ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Εισαγωγή στην παράλληλη επεξεργασία με έμφαση στις εφαρμογές μηχανικής μάθησης

Ακαδημαϊκό έτος 2019-20

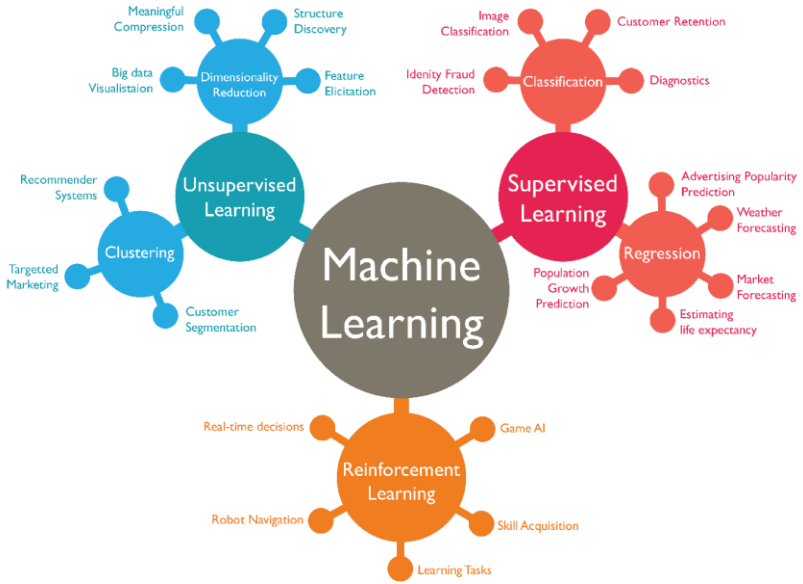
Εισαγωγή

- Μηχανική μάθηση και εφαρμογές
- Μηχανική μάθηση και παράλληλη επεξεργασία
- Εισαγωγή στις παράλληλες αρχιτεκτονικές
- Τάσεις στην παράλληλη επεξεργασία αλγορίθμων βαθιάς μηχανικής μάθησης

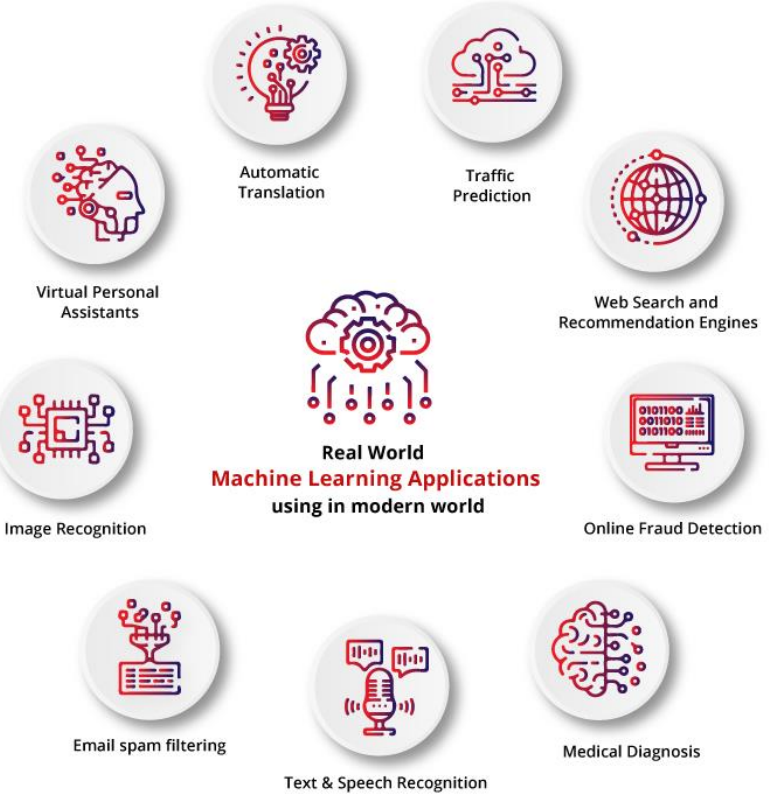
Μηχανική μάθηση και εφαρμογές

Μηχανική μάθηση

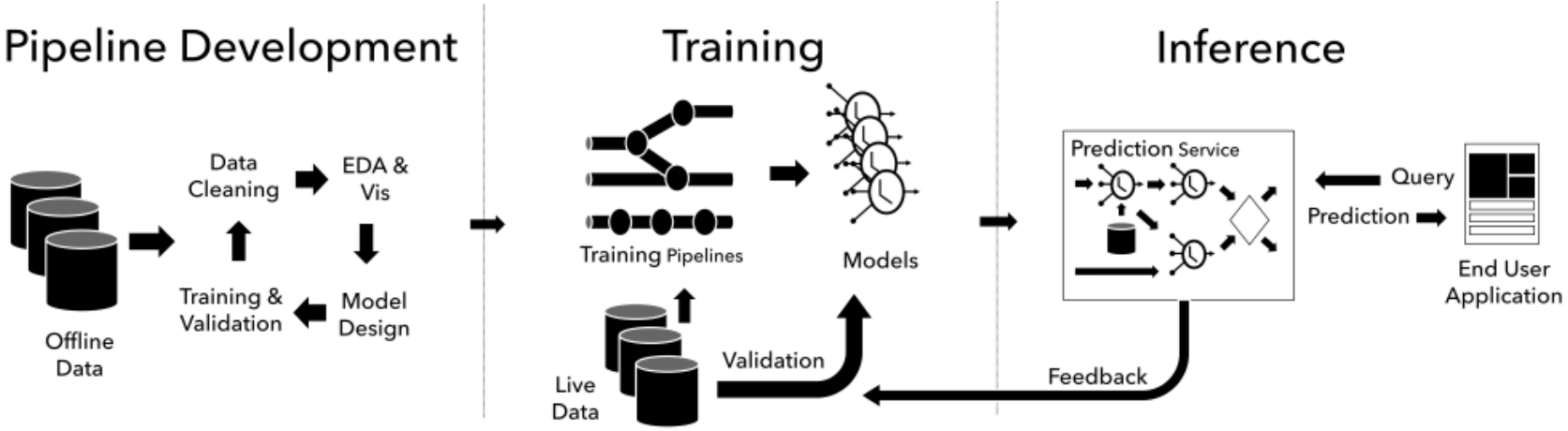
- Διερευνά τη μελέτη και την κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά



Εφαρμογές μηχανικής μάθησης

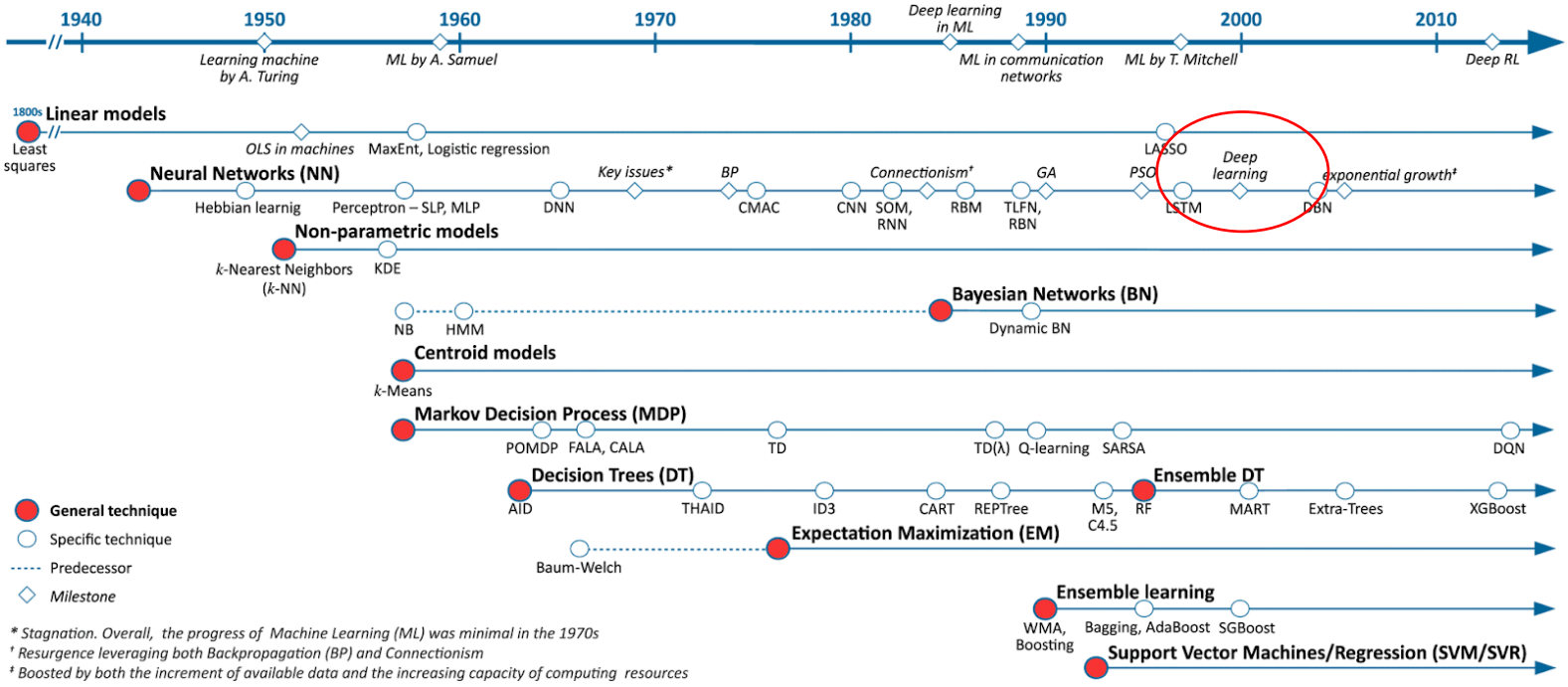


Ανάπτυξη εφαρμογών μηχανικής μάθησης



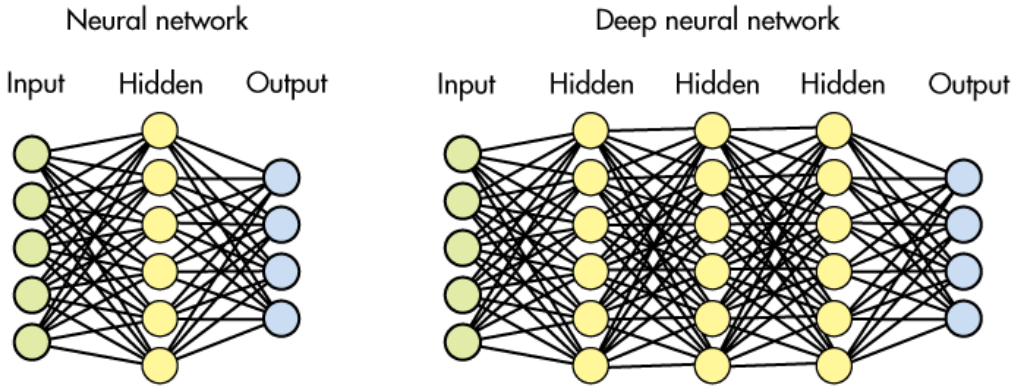
Πηγή: https://rise.cs.berkeley.edu/wp-content/uploads/2018/06/ML_Lifecycle-1.png

Ιστορική αναδρομή τεχνικών μηχανικής μάθησης



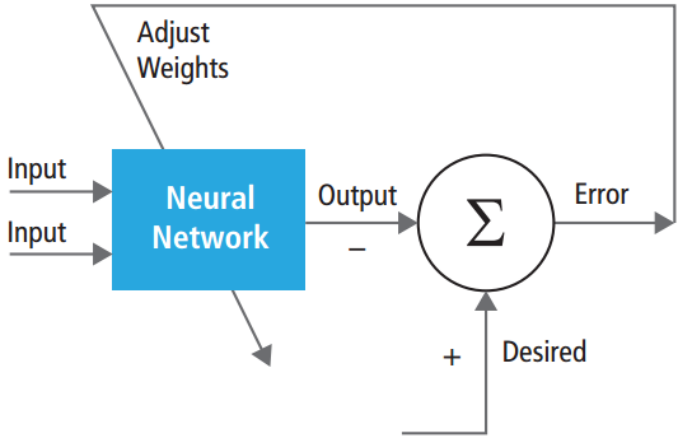
Βαθιά μηχανική μάθηση

- Εκπαίδευση Νευρωνικών Δικτύων με πολλά επίπεδα
- Ιδιαίτερα δημοφιλής τεχνική τα τελευταία χρόνια
  - Υψηλή επίδοση σε πολλές οικογένειες εφαρμογών, π.χ. αναγνώριση εικόνας, ήχου, κλπ.



