

Συνέλιξη Διανυσμάτων

Προαιρετική Προγραμματιστική Εργασία

Εξεταζόμενο Μάθημα: Σήματα & Συστήματα

Επιμέλεια: Παρμενίων Χαριστός

ΑΕΜ: 3173

Στην εργασία αυτή κληθήκαμε να προγραμματίσουμε σε C++ μία συνάρτηση υπολογισμού συνέλιξης διανυσμάτων και να την εφαρμόσουμε σε μερικά .wav αρχεία ήχου.

Αρχικά, ωστόσο, κληθήκαμε να επαληθεύουμε την λειτουργία της συνάρτησης παράγοντας ένα διάνυσμα A, τυχαίων αριθμών και συνελίσσοντας το με το δοθέν B = {1/5 1/5 1/5 1/5 1/5}.

Καταρχάς, λοιπόν, το πρόγραμμα λαμβάνει ως είσοδο από το πληκτρολόγιο του χρήστη, τον αριθμό των διαστάσεων του διανύσματος A. Πρέπει να είναι μεγαλύτερος του 10. Αν δώσει μικρότερη, ξαναζητάται η είσοδος, ενώ αν δεν δοθεί είσοδος με έγκυρο format (eg. "abc", 12.2), επιλέγεται ως default μήκος το 100.

Μόλις διευκρινιστούν και δημιουργηθούν τα A, B καλείται η συνάρτηση convolve() η οποία δέχεται δύο vectors και υλοποιεί τη συνέλιξή τους, επιστρέφοντάς την σε ένα νέο vector. Για τον προγραμματισμό της, αξιοποιήθηκε ο τύπος που περιγράφεται στην παρακάτω διαφάνεια της [παρουσίασης](#) του πανεπιστημίου του Εδιμβούργου.

Έπειτα, το αποτέλεσμα τυπώνεται σε ένα αρχείο εξόδου ABconvolution.txt.

Convolutions of Discrete Functions
Convolutions of Continuous Functions

Convolution of Vectors
Mid-lecture Problem
Convolution of Matrices

Convolution of Vectors

If a function f ranges over a finite set of values $\mathbf{a} = a_1, a_2, \dots, a_n$, then it can be represented as vector $[a_1 \ a_2 \ \dots \ a_n]$.

Definition: Convolution of Vectors

If the functions f and g are represented as vectors $\mathbf{a} = [a_1 \ a_2 \ \dots \ a_m]$ and $\mathbf{b} = [b_1 \ b_2 \ \dots \ b_n]$, then $f * g$ is a vector $\mathbf{c} = [c_1 \ c_2 \ \dots \ c_{m+n-1}]$ as follows:

$$c_x = \sum_u a_u b_{x-u+1}$$

where u ranges over all legal subscripts for a_u and b_{x-u+1} , specifically $u = \max(1, x - n + 1) \dots \min(x, m)$.

Frank Keller

Computational Foundations of Cognitive Science

5

Μερικά παραδείγματα εκτέλεσης για αυτό το ζητούμενο παρατίθενται παρακάτω.

```
Enter vector size:11

The selected length is 11.
A: 6977 8787 4490 18898 5533 258 3187 4978 3051 7632 19232
B: 0.2 0.2 0.2 0.2 0.2
A*B: 1395.4 3152.8 4050.8 7830.4 8937 7593.2 6473.2 6570.8 3401.4 3821.2 7616 6978.6 5983 5372.8 3846.4
The first task has been completed. You can find the result of the convolution in AB_convolution.txt of the /out folder.
```

Ορθή Είσοδος και Υπολογισμός

```
Enter vector size:11

The selected length is 11.
A: 10410 402 3670 12545 32711 31251 30341 8604 22461 1309 27623
B: 0.2 0.2 0.2 0.2 0.2
A*B: 2082 2162.4 2896.4 5405.4 11947.6 16115.8 22103.6 23090.4 25073.6 18793.2 18067.6 11999.4 10278.6 5786.4 5524.6
The first task has been completed. You can find the result of the convolution in AB_convolution.txt of the /out folder.
```

Ορθή Είσοδος και Υπολογισμός

```
Enter vector size:0
Enter vector size:2
Enter vector size:-1
Enter vector size:10
Enter vector size:a

The selected length is 100.
```

Παράδειγμα Αντιμετώπισης Άκυρης Εισόδου

Για το Μέρος Β, αξιοποιείται η ανοιχτή βιβλιοθήκη AudioFile (συμπεριλαμβάνεται στο παραδοτέο το header file της). Με τη βοήθειά της, δημιουργούμε αντικείμενα AudioFile στα οποία φορτώνουμε (συνάρτηση load) τα αντίστοιχα .wav files.

Έπειτα, αρχικοποιούμε αντίστοιχα vectors στα οποία εκχωρούμε τα στοιχεία του samples[0] array του αντίστοιχου AudioFile. Αυτό περιλαμβάνει όλα τα δείγματα που επιλέγονται με το default sampling rate για το συγκεκριμένο αρχείο. Επομένως, έτσι, έχουμε μετατρέψει τα αρχικά .wav files σε συμβατικά vectors τα οποία μπορούμε να χρησιμοποιήσουμε στην convolve() συνάρτησή μας.

Με τον τρόπο αυτό, υπολογίζεται το ζητούμενο Pink_Noise_Sample vector, το οποίο μετατρέπουμε σε ένα νέο AudioFile εκχωρώντας το στο samples array του. Τέλος, δημιουργούμε το πραγματικό .wav file με τη συνάρτηση save και την παράμετρο Wave της AudioFileFormat.

Για το τελευταίο πανομοιότυπο ερώτημα, ακολουθείται η ίδια διαδικασία, ωστόσο προηγουμένως δημιουργούμε ένα δικό μας σήμα λευκού θορύβου, δηλαδή το αντίστοιχο vector, με τη βοήθεια των εργαλείων της βιβλιοθήκης <random>.

Συγκεκριμένα, θέτουμε το επιθυμητό μέγεθος samples (ίσο με το αντίστοιχο του δοθέντος Pink Noise για ευκολία), και εκχωρούμε επαναληπτικά στο νέο διάνυσμα που αντιπροσωπεύει το σήμα λευκού θορύβου, τις τιμές που παράγει το random_device μας, οι οποίες ακολουθούν ομοιόμορφη κατανομή (uniform_real_distribution) στο διάστημα [0,10].

Για περαιτέρω λεπτομέρειες, απευθυνθείτε στο source file main.cpp, το οποίο είναι λεπτομερώς σχολιασμένο. Η εκτέλεση, καθώς δεν χρησιμοποιεί παράλληλα threads / CUDA, διαρκεί περίπου 20' για όλα τα ζητούμενα στον i7 8^{ης} γενιάς επεξεργαστή στον οποίο έγινε το testing.

Ωστόσο παραδείγματα εξόδων (8s .wav files + .txt file με το αποτέλεσμα της συνέλιξης A*B του A μέρους) που εξήχθησαν από την εκτέλεση του προγράμματος παρατίθενται στο παραδοτέο, στο out/ folder.