

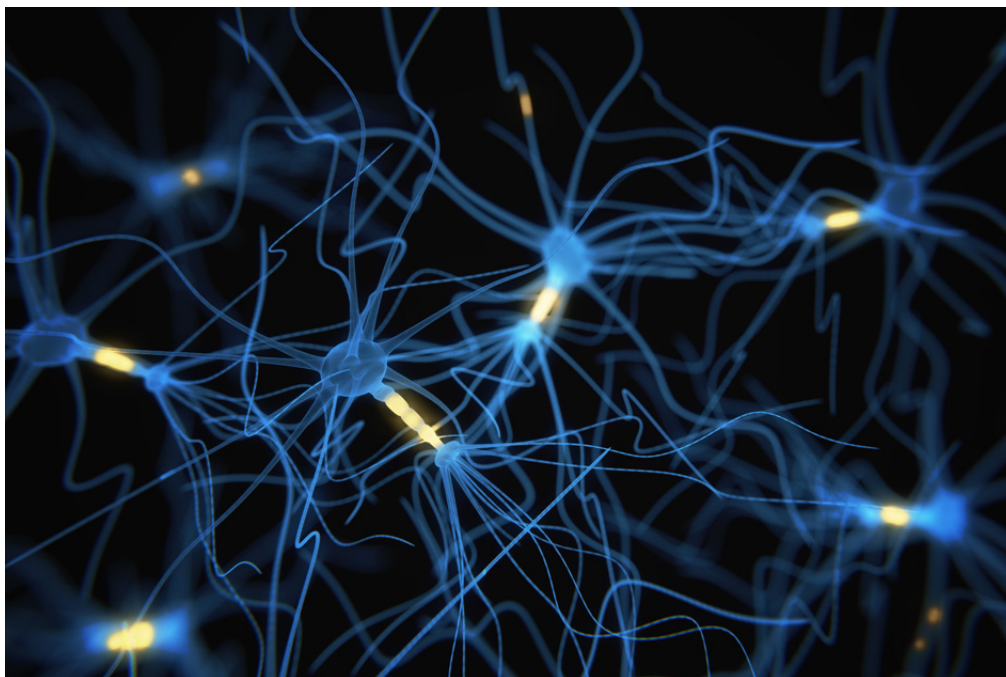


ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ

Παράλληλη Επεξεργασία

Εργασία Εαρινού Εξαμήνου 2017-2018



Σταυρούλα Δρίτσα, 6040
dritsa@ceid.upatras.gr

Νικόλαος Ζαχαρόπουλος, 6048
zachar@ceid.upatras.gr

Δαμιανός Ντούμη - Σιγάλας, 6157
nsigalas@ceid.upatras.gr

Ακολουθιακό Πρόγραμμα

Ζητούμενο 1Α - Βελτιστοποίηση Αντιγραφών

Στον κώδικα που μας δίνεται παρατηρούμε ότι στο τέλος κάθε επανάληψης η νέα τιμή δυναμικού του κάθε νευρώνα (διάνυσμα *u*plus), αντιγράφεται στην προηγούμενη (διάνυσμα *u*). Αυτή η διαδικασία εισάγει έναν μεγάλο αριθμό επαναλήψεων ο οποίος μπορεί να αποφευχθεί εναλλάσσοντας την τιμή των δεικτών των διανυσμάτων *u*plus και *u* μεταξύ τους, όπως φαίνεται στον κώδικα παρακάτω.

```
--- ./original/lif1d.c
+++ ./1a/lif1d.c
@@ -49,7 +49,7 @@
     double      mu;
     double      s_min;
     double      s_max;
-   double      *u, *uplus, *sigma, *omega, *omegal;
+   double      *u, *uplus, *sigma, *omega, *omegal, *temp;
     double      sum;
     double      time;
     struct timeval global_start, global_end, IO_start, IO_end;
@@ -326,8 +326,13 @@
     /*
      * Update network elements and set u[i] = 0 if u[i] > uth
      */
+
+   //CEID 1a
+   temp = uplus;
+   uplus = u;
+   u = temp;
+
     for (i = 0; i < n; i++) {
-       u[i] = uplus[i];
       if (u[i] > uth) {
         u[i] = 0.0;
       }
     }
@@ -379,4 +379,3 @@
     return 0;
 }
-
```

Τα αποτελέσματα που παίρνουμε από την εκτέλεση του προγράμματος μετά τις αλλαγές έχουν ως εξής:

gcc -O3 -Wall -Wextra lif1d.c				
n	r	Calculation Time	I/O Time	Execution Time
1000	350	142,934061	0,010088	142,944149
2000	700			0,000000
3000	1000			0,000000
4000	1300			0,000000
5000	1600			0,000000

gcc -O3 -Wall -Wextra lif1d.c				
n	r	Calculation Time	I/O Time	Execution Time
1000	350	142,934061	0,010088	142,944149
2000	700			0,000000
3000	1000			0,000000
4000	1300			0,000000
5000	1600			0,000000

