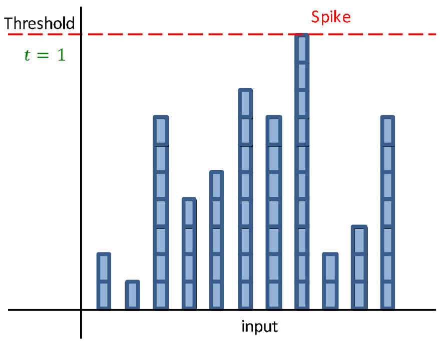
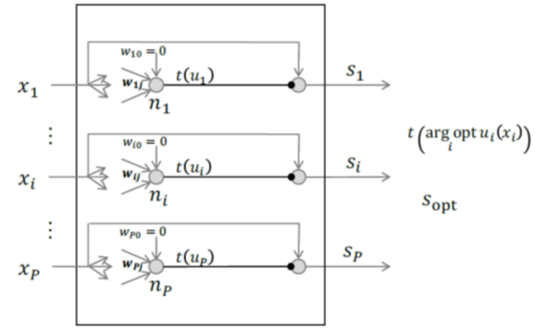
SAND2017-4783C





Optimization-based computation with spiking neurons

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

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.

Outline

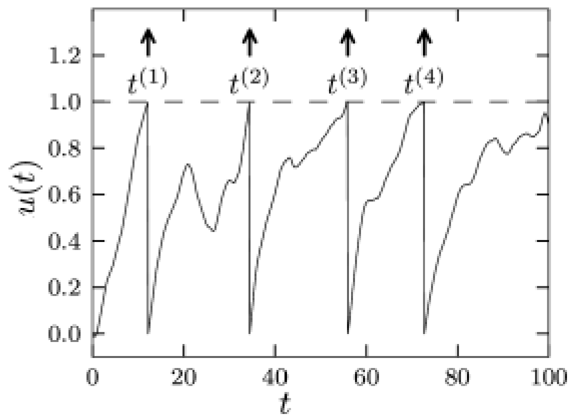
1. Introduction to Spiking
2. SpikingSort and SpikeMin
3. Optimization Using Spikes: SpikeOpt(Median)
4. Complexity results
5. Application: Median Filtering
6. Further work

INTRODUCTION TO SPIKING

Leaky Integrate-and-Fire Neuron Model

$$\tau_m \frac{du}{dt} = -u(t) + RI(t)$$

- $I(t)$ = input to neuron
- $u(t)$ = potential at time t
- τ_m = time constant
- R = resistance



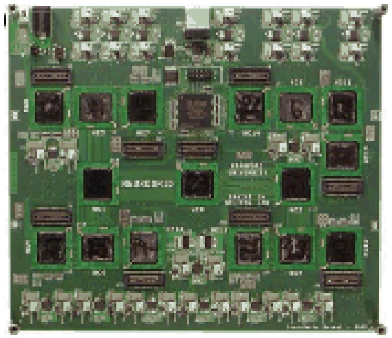
We use

$$u(t+1) = (1-\lambda)\big(u(t)\big)\Big(1 - z\big(u(t)\big)\Big) + I(t)$$

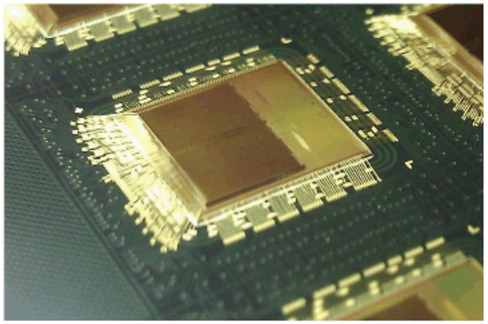
where $z\big(u(t)\big) = \begin{cases} 1 & \text{if } u(t) > \text{threshold} \\ 0 & \text{otherwise} \end{cases}$

Benefits of Neural Computing

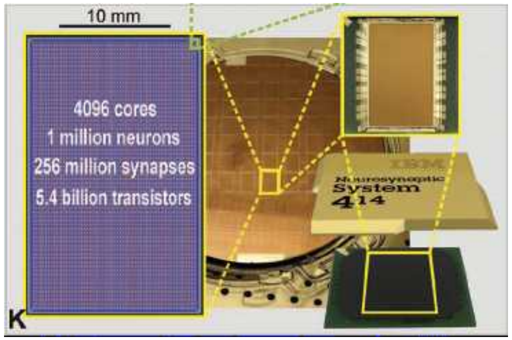
- Low power
- High speed (inherently parallel)
- Addresses gap between neuromorphic architectures (Neurogrid, SpiNNaker, TrueNorth) and algorithms which make effective use of the hardware



B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, Neurogrid, 2014



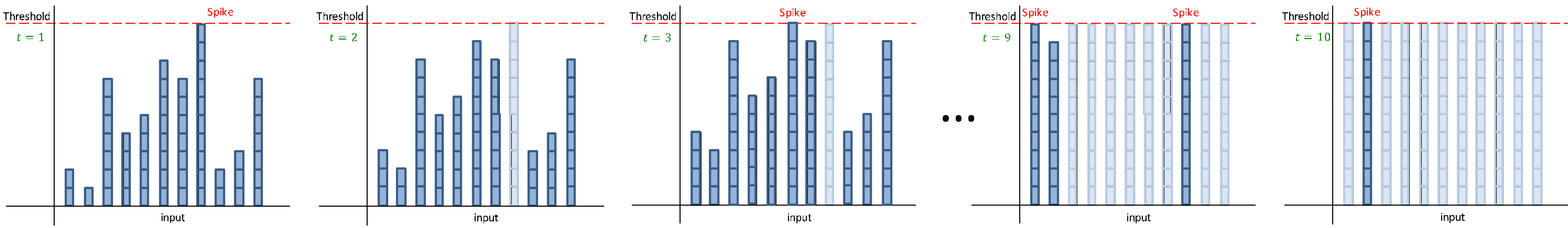
S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, SpiNNaker, 2014

