Υπολογιστική Νοημοσύνη
Εργαστηριακές Ασκήσεις ακ. έτους 2018-19

Να κατασκευάσετε σύνολα δεδομένων για τα ακόλουθα προβλήματα:
Σ1) **Πρόβλημα ταξινόμησης τριών κατηγοριών**: Θα δημιουργήσετε 6000 παραδείγματα (σημεία (x_1,x_2) στο επίπεδο) ως εξής: 3000 τυχαία σημεία μέσα στο τετράγωνο [0,2]x[0,2] (1500 για εκπαίδευση και 1500 για έλεγχο) και 3000 τυχαία σημεία μέσα στο τετράγωνο [-2,0]x[-2,0] (1500 για εκπαίδευση και 1500 για έλεγχο). Ετσι προκύπτει ένα σύνολο εκπαίδευσης με 3000 παραδείγματα και ένα σύνολο ελέγχου με 3000 παραδείγματα. Στη συνέχεια κάθε παράδειγμα (x_1,x_2) (από τα 6000 παραδείγματα) κατατάσσεται σε μια κατηγορία ως εξής:
1) εάν $(x_1-1)^2+(x_2-1)^2<=0.16$, τότε το (x_1,x_2) κατατάσσεται στην κατηγορία C1,
2) εάν $(x_1+1)^2+(x_2+1)^2<=0.16$, τότε το (x_1,x_2) κατατάσσεται στην κατηγορία C1,
3) εάν $(x_1-1)^2+(x_2-1)^2>0.16$ και $(x_1-1)^2+(x_2-1)^2<0.64$ τότε το (x_1,x_2) κατατάσσεται στην κατηγορία C2,
4) εάν $(x_1+1)^2+(x_2+1)^2>0.16$ και $(x_1+1)^2+(x_2+1)^2<0.64$ τότε το (x_1,x_2) κατατάσσεται στην κατηγορία C2
3) αλλιώς κατατάσσεται στην κατηγορία C3.

Σ2) **Πρόβλημα ομαδοποίησης πέντε ομάδων** (500 παραδείγματα): δημιουργούμε **τυχαία** α) 100 σημεία στο τετράγωνο [-0.3,0.3]x[-0.3,0.3], β) 100 σημεία στο τετράγωνο [-1.1,-0.5]x[0.5,1.1], γ) 100 σημεία στο τετράγωνο [-1.1,-0.5] x [-1.1,-0.5], δ) 100 σημεία στο τετράγωνο [0.5,1.1] x [-1.1,-0.5], ε) 100 σημεία στο τετράγωνο [0.5,1.1] x [0.5,1.1].

Στη συνέχεια να κατασκευάσετε:

Π1) **Πρόγραμμα ταξινόμησης** βασισμένο στο **πολυεπίπεδο perceptron** (MLP) με **δύο κρυμμένα επίπεδα** με **κρυμμένους νευρώνες που έχουν** συνάρτηση ενεργοποίησης τη λογιστική (σ(u)) ή την υπερβολική εφαπτομένη (tanh(u)) και νευρώνες εξόδου που έχουν λογιστική συνάρτηση ενεργοποίησης. Το πρόγραμμα θα πρέπει να αποτελείται από τις ακόλουθες μονάδες:

1) Με χρήση της εντολής define, καθορισμός αριθμού εισόδων (d), αριθμού κατηγοριών (K), αριθμού νευρώνων στο πρώτο κρυμμένο επίπεδο (H1), αριθμού νευρώνων στο δεύτερο κρυμμένο επίπεδο (H2) και είδος συνάρτησης ενεργοποίησης (λογιστική ή tanh) για όλους τους κρυμμένους νευρώνες.
2) Φόρτωμα των συνόλων εκπαίδευσης και ελέγχου (από αντίστοιχα αρχεία) και κωδικοποίηση των κατηγοριών (ορισμός των επιθυμητών εξόδων για κάθε κατηγορία).
3) Καθορισμός της αρχιτεκτονικής του δικτύου MLP. Ορισμός των απαιτούμενων πινάκων και άλλων δομών ως καθολικών μεταβλητών.
4) Υλοποίηση της συνάρτησης forward-pass (float *x, int d, float *y, int K) η οποία υπολογίζει το διάνυσμα εξόδου y (διάστασης K) του MLP δοθέντος του διανύσματος εισόδου x (διάστασης d).
5) Υλοποίηση της συνάρτησης backprop(float *x, int d, float *t, int K) η οποία λαμβάνει τα διανύσματα x διάστασης d (είσοδος) και t διάστασης K (επιθυμητή έξοδος) και υπολογίζει τις παραγώγους του σφάλματος ως προς οποιαδήποτε παράμετρο (βάρος ή πόλωση) του δικτύου ενημερώνοντας τους αντίστοιχους πίνακες.
6) Χρησιμοποιώντας τα παραπάνω να υλοποιήσετε τον **αλγόριθμο εκπαίδευσης gradient descent με ενημέρωση ανά ομάδες των L παραδειγμάτων (mini-batches) θεωρώντας τα N παραδείγματα του συνόλου εκπαίδευσης** (όπου το L διαιρέτης του N και ορίζεται στην αρχή του προγράμματος). Σημειώστε ότι εάν L=1 έχουμε σειριακή ενημέρωση, ενώ εάν L=N έχουμε ομαδική ενημέρωση. Στο τέλος **κάθε εποχής** θα πρέπει υποχρεωτικά να υπολογίζετε και να τυπώνετε την τιμή του συνολικού τετραγωνικού σφάλματος εκπαίδευσης. Ως κριτήριο τερματισμού θεωρούμε τη διαφορά της τιμής του σφάλματος εκπαίδευσης μεταξύ δύο εποχών, αφού ο αλγόριθμος έχει τρέξει για τουλάχιστον 500 εποχές.
7) Αφού τερματιστεί η εκπαίδευση του δικτύου να γίνεται μέτρηση της ικανότητας γενίκευσης του δικτύου που προκύπτει, χρησιμοποιώντας το σύνολο ελέγχου.