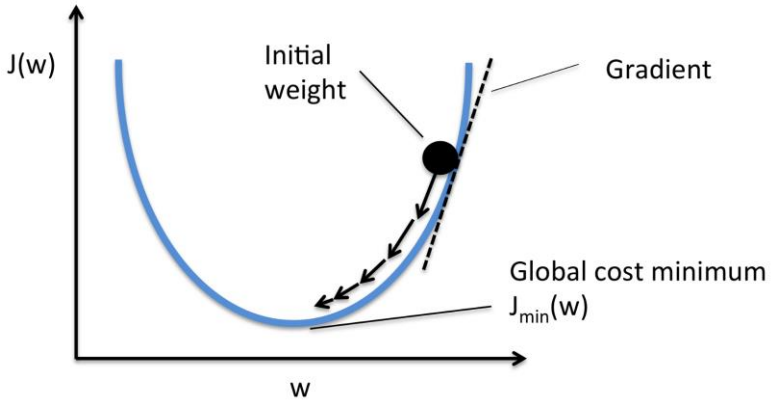
Εκπαίδευση Βαθιών Νευρωνικών Δικτύων

- Κάθε νευρώνας του δικτύου αποτελείται από κάποιες παραμέτρους (weights) οι τιμές των οποίων προσαρμόζονται στην εκάστοτε εφαρμογή
- Στόχος του αλγορίθμου εκπαίδευσης είναι η προσαρμογή των παραμέτρων μέσα από την ελαχιστοποίηση μιας συνάρτησης σφάλματος (cost function)
- Η συνάρτηση σφάλματος είναι μία συνάρτηση των παραμέτρων του δικτύου



Αλγόριθμος Gradient Descent

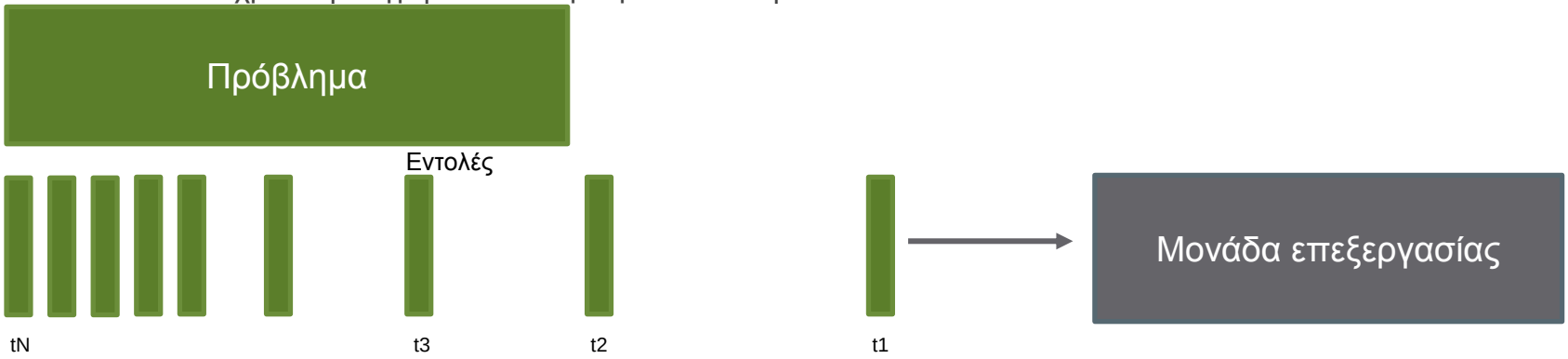
- Επαναληπτικός αλγόριθμος βελτιστοποίησης για την ελαχιστοποίηση συνάρτησης
- Χρησιμοποιείται στην εκπαίδευση νευρωνικών δικτύων για την ελαχιστοποίηση της συνάρτησης σφάλματος
- Σε κάθε επανάληψη:
  1. Υπολογίζει την τιμή του σφάλματος για ένα δεδομένο εισόδου (forward pass) και
  2. Υπολογίζει την παράγωγο της συνάρτησης σφάλματος ως προς κάθε παράμετρο του δικτύου και ενημερώνει τις τιμές των παραμέτρων (backward pass)



Μηχανική μάθηση
και παράλληλη επεξεργασία

Σειριακή εκτέλεση

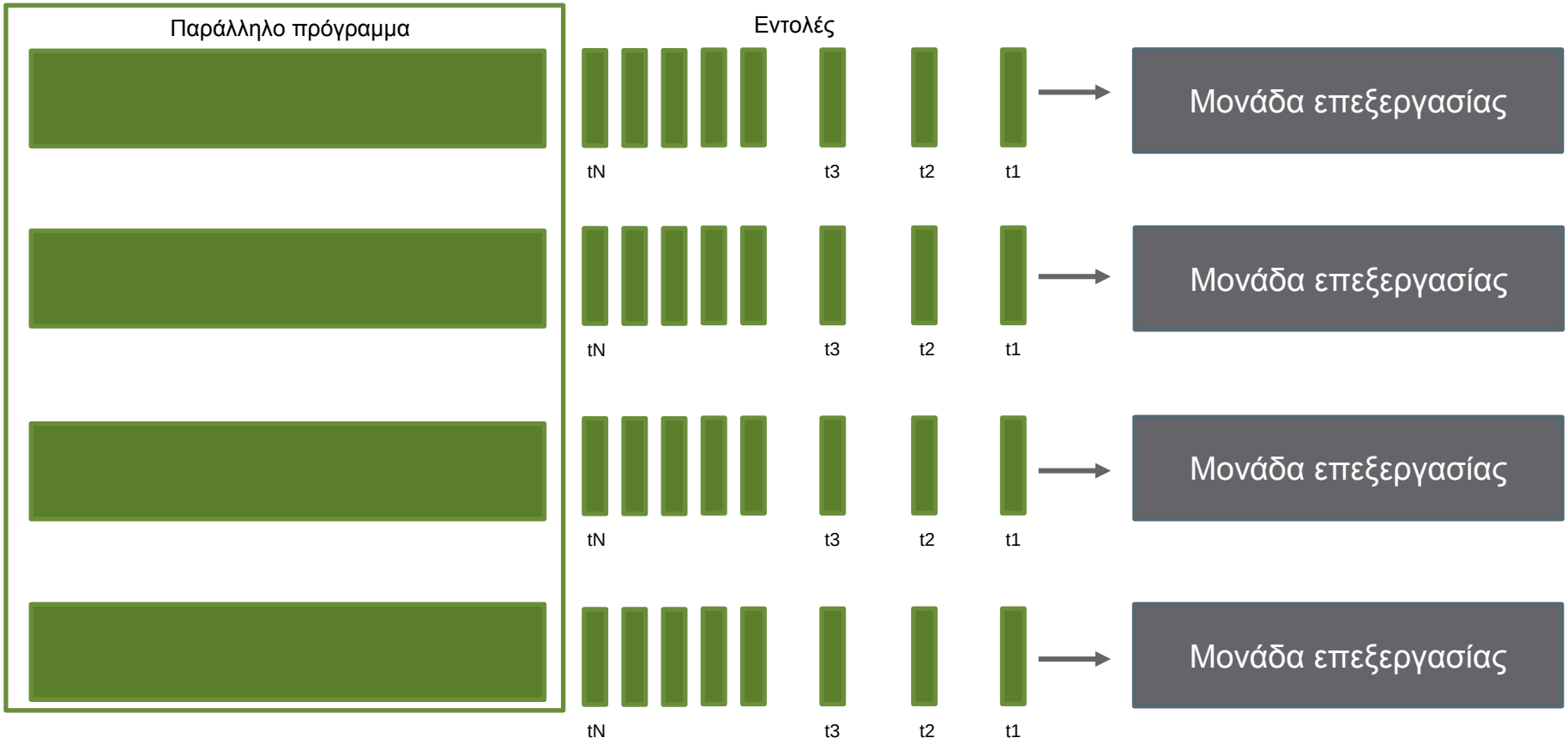
- Πρόβλημα: Κώδικας + Δεδομένα
- Μονάδα επεξεργασίας: Επεξεργαστής + Μνήμη
- Σειριακή εκτέλεση:
  - Το πρόβλημα είναι μια σειρά εντολών (υπολογισμοί και μεταφορές δεδομένων)
  - Οι εντολές εκτελούνται ακολουθιακά - η μία μετά την άλλη
  - Το πρόβλημα εκτελείται σε έναν μόνο επεξεργαστή
  - Κάθε χρονική στιγμή εκτελείται μία μόνο εντολή



Παράλληλη εκτέλεση

- Παράλληλη επεξεργασία:
  - Το πρόβλημα διαιρείται σε κομμάτια που μπορούν να εκτελεστούν ταυτόχρονα/παράλληλα
  - Κάθε κομμάτι του προβλήματος είναι ένα υπο-σύνολο εντολών
  - Οι εντολές κάθε κομματιού μπορούν να εκτελεστούν παράλληλα σε διαφορετικές μονάδες επεξεργασίας
  - Υπάρχει κάποιος μηχανισμός ελέγχου ή/και συγχρονισμού

Παράλληλη επεξεργασία

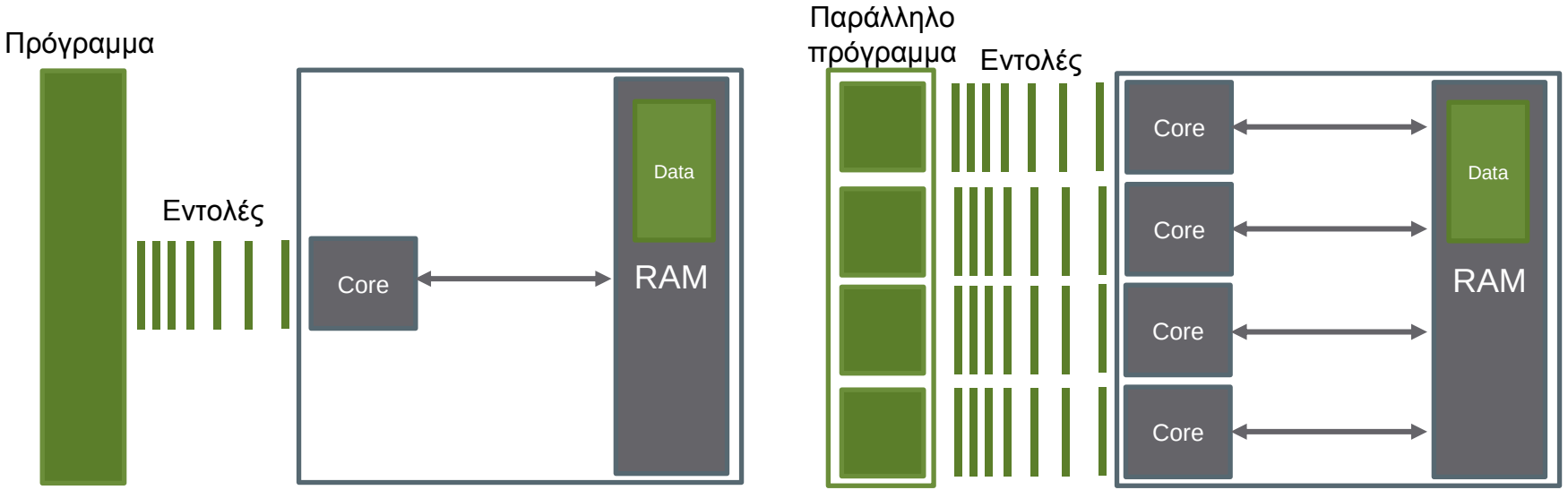


Γιατί παράλληλη επεξεργασία;

- Μπορώ να επιταχύνω τους υπολογισμούς και να μειώσω το χρόνο εκτέλεσης
- Μπορώ να φτιάξω συστοιχίες από μονάδες επεξεργασίας και να αυξήσω τη διαθέσιμη υπολογιστική δύναμη και μνήμη

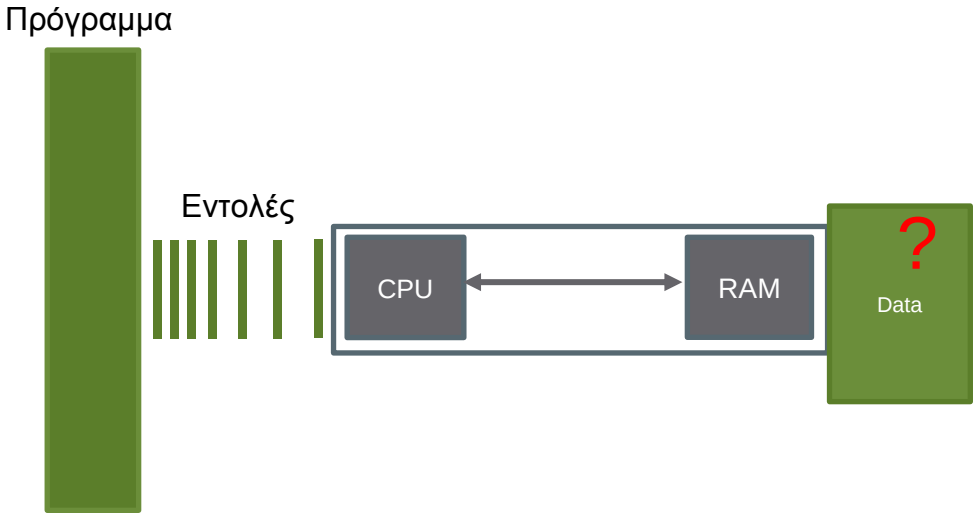
Γιατί παράλληλη επεξεργασία;

