Pronunciation generation using WaveGANs

Generating correct pronunciations in a Computer Aided Pronunciation Training (CAPT) System in the learner's voice (for isolated words in Kannada)

Ishaan Lagwankar
Student, 2nd Year, B.Tech, CSE
PES University, Banashankari
Bangalore
SRN: PES1201700150

Arpit Agarwal
Student, 2nd Year, B.Tech, CSE
PES University, Banashankari
Bangalore
SRN: PES1201701084

Varun Venkatesh
Student, 2nd Year, B.Tech, CSE
PES University, Banashankari
Bangalore
SRN: PES1201701403

Dinkar Sitaram
Professor, Centre for Cloud Computing and Big Data
PES University, Banashankari
Bangalore

Savitha Murthy
Professor, Centre for Cloud Computing and Big Data
PES University, Banashankari
Bangalore

***Abstract*—Generative models are successfully used for image synthesis in recent years, but when it comes to other modifications such as audio, video etc. little progress has been made. In this paper we further the concept of a WaveGAN architecture, applying it to regional language datasets and generating audio samples for a varied dataset of mispronounced words and generating their correct pronunciations in the learner's voice. Generating correct pronunciations in the learners voice for isolated words in Kannada has two parts to it. Firstly, recognizing the incorrectly pronounced word and classifying it into word based groups which map to the correctly pronounced words. Secondly, modelling that correctly pronounced word and outputting it in the speakers voice itself. We used a double discriminator model to approach this problem. The first discriminator is trained on the word spoken, hence to map it to the correct pronunciation of the word. The second discriminator is trained on the speakers voice, thus modeling the audio to be generated with a correct pronunciation of the word in the speakers voice. The key point to be driven is the dataset operated upon is not of accurate words, but slightly mispronounced words. Our architecture drives this as a key application of the generative models to operate on such data and with a preprocessor, allow them to use these as accuracy models. This provides a huge boost in language learning tools as it can be used as an interface to teach the user in his language what the word he said is pronounced like.**

***Index Terms*—Generative Adversarial Networks, WaveGenerative Adversarial Networks**

I. Introduction

Dealing with the norms of traditional neural network models with audio has always been a tedious process, generating such audio an even bigger challenge to primitive networks. The advent of an adversarial approach boosted the advantages of using a model of a Generative Adversarial Network to both generate and classify audio based on feature sampling coupled with traditional convolutional networks. Although this is still not a widely used approach for audio generation, the strength of an adversarial framework enables the system to challenge itself with more examples, hence, giving the system a higher accuracy.

The existing models that use the adversarial frameworks for the generation of raw audio usually rely on the accuracy and the consistency of the audio samples provided to them. An accurate model generated for training is usually as accurate as it's average input accuracy. Here, however, the samples used are not completely accurate, making it a challenge to classify the words under correct pronunciations and incorrect ones.

The proposed model advances the research so far done with audio files by proposing a technique to classify and featurize audio that is not a hundred per cent accurate to the model described. The model designs on it's ability to generate the correct pronunciation for a mispronounced word by first classifying the mispronounced word and produces the correct pronunciation in the speaker's voice to guide the speaker on how to pronounce it than a trivial computer generated voice. In brief terms, it adds another discriminator network to the existing generative model to scrutinize the audio based on word-specific rules and speaker-specific rules, trained independently, theorizing that the output generated will be one that passes the constraints of both these models, bearing an accurate result.

A. GAN overview

GANs, deep neural nets consisting of two networks pitting against each other. Discriminative algorithms try to classify the input data, that is, given the input features which is essentially the data set, the predict a label or a category to which set that data belongs. Generator model generates new data instances starting from a random input tensor. This data is sent to the discriminator which classifies the data as fake or real, that is, the discriminator evaluates for authenticity. The adversarial architecture initially proposed assumed the use of multi-layered perceptrons as models in a dueling fashion. The generator distribution p_g is first modeled over data x, with defining noise variables, and representing this structure as a mapping to a data space as G(z, θ_g), where G is the differentiable function represented by a multi-layered perceptron with parameters θ_g.

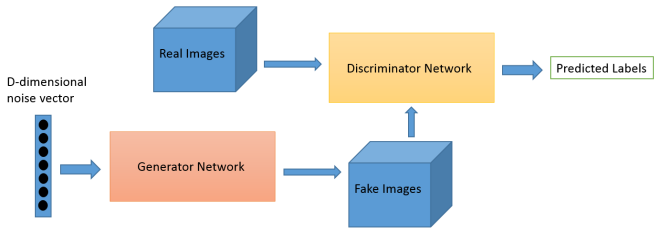

Fig. 1. GAN architecture initially proposed

Similarly another multi-layered perceptron D(x, θ_d) represents the data that came from the data rather than p_g. D is trained to maximize the possibility of assigning correct labels to data and simultaneously training G to minimize log(1-D(G(z)). The idea is to maximize the value of

$$\min_G \max_D V(D,G) = E_D + E_G \tag{1}$$

where

$$E_D = E_{x\ p(b)}[logD(x)] \tag{2}$$

and

$$E_{z\ p_z(z)}[log(1 - D(G(z))] \tag{3}$$

where E denotes the expectation of D and G respectively.

