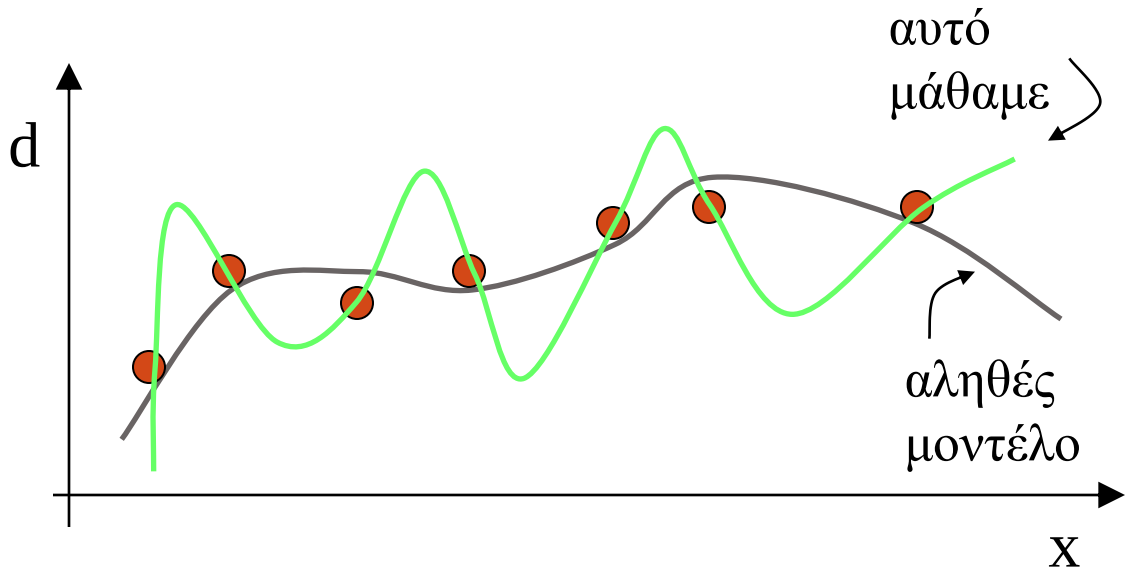
Μάθηση και Γενίκευση

Κώστας Διαμαντάρας
Τμήμα Πληροφορικής
ΤΕΙ Θεσσαλονίκης

Στόχος της μάθησης

- Να μάθουμε ακριβώς τις τιμές των στόχων $\mathbf{d}_i$ γιά κάθε διάνυσμα εισόδου $\mathbf{x}_i$;



Στόχος της μάθησης (2)

- Να μάθουμε την κρυμμένη «αλήθεια» δηλαδή το στατιστικό μοντέλο που παρήγαγε τα δεδομένα.
- Μην ξεχνάμε ότι υπάρχει θόρυβος παρατήρησης
- Στόχος η **γενίκευση**.

Γενίκευση: η ικανότητα να εκτιμάμε τη σωστή έξοδο $\mathbf{d}_j$ για πρότυπα εισόδου $\mathbf{x}_j$ που δεν έχουμε δει κατά την εκπαίδευση.

