SAND2014-20016PE

Review of “cross neural” research effort

Erik P. DeBenedictis
Bradley J. Aimone

/*No public release at the moment
SAND SAND2014-XXXXX C*/

Sandia National Laboratories is a multi-program 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.

Outline

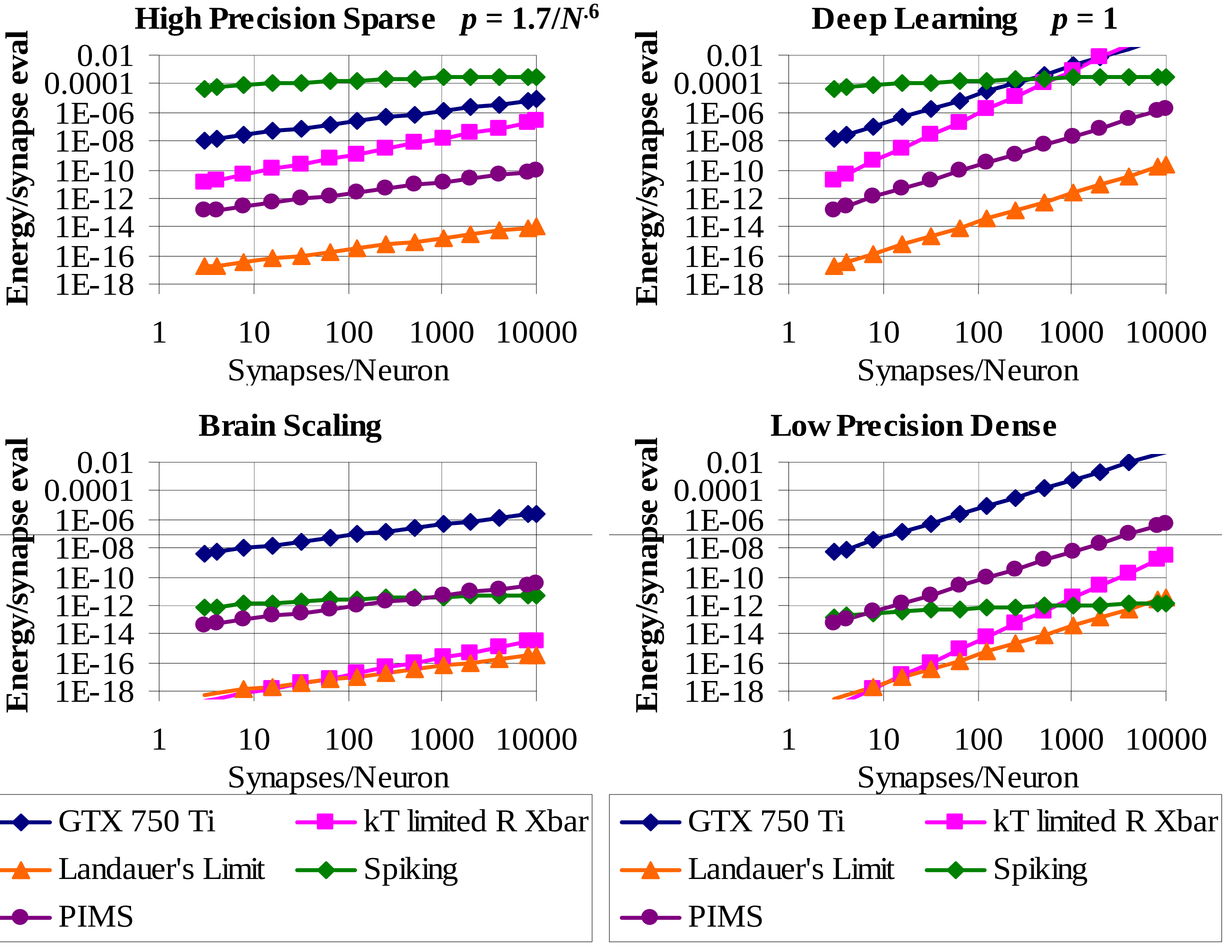
- Preview
- CMOS limits
- Resistive crossbar
- Improving energy efficiency and eating words
- Algorithm example
- Spiking
- Roll up

“Beyond Moore” issue preview

- Claimed objectives
  - Beat CMOS energy efficiency
  - Scale up
- Analytical approach
  - Devise theoretical performance limits
  - CMOS will approach its limits
  - Others approaches ought to have the theoretical ability to beat CMOS at limits, or they have no chance in practice

B = 16; 65536 levels

B = 3; 8 levels



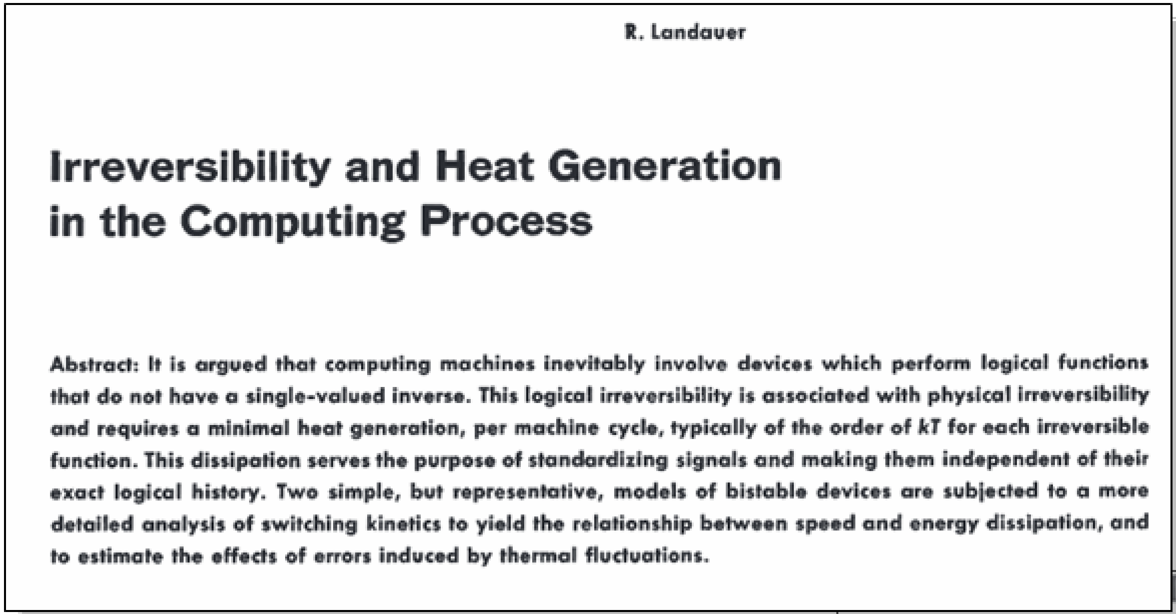
Test cases

Implementation	Energy ($N \times M$ matrix, $N = M$, L-level values, $B = \log_2 L$)
nVidia GTX 750 Ti	A simple analysis indicates the nVidia GTX 750 Ti (a state of the art consumer GPU at the time of this writing, costing $150) will be memory bandwidth limited. The computational strategy is the assume it will consume its standard 60 Watts and process data as fast as it can be read from memory. The memory rate will be based on synapse values of the specified number of bits + 4 bits of sparsity control information from [(appendix 1)].
Resistive crossbar	1/24 $N^2 L^2 p^3 M$ kT, as discussed in text.
Landauer's Limit	Full adder is 3 kT. Energy is 3 $N^2 p^2 B^2$ kT ln 2
Neuristor spike train	$L^2 \log_2 N$ spike energy = $2^{2B} \log_2 N\, E_{spike}$; 6-60 fJ/spike for a neuristor (we use 6 fJ), as disclosed
PIMS	See author's other paper; TFET logic and adiabatic memory

Outline

- Preview
- CMOS limits
- Resistive crossbar
- Improving energy efficiency and eating words
- Algorithm example
- Spiking
- Roll up

Theoretical limits for CMOS

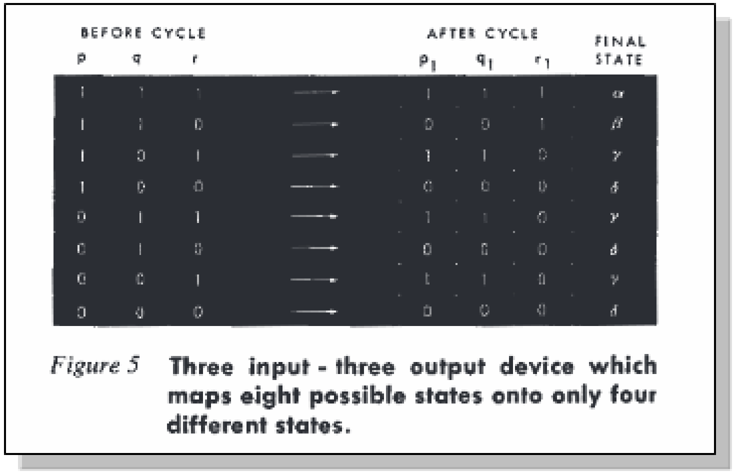


R. Landauer

Irreversibility and Heat Generation in the Computing Process

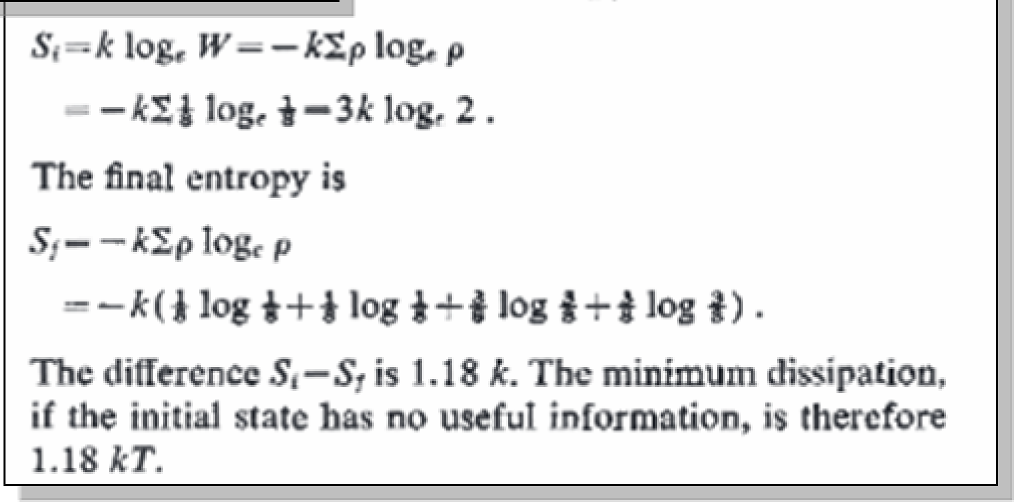
Abstract: It is argued that computing machines inevitably involve devices which perform logical functions that do not have a single-valued inverse. This logical irreversibility is associated with physical irreversibility and requires a minimal heat generation, per machine cycle, typically of the order of kT for each irreversible function. This dissipation serves the purpose of standardizing signals and making them independent of their exact logical history. Two simple, but representative, models of bistable devices are subjected to a more detailed analysis of switching kinetics to yield the relationship between speed and energy dissipation, and to estimate the effects of errors induced by thermal fluctuations.