Ζητούμενο 1B - Αναδιοργάνωση Υπολογισμών

Η σχέση που μας δίνει το νέο δυναμικό του νευρώνα i την χρονική στιγμή $t + \Delta t$ είναι η εξής:

$$u_i(t + \Delta t) = u_i(t) + \Delta t \cdot \left(\mu - u_i(t) + \frac{1}{N_c} \sum_{j \in \{N_i\}}^{N_c} \sigma_{ij} \cdot [u_j(t) - u_i(t)] \right)$$

Στην σχέση αυτή θα προσπαθήσουμε να αναδιοργανώσουμε τους υπολογισμούς που γίνονται στο άθροισμα:

$$\sum_{j \in \{N_i\}}^{N_c} \sigma_{ij} \cdot [u_j(t) - u_i(t)] = \sum_{j \in \{N_i\}}^{N_c} \sigma_{ij} u_j(t) - \sum_{j \in \{N_i\}}^{N_c} \sigma_{ij} u_i(t) = \sum_{j \in \{N_i\}}^{N_c} \sigma_{ij} u_j(t) - u_i(t) \sum_{j \in \{N_i\}}^{N_c} \sigma_{ij}$$

Στο δεύτερο άθροισμα που έχει προκύψει παρατηρούμε ότι ο όρος $u_i(t)$ είναι ανεξάρτητος του j και επομένως μπορεί να βγει έξω από αυτό. Συνεπώς για κάθε j χρησιμοποιείται το άθροισμα της i -οστής γραμμής του μητρώου σ . Ο όρος αυτός είναι σταθερός και ανεξάρτητος από κάθε επανάληψη της προσομοίωσης μας. Γι' αυτόν τον τον λόγο και με σκοπό να εξοικονομήσουμε περιττούς υπολογισμούς δημιουργούμε πριν την έναρξη της προσομοίωσης ένα διάνυσμα `sigma_vector` μήκους n στο οποίο υπολογίζουμε σε κάθε θέση το άθροισμα της αντίστοιχης γραμμής του μητρώου. Επειτα όπου χρειάζεται αντί να υπολογίζουμε εκ νέου την τιμή θα ανατρέχουμε σε αυτό. Ουσιαστικά μας ενδιαφέρει πως κάθε νευρώνας i επηρεάζεται από το δυναμικό των j γειτόνων του, με την μορφή του διανύσματος να είναι η παρακάτω:

$$\boxed{\sum \sigma_{1j}} \quad \boxed{\sum \sigma_{2j}} \quad \boxed{\sum \sigma_{3j}} \quad \dots \quad \boxed{\sum \sigma_{nj}}$$

Οι παραπάνω αλλαγές στον κώδικα συνοψίζονται ως εξής:

```
--- ./1a/lifid.c
+++ ./1b/lifid.c
@@ -36,7 +36,7 @@
 {
     FILE          *output1, *output2;
     long          n, r;
-    long          i, j;
+    long          i, j, var;
     long          it;
     double        divide;
     double        dt;
@@ -49,8 +49,8 @@
     double        mu;
     double        s_min;
     double        s_max;
-    double        *u, *uplus, *sigma, *omega, *omegal, *temp;
-    double        sum;
+    double        *u, *uplus, *sigma, *omega, *omegal, *temp, *sigma_vector;
+    double        sum, sum1;
     double        time;
     struct timeval global_start, global_end, IO_start, IO_end;
     double        global_usec, IO_usec = 0.0;
@@ -306,6 +306,17 @@
     /*
      * Temporal iteration.
      */
+    //CEID 1b
+    double var2 = dt / divide;
+
     sigma_vector = (double *)calloc(n, sizeof(double));
     for (i = 0; i < n; i++) {
         sigma_vector[i] = 0.0;
         for (j = 0; j < n; j++) {
             sigma_vector[i] += (-sigma[i*n + j]);
         }
     }
 }
```

```

+
+ gettimeofday(&global_start, NULL);
+ for (it = 0; it < itime; it++) {
+     /*
@@ -314,13 +325,18 @@
+     for (i = 0; i < n; i++) {
+         uplus[i] = u[i] + dt * (mu - u[i]);
+         sum = 0.0;
+         sum1 = 0.0;
+         /*
+         * Iteration over neighbouring neurons.
+         */
+         var = i*n;
+         for (j = 0; j < n; j++) {
-             sum += sigma[i * n + j] * (u[j] - u[i]);
+             sum1 += sigma[var + j] * u[j];
+         }
-         uplus[i] += dt * sum / divide;
+
+         sum = sum1 + u[i]*sigma_vector[i];
+
+         uplus[i] += var2 * sum;
+     }
+
+     /*

```

Τα αποτελέσματα που παίρνουμε από την εκτέλεση του προγράμματος μετά τις αλλαγές έχουν ως εξής:

gcc -O3 -Wall -Wextra lif1d.c				
n	r	Calculation Time	I/O Time	Execution Time
1000	350	142,934061	0,010088	142,944149
2000	700			0,000000
3000	1000			0,000000
4000	1300			0,000000
5000	1600			0,000000

gcc -O3 -Wall -Wextra lif1d.c				
n	r	Calculation Time	I/O Time	Execution Time
1000	350	142,934061	0,010088	142,944149
2000	700			0,000000
3000	1000			0,000000
4000	1300			0,000000
5000	1600			0,000000