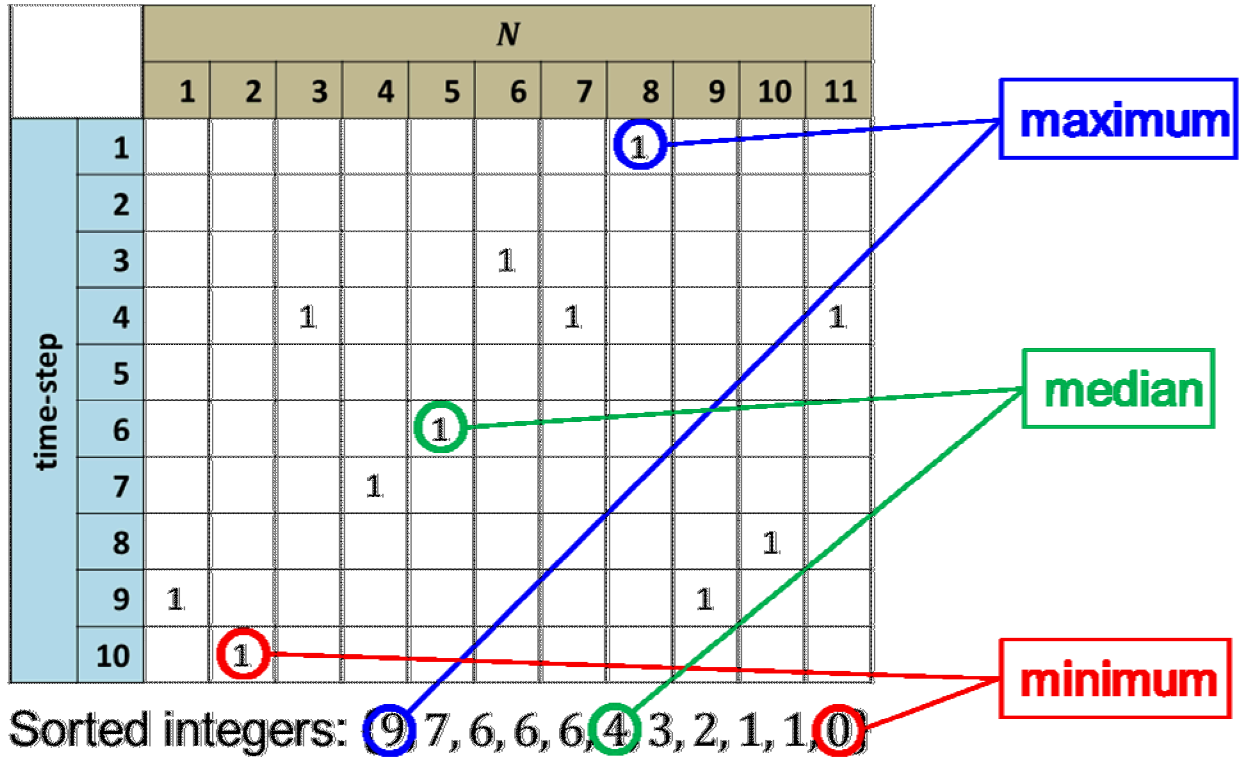


P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo,I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, TrueNorth, 2014

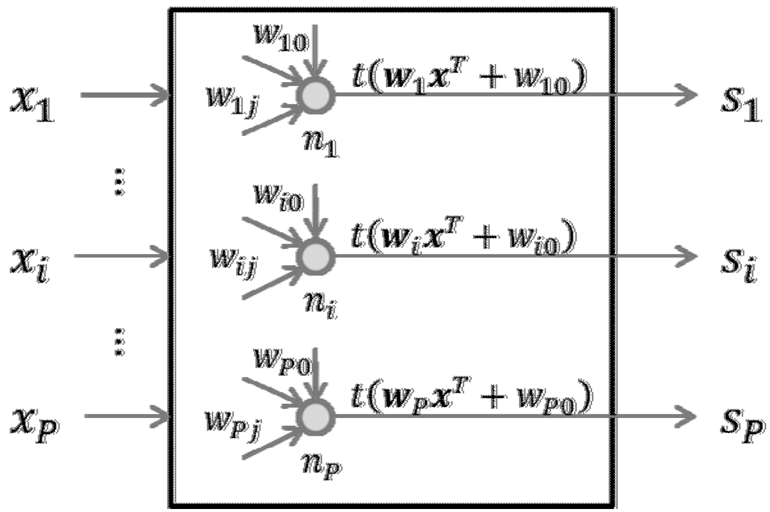
SPIKINGSORT AND SPIKEMIN

SpikingSort



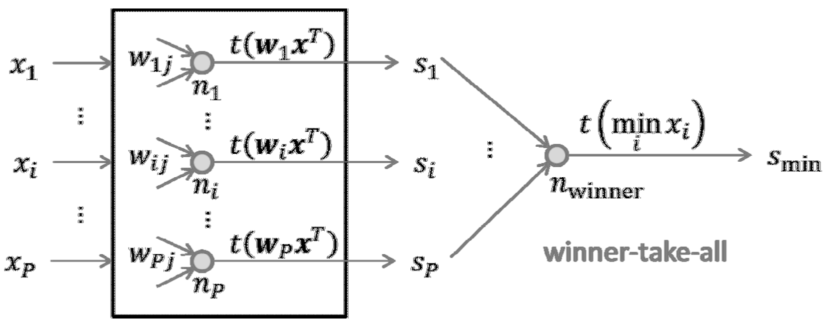


SpikingSort Neural Module

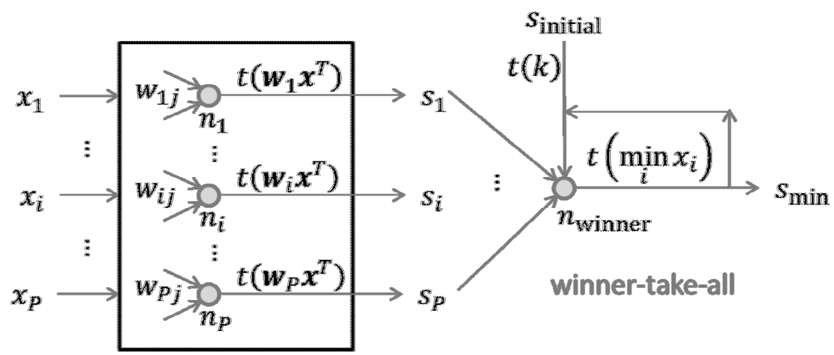


SpikeMin

Finding the min where $P \geq N$



Finding the min where $P < N$



OPTIMIZATION USING SPIKES:

