



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

## Παράλληλες Αρχιτεκτονικές Υπολογισμού για Μηχανική Μάθηση

Ε.ΔΕ.Μ<sup>2</sup>

Ακαδημαϊκό Έτος 2019-20

<http://www.cslab.ece.ntua.gr/courses/parml>

### ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1

#### Εισαγωγή στην Παράλληλη Επεξεργασία

#### 1 Σκοπός της Άσκησης

Στα πλαίσια αυτής της εργαστηριακής άσκησης θα εξοικειωθείτε με βασικές έννοιες της παράλληλης επεξεργασίας μέσα από την επίλυση ενός προβλήματος που μπορεί να αξιοποιήσει τεχνικές μηχανικής μάθησης. Συγκεκριμένα, θα καταπιαστείτε με την αναγνώριση χειρόγραφων αριθμητικών ψηφίων, όπου και θα εκπαιδευστείτε Νευρωνικά Δίκτυα (Neural Networks) για την επίλυσή του. Ως δεδομένα εκπαίδευσης για τα μοντέλα σας, θα χρησιμοποιήσετε τη βάση δεδομένων MNIST, η οποία περιλαμβάνει 70000 εικόνες χειρόγραφων αριθμητικών ψηφίων με 784 χαρακτηριστικά.

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 6 | 6 | 8 | 3 | 6 | 8 | 9 | 4 |
| 2 | 2 | 0 | 2 | 8 | 5 | 0 | 5 | 5 | 7 |
| 6 | 3 | 8 | 8 | 0 | 1 | 5 | 4 | 1 | 5 |
| 2 | 1 | 9 | 8 | 0 | 3 | 3 | 6 | 4 | 1 |
| 7 | 9 | 1 | 4 | 9 | 9 | 2 | 4 | 5 | 1 |
| 3 | 7 | 3 | 9 | 3 | 6 | 7 | 2 | 4 | 3 |
| 3 | 5 | 1 | 9 | 7 | 4 | 4 | 3 | 4 | 9 |
| 0 | 1 | 6 | 0 | 5 | 2 | 8 | 0 | 5 | 7 |
| 5 | 6 | 7 | 2 | 9 | 7 | 0 | 2 | 8 | 9 |
| 0 | 4 | 7 | 1 | 2 | 6 | 6 | 0 | 7 | 0 |

Σχήμα 1: Χειρόγραφα αριθμητικά ψηφία

Για την εκπαίδευση των μοντέλων, θα χρησιμοποιήσετε το εργαλείο μηχανικής μάθησης `scikit-learn`. Σας δίνονται έτοιμα `script` σε `Bash` και `Python` για την εκτέλεση των πειραμάτων σας, τα οποία θα βρείτε στους κατάλογο `/home/parml/shared/lab1/mnist`. Για την ενότητα αυτή, θα χρησιμοποιήσετε τα μηχανήματα `clones`, που ανήκουν στην ουρά `parlab` (δείτε τον Εργαστηριακό Οδηγό).

Για κάθε πρόβλημα, θα εξετάσουμε διάφορους τρόπους παραλληλοποίησης των πειραμάτων υποθέτοντας κάθε φορά ότι διαθέτουμε διαφορετικές υπολογιστικές υποδομές. Προκειμένου ωστόσο να

κατανοήσετε τα πλεονεκτήματα της παραλληλοποίησης, θα πρέπει πρώτα να εκτελείτε σειριακά τα πειράματά σας και να καταγράφετε το συνολικό χρόνο εκτέλεσης. Συγκρίνοντας τον χρόνο της σειριακής εκτέλεσης με αυτόν της παράλληλης, προκύπτει η μετρική της επιτάχυνσης (speedup), η οποία είναι χρήσιμη για την αξιολόγηση της παραλληλοποίησης. Πιο συγκεκριμένα, ορίζουμε ως επιτάχυνση  $S$  το λόγο του σειριακού χρόνου εκτέλεσης  $T_s$  προς το παράλληλο χρόνο εκτέλεσης  $T_p$  με  $P$  μονάδες επεξεργασίας:

$$S = \frac{T_s}{T_p} \quad (1)$$

Θα χρησιμοποιήσουμε, επιπλέον, την μετρική της αποδοτικότητας (efficiency)  $E$  η οποία δείχνει πόσο επιτυχημένη είναι η παραλληλοποίηση – τι ποσοστό του χρόνου κάθε επεξεργαστής κάνει χρήσιμη δουλειά – και ορίζεται ως εξής:

$$E = \frac{S}{P} \quad (2)$$

Τέλος, ο νόμος του Amdahl μας βοηθά να εκτιμήσουμε θεωρητικά τη μέγιστη επιτάχυνση που μπορούμε να επιτύχουμε για ένα πρόγραμμα. Αν  $f$  είναι το κλάσμα του χρόνου ενός προγράμματος που δεν παραλληλοποιείται τότε:

$$T_p = f \cdot T_s + \frac{(1-f) \cdot T_s}{P} \quad (3)$$

$$S = \frac{T_s}{T_p} = \frac{1}{f + \frac{(1-f)}{P}} \quad (4)$$

## 2 Αναγνώριση χειρόγραφων αριθμητικών ψηφίων

Έστω ότι έχετε αποφασίσει να εκτελέσετε συνολικά 8 ανεξάρτητα πειράματα εκπαίδευσης Νευρωνικών Δικτύων για να βρείτε αυτό με την υψηλότερη ακρίβεια.

