

Memristor Crossbar Array Simulation for Deep Learning Applications

Supplemental material

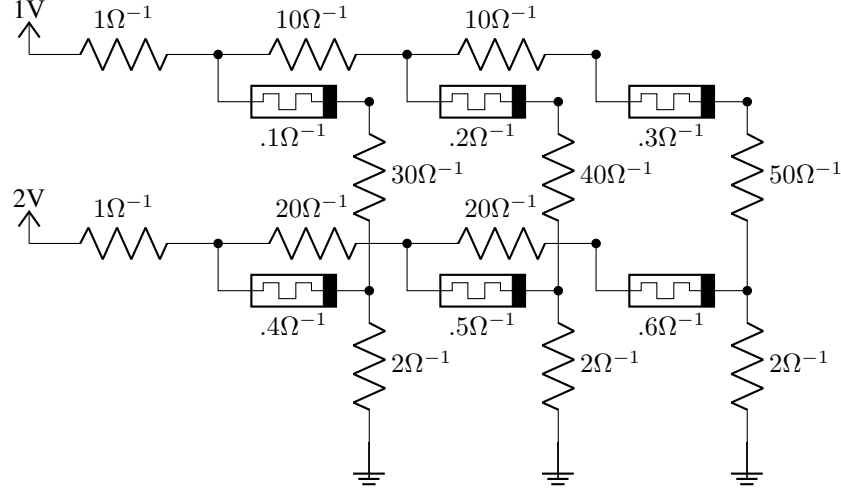


Fig. 1. Memristor Crossbar Array (2 rows, 3 columns)

I. EXAMPLE: CONSTRUCTION OF \mathbf{G}_{ABCD}

Here we show a very small example of how to construct \mathbf{G}_{ABCD} , indexed as indicated in Fig. 2 in [1].

From Fig. 1 we derive the following equations:

$$\begin{aligned}
 i_{1,0}^{t1} &= i_{1,1}^{t1} + i_{1,1} \rightarrow \frac{v_{1,0}^{t1} - v_{1,1}^{t1}}{r_{1,0}^{t1}} = \frac{v_{1,1}^{t1} - v_{1,2}^{t1}}{r_{1,1}^{t1}} + \frac{v_{1,1}^{t1} - v_{1,1}^{b1}}{r_{1,1}} \\
 i_{1,1}^{t1} &= i_{1,2}^{t1} + i_{1,2} \rightarrow \frac{v_{1,1}^{t1} - v_{1,2}^{t1}}{r_{1,1}^{t1}} = \frac{v_{1,2}^{t1} - v_{1,3}^{t1}}{r_{1,2}^{t1}} + \frac{v_{1,2}^{t1} - v_{1,2}^{b1}}{r_{1,2}} \\
 i_{1,2}^{t1} &= i_{1,3}^{t1} + i_{1,3} \rightarrow \frac{v_{1,2}^{t1} - v_{1,3}^{t1}}{r_{1,2}^{t1}} = \frac{v_{1,3}^{t1} - v_{1,4}^{t1}}{r_{1,3}^{t1}} + \frac{v_{1,3}^{t1} - v_{1,3}^{b1}}{r_{1,3}} \\
 i_{2,0}^{t1} &= i_{2,1}^{t1} + i_{2,1} \rightarrow \frac{v_{2,0}^{t1} - v_{2,1}^{t1}}{r_{2,0}^{t1}} = \frac{v_{2,1}^{t1} - v_{2,2}^{t1}}{r_{2,1}^{t1}} + \frac{v_{2,1}^{t1} - v_{2,1}^{b1}}{r_{2,1}} \\
 i_{2,1}^{t1} &= i_{2,2}^{t1} + i_{2,2} \rightarrow \frac{v_{2,1}^{t1} - v_{2,2}^{t1}}{r_{2,1}^{t1}} = \frac{v_{2,2}^{t1} - v_{2,3}^{t1}}{r_{2,2}^{t1}} + \frac{v_{2,2}^{t1} - v_{2,2}^{b1}}{r_{2,2}} \\
 i_{2,2}^{t1} &= i_{2,3}^{t1} + i_{2,3} \rightarrow \frac{v_{2,2}^{t1} - v_{2,3}^{t1}}{r_{2,2}^{t1}} = \frac{v_{2,3}^{t1} - v_{2,4}^{t1}}{r_{2,3}^{t1}} + \frac{v_{2,3}^{t1} - v_{2,3}^{b1}}{r_{2,3}}
 \end{aligned}$$