B. References

Since the field of GANs is relatively new and is still being further explored into, there are some other applications and references which can be implemented using the concept of GANs.

- Data Augmentation - The most obvious application with GANs is data augmentation. This is essentially training a model to generate new samples from the data to augment the data set. NVIDIA for example used this approach and showed an improve in results for sensitivity and specificity for brain CT images by close to seven percent by adding synthetic data augmentation.
- Domain Adaption - In practice we almost never have the exact data for training and running them in the real world environment. For example in Computer Vision, different camera angles, lighting conditions can make even the best model useless. GANs can generative all the different set of data to be used even in extreme conditions for training.
- Drug Development - While others apply generative adversarial networks to images and videos, researchers have proposed an approach of artificially intelligent drug discovery using GANs. The goal is to train the Generator to sample drug candidates for a given disease as precisely as possible to existing drugs from a Drug Database, generate a drug for a previously incurable disease and using the Discriminator to determine whether the sampled drug actually cures the given disease.
- Game Development - Game development and animation production are expensive and hire many production artists for relatively routine tasks. GANs can auto-generate and colorize any game graphics such as locations and characters. The generator and the discriminator composes of many layers of convolutional layers, batch normalization and ReLU with skip connections.
- CycleGAN - Cross-domain transfer GANs will be likely the first batch of commercial applications. These GANs could be used to transfer images from one domain to another. CycleGAN builds 2 networks G and F to construct images from one domain to another and in the reverse direction. It uses discriminators to critic how well the generated images are. For example, the generator converts real images to Van Gogh style painting and the discriminator is used to distinguish whether the image is real or generated, hence causing an adversarial type of learning.
- Security - A constant concern for industrial applications are cyber attacks. GANs are directly addressing this concern of "adversarial attacks". These adversarial attacks use many techniques to fool the deep learning architectures. A technique called SSGAN is used to do steganalysis of images and detect harmful encoding which should not have been there.
- Single Image Super-Resolution - We often face problems with low-resolution images as they are not clear, GANs helps us to create High-Resolution images from a single low-resolution image. For this problem a GAN called SRGAN is used. SRGAN is the first framework capable of inferring photo-realistic natural images for 4 up scaling factors. The discriminator is used to train how to differentiate between super-resolved and original photo-realistic images.

II. Related Work

The introduction of an Adversarial strategy using Generative Adversarial Networks (GANs) [2] opened avenues to use an unsupervised framework to map low-dimensional latent vectors to high-dimensional data. Data augmentation [5] has always been a strong use in speech recognition systems. They also provide a bonus of rapid and straightforward sampling of large amounts of audio.

Autoregression [3] has also been a useful way to operate on raw audio. However, an autoregressive model is quite slow compared to the GAN as it requires audio samples to be fed sequentially in a single-file fashion to generate new audio.

The work in the developement of a synthetic audio waveform has been closely linked to the idea of using encoded tensors of audio as raw waveforms and compiling them into waveforms and spectrograms [1] to model the data and parse to the generator, allowing the generator to model a function that transforms random waveforms to audio waveforms resembling that of the dataset. The WaveGAN provides a powerful way to flatten the deep convolutional GAN [4] architecture, giving the model a single dimensional approach to a two dimensional problem.

This idea is furthered in our research by making the functionality extend to audio which is not deemed as accurate.

