Introduction

Tacotron 2 is a text-to-speech system developed by Google. It's used in the website uberduck.ai, which is a platform with voice models for various characters. It's also used in the Japanese TTS TALQu, made by Haruqa.
There are beginners' guides out there for making English models for Tacotron 2, but I couldn't find any clear guides on Japanese speech synthesis, so I researched and experimented and eventually figured out this method. I hope this guide can help you develop your own Japanese speech model!

Part 1 - Recording

- Record/gather your audio. If you'd like, I have a reclist and transcription based on the list included with SHABERU.
- Save your clips as 16-bit 22050hz mono wav files.
- Do not include spaces in filenames. Files should only include alphanumerics (half-width) and underscores. This means no Japanese filenames.
- If you're working off of long audio segments such as a speech or dialogue from a game, separate each clip by sentence. If a clip is longer than 10 seconds, split it up so it's shorter.
- Ideally you should have at least 15 clips, but even that won't yield great results. Try to have at least 15 minutes of audio.
- Do not include unclear audio, such as screaming, whispering, and laughter. Remove any background music and noise.

Part 2 - Transcription

This is probably the most tedious part due to the phoneme notation used. It's mostly the same as normal romaji, but there are some exceptions, so please use the correct g2p. If you wish to use a different phonetic system, such as the default openjtalk phonetics, you can modify my g2p or make your own (if you know what you're doing). Neutrogic also has an ljspeech-format g2p with default pyopenjtalk phonetics.

1. To make a transcription file, create a new text document (.txt). Open the document, and add lines in the following format:
   - wavs/<name_of_file>.wav|[Japanese text here]
   - Japanese text can be in kana or kanji, but *not* in romaji.
2. Open my Japanese g2p for TALQu phonetics.
3. Run the cell to install pyopenjtalk. Wait for it to finish executing.
4. Upload your transcription to the file tab and right click on it. Rename it to "list.txt".
5. Run the Option 1 cell.
6. Right click and download the file called "written.txt". This is your transcription.

Part 3 - Training

Congrats, the hard work is behind you! Now you just need to be patient during training. It can take several hours. My port of KYE took four hours to train! Despite it taking a long time, it's a pretty easy process, though. Make sure each cell is finished running before you move onto the next step.

1. Open the Japanese Tacotron 2 Training notebook.
2. Run the cell called "Check GPU type". If you get the K80 or P4 GPU, go to the Runtime menu at the top of the page and select "Disconnect and delete runtime" then run this cell again. You don't *have* to do this, but if you get the right GPU it speeds up training a lot.
3. Run the cell called "Anti-Disconnect for Google Colab".
4. Scroll down and run the cell called "Mount your Google Drive". It will prompt you to connect your Google account.
5. Run "Download pretrained model and install tacotron 2". Click on the folder icon on the left sidebar. You should see a folder inside there called "wavs". Drag and drop your wav files into that folder and wait for them to upload. You'll know it's done when the list of files with orange progress wheels is gone. If you have a lot of wav files or a slow upload speed, zip up your wav files locally and run the optional "Unzip file" cell.
6. Set your model parameters in the "This is for your training configuration" cell.
    - transcription: Upload your transcription file to tacotron2/filelists. Right click on the uploaded file and select "Copy path". Paste the path into this box.
    - batchsize: Ideally, set this to an amount that cleanly divides into the amount of wav files you have. You can Google "factors of [amount of wavs]" to find out what some possible amounts could be. For example, I recently trained a model with 15 wavs, and I set the batch size to 5. You don't need to set it to a size that divides cleanly, but it is better when possible. If you have the T4 GPU in your Colab session, the batch size should be 14 or less.
7. Run the "Launch TensorBoard" cell. You'll be able to look at and refresh the images tab to see how training is going. If the graph looks like a straight diagonal line, training is going well.
8. In the cell called "Begin training", change "output_directory" to a folder in your drive where you want to save your model. Stop the cell once your model has saved at the proper validation loss, which will display as the training continues. **Do NOT let the model train until the epochs have run out!**
    - A guide for validation loss: less than 30 files = under 0.07; 30-100 files = under 0.09; 150+ files = under 0.1; more than 30 min of data = under 0.14
    - You can see when the model saves because it will display "Saving model checkpoint to [path]".
    - As the training progresses, your graphs in the images tab on TensorBoard should begin to look like a diagonal line. If the diagonal line cuts off, that's okay - it just means the graph is plotting a shorter file. If your graphs don't begin to look like a

diagonal line, your recordings are low quality, you have too little data, your transcription is wrong, or you have too much silence at the ends of your files.