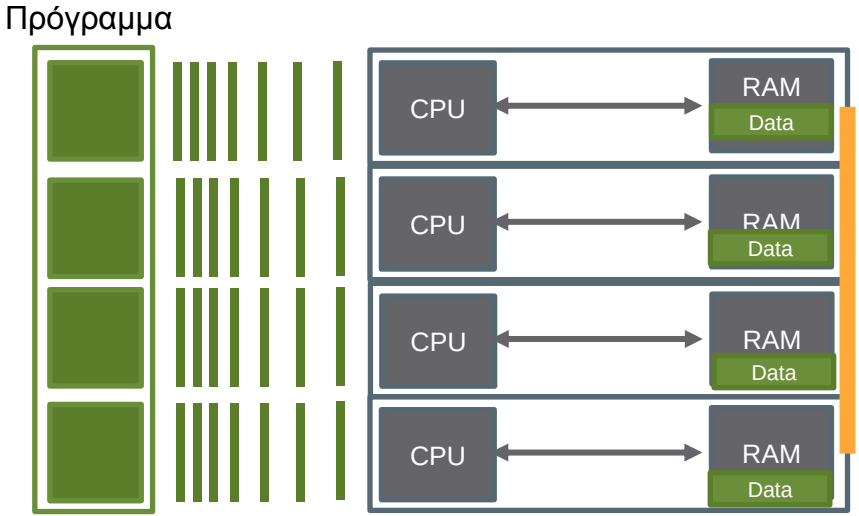
- Για να μειώσω το χρόνο εκτέλεσης
  - 1 μονάδα επεξεργασίας = 1 εργάτης
  - πολλές μονάδες επεξεργασίας = πολλοί εργάτες



Γιατί παράλληλη επεξεργασία;

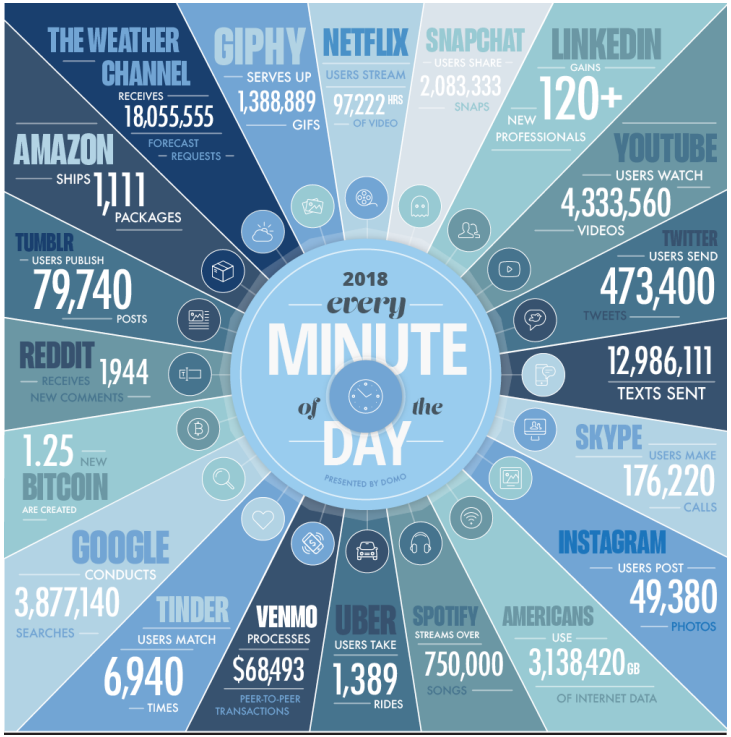
- Γιατί τα δεδομένα του προγράμματος δε χωράνε στη μνήμη ενός επεξεργαστή
  - Μοιράζω τα δεδομένα του προγράμματος στη μνήμη περισσότερων μονάδων επεξεργασίας





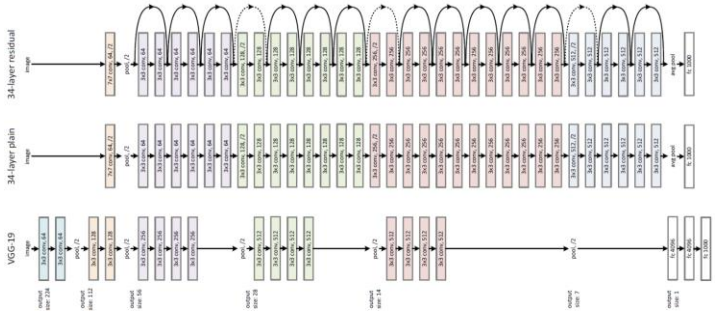
Παράλληλη επεξεργασία και μηχανική μάθηση

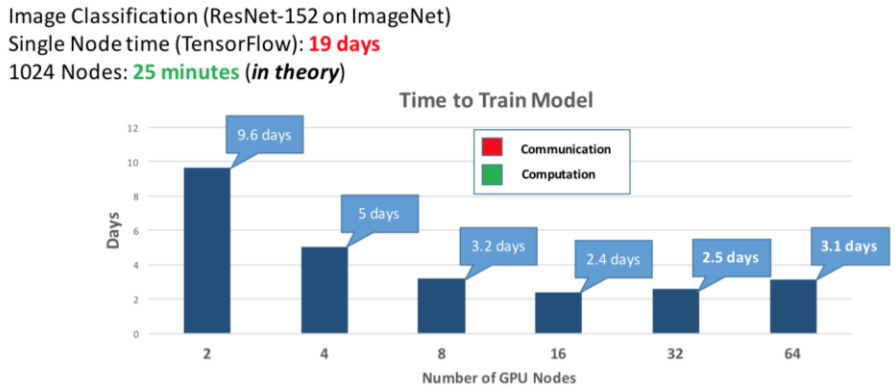
- Αν το πλήθος των δεδομένων εισόδου για εκπαίδευση είναι μεγάλο, απαιτείται παράλληλη επεξεργασία
  - Συλλογές δεδομένων με εκατομμύρια/δισεκατομμύρια καινούργια γεγονότα/entries τη μέρα (internet, finance) => εκατοντάδες TB δεδομένων τη μέρα
  - Τα δεδομένα δε χωράνε στη μνήμη μιας μονάδας επεξεργασίας/ενός υπολογιστικού κόμβου
    - … ή τα δεδομένα δε χωράνε στο δίσκο ενός υπολογιστικού κόμβου => big data => κατανεμημένη επεξεργασία!



Παράλληλη επεξεργασία και μηχανική μάθηση

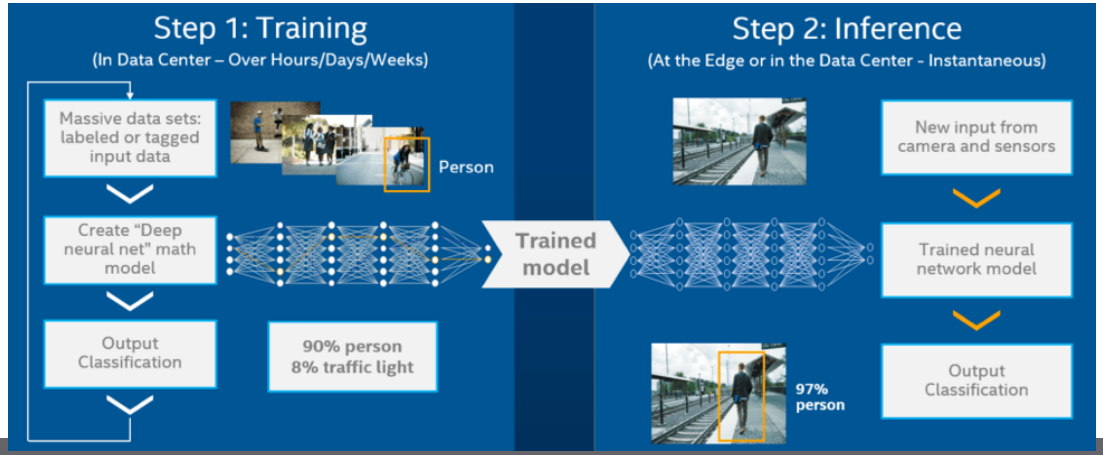
2. Αν ο αλγόριθμος ή το μοντέλο έχει μεγάλη υπολογιστική πολυπλοκότητα, απαιτείται παράλληλη επεξεργασία
   - Ο χρόνος εκτέλεσης σε έναν επεξεργαστή μπορεί να είναι υπερβολικά μεγάλος ή και αποτρεπτικός
     - …και μοντέλα με πολλές παραμέτρους δε χωράνε στη μνήμη ενός κόμβου!
   - Ο διαμοιρασμός των υπολογισμών σε πολλούς επεξεργαστές (ή και η χρήση επιταχυντών) καθιστούν εφικτή την ολοκλήρωση της εκτέλεσης σε λογικό χρόνο





Παράλληλη επεξεργασία και μηχανική μάθηση

3. Αν υπάρχουν περιορισμοί στο χρόνο του inference, η παράλληλη επεξεργασία είναι απαραίτητη
   - Παράδειγμα: έχω εκπαιδεύσει ένα νευρωνικό δίκτυο για αναγνώριση εικόνας και θέλω να το χρησιμοποιήσω σε κάποια εφαρμογή για αναγνώριση εικόνων σε πραγματικό χρόνο (milliseconds)
   - Παράλληλες αρχιτεκτονικές ειδικού σκοπού (GPUs, FPGAs, TPUs)

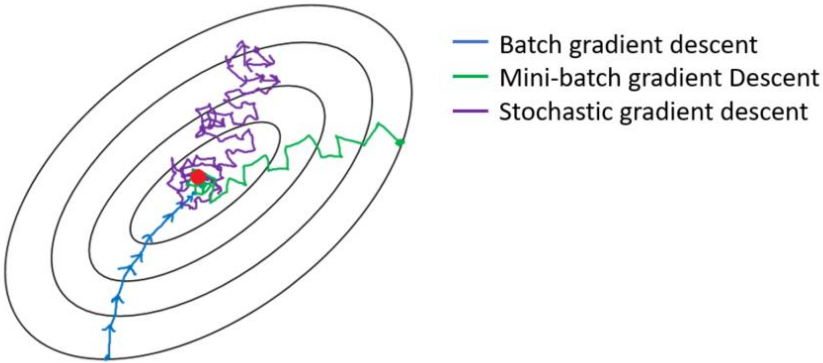


Παράλληλη επεξεργασία και μηχανική μάθηση

- Αν θέλω να επιλέξω μοντέλο ή παραμέτρους, μπορώ να εκπαιδεύσω πολλά μοντέλα παράλληλα
  - ”Embarassingly parallel”: η εκπαίδευση διαφορετικών μοντέλων σε πολλούς επεξεργαστές είναι εύκολη
    - Δεν απαιτεί συνεργασία μεταξύ των επεξεργαστών
    - Μειώνει το χρόνο σε 1/n, όπου n ο αριθμός των διαθέσιμων μονάδων επεξεργασίας

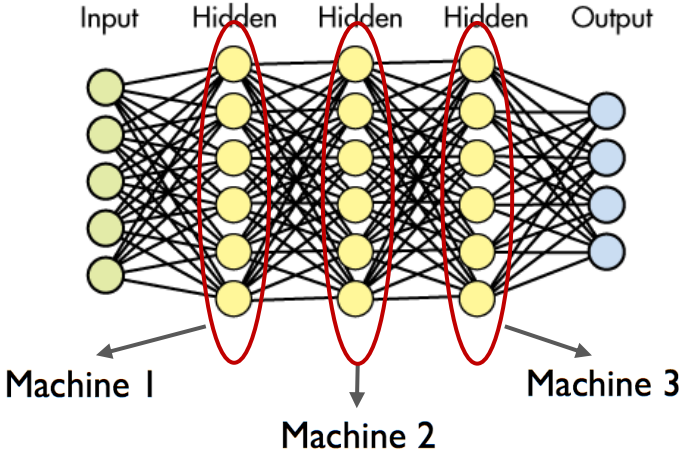
Παράδειγμα: αλγόριθμος εκπαίδευσης Gradient Descent

