Optimization-based computation with spiking neurons

Stephen J. Verzi, Craig M. Vineyard, Eric D. Vugrin, Meghan Galiardi
Conrad D. James and James B. Aimone

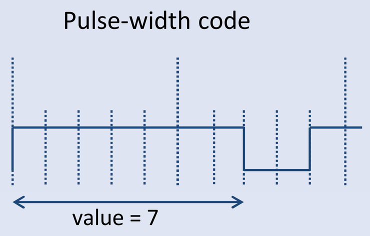
Unary Coding of Numbers

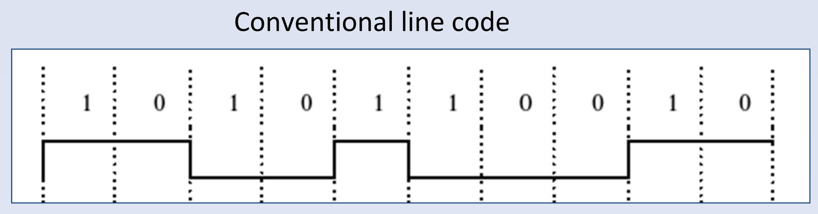
- Reducing communication cost (energy/bit) is crucial
- Unary codes
  - Fixed-length (k): for k=4 →0 is 0000 and 4 is 1111
  - Saves energy
  - Costs either more space or more time

	Binary Serial	Binary Parallel	Unary Serial	Unary Parallel
Space	$O(1)$	$O(\log k)$	$O(1)$	$O(k)$
Time	$O(\log k)$	$O(1)$	$O(k)$	$O(1)$
Energy	$O\left(\frac{\log k}{2}\right)$	$O\left(\frac{\log k}{2}\right)$	$O(1)$	$O(1)$

Temporal Coding

Small difference in time of signal edge conveys information



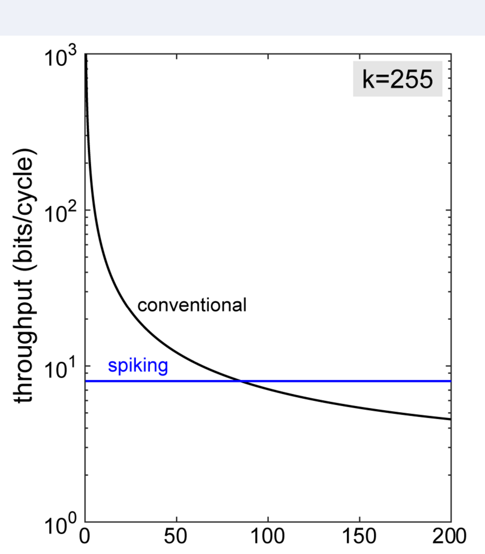


	Throughput (bits/second)	Energy (transitions/bit)
Pulse-width modulation or Phase shift (internal or global)	$O\left(\frac{P\log(k)}{w+\tau_{\text{spike}}}\right)$	$O\left(\frac{1}{\log(k)}\right)$
Winner-take-all	$O\left(\frac{\log(P)}{\tau_{\text{spike}}}\right)$	$O\left(\frac{P}{\log(P)}\right)$
Spike count	$O\left(\frac{\log(P+1)}{\tau_{\text{spike}}}\right)$ ‡	$O\left(\frac{P}{\log(P+1)}\right)$
Weighted spike count – binary	$O\left(\frac{P}{\tau_{\text{spike}}}\right)$ ‡	$O(1)$
Weighted spike count – latency	$O\left(\frac{P\log\left(\frac{\tau_{\text{spike}}}{\tau_{\text{step}}}\right)}{\tau_{\text{spike}}}\right)$ ‡	$O\left(\frac{1}{\log\left(\frac{\tau_{\text{spike}}}{\tau_{\text{step}}}\right)}\right)$
Rank order coding	$O\left(\frac{\log(P!)}{\tau_{\text{spike}}}\right)$ ‡	$O\left(\frac{P}{\log(P!)}\right)$
Synchrony group coding (g groups)	$O\left(\frac{\log(g^P)}{\tau_{\text{spike}}}\right)$ ‡	$O\left(\frac{P}{\log(g^P)}\right)$
Conventional digital	$O\left(\frac{P}{\tau_{\text{spike}}}\right)$	$O(1)$

