Domain Specific Text-to-Speech with Machine Learning

PlaneEnglish

Joy Gao, Yuan Gao, Laura Marie Lopez, Karthik Ravishankar, Grant Rivera, and Saurabh Singhal

BACKGROUND

PlaneEnglish develops training solutions for pilots, including ARSim, an aviation radio simulation app. ARSim allows pilots to practice communicating with Air Traffic Control (ATC) using interactive lessons. **Our goal** is to find a way to generate speech that sounds more like an air traffic controller voice than a typical voice for ARSim.

We used the following approach:

1. Research existing text-to-speech models that fit our needs
2. Find ATC data for training and develop tools for formatting
3. Train model with our aviation-specific dataset
4. Evaluate performance and make necessary adjustments

OVERVIEW OF NEURAL TEXT-TO-SPEECH

Recent models vary in scope and can accomplish tasks in end-to-end processes

- Process starts with input text which is converted to phonemes and finally to waveform audio
- We focused on Tacotron2 and FastSpeech2 which are acoustic models
- These models are recent enough to offer performance benefits but established enough to have a strong support community

MACHINE LEARNING FOR TEXT-TO-SPEECH

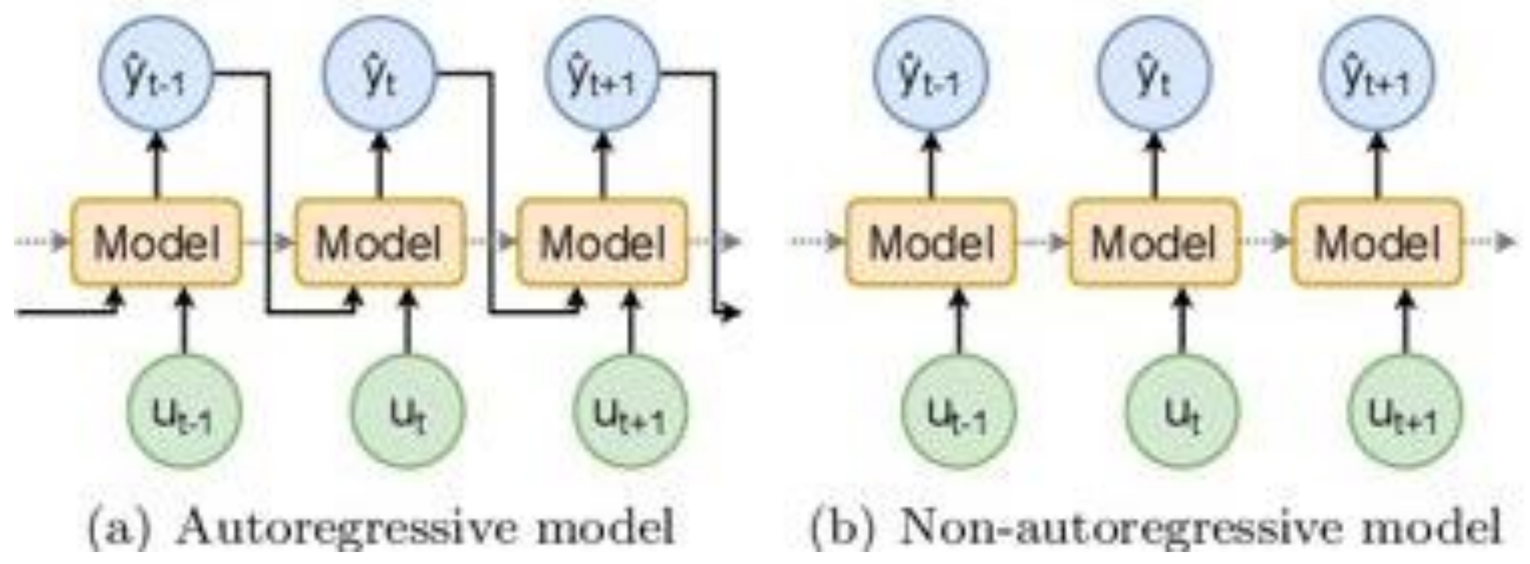
Data Collection and Training on Domain Specific Data

- The LJSpeech dataset links audio data to corresponding transcripts
- Developed tools to convert both the audio and text data into the LJSpeech format