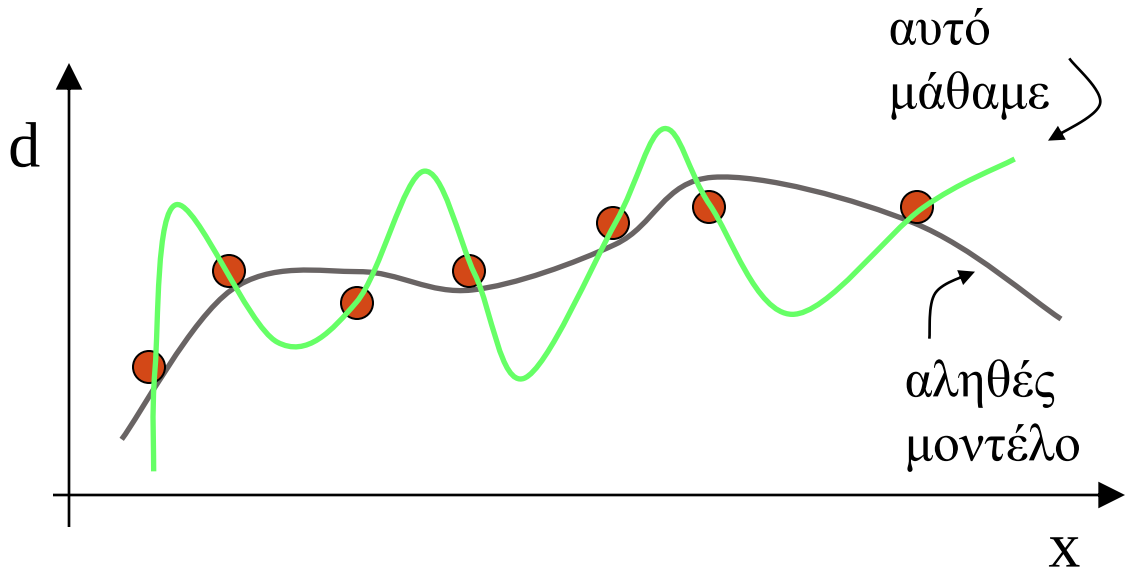
Επιλογή μοντέλου

Έστω ότι χρησιμοποιούμε MLP και Back-Propagation. (Το ίδιο ισχύει και για τα δίκτυα RBF ή για οποιοδήποτε άλλο δίκτυο που εκπαιδεύεται με επίβλεψη)

- Χρήση πολλών νευρώνων → το δίκτυο μπορεί να περιγράψει ιδιαίτερα πολύπλοκες καμπύλες (ή επιφάνειες)
- Χρήση λίγων νευρώνων → το δίκτυο μπορεί να περιγράψει απλές καμπύλες (ή επιφάνειες)
- Ποια επιλογή να κάνω?

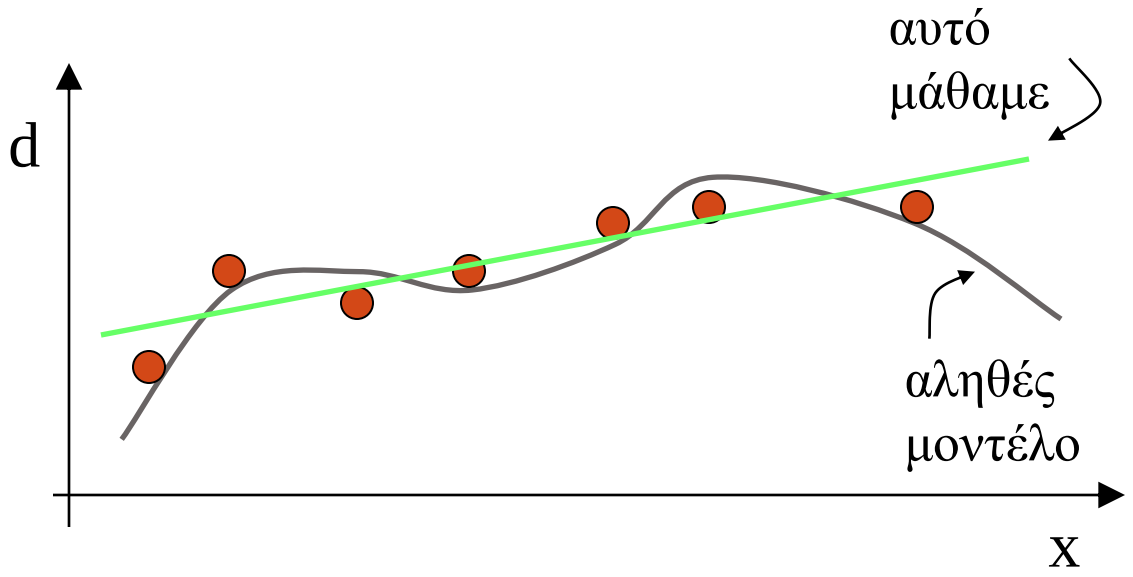
Λάθος 1: περισσότεροι νευρώνες απ' όσους πρέπει

Σφάλμα υπερ-μοντελοποίησης (over-modeling)



