Παράλληλος Προγραμματισμός σε Συστήματα Μηχανικής Μάθησης

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Εργαστήριο 4:

Εκπαίδευση νευρωνικών δικτύων με τον αλγόριθμο οπισθοδρομικής διάδοσης του σφάλματος (Error backpropagation)

Δασούλας Ιωάννης – 1053711 – 5ο Έτος

Εισαγωγή

Η αναφορά αφορά την υλοποίηση τεσσάρων εργασιών για την εκπαίδευση νευρωνικών δικτύων με τον αλγόριθμο της οπισθοδρομικής διάδοσης του σφάλματος. Για κάθε εργασία έχει δημιουργηθεί ένα ξεχωριστό αρχείο c (nn1.c για την Εργασία 1, nn2.c για την Εργασία 2 και ούτω καθ' εξής). Αναλυτικά εξηγήσεις και σχόλια για τα βήματα που ακολουθούνται υπάρχουν σε κάθε αρχείο, στα αγγλικά γιατί οι ελληνικοί χαρακτήρες δεν διαβάζονται από όλους τους compilers. Οι μετρήσεις χρόνου έγιναν σε προσωπικό υπολογιστή καθώς το μηχάνημα ήταν εκτός λειτουργίας, οπότε οι αλγόριθμοι είναι λίγο πιο αργοί.

Εργασία 1

Στο πρώτο πρόγραμμα, αρχικά δημιουργούνται οι απαραίτητοι πίνακες για αποθήκευση των εισόδων, των βαρών, των σταθερών (biases), των εσωτερικών καταστάσεων των νευρώνων και των εξόδων των νευρόνων.

Έπειτα, αρχικοποιείται το διάνυσμα εισόδου με αριθμούς από -1 έως 1 και τα διανύσματα με τα βάρη και τα biases με αριθμού από 0 έως 0.1. Επιλέχτηκαν μικρά βάρη για να μην δημιουργηθεί πρόβλημα στους υπολογισμούς με τη συνάρτηση ενεργοποίησης sigmoid, καθώς για μεγάλα βάρη δεν είναι έγκυρα τα αποτελέσματα λόγω της μορφής της sigmoid.

Στην συνέχεια καλείται η συνάρτηση ActivateNN που παίρνει ως όρισμα το διάνυσμα εισόδου. Για τα δύο επίπεδα υπολογίζει όλες τις ενδιάμεσες καταστάσεις και έπειτα μέσω της συνάρτησης sigmoid, την έξοδο των νευρώνων.

Η ενδιάμεση κατάσταση z υπολογίζεται με τον τύπο:

$$z = x.w + b$$

όπου x το διάνυσμα εισόδου, w, το διάνυσμα με τα βάρη και b το bias. Η έξοδος του νευρώνα υπολογίζεται με τον τύπο της συνάρτησης sigmoid:

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Εργασία 2

Στην εργασία 2 προστέθηκε ο πίνακας desired 10 θέσεων που περιέχει τις επιθυμητές τιμές του δικτύου. Επιλέχτηκαν 10 τυχαίες τιμές στο [0, 1] που είναι το πεδίο τιμών της sigmoid.

Στην συνέχεια προστέθηκε η συνάρτηση TrainNN που παίρνει ως όρισμα το διάνυσμα των τιμών εισόδου και το διάνυσμα των επιθυμητών τιμών εξόδου. Για το ορισμένο αριθμό επαναλήψεων, αρχικά ενεργοποιείται το νευρωνικό δίκτυο και υπολογίζονται οι έξοδοι για να ακολουθήσει ο αλγόριθμος οπισθοδρομικής διάδοσης λάθους. Για κάθε έξοδο του νευρώνα εξόδου, υπολογίζεται το σφάλμα, δηλαδή η διαφορά με την επιθυμητή τιμή και πολλαπλασιάζεται με την παράγωγο της συνάρτησης sigmoid για την έξοδο, με τον τύπο:

error2(i) = (desired – output) * derivative(output(i)), όπου i νευρώνας του δεύτερου επιπέδου

Αντίστοιχα, για το σφάλμα του πρώτου επιπέδου:

error1(j) = Sum(error2(j)*weights(i, j)) * derivative(output(j)))), όπου i νευρώνας του δεύτερου επιπέδου και j νευρώνας του πρώτου.

