



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας

Τμήμα Ηλεκτρονικών Μηχανικών

ΧΡΗΣΙΜΕΣ ΕΝΤΟΛΕΣ ΓΙΑ ΤΟ NEURAL TOOLBOX ΤΟΥ MATLAB

Επιμέλεια:

*Αλεξανδρίδης Αλέξανδρος
Αναπληρωτής Καθηγητής TEI Αθήνας*

Τυχαία αναδιάταξη αριθμών (permutation)

`perm=randperm(n)`; όπου:

n: Ένας ακέραιος αριθμός

perm: Μια τυχαία αναδιάταξη όλων των ακεραίων αριθμών από 1 έως n

Κανονικοποίηση δεδομένων ανάμεσα σε σταθερές τιμές (-1 και 1)

- **Απ' ευθείας κανονικοποίηση:**

`[xsc,ps_xsc] = mapminmax(x)`;

όπου:

x: ο πίνακας με τα δεδομένα εισόδου (ή εξόδου) (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

xsc: ο πίνακας με τα κανονικοποιημένα δεδομένα εισόδου (ή εξόδου) (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

ps_xsc: Δομή που περιέχει τις σταθερές κανονικοποίησης για την είσοδο (ή την έξοδο)

- **Μετασχηματισμός νέων δεδομένων βάση δοσμένων σταθερών κανονικοποίησης:**

`xnsc = mapminmax('apply',xn,ps_xsc)`;

όπου:

xn: ο πίνακας με τα νέα δεδομένα εισόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

xnsc: ο πίνακας με τα μετασχηματισμένα νέα δεδομένα εισόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

ps_xsc: Δομή που περιέχει τις ήδη υπολογισμένες σταθερές κανονικοποίησης για την είσοδο

- **Απο-κανονικοποίηση:**

```
y = mapminmax('reverse', ysc, ps_ysc);
```

όπου:

ysc: ο πίνακας με τα κανονικοποιημένα δεδομένα εξόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

y: ο πίνακας με τα αποκανονικοποιημένα δεδομένα εξόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

ps_ysc: Δομή που περιέχει τις ήδη υπολογισμένες σταθερές κανονικοποίησης για την έξοδο

Κανονικοποίηση δεδομένων με μηδενικό μέσο και μοναδιαία διασπορά

Στη περίπτωση αυτή χρησιμοποιείται η εντολή `mapstd`, χρησιμοποιώντας ακριβώς το ίδιο syntax με την εντολή `mapminmax`.

Νευρωνικά δίκτυα MLP – Προσέγγιση συναρτήσεων

- **Δημιουργία νευρωνικού δικτύου MLP**

```
net = feedforwardnet(hl);
```

όπου:

hl: διάνυσμα με τους αριθμούς των νευρώνων ανά κρυφή στοιβάδα. Π.χ. για ένα δίκτυο με δύο κρυφές στοιβάδες και 10 νευρώνες στην πρώτη κρυφή στοιβάδα και 5 στη δεύτερη, `hl=[10 5]`

net: αντικείμενο που περιέχει το νευρωνικό δίκτυο που έχει δημιουργηθεί

- **Επισκόπηση νευρωνικού δικτύου MLP**

`view(net)`

όπου:

net: αντικείμενο που περιέχει το νευρωνικό δίκτυο

- **Εκπαίδευση νευρωνικού δικτύου MLP**

`[net,tr] = train(net, xsc, ysc);`

όπου:

net: αντικείμενο που περιέχει το νευρωνικό δίκτυο

xsc: πίνακας με τα κανονικοποιημένα δεδομένα εισόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

ysc: πίνακας με τα κανονικοποιημένα δεδομένα εξόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

tr: Δομή που περιέχει στατιστικά στοιχεία για την εκπαίδευση

- **Καθορισμός δεδομένων εκπαίδευσης-αξιολόγησης-ελέγχου (για χρήση κατά την εκπαίδευση του δικτύου)**

`net.divideFcn = 'divideind';`

`net.divideParam.trainInd = [a:b];`

`net.divideParam.valInd = [c:d];`

`net.divideParam.testInd = [e:f];`

όπου:

a,b: δείκτες που δείχνουν το σημεία που αρχίζουν και τελειώνουν αντίστοιχα τα δεδομένα εκπαίδευσης μέσα στους πίνακες `xsc`, `ysc`

c,d: δείκτες που δείχνουν το σημεία που αρχίζουν και τελειώνουν αντίστοιχα τα δεδομένα αξιολόγησης μέσα στους πίνακες `xsc`, `ysc`

e,f: δείκτες που δείχνουν το σημεία που αρχίζουν και τελειώνουν αντίστοιχα τα δεδομένα ελέγχου μέσα στους πίνακες `xsc`, `ysc`

