Learning with audio data

Tom Södahl Bladsjö

October 23, 2023

My goals:

- Learn about a new kind of data
  - Preprocessing
  - Augmentation
  - Model architectures
- Get comfortable with more complex model architectures
- Training procedure
  - Logging
  - Checkpointing
  - Early stopping
- Sequence prediction (for ASR)

What is sound?

Pressure waves (generally in air), which

- have amplitude and frequency
- can form complex wave patterns when waves of different frequencies occur together

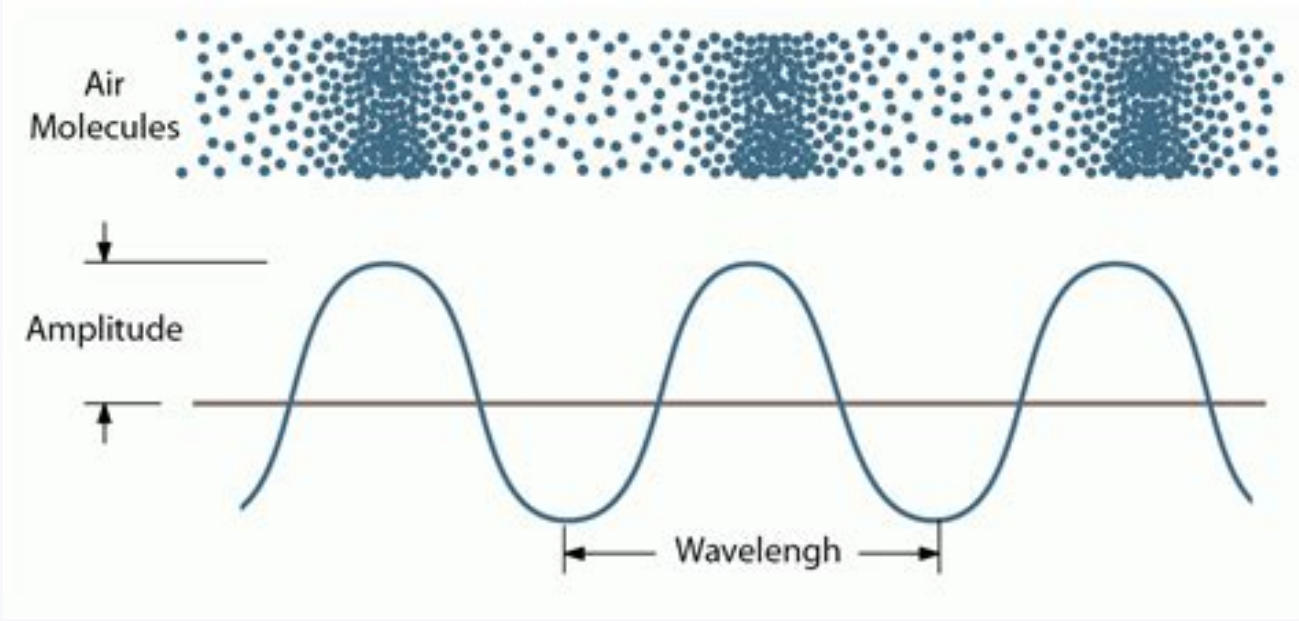

Image source: `https://www.soundproofingcompany.com/soundproofing101/what-is-sound/`

