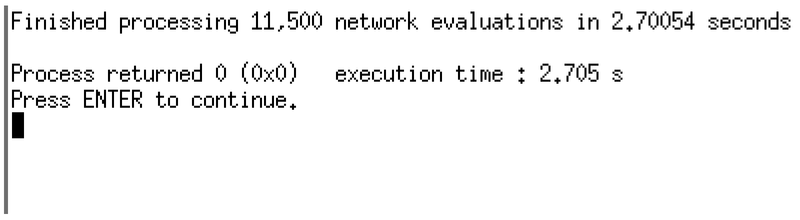
Bucky Frost
CS 405
Homework 1
2014-02-05

Timing Analysis

The Chrono library was used to preform the timing analysis of the neural network. The timing was generated by starting the timing before a for loop that calculated 11,500 iterations of processing the input of the neural network. It took approximately 2.70054 seconds for the 11,500 evaluations. This comes to about 4,258 network evaluations per second. This number of evaluations has no statistical significance as just one timing was conducted on the for loop with no other statistical analysis. The output from the timing in the main function is shown in Figure 1.



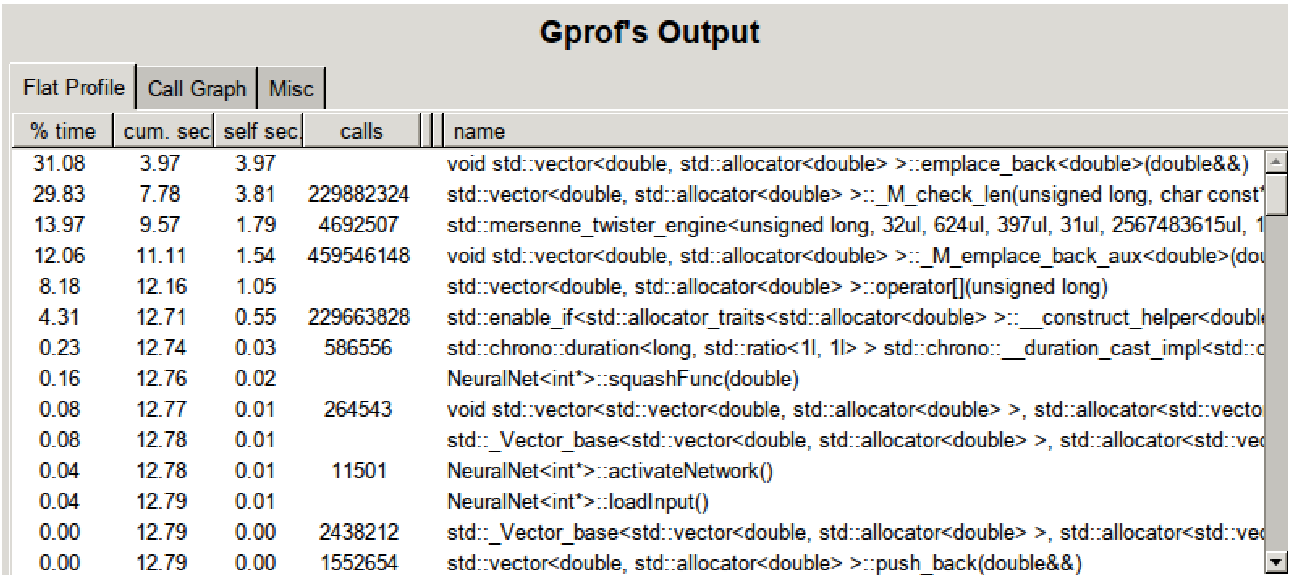
```
Finished processing 11,500 network evaluations in 2.70054 seconds

Process returned 0 (0x0)   execution time : 2.705 s
Press ENTER to continue.
```

Figure 1: Timing of Neural Network Evaluations

Profile

The program used to generate a profile of the neural network was Gprof. Figure 2 below shows the profile generated through analysis using Codeblocks. As shown, the majority of the time was involved in handling the overhead of the vector data structures. This could be improved by using another structure, such as std::array. Another reason for the vector manipulation being the bottleneck of the program is inefficient algorithm for computing the values through the layers of the neural network. The network implementation needs to be based on a new data structure and redo the functionality of calculating the weights through the network. The other functionality of the program that took a significant amount of processing time was the mersenne_twister_engine. Due to the lack of research, I can not say for sure if this is common or improper usage of the function.



Gprof's Output

Flat Profile | Call Graph | Misc

% time	cum. sec	self sec	calls	name
31.08	3.97	3.97		void std::vector<double, std::allocator<double> >::emplace_back<double>(double&&)
29.83	7.78	3.81	229882324	std::vector<double, std::allocator<double> >::_M_check_len(unsigned long, char const*
13.97	9.57	1.79	4692507	std::mersenne_twister_engine<unsigned long, 32ul, 624ul, 397ul, 31ul, 2567483615ul, 1
12.06	11.11	1.54	459546148	void std::vector<double, std::allocator<double> >::_M_emplace_back_aux<double>(dou
8.18	12.16	1.05		std::vector<double, std::allocator<double> >::operator[](unsigned long)
4.31	12.71	0.55	229663828	std::enable_if<std::allocator_traits<std::allocator<double> >::__construct_helper<doubl
0.23	12.74	0.03	586556	std::chrono::duration<long, std::ratio<1l, 1l> > std::chrono::__duration_cast_impl<std::c
0.16	12.76	0.02		NeuralNet<int*>::squashFunc(double)
0.08	12.77	0.01	264543	void std::vector<std::vector<double, std::allocator<double> >, std::allocator<std::vecto
0.08	12.78	0.01		std::_Vector_base<std::vector<double, std::allocator<double> >, std::allocator<std::vec
0.04	12.78	0.01	11501	NeuralNet<int*>::activateNetwork()
0.04	12.79	0.01		NeuralNet<int*>::loadInput()
0.00	12.79	0.00	2438212	std::_Vector_base<std::vector<double, std::allocator<double> >, std::allocator<std::vec
0.00	12.79	0.00	1552654	std::vector<double, std::allocator<double> >::push_back(double&&)

Figure 2: Gprof Profile of Neural Network