SPIKEOPT(MEDIAN)

Optimization Formula for the Median

- Given a set of floating point numbers $\mathrm{X} = \{x_1, x_2, \dots, x_N\}$
- Compute the Signed Rank function

$$\underset{x \in \{x_i\}}{\tilde{R}(x)} = \sum_{i=1}^{N} \mathrm{sign}(x - x_i)$$

- The median, $\tilde{x}$, is such that $\tilde{R}(\tilde{x})$ is closest to 0

SpikeOpt(Median) Algorithm

Input: Set of integers, $\{x_1, x_2, \dots, x_N\}$ where N is odd

Output: median integer, $m = \text{median}(x_i)$

typedef enum {INITIAL, SPIKING, DONE} is State

State $state \leftarrow SPIKING$ $\triangleright$ initialize state to SPIKING

for $i \leftarrow 1$ to N, in parallel **do**

$u_i = \sum_{j=1}^{N} \text{sign}(x_i - x_j)$

while $state \neq$ DONE **do**

if $u_i == 0$ **then**

$m = x_i$

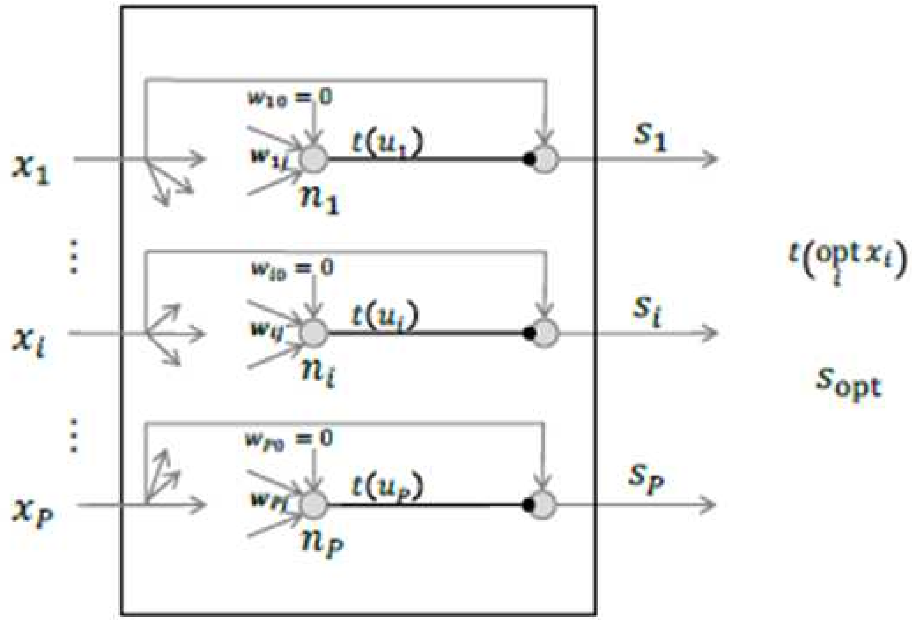
$state =$ DONE

else

$u_i = u_i - \text{sign}(u_i)$

SpikeOpt(Median) Architecture

- Let $w_{ij} = \text{sign}(x_i - x_j)/x_j$



Complexity Analysis

- Signed rank value will be in the range 0 to $\frac{N-1}{2}$
- Worst Case
  - SpikeOpt(Median) will operate for at most $\frac{N+1}{2}$ clock cycles
  - Total work $T_1 = O(N^2)$
  - Work per processor $T_P = O(N)$
  - Speedup $\frac{T_1}{T_P} = O(N)$
  - This is optimal when $P = N$
- Best Case
  - SpikeOpt(Median) will operate for at a minimum 1 clock cycle
  - Work per processor $T_P = O(1)$,

Complexity Analysis

Theorem 1 – The SpikeOpt(median) algorithm achieves optimal runtime with the PRAM framework for a symmetric probability distribution.

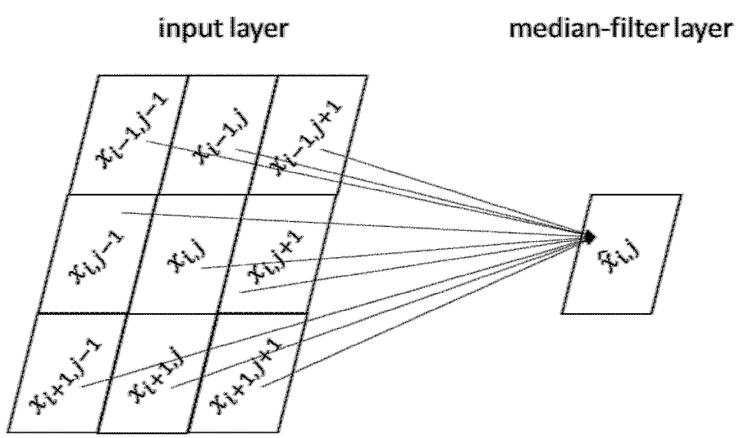
Theorem 2 - The SpikeOpt(median) algorithm achieves optimal runtime with the PRAM framework if each integer x_i is unique.

