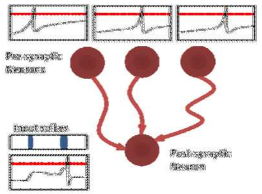
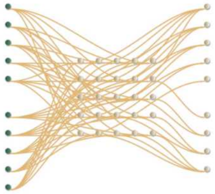
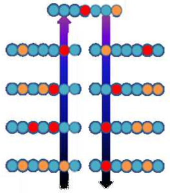
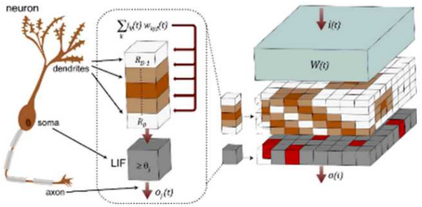
SAND2019-14017C

Neural-inspired Computing at Sandia Labs – Enabling and Performing Advanced Computation









PRESENTED BY

Craig M. Vineyard, PhD

cmviney@sandia.gov

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SANDIA ADDRESSES NATIONAL SECURITY CHALLENGES

- Sandia develops advanced technologies to ensure global peace
- Computing is critical

1987
1st ACM Gordon Bell Prize – 1000x speedup of scientific applications

1989
Solid mechanics code developed expanding role of HPC

1994
Intel XP/S 140 Paragon @ Sandia #1 on Top500

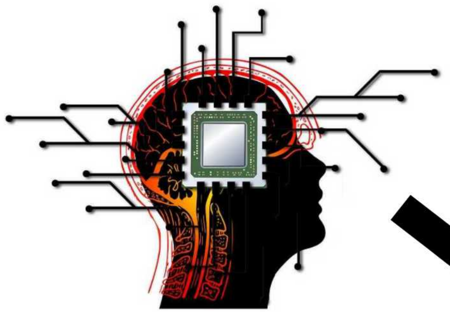
1997
ASCI Red #1 on Top500 First teraflop/s machine

2005
Thunderbird HPC expanding foundational research areas

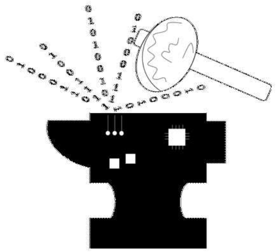
2018
ASTRA supercomputer

?

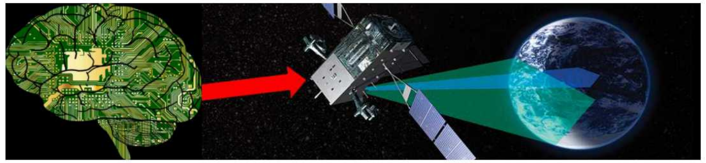
Exploring the path to neural computing impact



