Summary: Data Engineering

What is Data Engineering?

Data engineering is a critical component of supervised learning, and consists of defining requirements, recording data, processing it, and improving the dataset. The quality and quantity of collected data determines the tractability of a machine learning objective. Training examples need to include enough salient features, along with representative noise induced by the surrounding environment (for example, day or night in images, or quiet and loud background noise for audio), for an ML algorithm to accurately distinguish between classes when deployed into the real world.

Data engineers need a rigorous problem definition in order to know what data should be collected, and must identify the potential sources of data. Data might come from on-device sensors, product users, or paid or unpaid contributors, and each may introduce potential licensing or privacy restrictions. This data must be labeled, and this usually requires manual effort by individual workers. It may also require domain expertise, for example, when labeling medical images. Mislabeled or garbled data may also need to be filtered out through manual inspection. Data engineers must also manage changing needs for a dataset, for example, in order to support additional languages.

Speech Commands

Speech Commands is a keyword spotting (KWS) dataset developed by Pete Warden at Google, and we will consider it as a practical example of the steps required in TinyML data engineering.

Requirements: It established a new standard for public, comparable keyword spotting research - previous datasets for KWS were often restrictively licensed or proprietary to individual companies. Speech Commands is available for anyone to use, and allows ML researchers to compare their ML algorithm's performance on the same data.

Collection: Speech Commands contains thousands of recorded examples for 35 keywords, collected by over 2,600 volunteers with a variety of accents. All volunteers agree to have their voices redistributed. Data collection was done in the browser, since requiring installation of an app might discourage contributors.

Refinement: As with any dataset, some data collected will be unusable (for example, if an incorrect word is spoken or the microphone gain was too quiet). Some automated techniques were applied (removing low volume recordings and extracting the loudest 1sec from 1.5sec examples), but the remaining 105,829 recordings were manually verified through crowdsourcing.

Sustainment: Speech Commands has been expanded to 35 keywords in Version 2, from the original 25 words in Version 1. Care was also taken to ensure the same recordings remain in the same train, validation, and test splits across the two versions.

The careful construction of the Speech Commands dataset has allowed keyword spotting researchers to compare the performance of different TinyML neural architectures on the same reference data. This benefits many aspects of KWS research, such as reproducibility.

Crowdsourcing Data for the Long Tail

Speech Commands only contains English data, but many languages are spoken across the world. Paying contributors to record and verify keyword data for every language rapidly becomes cost-limited. Companies which pay to collect speech data may prefer to keep this data in-house and proprietary. An alternative approach is crowdsourcing speech data.

Community contributions have led to the success of other open-source projects such as Linux and TensorFlow, and this model can also work for dataset generation. CommonVoice is a Mozilla-led effort which seeks to attract community contributions for speech data. So far, over 50,000 volunteers have contributed speech data in 54 languages to CommonVoice. A key draw for potential contributors is the promise of bringing modern advancements in speech processing to underserved languages.

CommonVoice data is permissively licensed for anyone to use. The majority of data in Common Voice consists of full sentences read aloud by volunteers, and verified to be discernible by other volunteers through a voting process. CommonVoice also contains a single-word target segment dataset (in the same style as Speech Commands) which contains recordings in 18 different languages.

CommonVoice is an ambitious project to bring automatic speech recognition, voice-based interfaces, and other speech processing technology to the whole planet. Importantly, users can add support for new languages without needing software engineering expertise, as CommonVoice also provides tools for translating their data collection interface to new languages. For data engineers, CommonVoice provides a useful example of how to expand data collection to a worldwide scale.

Repurposing Existing Datasets for TinyML

We've seen how challenging it can be to collect brand new datasets. ML research has exploded in popularity and there are many datasets available for a wide variety of tasks already. TinyML research can be accelerated by taking advantage of these existing datasets and repurposing them for embedded tasks. A useful overview of available datasets is provided by TensorFlow's Datasets Catalog, which covers a wide gamut of machine learning problems, and are ready-to-use for training with the TensorFlow API.

In the next section we will develop a vision-based TinyML tool for person-detection. The dataset for this task, Visual Wake Words, was developed from an existing popular dataset, Common Objects In Context (COCO). By specializing an existing dataset, you can avoid the need to collect data from scratch.

We will also discuss transfer learning in the next section. Transfer learning allows researchers to avoid training models from scratch, since the same features are shared across many tasks in speech or vision. It is faster to transfer features than to train from scratch, and transfer learning also greatly reduces the amount of data which needs to be collected for a new task. Pretrained models can also be used to help curate data for new tasks - ImageNet was built in part using results from Google image search. You may even be able to generate data for your needs using a simulator.

Responsible Data Collection

Data engineers must consider how to reduce bias when collecting datasets. We have seen several potential sources of bias in speech recognition - for example, your data may not contain enough examples from various age groups, accents, or gender. Efforts such as CommonVoice attempt to tackle the problem of underserved languages. Data engineers must determine how to debias data collection efforts for a given ML task, and additionally ensure these debiasing requirements are maintained if an intermediary is used in data collection (for example, paying gig workers to collect data on Amazon’s MechanicalTurk)