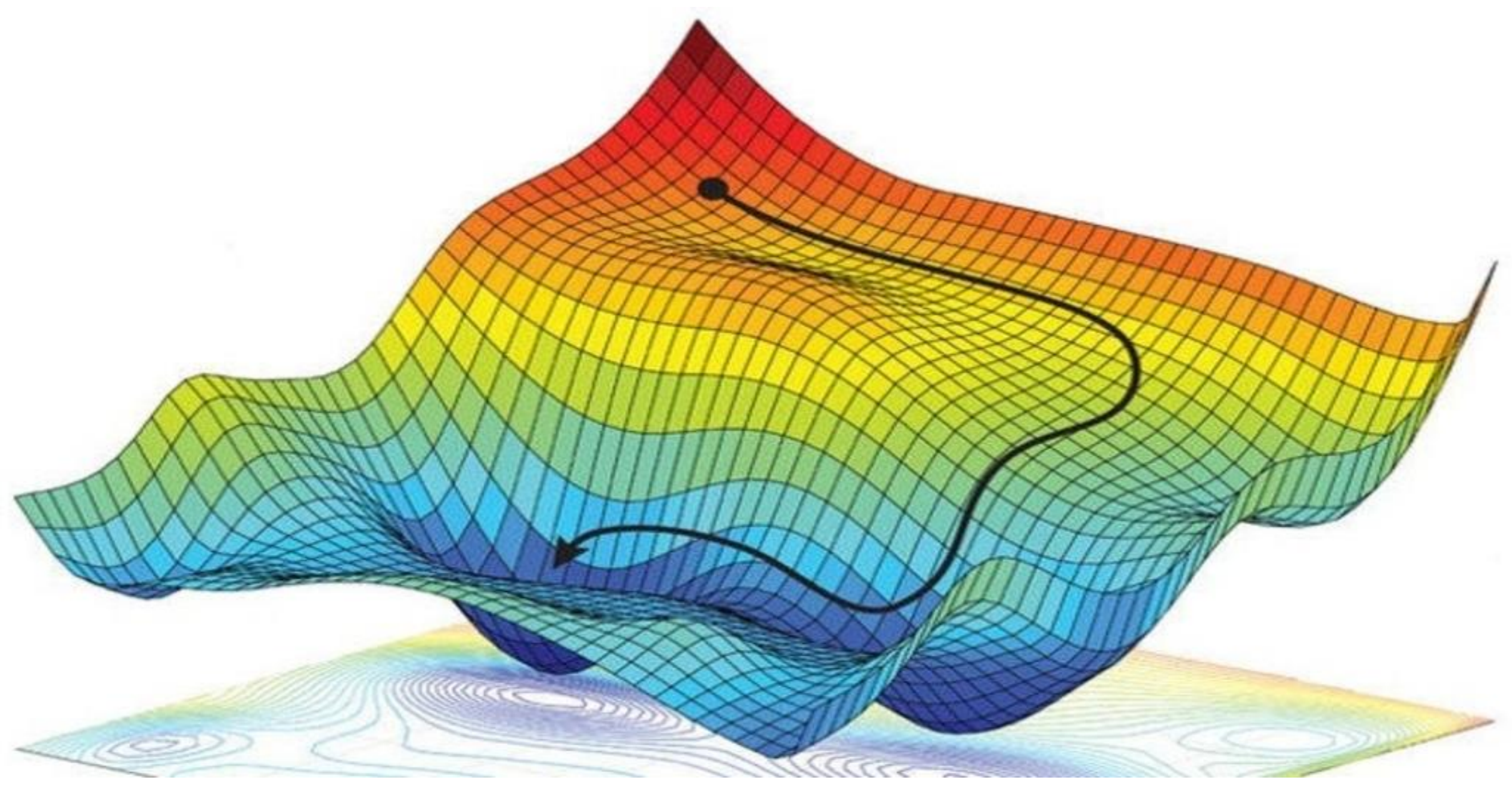
Training Process and Improving Performance

- Trained Tacotron2 model using ATC data with around 1,100 samples
- Model struggled initially, but by adjusting some parameters and giving the model more time to train, noticeable improvements were made
- We noticed that performance eventually plateaued, possibly due to the model exhausting all of the training examples in our dataset, which is relatively small for a TTS dataset (more on this in Future Goals)

