

MP Project

Amir Hossein Pashae Hir 9731013

The background is a blue gradient with a complex network of white lines and dots, resembling a molecular or network structure. The lines connect various points, some of which are highlighted with small white dots. The overall effect is a sense of interconnectedness and geometric complexity.

GENANN

Key functions

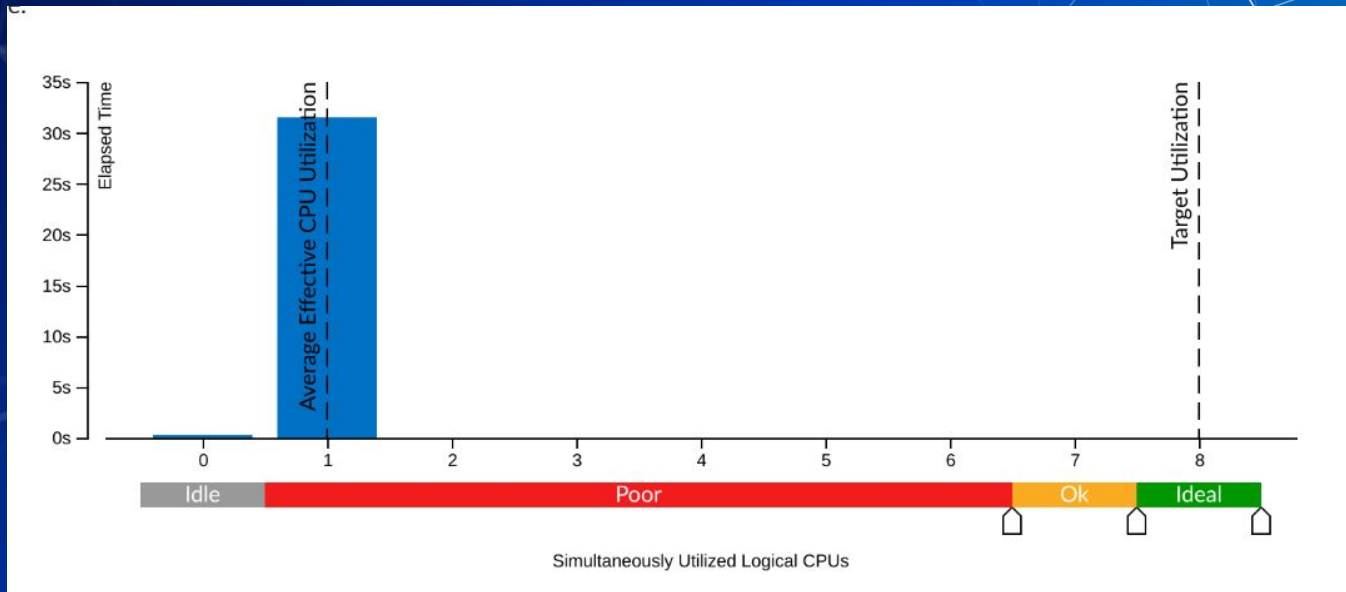
- `genann_run`
- `genann_train`
- Lookup table for sigmoid
- Cache friendly

Analysis

Function	Module	CPU Time ?	% of CPU Time ?
genann_train	a.out	17.224s	54.6%
genann_run	a.out	9.966s	31.6%
__isoc99_fscanf	libc.so.6	3.230s	10.2%
genann_act_sigmoid_cached	a.out	0.988s	3.1%
__isnan	a.out	0.052s	0.2%
[Others]	N/A*	0.060s	0.2%

**N/A is applied to non-summable metrics.*

Analysis (Cont.)



Analysis (Cont.)

234			
235	/* Figure input layer */		
236	for (j = 0; j < ann->hidden; ++j) {		
237	double sum = *w++ * -1.0;		
238	for (k = 0; k < ann->inputs; ++k) {	4.1%	1.288s
239	sum += *w++ * i[k];	15.4%	4.848s
240	}		
241	*o++ = genann_act_hidden(ann, sum);	0.1%	0.024s
242	}		
243			

Analysis (Cont.)

382			
383	for (j = 0; j < ann->hidden; ++j) {	0.1%	0.024s
384	*w++ += *d * learning_rate * -1.0;	0.2%	0.056s
385	for (k = 1; k < (h == 0 ? ann->inputs : ann->hidden) + 1; ++k) {	0.1%	0.016s
386	*w++ += *d * learning_rate * i[k-1];	19.6%	6.164s
387	}		
388	++d;		
389	}		
280			



GEANN **Parallel**

Key idea

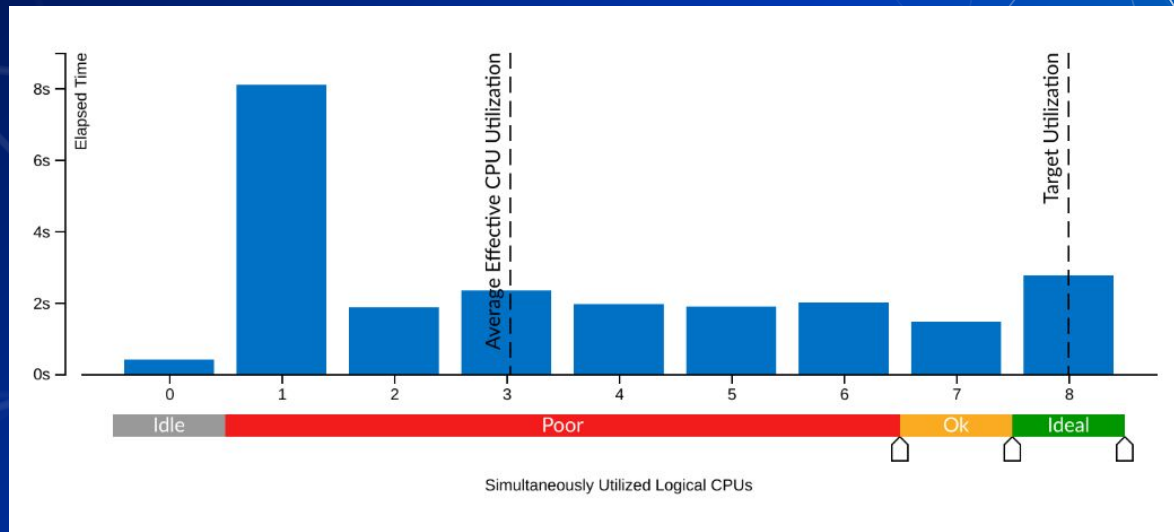
- Find loop with independ body
 - All loops except loops over hidden layers
- Parallel outermost loop
 - Each thread has its computation area
 - Cache friendly
 - Job distribution is uniform
 - Minimize context switch