Τα biases κάθε επιπέδου ενημερώνονται με τον τύπο:
bias += error*learning_rate

Τα βάρη του δεύτερου επιπέδου ενημερώνονται με τον τύπο:

weights(i, j) += output(j)*error2(i)*learning_rate

Αντίστοιχα, τα βάρη του πρώτου επιπέδου ενημερώνονται με τον τύπο:

weights(k, j) += input(k)*error1(j)*learning_rate, όπου k ο αριθμός της εισόδου.

Το learning rate ορίστηκε ίσο με 0.1 αλλά το πρόγραμμα είναι παραμετροποιήσιμο. Αυτή η διαδικασία επαναλαμβάνεται για κάθε επανάληψη. Στο τέλος της επανάληψης τυπώνεται το άθροισμα των απόλυτων τιμών των διαφορών των εξόδων με τις επιθυμητές εξόδους.

Για 1000 επαναλήψεις παρατηρείται το σφάλμα μειώνεται συνέχεια. Στις πρώτες επαναλήψεις με γρήγορο ρυθμό, όπως φαίνεται στην εικόνα:

```
Σφάλμα εκπαίδευσης επανάληψης 1: 5.184469
Σφάλμα εκπαίδευσης επανάληψης 2: 5.123297
Σφάλμα εκπαίδευσης επανάληψης 3: 5.051526
Σφάλμα εκπαίδευσης επανάληψης 4: 4.966401
Σφάλμα εκπαίδευσης επανάληψης 5: 4.864374
Σφάλμα εκπαίδευσης επανάληψης 6: 4.741059
Σφάλμα εκπαίδευσης επανάληψης 7: 4.591483
Σφάλμα εκπαίδευσης επανάληψης 8: 4.411005
Σφάλμα εκπαίδευσης επανάληψης 9: 4.197153
Σφάλμα εκπαίδευσης επανάληψης 10: 3.951868
Σφάλμα εκπαίδευσης επανάληψης 11: 3.682535
Σφάλμα εκπαίδευσης επανάληψης 12: 3.400467
Σφάλμα εκπαίδευσης επανάληψης 13: 3.117388
Σφάλμα εκπαίδευσης επανάληψης 14: 2.841934
Σφάλμα εκπαίδευσης επανάληψης 15: 2.578577
Σφάλμα εκπαίδευσης επανάληψης 16: 2.329161
Σφάλμα εκπαίδευσης επανάληψης 17: 2.094740
Σφάλμα εκπαίδευσης επανάληψης 18: 1.876259
Σφάλμα εκπαίδευσης επανάληψης 19: 1.674424
Σφάλμα εκπαίδευσης επανάληψης 20: 1.489509
Σφάλμα εκπαίδευσης επανάληψης 21: 1.321365
Σφάλμα εκπαίδευσης επανάληψης 22: 1.169562
Σφάλμα εκπαίδευσης επανάληψης 23: 1.033478
```

Στις 272 επαναλήψεις το σφάλμα μηδενίζεται και η εκπαίδευση θεωρείται επιτυχής, όπως φαίνεται στην εικόνα:

```
Σφάλμα εκπαίδευσης επανάληψης 259: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 260: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 261: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 262: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 263: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 264: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 265: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 266: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 267: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 268: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 269: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 270: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 271: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 272: 0.000001
Σφάλμα εκπαίδευσης επανάληψης 273: 0.000000
Σφάλμα εκπαίδευσης επανάληψης 274: 0.000000
Σφάλμα εκπαίδευσης επανάληψης 275: 0.000000
Σφάλμα εκπαίδευσης επανάληψης 276: 0.000000
Σφάλμα εκπαίδευσης επανάληψης 277: 0.000000
Σφάλμα εκπαίδευσης επανάληψης 278: 0.000000
Σφάλμα εκπαίδευσης επανάληψης 279: 0.000000
Σφάλμα εκπαίδευσης επανάληψης 280: 0.000000
Σφάλμα εκπαίδευσης επανάληψης 281: 0.000000
Σφάλμα εκπαίδευσης επανάληψης 282: 0.000000
```