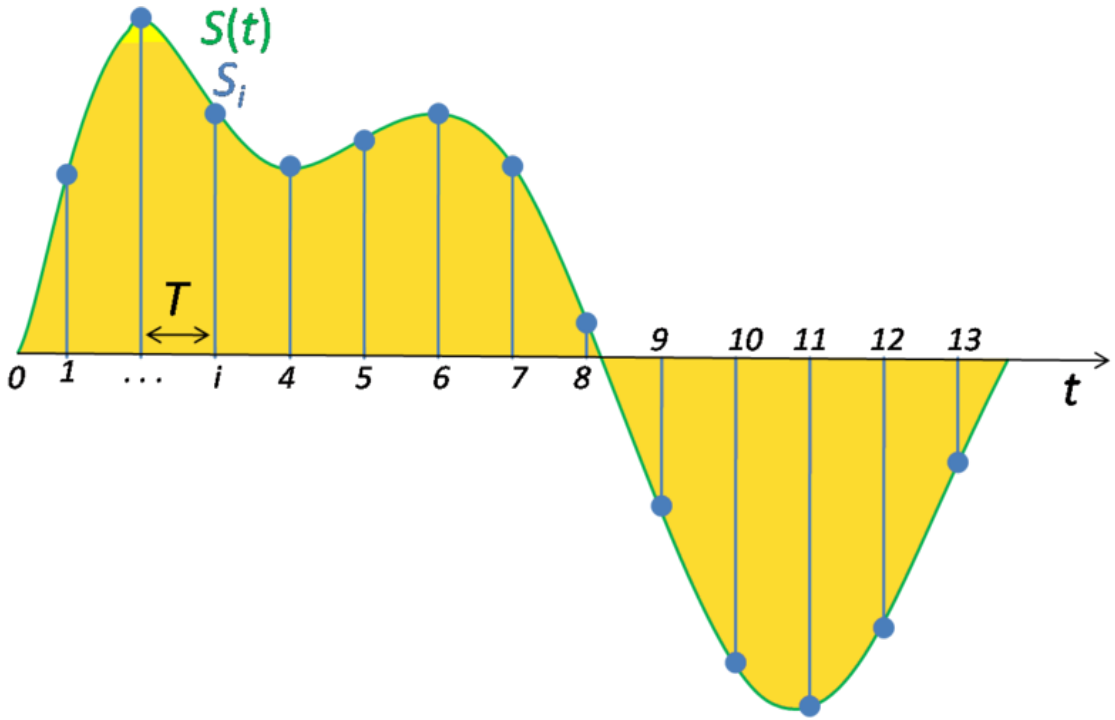

Figure: Sample measurements at regular time intervals

Image source: `https://commons.wikimedia.org/wiki/File:Signal_Sampling.png`

Fourier Transforms

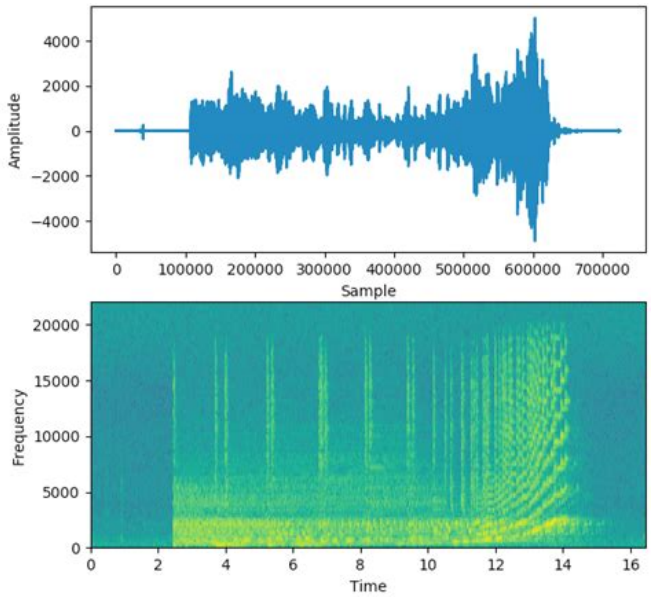

Figure: Amplitude vs frequency

This is an image...

...so we can use a CNN to extract feature maps!

Image source: `https://ketanhdoshi.github.io/Audio-Intro/`

ASR then and now

- Combined acoustic models and HMMs
- End-to-end architectures
  - CNN + RNN
  - Transformers

Training vs inference

- CTC-loss
- CTC-decoding, which can be done with or without...
  - ...a lexicon file
  - ...beam search
  - ...a connected language model

Testing

- WER

First experiment: Birds

Image: https://tailandfur.com/

Data

Warblr dataset

- Crowdsourced data
- 10,000 ten-second smartphone audio recordings
- 80/20 train/validation split
- Binary classification – is there or isn't there birdsong in this audio clip?

