Unwrapping Codec 2 Data Manifolds

October 30, 2023

> Uncrumpling paper balls is what machine learning is all about: finding neat representations for complex, highly folded data manifolds
>
> Francois Chollet [2]

Speech coding can be defined as the art of representing speech using a small number of parameters that can be efficiently sent over a channel.

Codec 2 700C resamples the pitch dependent, time varying $L = F_s/(2F_0), L = 20..80$ harmonic magnitude samples $\{A_m\}, m = 1..L$ to a set of fixed length $K = 20$ samples $\{B_k\}, k = 1..K$. These are then vector quantised $\hat{B_m} = Q(B_m)$. Harmonic phase is recovered using a minimum phase model. The recovered phase spectra and hence energy distribution across the pitch cycle is dependent on the magnitude spectra available at the receiver.

There is some evidence that the distribution of energy over a pitch cycle is important for speech perception [1]. To adequately represent male speech, the formant energy must be spread in time. As Codec 2 recover phase using a minimum phase model - this implies narrow formant bandwidths must be maintained [1]. Narrow formant bandwidths lead to long decay times for filters "ringing" with formant energy. When low resolution spectral sampling is employed (e.g. smoothed vector quantised spectra or low bit rate LSP quantisation of LPC models), male speech becomes buzzy, muffled and less intelligible. Speech with a short pitch period (e.g. female) has less time to decay between pitch cycles so is less sensitive to formant bandwidths, the short pitch period naturally leads to a more uniform distribution of energy over time.

At the Codec 2 receiver, a post filter $P()$ is applied which improves the perceptual quality, in particular for low pitched (male) speakers. The post filter is a form of ad-hoc non-linear filtering, using experimentally derived constants. It raises formants and lowers anti-formants, effectively reducing formant

[1] Other ways of spreading time domain energy are possible, for example generating phases through heuristics, non-minimum phase models, manipulation of phases, or time domain compression [1]

bandwidth and more widely distributing energy in time (each formant "rings" longer).

Neural codecs have shown high quality speech can be synthesised using vectors of sparsely sampled frequency domain samples (MFCCs) [3]. Applying linear transformations (such as IDCT and interpolation) to recover $\{A_m\}$ from MFCCs results in poor quality synthesised speech in sinusoidal codecs. This implies neural codecs are employing a non-linear transformation, made possible by modern deep learning techniques.

From an information theory perspective there is no reason to believe there is more information in speech from low pitch speakers than high pitched speakers. Therefore with an appropriate transformation we should be able to synthesise equivalent quality speech regardless of pitch.

This document describes two experiments to determine if narrow bandwidth formants can be preserved for low pitch speakers using non-linear transformations, resulting in reasonable quality speech when synthesised using Codec 2 (without the use of a post filter).

1 Vector Quantisation

Vector quantisers can take advantage of linear and non-linear dependencies to reduce bit rate [4]. They can efficiently exploit information that is uncorrelated in a linear sense (no linear transformation exists), but is statistically dependant.

The 700C resampling to a fixed length $K << L_{max}$ is a linear transformation aimed at fixing the dimension of the data; reducing the dimension of the data to make it easier to quantise, and reducing VQ storage requirements. However the resampling is a filtering/smoothing/aliasing operation, so some information is lost, for example narrow bandwidth formants cannot be recovered by a linear transformation.

In this experiment we will upsample the variable rate L vectors to a fixed length $K = 80$, then vector quantise. In this way no information is lost from the source vectors. The aim is to utilise the non-linear dependency matching properties of VQ to preserve high quality speech, while still maintaining a similar bit rate to $K = 20$ quantisation. We will ignore storage concerns for this experiment.

2 Machine Learning

In this approach we will attempt to take a $K = 20$ vector and resample it to $K = 80$ using a small neural network. We hope the network will discover any non-linear dependencies, and produce narrow bandwidth formants and (when synthesised using Codec 2) reasonable quality speech for low pitched speakers.