NERL exploring algorithm & architecture benefits



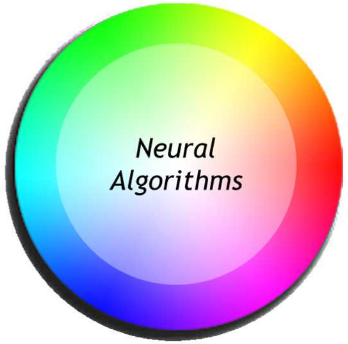
Forge architecture search capability

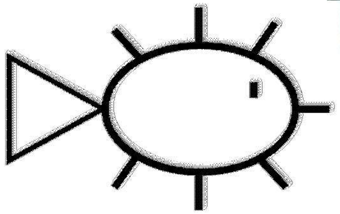


Applications

WHETSTONE

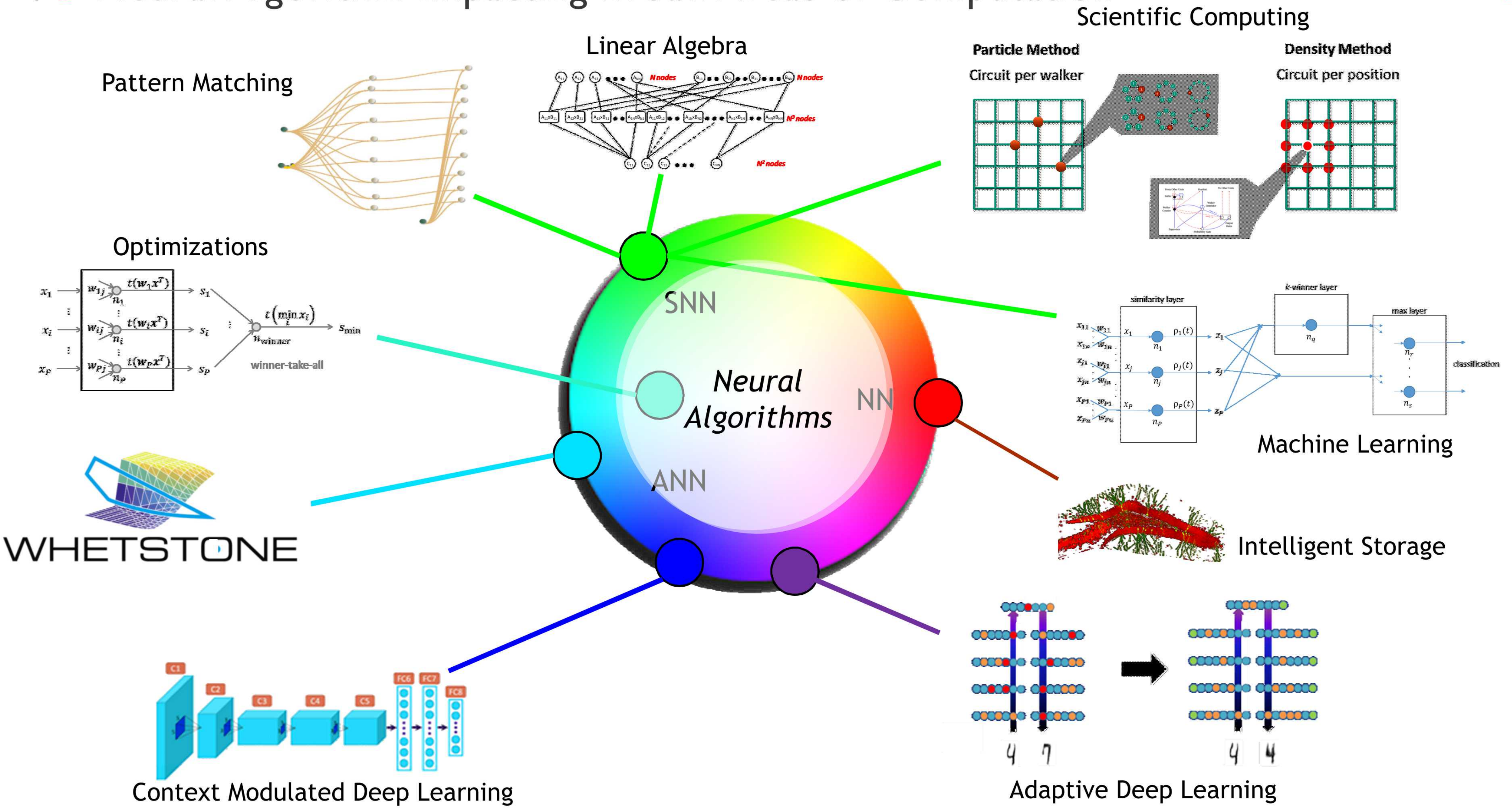
DNN to spiking communication tool





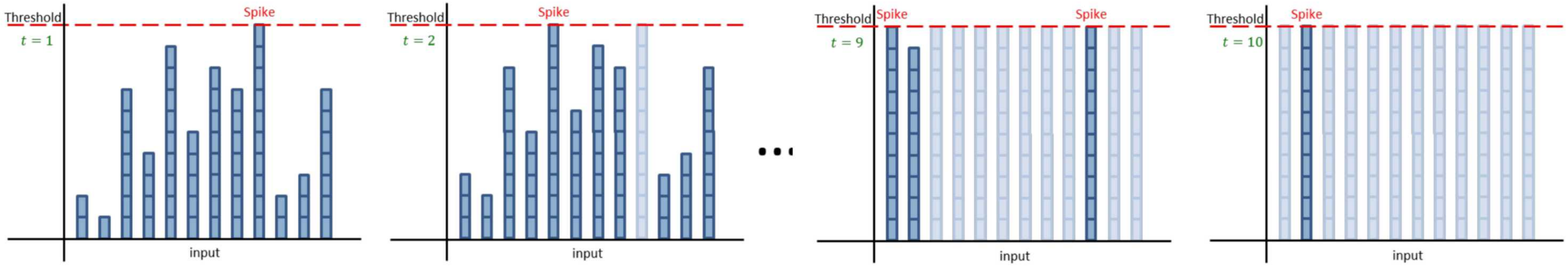
Fugu framework for composing spiking neural circuits