- Landauer’s paper derives limits that do not depend on device properties



BEFORE CYCLE				AFTER CYCLE			FINAL STATE
p	q	r		p_1	q_1	r_1	
1	1	1	→	1	1	1	α
1	1	0	→	0	0	1	β
1	0	1	→	1	1	0	γ
1	0	0	→	0	0	0	δ
0	1	1	→	1	1	0	γ
0	1	0	→	0	0	0	δ
0	0	1	→	1	1	0	γ
0	0	0	→	0	0	0	δ

Figure 5 Three input - three output device which maps eight possible states onto only four different states.



The initial entropy was

$$S_i = k \log_e W = -k\Sigma\rho \log_e \rho$$

$$= -k\Sigma\tfrac{1}{8} \log_e \tfrac{1}{8} = 3k \log_e 2 \,.$$

The final entropy is

$$S_f = -k\Sigma\rho \log_e \rho$$

$$= -k(\tfrac{1}{8} \log \tfrac{1}{8} + \tfrac{1}{8} \log \tfrac{1}{8} + \tfrac{3}{8} \log \tfrac{3}{8} + \tfrac{3}{8} \log \tfrac{3}{8}) \,.$$

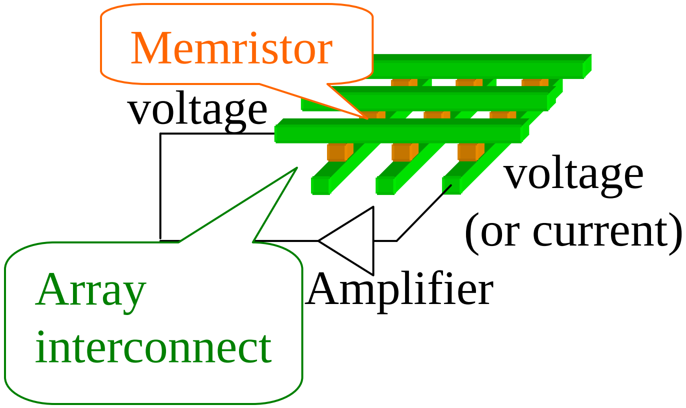
The difference $S_i - S_f$ is 1.18 k. The minimum dissipation, if the initial state has no useful information, is therefore 1.18 kT.

Outline

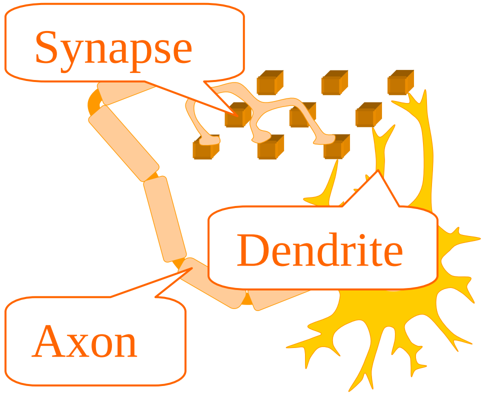
- Preview
- CMOS limits
- Resistive crossbar
- Improving energy efficiency and eating words
- Algorithm example
- Spiking
- Roll up

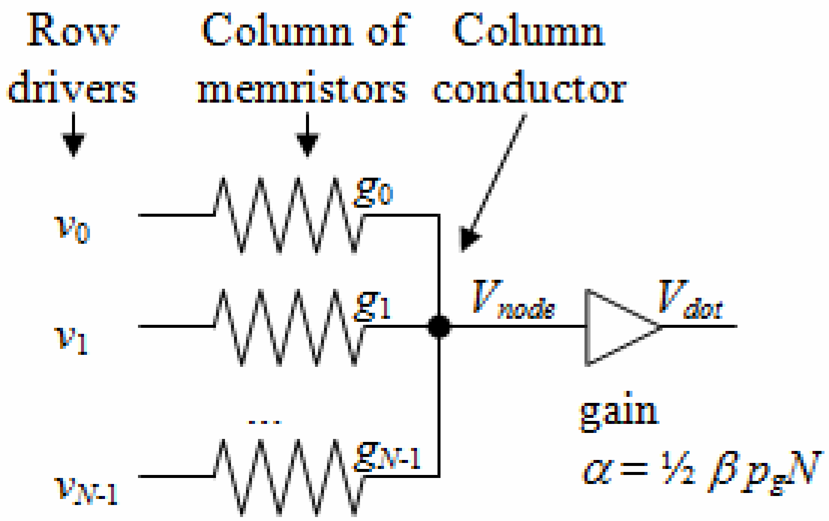
Resistive crossbar

A. Artificial neural network:



B. Natural neural network deformed to show equivalence:





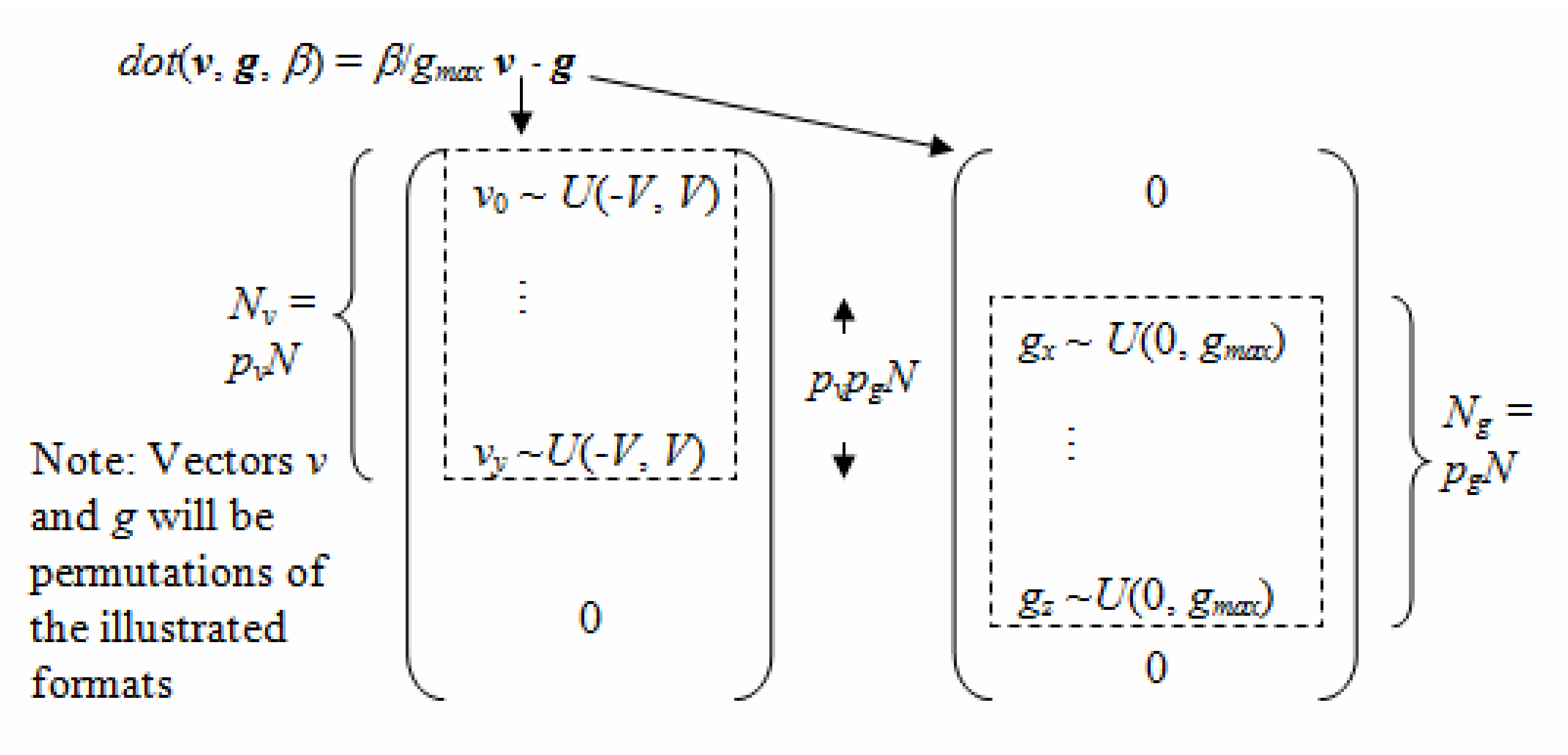
$$V_{node} = \frac{\sum_i v_i g_i}{\sum_i g_i}$$

Assume

$$\sum_i g_i = ½\, g_{max}\, p_g N$$

$$V_{dot} = \beta / g_{max} \sum_i v_i g_i$$

Sparse dot product problem setup



$dot(\mathbf{v}, \mathbf{g}, \beta) = \beta / g_{max}\, \mathbf{v} \cdot \mathbf{g}$