APPLICATION:

MEDIAN FILTERING

Median-Filtering

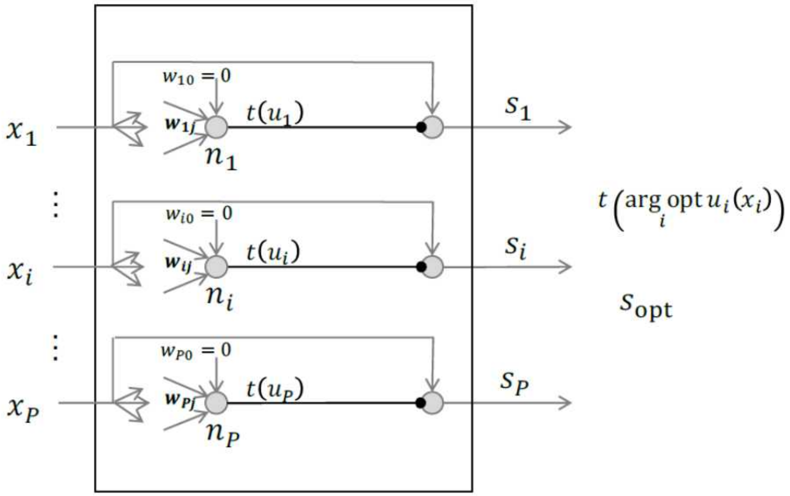
- Median-filtering is an algorithm to perform noise reduction on an image or signal
- Run through image, pixel by pixel, and replace the current value us the value of the median of the neighbors
- Maximum size for each median operation is 9 which means we can we can compute the median filtered image in constant time using SpikeOpt(Median)



Median-Filtering

- Original image

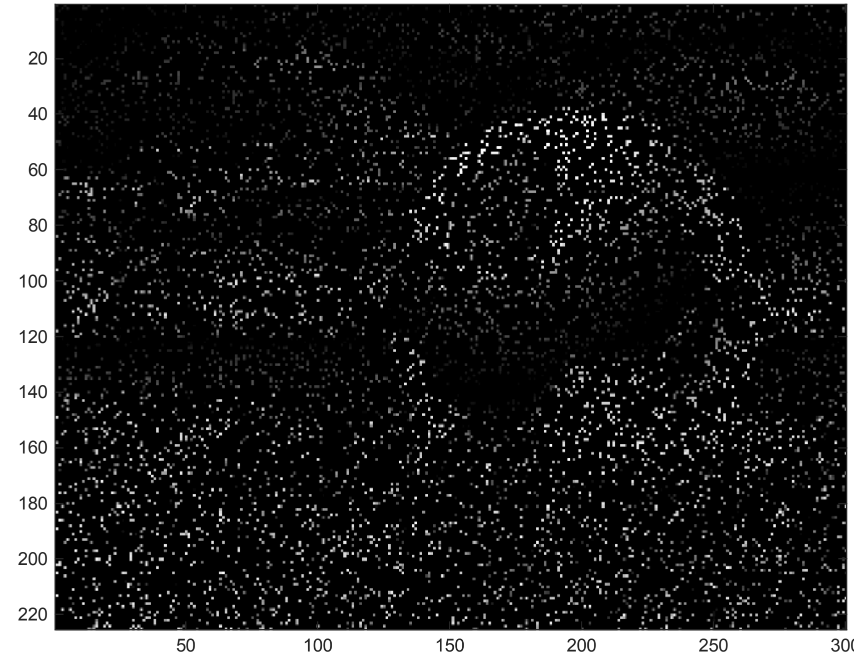
L. Fei-Fei, R. Fergus, and P. Perona, Caltech 101, 2004



SpikeOpt network using decay

Median-filtering

- Noisy image



percent pixels different = 9.85

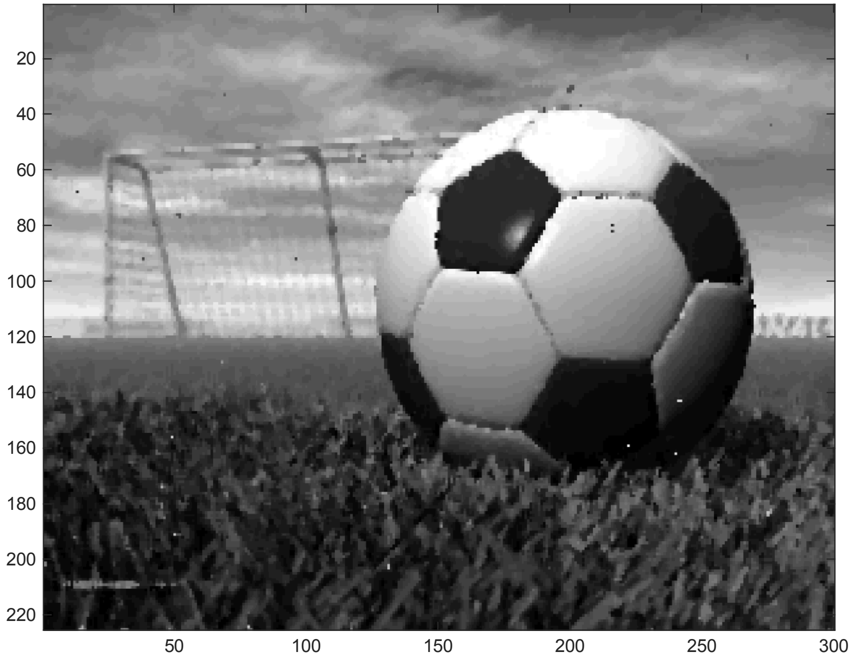
number different pixels = 6651

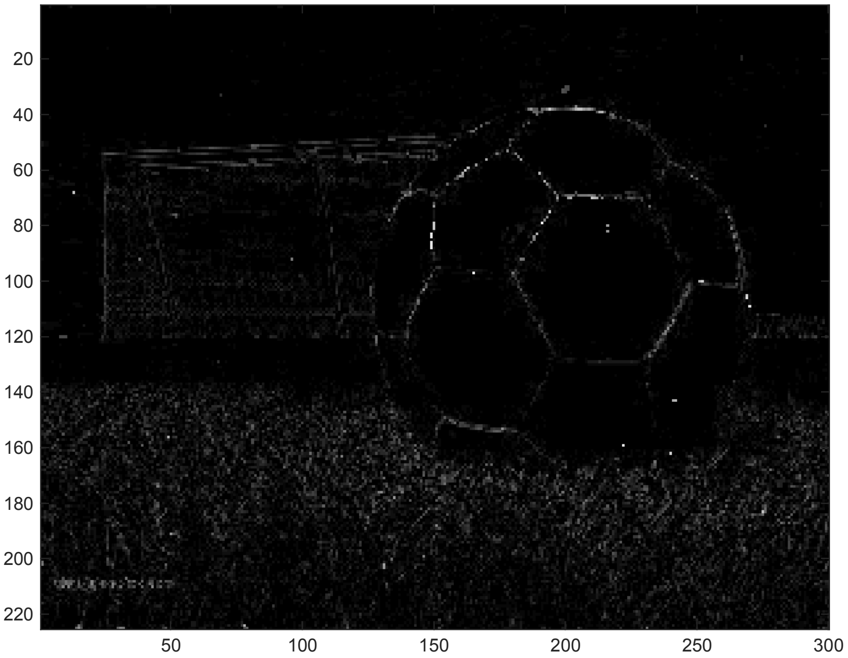
total difference = 780735

average difference = 117.386

Median-filtering

- Median-filtered image (1st iteration)