Neural Algorithm Impacting Broad Areas of Computation



Complexity Analysis

Spiking Sort



Algorithm 1 spiking-sort

Input: set of integers, $\{x_1, x_2, \ldots, x_P\}$; k ▷ largest possible integer is $k-1$

Output: sparse bit matrix of spikes, **S**

$w = 0$ ▷ initialize weight matrix to all zeros

for $j \leftarrow 1$ to P, in parallel **do**

  $w_{0j} = 1$ ▷ initialize bias weights

  $\theta_j = k$ ▷ set neuron threshold

  $u_j = x_j$ ▷ directly inject initial value as neuron potential

$x_0 = 1$ ▷ initialize bias input

$\mathbf{S} = 0$ ▷ initialize bit matrix to all zeros

for $j \leftarrow 1$ to P, in parallel **do**

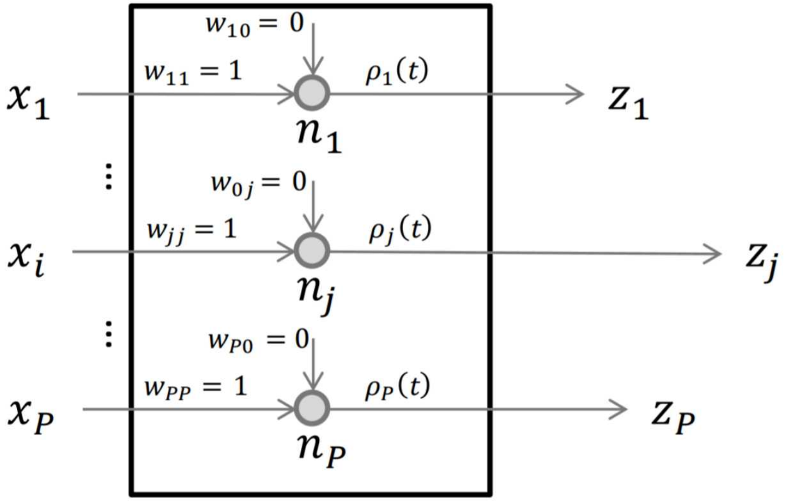
  for $\tau \leftarrow 1$ to k **do**

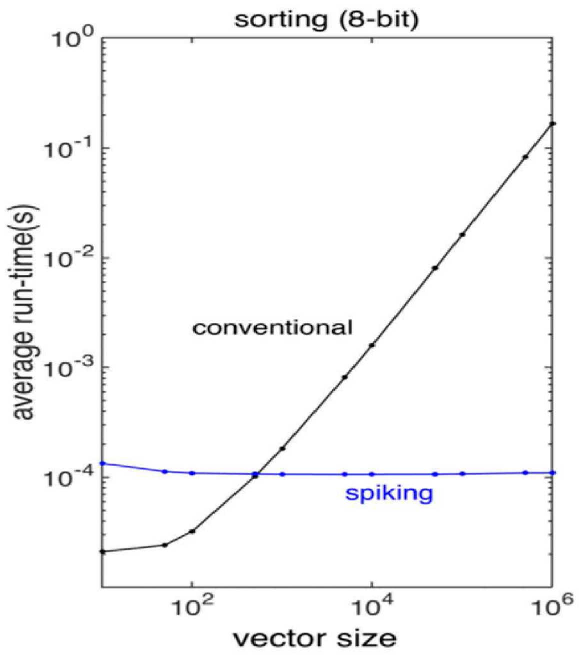
    $u_j = u_j + w_{0j}x_0$ ▷ neuron potential update (discretized LIF)