$N_v =$ $p_v N$

$v_0 \sim U(-V, V)$

$\vdots$

$v_y \sim U(-V, V)$

0

$p_v p_g N$

0

$g_x \sim U(0, g_{max})$

$\vdots$

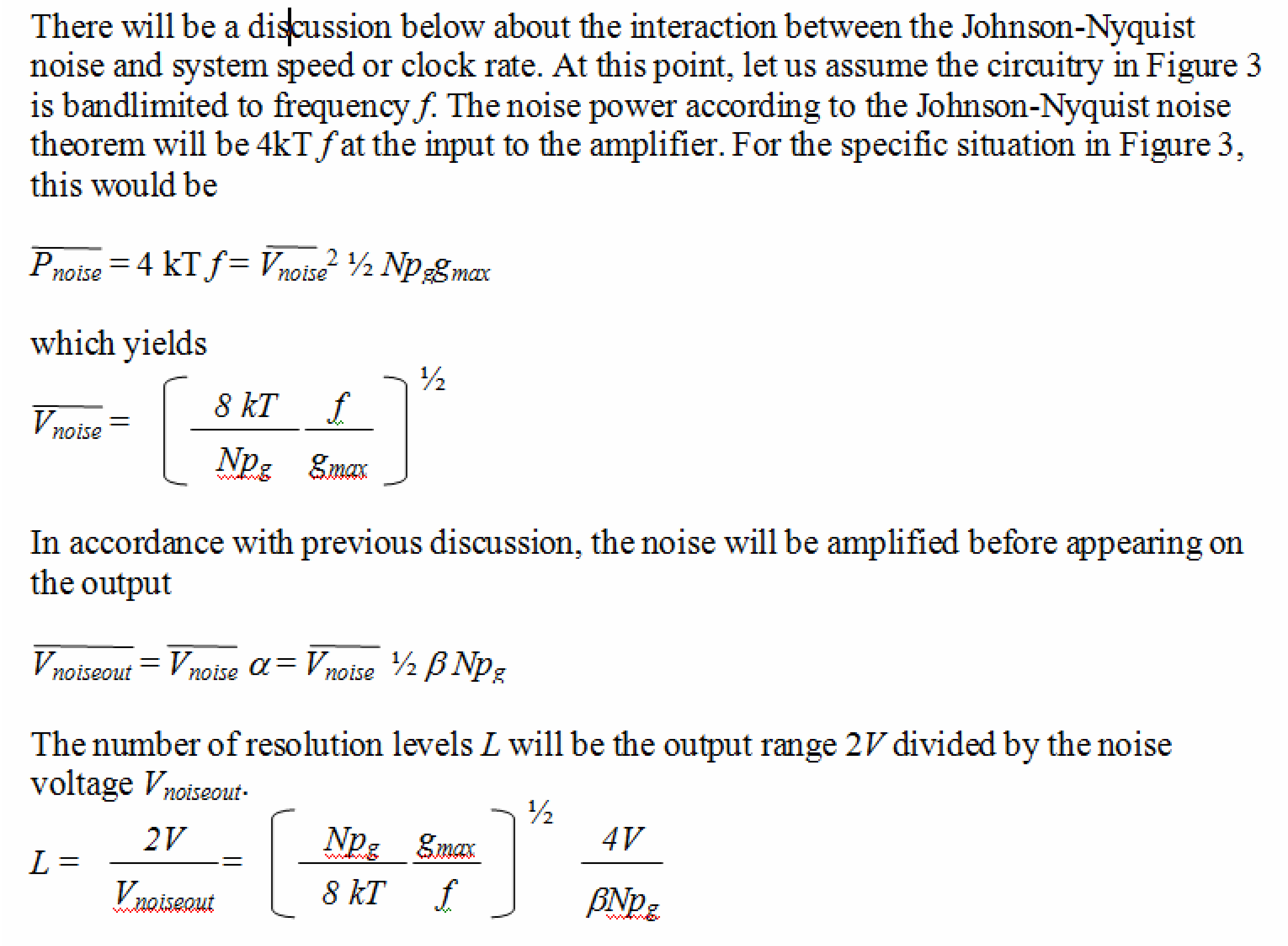
$g_z \sim U(0, g_{max})$

0

$N_g =$ $p_g N$

Note: Vectors *v* and *g* will be permutations of the illustrated formats

Algebra



There will be a discussion below about the interaction between the Johnson-Nyquist noise and system speed or clock rate. At this point, let us assume the circuitry in Figure 3 is bandlimited to frequency *f*. The noise power according to the Johnson-Nyquist noise theorem will be 4kT *f* at the input to the amplifier. For the specific situation in Figure 3, this would be

$$\overline{P_{noise}} = 4\ \mathrm{kT}\ f = \overline{V_{noise}}^{2}\ \tfrac{1}{2}\ Np_{g}g_{max}$$

which yields

$$\overline{V_{noise}} = \left[\frac{8\ kT}{Np_{g}}\ \frac{f}{g_{max}} \right]^{\frac{1}{2}}$$

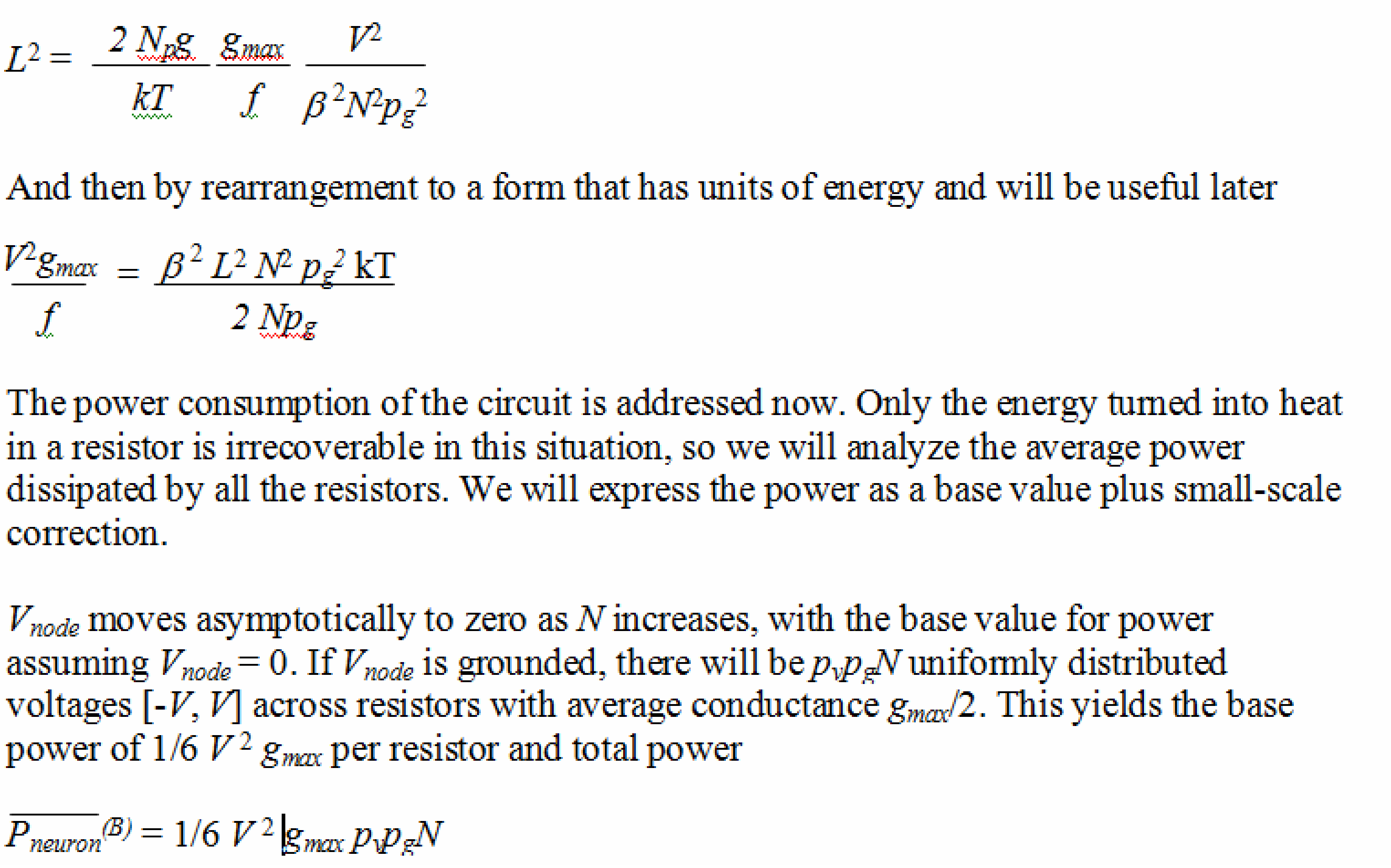
In accordance with previous discussion, the noise will be amplified before appearing on the output

$$\overline{V_{noiseout}} = \overline{V_{noise}}\ \alpha = \overline{V_{noise}}\ \tfrac{1}{2}\ \beta\, Np_{g}$$

The number of resolution levels *L* will be the output range 2*V* divided by the noise voltage $V_{noiseout}$.

$$L = \frac{2V}{V_{noiseout}} = \left[\frac{Np_{g}}{8\ kT}\ \frac{g_{max}}{f} \right]^{\frac{1}{2}} \frac{4V}{\beta Np_{g}}$$

Algebra



$$L^2 = \frac{2\,N_p g}{kT}\;\frac{g_{max}}{f}\;\frac{V^2}{\beta^2 N^2 p_g^2}$$

And then by rearrangement to a form that has units of energy and will be useful later