Ζητούμενο 1C - Χρήση βιβλιοθήκης BLAS

Στο προηγούμενο ερώτημα ασχοληθήκαμε με την βελτιστοποίηση του δεύτερου αθροίσματος όπως προέκυψε μετά την αναδιοργάνωση των υπολογισμών. Τώρα θα προσπαθήσουμε να βελτιστοποιήσουμε το πρώτο άθροισμα κάνοντας χρήση της βιβλιοθήκης συναρτήσεων BLAS. Το άθροισμα αυτό έχει την μορφή:

$$\sum_{j \in \{N_i\}}^{N_c} \sigma_{ij} u_j(t)$$

Παρατηρούμε ότι για κάθε νευρώνα i υπολογίζουμε πως επηρεάζεται από το δυναμικό του νευρώνα j . Η πράξη αυτή αντιστοιχεί με τον πολλαπλασιασμό του μητρώου σ επί διάνυσμα u . Θα χρησιμοποιήσουμε την βιβλιοθήκη BLAS η οποία παρέχει συναρτήσεις για τον υπολογισμό τέτοιου είδους πράξεων με πολύ πιο αποδοτικό τρόπο. Συγκεκριμένα θα χρειαστούμε την συνάρτηση `dgemv` η οποία εκτελεί την παρακάτω σχέση για `double` ακρίβειας δεδομένα:

$$y = \alpha Ax + \beta y$$

όπου A είναι το μητρώο εισόδου διαστάσεων $N \times M$, x το διάνυσμα, α και β βαθμωτοί και y το τελικό αποτέλεσμα. Επομένως η κλήση της συνάρτησης για τα δικά μας δεδομένα θα έχει την μορφή:

```
cblas_dgemv(CblasRowMajor, CblasNoTrans, n, n, 1, sigma, n, u, 1, 0, mysum1, 1);
```

όπου:

CblasRowMajor το μητρώο σ είναι δομημένο κατά γραμμές

CblasNoTrans δεν θέλουμε το ανάστροφο του μητρώου σ

n η διάσταση M του μητρώου

n η διάσταση N του μητρώου

1 η τιμή του βαθμωτού α

sigma το μητρώο εισόδου

n η διάσταση M του διανύσματος u ($n \times 1$)

u το διάνυσμα

1 `incX`, το βήμα για τον εντοπισμό του επόμενου στοιχείου του διανύσματος

0 η τιμή του βαθμωτού β

mysum1 το διάνυσμα y στο οποίο αποθηκεύεται το αποτέλεσμα

1 `incY`, το βήμα για τον εντοπισμό του επόμενου στοιχείου του διανύσματος y

Συνολικά οι αλλαγές στον κώδικα είναι οι εξής:

```
--- ./1b/lif1d.c
+++ ./1c/lif1d.c
@@ -3,6 +3,7 @@
#include <getopt.h>
#include <stdlib.h>
#include <sys/time.h>
+#include <cblas.h>

#define MIN_NUM_OF_NEURONS (1L)
#define DEF_NUM_OF_NEURONS (1000L)
@@ -36,7 +37,7 @@
{
```

```

FILE      *output1, *output2;
long      n, r;
- long    i, j, var;
+ long    i, j;
long      it;
double    divide;
double    dt;
@@ -50,7 +51,7 @@
double    s_min;
double    s_max;
double    *u, *uplus, *sigma, *omega, *omegal, *temp, *sigma_vector;
- double  sum, sum1;
+ double  sum;
double    time;
struct timeval global_start, global_end, IO_start, IO_end;
double    global_usec, IO_usec = 0.0;
@@ -317,24 +318,26 @@
    }
}

+ double *mysum1;
+ mysum1 = (double *)calloc(n, sizeof(double));
+
+
gettimeofday(&global_start, NULL);
for (it = 0; it < itime; it++) {
    /*
     * Iteration over elements.
     */
+ cblas_dgemv(CblasRowMajor, CblasNoTrans, n, n, 1, sigma, n, u, 1, 0, mysum1, 1);
+
    for (i = 0; i < n; i++) {
        uplus[i] = u[i] + dt * (mu - u[i]);
        sum = 0.0;
-        sum1 = 0.0;
+
        /*
         * Iteration over neighbouring neurons.
         */
-        var = i*n;
-        for (j = 0; j < n; j++) {
-            sum1 += sigma[var + j] * u[j];
-        }
-
-        sum = sum1 + u[i]*sigma_vector[i];
+        sum = mysum1[i] + u[i]*sigma_vector[i];
+
        uplus[i] += var2 * sum;
    }
@@ -349,6 +352,7 @@
    u = temp;

    for (i = 0; i < n; i++) {
+        //u[i] = uplus[i];
        if (u[i] > uth) {
            u[i] = 0.0;
        }
    }
}

```