Λάθος 2: λιγότεροι νευρώνες απ' όσους πρέπει

Σφάλμα υπο-μοντελοποίησης (under-modeling)



Δύο Συναρτήσεις Κόστους

Συνάρτηση Κόστους ή Συνάρτηση Σφάλματος Εκπαίδευσης

$$J_{train} = \sum_{p=1}^{P} \sum_{i=1}^{M} (d_i^{(p)} - y_i^{(p)})^2$$

P = το πλήθος των προτύπων στα οποία εκπαιδεύτηκε το δίκτυο (training set)

M = το πλήθος των εξόδων του δικτύου

$d_i^{(p)}, y_i^{(p)}$ = στόχοι και έξοδοι του δικτύου για το πρότυπο p

Δύο Συναρτήσεις Κόστους (2)

Συνάρτηση Κόστους ή Συνάρτηση Σφάλματος Ελέγχου

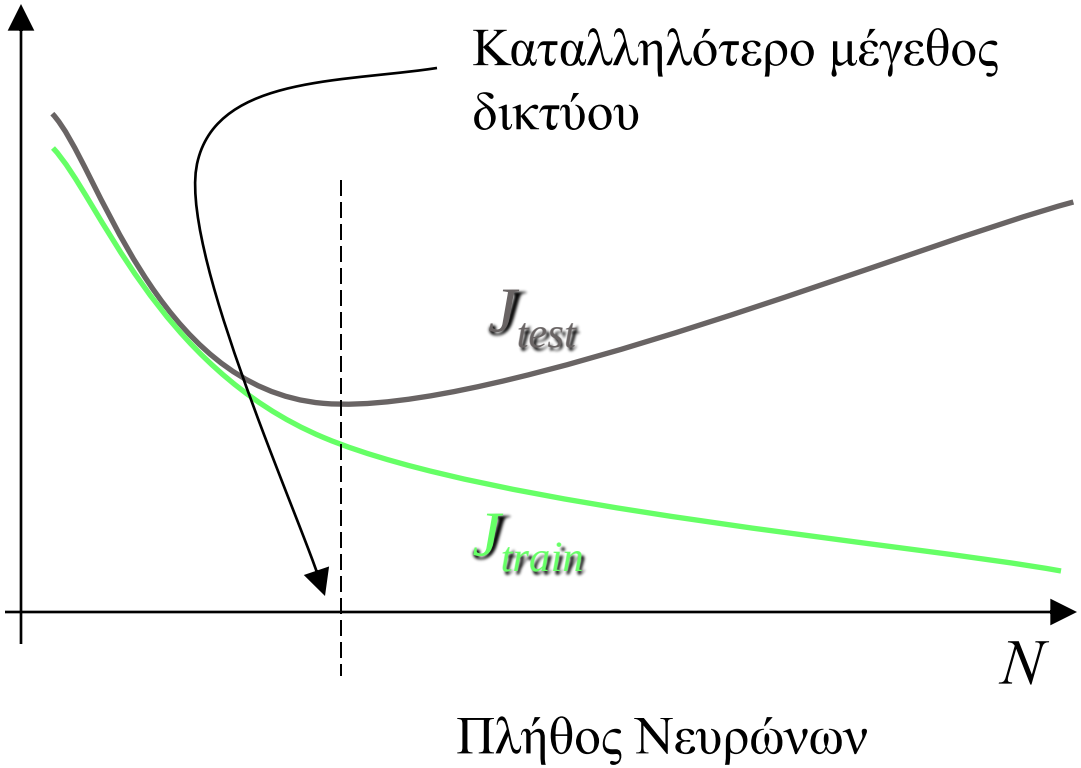
$$J_{test} = \sum_{t=1}^{T} \sum_{i=1}^{M} (d_i^{(t)} - y_i^{(t)})^2$$

T = το πλήθος των προτύπων στα οποία **δεν** εκπαιδεύτηκε το δίκτυο αλλά τα χρησιμοποιούμε για έλεγχο γενίκευσης (test set)