We will include pitch as a feature (perhaps via an embedding network), arguing that the formant bandwidth is a function of pitch (F_0). This is information the VQ in Section 1 does not consider, which implies better expected results

from the ML approach. Using a fixed length target vector simplifies any issues around variable vector length. This is similar to the decoder side of an autoencoder network. Autoencoder designs could therefore be used as candidate architectures.

We argue that essentially the same information (in the form of MFCCs) is used for high quality speech synthesis with neural vocoders, therefore these networks must be performing a similar non-linear mapping of coarsely sampled, smoothed spectral information to speech spectra that includes narrow bandwidth formants. A counter argument is a simple neural network may not be capable of representing the non-linear function that maps between the two vectors. However we are not trying to synthesise speech here - just the speech spectral envelope. Other differences: the network proposed here outputs linear values using regression, more sophisticated neural vocoders employ sophisticated conditional probability models and output a PDF; this network considers just one frame, rather than utilising an autoregressive design that considers many past samples.

3 Evaluation

Evaluation methods considered:

1. Informal listening tests and ranking across a small number of samples.
2. Evaluation of the results using objective measures such as Spectral Distortion (SD), however the relationship to subjective quality may be complex. For a given bit rate SD may be larger at $K = 80$ but the synthesised speech may sound better.
3. Another candidate objective measure is Peak Average Power Ratio (PAPR). This tends to be higher when the formant bandwidths are not preserved.
4. Visual inspection of speech spectra and waveforms of speech synthesised from both methods (prior to post filtering) would indicate if narrow formants have been preserved for males.

Success would be indicated by evidence of formant bandwidth being preserved, and higher quality speech from male speakers compared to linear $K = 20$ approaches. Speech should be synthesised using phases recovered from the output magnitude spectra using the minimum phase model - the degradation in quality is less obvious when original harmonic phases are used.

For controls we could use:

1. Speech resampled through the $K = 20$ "bottleneck" (without postfiltering).
2. Speech VQed at $K = 20$ and $K = 80$.
3. Codec 2 3200 - this uses finely quantised LSPs that preserve formant shapes (and a post filter).

3.1 Notes and Further Work

Notes:

1. Do we need an embedding layer for pitch? Research embedding networks.
2. Should we test with some contrived data? Introduce expected non-linearity and see if network can train as a check.
3. Just include voiced frames? This seems like a good idea. For UV frames pitch may not be meaningful, and may cause random noise. Set UV frames to a constant F_0 (like 0) so at least the network "knows".
4. Make sure we use energy normalisation [1] when resampling to rate K.
5. Used mean removed vectors, just consider $200 < f < 3600$ Hz, for consistency with recent Codec 2 VQ work in [1].
6. Add a block diagram of Codec 2 700C processing.
7. Add to text $K = 80$ are linear spaced, $K = 20$ Mel spaced.

Further work:

1. try bigger VQ training databases, concat some vectors (time-freq).
2. If fixed K works, try to come up with a network that handles sparse vectors.
3. A network that uses PAPR as part of the cost function. It would be useful if we can directly move across this latent space, e.g. to add compression and reduce dynamic range. For V or UV, we should never have highly concentrated energy in time (clicks).
4. Are MFCCs good choices for input features to neural vocoders? They are smoothed, linear transformations of the input speech. Other latent spaces may be more suitable.
5. A linear transformation (like a DCT or PCA) followed by VQ is a good idea - DCT will reduce the dimension, making storage easier, and VQ will capture any non-linear dependencies.
6. Can we avoid gradient shift with multiple stage VQs? Does mbest help?

References

[1] FreeDV-015 Codec 2 Rate K Resampler. `https://github.com/drowe67/misc/ratek_resampler/ratek_resampler.pdf`.

[2] Francois Chollet. Deep learning with python. 2017. *Shelter Island, NY: Manning*, 2018.

[3] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[4] John Makhoul, Salim Roucos, and Herbert Gish. Vector quantization in speech coding. *Proceedings of the IEEE*, 73(11):1551–1588, 1985.