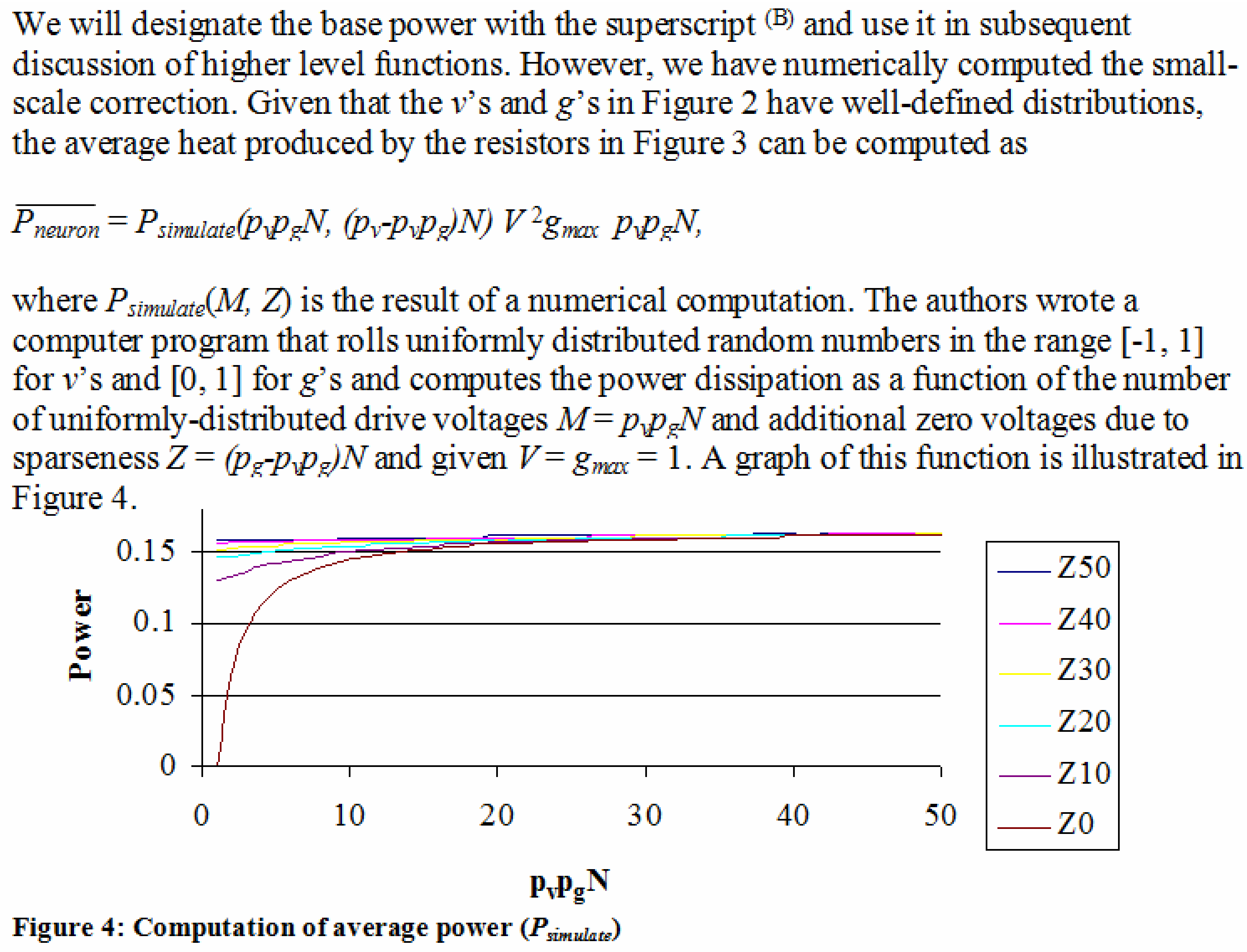
$$\frac{V^2 g_{max}}{f} = \frac{\beta^2 L^2 N^2 p_g^2\,\mathrm{kT}}{2\,Np_g}$$

The power consumption of the circuit is addressed now. Only the energy turned into heat in a resistor is irrecoverable in this situation, so we will analyze the average power dissipated by all the resistors. We will express the power as a base value plus small-scale correction.

V_{node} moves asymptotically to zero as N increases, with the base value for power assuming $V_{node} = 0$. If V_{node} is grounded, there will be $p_V p_g N$ uniformly distributed voltages $[-V, V]$ across resistors with average conductance $g_{max}/2$. This yields the base power of $1/6\ V^2\ g_{max}$ per resistor and total power

$$\overline{P_{neuron}}^{(B)} = 1/6\ V^2 g_{max}\, p_V p_g N$$

Algebra



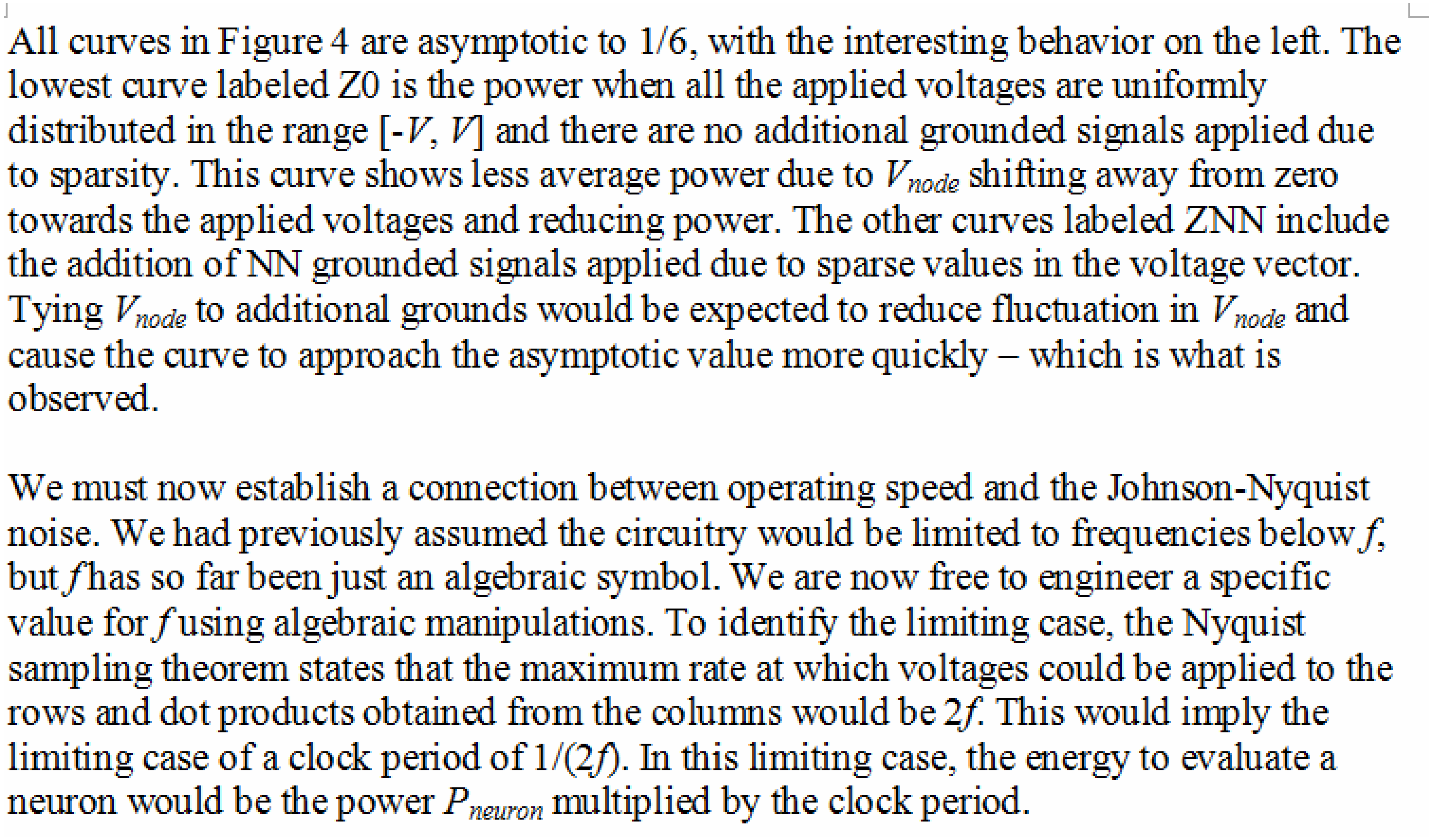
We will designate the base power with the superscript (B) and use it in subsequent discussion of higher level functions. However, we have numerically computed the small-scale correction. Given that the *v*'s and *g*'s in Figure 2 have well-defined distributions, the average heat produced by the resistors in Figure 3 can be computed as

$$\overline{P_{neuron}} = P_{simulate}(p_v p_g N,\ (p_v\text{-}p_v p_g)N)\ V^2 g_{max}\ \ p_v p_g N,$$

where $P_{simulate}(M, Z)$ is the result of a numerical computation. The authors wrote a computer program that rolls uniformly distributed random numbers in the range [-1, 1] for *v*'s and [0, 1] for *g*'s and computes the power dissipation as a function of the number of uniformly-distributed drive voltages $M = p_v p_g N$ and additional zero voltages due to sparseness $Z = (p_g\text{-}p_v p_g)N$ and given $V = g_{max} = 1$. A graph of this function is illustrated in Figure 4.

