Дефолт

#include <math.h>

#include <iostream>

#include <time.h>

#define N 2048

#define M 10

float A[N][N];

float B[N][N];

float R[N][N];

float AA[N][N];

float Save[N][N];

int main() {

int a = time(NULL);

srand(time(NULL));

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++) {

A[i][j] = rand() % 10;

Save[i][j] = A[i][j];

}

float MaxCol = 0;

float MaxStr = 0;

float TmpStr, TmpCol;

for (int i = 0; i < N; i++) {

TmpStr = 0;

TmpCol = 0;

for (int j = 0; j < N; j++) {

TmpStr += A[i][j];

TmpCol += A[j][i];

}

if (TmpStr > MaxStr)

MaxStr = TmpStr;

if (TmpCol > MaxCol)

MaxCol = TmpCol;

}

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

B[i][j] = A[j][i] / (MaxCol \* MaxStr);

float Tmp;

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++) {

Tmp = 0;

for (int k = 0; k < N; k++)

Tmp += B[i][k] \* A[k][j];

R[i][j] = -Tmp;

AA[i][j] = -Tmp;

}

for (int i = 0; i < N; i++) {

R[i][i]++;

AA[i][i]++;

}

for (int it = 0; it < M - 2; it++)

{

if (it % 2 == 0)

{

for (int i = 0; i < N; i++)

AA[i][i]++;

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

for (int k = 0; k < N; k++)

Tmp += R[i][k] \* AA[k][j];

A[i][j] = Tmp;

}

}

else {

for (int i = 0; i < N; i++)

A[i][i]++;

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

for (int k = 0; k < N; k++)

Tmp += R[i][k] \* A[k][j];

AA[i][j] = Tmp;

}

}

}

if (M % 2 == 0)

{

for (int i = 0; i < N; i++)

AA[i][i]++;

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

for (int k = 0; k < N; k++)

Tmp += AA[i][k] \* B[k][j];

A[i][j] = Tmp;

}

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

for (int k = 0; k < N; k++)

Tmp += Save[i][k] \* A[k][j];

AA[i][j] = Tmp;

}

for (int i = 0; i < N; i++) {

float al = 0;

for (int j = 0; j < N; j++) {

al += fabs(AA[i][j]);

}

if (al < 0.9 || al>1.1) {

printf("Not OK");

return 0;

}

}

}

else {

for (int i = 0; i < N; i++)

A[i][i]++;

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

for (int k = 0; k < N; k++)

Tmp += A[i][k] \* B[k][j];

AA[i][j] = Tmp;

}

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

for (int k = 0; k < N; k++)

Tmp += Save[i][k] \* AA[k][j];

A[i][j] = Tmp;

}

for (int i = 0; i < N; i++) {

float al = 0;

for (int j = 0; j < N; j++) {

al += fabs(A[i][j]);

}

if (al < 0.9 || al>1.1) {

printf("Not OK");

return 0;

}

}

}

int b = time(NULL);

std::cout << b - a<<std::endl;

return 0;

}

-O3 206

715

Simd расширения

#include <math.h>

#include <iostream>

#include <xmmintrin.h>

#include <time.h>

#define N 2048

#define M 10

float MulVec(float\* x, float\* y, int n) {

\_\_m128\* xx, \* yy;

\_\_m128 p, s;

xx = (\_\_m128\*)x;

yy = (\_\_m128\*)y;

s = \_mm\_setzero\_ps();

for (int i = 0; i < n / 4; i++) {

p = \_mm\_mul\_ps(xx[i], yy[i]);

s = \_mm\_add\_ps(s, p);

}

p = \_mm\_movehl\_ps(p, s);

s = \_mm\_add\_ps(s, p);

p = \_mm\_shuffle\_ps(s, s, 1);

s = \_mm\_add\_ss(s, p);

float sum;

\_mm\_store\_ss(&sum, s);

return sum;

}

float A[N][N];

float Save[N][N];

float B[N][N];

float R[N][N];

float AA[N][N];

float TM[N][N];

int main() {

int a = time(NULL);

srand(time(NULL));

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++) {

A[i][j] = rand() % 10;

Save[i][j] = A[i][j];

}

float MaxCol = 0;

float MaxStr = 0;

float TmpStr, TmpCol;

for (int i = 0; i < N; i++) {

TmpStr = 0;

TmpCol = 0;

for (int j = 0; j < N; j++) {

TmpStr += A[i][j];

TmpCol += A[j][i];

}

if (TmpStr > MaxStr)

MaxStr = TmpStr;

if (TmpCol > MaxCol)

MaxCol = TmpCol;

}

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

B[j][i] = A[i][j] / (MaxCol \* MaxStr);

float Tmp;

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

TM[j][i] = A[i][j];

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++) {

Tmp = 0;

Tmp = MulVec(B[i], TM[j], N);

R[i][j] = -Tmp;

AA[i][j] = -Tmp;

}

for (int i = 0; i < N; i++) {

R[i][i]++;

AA[i][i]++;

}

for (int it = 0; it < M - 2; it++)