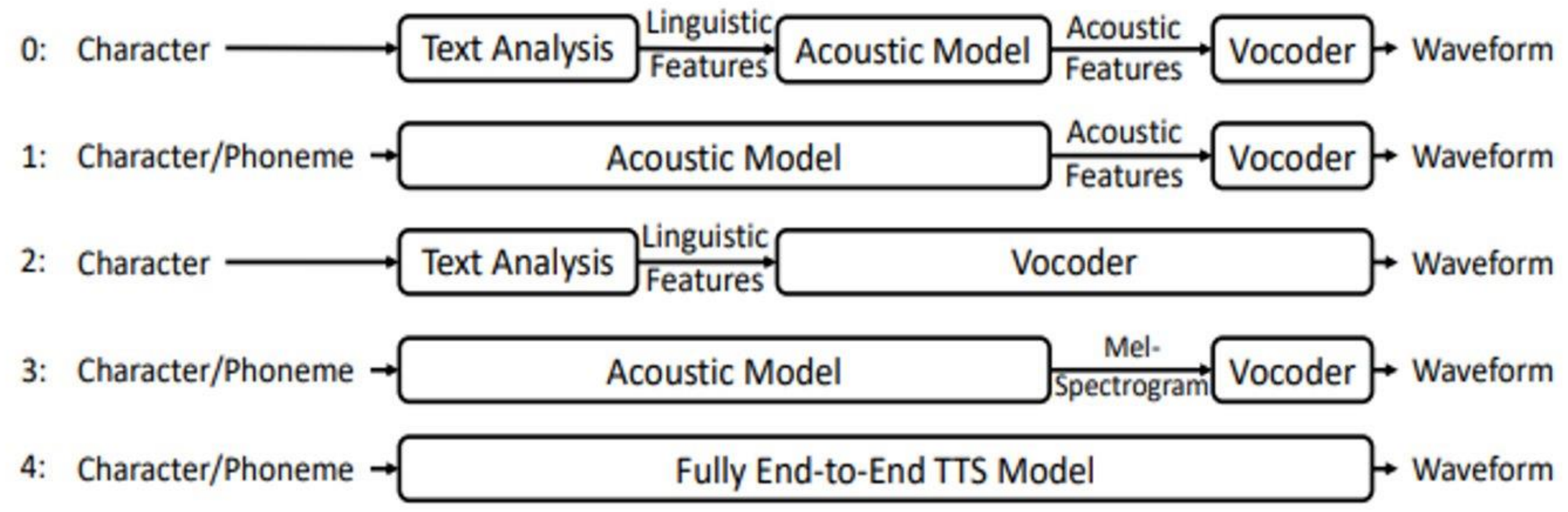
Examples of a recurrent neural network, which tracks its previous state to have a short-term sense of "memory."



A neural network tries to minimize its "loss," which is a function of its error with respect to its weights and biases. Sometimes the loss can converge at a local minimum.



Some types of text-to-speech models and their scopes; we went with acoustic models, which offer flexibility and lots of available documentation



