Memristor Crossbar Array Simulation for Deep Learning Applications

Supplemental material

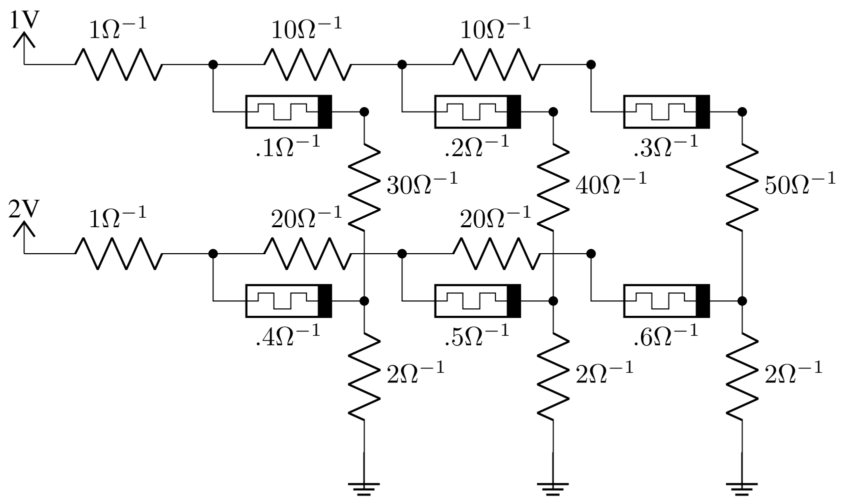

Fig. 1. Memristor Crossbar Array (2 rows, 3 columns)

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 the main paper. From Fig. 1 we derive the following equations:

$$i^{\mathtt{tl}}_{1,0} = i^{\mathtt{tl}}_{1,1} + i_{1,1} \rightarrow \frac{v^{\mathtt{tl}}_{1,0} - v^{\mathtt{tl}}_{1,1}}{r^{\mathtt{tl}}_{1,0}} = \frac{v^{\mathtt{tl}}_{1,1} - v^{\mathtt{tl}}_{1,2}}{r^{\mathtt{tl}}_{1,1}} + \frac{v^{\mathtt{tl}}_{1,1} - v^{\mathtt{bl}}_{1,1}}{r_{1,1}}$$

$$i^{\mathtt{tl}}_{1,1} = i^{\mathtt{tl}}_{1,2} + i_{1,2} \rightarrow \frac{v^{\mathtt{tl}}_{1,1} - v^{\mathtt{tl}}_{1,2}}{r^{\mathtt{tl}}_{1,1}} = \frac{v^{\mathtt{tl}}_{1,2} - v^{\mathtt{tl}}_{1,3}}{r^{\mathtt{tl}}_{1,2}} + \frac{v^{\mathtt{tl}}_{1,2} - v^{\mathtt{bl}}_{1,2}}{r_{1,2}}$$

$$i^{\mathtt{tl}}_{1,2} = i^{\mathtt{tl}}_{1,3} + i_{1,3} \rightarrow \frac{v^{\mathtt{tl}}_{1,2} - v^{\mathtt{tl}}_{1,3}}{r^{\mathtt{tl}}_{1,2}} = \frac{v^{\mathtt{tl}}_{1,3} - v^{\mathtt{tl}}_{1,4}}{r^{\mathtt{tl}}_{1,3}} + \frac{v^{\mathtt{tl}}_{1,3} - v^{\mathtt{bl}}_{1,3}}{r_{1,3}}$$

$$i^{\mathtt{tl}}_{2,0} = i^{\mathtt{tl}}_{2,1} + i_{2,1} \rightarrow \frac{v^{\mathtt{tl}}_{2,0} - v^{\mathtt{tl}}_{2,1}}{r^{\mathtt{tl}}_{2,0}} = \frac{v^{\mathtt{tl}}_{2,1} - v^{\mathtt{tl}}_{2,2}}{r^{\mathtt{tl}}_{2,1}} + \frac{v^{\mathtt{tl}}_{2,1} - v^{\mathtt{bl}}_{2,1}}{r_{2,1}}$$