$$\begin{aligned}
i_{0,1}^{bl} + i_{1,1} &= i_{1,1}^{bl} \rightarrow \frac{v_{0,1}^{bl} - v_{1,1}^{bl}}{r_{0,1}^{bl}} + \frac{v_{1,1}^{tl} - v_{1,1}^{bl}}{r_{1,1}} = \frac{v_{1,1}^{bl} - v_{2,1}^{bl}}{r_{1,1}^{bl}} \\
i_{0,2}^{bl} + i_{1,2} &= i_{1,2}^{bl} \rightarrow \frac{v_{0,2}^{bl} - v_{1,2}^{bl}}{r_{0,2}^{bl}} + \frac{v_{1,2}^{tl} - v_{1,2}^{bl}}{r_{1,2}} = \frac{v_{1,2}^{bl} - v_{2,2}^{bl}}{r_{1,2}^{bl}} \\
i_{0,3}^{bl} + i_{1,3} &= i_{1,3}^{bl} \rightarrow \frac{v_{0,3}^{bl} - v_{1,3}^{bl}}{r_{0,3}^{bl}} + \frac{v_{1,3}^{tl} - v_{1,3}^{bl}}{r_{1,3}} = \frac{v_{1,3}^{bl} - v_{2,3}^{bl}}{r_{1,3}^{bl}} \\
i_{1,1}^{bl} + i_{2,1} &= i_{2,1}^{bl} \rightarrow \frac{v_{1,1}^{bl} - v_{2,1}^{bl}}{r_{1,1}^{bl}} + \frac{v_{2,1}^{tl} - v_{2,1}^{bl}}{r_{2,1}} = \frac{v_{2,1}^{bl} - v_{3,1}^{bl}}{r_{2,1}^{bl}} \\
i_{1,2}^{bl} + i_{2,2} &= i_{2,2}^{bl} \rightarrow \frac{v_{1,2}^{bl} - v_{2,2}^{bl}}{r_{1,2}^{bl}} + \frac{v_{2,2}^{tl} - v_{2,2}^{bl}}{r_{2,2}} = \frac{v_{2,2}^{bl} - v_{3,2}^{bl}}{r_{2,2}^{bl}} \\
i_{1,3}^{bl} + i_{2,3} &= i_{2,3}^{bl} \rightarrow \frac{v_{1,3}^{bl} - v_{2,3}^{bl}}{r_{1,3}^{bl}} + \frac{v_{2,3}^{tl} - v_{2,3}^{bl}}{r_{2,3}} = \frac{v_{2,3}^{bl} - v_{3,3}^{bl}}{r_{2,3}^{bl}}
\end{aligned}$$