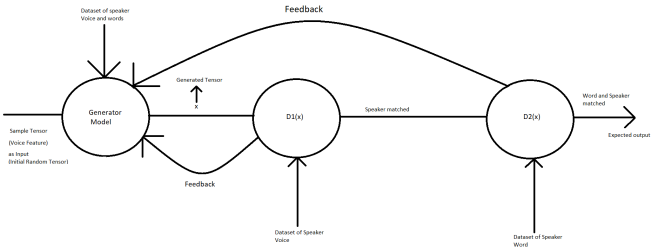

Fig. 2. Structure of proposed model

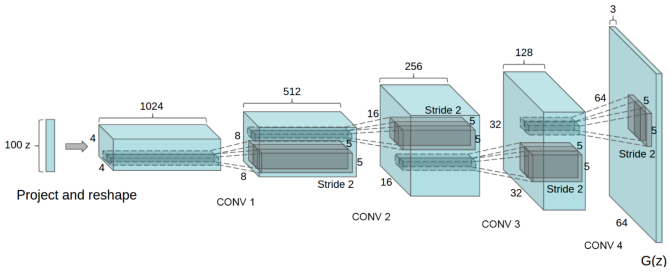

Fig. 3. DCGAN architecture

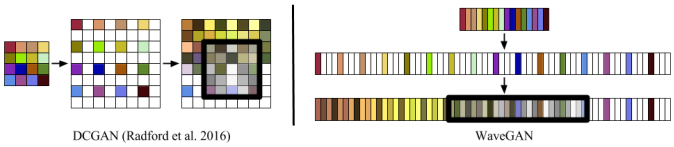

Fig. 4. WaveGAN modelling

The WaveGAN architecture remains the same, although two networks are added to this preceding architecture to better model the networks to work concurrently with each other.

III. Proposed Method

The model is essentially a stringent classifier for first classifying the audio files into word-based groups, where each group resembles the correctly pronounced word, modeled by features encompassing all the audio files belonging to the same group. The main GAN architecture uses the existing WaveGAN model proposed coupled with an additional discriminator. The two discriminators are identical in nature, but trained separately. The first discriminator is trained on samples modeling the word spoken, and the second trained on samples of the speakers voice. The proposed architecture models the to-be-generated audio to pass both the tests to give an output that closely matches the correct pronunciation in the speaker's voice.

Figure 4 shows the basic structure of the proposed model.

A. WaveGAN architecture with proposed modelling

The WaveGAN architecture is mostly the same architecture as proposed in the DCGAN model applied to audio synthesis instead of the traditional image synthesis. The DCGAN approach used a transposed convolution operation to iteratively upsample low-resolution feature maps into a high resolution image. The same technique is used in the waveGAN model, but with a widened receptive field by using longer one-dimensional filters of length 25, instead of two dimensional layers of 5x5, upsampled by a factor of 4. The discriminators are updated in a similar fashion, using 25 one-dimensional filters with a updated stride of 4. Instead of using the normal 64x64 sample that the DCGAN utilises, which only models about 4096 samples, an updated additional layer allows 16284 samples, slightly more than a second of audio at 16kHz.

The second layer is an exact replica of this layer, but these two models are trained independently, and instances of these two models are used in the final architecture. An extension to this model is the audio classifier, which is a convolutional neural network model[7], whose implementation will be not discussed here.

The audio database is sampled through this classifier, which makes sample groups of audio samples having similar features based on standard audio features checked by the audio classifier, and the updated data is classified further onto similar voice models and similar word models. A deterministic and rather manual approach has been used as of now to detect the speaker speaking, by coupling the input with the audio classifier to find the pool of voices the speaker belongs to. We make a formal assumption here that the speaker's voice is already modeled by the audio classifier. The discriminators are trained on the voice samples and word-based audio samples, each trained seperately, and the final models are used coupled with a generator with little initial training on the whole audio dataset. The generator adversarially trains with these two discriminators, and over the training period provides an output that represents a combination of true data from both the discriminators. The generator of the model proposed gives slices in lengths of 16,384, with a standard dimension of 64. The kernel length remains at at 25. Reshaping of this for convolution upsamples every 100 dimensional vector into a 16x1024 vector matrix for convolution. Similarly, every layer of the architecture is upsampled for consistent convolution. Five layers are used in the initial generator model. A leakyRELU activation is applied on the architecture with a phaseshuffle constraint. The discriminator architecture has a kernel length of 25 and the dimension of latent vectors extends

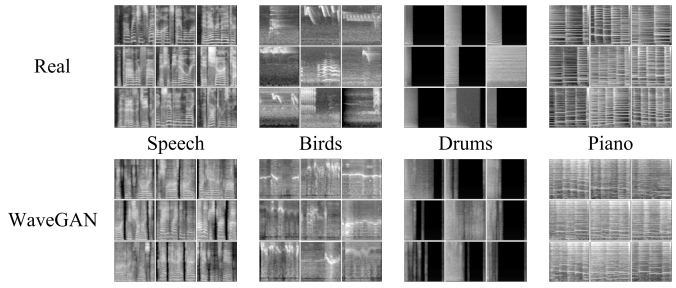

Fig. 5. Independent training sets and results on WaveGAN proposed by [1]