Figure 4: Computation of average power ($P_{simulate}$)

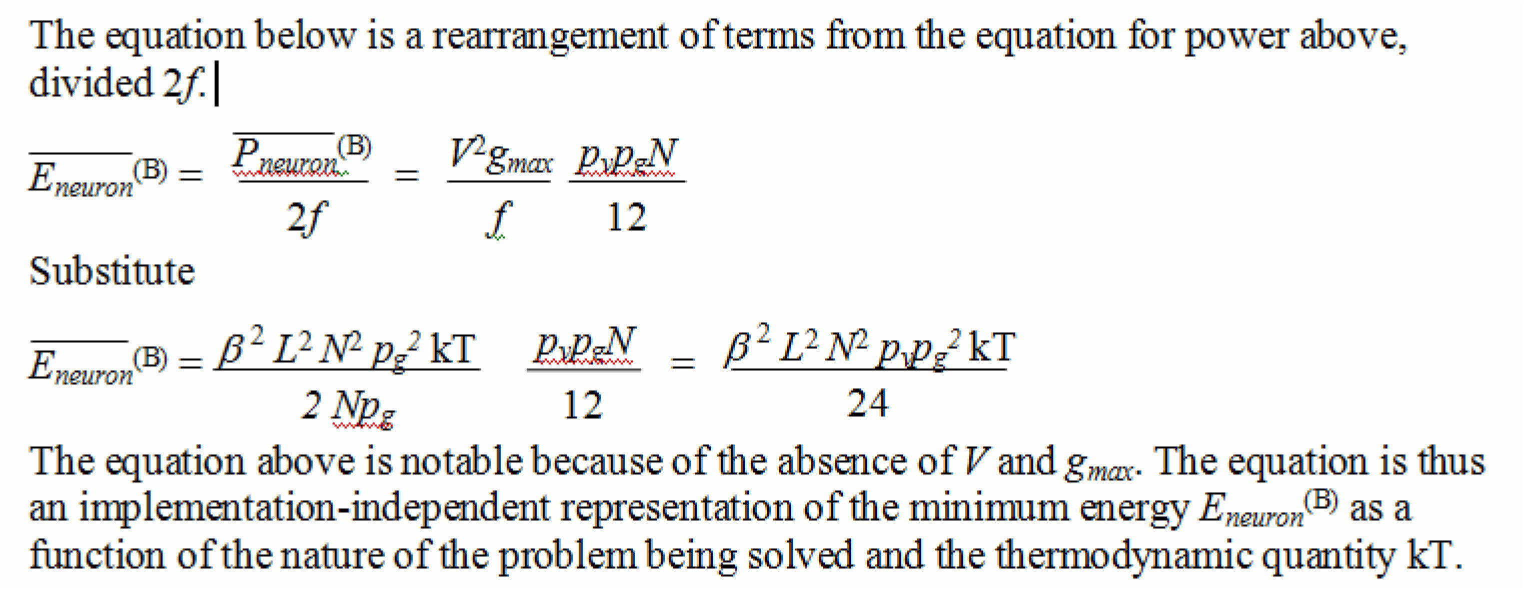
Algebra



All curves in Figure 4 are asymptotic to 1/6, with the interesting behavior on the left. The lowest curve labeled Z0 is the power when all the applied voltages are uniformly distributed in the range [-*V*, *V*] and there are no additional grounded signals applied due to sparsity. This curve shows less average power due to V_{node} shifting away from zero towards the applied voltages and reducing power. The other curves labeled ZNN include the addition of NN grounded signals applied due to sparse values in the voltage vector. Tying V_{node} to additional grounds would be expected to reduce fluctuation in V_{node} and cause the curve to approach the asymptotic value more quickly – which is what is observed.

We must now establish a connection between operating speed and the Johnson-Nyquist noise. We had previously assumed the circuitry would be limited to frequencies below *f*, but *f* has so far been just an algebraic symbol. We are now free to engineer a specific value for *f* using algebraic manipulations. To identify the limiting case, the Nyquist sampling theorem states that the maximum rate at which voltages could be applied to the rows and dot products obtained from the columns would be 2*f*. This would imply the limiting case of a clock period of 1/(2*f*). In this limiting case, the energy to evaluate a neuron would be the power P_{neuron} multiplied by the clock period.

Algebra



The equation below is a rearrangement of terms from the equation for power above, divided 2*f*.

$$\overline{E_{neuron}}^{(B)} = \frac{\overline{P_{neuron}}^{(B)}}{2f} = \frac{V^2 g_{max}}{f} \frac{p_v p_g N}{12}$$

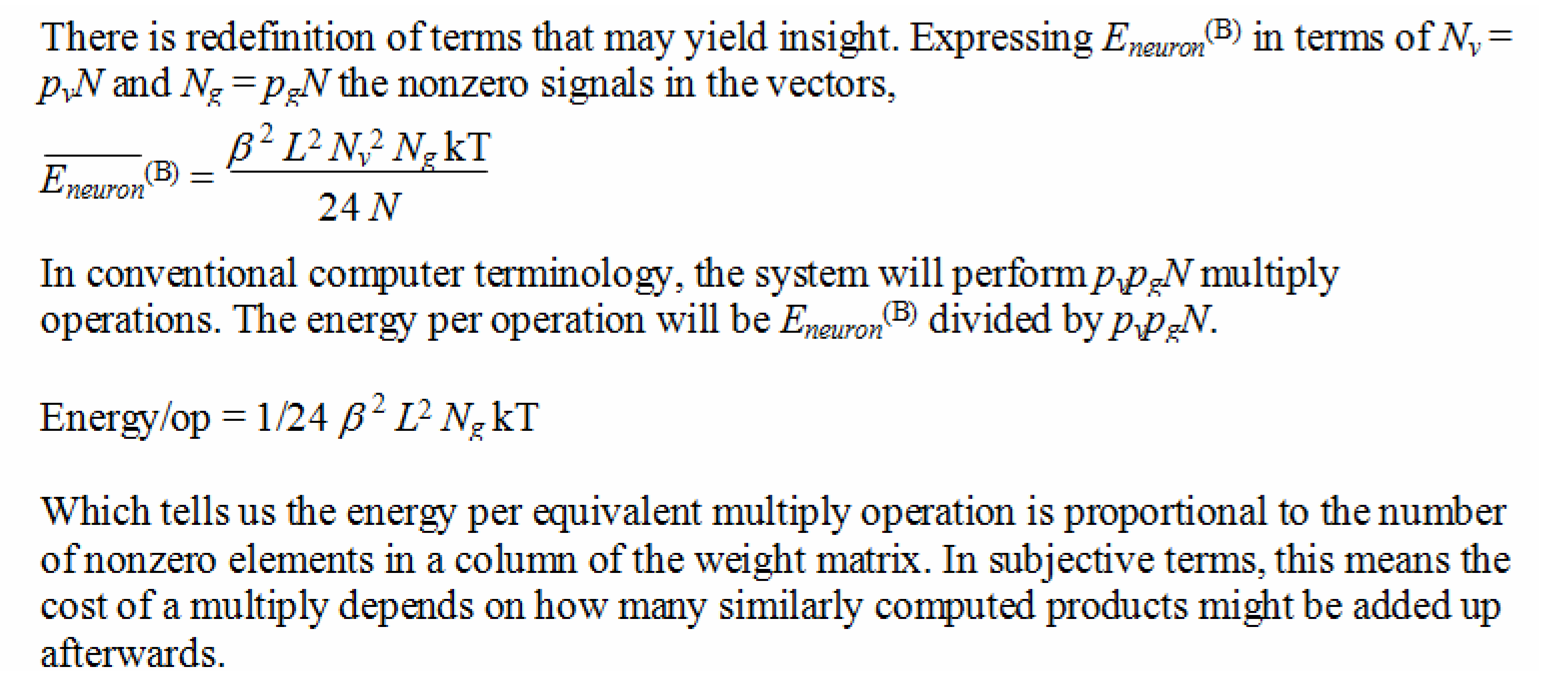
Substitute

$$\overline{E_{neuron}}^{(B)} = \frac{\beta^2 L^2 N^2 p_g^2 \mathrm{kT}}{2\, N p_g} \frac{p_v p_g N}{12} = \frac{\beta^2 L^2 N^2 p_v p_g^2 \mathrm{kT}}{24}$$

The equation above is notable because of the absence of V and g_{max}. The equation is thus an implementation-independent representation of the minimum energy $E_{neuron}^{(B)}$ as a function of the nature of the problem being solved and the thermodynamic quantity kT.

- $N \times M$ matrix; L-level signal resolution
- β extra gain
- p_v p_g sparsity proportion on v and g

Reinterpretation



There is redefinition of terms that may yield insight. Expressing $E_{neuron}^{(B)}$ in terms of $N_v = p_v N$ and $N_g = p_g N$ the nonzero signals in the vectors,

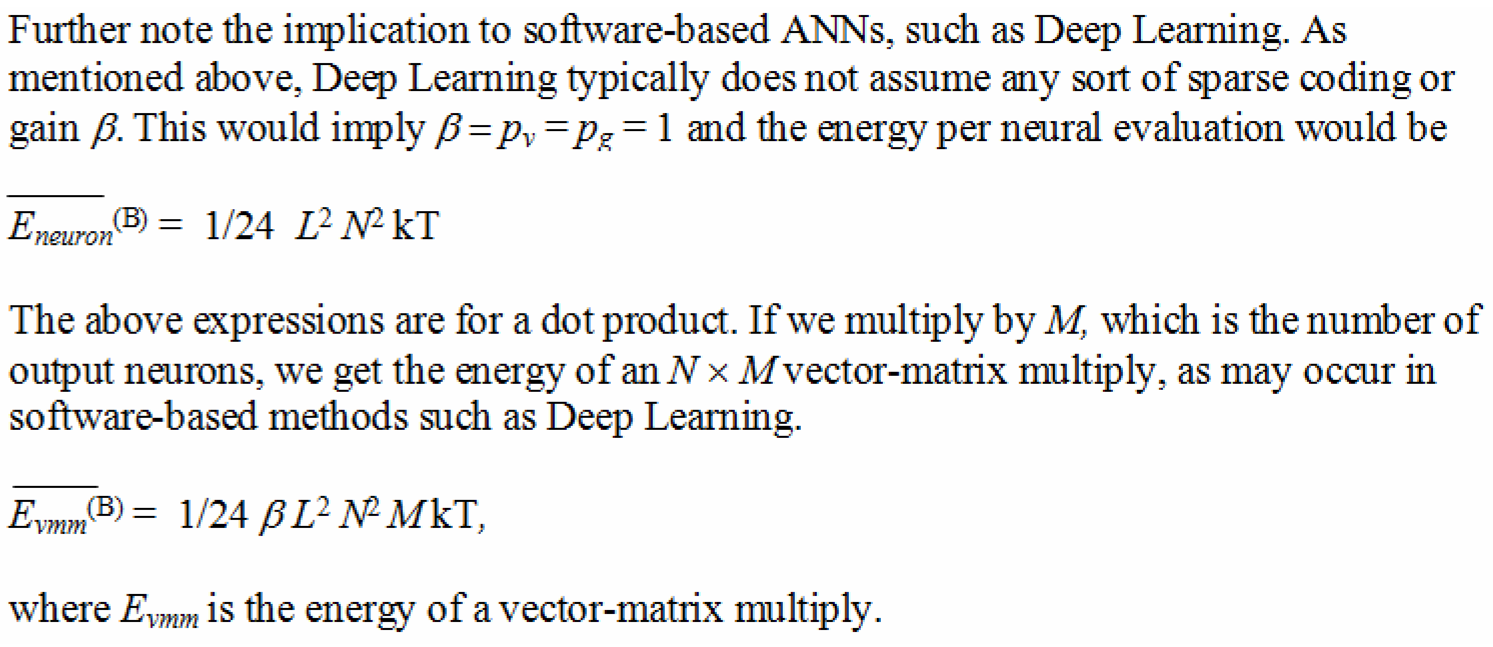
$$\overline{E_{neuron}}^{(B)} = \frac{\beta^2 L^2 N_v^2 N_g \mathrm{kT}}{24\, N}$$