percent pixels different = 67.7

number different pixels = 45704

total difference = 578625

average difference = 12.6603

FURTHER WORK

Further Work

- Apply SpikingOpt to other types of optimization problems
- Enhance SpikeOpt/SpikeMin to handle real-valued numbers
- Incorporate memory and learning

Adaptation

- Can we have the SpikingOpt architecture adapt to learn the weights instead of hard coding them for a specific application?
- Can we adapt to learn median filtering?
- Can we use SpikingOpt to adapt other networks?
  - Given a network, we want to optimize it to do something
  - Use SpikingOpt to allow the network to adapt to optimal conditions
  - Similar to GANs

References

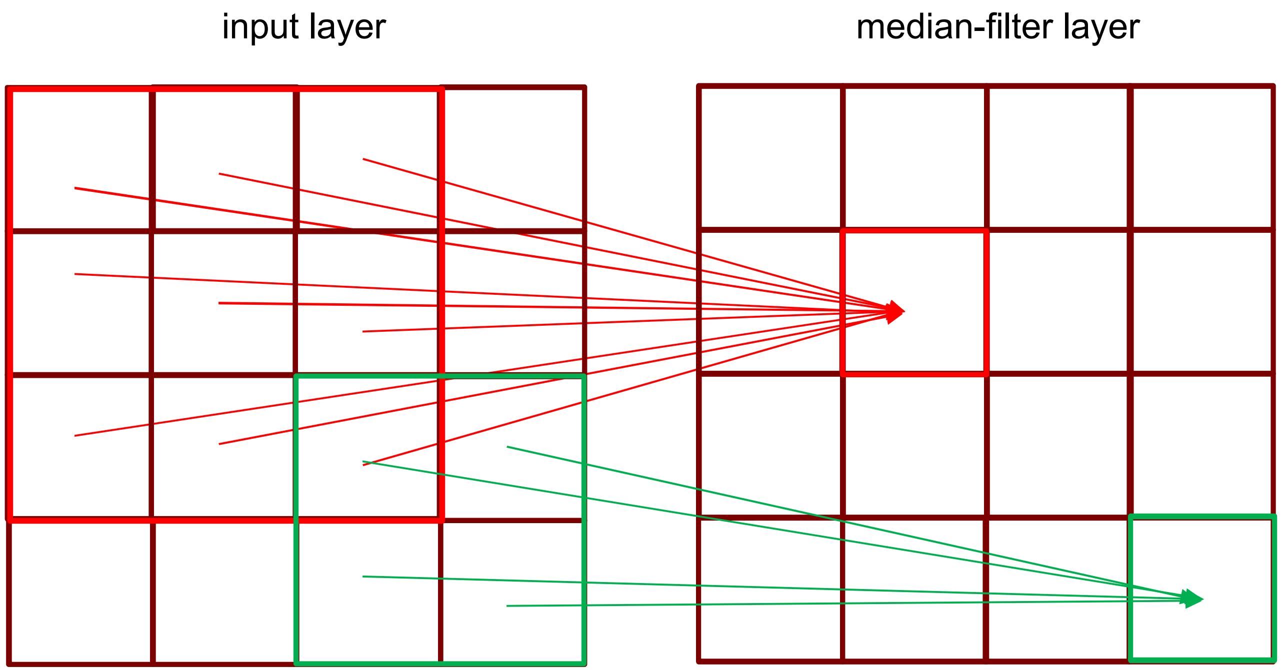
- L. F. Abbott, “Lapicque’s introduction of the integrate-and-fire model neuron (1907),” Brain Research Bulletin, vol. 50, no. 5, pp. 303–304, 1999. [Online]. Available: http://dx.doi.org/10.1016/S0361-9230(99)00161-6
- H. Oja, Multivariate Nonparametric Methods with R, An Approach Based on Spatial Signs and Ranks, ser. Lecture Notes in Statistics. New York City, NY: Springer, 2010, vol. 199. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-0468-3
- S. J. Verzi, F. Rothganger, O. D. Parekh, T.-T. Quach, N. E. Miner, C. D. James, and J. B. Aimone, “Computing with spikes: The advantage of fine-grained timing,” , submitted.

BACKUP SLIDES

Parallel computational complexity comparison of algorithms for finding the median.

	T_P	P	cost
Akl, for $0 < x < 1$	$O(N^{1-x})$	$O(N^2)$	$O(N)$
Cole & 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, worst-case	$O(N/2)$	$O(N)$	$O(N^2/2)$
SpikeOpt, symmetric	$O(1)$	$O(N)$	$O(N)$
SpikeOpt, $\|X\| = d$	$O(1)$	$O(N)$	$O(N)$