$$i^{\mathtt{tl}}_{2,1} = i^{\mathtt{tl}}_{2,2} + i_{2,2} \rightarrow \frac{v^{\mathtt{tl}}_{2,1} - v^{\mathtt{tl}}_{2,2}}{r^{\mathtt{tl}}_{2,1}} = \frac{v^{\mathtt{tl}}_{2,2} - v^{\mathtt{tl}}_{2,3}}{r^{\mathtt{tl}}_{2,2}} + \frac{v^{\mathtt{tl}}_{2,2} - v^{\mathtt{bl}}_{2,2}}{r_{2,2}}$$

$$i^{\mathtt{tl}}_{2,2} = i^{\mathtt{tl}}_{2,3} + i_{2,3} \rightarrow \frac{v^{\mathtt{tl}}_{2,2} - v^{\mathtt{tl}}_{2,3}}{r^{\mathtt{tl}}_{2,2}} = \frac{v^{\mathtt{tl}}_{2,3} - v^{\mathtt{tl}}_{2,4}}{r^{\mathtt{tl}}_{2,3}} + \frac{v^{\mathtt{tl}}_{2,3} - v^{\mathtt{bl}}_{2,3}}{r_{2,3}}$$

$$i^{\texttt{bl}}_{0,1} + i_{1,1} = i^{\texttt{bl}}_{1,1} \rightarrow \frac{v^{\texttt{bl}}_{0,1} - v^{\texttt{bl}}_{1,1}}{r^{\texttt{bl}}_{0,1}} + \frac{v^{\texttt{tl}}_{1,1} - v^{\texttt{bl}}_{1,1}}{r_{1,1}} = \frac{v^{\texttt{bl}}_{1,1} - v^{\texttt{bl}}_{2,1}}{r^{\texttt{bl}}_{1,1}}$$

$$i^{\texttt{bl}}_{0,2} + i_{1,2} = i^{\texttt{bl}}_{1,2} \rightarrow \frac{v^{\texttt{bl}}_{0,2} - v^{\texttt{bl}}_{1,2}}{r^{\texttt{bl}}_{0,2}} + \frac{v^{\texttt{tl}}_{1,2} - v^{\texttt{bl}}_{1,2}}{r_{1,2}} = \frac{v^{\texttt{bl}}_{1,2} - v^{\texttt{bl}}_{2,2}}{r^{\texttt{bl}}_{1,2}}$$

$$i^{\texttt{bl}}_{0,3} + i_{1,3} = i^{\texttt{bl}}_{1,3} \rightarrow \frac{v^{\texttt{bl}}_{0,3} - v^{\texttt{bl}}_{1,3}}{r^{\texttt{bl}}_{0,3}} + \frac{v^{\texttt{tl}}_{1,3} - v^{\texttt{bl}}_{1,3}}{r_{1,3}} = \frac{v^{\texttt{bl}}_{1,3} - v^{\texttt{bl}}_{2,3}}{r^{\texttt{bl}}_{1,3}}$$

$$i^{\texttt{bl}}_{1,1} + i_{2,1} = i^{\texttt{bl}}_{2,1} \rightarrow \frac{v^{\texttt{bl}}_{1,1} - v^{\texttt{bl}}_{2,1}}{r^{\texttt{bl}}_{1,1}} + \frac{v^{\texttt{tl}}_{2,1} - v^{\texttt{bl}}_{2,1}}{r_{2,1}} = \frac{v^{\texttt{bl}}_{2,1} - v^{\texttt{bl}}_{3,1}}{r^{\texttt{bl}}_{2,1}}$$

$$i^{\texttt{bl}}_{1,2} + i_{2,2} = i^{\texttt{bl}}_{2,2} \rightarrow \frac{v^{\texttt{bl}}_{1,2} - v^{\texttt{bl}}_{2,2}}{r^{\texttt{bl}}_{1,2}} + \frac{v^{\texttt{tl}}_{2,2} - v^{\texttt{bl}}_{2,2}}{r_{2,2}} = \frac{v^{\texttt{bl}}_{2,2} - v^{\texttt{bl}}_{3,2}}{r^{\texttt{bl}}_{2,2}}$$