M = το πλήθος των εξόδων του δικτύου

$d_i^{(t)}, y_i^{(t)}$ = στόχοι και έξοδοι του δικτύου για το πρότυπο t

Επιλογή μεγέθους δικτύου



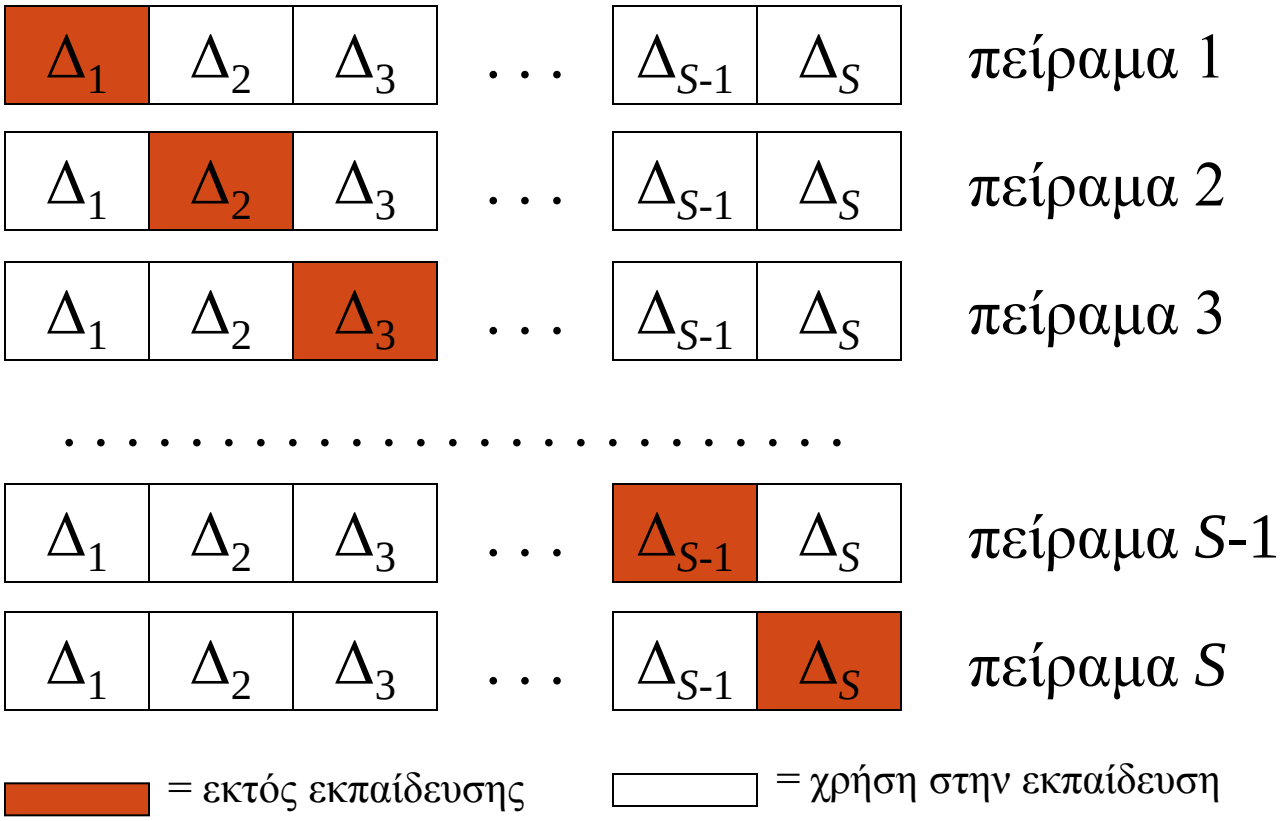
Η μέθοδος της διασταύρωσης (cross-validation)

- Διαθέτουμε πεπερασμένα δεδομένα : P ζεύγη $[\mathbf{x}^{(p)},\mathbf{d}^{(p)}]$, $p = 1, 2, \ldots, P$.
- Τεμαχίζουμε το σύνολο των δεδομένων σε S ισομεγέθη τμήματα, $\Delta_1, \Delta_2, \ldots, \Delta_S$.
- Εκπαιδεύουμε το δίκτυο χρησιμοποιώντας τα δεδομένα από τα $S-1$ τμήματα και αφήνουμε το τμήμα που δε χρησιμοποιήθηκε για έλεγχο (testing)
- Υπολογίζουμε το σφάλμα ελέγχου J_{test} για το συγκεκριμένο πείραμα.