Sandia National Laboratories

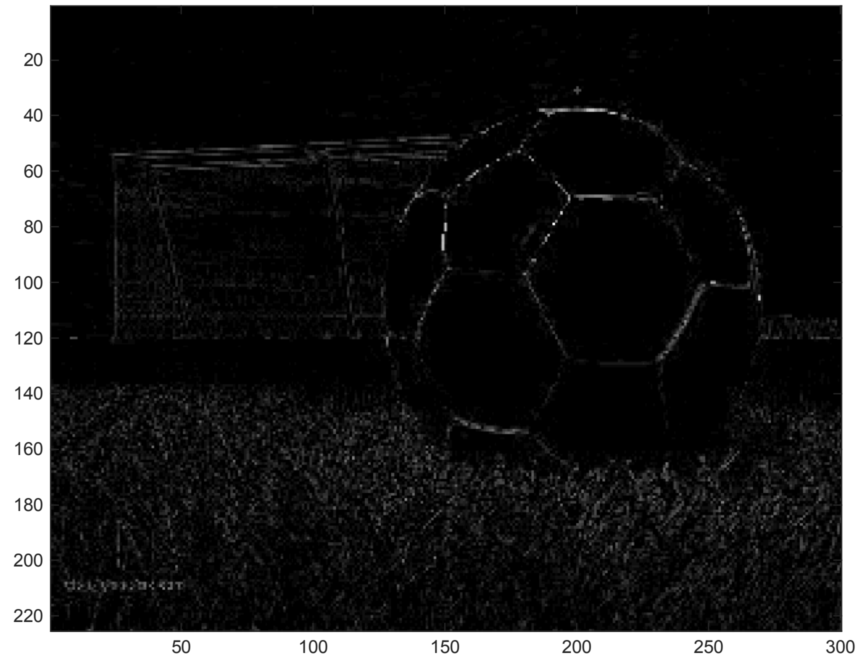


$x_{i,j}$

$\hat{x}_{i,j} = \underset{i-1 \le p \le i+1, j-1 \le q \le j+1}{\text{median}} x_{p,q}$

Median-filtering

- Median-filtered image (2^{nd} iteration)



L. Fei-Fei, R. Fergus, and P. Perona, Caltech 101, 2004

percent pixels different = 69.2

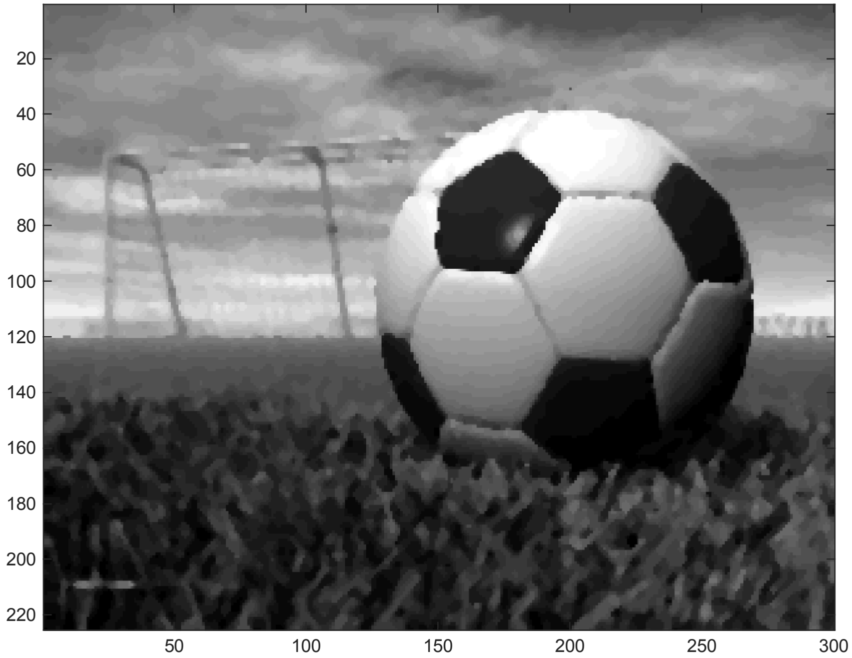
number different pixels = 46682

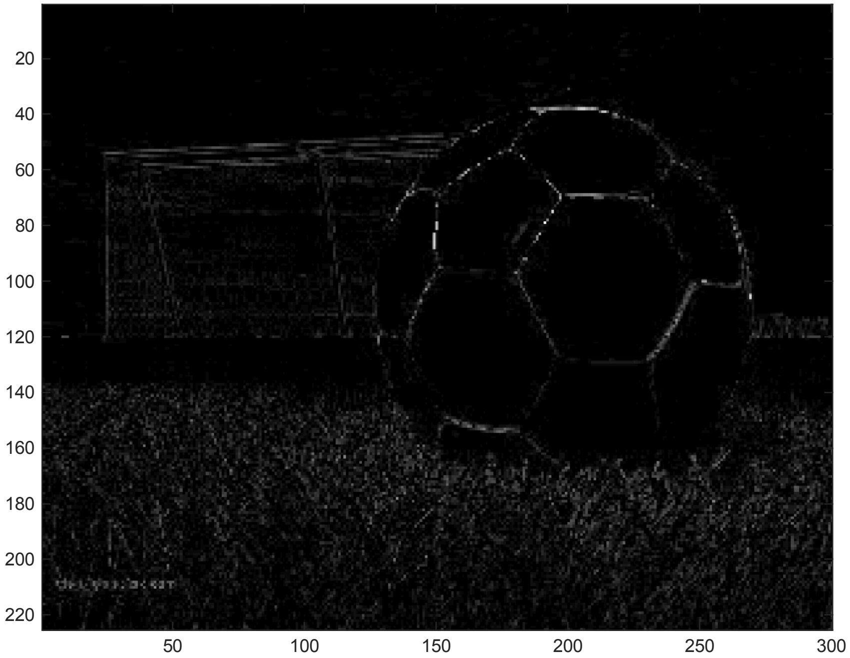
total difference = 578948

average difference = 12.4020

Median-filtering

- Median-filtered image (3rd iteration)





