Question 1:

Parallel Stochastic Gradient Descent

Team member: Yue Sun

Below is my plan for the project:

week1:
Implement serial SGD and locked parallel SGD in OpenMP.

week2:

Implement hogwild algorithm in OpenMP.
Compare the three algorithms on small and large datasets.

Week3:
Compare the performance with Tensorflow.

Week4:
If time permits, I will implement the three algorithms in CUDA.

GitHub:
`https://github.com/syalexandra/high-performance-computing.git`

The processor :
2.8 GHz Quad-Core Intel Core i7

g++ version:
gcc version 9.2.0 (Homebrew GCC 9.2.0_3)

Question 2:

```
g++ -std=c++11 -O3 -march=native -fopenmp -fno-tree-vectorize fast-sin.cpp  -o fast-sin

Reference time: 0.2293
Taylor time:    1.9021      Error: 6.927903e-12
```

```
Intrin time:    0.0025      Error: 6.927903e-12
Vector time:    0.0019      Error: 6.928014e-12
```

Below is the testing for extra point.

```
Reference time: 0.2506
Taylor time:    19.9274      Error: 6.928680e-12
Intrin time:    4.2626      Error: 6.928680e-12
```

Extra point algorithm(Vectorize by AVX):

1. Map x on the full range to [-pi/4, 7pi/4].
   x=x-floor[(x+pi/4)/2pi]*2pi
2. Map x to range [-pi/4, pi/4].
   k=floor[(x+pi/4)/(pi/2)]
   x=x-k*pi/2
3. After the above transformation, k=0,1,2,3. x in [-pi/4, pi/4]
4. Now we can do the approximation using Taylor expansion.

	1	x	-x^2/2!	-x^3/3!	……	-x^12/12!
x:[-pi/4,pi/4] k=0	0	1	0	1		0
x:[pi/4,3pi/4] k=1	1	0	1	0		1
x:[3pi/4,5pi/4] k=2	0	-1	0	-1		0
x:[5pi/4,7pi/4] k=3	-1	0	-1	0		-1

So we can generalize the function to
f(x)=A*(X0+X2+X4+X6+X8+X10+X12)+B*(X1+X3+X5+X7+X9+X11)
Where X0=1, X1=x,X2=-x^2/2!, X3=-x^3/3!,……,X12=-x^12/12!
A=[1-(k//2)*2]*[(k%2)]
B=[1-(k//2)*2]*[1-(k%2)]

Question 3:

```
g++ -std=c++11 -O3 -march=native -fopenmp -fno-tree-vectorize omp-scan.cpp  -o omp-scan
```

```
2 threads:


sequential-scan = 0.335280s
parallel-scan   = 0.262539s
error = 0



4 threads:

sequential-scan = 0.339973s
parallel-scan   = 0.177510s
error = 0


8 threads:

sequential-scan = 0.329807s
parallel-scan   = 0.151668s
error = 0
```