Η μέθοδος της διασταύρωσης (cross-validation) (2)

- Επαναλαμβάνουμε το πείραμα αφήνοντας κάποιο άλλο τμήμα για έλεγχο, δηλ. δεν το χρησιμοποιούμε στην εκπαίδευση. Χρησιμοποιούμε τα υπόλοιπα $S-1$ τμήματα για εκπαίδευση ξανά από την αρχή.
- Υπολογίζουμε ξανά το σφάλμα ελέγχου J_{test} για το συγκεκριμένο πείραμα.
- Επαναλαμβάνουμε το πείραμα S φορές αφήνοντας κάθε φορά και άλλο τμήμα για έλεγχο.
- Παίρνουμε το μέσο όρο J_{mean} των σφαλμάτων ελέγχου J_{test} από όλα τα S πειράματα.

Η μέθοδος της διασταύρωσης (cross-validation) (3)



Η μέθοδος της διασταύρωσης (cross-validation) (4)

- Επαναλαμβάνουμε ολόκληρη τη μέθοδο για ένα άλλο δίκτυο με περισσότερους ή λιγότερους νευρώνες και υπολογίζουμε το μέσο όρο των σφαλμάτων ελέγχου J'_{mean}
- Τυπώνουμε την καμπύλη του μέσου σφάλματος ελέγχου σαν συνάρτηση του πλήθους των νευρώνων
- Επιλέγουμε το δίκτυο εκείνο με το μικρότερο μέσο σφάλμα ελέγχου.

Η μέθοδος της διασταύρωσης (cross-validation) (5)

- Πλεονέκτημα της μεθόδου = η καλή εκτίμηση του κατάλληλου μεγέθους του δικτύου
- Λειτουργεί και με λίγα δεδομένα.
- Μειονέκτημα = οι πολλές επαναλήψεις. Εκπαιδεύουμε κάθε δίκτυο S φορές.
- Συνήθης επιλογή $S = 10$. Αλλά μπορεί να χρησιμοποιηθεί και $S = P =$ όλα τα δεδομένα

Κανονικοποίηση (Regularization)

- Για την αποφυγή υπερ-μοντελοποίησης προσθέτουμε ένα επί πλέον όρο Ω στο σφάλμα εκπαίδευσης J. Ο όρος αυτός είναι μεγάλος όταν το δίκτυο είναι μεγάλο και μικρός όταν το δίκτυο είναι μικρό.

- Η σχετική «βαρύτητα» του όρου αυτού ρυθμίζεται από την παράμετρο κανονικοποίησης λ.
- Έτσι το δίκτυο μαθαίνει να αποφεύγει τους πολλούς νευρώνες.

$$J = J + \lambda\Omega$$

Κανονικοποίηση (Regularization) (2)

- Παράδειγμα:

$$\Omega = \frac{1}{2}\sum_i w_i^2$$

w_i = τα συναπτικά βάρη του δικτύου

- Τιμωρούμε τα μεγάλα βάρη
- Αν τα βάρη του δικτύου είναι 0 είναι σα να βγάζουμε τους νευρώνες εκτός λειτουργίας.

Κανονικοποίηση (Regularization) (3)

- Παράδειγμα (συνέχεια): Τετραγωνικό σφάλμα με κανονικοποίηση

$$J = \sum_{p=1}^{P} \left\| \mathbf{d}^{(p)} - \mathbf{y}^{(p)} \right\|^2 + \frac{\lambda}{2} \sum_i w_i^2$$

