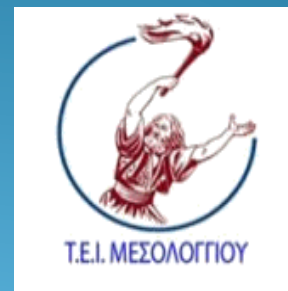


# Υπολογιστική Νοημοσύνη: Υλοποίηση Νευρωνικών Δικτύων

Ανδρέας Παπαζώης

Τμ. Διοίκησης Επιχειρήσεων





# Περιεχόμενα Εργ. Μαθήματος

- Εκπαίδευση (μάθηση) Νευρωνικών Δικτύων
- Απλός αισθητήρας
- Παράδειγμα εκπαίδευσης
- Θέματα υλοποίησης Νευρωνικών Δικτύων



# Διαδικασία Μάθησης

- Η διαδικασία της εκπαίδευσης (μάθησης) αποσκοπεί στη βελτίωση των συναπτικών βαρών των Νευρωνικών Δικτύων
- Υπάρχουν διάφορες μέθοδοι μάθησης, με κάθε μία να παρουσιάζει συγκεκριμένα:
  - πλεονεκτήματα έναντι μειονεκτημάτων
- «Εποχή» λέγεται η διαδικασία της παροχής εισόδων στα δίκτυα και της τροποποίησης των συναπτικών βαρών:
  - Το πλήθος των εποχών είναι πρακτικά η μονάδα μέτρησης για την αξιολόγηση του πόσο γρήγορα προσαρμόζεται ένα δίκτυο
- Οι μέθοδοι μάθησης μπορούν να κατηγοριοποιηθούν σε δύο κατηγορίες ως εξής:
  - Μάθηση με επίβλεψη
  - Μάθηση χωρίς επίβλεψη

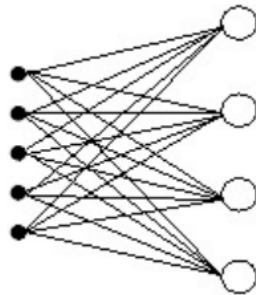


# Κατηγορίες Μάθησης

- Μάθηση με επίβλεψη:
  - παρέχονται εισοδοι και οι επιθυμητές έξοδοι
  - υπολογίζονται οι έξοδοι
  - τα βάρη τροποποιούνται για να μειωθεί η διαφορά μεταξύ των τρεχουσών και των επιθυμητών τιμών εξόδου
- Μάθηση χωρίς επίβλεψη:
  - παρέχονται μόνο εισοδοι
  - τα βάρη τροποποιούνται ώστε παρόμοιες εισοδοι να προκαλούν παρόμοιες εξόδους
  - Το δίκτυο μπορεί να βρίσκει πρότυπα και διαφοροποιήσεις στις εισόδους χωρίς εξωτερική βοήθεια

# Απλοί Αισθητήρες

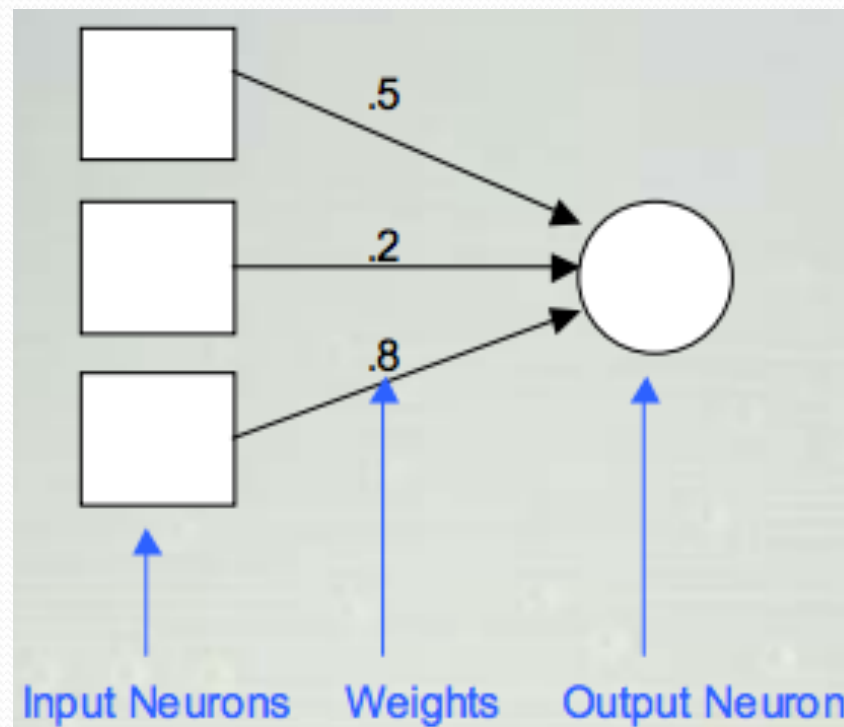
- Οι απλοί αισθητήρες (perceptrons) είναι τα απλούστερα νευρωνικά δίκτυα με δυνατότητα μάθησης
- Αποτελούνται από ένα επίπεδο νευρώνων:



- Οι είσοδοι έχουν 2 δυνατές τιμές (ON/OFF ή 0/1)
- Η συνάρτηση ενεργοποίησης είναι μία απλή συνάρτηση με κατώφλι:
  - 0 ή 1 ανάλογα με το αν το ζυγισμένο άθροισμα περνά ή όχι μία τιμή)
  - Συνήθως είναι η βηματική συνάρτηση
- Μπορούν να λύσουν μόνο απλά προβλήματα:
  - έχουν πλέον περιορισμένες εφαρμογές

# Παράδειγμα Απλού Αισθητήρα

- Παράδειγμα αισθητήρα με 3 εισόδους και 1 έξοδο







# Μάθηση Αισθητήρων

- Οι αισθητήρες χρησιμοποιούν μάθηση με επίβλεψη
- Αρχικά τα βάρη έχουν:
  - μικρές τιμές, που έχουν αποδοθεί με τυχαίο τρόπο
  - για όλο τον αισθητήρα την τιμή  $w$
- Σε κάθε εποχή μάθησης:
  - Δίνεται η είσοδος *input* στον αισθητήρα
  - Υπολογίζεται η πραγματική έξοδος *output*
  - Συγκρίνεται η πραγματική έξοδος *output* με την επιθυμητή έξοδο *desired*
  - Αν είναι ίδιες, δεν υπάρχει αλλαγή στα βάρη
  - Αν δεν είναι ίδιες, προκαλείται αλλαγή σε κάθε βάρος
- Η διαδικασία επαναλαμβάνεται έως ότου:
  - Ο αισθητήρας απαντά σωστά σε όλες τις εισόδους
  - Δεν υπάρχει περαιτέρω βελτίωση στα βάρη



# Διόρθωση Βαρών

- Η νέα τιμή των βαρών  $w_{new}$  προκύπτει από την προηγούμενη τιμή  $w_{old}$  με χρήση του εξής τύπου:

$$w_{new} = w_{old} + \alpha(desired - output) * input$$

- $\alpha$  είναι μία θετική σταθερά:
  - ονομάζεται ρυθμός μάθησης
  - καθορίζει την «ευαισθησία» του αισθητήρα σε λάθη
  - όσο πιο μεγάλη τιμή έχει, τόσο πιο πολύ διαφοροποιούνται οι τιμές των βαρών για το ίδιο λάθος
  - μεγάλο  $\alpha$  μπορεί να οδηγήσει σε γρηγορότερη σύγκλιση, αλλά και σε παλινδρόμηση γύρω από τις βέλτιστες τιμών βαρών
  - μικρό  $\alpha$  οδηγεί σε πιο αργή σύγκλιση, ενώ μπορεί να οδηγήσει σε παγίδευση σε τοπικά ακρότατα
- Οι αισθητήρες συγκλίνουν και δίνουν ικανοποιητικά αποτελέσματα στην πλεοψηφία των προβλημάτων
  - Δεν είναι κατάλληλοι για προβλήματα που δεν είναι γραμμικά διαχωρίσιμα





# Παράδειγμα Διαδικασίας για AND

x	y	wx	wy	wz	weighted sum	output	desired	wx (new)	wy (new)	wz (new)
0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0.1	0.1	0.1
0	0	0	0	0	0.1	1	0	0.1	0.1	0
0	1	0	0	0	0.1	1	0	0.1	0	-0.1
1	0	0	0	-0	0	0	0	0.1	0	-0.1
1	1	0	0	-0	0	0	1	0.2	0.1	0
0	0	0	0	0	0	0	0	0.2	0.1	0
0	1	0	0	0	0.1	1	0	0.2	0	-0.1
1	0	0	0	-0	0.1	1	0	0.1	0	-0.2
1	1	0	0	-0	-0.1	0	1	0.2	0.1	-0.1
0	0	0	0	-0	-0.1	0	0	0.2	0.1	-0.1
0	1	0	0	-0	0	0	0	0.2	0.1	-0.1
1	0	0	0	-0	0.1	1	0	0.1	0.1	-0.2
1	1	0	0	-0	0	0	1	0.2	0.2	-0.1
0	0	0	0	-0	-0.1	0	0	0.2	0.2	-0.1
0	1	0	0	-0	0.1	1	0	0.2	0.1	-0.2
1	0	0	0	-0	0	0	0	0.2	0.1	-0.2
1	1	0	0	-0	0.1	1	1	0.2	0.1	-0.2
0	0	0	0	-0	-0.2	0	0	0.2	0.1	-0.2