{

if (it % 2 == 0)

{

for (int i = 0; i < N; i++)

AA[i][i]++;

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

TM[j][i] = AA[i][j];

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

Tmp = MulVec(R[i], AA[j], N);

A[i][j] = Tmp;

}

}

else {

for (int i = 0; i < N; i++)

A[i][i]++;

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

TM[j][i] = A[i][j];

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

Tmp = MulVec(R[i], A[j], N);

AA[i][j] = Tmp;

}

}

}

if (M % 2 == 0)

{

for (int i = 0; i < N; i++)

AA[i][i]++;

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

TM[j][i] = B[i][j];

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

Tmp = MulVec(AA[i], TM[j], N);

A[i][j] = Tmp;

}

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

TM[j][i] = A[i][j];

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

Tmp = MulVec(Save[i], TM[j], N);

AA[i][j] = Tmp;

}

for (int i = 0; i < N; i++) {

float al = 0;

for (int j = 0; j < N; j++) {

al += fabs(AA[i][j]);

}

if (al < 0.9 || al>1.1) {

printf("Not OK");

return 0;

}

}

}

else {

for (int i = 0; i < N; i++)

A[i][i]++;

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

TM[j][i] = B[i][j];

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

Tmp = MulVec(A[i], TM[j], N);

AA[i][j] = Tmp;

}

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

TM[j][i] = AA[i][j];

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

Tmp = 0;

Tmp = MulVec(Save[i], TM[j], N);

A[i][j] = Tmp;

}

for (int i = 0; i < N; i++) {

float al = 0;

for (int j = 0; j < N; j++) {

al += fabs(A[i][j]);

}

if (al < 0.9 || al>1.1) {

printf("Not OK");

return 0;

}

}

}

int b = time(NULL);

std::cout <<"All right "<< b – a<<std::endl;

return 0;

}

-O3 34

102

Blas библиотека

#include <math.h>

#include <stdio.h>

#include <time.h>

#include <mkl.h>

#define N 2048

#define M 10

float A[N][N];

float Save[N][N];

float B[N][N];

float R[N][N];

float AA[N][N];

float TM[N][N];

float Tmp;

int main() {

int a = time(NULL);

srand(time(NULL));

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

A[i][j] = rand() % 10;

Save[i][j] = A[i][j];

}

float MaxCol = 0;

float MaxStr = 0;

float TmpStr, TmpCol;

for (int i = 0; i < N; i++) {

TmpStr = 0;

TmpCol = 0;

for (int j = 0; j < N; j++) {

TmpStr += A[i][j];

TmpCol += A[j][i];

}

if (TmpStr > MaxStr)

MaxStr = TmpStr;

if (TmpCol > MaxCol)

MaxCol = TmpCol;

}

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

B[i][j] = A[j][i] / (MaxCol \* MaxStr);

cblas\_sgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, N, N, N, -1.0, &B[0][0], N, &A[0][0], N, 0.0, &AA[0][0], N);

cblas\_sgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, N, N, N, -1.0, &B[0][0], N, &A[0][0], N, 0.0, &R[0][0], N);

for (int i = 0; i < N; i++) {

R[i][i]++;

AA[i][i]++;

}

for (int it = 0; it < M - 2; it++)

{

if (it % 2 == 0)

{

for (int i = 0; i < N; i++)

AA[i][i]++;

cblas\_sgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, N, N, N, 1.0, &R[0][0], N, &AA[0][0], N, 0.0, &A[0][0], N);

}

else {

for (int i = 0; i < N; i++)

A[i][i]++;

cblas\_sgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, N, N, N, 1.0, &R[0][0], N, &A[0][0], N, 0.0, &AA[0][0], N);

}

}

if (M % 2 == 0)

{

for (int i = 0; i < N; i++)

AA[i][i]++;

cblas\_sgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, N, N, N, 1.0, &AA[0][0], N, &B[0][0], N, 0.0, &A[0][0], N);

cblas\_sgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, N, N, N, 1.0, &Save[0][0], N, &A[0][0], N, 0.0, &AA[0][0], N);

for (int i = 0; i < N; i++) {

float al = 0;

for (int j = 0; j < N; j++) {

al += fabs(AA[i][j]);

}

if (al < 0.9 || al>1.1) {

printf("Not OK");

return 0;

}

}

}

else {

for (int i = 0; i < N; i++)

A[i][i]++;

cblas\_sgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, N, N, N, 1.0, &A[0][0], N, &B[0][0], N, 0.0, &AA[0][0], N);

cblas\_sgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, N, N, N, 1.0, &Save[0][0], N, &AA[0][0], N, 0.0, &A[0][0], N);

for (int i = 0; i < N; i++) {

float al = 0;

for (int j = 0; j < N; j++) {

al += fabs(A[i][j]);

}

if (al < 0.9|| al>1.1) {

printf("Not OK");

return 0;

}

}

}

int b = time(NULL);

printf("ALL RIGHT\n%d\n", b - a);

return 0;

}

1.3 sec

Icx name.c –qmkl –o name.out