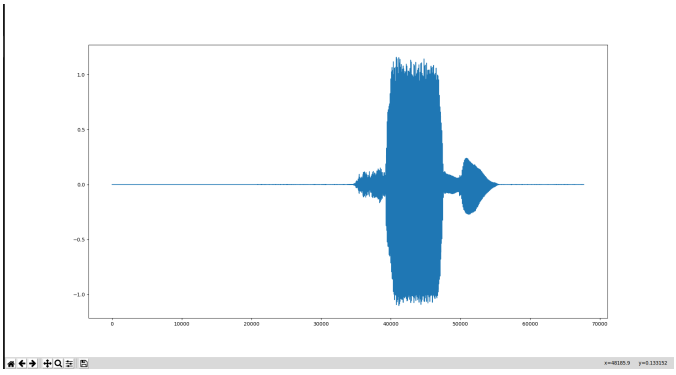

Fig. 6. Real audio used for testing

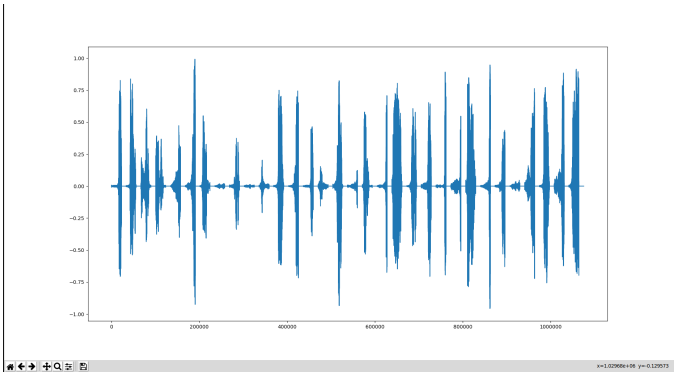

Fig. 7. Generated voice model

to 64, just as the generator. No batch normalization is applied as the layers are downsampled from the [16384,1] sample to a [16,2084] sample via intermediary downsampled layers, to a final dimension space where the comparison occurs and the layers are flattened by reshaping them. A double discriminator model is applied here. Both the discriminators apply the same architecture described above. The whole architecture is trained concurrently, but the networks referencing different data sets. The generator works on the entire dataset containing audio samples of speakers speaking in Kannada, in discrete words that are all spoken uniformly. The generator tries to model a generic function that can generate words resembling the words it sees in the voices it sees using a many-to-many mapping. The first discriminator trains on the dataset containing words only. It samples functions for each word and collects information on how to distinguish one word from the other. The second discriminator works in a similar fashion, but with speaker voices rather than words. The architecture trained simultaneously is theorized to give audio samples that filter through both the discriminators of voice and word to give the word mispronounced in the speakers voice. The model assumes that the words are all correctly pronounced initially, then upon training is able to map a feature graph of what each word has in common with other audio samples in it's pool. It is highly unlikely that every speaker pronounces a word incorrectly in the same exact time frame. If it is observed that two or more speakers use the exact same type of pronunciation for a specific time stamp, the model assumes that the pronunciation for that word at that time stamp is the correct pronunciation as given. Hence the discriminator for words is able to correctly map the pronunciations to real pronunciations. This graphing is done independently and for testing uses a correctly pronounced words data set for the same words to compare it's outputs generated.

IV. Experimental Data and Results

Due to the nature of the problem and its key dependency onto natural language and speech, the dataset obtained of real genuine speakers in a regional language was minimal. Trained on a dataset of about 20 speakers, each speaking 25 words each, the model was able to obtain close to 62% accuracy upon training. Inferences can be drawn from 6 and 7, where the generated waveform can be split into discrete waveforms that try to model the real waveform via transformative functions the generator understands after training. The training was done on a GTX 1050Ti for about 7 hours to yield these results. There were some initial overfits but the model trained gradually on the dataset yielding fairly accurate generative models. The figure 7 depicts the same word spoken by a speaker over and over. The model makes a set of checkpoints every batch inputted, making about 16,000 checkpoints over the lifetime of the training. The key inferences drawn from the architecture drawn is its unique ability to take mispronounced words as input and then find the correct pronounciation and its ability to generate the correct pronunciation of the spoken word in the speaker's voice. Usually these network architectures deal with highly accurate data models. But the real world models are very far from accurate most of the time, so the modelling of real world models is done much more closely using an architecture such as this, especially for audio featurizing, where there is not much work done to collect an accurate dataset of spoken word in regional languages. The project symbolises a huge solve for regional language learner tools, and allows people in countries of multiple dialects, languages and different voices to communicate seamlessly with one another. The model depicts reusability in real time as the model once trained can be used as an application interface in existing language learning tools.

Acknowledgment

The project comes with immense support from the Centre of Cloud Computing and Big Data, PES University, with our mentor professors playing a key role in building the proposed architecture for the specification given. The previous work done on the model, especially the work done on the WaveGAN [1] has been instrumental to our work, and the model proposed would not be possible without the initial architecture provided. The Computer Aided Pronunciation Tool mentioned also played a key role in obtaining a usable dataset for the model to train on.

REFERENCES

[1] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *ICLR*, 2019.

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[3] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech, 2018.

[4] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.

[5] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. Sgm: Sequence generation model for multi-label classification, 2018.