### 2.1 Παραλληλοποίηση εργασιών με τη χρήση ενός υπολογιστικού κόμβου

Έστω ότι διαθέτετε έναν υπολογιστικό κόμβο που διαθέτει δύο CPUs με 4 πυρήνες (αρχιτεκτονική κοινής μνήμης UMA). Σε αυτή την περίπτωση, εφόσον τα πειράματά σας είναι ανεξάρτητα μπορείτε να εκτελέσετε κάθε πείραμα σε διαφορετικό πυρήνα. Προκειμένου να εξασφαλιστεί ότι κάθε πείραμα εκτελείται σε διαφορετικό πυρήνα, θα γίνει χρήση του εργαλείου taskset. Το εργαλείο αυτό μας επιτρέπει να ορίσουμε ποιος πυρήνας θα εκτελέσει το πρόγραμμά μας.

1. Εκτελέστε όλα τα πειράματα σε έναν πυρήνα ενός κόμβου μέσω της εντολής:

```
$ qsub -q parlab -l nodes=1:ppn=1 run_all_serial.sh
```

και καταγράψτε το συνολικό χρόνο εκτέλεσης της εκπαίδευσης των πειραμάτων με βάση το αρχείο εξόδου mnist\_run\_all\_serial.out. Ο συνολικός χρόνος εκτέλεσης που αφορά αποκλειστικά την εκπαίδευση (εντολή mlp.fit(X\_train, y\_train)) είναι το άθροισμα των επιμέρους χρόνων εκπαίδευσης κάθε πειράματος.

2. Εκτελέστε κάθε πείραμα σε διαφορετικό πυρήνα του ίδιου κόμβου αξιοποιώντας το σύστημα Torque μέσω της εντολής:

```
$ qsub -q parlab -l nodes=1:ppn=8:highmem run_all_parallel_torque.sh
```

και καταγράψτε το χρόνο εκτέλεσης κάθε πειράματος με βάση το αρχείο εξόδου mnist\_run\_all\_parallel\_torque.out. Πώς συγκρίνονται με τους αντίστοιχους χρόνους σειριακής εκτέλεσης από το Βήμα 1;

3. Εκτελέστε κάθε πείραμα σε διαφορετικό πυρήνα του ίδιου κόμβου αξιοποιώντας το multiprocessing module της Python μέσω της εντολής:

```
$ qsub -q parlab -l nodes=1:ppn=8 run_all_parallel_in_python.sh
```

και καταγράψτε το συνολικό χρόνο εκτέλεσης με βάση το αρχείο εξόδου mnist\_run\_all\_parallel\_in\_python.out. Πώς συγκρίνεται ο συνολικός χρόνος εκτέλεσης σε σχέση με τον χρόνο εκτέλεσης των επιμέρους πειραμάτων; Επιπλέον, συγκρίνετέ τον χρόνο εκτέλεσης κάθε πειράματος με τον αντίστοιχο χρόνο του στη σειριακή εκτέλεση από το Βήμα 1, αλλά και με τον αντίστοιχο χρόνο από το Βήμα 2. Παρατηρείτε κάποια διαφορά; Υπολογίστε την επιτάχυνση. Επιτύχατε τη θεωρητικά αναμενόμενη επιτάχυνση;

## 2.2 Παραλληλοποίηση υπολογισμών ανά εργασία με τη χρήση ενός υπολογιστικού κόμβου

Έστω ότι διαθέτετε έναν υπολογιστικό κόμβο που διαθέτει δύο CPUs με 4 πυρήνες (αρχιτεκτονική κοινής μνήμης UMA) και ένα υποσύνολο των υπολογισμών που εκτελούνται κατά την εκπαίδευση μοντέλων είναι “embarrassingly parallel”. Σε αυτή την περίπτωση, μπορούμε να αξιοποιήσουμε τους πολλαπλούς πυρήνες για να επιταχύνουμε τους υπολογισμούς με τη χρήση πολλαπλών νημάτων εκτέλεσης (threads).

1. Εκτελέστε τον αλγόριθμο SGD αξιοποιώντας αριθμό πυρήνων/νημάτων 1,2,4,8 μέσω των εντολών:

```
$ qsub -q parlab -l nodes=1:ppn=8 run_sgd_multithreaded.sh
```

και καταγράψτε το χρόνο εκτέλεσης κάθε πειράματος με βάση το αντίστοιχο αρχείο εξόδου.

2. Υπολογίστε την απόδοση (efficiency) για κάθε αριθμό νημάτων και εξετάστε την κλιμακωσιμότητα (scalability) του αλγορίθμου εκπαίδευσης του πειράματος. Επιτύχατε τη θεωρητικά αναμενόμενη απόδοση για κάθε αριθμό νημάτων;
3. Μελετήστε τον χρόνο εκτέλεσης όλου του πειράματος σε σχέση με τον χρόνο εκτέλεσης της εκπαίδευσης για κάθε αριθμό νημάτων. Επιτύχατε τη θεωρητικά αναμενόμενη επιτάχυνση για κάθε αριθμό νημάτων; Αν όχι, δικαιολογήστε το φαινόμενο με τη βοήθεια του νόμου του Amdahl.