Part 4 - Fine-tuning HiFiGAN

This step is optional, but it's very useful for making your model sound significantly better. Without this step, you'll get weird artifacting during your synthesis, and your model won't sound very realistic.

1. Open the HiFiGAN fine-tuning notebook.
2. Run Step 1. If you get the P4 or K80 GPU, select "Disconnect and delete runtime" from the Runtime menu at the top of the page, then try again.
3. Run Step 2 and authorize Colab to access your Drive.
4. Upload a zip file with the wavs used for training your model to the Colab session. Upload the transcription used, too. Set the parameters in the text boxes, then run the Step 3 cell.
    - tacotron_model: Find your trained model in drive/MyDrive and right-click on it, then select "Copy path".
    - tacotron_dataset: Right-click on your zipped wavs in Colab, and copy the path.
    - train_filelist: Copy the path of your transcription txt.
    - val_filelist: Same as train_filelist.
5. Run Step 4.
6. Run Step 5.
7. Change "output_dir" in the Step 6 cell to a place in your drive where you want to save your HiFiGAN model. Then, run the cell. Stop it after it gets beyond 5,000 steps.

Part 5 - Synthesizing

You're finished creating your model! This is the step where you get to hear it talk. There are two main ways to do this: in Colab, and in TALQu.

- Synthesizing in Colab:
    - Open the Tacotron 2 synthesis notebook.
    - Run the cells in the "Setup to run on CPU" category.
    - Change the params in the "Inferencing" cell, then run the cell.
        - tacotron_id: Find the latest checkpoint of your model in Google Drive (you can delete the old checkpoints), and select share. Change to "Anyone with the link", then copy the part of the link with random numbers and letters and put it in this text box.
        - hifigan_id: Do the same as for the Tacotron model, but with the g_00000000 file generated by the HiFiGAN notebook. If you didn't follow that part, leave it as "universal" or use the HiFiGAN model from a different Tacotron model.
        - pronounciation_dictionary: Check off the check box.

    - After you run the Inferencing cell, it will load for a bit before a text box shows up. Use the g2p in Part 2 to get the phonetic input for the text you want to synthesize (this time, use the Option 2 cell). A sentence I use to test is the following: “myizuomareesyiakarakawanaktewanaranainodes.”
    - It will output audio, which you can play back or download. You can enter another sentence after that, or stop the cell.
- Synthesizing in TALQu:
    - Download and install the latest version of TALQu.
    - Download your Tacotron model, as well as the g_00000000 file and config.json generated by the HiFiGAN notebook (unless you didn’t follow that part).
    - Make a new folder in TALQu/Models and name it whatever you want. Put your Tacotron model, g_00000000 file (if applicable), and config.json file (if applicable) into the folder. Rename config.json to “g_00000000_config.json”.
    - Copy a Config.csv model to the folder and edit the information in it:
        - SpkName: The name you want your model to have with TALQu.
        - Icon: If you’d like, add a square .png file to the folder. Put the name of the png in this entry. If you don’t have/want an icon, delete this line.
        - Author: The name of the model’s author (probably you).
        - URL: A website either related to the model or related to the other, followed by a comma, then a small description of what the website is.
        - NTacotron2Model: The name of your Tacotron 2 model.
        - HiFiGANModel: “g_00000000”
    - Open TALQu, and type in a sentence you want to synthesize. One I use for testing is: “水をマレーシアから買わなくてはならないのです。”
- Synthesizing with Uberduck:
    - Go to https://app.uberduck.ai/submit.
    - Fill out the form with the applicable information.
    - Wait for your model to be approved.
    - Use your model at https://app.uberduck.ai/speak#mode=tts-basic.
        - Use phonetic input, which can be generated from plain Japanese text using the old version of TALQu.

Contact

Contact the author of this tutorial with questions, suggestions, or concerns.

- Email
- Discord: Kei Wendt#6210
- Twitter
- Soundcloud

References

- Uberduck Tacotron 2 training notebook

- Japanese speech synthesis starting with Tacotron2
- Tacotron 2: Generating Human-like Speech from Text
- TALQu
- TACOTRON FOR VOCAL SYNTH USERS
- SHABERU
- FlatBaseModel (Japanese pretrained model)
- PyTorch implementation of Tacotron 2