3D Crossbar Algorithms

1 Original

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

$i \leftarrow [0\ to\ M-1]$
$j \leftarrow [1\ to\ K]$
$inp \leftarrow [1\ to\ p]$
$AM_c \leftarrow column\ of\ AM$
for `each i in IB` **do**
  $C_i \leftarrow i^{th}$ column of NN-chip
  for `each j in IB[i]` **do**
    for `each inp in IB[i][j]` **do**
      AM← Weight_Update(AM, inp, S_{th}, AM_c)
    end for
    W_{AM} ← weights of AM
    NN-chip← Weight_Update(NN-chip, W_{AM}, P_{th}, C_i)
    RESET AM
  end for
end for
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×M size such that i×k=N. We can use an AM crossbar of dimeansions i×1.

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

$inp \leftarrow [1\ to\ p]$
$AM_c \leftarrow column\ of\ AM$
for `each q in IB` **do**
  $C_q \leftarrow q^{th}$ column of NN-chip
  for `each j in IB[q]` **do**
    for `b from 0 to k-1` **do**
      for `each inp in IB[q][j]` **do**
        AM ← Weight_Update(AM, inp[i×b:i×(b+1)], S_{th}, AM_c)
      end for
      W_{AM} ← weights of AM
      NN-chip ← Weight_Update(NN-chip, W_{AM}, P_{th}, C_q)
      RESET AM
    end for
  end for
end for
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×N size such that i×k=M. Here the dimensions of the AM Crossbar remain the same i.e. N×1.
Note: 'K' is number of sub-batches and 'k' is the distribution factor.

Input: i×k×N Crossbar (NN-chip), AM Crossbar, IB, P_{th} and S_{th}
Output: Weighted NN-chip

1: $j \leftarrow [1\ to\ K]$
2: $AM_c \leftarrow column\ of\ AM$
3: **for** `b from 0 to k-1` **do**
4: **for** `a from 1 to i` **do**
5: c← 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← Weight_Update(AM, inp, S_{th}, AM_c)
10: **end for**
11: $W_{AM} \leftarrow$ weights of AM
12: NN-chip ← Weight_Update(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
