JI Xuanjin

16251180

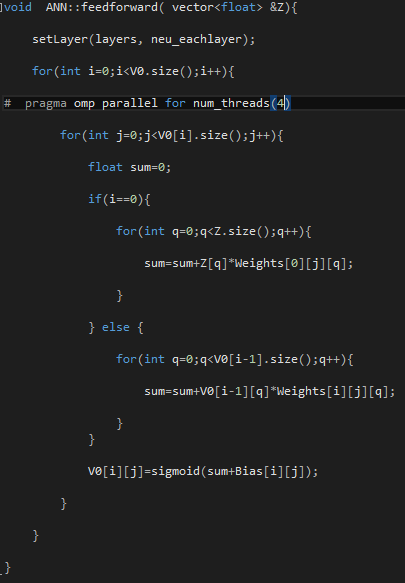
COMP3046\_Phase4

Enhancement of Artificial Neural Network by OpenMP

1. Design of OpenMP
   1. In **feedforward()** part, we need to calculate the value for each neuron by previous layer’s neurons’ values multiplied by current layer’s weights. It is the **matrix \* matrix** part. Here, we use

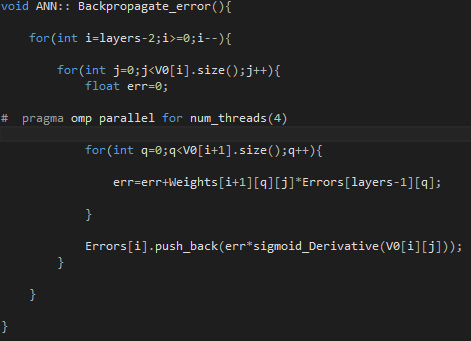
**# pragma omp parallel for num\_threads(4)** to create 4 threads to do the calculation .

Inside each layer, we use 4 threads to calculate the value for neuron



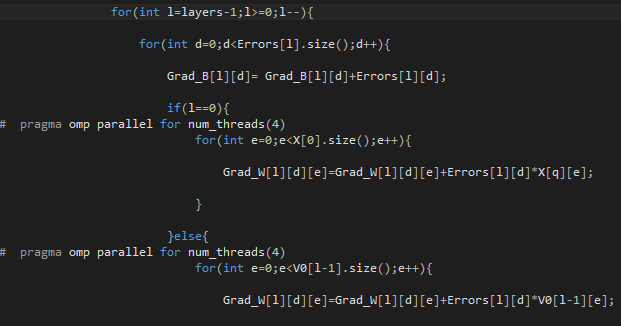
* 1. In **Backpropagate()** part, , we need to calculate the value for each neuron’s error current layer’s neurons’ error and (current+1) layer’s weights and error. It is the **matrix\* matrix** part. Here, we use

**# pragma omp parallel for num\_threads(4)** to create 4 threads to do the calculation



* 1. In **train()** part, , we need to calculate the gradients by (current-1) layer’s sigmoid value and current layer error. It is the **matrix\* matrix** part. Here, we use

**# pragma omp parallel for num\_threads(4)** to create 4 threads to do the calculation

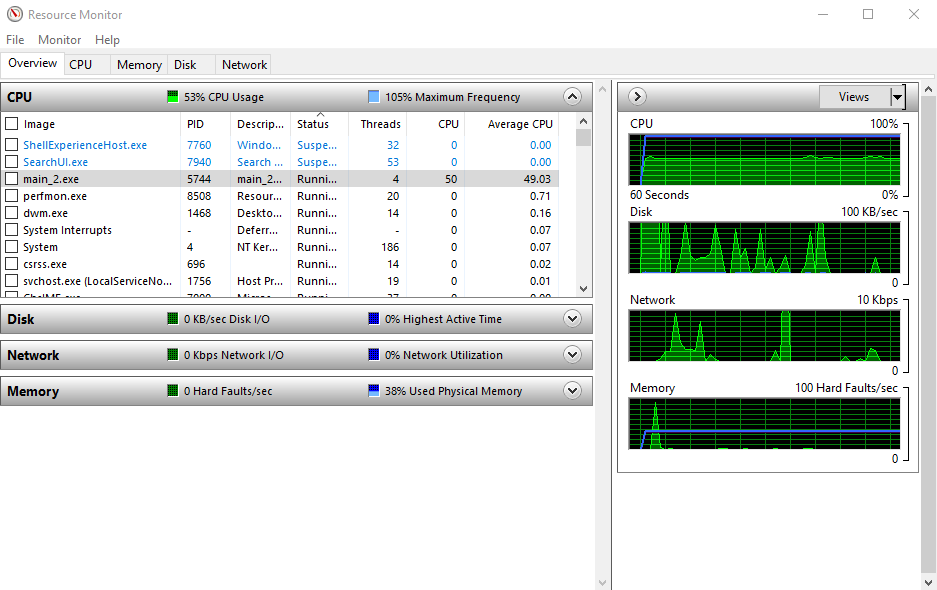


1. How to compile?

For Windows,

**cl -openmp main.cpp ANN.cpp**

1. Experimental Results



* 1. As the picture showed, 4 threads and average CPU is 53
  2. CPU time per epoch is around (10000, 14000) milliseconds (Using RRS638 computer)
  3. Speedup=T1/Tp=1548000/683744=2.26
  4. Hardware environment:

Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz

Intel64 Family 6 Model 60