From which we solve for the variables we use as inputs and outputs (marked in blue), those at the edge of the circuit, and then gather conductances together and separate the voltages, leaving us with a matrix equation such as:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \mathbf{v}^{\text{ext}} = \begin{bmatrix} \vdots \\ \text{Part of } \mathbf{v}^{\text{ext}} \\ \underbrace{-g_{i,j-1}^{tl} \overbrace{v_{i,j-1}^{tl}} + (g_{i,j-1}^{tl} + g_{i,j}^{tl} + g_{i,j+1}^{tl}) \overbrace{v_{i,j}^{tl}} - g_{i,j+1}^{tl} \overbrace{v_{i,j+1}^{tl}}}_{\text{Part of A}} \underbrace{-g_{i,j}^{bl} \overbrace{v_{i,j}^{bl}}}_{\text{Part of B}} \\ \vdots \\ \underbrace{g_{i,j}^{bl} \overbrace{v_{i,j}^{bl}}}_{\text{Part of C}} + \underbrace{g_{i-1,j}^{bl} \overbrace{v_{i-1,j}^{bl}} - (g_{i-1,j}^{bl} + g_{i,j}^{bl} + g_{i+1,j}^{bl}) \overbrace{v_{i,j}^{bl}} + g_{i+1,j}^{bl} \overbrace{v_{i+1,j}^{bl}}}_{\text{Part of D}} \\ \vdots \end{bmatrix} = \begin{bmatrix} v_{1,0}^{tl} \cdot g_{1,0}^{tl} \\ 0 \\ v_{1,3}^{tl} \cdot g_{1,4}^{tl} \\ v_{2,0}^{tl} \cdot g_{2,0}^{tl} \\ 0 \\ v_{2,3}^{tl} \cdot g_{2,4}^{tl} \\ v_{0,1}^{bl} \cdot g_{0,1}^{bl} \\ v_{0,2}^{bl} \cdot g_{0,2}^{bl} \\ v_{0,3}^{bl} \cdot g_{0,3}^{bl} \\ v_{2,1}^{bl} \cdot g_{2,1}^{bl} \\ v_{2,2}^{bl} \cdot g_{2,2}^{bl} \\ v_{2,3}^{bl} \cdot g_{2,3}^{bl} \end{bmatrix} \quad (1)$$

Note that in this example $v_{i,0}^{tl}$ are considered the inputs, and $v_{i,4}^{tl} = v_{0,j}^{bl} = v_{3,j}^{bl} = 0$. Similarly, for the branches that are not present, we can consider infinite resistance ($r^{-1} = g = 0$), and therefore $g_{i,3}^{tl} = g_{0,j}^{bl} = 0$, which results in a sparser vector. This is generally the case when implementing a MCA, but to mitigate the attenuation of the signal due to wire resistance, $v_{i,4}^{tl}$ might also act as an input.

From (I) we extract \mathbf{G}_{ABCD}

$$\mathbf{G}_{ABCD} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \left[\begin{array}{cccc|cccc} 21.1 & -10 & & & -0.1 & & & \\ & -10 & 20.2 & -10 & & -0.2 & & \\ & & -10 & 20.3 & & & -0.3 & \\ & & & 41.4 & -20 & & & -0.4 \\ & & & -20 & 40.5 & -20 & & -0.5 \\ & & & & -20 & 40.6 & & -0.6 \\ -0.1 & & & & 60.1 & & -30 & \\ & -0.2 & & & & 80.2 & & -40 \\ & & -0.3 & & & & 100.3 & -50 \\ & & & -0.4 & -30 & & 62.4 & \\ & & & & -40 & & & 82.5 \\ & & & & & -50 & & 102.6 \end{array} \right]$$

This matrix can be scaled so that the all wire resistances equal -1 during computation, which therefore do not need to be stored, and do not contribute to floating point errors during multiplication.

More generally, the matrix can be constructed as:

$$\mathbf{G}_{ABCD} = \left[\begin{array}{cccc|cccc} \mathbf{A}_1 & 0 & \cdots & & 0 & \mathbf{B}_1 & 0 & \cdots & & 0 \\ 0 & \mathbf{A}_2 & & & & 0 & \mathbf{B}_2 & & & \\ \vdots & & \ddots & & \vdots & \vdots & & \ddots & & \vdots \\ & & & \mathbf{A}_i & & & & & \mathbf{B}_i & \\ & & & & \ddots & & & & & \ddots & 0 \\ 0 & & \cdots & 0 & \mathbf{A}_m & 0 & & \cdots & 0 & \mathbf{B}_m \\ \hline \mathbf{C}_1 & 0 & \cdots & & 0 & \mathbf{D}_1 & -\mathbf{G}_1^D & 0 & \cdots & 0 \\ 0 & \mathbf{C}_2 & & & & -\mathbf{G}_1^D & \mathbf{D}_2 & -\mathbf{G}_2^D & & \vdots \\ \vdots & & \ddots & & \vdots & 0 & & \ddots & & \\ & & & \mathbf{C}_i & & & & -\mathbf{G}_{i-1}^D & \mathbf{D}_i & -\mathbf{G}_i^D & 0 \\ & & & & \ddots & \vdots & & & & \ddots & \\ 0 & & \cdots & 0 & \mathbf{C}_m & 0 & \cdots & & 0 & -\mathbf{G}_{m-1}^D & \mathbf{D}_m \end{array} \right]$$