Εργασία 3

Για την εργασία 3 προστέθηκαν εντολές openMP στις διαδικασίες της ενεργοποίησης και της διάδοσης του σφάλματος του νευρωνικού δικτύου. Οι εντολές αφορούσαν τους βρόγχους υπολογισμού, ενώ χρησιμοποιήθηκαν και εντολές simd μιας και οι περισσότεροι υπολογισμοί αφορούσαν τιμές πινάκων.

Παρόλα αυτά, η μορφή του προβλήματος είναι τέτοια ώστε να μην ενδείκνυται για παραλληλοποίηση, αφού ο αλγόριθμος απαρτίζεται από πολλές επαναλήψεις εκπαίδευσης, αλλά κάθε επανάληψη εμπεριέχει πολλούς μικρούς βρόγχους υπολογισμού, λόγω των μικρών αριθμών νευρώνων. Ως αποτέλεσμα, καλούνται πολλές φορές οι δηλώσεις της openMP, γεγονός που απαιτεί κάποιον χρόνο, αλλά έχουν μικρό σχετικά έργο, κι έτσι δεν υπάρχει ουσιαστική βελτίωση. Δοκιμάστηκαν και άλλες τεχνικές όπως παράλληλα sections, αλλά πάλι δεν υπήρχε ικανοποιητική βελτίωση.

```
#pragma omp parallel for private(error) reduction(+: error_sum)
for(i=0; i<N2; i++){                                //For every output
    error = (desired[i] - OL2[i]);                  //Error calculation
    if(error>0)
        error_sum += error;
    else
        error_sum += -error;
    delta2[i] = error * DerSigmoid(OL2[i]);         //Calculation of delta array for output layer
}

#pragma omp parallel for private(error)
for(i=0; i<N1; i++){                                //Back propagation to the hidden layer
    error = 0;
    #pragma omp simd reduction(+: error)
    for(j=0; j<N2; j++){
        error += delta2[j]*weights2[i][j];          //Error back propagation
    }
    delta1[i] = error*DerSigmoid(OL1[i]);           //Calculation of delta array for hidden layer
}

#pragma omp parallel for
for(i=0; i<N2; i++){                                //Weight updating for ouput layer
    biases2[i] += delta2[i]*LR;                     //Biases updating
    #pragma omp simd
    for(j=0; j<N1; j++){
        weights2[j][i] += OL1[j]*delta2[i]*LR;      //Weight updating
    }
}
```

Εργασία 4

Στην εργασία 4 χρησιμοποιήθηκε το δίκτυο για κατηγοριοποίηση MNIST δεδομένων. Αρχικά, προστέθηκαν οι συναρτήσεις ανάγνωσης των δύο MNIST αρχείων, με τα δεδομένα εκπαίδευσης και αξιολόγησης αντίστοιχα. Οι συναρτήσεις ανοίγουν τα αρχεία, και αποθηκεύουν τις κλάσεις των δεδομένων καθώς και τα pixels κάθε εικόνας, κανονικοποιημένα, για να γίνονται ευκολότερα και πιο αποτελεσματικά οι υπολογισμοί. Η κανονικοποίηση γίνεται διαιρώντας το κάθε pixel με το 255. Με αυτόν τον τρόπο όλες οι τιμές των pixel είναι από 0 έως 1, διευκολύνοντας έτσι τους υπολογισμούς.