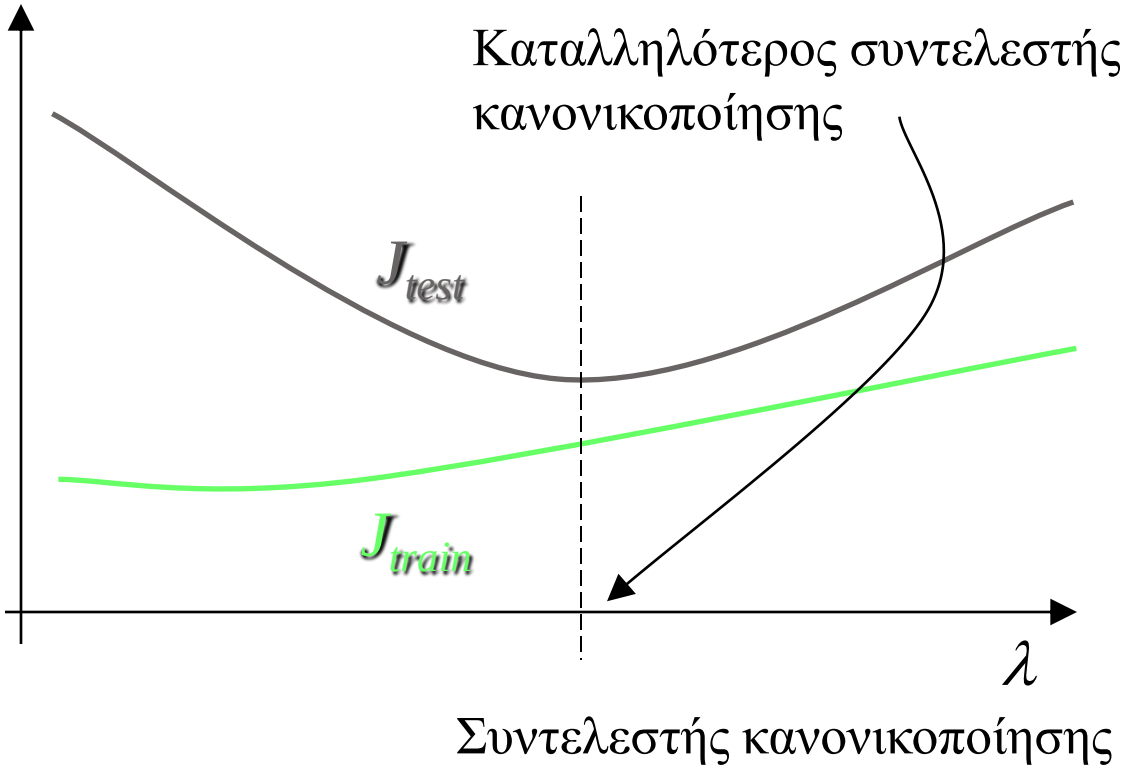
- Κανόνας Back-Propagation με κανονικοποίηση

$$w_{ij}(k+1) = w_{ij}(k) + \beta \Delta w_{ij} - \lambda w_{ij}(k)$$

όπως ο κανονικός BP

όρος κανονικοποίησης

Επίδραση της κανονικοποίησης



Επαύξηση και κλάδεμα

- Επαύξηση (growing): πρόσθεση κόμβων σε ένα ήδη υπάρχον δίκτυο
- Κλάδεμα (pruning): αφαίρεση κόμβων από ένα δίκτυο