- *Batch (BGD)*: χρήση *όλων* των δεδομένων εισόδου σε ένα πέρασμα
  - Υψηλές απαιτήσεις σε υπολογισμούς και μνήμη
  - Χωράνε όλα τα δεδομένα σε έναν κόμβο;
- *Stochastic (SGD)*: χρήση *ενός* δεδομένου εισόδου σε ένα πέρασμα
  - Χαμηλές απαιτήσεις σε υπολογισμούς και μνήμη
  - Χωράνε περισσότερα δεδομένα σε έναν κόμβο;
- *Mini-Batch (MB-GD)*: χρήση ενός *μικρού συνόλου* δεδομένων εισόδου σε ένα πέρασμα
  - Συμβιβασμός ανάμεσα σε BGD και SGD



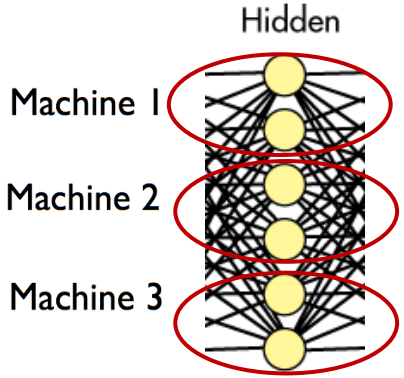
Παράδειγμα: παραλληλισμός στην εκπαίδευση νευρωνικών δικτύων

- **Inter-layer parallelism**: παραλληλισμός σε επίπεδο επεξεργασίας ενός δεδομένου εισόδου (ή mini-batch) ανάμεσα σε διαφορετικά επίπεδα του δικτύου
  - Κάθε μονάδα επεξεργασίας αναλαμβάνει τους υπολογισμούς που αντιστοιχούν στους νευρώνες ενός επιπέδου του δικτυού για τα δεδομένα εισόδου



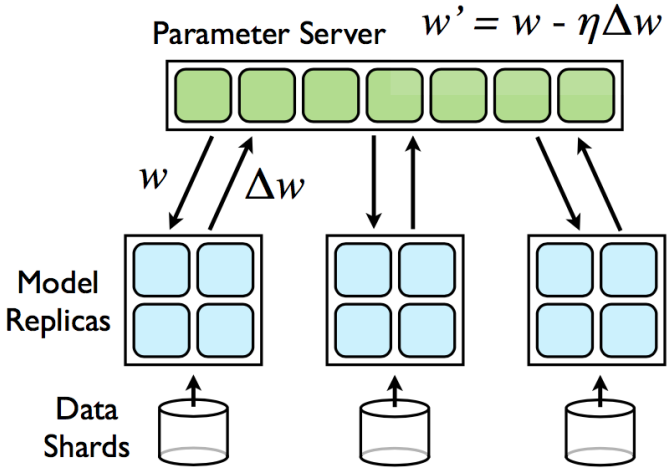
Παράδειγμα: παραλληλισμός στην εκπαίδευση νευρωνικών δικτύων

- **Intra-layer parallelism**: παραλληλισμός σε επίπεδο επεξεργασίας ενός δεδομένου εισόδου (ή mini-batch) εντός ενός επιπέδου του δικτύου
  - Κάθε μονάδα επεξεργασίας αναλαμβάνει τους υπολογισμούς που αντιστοιχούν σε ένα υποσύνολο νευρώνων ενός επιπέδου του δικτύου για τα δεδομένα εισόδου



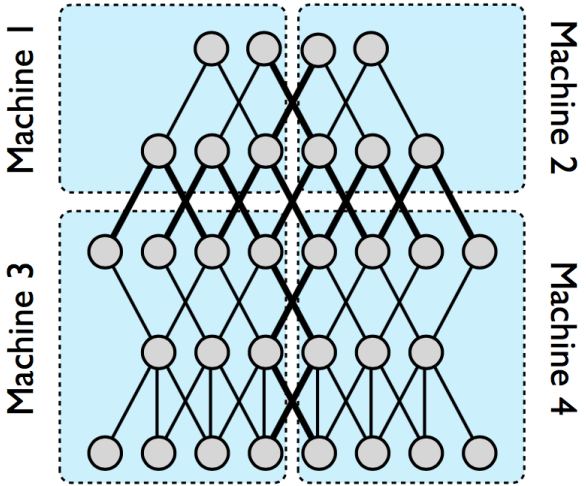
Παράδειγμα: παραλληλισμός στην εκπαίδευση νευρωνικών δικτύων

- **Data parallelism**: παραλληλισμός σε επίπεδο δεδομένων εισόδου
  - Κάθε μονάδα επεξεργασίας διατηρεί ένα αντίγραφο του μοντέλου και εκπαιδεύει τις παραμέτρους με ένα υποσύνολο των δεδομένων εισόδου



Παράδειγμα: παραλληλισμός στην εκπαίδευση νευρωνικών δικτύων

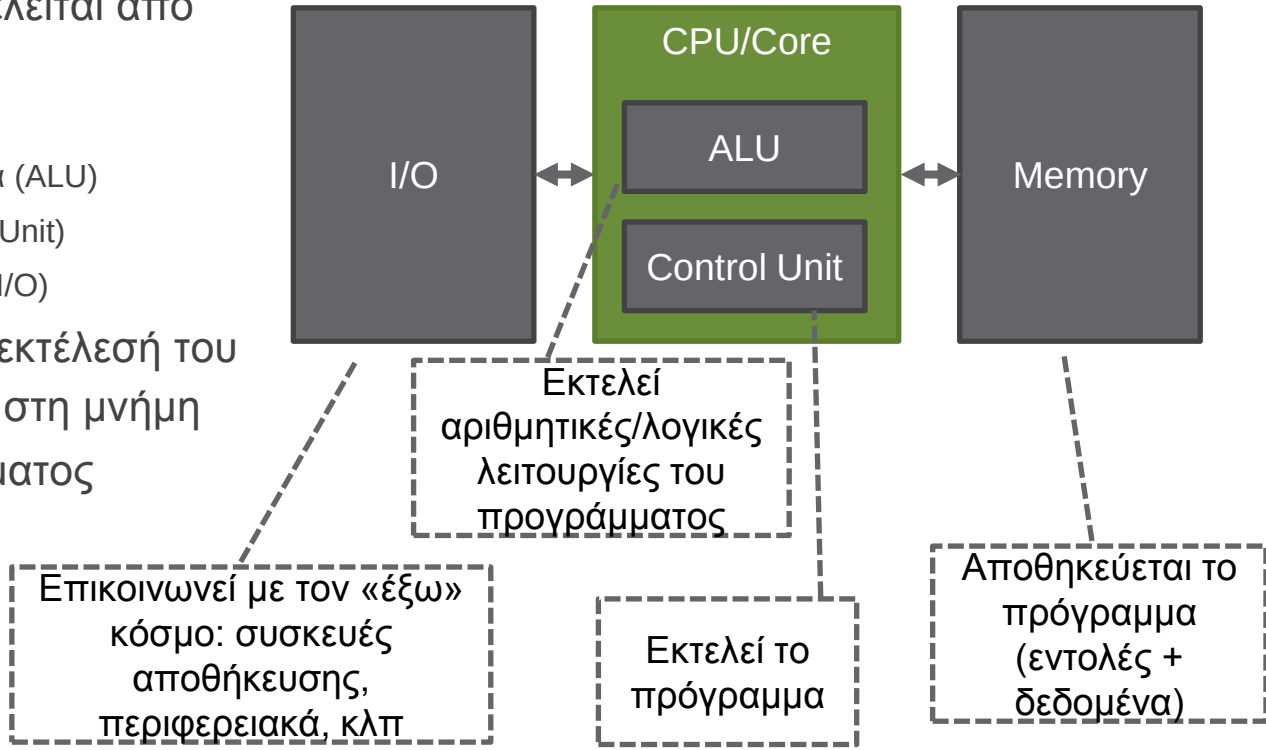
- **Model parallelism**: παραλληλισμός σε επίπεδο μοντέλου
  - Κάθε μονάδα επεξεργασίας αναλαμβάνει να εκπαιδεύσει ένα υποσύνολο των παραμέτρων του μοντέλου με όλα τα δεδομένα εισόδου



Εισαγωγή
στις παράλληλες αρχιτεκτονικές

Σκονάκι: Αρχιτεκτονική Von Neumann

- Ένας υπολογιστής αποτελείται από 4 υπο-συστήματα:
  - Μνήμη (Memory)
  - Αριθμητική/Λογική Μονάδα (ALU)
  - Μονάδα Ελέγχου (Control Unit)
  - Σύστημα εισόδου/εξόδου (I/O)
- Το πρόγραμμα κατά την εκτέλεσή του βρίσκεται αποθηκευμένο στη μνήμη
- Οι εντολές του προγράμματος εκτελούνται ακολουθιακά



Σκονάκι: Εντολές

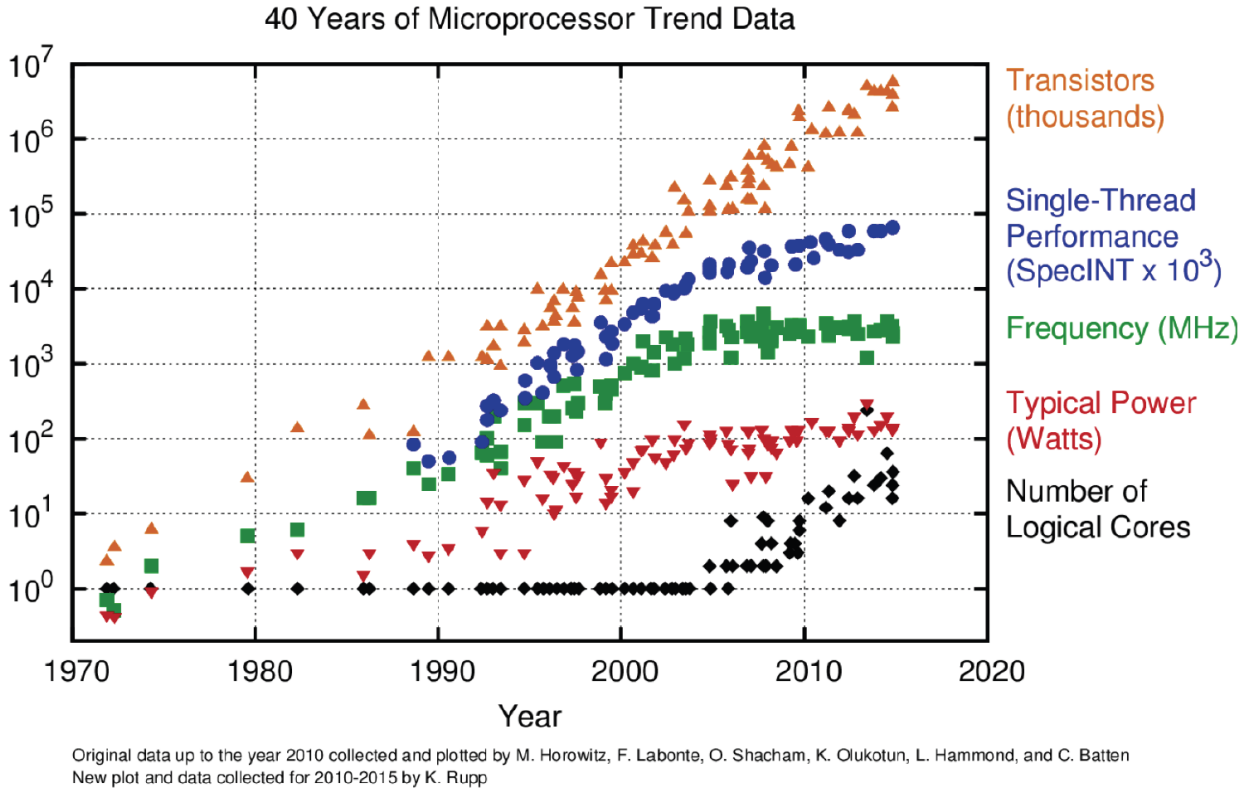
- Μια εντολή είναι μια θεμελιώδης μονάδα εργασίας
- Τρία βασικά είδη:
  - Εντολές υπολογισμού
    - Αριθμητικές/Λογικές πράξεις που εκτελούνται στην ALU
  - Εντολές μεταφοράς δεδομένων
    - Μεταφορά δεδομένων από τη μνήμη στη CPU και αντίστροφα (φόρτωση/αποθήκευση)
  - Εντολές ελέγχου
- Πόσο χρόνο παίρνει μια εντολή;
  - CPU με έναν πυρήνα με συχνότητα 2GHz
    - Βέλτιστη περίπτωση: 1 εντολή ανά κύκλο ~ 0.5ns ανά εντολή
    - Πιο σύνθετες πράξεις κοστίζουν περισσότερους κύκλους
    - Προσβάσεις στη μνήμη κοστίζουν σημαντικά περισσότερους κύκλους