Χρησιμοποιώντας το πρόγραμμα (Π1) να μελετήσετε το πρόβλημα (Σ1). Να εξετάσετε πώς μεταβάλλεται η γενικευτική ικανότητα του δικτύου (ποσοστό επιτυχίας στο σύνολο ελέγχου) θεωρώντας τους συνδυασμούς τιμών (H1,H2) π.χ.{(5,3),(7,4),(8,5)}, για κρυμμένους νευρώνες με συνάρτηση ενεργοποίησης λογιστική ή tanh και για τιμές του L={1, N/10, N} (συνολικά 3x2x3=18 περιπτώσεις). Για το δίκτυο με την καλύτερη γενικευτική ικανότητα που θα βρείτε, να τυπώσετε τα παραδείγματα του συνόλου ελέγχου χρησιμοποιώντας διαφορετικό στυλ (πχ + και -) ανάλογα με το αν το παράδειγμα ταξινομείται από το δίκτυο στη σωστή κατηγορία ή όχι.

Π2) **Πρόγραμμα ομαδοποίησης** με M ομάδες (το M θα ορίζεται με την εντολή #define) βασισμένο στον **αλγόριθμο k-means**. Το πρόγραμμα θα φορτώνει το αρχείο με τα παραδείγματα, θα εκτελεί τον αλγόριθμο k-means με M κέντρα και στο τέλος θα αποθηκεύει τις συντεταγμένες των κέντρων των ομάδων. Επίσης **θα πρέπει στο τέλος να υπολογίζεται η συνολική διασπορά των ομάδων**. Η αρχική θέση κάθε κέντρου να γίνεται επιλέγοντας τυχαία κάποιο από τα παραδείγματα.

Π3) **Πρόγραμμα ομαδοποίησης** με M ομάδες (το M θα ορίζεται με την εντολή #define) βασισμένο στον **αλγόριθμο LVQ** για ομαδοποίηση. Το πρόγραμμα θα φορτώνει το αρχείο με τα παραδείγματα, θα εκτελεί τον αλγόριθμο LVQ με M κέντρα και στο τέλος θα αποθηκεύει τις συντεταγμένες των κέντρων των ομάδων. Επίσης θα **πρέπει στο τέλος να υπολογίζεται η συνολική διασπορά των ομάδων.** Η αρχική θέση κάθε κέντρου να γίνεται επιλέγοντας τυχαία κάποιο από τα παραδείγματα. Ο ρυθμός μάθησης n να μειώνεται στο τέλος κάθε **εποχής** (π.χ. n(t+1)=0.95 n(t)) ξεκινώντας από μια κατάλληλη **αρχική τιμή** (π.χ. n=0.1).

Να εφαρμόσετε τα προγράμματα Π2 και Π3 στο σύνολο δεδομένων (Σ2) για M=2,3,5,7 ομάδες.

Για **κάθε** πρόγραμμα Π2 ή Π3 και για **κάθε** τιμή του M να κάνετε τα εξής:
α) Να εκτελέσετε 5 τρεξίματα του προγράμματος και να κρατήσετε τη **λύση με τη μικρότερη διασπορά**. β) Στη συνέχεια να εμφανίσετε (plot) στο ίδιο σχήμα τα παραδείγματα (π.χ. με '+') και τις θέσεις των κέντρων που βρήκατε (π.χ. με '*').