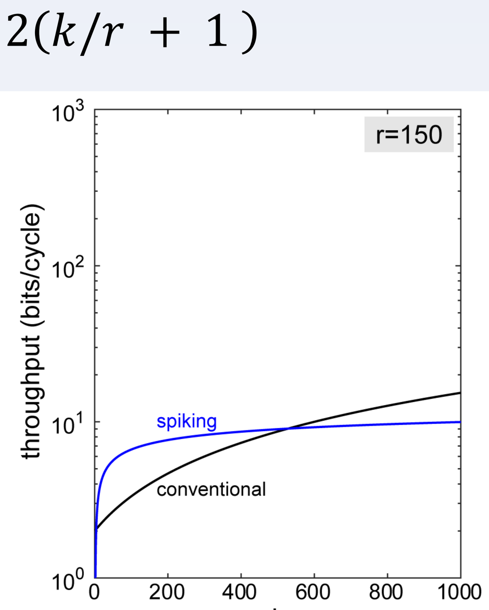
Comparison: Temporal vs. Conventional

Compare bits per pulse-width cycle ($w+\tau_{\text{spike}}$)

Temporal = $\log k$
Conventional = $2(k/r+1)$



Benefit of finer edge detection (with k=255)

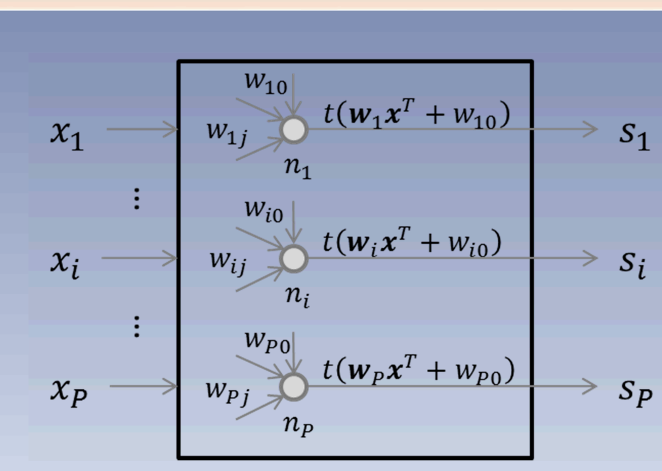


Benefit of smaller number range (with $r=\tau_{\text{spike}}/\tau_{\text{step}}=150$)

References

1. S. J. Thorpe. Spike arrival times: A highly efficient coding scheme for neural networks. In G. Hartmann, R. Eckmiller, and G. Hauske, editors, Parallel Processing in Neural Systems and Computers, pages 91–94. North-Holland Elsevier, 1990.
2. J. J. Hopfield. "Pattern recognition computation using action potential timing for stimulus representation". Nature, 376:33–36. 1995.
3. W. Maass. Networks of spiking neurons: The third generation of neural network models. Neural Networks, 10(9):1659–1671, 1997.
4. S. J. Thorpe, A. Delorme, and R. Van Rullen. Spike-based strategies for rapid processing. Neural Networks, 14(6–7):715–725, 2001.
5. Verzi, SJ., Rothganger, F., Parekh, OD., Quach, T. Miner, NE., James, CD., and Aimone, JB. (2016). "Computing with spikes: The advantage of fine-grained timing". Neural Networks, submitted.

Neural Spiking Module



leaky integrate-and-fire

- Convert from input space to spiking
- Strength of input signal relates to timing of spike

Parallel Random Access Machine (PRAM)

- Shared memory abstract computation architecture
- Metrics
  - Time (T_P) – time required using P processors
    - T_1 – time required by serial processing of algorithm
  - Processors (P) – number of processors needed
  - Work (W) – total effort of algorithm ($W=T_1$)
  - cost – product of time & # processors (cost $=T_P\times P$)
  - Speedup (S_P) – improvement achieved using P processors and parallel processing ($S_P=T_1/T_P$)

SpikingSort PRAM Algorithm

Input : set of integers, $\{x_1,x_2,\ldots,x_N\}$
Output : sparse bit matrix of sorted sequence of spikes, $\boldsymbol{S}$
$\boldsymbol{S}=\boldsymbol{0}$
for $\tau\leftarrow k$ **to** 0
  do for $i\leftarrow 1$ **to** N, in parallel
    if $x_i==\tau$ **then**
      $S(\tau,i)=1$

Comparison of Sorting Algorithms

	T_P	P	cost
Parallel merge sort	$O(\log N)$	$O(N)$	$O(N\log N)$
Reif	$O(\log N)$	$O\left(\frac{N}{\log N}\right)$	$O(N)$
SpikingSort	$O(k)$	$O(N)$	$O(kN)$
SpikingSort, two-layer network	$O(2k)$	$O((k+1)N)$	$O(2k(k+1)N)$
SpikingSort, c-layer network	$O(ck)$	$O\left(\sum_{i=0}^{c-1}k^i N\right)$	$O\left(ck\sum_{i=0}^{c-1}k^i N\right)$
SpikingSort, constant k	$O(1)$	$O(N)$	$O(N)$

SpikeMax PRAM Algorithm

Input : set of integers, $\{x_1,x_2,\ldots,x_N\}$
Output : the maximum, $m=\max_i x_i$
$done$ = **False**
for $\tau\leftarrow k$ **to** 0
  if $done$ == **False then**
    do for $i\leftarrow 1$ **to** N, in parallel
      if $x_i==\tau$ **then**
        $m=x_i$
        $done$ = **True**

