CSE141L Lab 2 Characterizing a Perceptron

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. There are 75 points total for the write up portion of the lab.

Enabling the Profiler

P1 (1pt) Which function accounts for the most time?

A: _____

P2 (1pt) What percentage of time does it account for?

A: _____

P3 (1pt) According to Amdahl's Law, how much speedup could you possibly achieve by optimizing this function?

```
Your work here
```

Taking a Closer Look at the Code

P3 (1pt) We noticed that fc_layer_t::activate(tensor_t const&) took 46.7% of the total execution time. If we speed up this function by 3x, how much will the total program be sped up by?

```
Your work here
```

What's the Compiler Doing?

P1 (1pt) How many instructions does the function 'point_t::point_t(int, int, int)' execute when called?

A: _____

Looking at Performance Counters

P1 (4pt) Compute the instruction mix for each of the dataset and enter them below (as %)

Dataset	Memory insts	Branches	uncond. branches
mnist	_____	_____	_____
emnist	_____	_____	_____
cifar10	_____	_____	_____
cifar100	_____	_____	_____
imagenet	_____	_____	_____

P2 (4pt) Fill out the table (total data processed is the product of the model size and the number of training inputs)

Dataset	Model size (B)	training_inputs_count	total data processed	Memory ops
mnist	_____	_____	_____	_____
emnist	_____	_____	_____	_____
cifar10	_____	_____	_____	_____
cifar100	_____	_____	_____	_____
imagenet	_____	_____	_____	_____

P3 (4pt) Prepare a bar graph from your table that plots the number of memory operation/byte of data processed and the number of branches per byte of data processed for each workload.

```
Your graph here
```

Asking the Compiler to Do More

P1 (1pt) Look at the demangled, optimized version of code.s. Count the total number of instructions in the inner loop body of fc_layer_t::activate. How many are there?

A: _____

P2 (4pt) Compute the instruction mix for each of the dataset for the optimized code and enter them below (as %)

Dataset	Memory insts	Branches	uncond. branches
mnist	_____	_____	_____
emnist	_____	_____	_____
cifar10	_____	_____	_____
cifar100	_____	_____	_____
imagenet	_____	_____	_____

P3 (4pt) Fill out the table for the optimized code (total data processed is the product of the model size and the number of training inputs)

Dataset	Model size (B)	training_inputs_count	total data processed	Memory ops
mnist	_____	_____	_____	_____
emnist	_____	_____	_____	_____
cifar10	_____	_____	_____	_____
cifar100	_____	_____	_____	_____
imagenet	_____	_____	_____	_____

P4 (4pt) Prepare a bar graph from your table that plots the number of memory operation/byte of data processed and the number of branches per byte of data processed for each workload for the optimized code.

```
Your graph here
```

For the following questions, compute the answers based on the total number of instructions, cycles, etc. across all the workloads.

P5 (1pt) Based on the data in optimized-pe.csv, how much speedup from '-O3' do you expect due to change in IC?

A: _____

P6 (1pt) Based on the data in optimized-pe.csv, how much speedup from '-O3' do you expect due to change in CPI?

A: _____

P7 (1pt) Based on the data in optimized-pe.csv, how much speedup from '-O3' do expect from the combination of IC and CPI?

A: _____

P8 (4pt) Fill in the data below

Assembly Code	**Unoptimized**	**Optimized**
Instruction count	_____	_____
Cycle count	_____	_____
Cycle time	_____	_____
Projected execution time	_____	_____
Projected speedup vs unoptimized	_____	_____
Actual execution time	_____	_____
Actual speedup vs unoptimized	_____	_____

P9 (4pt) How accurately did the PE accurately model the performance of this program on these workloads?

A: _____

P10 (4pt) Based on profile data with -O3 turned on, which functions should you target for optimization.

A: _____

P11 (4pt) For the functions you listed, what's the largest speed up you could hope to achieve?

```
Your work here
```

Measuring Actual Performance

For the following questions, compute the answers based on the total number of instructions, cycles, etc. across all the workloads.

P1 (3pt) How much overhead (i.e., increase) does gprof cause in terms of the following?

IC: _____ CPI: _____ ET: _____

Reasoning About Performance

P1 (1pt) Which function accounts for the largest fraction of time in optimized gprof data?

A: _____

P2 (1pt) What's the O() complexity of that function?

A: _____

P3 (4pt) Fill out this table using data from your per-workload gprof outputs for your hot function.

dataset	measured ET	ET rel. to mnist Big-O estimate
mnist	_____	_____
emnist	_____	_____
cifar10	_____	_____
cifar100	_____	_____
imagenet	_____	_____

P4 (4pt) Draw a scatter plot with the relative values of m*n on the x-axis and relative execution time on the y-axis. Plot the data for measured ET and your O() estimate.

Your Graph here

P5 (1pt) How well does your O() match actual performance?

A: _____

Changing the Clock Rate and Measuring Power

P1 (4pt) Draw a line graph with clock speed on the x-axis and execution time on the y-axis.

```
Your Graph here
```

P2 (4pt) Draw a line graph with clock speed on the x-axis and energy on the y-axis.

```
Your Graph here
```

P3 (4pt) Draw a line graph with clock speed on the x-axis and power on the y-axis.

```
Your Graph here
```