    if $u_j >= \theta_j$ **then**

      $\mathbf{S}(\tau, j) = 1$

      $u_j = 0$ ▷ reset neuron potential after spike

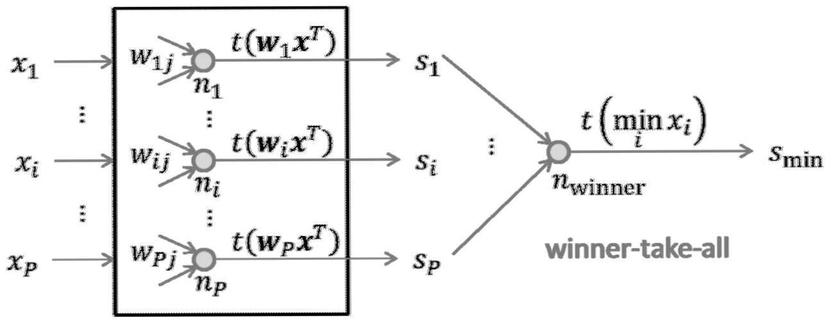




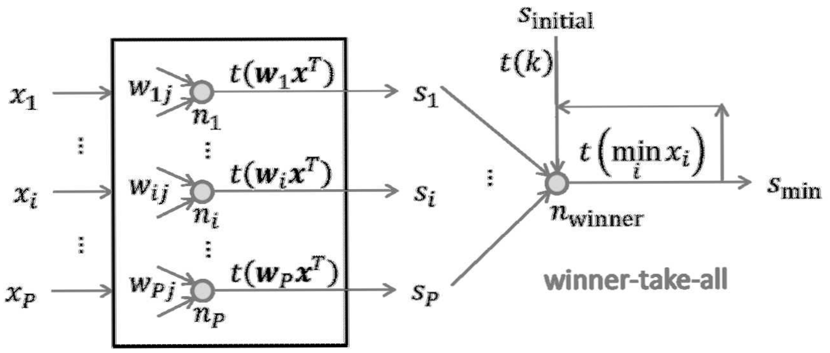
Complexity Analysis

SpikeMin