$$i^{\texttt{bl}}_{1,3} + i_{2,3} = i^{\texttt{bl}}_{2,3} \rightarrow \frac{v^{\texttt{bl}}_{1,3} - v^{\texttt{bl}}_{2,3}}{r^{\texttt{bl}}_{1,3}} + \frac{v^{\texttt{tl}}_{2,3} - v^{\texttt{bl}}_{2,3}}{r_{2,3}} = \frac{v^{\texttt{bl}}_{2,3} - v^{\texttt{bl}}_{3,3}}{r^{\texttt{bl}}_{2,3}}$$

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:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \mathbf{v}^{\text{ext}} = \begin{bmatrix} \vdots \\ \underbrace{\underbrace{-g^{\texttt{tl}}_{i,j-1}} \overbrace{v^{\texttt{tl}}_{i,j-1}} + \underbrace{(g^{\texttt{tl}}_{i,j-1} + g_{i,j} + g^{\texttt{tl}}_{i,j+1})} \overbrace{v^{\texttt{tl}}_{i,j}} \underbrace{-g^{\texttt{tl}}_{i,j+1}} \overbrace{v^{\texttt{tl}}_{i,j+1}}}_{\text{Part of } \mathbf{A}} \quad \underbrace{\underbrace{-g_{i,j}} \overbrace{v^{\texttt{bl}}_{i,j}}}_{\text{Part of } \mathbf{B}} \\ \text{Part of } \mathbf{v}^{\text{ext}} \\ \vdots \\ \text{Part of } \mathbf{v}^{\text{ext}} \\ \underbrace{\underbrace{g_{i,j}} \overbrace{v^{\texttt{tl}}_{i,j}}}_{\text{Part of } \mathbf{C}} + \underbrace{\underbrace{g^{\texttt{bl}}_{i-1,j}} \overbrace{v^{\texttt{bl}}_{i-1,j}} \underbrace{-(g^{\texttt{bl}}_{i-1,j} + g_{i,j} + g^{\texttt{bl}}_{i+1,j})} \overbrace{v^{\texttt{bl}}_{i,j}} + \underbrace{g^{\texttt{bl}}_{i+1,j}} \overbrace{v^{\texttt{bl}}_{i+1,j}}}_{\text{Part of } \mathbf{D}} \\ \vdots \end{bmatrix} = \left[\begin{array}{c} v^{\texttt{tl}}_{1,0} \cdot g^{\texttt{tl}}_{1,0} \\ 0 \\ v^{\texttt{tl}}_{1,3} \cdot g^{\texttt{tl}}_{1,4} \\ v^{\texttt{tl}}_{2,0} \cdot g^{\texttt{tl}}_{2,0} \\ 0 \\ v^{\texttt{tl}}_{2,3} \cdot g^{\texttt{tl}}_{2,4} \\ \hline v^{\texttt{bl}}_{0,1} \cdot g^{\texttt{bl}}_{0,1} \\ v^{\texttt{bl}}_{0,2} \cdot g^{\texttt{bl}}_{0,2} \\ v^{\texttt{bl}}_{0,3} \cdot g^{\texttt{bl}}_{0,3} \\ v^{\texttt{bl}}_{2,1} \cdot g^{\texttt{bl}}_{2,1} \\ v^{\texttt{bl}}_{2,2} \cdot g^{\texttt{bl}}_{2,2} \\ v^{\texttt{bl}}_{2,3} \cdot g^{\texttt{bl}}_{2,3} \end{array}\right] \tag{1}$$

Note that in this example $v^{\texttt{tl}}_{i,0}$ are considered the inputs, and $v^{\texttt{tl}}_{i,4} = v^{\texttt{bl}}_{0,j} = v^{\texttt{bl}}_{3,j} = 0$. Similarly, for the branches that are not present, we can consider infinite resistance ($r^{-1} = g = 0$), and therefore $g^{\texttt{tl}}_{i,3} = g^{\texttt{bl}}_{0,j} = 0$, 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, $v^{\texttt{tl}}_{i,4}$ might also act as an input.