Where $\mathbf{G}_{ABCD} \in \mathbb{R}^{2mn \times 2mn}$, $(\mathbf{A}, \mathbf{D}) \in \mathbb{R}^{mn \times mn}$, $(\mathbf{B}, \mathbf{C}) \in \mathbb{R}_{(-)}^{mn \times mn}$, and $\mathbf{G}_i^D \in \mathbb{R}_{(+)}^{n \times n}$.

$$\mathbf{A}_i = \begin{bmatrix} g_{i,0}^{\text{t}1} + g_{i,1} + g_{i,1}^{\text{t}1} & -g_{i,1}^{\text{t}1} & & & & \\ -g_{i,1}^{\text{t}1} & g_{i,1}^{\text{t}1} + g_{i,2} + g_{i,2}^{\text{t}1} & \ddots & & & \\ & -g_{i,2}^{\text{t}1} & \ddots & -g_{i,(j-1)}^{\text{t}1} & & \\ & & \ddots & g_{i,(j-1)}^{\text{t}1} + g_{i,j} + g_{i,j}^{\text{t}1} & \ddots & \\ & & & -g_{i,j}^{\text{t}1} & \ddots & -g_{i,(n-1)}^{\text{t}1} \\ & & & & \ddots & g_{i,(n-1)}^{\text{t}1} + g_{i,n} + g_{i,n}^{\text{t}1} \end{bmatrix}$$

$$\mathbf{B}_i = \mathbf{C}_i^\top = -\text{Diag}(\mathbf{g}_{i^t}) = \begin{bmatrix} -g_{i,1} & & \\ & \ddots & \\ & & -g_{i,n} \end{bmatrix}$$

$$\mathbf{D}_i = \mathbf{G}_{i-1}^D + \text{Diag}(\mathbf{g}_{i^t}) + \mathbf{G}_i^D$$

Where $\mathbf{A}_i \in \mathbb{R}^{n \times n}$, $(\mathbf{B}_i, \mathbf{C}_i) \in \mathbb{R}_{(-)}^{n \times n}$, and $\mathbf{D}_i \in \mathbb{R}_{(+)}^{n \times n}$.

II. APPROXIMATION OF $\rho(\mathbf{G}_{ABCD})$

In [1], the upper left quadrant of \mathbf{M} is a block diagonal, the lower right quadrant can be multiplied by permutation matrices to be such as well, but with sides of size $n \times n$ instead. Since the largest eigenvalue of a block diagonal is the largest value between each of the block, we will only demonstrate the solution for one block $\mathbf{M}^{(i)}$.

$$\frac{\mathbf{M}^{(i)}}{g_{i,i}^{\dagger 1}} = \mathbf{T}'_m(-1, 2 + \frac{\alpha}{g_{i,i}^{\dagger 1}}, -1, \mathbf{S}) \quad (2)$$

$$\text{where } \mathbf{S} = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{g^{\mathbb{W}}}{g_{i,i}^{\dagger 1}} & \\ & \frac{g^{\mathbb{E}}}{g_{i,i}^{\dagger 1}} \end{bmatrix} \begin{bmatrix} 1 & \cdots & 0 \\ 0 & \cdots & 1 \end{bmatrix} \quad (3)$$

If we want to know the complexity of solving this, we need to approximate the amount of iterations we need to reach an acceptable error. We will describe the error as the 2-norm ($\|\cdot\|_2$) in this paper, but in code we use the Frobenius norm since it is less computationally taxing and $\|\mathbf{A}\|_2 \leq \|\mathbf{A}\|_F \leq \sqrt{\text{rank}(\mathbf{A})}\|\mathbf{A}\|_2$.