Comparison of Algorithms for Finding the Max

	T_P	P	cost
Shiloach and Vishkin	$O(1)$	$O(N^2)$	$O(N^2)$
Valiant	$O(\log N)$	$O\left(\frac{N}{\log N}\right)$	$O(N)$
SpikeMax	$O(k)$	$O(N)$	$O(kN)$
SpikeMax, when $k-d\le\max_i x_i\le k$	$O(d)$	$O(N)$	$O(dN)$
SpikeMax, when $N\gg k$ for constant k	$O(1)$	$O(N)$	$O(N)$

SpikeOpt PRAM Algorithm

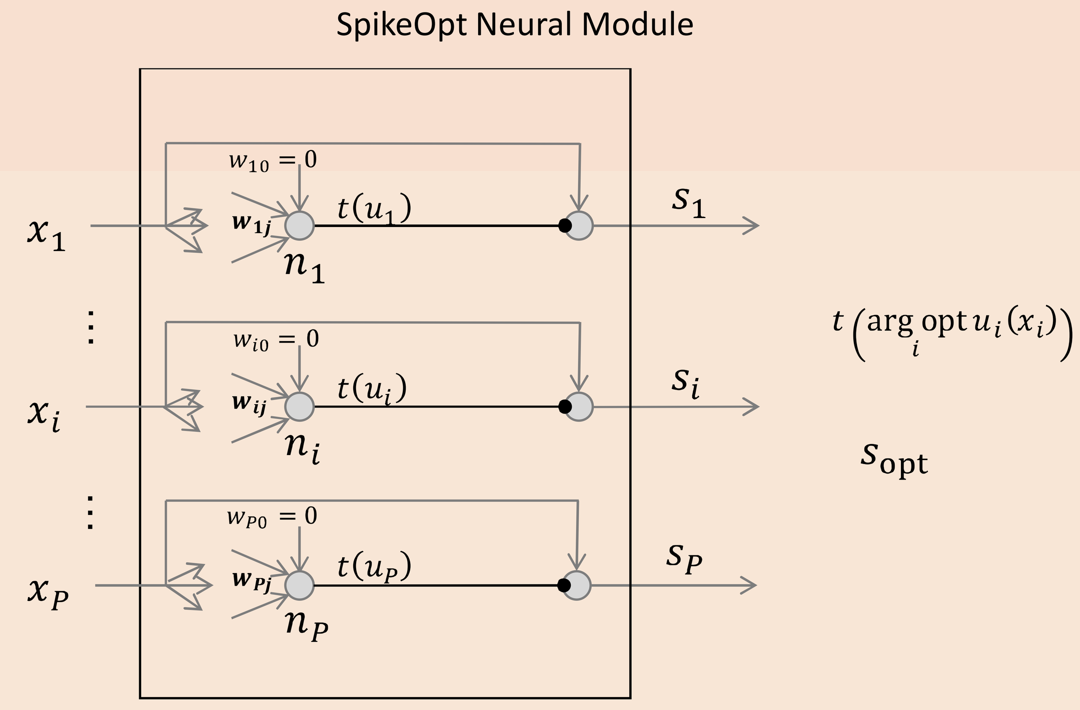
Input : set of integers, $\{x_1,x_2,\ldots,x_N\}$, where N is odd
Output : median integer, $m=\text{median}_i\, x_i$
typedef enum {**INITIAL**, **SPIKING**, **DONE**} is State
$state$ = **SPIKING**
do for $i\leftarrow 1$ **to** N, in parallel
  $u_i=\sum_{j=1}^{N}\text{sign}(x_i-x_j)$
  while $state\neq$ **DONE**
    if $u_i==0$ **then**
      $m=x_i$
      $state$ == **DONE**
    else
      $u_i=u_i-\text{sign}(u_i)$