percent pixels different = 71.8

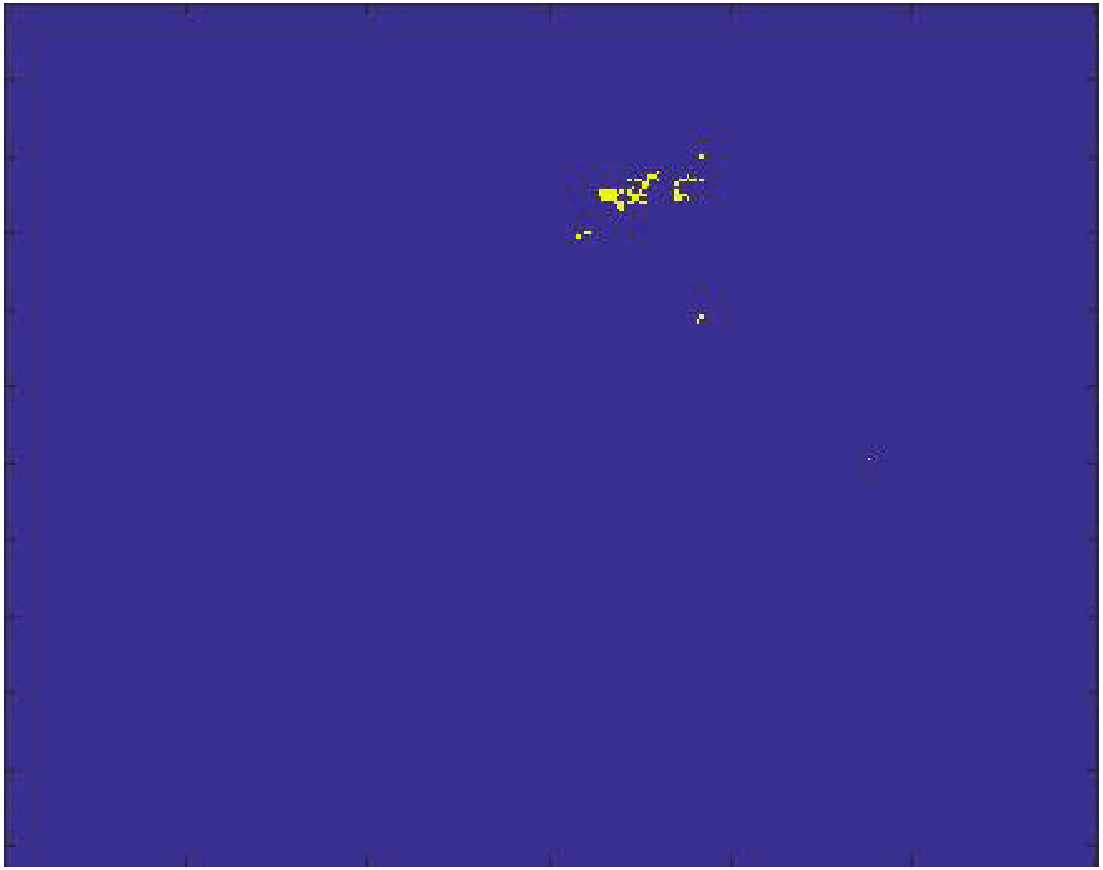
number different pixels = 48489

total difference = 619310

average difference = 12.7721

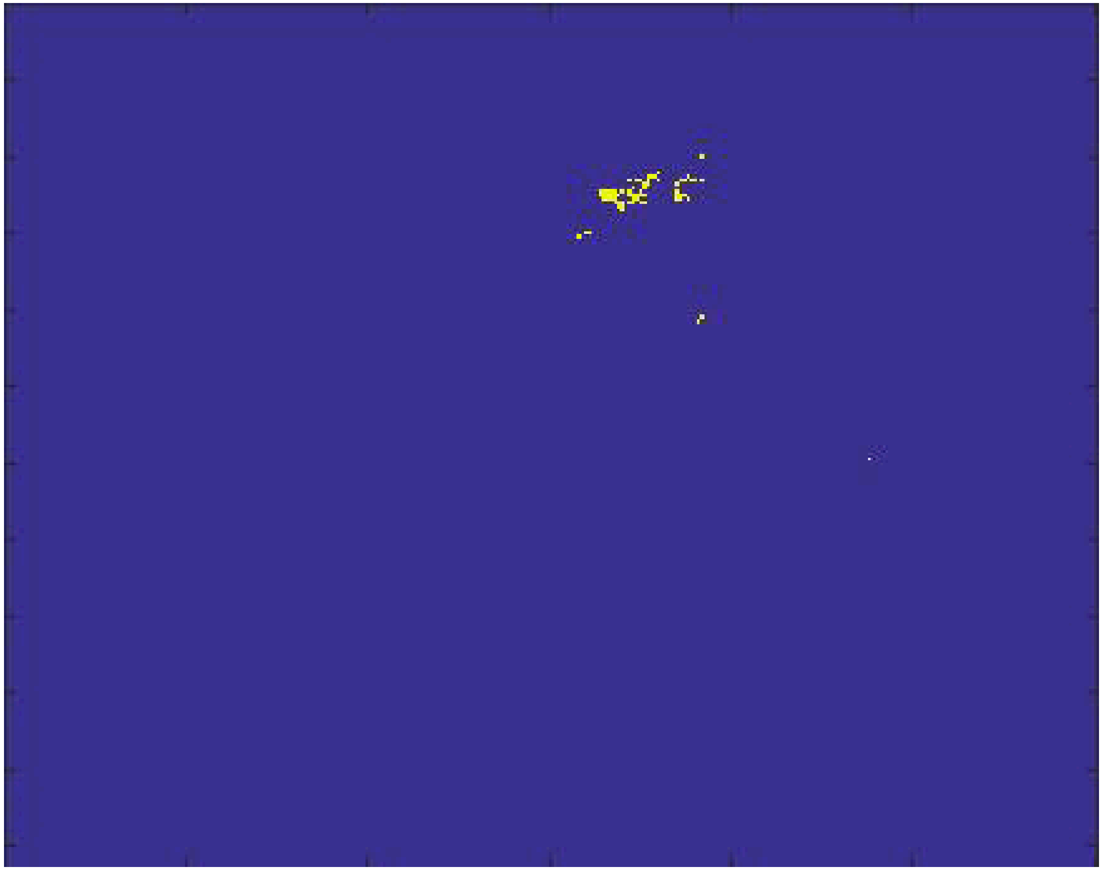
Demonstration of temporal-coding representational capacity

