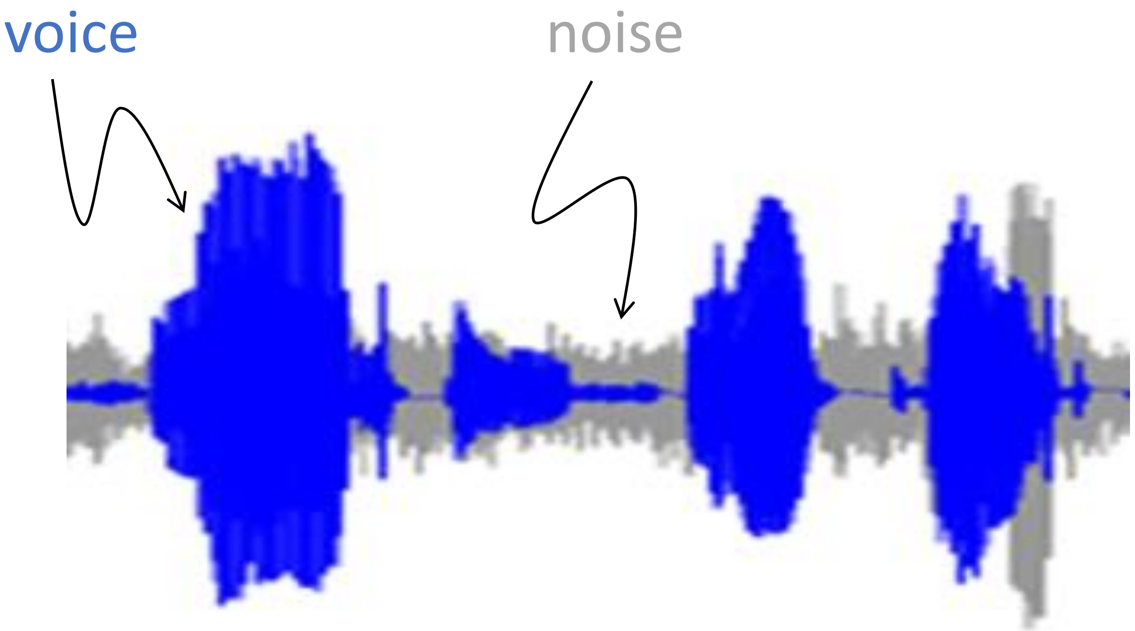
Efficient Voice Activity Detection via Binarized Neural Networks

Jong Hwan Ko   Josh Fromm   Matthai Philipose
Shuayb Zarar   Ivan Tashev

Microsoft   Georgia Tech   U of Washington

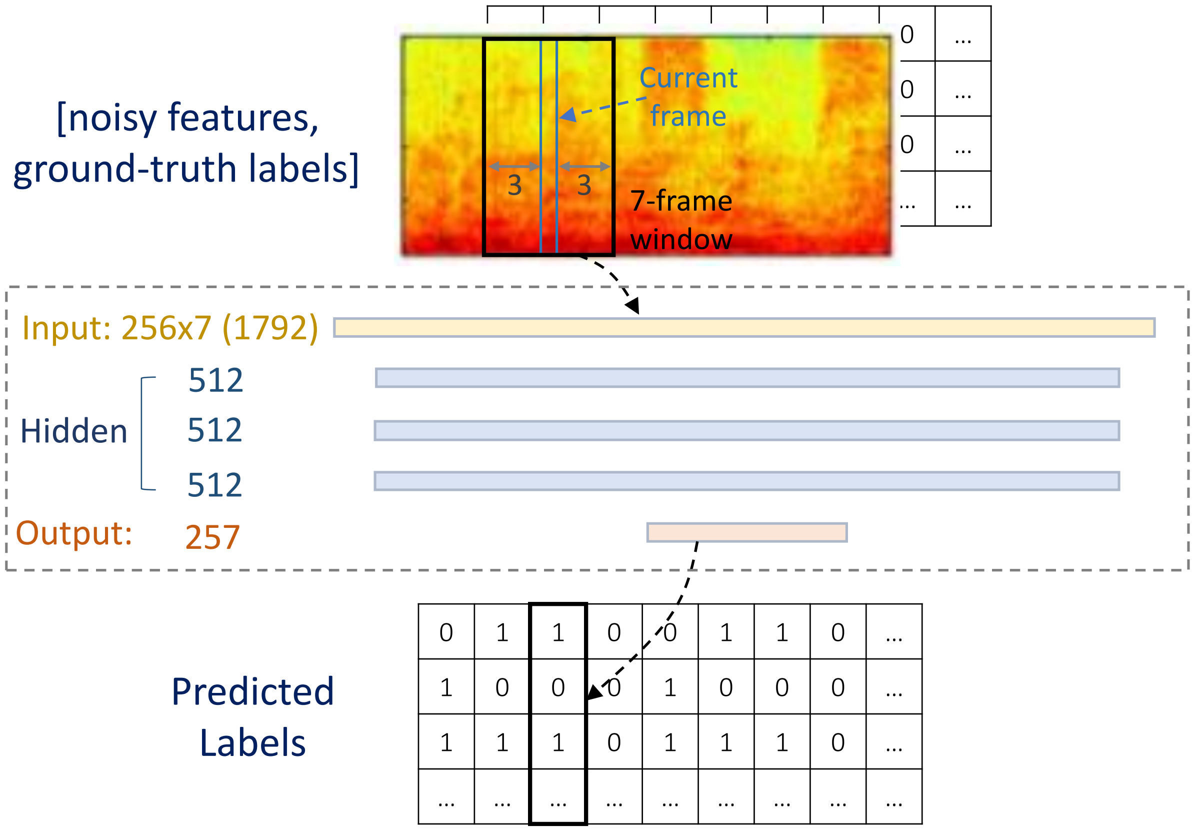
Voice Activity Detection (VAD)



