Deep Redundancy for the Opus Codec
`draft-ietf-mlcodec-opus-dred-04`

Presenter: Jean-Marc Valin
jeanmarcv@google.com

IETF 123

Technical Changes Since IETF 122

- Training procedure improvements
  - Better schedule for IS rate lambda
  - Soft quantization
- Robustness
  - Added noise/reverb augmentation
  - Tried denoising network → not a good idea
- New model: shrunk by 3x again
  - Available in `exp_dred_newconv6` branch (not in main)

Changes (New Model)

- New, smaller model
  - Encoder: 200k weights, 245 kB (was 980k / 1060 kB)
  - Decoder: 290k weights, 365 kB (was 660k / 740 kB)
- Available in `exp_dred_newconv6` branch (not in main)

Testvectors Proposal

- Set of DRED payloads in simple binary format (similar to opus_demo format)
  - https://media.xiph.org/opus/ietf/dred_testvectors.tar.gz
  - Provides corresponding decoded float features
- Tools
  - `fargan_demo -dred-decoding` decodes DRED payloads
  - `dred_compare -features` compares with test decoded features
    - Considering total MSE, max feature error, pitch error
  - tests/dred_vectors.sh script checks the error
- Thresholds slightly more strict than quantization noise at Q0

Vocoder Testvectors

- Testvectors also include vocoder testvectors
  - Feature file along with the original audio
- `Tools`
  - `fargan_demo -fargan-synthesis` runs synthesis
  - `dred_compare -audio` compares with the reference
    - Considering average, “peak”, pitch error
  - Included in tests/dred_vectors.sh script
- Thresholds designed to pass for most correct vocoders

Integration Testvectors

- Testvectors also include vocoder testvectors
  - Feature file along with the original audio
- `Tools`
  - `opus_demo -d` runs Opus+DRED decoding
  - `dred_compare -audio` compares with the reference
  - Also included in tests/dred_vectors.sh script
- Should these be OPTIONAL/RECOMMENDED

Open Questions

- Where to publish weights?
- When/how to produce final weights?
- What else do we need to finish?
- Feedback welcome