Γιατί παραλληλισμός;

- Τον παλιό καλό καιρό, αν ήθελα να επιταχύνω την εκτέλεση του προγράμματός μου, μπορούσα απλά να περιμένω
  - **Moore's law** – το πλήθος των τρανζίστορ του επεξεργαστή διπλασιάζεται κάθε 18 μήνες
  - **Dennard scaling** – όσο το μέγεθος των τρανζίστορ μειώνεται, η πυκνότητα ισχύος παραμένει σταθερή - ή αλλιώς, η επίδοση ανά Watt διπλασιάζεται κάθε ~18 μήνες
  - **Φυσικό επακόλουθο** – περιοδική αύξηση της συχνότητας/επίδοσης

- *"Free lunch":* η συνεχής αύξηση της επίδοσης ευνόησε την ανάπτυξη νέων εφαρμογών με περισσότερα δεδομένα

Moore's law



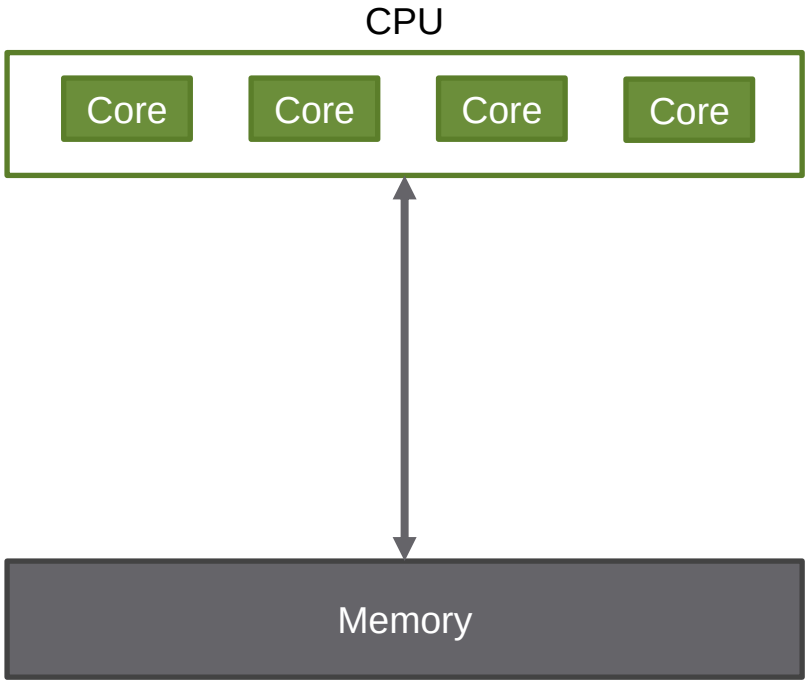
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Γιατί παραλληλισμός;

- *“The free lunch is over”* – Herb Sutter, 2005
  - **Power wall** - Καθώς μειώνεται το μέγεθος των τρανζίστορ, η πυκνότητα ισχύος αυξάνεται
    - Είναι αδύνατη η αύξηση της συχνότητας με τη μείωση του μεγέθους των τρανζίστορ
  - **ILP wall** - Η αύξηση των τρανζίστορ δε βοηθά πια στον παραλληλισμό σε επίπεδο μικροαρχιτεκτονικής
  - **Memory wall** – Η πρόσβαση στη μνήμη είναι πολύ πιο ακριβή από τις αριθμητικές πράξεις (200 κύκλοι για πρόσβαση στη RAM vs 4 κύκλοι για πολλαπλασιασμό)
- *Λύση:* **Παράλληλες αρχιτεκτονικές**
  - Ο νόμος του Moore συνεχίζει να ισχύει – η πυκνότητα των τρανζίστορ ανά επεξεργαστή αυξάνεται
  - Αντί για αύξηση της συχνότητας, αυξάνεται ο αριθμός των πυρήνων ανά επεξεργαστή

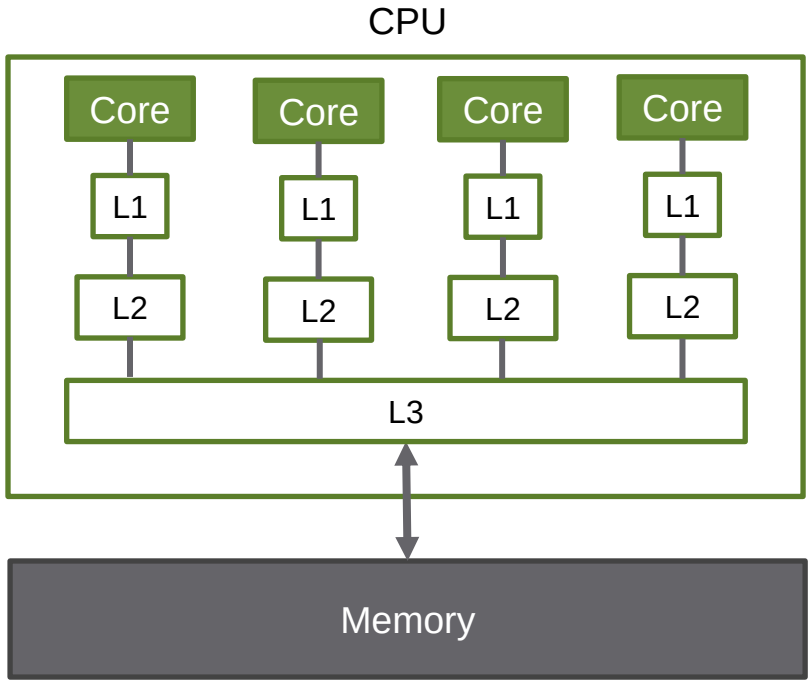
Σύγχρονες CPUs

- Πολλοί παράλληλοι πυρήνες



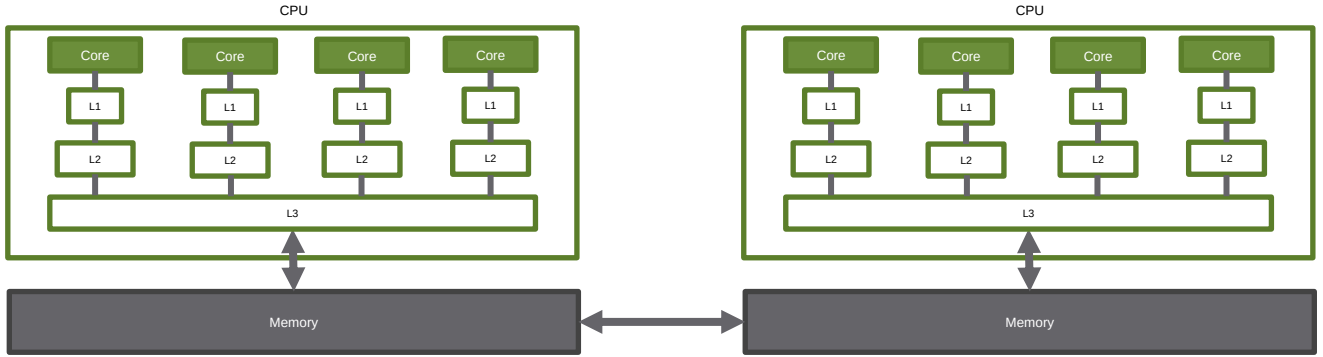
Σύγχρονες CPUs

- Πολλοί παράλληλοι πυρήνες
- Βαθιές ιεραρχίες κρυφής μνήμης



Σύγχρονες CPUs

- Πολλοί παράλληλοι πυρήνες
- Βαθιές ιεραρχίες κρυφής μνήμης
- Συχνά παράλληλες CPUs στον ίδιο κόμβο

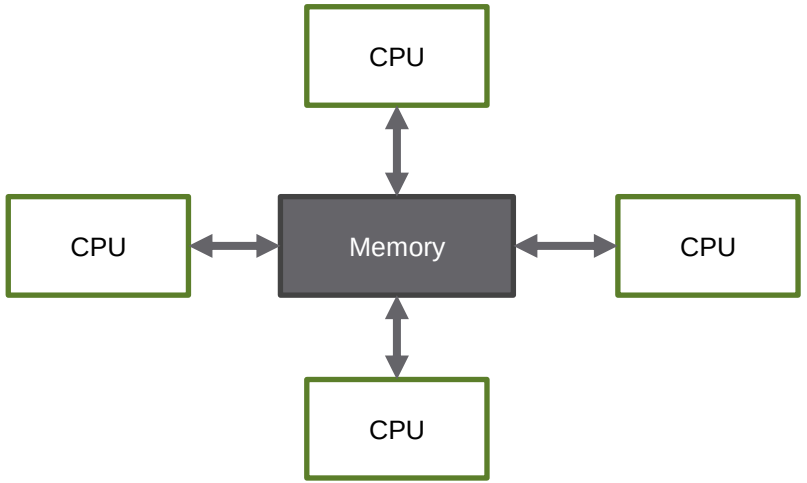


Αρχιτεκτονικές κοινής μνήμης

- Οι σύγχρονες CPUs είναι αρχιτεκτονικές κοινής μνήμης
- Όλοι οι πυρήνες βλέπουν τη μνήμη ως ενιαίο χώρο διευθύνσεων
- Όλοι οι πυρήνες μπορούν να εργαστούν ανεξάρτητα αλλά μοιράζονται τους ίδιους πόρους μνήμης
- Οι αρχιτεκτονικές κοινής μνήμης μπορεί να είναι UMA (uniform memory access) ή NUMA (non-uniform memory access), ανάλογα με το χρόνο πρόσβασης στην κύρια μνήμη

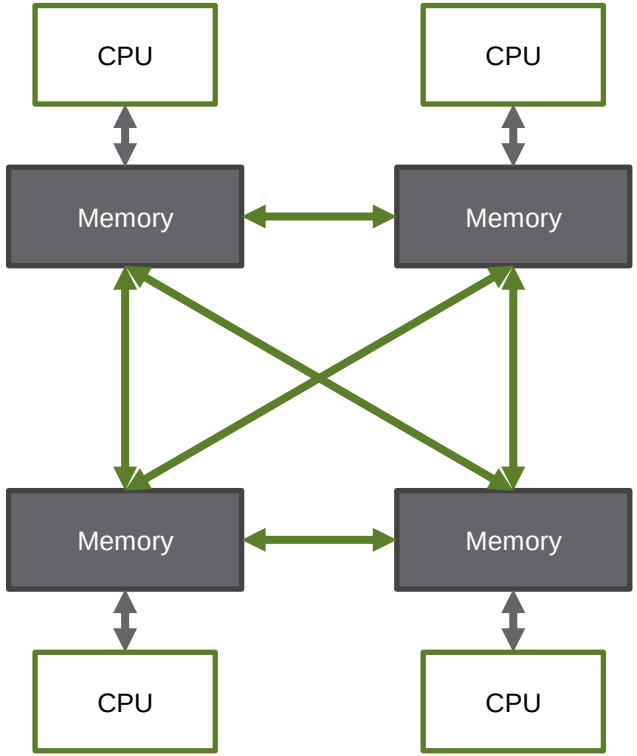
Αρχιτεκτονικές κοινής μνήμης (UMA)

- Αρχιτεκτονικές κοινής μνήμης με ομοιόμορφη πρόσβαση στη μνήμη – UMA
  - Αλλιώς symmetric multi-processors (SMPs)
  - Ίδιοι επεξεργαστές
  - Ισότιμη πρόσβαση και ίσος χρόνος πρόσβασης στην κύρια μνήμη
  - Cache-coherent: Αν κάποιος επεξεργαστής ενημερώσει κάποια θέση μνήμης στην κοινή μνήμη, όλοι οι επεξεργαστές ενημερώνονται



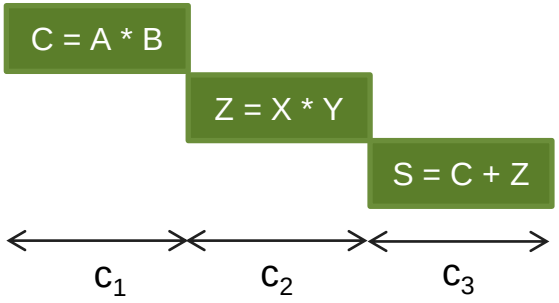
Αρχιτεκτονικές κοινής μνήμης (NUMA)