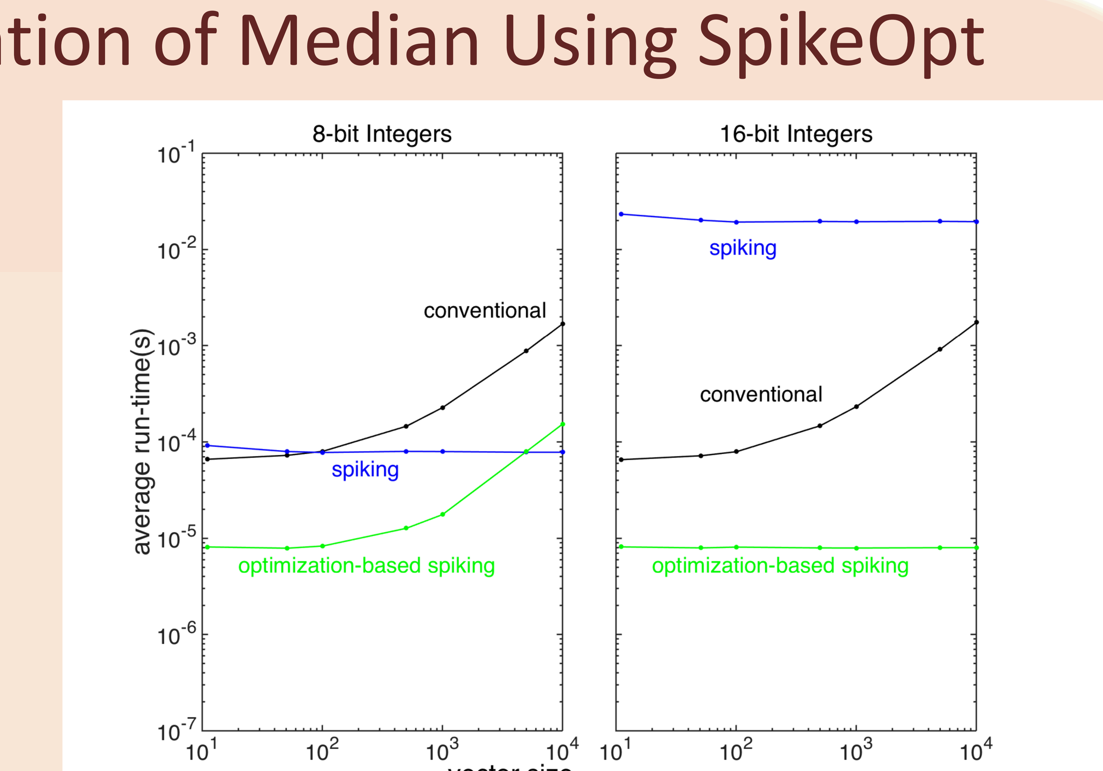
Comparison of Algorithms for Finding the Median

	T_P	P	cost
Akl, for $0<x<1$	$O(N^{1-x})$	$O(N^x)$	$O(N)$
Cole and Yap	$O((\log\log N)^2)$	$O(N)$	$O(N(\log\log N)^2)$
Tishkin	$O(\log\log N)$	$O(N)$	$O(N\log\log N)$
Beliakov	$O(1)$	$O(N)$	$O(N)$
SpikingMedian	$O(k)$	$O(N)$	$O(kN)$
SpikeOpt (median), worst case	$O(N/2)$	$O(N)$	$O(N^2)$
SpikeOpt (median), symmetric distribution	$O(1)$	$O(N)$	$O(N)$
SpikeOpt (median), $\lvert X\rvert=d$, constant d	$O(1)$	$O(N)$	$O(N)$

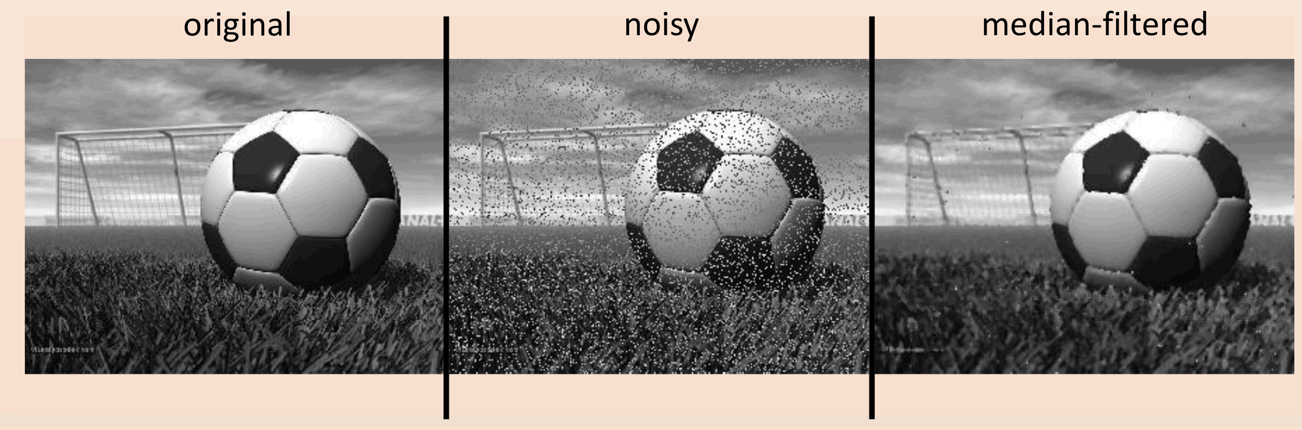
Optimization-based Computation of Median Using SpikeOpt

SpikeOpt Neural Module





Median-filtering with SpikeOpt



Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. XXX