RESULTS AND OPTIMIZATION

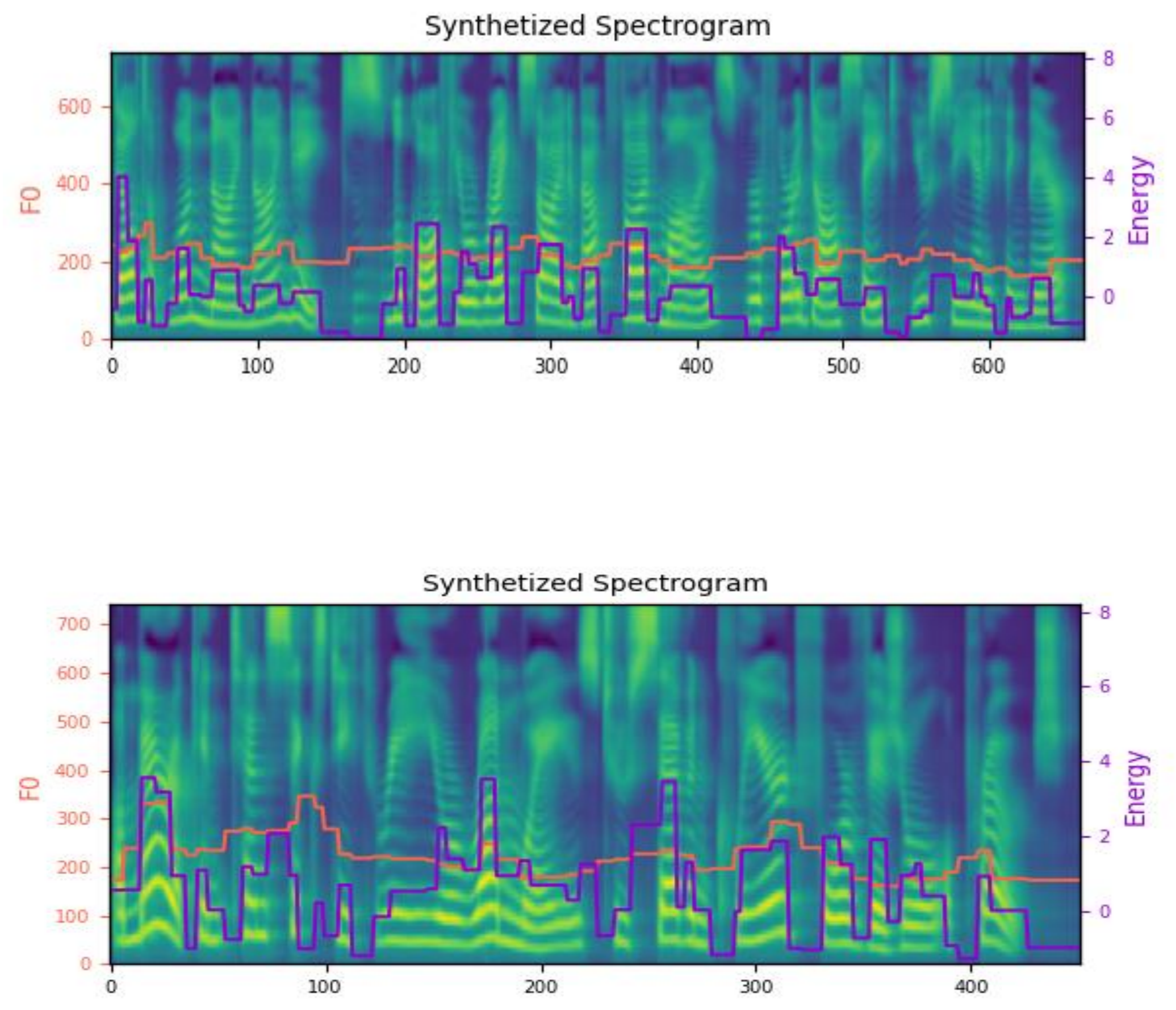
Evaluating Performance

- Model uses an error function (or a loss function) to judge how far it is from desired output
- Helpful for training, but can't capture subjective changes in voice audio quality

Generated Samples

- We tested our trained model with several sample phrases specific to aviation
- Improvements in key areas, such as a voice that sounds closer to an air traffic controller, as well as some typical noises in ATC communication
- Changing parameters helped improve some aspects of speech, but performance limited by dataset size

Example spectrogram outputs from the FastSpeech2 model, which is then used by a vocoder to generate waveform audio



REFERENCES AND AKNOWLEDGEMENT

Corporate Partner Mentor: Muharrem Mane

Corporate Partner TA: PK, Yuantong Li

Data Mine Staff: Kevin Amstutz, Shuennhau Chang, Dr. Ward, David Glass

ONGOING EFFORTS AND FUTURE GOALS

Optimizing Response Time

- Apply our domain specific data to the FastSpeech2 model which generates audio much faster than Tacotron2
- The goal is to generate speech quickly enough for use in a simulation

Measuring Voice Quality

- Neither loss or subjective evaluation fully capture output quality
- Developing a framework based on existing ATC fluency standards could achieve this
- ICAO language proficiency accounts for words per minute, filler words, other factors

Data Collection

- A larger dataset could help the model continue to improve
- A standard dataset for neural TTS, the LJSpeech dataset, has about 25 hours of training data
- Our model has shown improvements with only 2 hours of data, so looking for more is worthwhile

The Data Mine Corporate Partners Symposium 2022