- Αρχιτεκτονικές κοινής μνήμης με ανομοιόμορφη πρόσβαση στη μνήμη – NUMA
  - Συχνά κατασκευάζονται από δύο ή περισσότερους SMPs
  - Κάθε SMP μπορεί να έχει πρόσβαση στη μνήμη του άλλου
  - Οι επεξεργαστές δεν έχουν ίσους χρόνους πρόσβασης σε κάθε μνήμη
  - Cache-coherent NUMA: ομοίως με cache-coherent UMA, αν όλοι οι επεξεργαστές είναι ενήμεροι για κάποια ενημέρωση σε οποιαδήποτε μνήμη

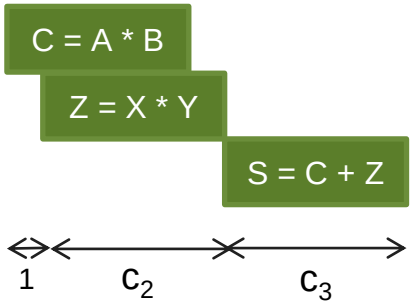


Παραλληλισμός στις CPUs

- Παραλληλισμός σε επίπεδο εντολών (instruction level parallelism)
  - Pipelining, Superscalar architectures
- Παράδειγμα: οι δύο πρώτες εντολές μπορούν να εκτελεστούν ταυτόχρονα
  - **C** = A * B
  - **Z** = X * Y
  - S = **C** + **Z**



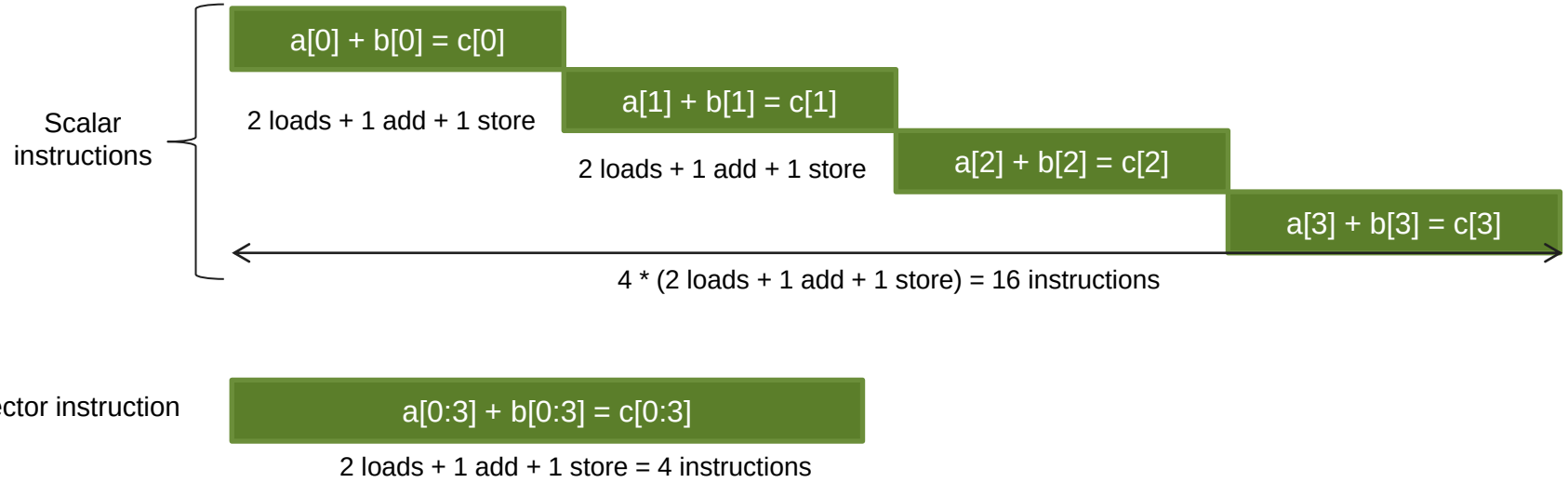
Χωρίς pipelining: $c_1 + c_2 + c_3$



Με pipelining: $1 + c_2 + c_3$

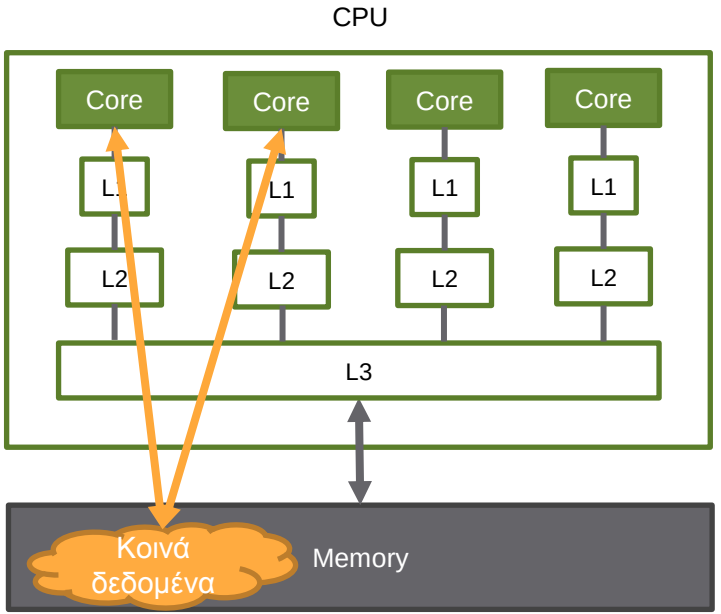
Παραλληλισμός στις CPUs

- Παραλληλισμός διανυσμάτων
  - Εντολές διανυσμάτων: εκτέλεση της **ίδιας πράξης** σε **διαφορετικά δεδομένα** παράλληλα
- Παράδειγμα: άθροισμα διανυσμάτων **c** = **a** + **b**



Παραλληλισμός στις CPUs

- Παραλληλισμός πολλών πυρήνων
  - Πολλοί ίδιοι πυρήνες σε έναν επεξεργαστή (multi-core CPUs)
- Οι πυρήνες μπορούν να εργάζονται ανεξάρτητα σε ανεξάρτητες παράλληλες εργασίες
- Επικοινωνούν μέσω της κοινής/μοιραζόμενης μνήμης
  - Γράφουν δεδομένα και διαβάζουν δεδομένα σε/από την ίδια μνήμη
  - Αξιοποιούν την ιεραρχία κρυφών μνημών

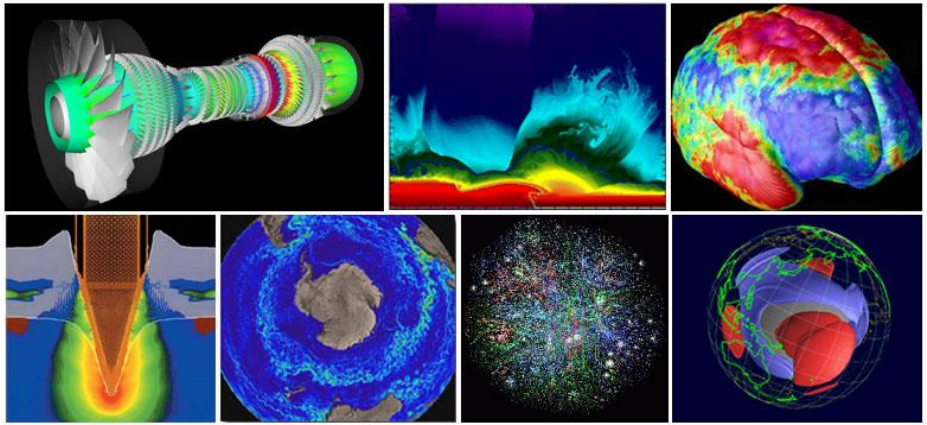


Παραλληλισμός στις CPUs

- Θετικά
  - + Παραλληλισμός σε πολλά σημεία της αρχιτεκτονικής – δυνατότητα υψηλής επίδοσης μέσω παράλληλης επεξεργασίας
  - + Κοινή μνήμη – κοινά δεδομένα – «εύκολος» παραλληλισμός – εύκολος προγραμματισμός
- Αρνητικά
  - - Κοινή μνήμη – κοινά δεδομένα – ανταγωνισμός/συμφόρηση κατά την πρόσβαση σ’αυτά
    - Περιορισμός από το υλικό: περιορισμένο εύρος ζώνης από/προς την κύρια μνήμη, περιορισμένη χωρητικότητα των κρυφών μνημών
  - - Κοινά δεδομένα – ανάγκη για συγχρονισμό
  - - Περιορισμένη κλιμακωσιμότητα της αρχιτεκτονικής

Γιατί παραλληλισμός;

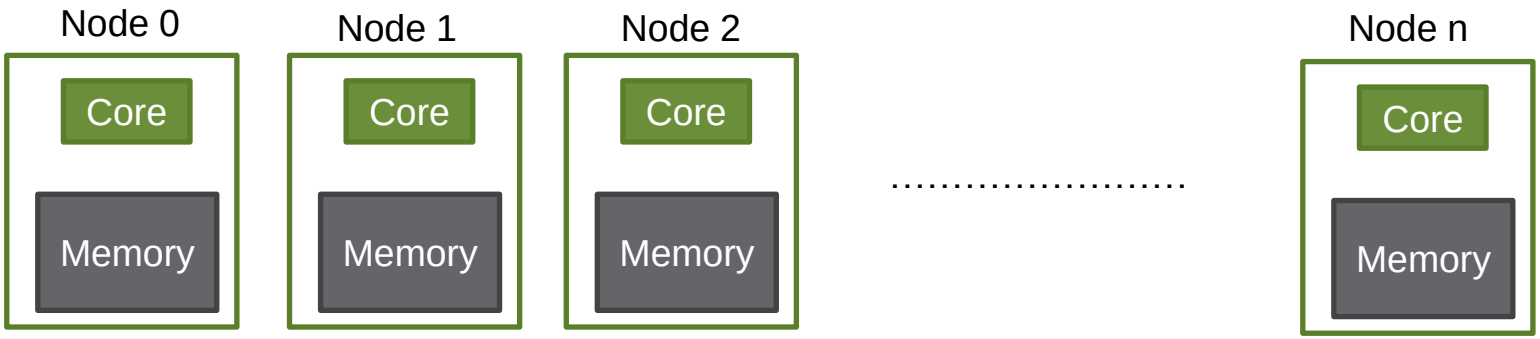
- Ακόμα και τον παλιό καλό καιρό, υπήρχαν προγράμματα που *δεν μπορούσαν να περιμένουν την εξέλιξη της τεχνολογίας*
  - «Δύσκολα» επιστημονικά προβλήματα με μεγάλη υπολογιστική πολυπλοκότητα και υψηλές απαιτήσεις σε μνήμη απαιτούσαν πάντα παράλληλη επεξεργασία
- «Παραδοσιακός» παραλληλισμός: *επεξεργασία υψηλών επιδόσεων* (**high-performance computing /supercomputing**)



Ατμόσφαιρα, Γη, Περιβάλλον
Φυσική
Βιοτεχνολογία, Γενετική
Χημεία, Μοριακές επιστήμες
Γεωλογία, Σεισμολογία
Μηχανολογία
…

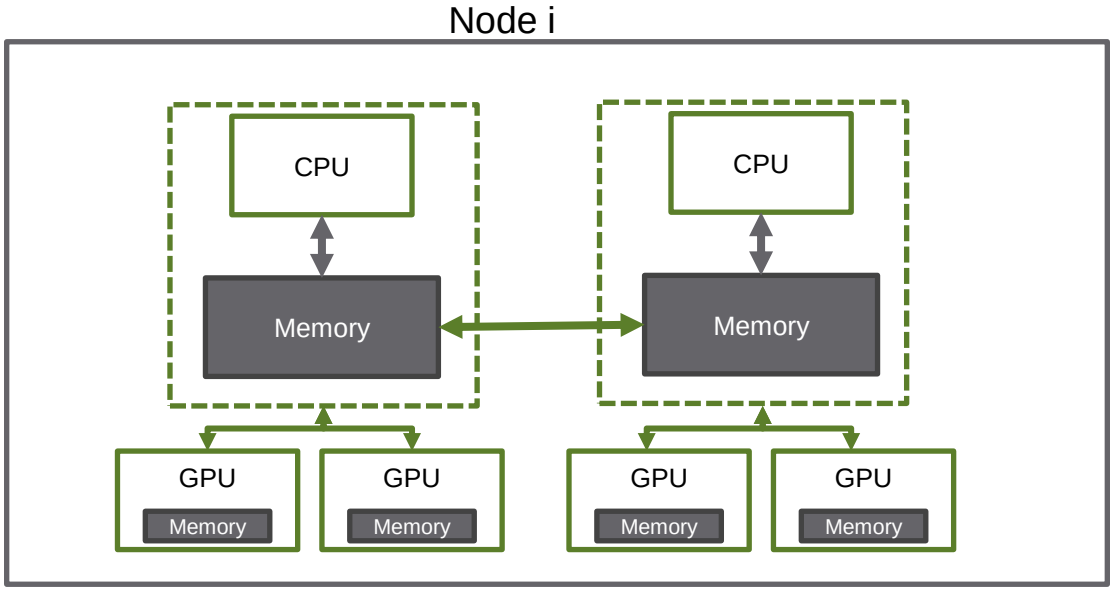
Συστοιχίες (clusters/supercomputers)