Επίσης, προστέθηκαν οι συναρτήσεις ReLu και Softmax. Η ReLu χρησιμοποιείται στο πρώτο κρυφό επίπεδο καθώς, αυξάνει την απόδοση σε σχέση με την Sigmoid για το συγκεκριμένο πρόβλημα μιας και οι τιμές που βγάζει δεν περιορίζονται στο [0,1] κι έτσι αποφεύγονται οι λάθος υπολογισμοί που μπορεί να συμβούν για μεγάλους αριθμούς πολλές φορές. Αντίστοιχα, η Softmax χρησιμοποιείται στο επίπεδο εξόδου, μιας και είναι η κατεξοχήν συνάρτηση για προβλήματα classification με πάνω από δύο κλάσεις, όπου το πρόβλημα δηλαδή δεν είναι δυαδικό. Με την Softmax κάθε έξοδος αντιμετωπίζεται ως πιθανότητα η είσοδος που δόθηκε να ανήκει σε μία από τις 10 κλάσεις. Το σύνολο των 10 εξόδων είναι ίσο με 1. Η έξοδος με την μεγαλύτερη τιμή αναπαριστά την μεγαλύτερη πιθανότητα να ανήκει το δείγμα στην αντίστοιχη κλάση. Έτσι ο πίνακας desired αποτελείται από μηδενικά κι έναν άσσο στην επιθυμητή κλάση, όταν γίνεται η εκπαίδευση του δικτύου. Στην αξιολόγηση μετριούνται οι πιθανότητες και αντιστοιχούνται σε μία κλάση. Για να υλοποιηθεί, άλλαξαν οι μαθηματικές σχέσεις στον error backpropagation όπως φαίνεται στο πρόγραμμα, καθώς και η συνάρτηση κόστους. Οι μαθηματικές σχέσεις γράφτηκαν μέσω της πηγής: The Softmax function and its derivative - Eli Bendersky's website (thegreenplace.net). Επίσης, χρησιμοποιήθηκε και η παράγωγος της ReLu στο πρώτο επίπεδο για την ανανέωση των δικών της βαρών.

Αφού φορτώνονται τα δεδομένα εκπαίδευσης, ξεκινάει η εκπαίδευση. Επειδή, παρατηρήθηκαν κάποια προβλήματα στην τυχαία επιλογή δειγμάτων για την εκπαίδευση συγκεκριμένα, η εκπαίδευση έγινε και με τα 60.000 δείγματα σε 100 εποχές, συνολικά δηλαδή 6.000.000 φορές. Μετά από κάθε επανάληψη υπολογίζεται η επιτυχία αναγνώρισης συγκρίνοντας την κλάση του αρχείου με την αντίστοιχη μεγαλύτερη πιθανότητα κλάσης που το δίκτυο δίνει.

Με αντίστοιχο τρόπο έγινε η αξιολόγηση, αυτή τη φορά με τυχαία δείγματα. Αφού φορτώνονται τα 10.000 τυχαία δείγματα, επιλέγονται τυχαία δείγματα εξ αυτών (αριθμός δειγμάτων ρυθμίζεται στις παραμέτρους, στο παραδοτέο τέθηκε ίσος με 10.000) και για κάθε δείγμα υπολογίζεται η επιτυχία αναγνώρισης.

Και για την εκπαίδευση και για την αξιολόγηση, οι επιτυχίες αθροίζονται και διαιρούνται με τον συνολικό αριθμό δειγμάτων, για να δημιουργηθεί το ποσοστό επιτυχίας της εκπαίδευσης και της αξιολόγησης, αντίστοιχα.

Στην εικόνα φαίνονται τα αποτελέσματα για 6.000.000 δείγματα εκπαίδευσης και 10.000 τυχαία δείγματα αξιολόγησης. Τα ποσοστά επιτυχίας επαληθεύουν τη σωστή λειτουργία και είναι αρκετά ικανοποιητικά. Χρησιμοποιήθηκε οδηγία βελτιστοποίησης -O3. Επίσης, δεν χρησιμοποιήθηκαν οι εντολές openMP του 3ου ερωτήματος, αλλά οι χρόνοι είναι αρκετά καλοί. Παρατηρήθηκαν και καλύτερα ποσοστά επιτυχίας από τα εικονιζόμενα για διαφορετικά μεγέθη στα βάρη και στον συντελεστή εκπαίδευσης, μεγέθη τα οποία μπορούν να επηρεάσουν το τελικό αποτέλεσμα. Παρατηρήθηκε και επιτυχία πάνω από 90%. Στην συγκεκριμένη εκτέλεση είναι 80.2%.