RNNoise, Neural Speech Enhancement, and the Browser

Jean-Marc Valin
jmvalin@jmvalin.ca

W3C Workshop on Web and Machine Learning
September 2020

(the audio for this talk is processed with RNNoise)
Speech Enhancement

• The signal processing (DSP) way
  – Spectral estimators, hand-tuned parameters
  – Works on stationary noise at mid to high SNR

• The new deep neural network (DNN) way
  – Data driven, often large models (tens of MBs)
  – Handles non-stationary noise, low SNR

• RNNoise: trying to get DNN quality with DSP complexity
RNNoise: A Hybrid Solution

- Start from conventional DSP approach
- Replace complicated estimators with an RNN
- Divide spectrum into 22 “critical bands”
  - Independently attenuate each band

- Use “pitch filter” to remove noise between harmonics
Results (Quality)

- Interactive Demo:
  https://people.xiph.org/~jm/demo/rnnoise/
Complexity (48 kHz)

- Requires 215 neurons, 88k weights
- Based on 10-ms frames
- Total complexity: \(\sim 40\) MFLOPS
  - DNN (matrix-vector multiply): 17.5 MFLOPS
  - FFT/IFFT: 7.5 MFLOPS
  - Pitch search (convolution): 10 MFLOPS
- Unoptimized C code
  - 1.3% CPU on x86, 14% CPU on Raspberry Pi 3
  - Real-time with asm.js via Emscripten
Looking Forward (And Bigger)

- **RNNoise**
  - DNN could still grow by 100x to 1000x
  - Need fast matrix-vector product, low overhead

- **Pure-DNN approaches**
  - Some approaches use large convolutional networks
  - Up to 10s of GFLOPS (may require GPU)

- **Vocoder-based re-synthesis**
  - TTS-like systems using denoised acoustic features
  - WaveRNN/LPCNet: 3-10 GFLOPS, sample latency
Resources

- RNNoise source code (BSD): https://github.com/xiph/rnnoise/
- Demo page: https://jmvalin.ca/demo/rnnoise/
- References