

3D Crossbar Algorithms

1 Original

Input: $N \times M$ Crossbar (NN-chip), AM, IB, P_{th} and S_{th}
Output: Weighted NN-chip

```
1:  $i \leftarrow [0 \text{ to } M - 1]$ 
2:  $j \leftarrow [1 \text{ to } K]$ 
3:  $inp \leftarrow [1 \text{ to } p]$ 
4:  $AM_c \leftarrow \text{column of } AM$ 
5: for each  $i$  in IB do
6:    $C_i \leftarrow i^{th}$  column of NN-chip
7:   for each  $j$  in IB[ $i$ ] do
8:     for each  $inp$  in IB[ $i$ ][ $j$ ] do
9:        $AM \leftarrow \text{Weight\_Update}(AM, inp, S_{th}, AM_c)$ 
10:    end for
11:     $W_{AM} \leftarrow \text{weights of } AM$ 
12:    NN-chip  $\leftarrow \text{Weight\_Update}(\text{NN-chip}, W_{AM}, P_{th}, C_i)$ 
13:    RESET AM
14:  end for
15: end for
16: return Weighted NN-chip
```

2 Approach: Distribution Of Features

Following the 6th section of the previous paper, each one of the k layers of the 3D crossbar will be of $i \times M$ size such that $i \times k = N$. We can use an AM crossbar of dimensions $i \times 1$.

Input: $i \times k \times M$ Crossbar (NN-chip), AM Crossbar, IB, P_{th} and S_{th}
Output: Weighted NN-chip

```
1:  $inp \leftarrow [1 \text{ to } p]$ 
2:  $AM_c \leftarrow \text{column of } AM$ 
3: for each  $i$  in IB do
4:    $C_i \leftarrow i^{th}$  column of NN-chip
5:   for each  $j$  in IB[ $i$ ] do
6:     for  $b$  from 0 to  $k-1$  do
7:       for each  $inp$  in IB[ $i$ ][ $j$ ] do
8:          $AM \leftarrow \text{Weight\_Update}(AM, inp[i \times b : i \times (b+1)], S_{th}, AM_c)$ 
9:       end for
10:       $W_{AM} \leftarrow \text{weights of } AM$ 
11:      NN-chip  $\leftarrow \text{Weight\_Update}(\text{NN-chip}, W_{AM}, P_{th}, C_i)$ 
12:      RESET AM
13:    end for
14:  end for
15: end for
16: return Weighted NN-chip
```

3 Approach: Distribution Of Classes

We can do the same distribution with classes instead of features. each one of the k layers of the 3D Crossbar will be of $i \times N$ size such that $i \times k = M$. Here the dimensions of the AM Crossbar remain the same i.e. $N \times 1$.

Note: 'K' is number of

Input: $i \times k \times N$ Crossbar (NN-chip), AM Crossbar, IB, P_{th} and S_{th}

Output: Weighted NN-chip

```
1:  $j \leftarrow [1 \text{ to } K]$ 
2:  $AM_c \leftarrow \text{column of } AM$ 
3: for  $b$  from 0 to  $k-1$  do
4:   for  $a$  from 1 to  $i$  do
5:      $c \leftarrow b*i + a$ 
6:      $C_c \leftarrow c^{th}$  column of NN-chip
7:     for each  $j$  in IB[ $c$ ] do
8:       for each  $inp$  in IB[ $c$ ][ $j$ ] do
9:          $AM \leftarrow \text{Weight\_Update}(AM, inp, S_{th}, AM_c)$ 
10:      end for
11:       $W_{AM} \leftarrow \text{weights of } AM$ 
12:      NN-chip  $\leftarrow \text{Weight\_Update}(\text{NN-chip}, W_{AM}, P_{th}, C_c)$ 
13:      RESET AM
14:    end for
15:  end for
16: end for
17: return Weighted NN-chip
```