- Πολλοί παράλληλοι υπολογιστικοί κόμβοι
  - «Παραδοσιακά» (πριν το 2005): κόμβος = 1 CPU/core + memory



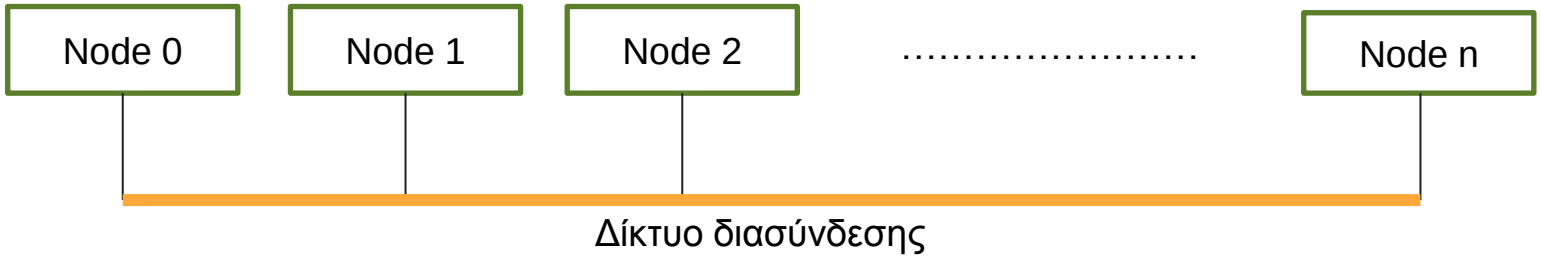
Σύγχρονες συστοιχίες (clusters/supercomputers)

- Πολλοί παράλληλοι υπολογιστικοί κόμβοι
  - Σύγχρονοι υπολογιστικοί κόμβοι: πολλαπλές CPUs, κοινή μνήμη (UMA ή NUMA), ένας ή περισσότεροι επιταχυντές



Σύγχρονες συστοιχίες (clusters/supercomputers)

- Πολλοί παράλληλοι υπολογιστικοί κόμβοι
- Κάθε κόμβος έχει τη δική του μνήμη
- Η μνήμη ενός κόμβου δεν είναι «ορατή» από άλλους κόμβους
  - Δεν είναι μοιραζόμενη
- Οι κόμβοι διασυνδέονται με κάποιο δίκτυο διασύνδεσης υψηλής επίδοσης



Αρχιτεκτονικές κατανεμημένης μνήμης

- Οι συστοιχίες υπολογιστών είναι αρχιτεκτονικές κατανεμημένης μνήμης
- Αλλαγές/Ενημερώσεις στη μνήμη ενός κόμβου δεν είναι ορατές σε άλλους κόμβους
- Η επικοινωνία μεταξύ των κόμβων γίνεται μέσω του δικτύου διασύνδεσης με ανταλλαγή μηνυμάτων
- Μια εφαρμογή μπορεί να αξιοποιήσει όλη τη διαθέσιμη υπολογιστική ισχύ και κατανεμημένη μνήμη με το κατάλληλο προγραμματιστικό μοντέλο

Παραλληλισμός σε συστοιχίες

- Κάθε πυρήνας, CPU, κόμβος μπορεί να είναι ένας «εργάτης» για το ίδιο πρόγραμμα
- Οι «εργάτες» επικοινωνούν μεταξύ τους με ανταλλαγή μηνυμάτων μέσω του δικτύου διασύνδεσης
    - «Εργάτες» στον ίδιο κόμβο μπορούν να επικοινωνήσουν μέσω της κοινής μνήμης του κόμβου

Παραλληλισμός στις συστοιχίες

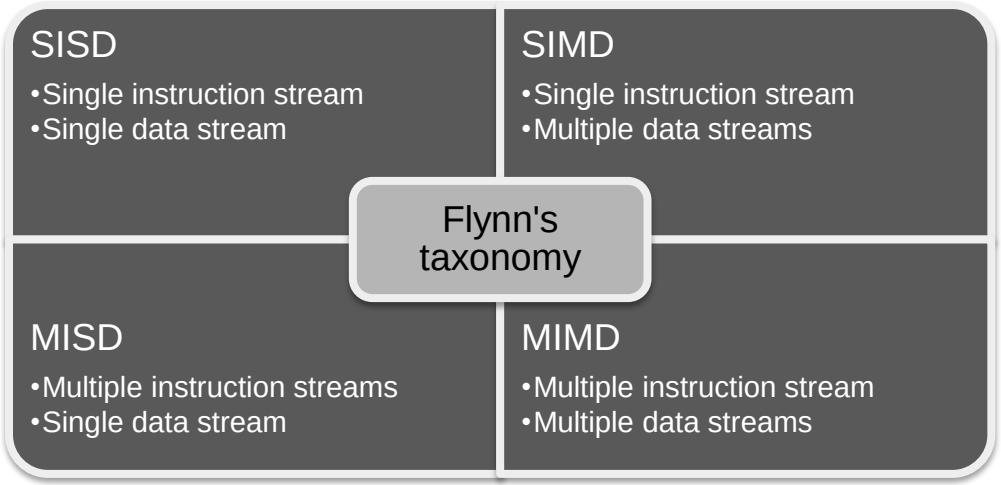
- Θετικά
  - + Παραλληλισμός μέσω των πολλαπλών κόμβων (και εντός του κόμβου) – δυνατότητα πολύ υψηλής επίδοσης μέσω παράλληλης επεξεργασίας
  - + Μεγάλη κλιμακωσιμότητα της αρχιτεκτονικής (χιλιάδες κόμβοι)
  - + Αξιοποίηση διαθέσιμης μνήμης πολλαπλών κόμβων – επίλυση μεγάλων προβλημάτων
- Αρνητικά
  - - Κατανεμημένη μνήμη – ανάγκη για επικοινωνία - ευθύνη του προγραμματιστή
  - - Κατανεμημένη μνήμη – ανάγκη για επικοινωνία – επιβραδύνσεις του προγράμματος από την επικοινωνία μέσω του δικτύου διασύνδεσης

Top500

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,397,824	143,500.0	200,794.9	9,783
2	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
3	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
4	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 , NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
5	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	387,872	21,230.0	27,154.3	2,384

Flynn’s taxonomy

- Ένας τρόπος ταξινόμησης παράλληλων αρχιτεκτονικών αφορά τη μνήμη (κοινή/κατανεμημένη)
- Εναλλακτικά: Flynn’s taxonomy
  - Ροές εντολών (Instruction streams): single/multiple
  - Ροές δεδομένων (Data streams): single/multiple



Flynn’s taxonomy: τα δύο άκρα

- SISD: μία CPU με έναν μόνο πυρήνα
  - Μπορεί να εκτελεί ένα μόνο πρόγραμμα (single instruction stream) σε δεδομένα που φέρνει από τη μνήμη σε ένα μόνο stream
    - Μπορώ να τρέξω ένα μόνο πρόγραμμα
    - Δεν μπορώ να τρέξω παράλληλα προγράμματα (δεν υπάρχει παραλληλισμός στις εντολές/στα δεδομένα)
- MIMD: μία σύγχρονη CPU με πολλούς πυρήνες
  - Μπορεί να εκτελεί πολλαπλά προγράμματα (multiple instruction streams) σε πολλαπλά δεδομένα που φέρνει ταυτόχρονα από τη μνήμη σε διαφορετικά streams
    - Μπορώ να τρέχω ένα διαφορετικό πρόγραμμα σε κάθε πυρήνα
    - Μπορώ να τρέξω παράλληλα προγράμματα

Flynn’s taxonomy: SIMD

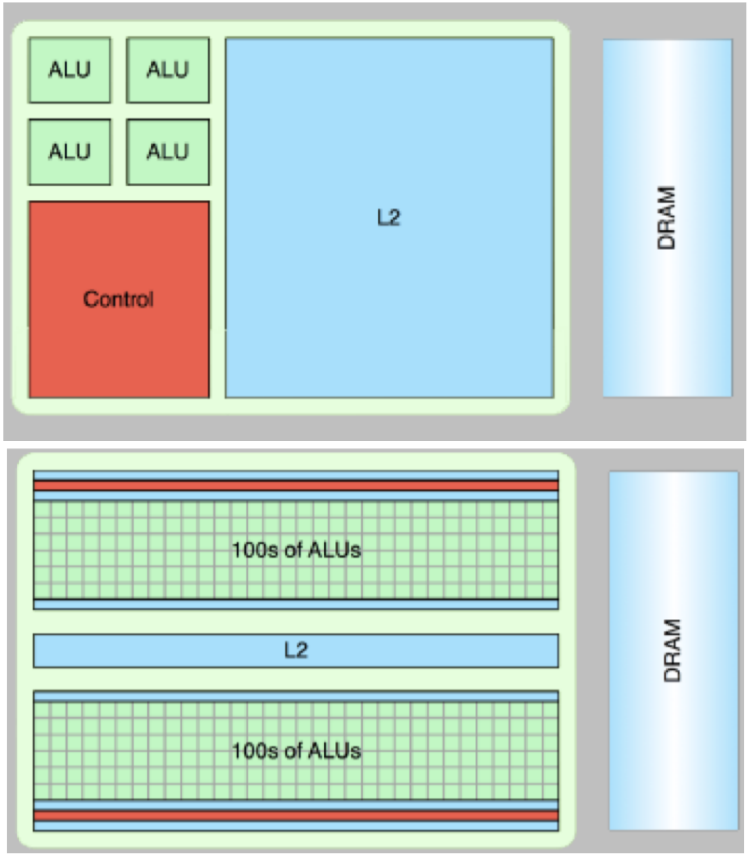
- MISD: δεν έχει κατασκευαστεί
  - Μπορεί να εκτελεί διαφορετικό πρόγραμμα (multiple instruction streams) στα ίδια δεδομένα (single instruction stream)
    - Θεωρητικό παράδειγμα: πολλοί διαφορετικοί αλγόριθμοι κρυπτογραφίας προσπαθούν να αποκρυπτογραφήσουν το ίδιο κρυπτογραφημένο μήνυμα
- SIMD
  - Μπορεί να εκτελεί την ίδια εντολή σε μια συγκεκριμένη χρονική στιγμή (single instruction stream) σε πολλαπλά δεδομένα που φέρνει ταυτόχρονα από τη μνήμη σε διαφορετικά streams
    - Μπορώ να εφαρμόζω την ίδια πράξη σε διαφορετικά δεδομένα παράλληλα
      - Παράδειγμα: Παραλληλισμός διανυσμάτων στις CPUs

Αρχιτεκτονικές SIMD

- GPUs: Χαρακτηριστικό παράδειγμα SIMD
  - Στα γραφικά/στην επεξεργασία εικόνας, μία πράξη (η ίδια) εκτελείται πολλές φορές σε διαφορετικά δεδομένα (π.χ. σε κάθε pixel)
- Τι είναι μία GPU;
  - Κάτι σαν μία CPU με πολλούς πυρήνες
  - Κάθε πυρήνας/thread της GPU είναι πολύ πιο απλός και εκτελεί **συγκεκριμένες πράξεις** πολύ πιο γρήγορα από ό,τι ο αντίστοιχος πυρήνας της CPU
  - Πολλοί περισσότεροι απλούστεροι πυρήνες σε μία GPU από ό,τι σε μία πολυπύρηνη CPU
    - “Massive parallelism”!