The eigenvalues, according to [2], of a matrix of a similar form to (2) are

$$\begin{aligned} \mathbf{T}'_m(c, b, a, \mathbf{0}) &= \mathbf{T}'_V \mathbf{T}'_D \mathbf{T}'_V^{-1} \\ t'_{D(k,k)} &= b + 2\sqrt{a}\sqrt{c} \cos \frac{(k-1)\pi}{m} \quad \forall k \in [1, m] \end{aligned}$$

And those of a toeplitz matrix are

$$\begin{aligned} \mathbf{T}_m(c, b, a) &= \mathbf{T}_V \mathbf{T}_D \mathbf{T}_V^{-1} \\ t_{D(k,k)} &= b + 2\sqrt{a}\sqrt{c} \cos \frac{k\pi}{m+1} \quad \forall k \in [1, m] \end{aligned}$$

By definition an induced, and therefore consistent, matrix norm is sub-multiplicative ($\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\|$) and sub-additive ($\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$).

$\mathbf{T}_m(-1, 2, -1)$, $\mathbf{T}'_m(-1, 2, -1, \mathbf{0})$ and \mathbf{S} are all (semi-)positive definite matrices. And in the case where $g^{\mathbb{W}} = g^{\mathbb{E}} = 1$

$$\frac{\alpha}{g_{i,i}^{\dagger 1}} \mathcal{I} + \mathbf{T}'_m(-1, 2, -1, g_{i,i}^{\dagger 1} \mathbf{S}) = \mathbf{T}_m(-1, 2 + \frac{\alpha}{g_{i,i}^{\dagger 1}}, -1)$$

Which is a convex operation, and therefore

$$\begin{aligned} \rho(\mathbf{M}^{(i)})^{-1} &= \frac{1}{g_{i,i}^{\dagger 1} \min_k |\text{eig}(\mathbf{T}'_m)|} \\ &\geq \frac{1}{\alpha + g_{i,i}^{\dagger 1} |t_{D(m,m)} - t'_{D(m,m)}|} \\ &= \frac{1}{\alpha + 2g_{i,i}^{\dagger 1} \left| \cos \frac{\pi}{m+1} - \cos \frac{\pi}{m} \right|} \end{aligned}$$

With big m we can use the small angle approximation of the cosine

$$\begin{aligned} \cos \frac{\pi}{m+1} - \cos \frac{\pi}{m} &\approx 1 - \frac{\pi^2}{2(m+1)^2} - \left(1 - \frac{\pi^2}{2m^2}\right) \\ &= \frac{\pi^2}{2m^2} - \frac{\pi^2}{2(m+1)^2} = \frac{\pi^2}{4} \frac{2m+1}{m^2(m^2+2m+1)} \end{aligned}$$

The \mathbf{N} on the other hand can be decomposed as:

$$\begin{aligned} \mathbf{N} &= \begin{bmatrix} \mathcal{I} & -\mathbf{BC}^{-1} \\ \mathcal{I} & \mathcal{I} \end{bmatrix} \begin{bmatrix} \alpha \mathcal{I} & \\ & \alpha \mathcal{I} + \mathbf{B} + \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathcal{I} & -\mathbf{BC}^{-1} \\ \mathcal{I} & \mathcal{I} \end{bmatrix}^{-1} \\ \rho(\mathbf{N}) &= \alpha \end{aligned}$$

Therefore,

$$\rho(\mathbf{M}^{-1} \mathbf{N}) = \|\mathbf{M}^{-1} \mathbf{N}\|_2 \leq \|\mathbf{M}^{-1}\|_2 \|\mathbf{N}\|_2 \leq \frac{\alpha}{\alpha + O\left(\frac{\pi^2 g_{i,i}^{\dagger 1}}{m^3}\right)}$$

which is a tighter bound of the spectral radius that provided in [1].

REFERENCES

- [1] E. D. Machado, J. L. Vicario, E. Miranda, and A. Morell, "Memristor crossbar array simulation for deep learning applications," *IEEE Transactions on Nanotechnology*, [Submitted] pending publication.
- [2] S. Kouachi, "Eigenvalues and eigenvectors of tridiagonal matrices," *The Electronic Journal of Linear Algebra*, vol. 15, Jan. 2006. [Online]. Available: <https://doi.org/10.13001/1081-3810.1223>