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

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

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:

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

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 $\mathbf{G}_i^D \in \mathbb{R}^{n \times n}_{(+)}$.

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

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

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

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 the main paper, 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 $\mathbf{M}^{(i)}$.

$$\frac{\mathbf{M}^{(i)}}{g^{\mathtt{tl}}_{i,i}} = \mathbf{T'}_m(-1, 2 + \frac{\alpha}{g^{\mathtt{tl}}_{i,i}}, -1, \mathbf{S}) \tag{2}$$

$$\text{where } \mathbf{S} = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{g^{\mathtt{W}}}{g^{\mathtt{tl}}_{i,i}} & \\ & \frac{g^{\mathtt{E}}}{g^{\mathtt{tl}}_{i,i}} \end{bmatrix} \begin{bmatrix} 1 & \cdots & 0 \\ 0 & \cdots & 1 \end{bmatrix} \tag{3}$$

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 $||\mathbf{A}||_2 \leq ||\mathbf{A}||_F \leq \sqrt{\mathtt{rank}(\mathbf{A})}||\mathbf{A}||_2$.

The eigenvalues, according to [1], of a matrix of a similar form to (2) are

$$\mathbf{T'}_m(c, b, a, \mathbf{0}) = \mathbf{T'}_V \mathbf{T'}_D \mathbf{T'}_V^{-1}$$

$$t'_{D(k,k)} = b + 2\sqrt{a}\sqrt{c}\cos\frac{(k-1)\pi}{m} \quad \forall k \in [1, m]$$

And those of a toeplitz matrix are

$$\mathbf{T}_m(c, b, a) = \mathbf{T}_V \mathbf{T}_D \mathbf{T}_V^{-1}$$

$$t_{D(k,k)} = b + 2\sqrt{a}\sqrt{c}\cos\frac{k\pi}{m+1} \quad \forall k \in [1, m]$$

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 $g^{\mathtt{W}} = g^{\mathtt{E}} = 1$

$$\frac{\alpha}{g^{\mathtt{tl}}_{i,i}}\mathcal{I} + \mathbf{T'}_m(-1, 2, -1, g^{\mathtt{tl}}_{i,i}\mathbf{S}) = \mathbf{T}_m(-1, 2 + \frac{\alpha}{g^{\mathtt{tl}}_{i,i}}, -1)$$

Which is a convex operation, and therefore

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

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

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

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

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

$$\rho(\mathbf{N}) = \alpha$$

Therefore,

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

III. Backpropagation with $\mathbf{G}_{ABCD}$

In a regular fully connected layer we train the weights by modifying them using the gradient calculated from the loss function, for example in a classification task the categorical cross-entropy between the output ($\hat{\mathbf{y}}$) of the neural network through the softmax function ($\sigma(\hat{\mathbf{y}})$ [probability vector]) and the ground truth ($\mathbf{y}$ [one-hot vector]).

$$\ell = \texttt{loss}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i \neq j} y_i \log_e(\sigma(\hat{y}_i)) \qquad \sigma(\hat{y}_i) = P(C = i|\hat{y}_i) = \frac{e^{\hat{y}_i}}{\sum_{k \neq i}^{C} e^{\hat{y}_k}}$$

$$\frac{\partial \sigma(\hat{y}_j)}{\partial \hat{y}_i} = \begin{cases} \sigma(\hat{y}_j)[1 - \sigma(\hat{y}_i)] & i = j \\ \sigma(\hat{y}_j)\sigma(\hat{y}_i) & i \neq j \end{cases} \qquad \sum_{k}^{C} y_k = \sum_{k}^{C} \sigma(\hat{y}_k) = 1$$

We would then calculate the loss with respect to our output, via the chain rule (a.k.a backpropagation):

$$\frac{\partial \ell}{\partial \hat{\mathbf{y}}} = \frac{\partial \ell}{\partial \sigma(\hat{\mathbf{y}})} \frac{\partial \sigma(\hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}} = -\sum_{k}^{C} y_k \frac{\partial \log_e(\sigma(\hat{y}_k))}{\partial \sigma(\hat{y}_k)} \frac{\partial \sigma(\hat{y}_k)}{\partial \hat{y}_i} =$$
$$= y_i[1 - \sigma(\hat{y}_i)] - \sum_{k \neq i}^{C} y_k \frac{1}{\sigma(\hat{y}_k)} (-\sigma(\hat{y}_k)\sigma(\hat{y}_i)) = \sigma(\hat{y}_i) - y_i$$