In conventional computer terminology, the system will perform $p_v p_g N$ multiply operations. The energy per operation will be $E_{neuron}^{(B)}$ divided by $p_v p_g N$.

Energy/op = $1/24\ \beta^2 L^2 N_g \mathrm{kT}$

Which tells us the energy per equivalent multiply operation is proportional to the number of nonzero elements in a column of the weight matrix. In subjective terms, this means the cost of a multiply depends on how many similarly computed products might be added up afterwards.

Reinterpretation



Further note the implication to software-based ANNs, such as Deep Learning. As mentioned above, Deep Learning typically does not assume any sort of sparse coding or gain β. This would imply $\beta = p_v = p_g = 1$ and the energy per neural evaluation would be

$$\overline{E_{neuron}}^{(B)} = \ 1/24 \ \ L^2 N^2 \, \mathrm{kT}$$

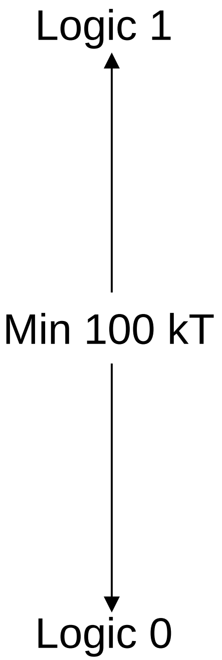
The above expressions are for a dot product. If we multiply by M, which is the number of output neurons, we get the energy of an $N \times M$ vector-matrix multiply, as may occur in software-based methods such as Deep Learning.

$$\overline{E_{vmm}}^{(B)} = \ 1/24 \ \beta \, L^2 N^2 M \, \mathrm{kT},$$

where E_{vmm} is the energy of a vector-matrix multiply.

This method has a problem

Digital representation:

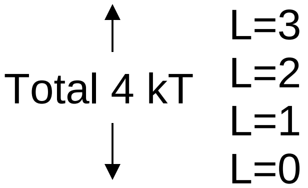


Logic 1

Min 100 kT

Logic 0

L=4 level analog representation:



Total 4 kT

L=3

L=2

L=1

L=0

Probability of digital “mistake”:
e^{-100}

Probability of analog signal mistake by one level:
$1/e$

Probably of analog signal being totally wrong:
e^{-4} or e^{-L}

Haven’t worked this out yet. Means analog doesn’t work as well as predicted.

Outline

- Preview
- CMOS limits
- Resistive crossbar
- Improving energy efficiency and eating words
- Algorithm example
- Spiking
- Roll up

Improving energy efficiency

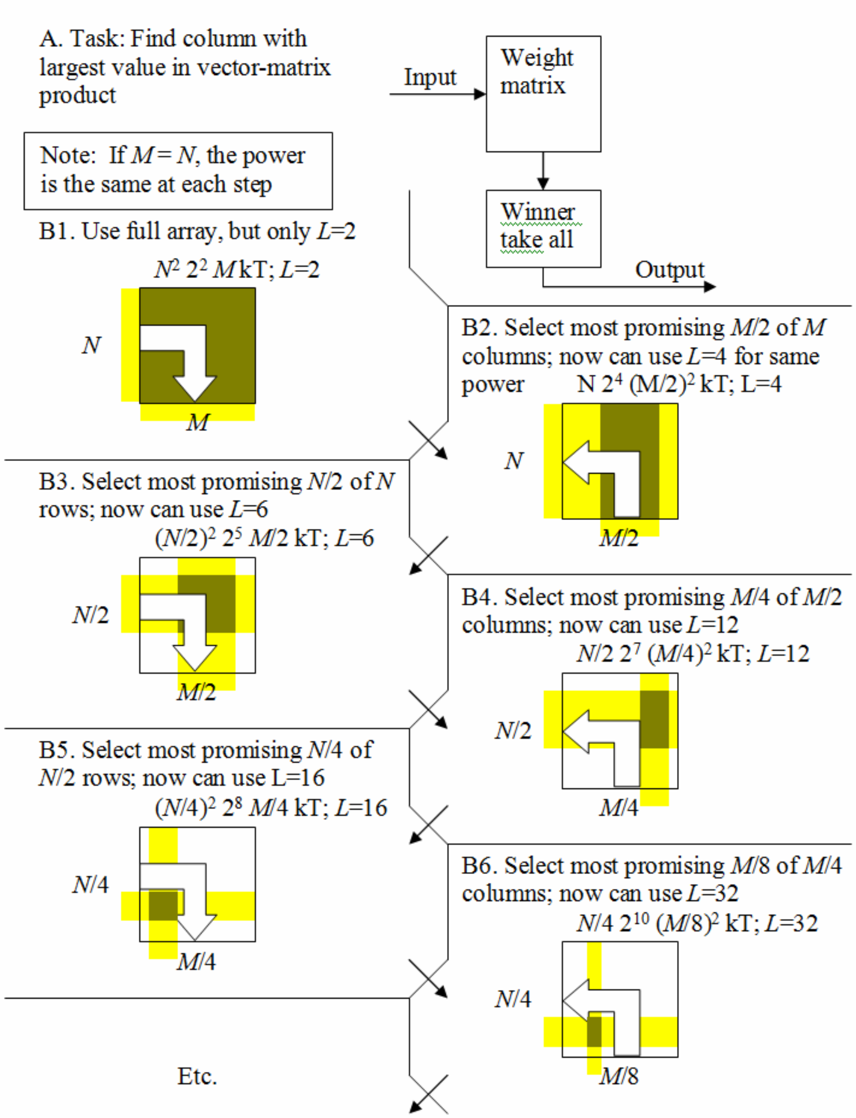
	E_{mvm}	p_g	Clock cycles	Clock period	P = *Power*
Analog array	$O(N^2 L^2 M)$ kT	1	1	T_{clock}	$\propto 1$
j steps × $1/j$ th size	$j\, O(N^2 L^2 p_g{}^2 M)$ kT + jME_{add} $= O(1/j \times N^2 L^2 M)$ kT + jME_{add} $\leq O((N^2 L + j^2)/j\ ML)$ kT	$1/j$	j	T_{clock}/j	$\propto 1/j$
$j = N$ full extension	$O(N L^2 M)$ kT + NME_{add}	$1/N$	N	T_{clock}/N	$\propto 1/N$

- However, the problem with this idea is that it turns an analog neural network into a the digital version
  - Works fine, but causes you to eat your words

Outline

- Preview
- CMOS limits
- Resistive crossbar
- Improving energy efficiency and eating words
- Algorithm example
- Spiking
- Roll up

