Φόρτωση των δεδομένων

Το Σύνολο Δεδομένων CIFAR-10

Το CIFAR-10 είναι μια βάση δεδομένων με εικόνες για εκπαίδευση νευρωνικών δικτύων. Αποτελείται συνολικά από 60.000 κατηγοριοποιημένες εικόνες σε 10 κλάσεις. Κάθε εικόνα είναι έγχρωμη με διαστάσεις 32x32. Χωρίζεται σε ένα σύνολο 50.000 εικόνων για εκπαίδευση (training data) και 10.000 εικόνες για έλεγχο (test data). CIFAR-10

Τα δεδομένα είναι αποθηκευμένα σε 5 δυαδικά αρχεία, 10.000 εικόνων το καθένα, `data_batch_1.bin, data_batch_2.bin, ..., data_batch_5.bin` και ένα αρχείο με τις εικόνες ελέγχου,`test_batch.bin` . Τα αρχεία έχουν την εξής μορφή:

- κάθε γραμμή του αρχείου περιέχει 3073 byte που αντιστοιχούν στα δεδομένα μίας εικόνας.
- το πρώτο byte είναι ένας αριθμός από το 0 εώς το 9 που αντιστοιχεί στην κατηγορία που ανήκει η εικόνα.
- Τα επόμενα 3072 bytes είναι οι τιμές των πίξελ της εικόνας. Τα πρώτα 1024 αντιστοιχούν στο κόκκινο χρωματικό κανάλι, τα επόμενα 1024 στο πράσινο και τα τελευταία 1024 στο μπλε. Επιπλέον, οι εικόνες είναι αποθηκευμένες γραμμή προς γραμμή (row-major order), δηλαδή τα πρώτα 32 byte είναι η πρώτη γραμμή της εικόνας στο κόκκινο κανάλι.

Συνάρτηση `load_data()`

Φορτώνουμε τις εικόνες του CIFAR-10 σε ένα πίνακα `image[N][IMAGE_PIXEL]`, όπου N είναι ο αριθμός των εικόνων και IMAGE_PIXEL ο αριθμός των πίξελ της κάθε εικόνας. Στη μεταβλητή `DATA_FOLDER` ορίζουμε τη διαδρομή για το φάκελο που βρίσκονται τα αρχεία στον υπολογιστή μας.

Αρχικά βρίσκουμε πόσα αρχεία από τα 5 θα χρειαστούμε με βάση τον αριθμό των εικόνων.

```
int batches = (N/MAX_BATCH_DATA)+1;
int samples = N % MAX_BATCH_DATA;
```

Στο `batches` αποθηκεύονται πόσα αρχεία θα διαβάσουμε, προσθέτουμε 1 γιατί η αρίθμηση ξεκινάει από το 1, και στο `samples` το υπόλοιπο των εικόνων που θα χρειαστεί να διαβάσουμε.

```
int n=0;
    for (int b=1;b<batches;b++){
        //Open file data_batch_'b'.bin

        for (int i = 0; i < MAX_BATCH_DATA; i++) {
        //read 10.000 images from batch-files 1 to 'batches-1'

        n++;
        }
    }
    if(samples!=0){
        //Open file data_batch_'batches'.bin
        for (int i = 0; i < samples; i++) {
            // read 'samples' images from batch
        n++;
```

```
        }
    }
```

Στο πρώτο βρόγχο διαβάζουμε ολόκληρο αρχείο με 10.000 εικόνες, αν το υπόλοιπο της διαίρεσης είναι διάφορο του 0, διαβάζουμε το υπόλοιπο των εικόνων από το επόμενο αρχείο `data_batch_*.bin`. Εάν οι εικόνες είναι λιγότερες από 10.000, η τιμή του `batches` είναι 1, ο πρώτος βρόγχος δεν εκτελείται, και διαβάζονται `samples` εικόνες από το 1ο αρχείο. Η μεταβλητή `n` αναφέρετε στην εικόνα που διαβάζουμε σε κάθε επανάληψη. Αρχικοποιείται έξω από τους δύο βρόγχους και αυξάνεται κάθε φορά που διαβάζεται μία εικόνα.

Η μεταβλητή `file_name` χρησιμοποιείται για να ορίζουμε κάθε φορά το όνομα του αρχείου το οποίο θα διαβάσουμε.

```
    sprintf(file_name, "%s/data_batch_%d.bin", DATA_FOLDER, b);
```

Μέσα σε κάθε βρόγχο ανοίγουμε το αντίστοιχο αρχείο, ελέγχοντας ότι υπάρχει:

```
FILE *fbin = fopen(file_name, "rb");
        if(fbin==NULL){
            printf("File not found. Error reading .bin files.\n");
            exit(EXIT_FAILURE);
        }
```

Διαβάζουμε γραμμή γραμμή το αρχείο, και αποθηκεύουμε τα δεδομένα σε έναν πίνακα `data` μεγέθους 3073 (`LINE_SIZE`). Το data είναι τύπου `uint8_t` που αντιστοιχεί σε ακέραιο χωρίς πρόσημο του ενός byte (unsigned integer).

```
uint8_t data[LINE_SIZE];
    for (int i = 0; i < MAX_BATCH_DATA; i++) {

        size_t bytesRead = fread(data, 1, LINE_SIZE, fbin);

        label[n] = data[0];
        size_t data_i=1;
        for (int j = 0; j < IMAGE_PIXELS && data_i<LINE_SIZE; j++) {
            image[n][j] = (float)data[data_i++]/255.0-0.5;
        }
        n++;
    }
```

Όπως είπαμε και παραπάνω, το πρώτο byte της γραμμής αντιστοιχεί στην κλάση, γι αυτό και το αποθηκεύουμε στον πίνακα `label`. Στη συνέχεια κανονικοποιούμε τα δεδομένα γύρω από το μηδέν, και τα αποθηκεύουμε στον πίνακα `image`.