And then continue applying the chain rule to get the gradient of other layers:

$$\frac{\partial \ell}{\partial \mathbf{x}} = \frac{\partial \ell}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{a}^{(k)}} \frac{\partial \mathbf{a}^{(k)}}{\partial \mathbf{a}^{(k-1)}} \cdots \frac{\partial \mathbf{a}^{(1)}}{\partial \mathbf{x}}$$

For each layer then we need to know the loss for the previous layer and how we should adjust the weights, that is $\frac{\partial \texttt{loss}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{G}^{(k)}}$, which for the case where $\mathbf{a}^{(k)} = \mathbf{G}^{(k)} \mathbf{a}^{(k-1)}$, the gradients would be:

$$\frac{\partial \ell}{\partial \mathbf{G}^{(k)}} = \frac{\partial \ell}{\partial \mathbf{a}^{(k)}} \frac{\partial \mathbf{a}^{(k)}}{\partial \mathbf{G}^{(k)}} = \frac{\partial \ell}{\partial \mathbf{a}^{(k)}} (\mathbf{a}^{(k-1)})^\top \qquad \frac{\partial \mathbf{a}^{(k)}}{\partial \mathbf{a}^{(k-1)}} = (\mathbf{G}^{(k)})^\top$$

Our case, however, where $\mathbf{v}^{\text{ext}} = \mathbf{G}_{ABCD}^{-1} \mathbf{i}^{\text{ext}}$, can be solved as follows, with the resulting backpropagation of the loss.

$$\frac{\partial \ell}{\partial \mathbf{i}^{\text{ext}}} = \mathbf{G}_{ABCD}^{-\top} \frac{\partial \ell}{\partial \mathbf{v}^{\text{ext}}}$$

$$\frac{\partial \ell}{\partial \mathbf{G}_{ABCD}} = -\left(\mathbf{G}_{ABCD}^{-\top} \frac{\partial \ell}{\partial \mathbf{v}^{\text{ext}}} \right) \left(\mathbf{G}_{ABCD}^{-1} \mathbf{i}^{\text{ext}} \right)^\top = -\frac{\partial \ell}{\partial \mathbf{i}^{\text{ext}}} (\mathbf{v}^{\text{ext}})^\top$$

Furthermore, $\frac{\partial \ell}{\partial \mathbf{v}^{\text{ext}}}$ is very sparse since we only know the n outputs at the edge of the crossbar array, as they are the only ones which we can compare, using the same reasoning as before, we only need to keep the rows of $\mathbf{G}_{ABCD}^{-1}$ that get multiplied by non-zero loss.

However, we are only interested in $\frac{\partial \ell}{\partial \mathbf{G}_{ABCD}}$ in so far as we can use it to compute $\frac{\partial \ell}{\partial \mathbf{G}}$. And while the former is dense, we only care about the element that depend on the latter, which are entirely located in the diagonals of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$.

$$\frac{\partial \ell}{\partial \mathbf{G}_{ABCD}} = \begin{bmatrix} \frac{\partial \ell}{\partial \mathbf{A}} & \frac{\partial \ell}{\partial \mathbf{B}} \\ \frac{\partial \ell}{\partial \mathbf{C}} & \frac{\partial \ell}{\partial \mathbf{D}} \end{bmatrix}$$

$$\frac{\partial \ell}{\partial \texttt{vec}(\mathbf{G})} = \texttt{diag} \left(\frac{\partial \ell}{\partial \mathbf{A}} + \frac{\partial \ell}{\partial \mathbf{B}} + \frac{\partial \ell}{\partial \mathbf{C}} + \frac{\partial \ell}{\partial \mathbf{D}} \right) \texttt{vec}(\mathbf{G})$$

References

[1] 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