Algorithm example



A. Task: Find column with largest value in vector-matrix product

Note: If $M = N$, the power is the same at each step

B1. Use full array, but only L=2

$N^2\ 2^2\ M\,\mathrm{kT};\ L=2$

B2. Select most promising M/2 of M columns; now can use L=4 for same power

$N\ 2^4\ (M/2)^2\ \mathrm{kT};\ L=4$

B3. Select most promising N/2 of N rows; now can use L=6

$(N/2)^2\ 2^5\ M/2\ \mathrm{kT};\ L=6$

B4. Select most promising M/4 of M/2 columns; now can use L=12

$N/2\ 2^7\ (M/4)^2\ \mathrm{kT};\ L=12$

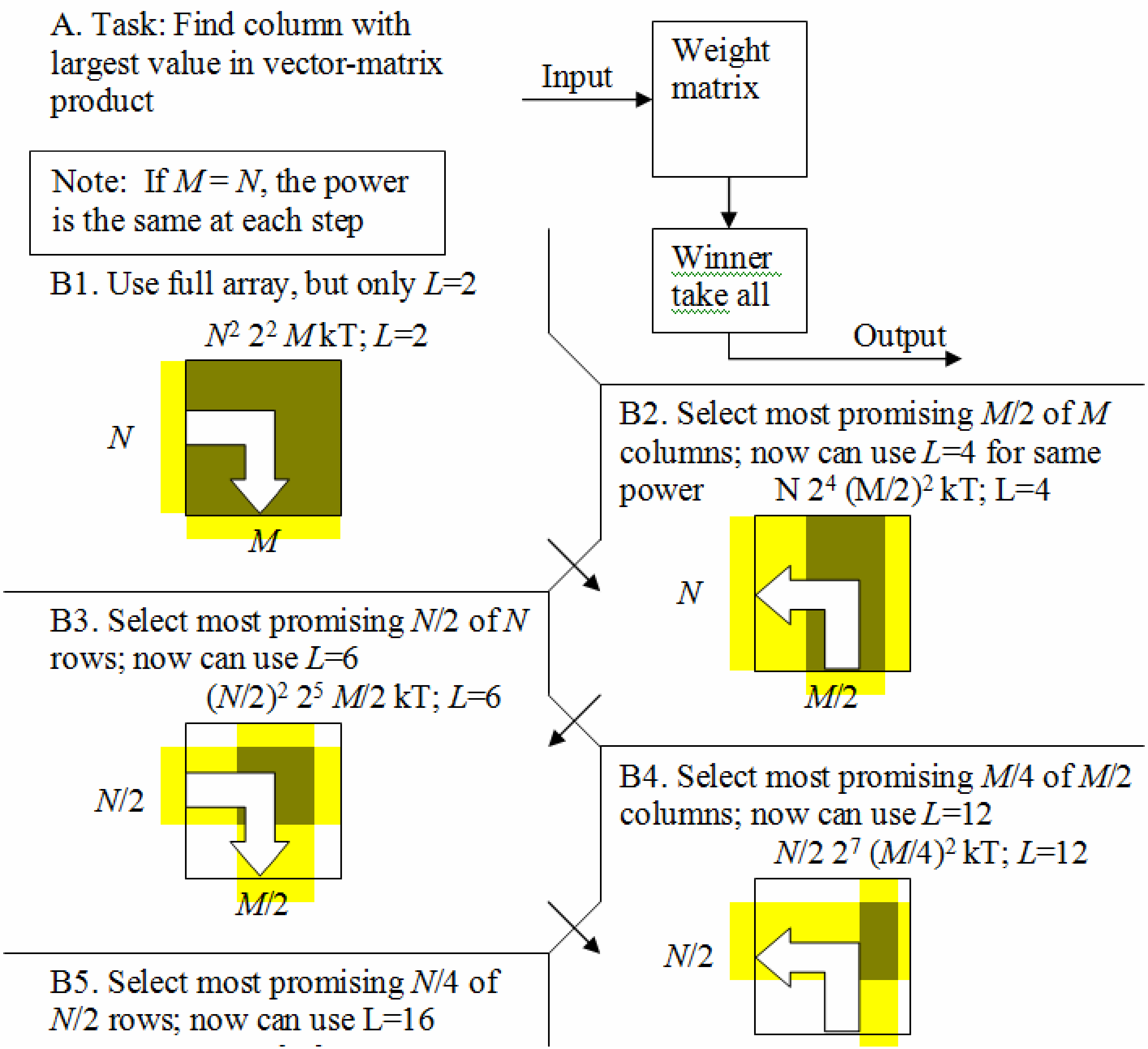
B5. Select most promising N/4 of N/2 rows; now can use L=16

$(N/4)^2\ 2^8\ M/4\ \mathrm{kT};\ L=16$

B6. Select most promising M/8 of M/4 columns; now can use L=32

$N/4\ 2^{10}\ (M/8)^2\ \mathrm{kT};\ L=32$

Etc.



A. Task: Find column with largest value in vector-matrix product

Note: If $M = N$, the power is the same at each step

B1. Use full array, but only L=2

$N^2\ 2^2\ M\,\mathrm{kT};\ L=2$

B2. Select most promising M/2 of M columns; now can use L=4 for same power

$\mathrm{N}\ 2^4\ (\mathrm{M}/2)^2\ \mathrm{kT};\ \mathrm{L}=4$

B3. Select most promising N/2 of N rows; now can use L=6

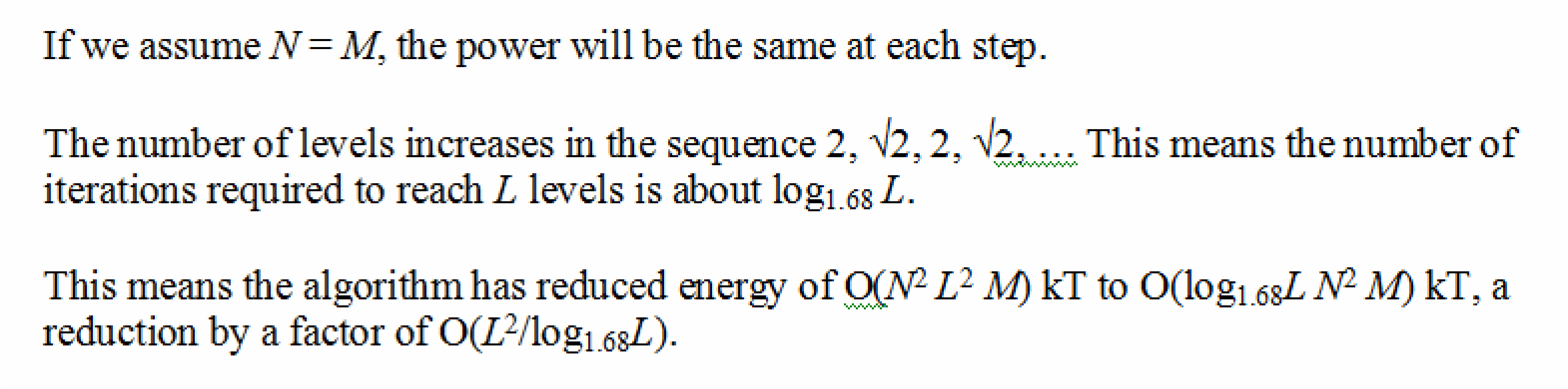
$(N/2)^2\ 2^5\ M/2\ \mathrm{kT};\ L=6$

B4. Select most promising M/4 of M/2 columns; now can use L=12

$N/2\ 2^7\ (M/4)^2\ \mathrm{kT};\ L=12$

B5. Select most promising N/4 of N/2 rows; now can use L=16

Algorithm quantitative benefit



If we assume $N = M$, the power will be the same at each step.

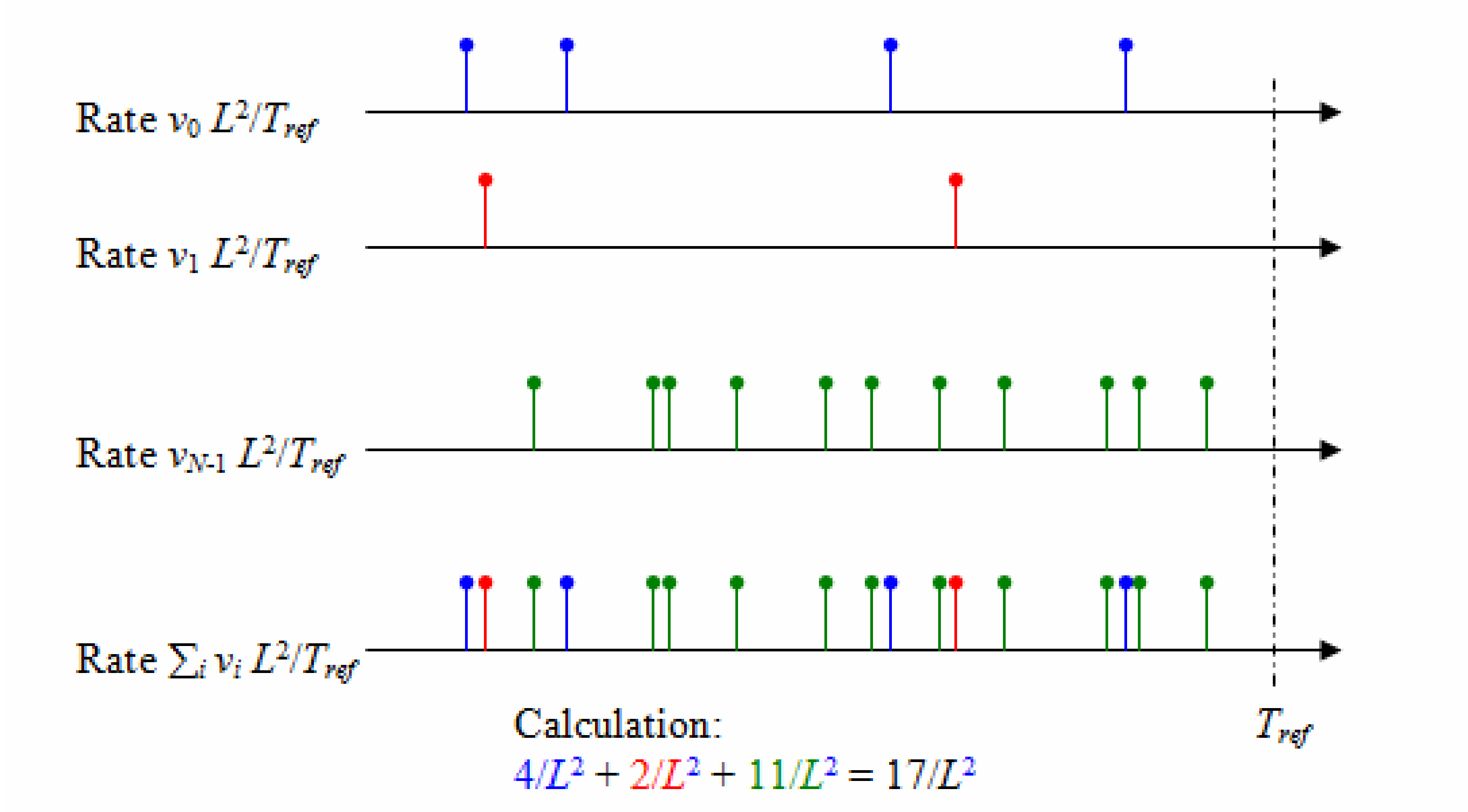
The number of levels increases in the sequence 2, √2, 2, √2, … This means the number of iterations required to reach L levels is about $\log_{1.68} L$.

This means the algorithm has reduced energy of $O(N^2 L^2 M)$ kT to $O(\log_{1.68} L\, N^2 M)$ kT, a reduction by a factor of $O(L^2/\log_{1.68} L)$.