Σύγχρονες GPUs

- Οι GPUs είναι μια βελτιστοποιημένη αρχιτεκτονική για προγράμματα που εκτελούν πολλαπλές αριθμητικές πράξεις
- Σε μία GPU, το υλικό για αριθμητικές πράξεις είναι πολύ περισσότερο από το υλικό για μνήμη



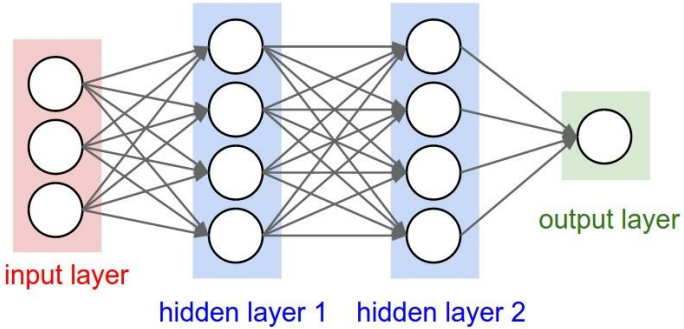
Παραλληλισμός στις GPUs

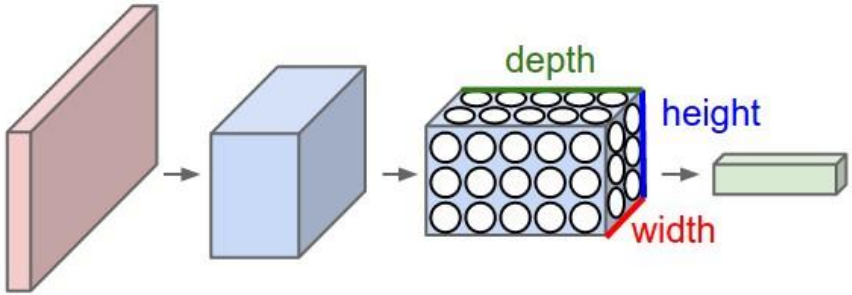
- Θετικά
  - \+ Παραλληλισμός μέσω των πολλαπλών απλών πυρήνων για αριθμητικές πράξεις
  - \+ Υψηλή επίδοση σε προγράμματα που στηρίζονται σε αριθμητικές πράξεις
- Αρνητικά
  - \- Πιο δύσκολος, εξειδικευμένος προγραμματισμός
  - \- Χαμηλή επίδοση σε προγράμματα που οι προσβάσεις στη μνήμη είναι ανομοιόμορφες ή τυχαίες

Τάσεις στην παράλληλη επεξεργασία αλγορίθμων βαθιάς μηχανικής μάθησης

Ανάλυση αλγορίθμων βαθιάς μηχανικής μάθησης

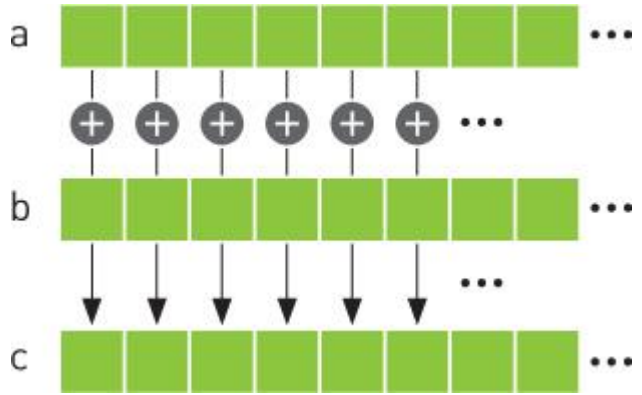
- Οι αλγόριθμοι βαθιάς μηχανικής μάθησης ανάγονται σε πράξεις γραμμικής άλγεβρας σε τανυστές (tensors)





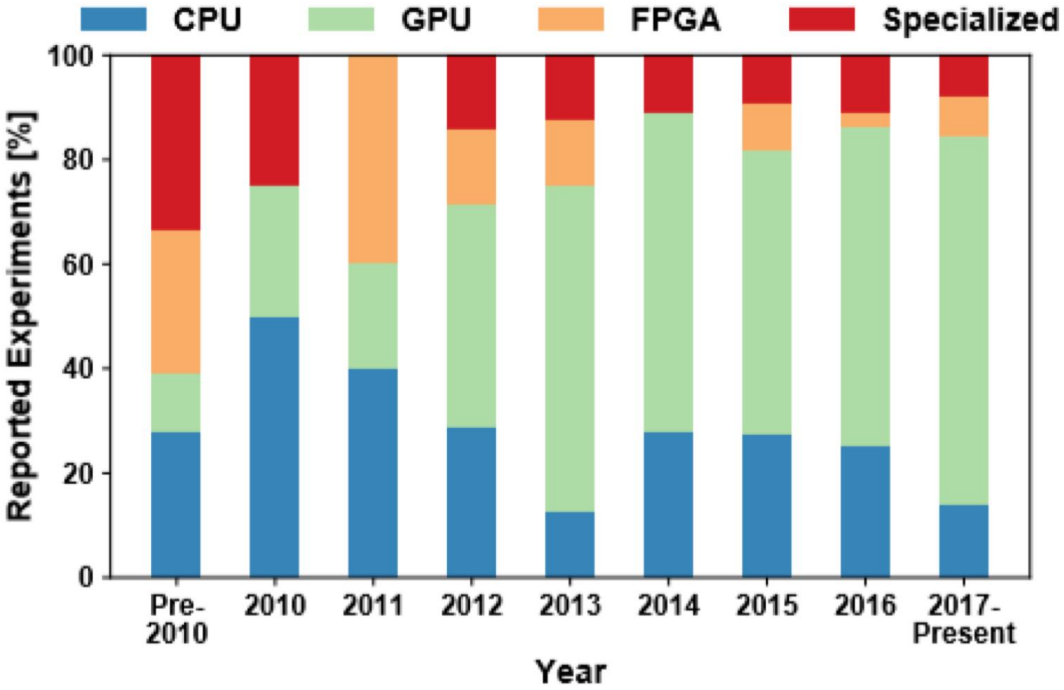
Χαρακτηριστικά υπολογισμών γραμμικής άλγεβρας

- Οι υπολογισμοί γραμμικής άλγεβρας ανάγονται σε εφαρμογή μιας γραμμικής συνάρτησης σε όλα τα δεδομένα του τανυστή
- *Παράδειγμα*: άθροισμα διανυσμάτων
  - Το διάνυσμα εξόδου προκύπτει από εφαρμογή της συνάρτησης f(x, y) = x + y (**S**ingle **I**nstruction) σε κάθε ζεύγος στοιχείων (a_i, b_i) των διανυσμάτων εισόδου (**M**ultiple **D**ata)
- GPUs = SIMD!
  - Άρα αποτελούν ιδανική μονάδα επεξεργασίας για πράξεις γραμμικής άλγεβρας



Βαθιά νευρωνικά δίκτυα και GPUs

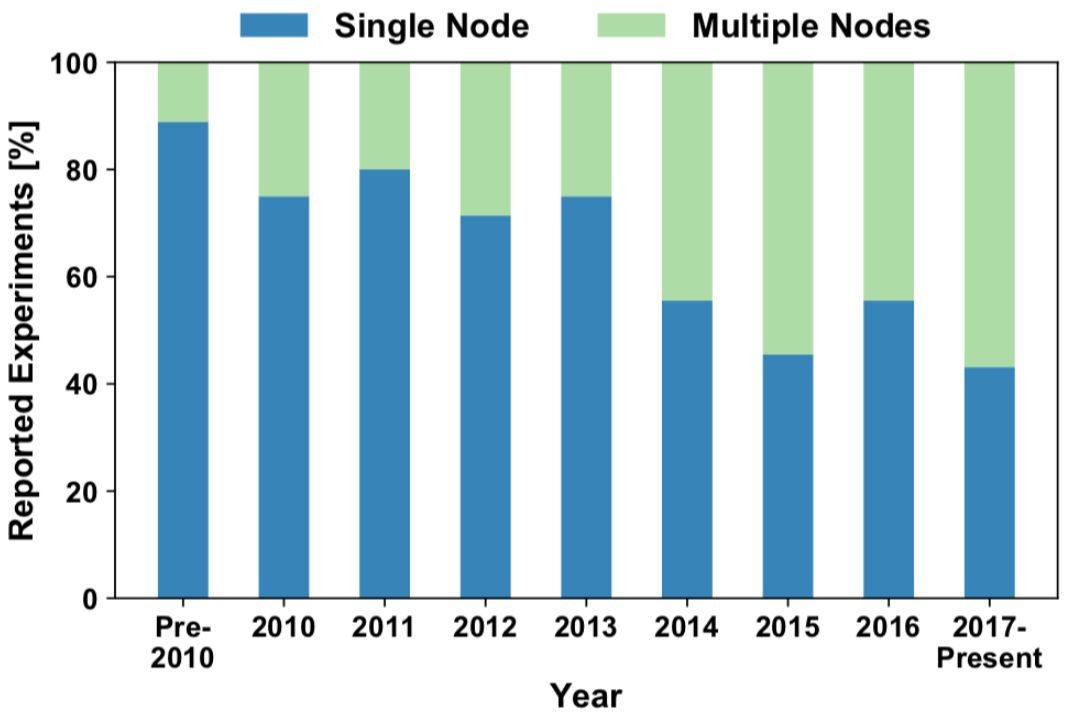
- Η εκπαίδευση βαθιών νευρωνικών δικτύων βασίζεται σε γνωστούς υπολογιστικούς πυρήνες γραμμικής άλγεβρας
- Οι GPUs χρησιμοποιούνται εδώ και αρκετά χρόνια σε συστήματα υψηλών επιδόσεων για την επίλυση προβλημάτων με ένταση στους υπολογισμούς (compute-intensive)
- *Οι GPUs κερδίζουν συνεχώς έδαφος στο deep learning*



Χρήση υλικού σε 227 papers που χρησιμοποιούν παράλληλα συστήματα για βαθιά νευρωνικά δίκτυα

Βαθιά νευρωνικά δίκτυα και πολλαπλοί κόμβοι

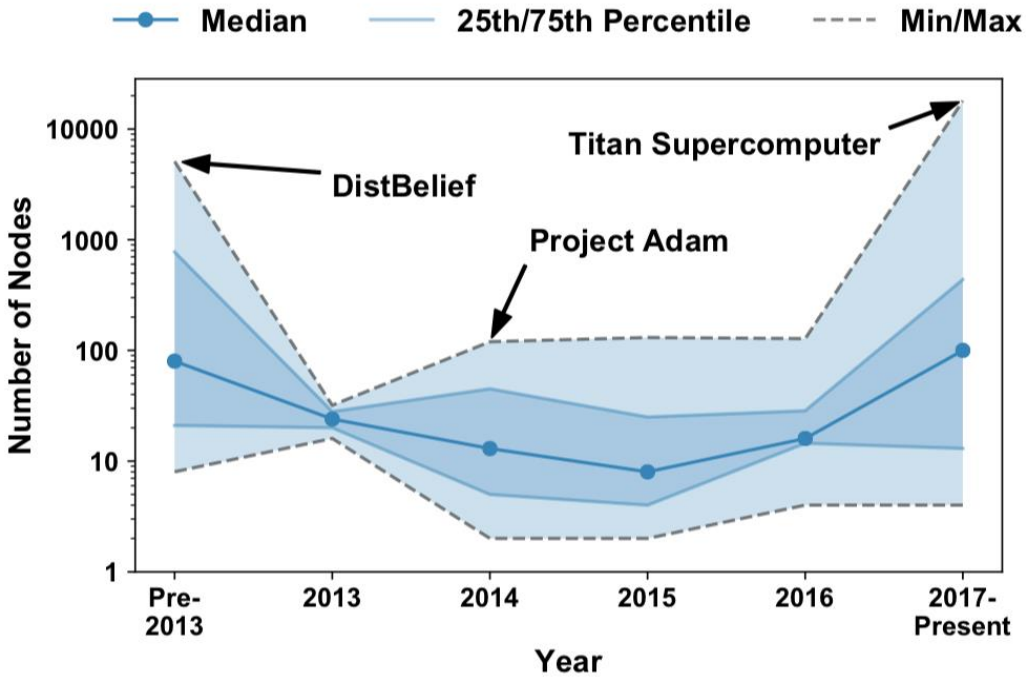
- Όλο και περισσότεροι ερευνητές χρησιμοποιούν συστοιχίες υπολογιστικών κόμβων για την εκπαίδευση βαθιών νευρωνικών δικτύων
  - Μεγάλες απαιτήσεις σε μνήμη
    - Παραλληλισμός σε επίπεδο δεδομένων
    - Παραλληλισμός σε επίπεδο μοντέλου
- Οι πολλαπλοί κόμβοι αυξάνουν τη διαθέσιμη υπολογιστική ισχύ προς αξιοποίηση



Χρήση συστοιχιών σε 227 papers που χρησιμοποιούν παράλληλα συστήματα για βαθιά νευρωνικά δίκτυα

Βαθιά νευρωνικά δίκτυα και συστήματα υψηλής επίδοσης

- Η ανάγκη για μείωση του χρόνου εκπαίδευσης οδηγεί σε χρήση πολλαπλών κόμβων για την εκπαίδευση βαθιών νευρωνικών δικτύων
- Η αύξηση στη χρήση ισχυρών GPUs μείωσε αυτή την ανάγκη για λίγο (2015)
- "***Employing Deep Learning Methods to Understand Weather Patterns" -*** Gordon Bell prize winner 2018
  - 4560 υπολογιστικοί κόμβοι
  - 27360 GPUs
  - ~65k images/s



Πλήθος κόμβων σε 227 papers που χρησιμοποιούν παράλληλα συστήματα για βαθιά νευρωνικά δίκτυα