Τμ. Διοίκησης Επιχειρήσεων  
ΤΕΙ Δυτικής Ελλάδας



# Συνεχείς Συναρτήσεις Ενεργοποίησης

- Η βηματική συνάρτηση δεν είναι η καλύτερη επιλογή ως συνάρτηση ενεργοποίησης
- Οι συναρτήσεις ενεργοποίησης εξυπηρετεί να είναι συνεχείς (παραγωγίσιμες)
- Αυτό γιατί η παράγωγος δείχνει το ρυθμό μεταβολής της συνάρτησης
  - Προσπαθούμε να αποφύγουμε βάρη, μικρές μεταβολές των οποίων προκαλούν μεγάλες μεταβολές στο σφάλμα, γιατί αυτά δημιουργούν ένα ασταθές νευρωνικό δίκτυο
  - Επομένως η παράγωγος είναι μία σημαντική ένδειξη για τις μεταβολές στο σφάλμα και μπορεί να χρησιμοποιηθεί στη μάθηση
- Παράδειγμα κατάλληλης σχετικής συνάρτησης ενεργοποίησης, η λογιστική συνάρτηση:
  - Τύπος συνάρτησης:  $f(x) = \frac{1}{1 + e^{-x}}$
  - Παράγωγος:  $\frac{d}{dx}f(x) = f(x) \cdot (1 - f(x))$ .



# Υλοποίηση: Classes

- Βασικές classes:
  - NeuralNetwork
    - Περιέχει 1 ή περισσότερα NeuronLayer
  - NeuronLayer
    - Περιέχεται σε ένα NeuralNetwork
    - Περιέχει 1 ή περισσότερα Neuron
  - Neuron
    - Περιέχεται σε 1 NeuronLayer
- Για τη σύνδεση των Neuron μπορεί να χρησιμοποιηθεί η class Synapse:
  - Κάθε Neuron δέχεται εισόδους από 1 ;ή περισσότερα Neuron και τροφοδοτεί με την έξοδό του 1 ή περισσότερα Neuron



# Υλοποίηση: Class Neuron

- Ιδιότητες:
  - bias
  - delta (η διαφορά μεταξύ τρεχόντων και νέων βαρών)
  - output
  - inputConnections
  - outputConnections
- Μέθοδοι:
  - updateDelta()
  - updateOutput()
  - updateWeights()
  - updateBias()



# Υλοποίηση: Class Synapse

- Ιδιότητες:
  - weight
  - fromNeuron
  - toNeuron
- Μέθοδοι:
  - `initiateWeight()` (μπορεί να θέτει με κάποιο τυχαίο τρόπο αρχική τιμή στο βάρος της σύναψης)
  - `updateWeight()`



# Υλοποίηση: Class NeuronLayer

- Ιδιότητες:
  - neurons (συλλογή από Neuron)
- Μέθοδοι:
  - ActivateLayer() (ενεργοποιεί τις συναρτήσεις ενεργοποίησης των νευρώνων και τα αποτελέσματα μπορούν να χρησιμοποιηθούν ως είσοδοι στο επόμενο επίπεδο ή ως έξοδος)
- Η παραπάνω class μπορεί να χρησιμοποιηθεί για τη δημιουργία subclasses:
  - InputLayer με ιδιότητα nextLayer και μέθοδο transferToNextLayer()
    - transferToNextLayer() τα αποτελέσματα των συναρτήσεων ενεργοποίησης τροφοδοτούν το επόμενο επίπεδο
  - HiddenLayer με ιδιότητες: previousLayer, nextLayer και μέθοδο transferToNextLayer():
    - transferToNextLayer() τα αποτελέσματα των συναρτήσεων ενεργοποίησης τροφοδοτούν το επόμενο επίπεδο
  - OutputLayer με ιδιότητες previousLayer και μέθοδο terminateIteration():
    - τα αποτελέσματα των συναρτήσεων ενεργοποίησης αποτελούν και τελικά αποτελέσματα του δικτύου
    - δεν υπάρχουν συνδέσεις εξόδου





# Υλοποίηση: Class NeuralNetwork

- Ιδιότητες:
  - hiddenLayers (Συλλογή από NeuronLayer)
  - inputLayer
  - outputLayer
- Μέθοδοι:
  - createNetwork()
  - getOutput()
  - TrainNetwork()



# Βοηθητικές Classes

- **TrainingSet**: Δεδομένα εισόδου/εξόδου για μάθηση με επίβλεψη
- **TrainingElement**: Δεδομένα εισόδου/εξόδου για μία εποχή
- **LearningRule**: Μέθοδος/τύποι που χρησιμοποιούνται για να προσαρμοστούν τα βάρη κατά την μάθηση
- **TransferFunction**: Ο τύπος της συνάρτησης ενεργοποίησης



# Ευχαριστώ!

Επικοινωνία: [parazois@ceid.upatras.gr](mailto:parazois@ceid.upatras.gr)

Τμ. Διοίκησης Επιχειρήσεων  
ΤΕΙ Δυτικής Ελλάδας