scheduler", ieee transactions on very large scale integration (vlsi) systems, vol. 22, no. 4, april 2014.

[17]. K.Gunasekaran and M.Manikandan," High Speed Reconfigurable FIR Filter using Russian Peasant Multiplier with Sklansky Adder", Research Journal of Applied Sciences, Engineering and Technology 8(24): 2451-2456, 2014.

[18]. Shidhartha Das, David Roberts, Seokwoo Lee, Sanjay Pant, David Blaau, Todd Austin, Krisztiá Flautner and Trevor Mudge , "Self-Tuning DVS Processor Using Delay-Error Detection and Correction", ieee journal of solid-state circuits, vol. 41, no. 4, april 2006.

[19]. J Britto Pari ,et al., "Reconfigurable Architecture Of RNS Based High Speed FIR Filter" , Indian Journal Of Engineering And Material Sciences,pp . 230-240, vol.21, april 2014.