- Spikes as they happen in time



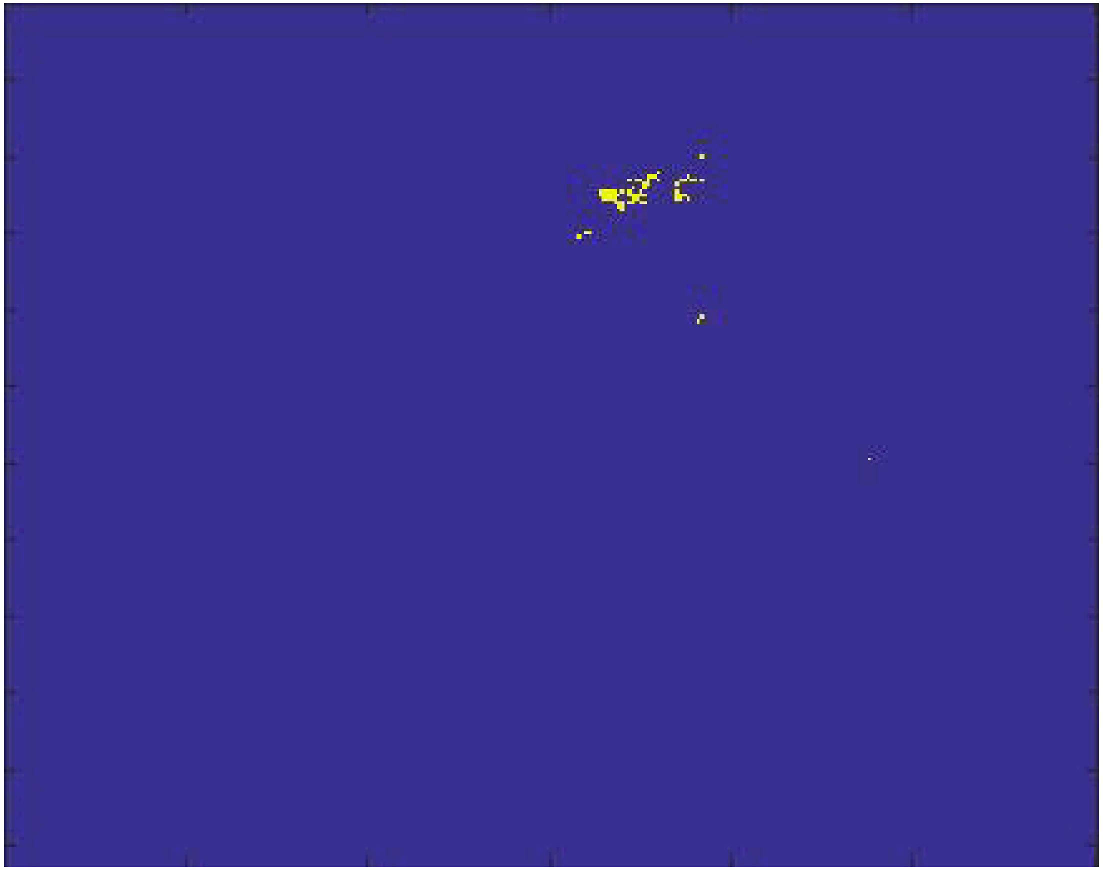
Demonstration of temporal-coding representational capacity

- Aggregation of spikes (from all 0's to all 1's)



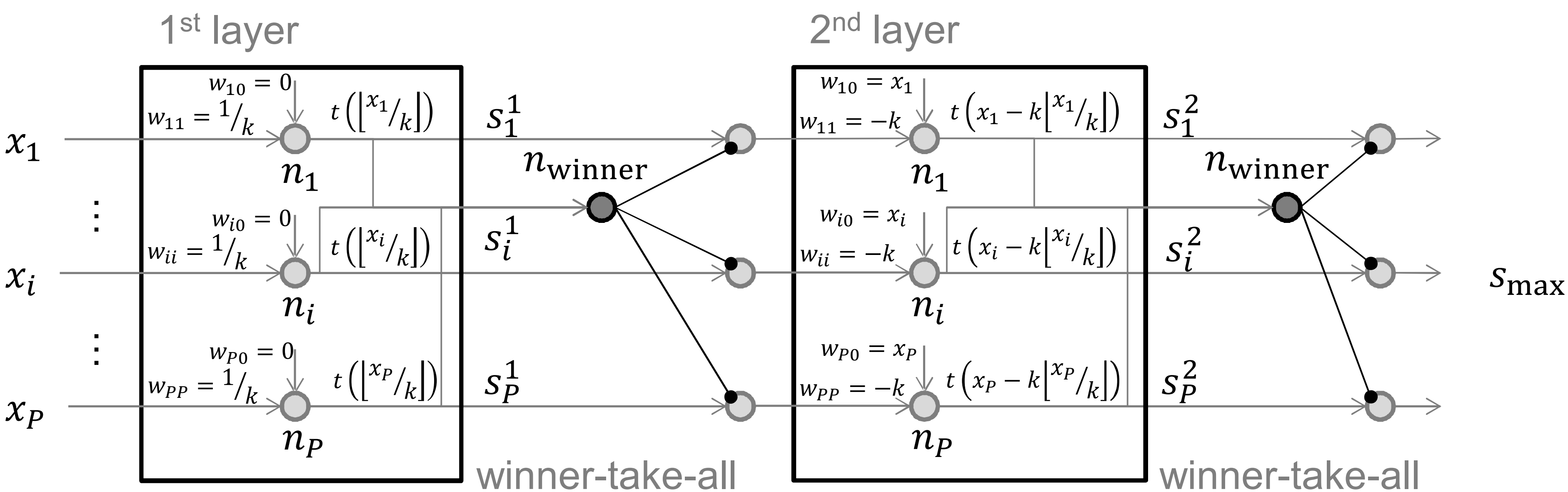
Demonstration of temporal-coding representational capacity

- Aggregation of spikes weighted by their temporal code value



		neuron												Temporal code
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	ρ
time	**1**									1				7
	2					1					1			6
	3							1						5
	4			1					1				1	4
	5													3
	6						1							2
	7	1			1									1
	8		1									1		0

Finding the max



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