Outline

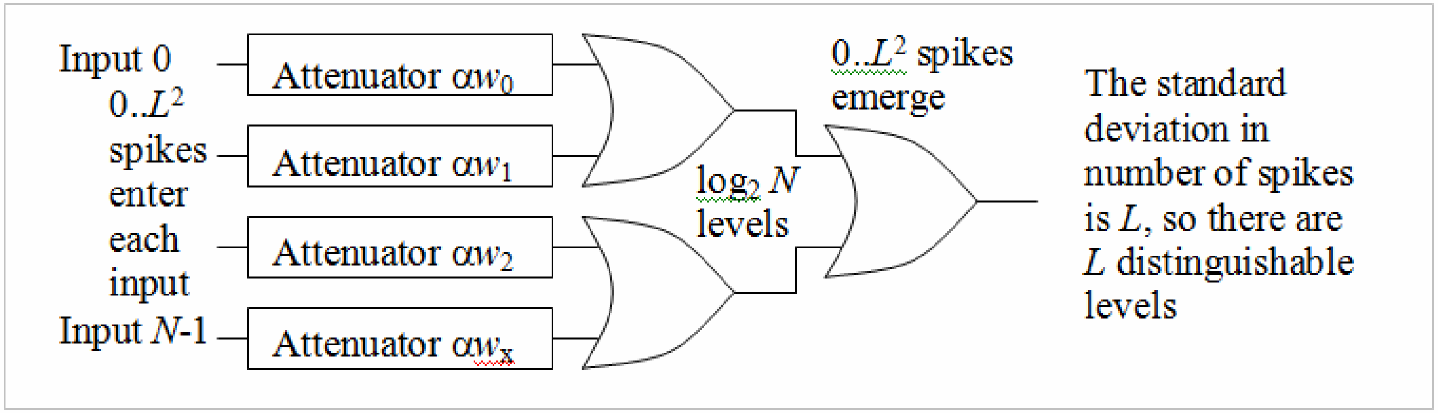
- Preview
- CMOS limits
- Resistive crossbar
- Improving energy efficiency and eating words
- Algorithm example
- Spiking
- Roll up

Spike representation


Rate $\nu_0 L^2/T_{ref}$
Rate $\nu_1 L^2/T_{ref}$
Rate $\nu_{N\text{-}1} L^2/T_{ref}$
Rate $\sum_i \nu_i L^2/T_{ref}$
Calculation:
$4/L^2 + 2/L^2 + 11/L^2 = 17/L^2$
T_{ref}

Backup: adiabatic memory (low) maturity level

A. Spiking dot product



Input 0

0..L^2 spikes enter each input

Input N-1

Attenuator αw_0

Attenuator αw_1

Attenuator αw_2

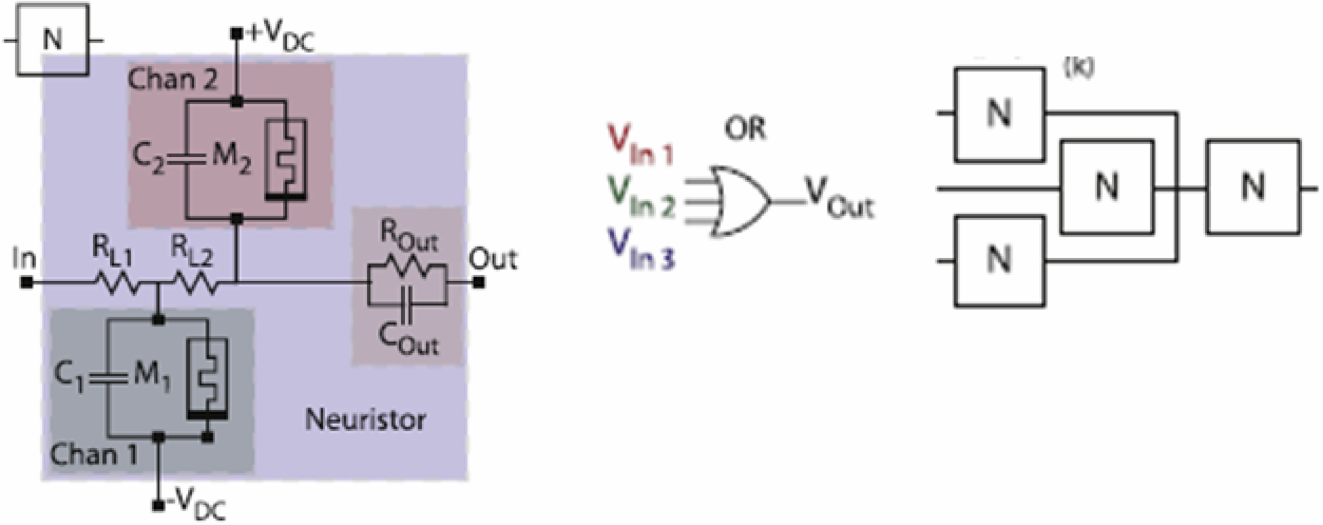
Attenuator αw_x

$\log_2 N$ levels

0..L^2 spikes emerge

The standard deviation in number of spikes is L, so there are L distinguishable levels

B. Neuristor OR gate (attenuator not known):



Outline

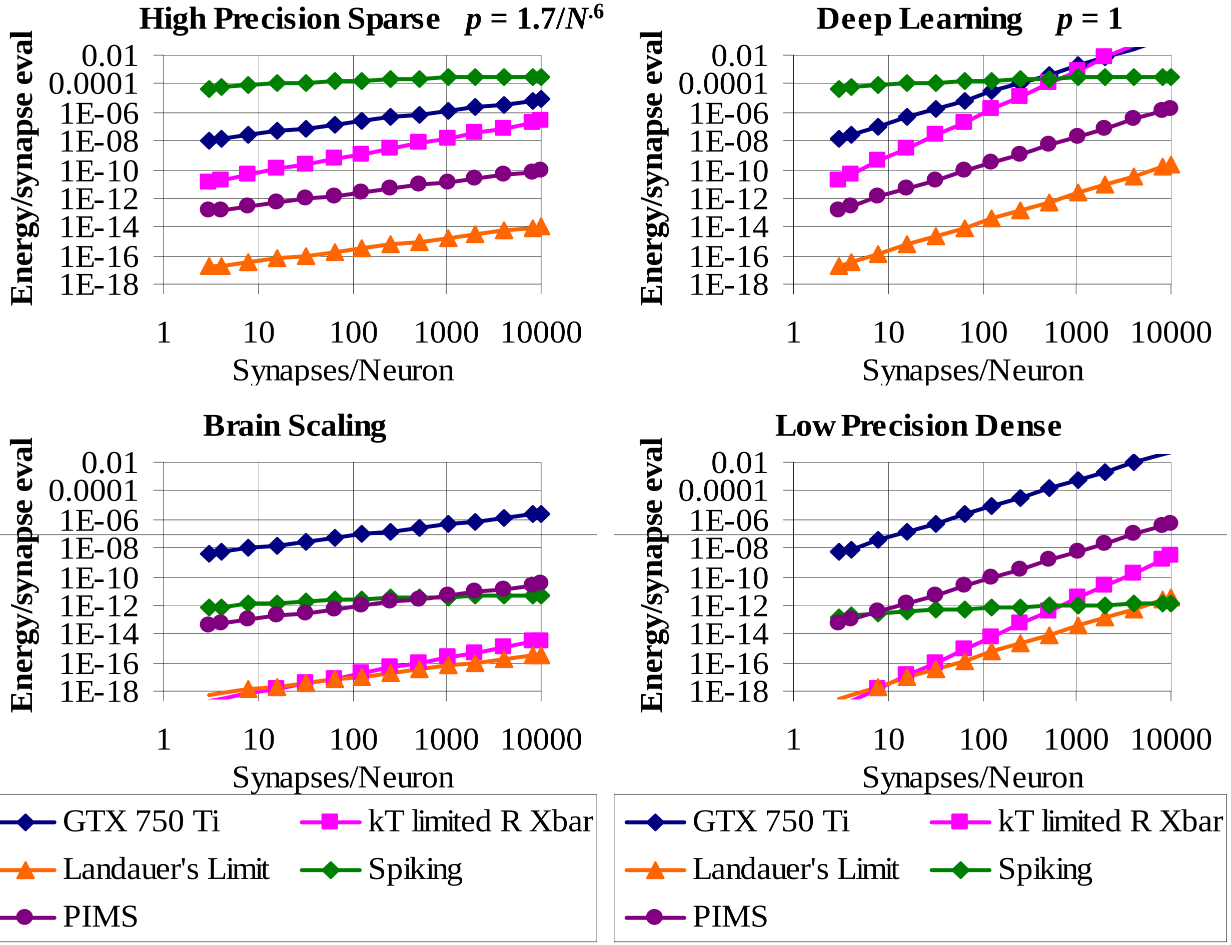
- Preview
- CMOS limits
- Resistive crossbar
- Improving energy efficiency and eating words
- Algorithm example
- Spiking
- Roll up

“Beyond Moore” issue preview

- Claimed objectives
  - Beat CMOS energy efficiency
  - Scale up
- Analytical approach
  - Devise theoretical performance limits
  - CMOS will approach its limits
  - Others approaches ought to have the theoretical ability to beat CMOS at limits, or they have no chance in practice

B **= 16; 65536 levels**

B **= 3; 8 levels**



High Precision Sparse ***p*** **= 1.7/*N*·6**

Deep Learning ***p*** **= 1**

Brain Scaling

Low Precision Dense

Conclusions

- Deep learning has evolved to be efficient on digital computers
  - High resolution and lack of control of sparsity make it a challenge for analog electronics
- Biology tried the level-based approach
  - Uses it for worms and retinas, but not much more. Maybe we have figured out why
- Spiking approach has very different characteristics
  - Biology uses it for higher-level thinking
  - Neuristor looks interesting, but no complementary synapse (?)
    - Looking for a circuit that deletes a spike with probability *w*.
- Algorithms stack seems feasible
- This is work in progress; still not sure about analog level separation and digital vs. analog reliability