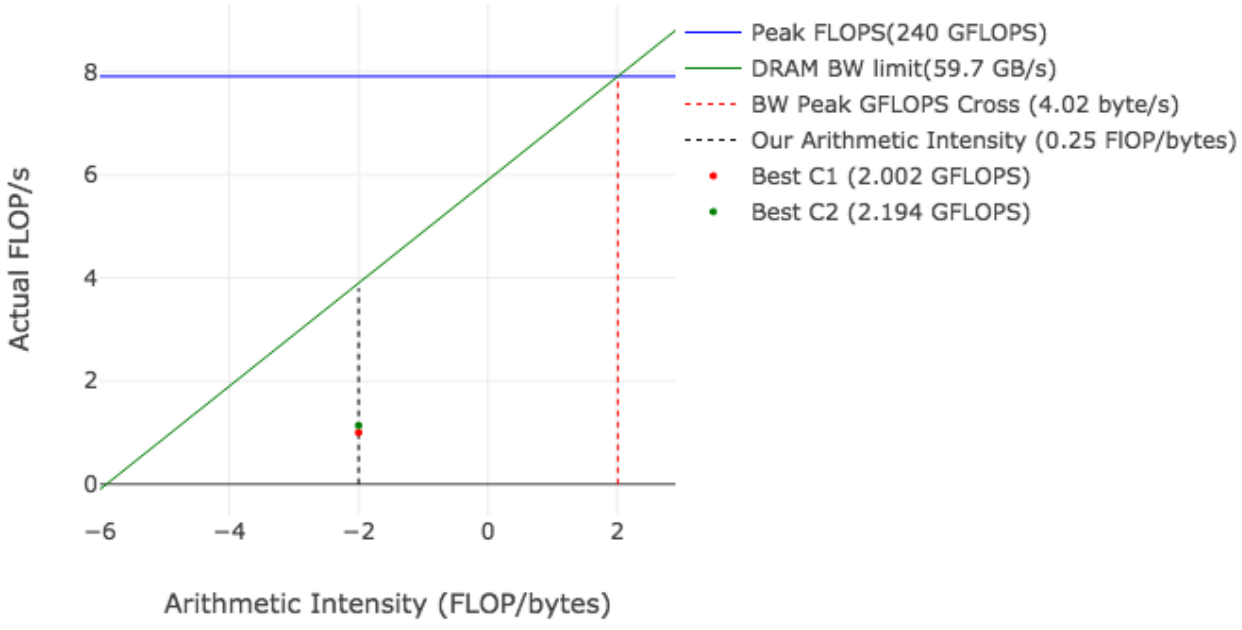
Q1. Why is the loop unrolling version faster?
Ans. The loop unrolling version is faster due to lesser amount of iteration overhead. The statements are also parallelized.

Q2. What can be done to obtain a higher memory BW using all the resources of the CPU?
Ans. Use all the cores for parallelism and divide the array into blocks of cache size and do all the operations that are to be done using that block that are to be done before switching.

Q3. Plot
Ans. Has been plotted in log2-log2 scale.

Parameter	In Normal units	In Log2 units
Peak CPU FLOPS	240 GFLOPS	8 GFLOPS
DRAM BW	59.7 GB/s	5.89 GB/s
BW Peak FLOPS cross	4.02 bytes/s	2 bytes/s
Our A.I line	0.25 FLOP/bytes	-2 FLOP/bytes
C1 Dot on A.I line	2.002 GFLOPS	1 GFLOPS
C2 Dot on A.I line	2.1947 GFLOPS	1.13 GFLOPS



Q4. What are the dimensions of Wl , xl , and zl for each layer l ?
Ans. x0 -> 256 * 256, W0 -> 65536 * 4000, z0 = x1 -> 1 * 4000 or 4000 * 1 depending on the implementation, W1 -> 4000 * 1000, z1 -> 1000 * 1or 1 * 1000 depending on the implementation.

Q5. Assuming double precision operations, what are Wl , x l, and z l sizes in memory for each layer l?
Ans. X0 -> 256 * 256 * 8 = 524,288 bytes
W0 -> 2,097,152,000 bytes
Z0 -> 32,000 bytes
W1 ->32,000,000 bytes
Z1 -> 8000 bytes

C1 output:

```
*C1*
Sum : 1333333332077485824.000000
Elapsed time : 2.003033 seconds
```

C2 output:

```
*C1*
Sum : 1333333332077485824.000000
Time : 1.997979 secs
Bw : 8.008092 GB/s
FLOPS: 2.002023 GFLOP/s
*C2*
Sum : 1333333332077485824.000000
Time : 1.822518 secs
Bw : 8.779061 GB/s
FLOPS: 2.194765 GFLOP/s
```

C3 output:

```
*C3*
Elapsed time: 159.56710474507418 secs
Checksum: 15235693999.999979
```

C4 output:

```
*C4*
Elapsed time: 0.11388553801225498 secs
Speedup: 1401.1182414390798 times
Checksum: 15235693999.999926
```

C5 output:

```
*C5*
Elapsed_time: 3.370286 secs
Speedup wrt C3: 47.345266 times
Sum is: 15235693999.999979
```

C6 output:

```
*C6*
Elapsed_time: 0.161201 secs
Speedup wrt C3: 989.863235 times
Sum is: 15235693999.999979
```