0 0 1 1 1 1 1 0 1 1 1 0 0 1 1 1 0 1 1 1 1 0

- Need to run on a fraction of a CPU
- Traditionally (pre-2016)
  - Based on Gaussian Mixture Models
  - Google WebRTC state of the art:
    - 20.5% error
    - 17 ms latency

VAD with DNNs



[noisy features, ground-truth labels]

0	...
0	...
0	...
...	...

Input: 256x7 (1792)

Hidden 512 512 512

Output: 257

Predicted Labels

0	1	1	0	0	1	1	0	...
1	0	0	0	1	0	0	0	...
1	1	1	0	1	1	1	0	...
...	...	...	...	...	...	...	...	...

- Simple DNN on audio spectrogram †

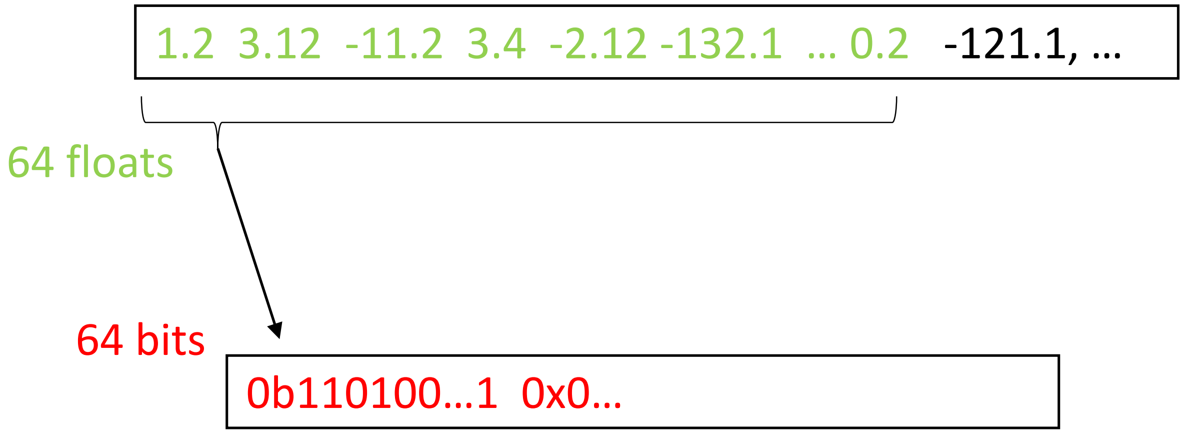
† I. Tashev and S. Mirsamadi, ITA 2016

- Results:
  - ☺ 5.6% error (from 20.5%)
  - ☹ **152ms (from 17ms)**

Idea: Quantize DNN to very low (1-3 bit) bitwidths

Implementing Binarized Arithmetic

- Quantize floats to +/-1
- 1.122 * -3.112 ==> 1 * -1
- Notice:
  - 1 * 1 = 1
  - 1 * -1 = -1
  - -1 * 1 = -1
  - -1*-1 = 1
- Replacing -1 with 0, this is just XNOR
- Retrain model to convergence



$$A[:64]\ .\ W[:64] == \mathrm{popc}(A_{/64}\ \mathrm{XNOR}\ W_{/64})$$

Cost/Benefit of Binarized Arithmetic

```
float x[], y[], w[];
...
for i in 1…N:
    y[j] += x[i] * w[i];
```

2N ops

⬇

```
unsigned long x[], y[], w[];
…
for i in 1…N/64:
    y[j] += 64 - 2*popc(not(x_b[i] xor w_b[i]));
```

3N/64 ops

~40x fewer ops
32x smaller

Problem: Optimized model *slower* when measured! ☹ ☹

Kang et al. ICASSP 2018

Try Again, With Custom GEMM Operation

Per-frame error

(WebRTC=20.46%)

feature quantization bits

weight quantization bits

Model	N32	N8	N4	N2	N1
W32	5.55				
W8		6.25	6.45	7.23	13.87
W4		6.16	6.47	7.32	14.11
W2		6.63	7.06	7.92	13.88
W1		7.91	8.47	8.97	14.95

Sweet spot:

☺ ~5ms latency (30.2x faster)

☺ additional 2.4% accuracy loss

Takeaway: Compilers (a la TVM/Halide) essential for new ops.