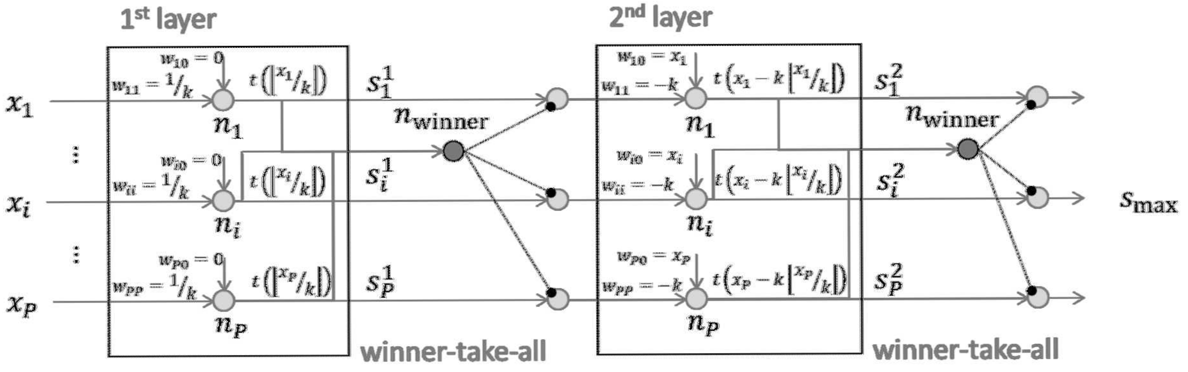
Finding the min where $P \geq N$



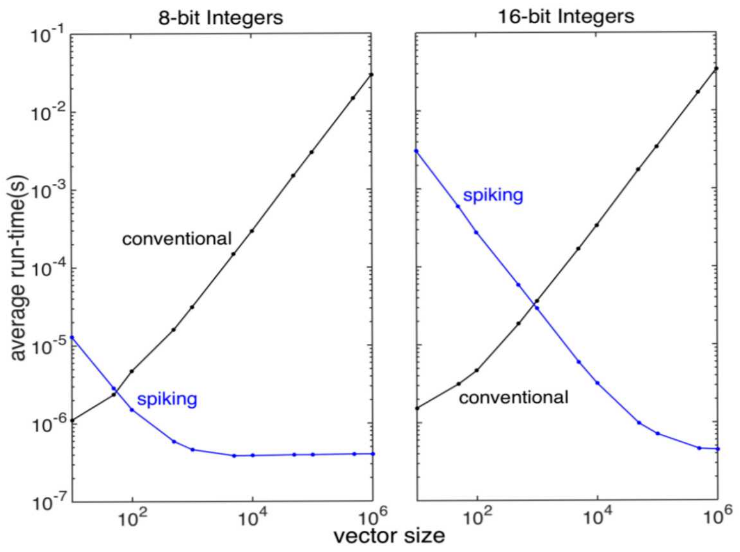
Finding the min where $P < N$



SpikeMax



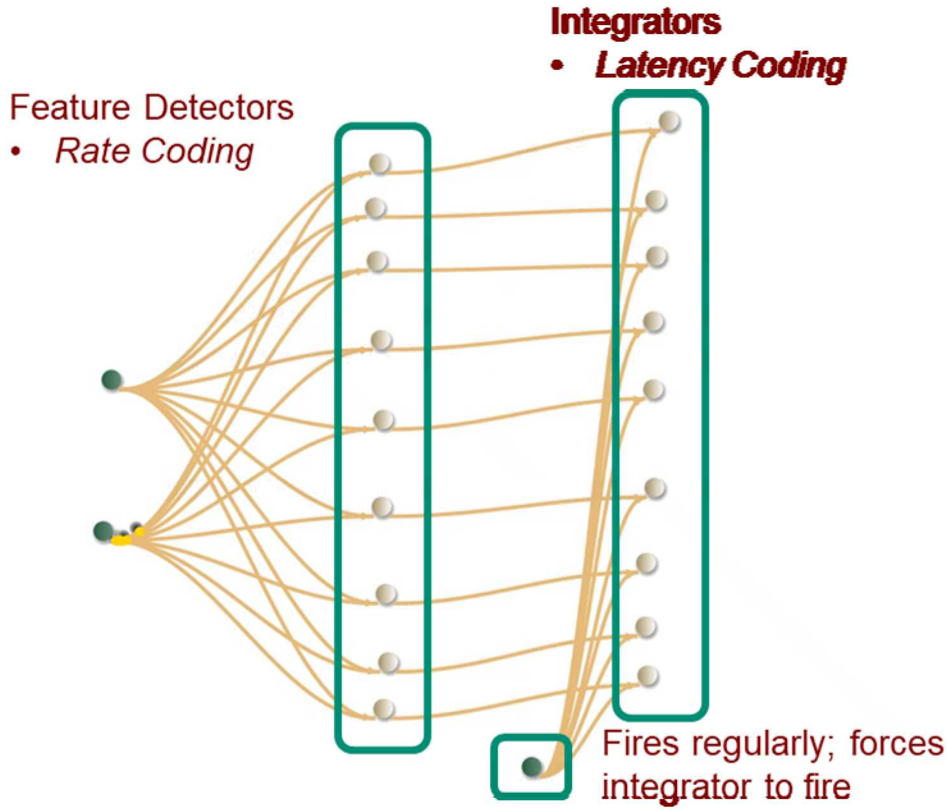
$$t\left(\max_i x_i\right) = ks_i^1 + s_i^2$$

