CSE141L Lab 3 Caching Optimizations Worksheet1

Name: ________________________ Student ID: ___________________

Instructions

- Complete this worksheet while reading/working through the lab write up. The worksheet doesn't make sense without the lab.
- The point values are listed for each question. Altering the size of the cells will cost you 1 point. The write up portion of the lab is 30% of your total point for the lab as shown in the lab's README.md

Cache and dataset characteristics

P1 (4pt) Find out the dimensions (number of data elements) of the following tensors/vectors used in `fc_layer_t::activate` for the cifar100 dataset and fill the following table

Tensor/Vector	Number of Data Elements
`in`	____
`out`	____
`weights`	____
`activation_input`	____

P2 (4pt) Calculate the size (in Bytes) of the following tensors/vectors used in `fc_layer_t::activate` for the cifar100 dataset and fill the following table

Tensor/Vector	Size in Bytes
`in`	____
`out`	____
`weights`	____
`activation_input`	____

P3 (4pt) How much of each of these data structures used in `fc_layer_t::activate()` will fit in the L1 and L2 cache (Note The cache size for our machine is L1-dCache: 32kb, L2: 256kb, L3: 8Mb)?

tensor	% that'll fit in L1	% that'll fit in L2
`in`	___________________	___________________
`out`	___________________	___________________
`weights`	___________________	___________________
`activation_input`	___________________	___________________

Understanding Tensor_t

Given `tensor_t<double> foo(tdsize(4,3,5,7))`, answer the following (double are 8 bytes)(Hint: Look at lecture slides) :

P1 (1pt) How many elements are there in `foo`?

P2 (1pt) What's the linear index of element (1,1,1,1)?

P3 (1pt) How far apart are elements that differ by 1 in each dimension?

dim.	distance in bytes	distance in linear index
x		
y		
z		
b		

Tier 1: Reordering and Tiling loops in `fc_layer_t::activate`

P1 (3pt) Fill out the following table. Report the Misses per Instruction by using the performance counters (there should be a column for "MPI" in the reported data when running with L1/2/3.cfg)

Cache-Level	Miss rate - Base	Miss rate - loop reordering	Miss rate - Tiling
L1			
L2			
L3			

P2 (4pt) Change the order of loops from `b i n` to `b n i` in `fc_layer_t::activate` and report the speedup.

Speedup after loop reordering : ______________

P3 (4pt) Block the loop `n` in `fc_layer_t::activate` with the tile sizes 1, 2, 4, 8, 16 and fill out the table below.

Dataset	Step size	Blocked implementation time	Speedup vs step size == 1
cifar100	1	____	_______
cifar100	2	____	_______
cifar100	4	____	_______
cifar100	8	____	_______
cifar100	16	____	_______

P4 (4pt) In a single line graph, plot the speed up against the different block sizes for blocking the loop `n` in `fc_layer_t::activate`. Block size is the independent vairable.

```
Your graph here
```

P5 (4pt) Consider the blocksize which gave maximum speedup in the previous question P4 and fill out the following table

1. Base implementation time : __________________________________
2. Implementation time of your optimized solution : __________________________________
3. Base implementation L1 miss rate : __________________________________
4. Your fastest solution L1 misse rate : __________________________________

P6 (3pt) Insert the memory access patterns (take screenshots from moneta) for loop orders b-i-n, b-n-i and nn-b-n-i. Do this for weights tensor and pass the runtime options that set scale to 4 and reps to 1 in config.env. The dataset should be cifar100 (which should be the default). Leave the cache lines and block size fields as they are but set the max accesses to 2 million. In the file opt_cnn.cpp, there is an example of where and what to put to run the moneta please take a look at it.

```
memory access pattern with loop order b-i-n
```

```
memory access pattern with loop order b-n-i
```

```
memory access pattern with loop order nn-b-n-i
```