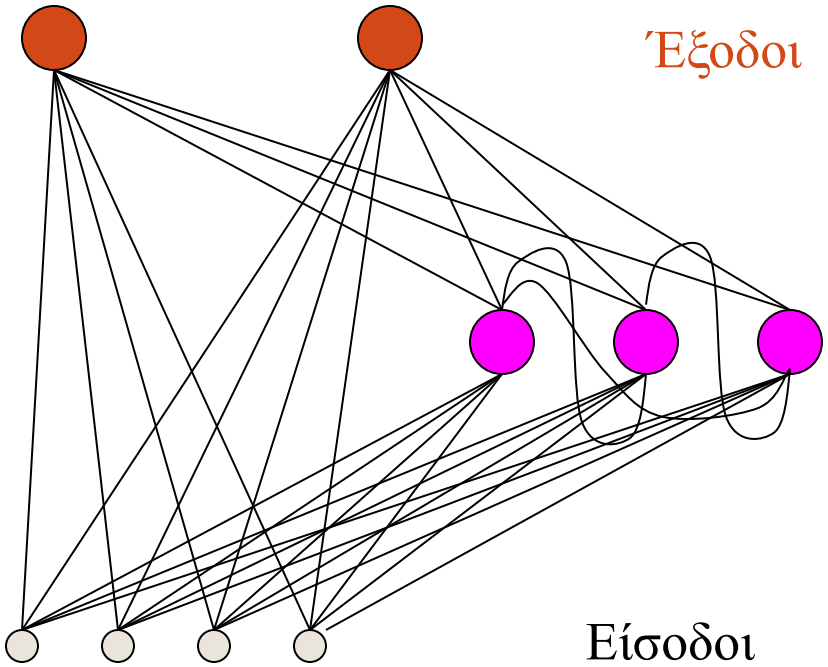
Μέθοδοι επαύξησης

- Η πιο απλή μέθοδος και πιο συχνά χρησιμοποιούμενη
- Ορίζουμε ένα δίκτυο MLP δύο στρωμάτων με M κρυφούς νευρώνες.
- Εκπαιδεύουμε το δίκτυο
- Αυξάνουμε τους κρυφούς νευρώνες σταδιακά και το ξαναεκπαιδεύουμε
- Σταματάμε όταν ικανοποιηθεί ένα συνολικό κριτήριο τερματισμού. Π.χ. όταν το σφάλμα ελέγχου J_{test} είναι κάτω από κάποιο όριο.

Η μέθοδος cascade-correlation

- Εναλλακτική μέθοδος κατασκευής ενός δικτύου
- Ξεκινάμε ενώνοντας όλες τις εισόδους με όλες τις εξόδους - χωρίς κρυφό στρώμα
- Εκπαιδεύουμε το δίκτυο αυτό
- Προσθέτουμε ένα κρυφό νευρώνα
- Βρίσκουμε τα βάρη που ενώνουν τις εισόδους με τον κρυφό νευρώνα χρησιμοποιώντας ένα κανόνα συσχέτισης
- Παγώνουμε τα βάρη του κρυφού νευρώνα και εκπαιδεύουμε τα βάρη που ενώνουν το κρυφό νευρώνα με τις εξόδους καθώς και τα βάρη που ενώνουν τις εισόδους με τις εξόδους
- Συνεχίζουμε προσθέτοντας κρυφούς νευρώνες μέχρι να ικανοποιηθεί κάποιο συνολικό κριτήριο τερματισμού

Cascade-correlation (2)



Μέθοδοι κλαδέματος

Κεντρική ιδέα:

- Ξεκινάμε εκπαιδεύοντας ένα μεγάλο δίκτυο.
- Αφού αυτό συγκλίνει αποφασίζουμε με κάποιο τρόπο ποιούς κόμβους ή συναπτικά βάρη θα αφαιρέσουμε
- Αφαιρούμε τους κόμβους ή τα συναπτικά βάρη και ξαναεκπαιδεύουμε το δίκτυο μέχρι να ικανοποιηθεί κάποιο συνολικό κριτήριο τερματισμού

Κλάδεμα βαρών

Μέθοδος 1: Εξαφάνιση βαρών (weight elimination)

- Χρήση κανονικοποίησης

$$\Omega = \frac{1}{2} \sum_{i} \frac{w_i^2}{c^2 + w_i^2}$$

c = μια σταθερά που επιλέγουμε εμείς

- Εκπαιδεύουμε το δίκτυο κανονικά με τον παραπάνω όρο Ω της κανονικοποίησης
- Αφαιρούμε όλα τα βάρη που είναι μικρότερα από c.

Κλάδεμα βαρών (2)

Μέθοδος 2: Σημαντικότητα βαρών (weight saliency)

- Εκπαιδεύουμε ένα μεγάλο δίκτυο μέχρι να συγκλίνει
- Μετά τη σύγκλιση βρίσκουμε τη «σημαντικότητα» του κάθε συναπτικού βάρους υπολογίζοντας τη διαφορά

$$s_{ij} = J(\text{χωρίς το βάρος } w_{ij}) - J(\text{με το βάρος } w_{ij})$$

- Αφαιρούμε τα βάρη με τη μικρότερη σημαντικότητα
- Είτε αφαιρούμε τα βάρη των οποίων η σημαντικότητα είναι μικρότερη από κάποιο όριο (κατώφλι) που ορίζουμε εμείς.
- Πρόβλημα: πολύς υπολογιστικός φόρτος