Model and training

CNN + Linear classifier

- 2 times CNN + ReLU + maxpool
- 2 Linear layers with ReLU

BCE loss with logits

Problem: label imbalance

- Accuracy never got above 0.7
- ...which turned out to be the proportion of positive samples in the data
- The model only ever predicted positive!

Solution: weighted loss

- torch.nn.BCEWithLogitsLoss has option positive weighting
- I could weight the loss by the proportion of positive samples in the data
- ...which improved performance **a lot**

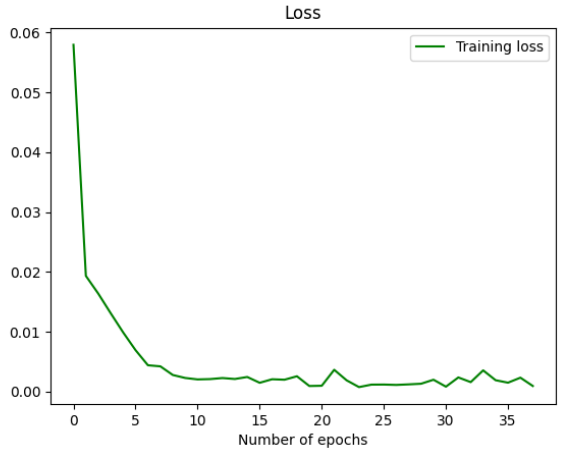

Figure: Training loss per epoch

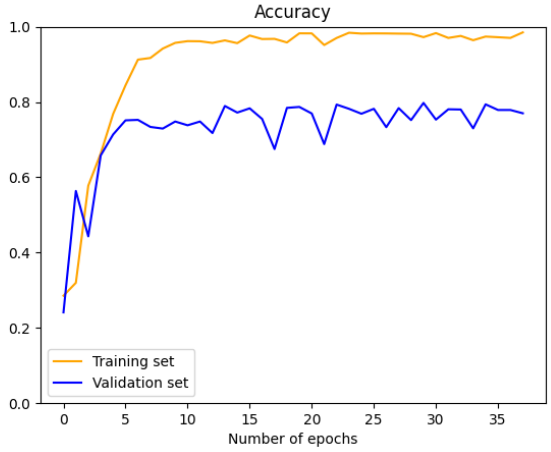

Figure: Accuracy per epoch on train- and validation data

Second experiment: Emotions

Image: https://dagshub.com/kingabzpro/EMOVO

Datasets

- EMOVO Corpus
  - Recordings of emotional speech by 6 actors (3 female and 3 male)
  - disgust, joy, fear, anger, surprise, sadness, neutral
  - I only used joy, anger, sadness and neutral (to match with the other datasets)
  - ... which amounts to 336 samples
  - uniform distribution of labels
- URDU dataset
  - Emotional utterances gathered from Urdu talk shows
  - anger, joy, sadness, neutral
  - 400 samples
  - uniform distribution of labels
- Estonian Emotional Speech Corpus
  - Recordings of emotional speech by a female voice
  - anger, joy, sadness, neutral
  - 1,234 samples
  - non-uniform distribution of labels, but not wildly imbalanced

Augmentation

- Time shift
- Time- and frequency masking
- Both performed on each sample = increased dataset size by 200%

Credit to `https://gist.github.com/ketanhdoshi` for augmentation examples in code

EMOVO corpus: `http://voice.fub.it/activities/corpora/emovo/index.html`

URDU dataset: `https://github.com/siddiquelatif/URDU-Dataset/tree/master`

Estonian Emotional Speech Corpus `https://metashare.ut.ee/repository/download/4d42d7a8463411e2a6e4005056b40024a19021a316b54b7fb707757d43d1a889/`