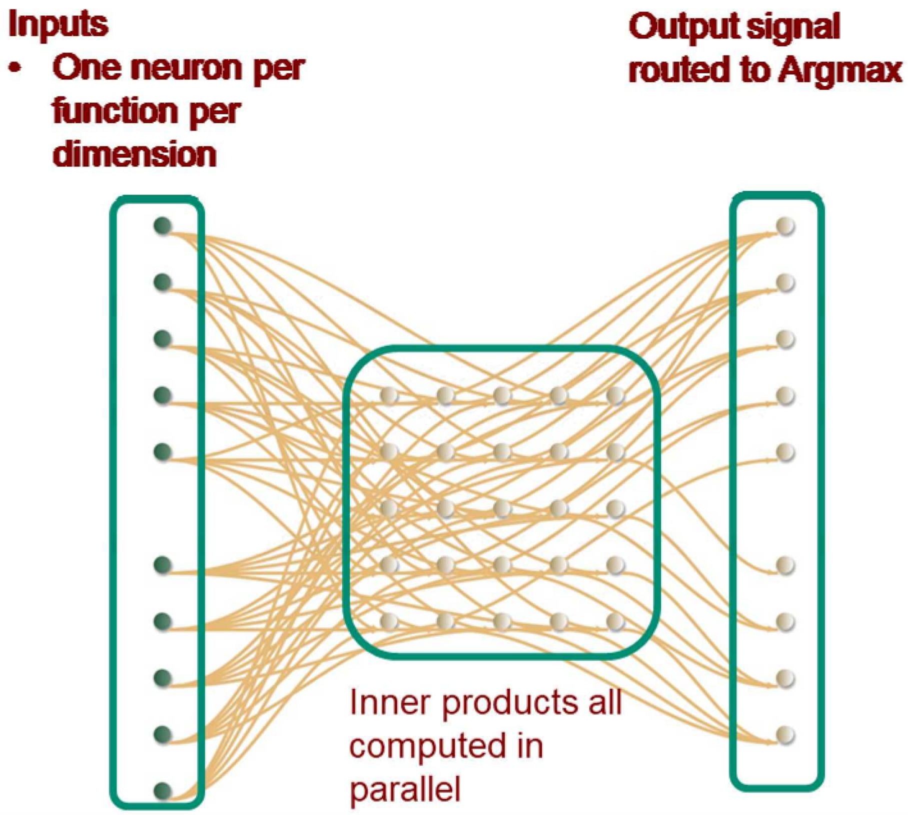


Average runtimes for 10000 simulations of the spike-max neural spiking algorithm

Complexity Analysis

Time Multiplexed Cross Correlation



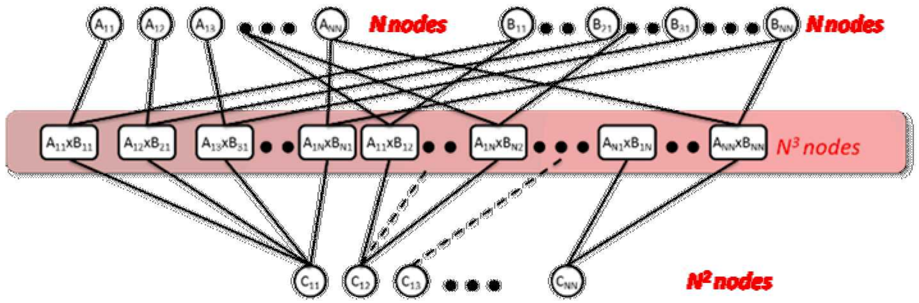


Temporal Coding: $O(n)$ neurons; $O(n)$ runtime
Parallelize inputs & timesteps: $O(n^2)$ neurons; $O(1)$ runtime

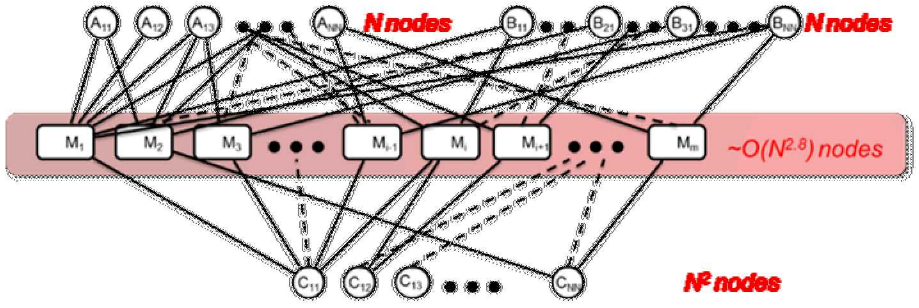
Complexity Analysis

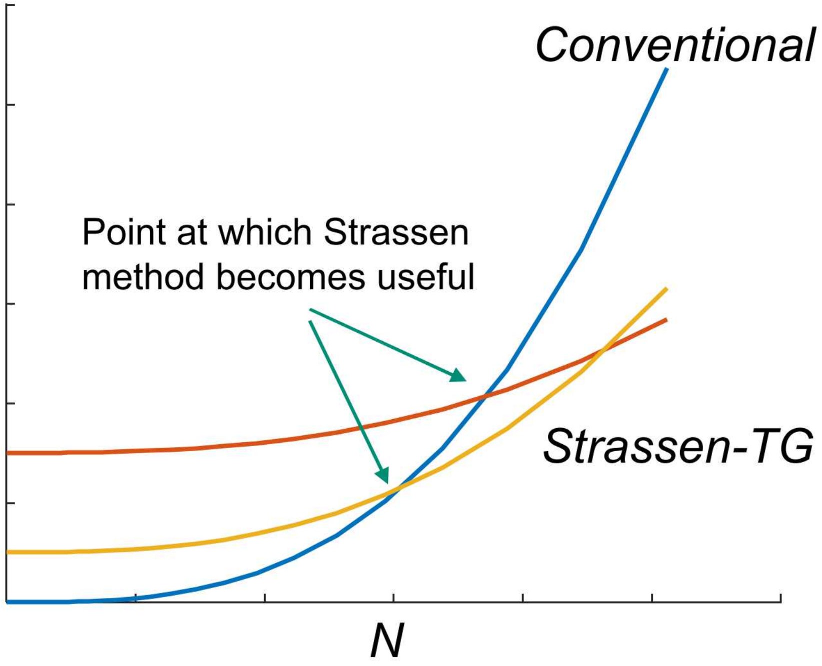
Strassen's Recursive Algorithm for Matrix Multiplication