Analysis

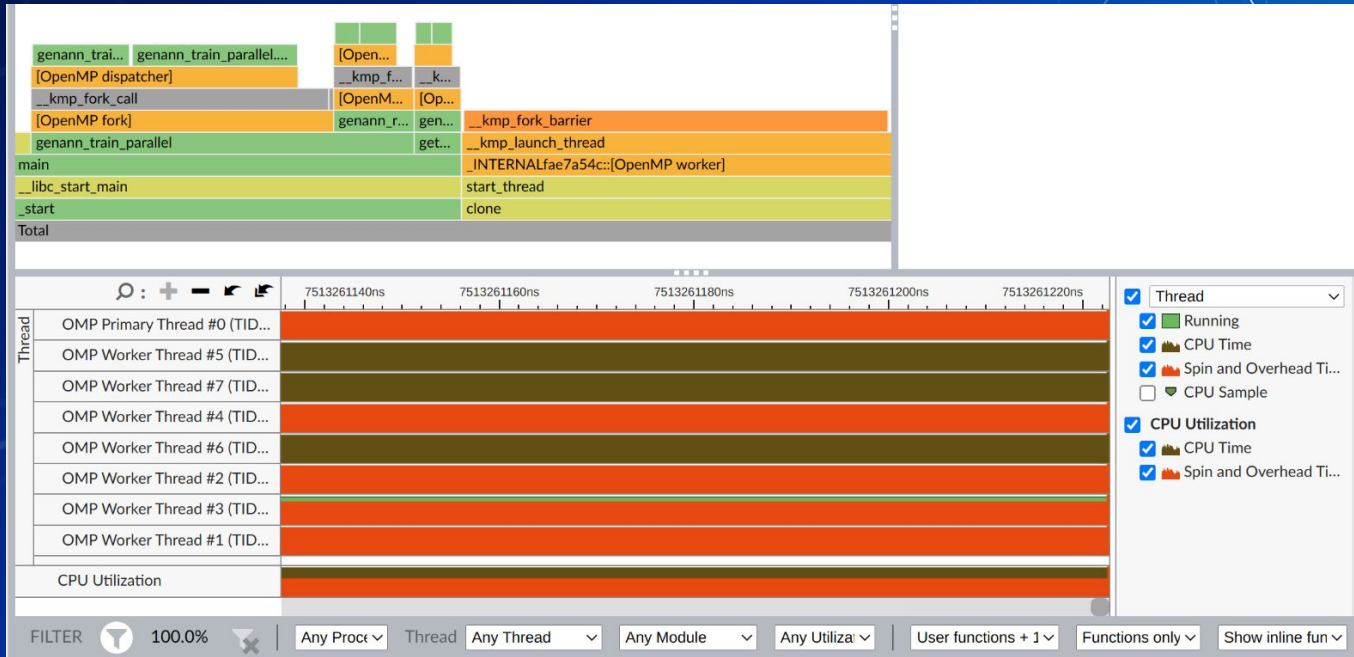
Function	Module	CPU Time ⓘ	% of CPU Time ⓘ
__kmp_fork_barrier	libiomp5.so	76.549s 🚩	48.2% 🚩
genann_train_parallel.extracted.33	a.out	29.800s	18.8%
genann_train_parallel.extracted	a.out	17.499s	11.0%
__kmp_fork_call	libiomp5.so	9.797s 🚩	6.2% 🚩
genann_run_parallel.extracted.19	a.out	9.671s	6.1%
[Others]	N/A*	15.345s	9.7%

*N/A is applied to non-summable metrics.

Analysis (Cont.)



Analysis (Cont.)



Batch learning

- Partition tests into batches
- Learn each test in a batch simultaneously
- Merge batches parallel
- Result:
 - Poor accuracy because of merging
 - Lots of memory copying



GEANN **On GPU**

Key idea

- Translate All pointers to indices
 - done in parallelizing of serial algorithm
- Find independ loop body
- Figure out linear algebra operations
- Categorize operations and implement them for GPU
- Consider memory coalescing and synchronization

Reduced operations

- Matrix multiplication to a vector
- Sigmoid of a vector
- Transpose a matrix
 - Use share memory in order to memory coalescing
- Update a matrix by multiplication of two vectors
- Vectors add/subtract

Results

Algorithm / Layers	1 32	1 512	2 512
Serial	4.82 91	31.12 94	59.70 94
Batch	31.12 48	24.14 76	48.21 20
Parallel	3.8 92	7.93 93	14.41 94
GPU	5.31 86	19.42 91	35.49 91

Config

- CPU: 8 cores
- GPU:
 - Model: MX450
 - Core: 896



THANKS!