Experiment

- Three identical models:
  - 2 times CNN + ReLU + average pooling (the second being adaptive average pool to handle varying input sizes)
  - Linear layer + Tanh
  - 2 layer biLSTM
  - Linear classifier
- Each trained on one of the datasets

  ...and then tested on all three

Results

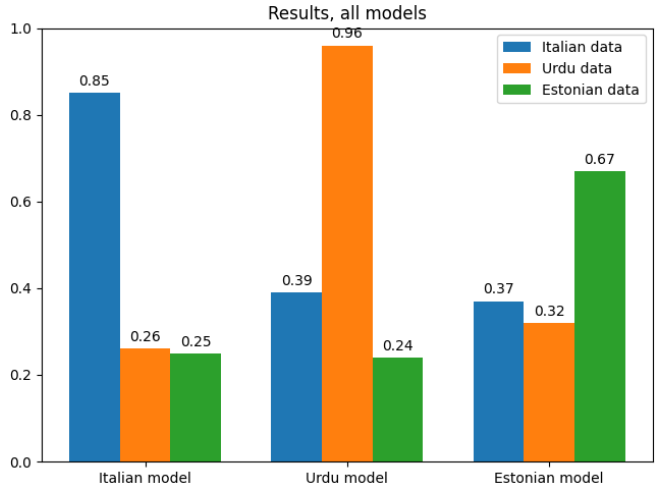

Figure: Accuracy of all three models on the different datasets

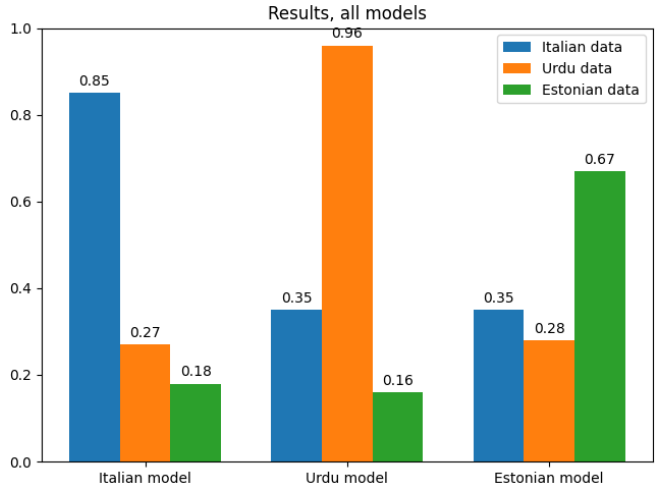

Figure: F1 of all three models on the different datasets

Third experiment: ASR

Image: https://developer.nvidia.com/

Data

LibriSpeech ASR Corpus

- Audiobooks from the LibriVox project, segmented and aligned
- Contains pre-divided train, dev and test data
- "Clean" and "other" (more challenging) speech data
- Total approximately 1000 hours, I used the smaller 100 hour train set

Model and training

- CNN + LSTM architecture
  - 3 times CNN + ReLU + average pooling
  - Linear + ReLU
  - 4 layer biLSTM
  - Linear classifier + logSoftmax
- CTC loss

Difficulties

- Training loss stops decreasing after first epoch
- Validation loss fluctuates but does not seem to decrease or increase
- Model does not seem to be learning (even after 100 epochs)

Reasons??

- Model architecture
- Batch size
- Learning rate (Using a scheduler might have helped)

So...

...I have not yet tried inference.

This means:

- no decoding
- no beam search
- no connecting an external language model

Conclusions

Revisiting my goals

Did I...

...learn about a new kind of data?

- Preprocessing — Yes
- Augmentation — Yes
- Model architectures — Yes

...get more comfortable with more complex model architectures? — Yes

...work on training procedures?

- Logging — Yes
- Checkpointing — Yes
- Early stopping — Yes

...do sequence prediction (for ASR)? — No

Where to go from here?

- Rerun the Emotion experiment with proper train/test splits
- Get the ASR model to learn, and then do inference
- Try some other sequence prediction task, such as machine translation (not as part of this specific project though)

Thank you!