Κλάδεμα βαρών (3)

Μέθοδος 3: Βέλτιστη εγκεφαλική βλάβη (optimal brain damage)

- Όμοια με τη μέθοδο 2. Βασίζεται στη σημαντικότητα των βαρών την οποία δεν υπολογίζει ακριβώς αλλά προσεγγιστικά. Χρήση των στοιχείων H_{ij} του πίνακα Hess

$$H_{ij} = \frac{\partial^2 J}{\partial w_i \partial w_j}$$

- Τα στοιχεία H_{ij} υπολογίζονται με κάποιο τύπο

Κλάδεμα βαρών (4)

Μέθοδος 3: Βέλτιστη εγκεφαλική βλάβη (συνέχεια)

- Εκπαιδεύουμε ένα μεγάλο δίκτυο μέχρι να συγκλίνει
- Για κάθε συναπτικό βάρος w_i υπολογίζουμε προσεγγιστικά τη σημαντικότητα από τον τύπο

$$\frac{1}{2} H_{ii} w_i^2$$

- Ταξινομούμε τα βάρη με βάση τη σημαντικότητά τους
- Αφαιρούμε τα βάρη με τη μικρότερη σημαντικότητα
- Επαναλαμβάνουμε από την αρχή τη μέθοδο μέχρι να ικανοποιηθεί κάποιο συνολικό κριτήριο τερματισμού

Κλάδεμα νευρώνων

- Βρίσκουμε τη «σημαντικότητα» του κάθε νευρώνα υπολογίζοντας τη διαφορά

  s_i = J(χωρίς το νευρώνα i) - J(με το νευρώνα i)
- Αν η αύξηση του κόστους J χωρίς το νευρώνα i είναι μηδαμινή τότε ο νευρώνας μπορεί να αποκοπεί.

Κλάδεμα νευρώνων (2)

Εναλλακτικά:

- Χρησιμοποιούμε μια συνάρτηση ενεργοποίησης f τέτοια ώστε $f(0)=0$

  πχ. $f(u) = \tanh(u) = (e^{u}-e^{-u})/(e^{u}+e^{-u})$
- Με κάθε νευρώνα i συνδέουμε μια μεταβλητή γ_i μεταξύ 0 και 1 η οποία είναι 0 αν θέλουμε να πετάξουμε το νευρώνα ή 1 αν θέλουμε να τον κρατήσουμε.
- Η ενεργοποίηση του νευρώνα i γίνεται

$$a_i = f\left(\gamma_i\left(\sum_j w_{ij}a_j + w_{i0}\right)\right)$$

Κλάδεμα νευρώνων (3)

Αν $\gamma_i = 0$ τότε

Αν $\gamma_i = 1$ τότε $a_i = f(0) = 0$

ο νευρώνας είναι σα να μην υπάρχει

$$a_i = f(\sum_j w_{ij} a_j + w_{i0})$$

ο νευρώνας είναι όπως ένας κανονικός νευρώνας

Κεντρική ιδέα:

- Οι συντελεστές γ_i να εκπαιδεύονται όπως και τα συναπτικά βάρη. Επέκταση του κανόνα Back-Propagation

Επιτροπές δικτύων (committees of networks)

- Κεντρική ιδέα: να εκπαιδεύσουμε πολλά δίκτυα με τα ίδια δεδομένα. Όταν μας δίνεται ένα πρότυπο εισόδου υπολογίζουμε την έξοδο παίρνοντας το μέσο όρο των εξόδων όλων των δικτύων.
- Τα διαφορετικά δίκτυα μπορεί να έχουν διαφορετικό πλήθος νευρώνων, διαφορετική τοπολογία ακόμα και διαφορετικό κανόνα εκπαίδευσης

Επιτροπές δικτύων (2)

- Έστω $J_1, J_2, \ldots, J_N$ τα σφάλματα εκπαίδευσης είτε τα σφάλματα ελέγχου των N δικτύων.
- Αποδεικνύεται ότι το σφάλμα J_{com} της επιτροπής είναι μικρότερο από το μέσο σφάλμα των δικτύων

$$J_{com} < \frac{1}{N}\sum_{i=1}^{N} J_i$$