Standard:
8Ms, 4As → $O(N^3)$



Strassen:
7Ms, 18A/Ss→ $O(N^{2+\varepsilon})$



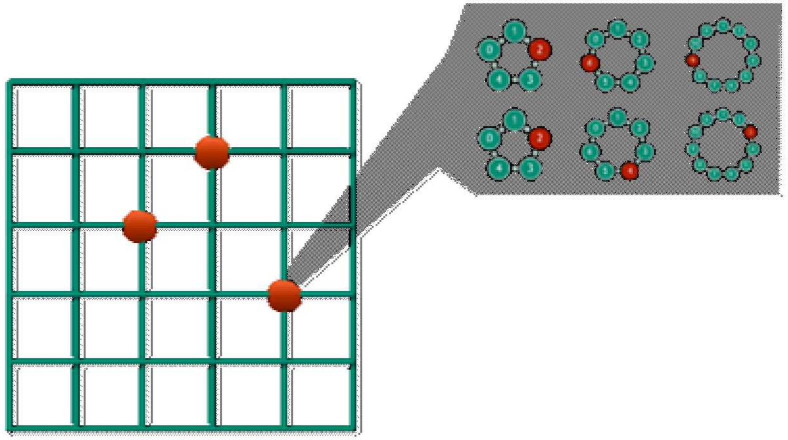


Complexity Analysis

Spiking Random Walk Algorithms

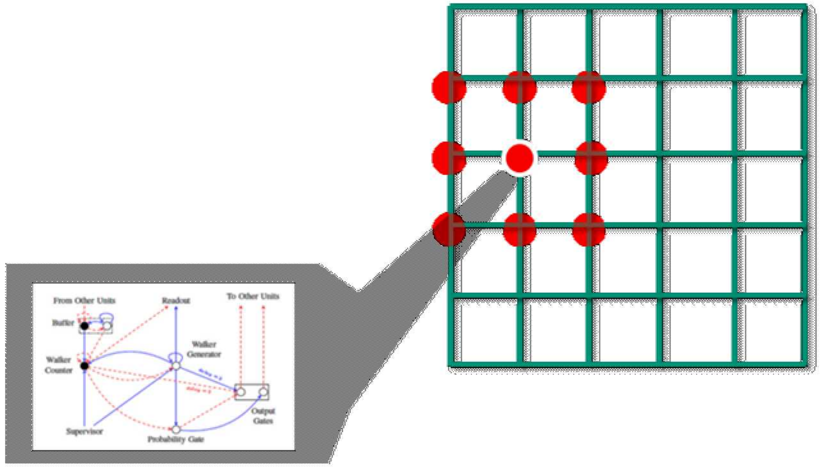
Particle method

- Circuit per walker
- Path dependent behavior is readily available
- Communication is entirely local within particles (embarrassingly parallel)
- With unlimited neurons, can run in constant time
- **Ideal for sparse particles in large spaces**



Density method

- Circuit per position
- Densities are readily available at all times
- Non-local or other complex graphs can easily be implemented
- With limited neurons, can tradeoff statistical approximation (i.e., number of walkers) with longer or shorter simulations
- **Ideal for dense particles in small spaces**



Conclusion

Sandia National Laboratories is pursuing neural-inspired computing as a transformative approach to computation

- Many opportunities at intersection of neuroscience, math, and computing
- Even loosely brain-inspired concepts have potential to be very impactful on computing applications
- Excited to see how neuromorphic at large-scale can enable breakthroughs

Thank you!