- **Χρήση νευρωνικού δικτύου MLP για προβλέψεις**

`ysc=net(xsc);`

όπου:

net: αντικείμενο που περιέχει το νευρωνικό δίκτυο

xsc: πίνακας με τα κανονικοποιημένα δεδομένα εισόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

ysc: πίνακας με τις κανονικοποιημένες προβλέψεις (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

Νευρωνικά δίκτυα MLP – Αναγνώριση προτύπων

- **Δημιουργία νευρωνικού δικτύου MLP**

`net = patternnet (hl);`

όπου:

hl: διάνυσμα με τους αριθμούς των νευρώνων ανά κρυφή στοιβάδα. Π.χ. για ένα δίκτυο με δύο κρυφές στοιβάδες και 10 νευρώνες στην πρώτη κρυφή στοιβάδα και 5 στη δεύτερη, `hl=[10 5]`

net: αντικείμενο που περιέχει το νευρωνικό δίκτυο που έχει δημιουργηθεί

- **Επισκόπηση νευρωνικού δικτύου MLP**

`view(net)`

όπου:

net: αντικείμενο που περιέχει το νευρωνικό δίκτυο

- **Εκπαίδευση νευρωνικού δικτύου MLP**

`[net,tr] = train(net, xsc, y);`

όπου:

net: αντικείμενο που περιέχει το νευρωνικό δίκτυο

xsc: πίνακας με τα κανονικοποιημένα δεδομένα εισόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

y: πίνακας με τα δεδομένα εξόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές). Κάθε μια μεταβλητή εξόδου (σειρές) αντιστοιχεί με μια από τις κλάσεις του προβλήματος και στην κάθε στήλη (δεδομένο) θα πρέπει να υπάρχει μια μονάδα στην κλάση στην οποία ανήκει πραγματικά το συγκεκριμένο δεδομένο και μηδέν σε όλες τις υπόλοιπες κλάσεις.

tr: Δομή που περιέχει στατιστικά στοιχεία για την εκπαίδευση

ΠΡΟΣΟΧΗ: Σε προβλήματα με υψηλές απαιτήσεις μνήμης θα πρέπει οπωσδήποτε της εντολής *train* να προηγείται η εντολή *net.trainFcn= 'trainscg'*

- **Καθορισμός δεδομένων εκπαίδευσης-αξιολόγησης-ελέγχου (για χρήση κατά την εκπαίδευση του δικτύου)**

```
net.divideFcn = 'divideind';
```

```
net.divideParam.trainInd = [a:b];
```

```
net.divideParam.valInd = [c:d];
```

```
net.divideParam.testInd = [e:f];
```

όπου:

a,b: δείκτες που δείχνουν το σημεία που αρχίζουν και τελειώνουν αντίστοιχα τα δεδομένα εκπαίδευσης μέσα στους πίνακες *xsc*, *ysc*

c,d: δείκτες που δείχνουν το σημεία που αρχίζουν και τελειώνουν αντίστοιχα τα δεδομένα αξιολόγησης μέσα στους πίνακες *xsc*, *ysc*

e,f: δείκτες που δείχνουν το σημεία που αρχίζουν και τελειώνουν αντίστοιχα τα δεδομένα ελέγχου μέσα στους πίνακες *xsc*, *ysc*

- **Χρήση νευρωνικού δικτύου MLP για προβλέψεις**

```
yp=net(xsc);
```

όπου:

net: αντικείμενο που περιέχει το νευρωνικό δίκτυο

xsc: πίνακας με τα κανονικοποιημένα δεδομένα εισόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)

yp: πίνακας με τις προβλέψεις (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές). Κάθε μια μεταβλητή εξόδου (σειρές) αντιστοιχεί με μια από τις κλάσεις του προβλήματος. Θεωρούμε ότι το συγκεκριμένο δεδομένο (στήλη) ταξινομήθηκε από το νευρωνικό δίκτυο στην κλάση (σειρά) στην οποία η έξοδος είναι μέγιστη.

- **Κατασκευή confusion matrix**

`plotconfusion(y,yp)`

όπου:

y: πίνακας με πραγματικά δεδομένα εξόδου (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές). Κάθε μια μεταβλητή εξόδου (σειρές) αντιστοιχεί με μια από τις κλάσεις του προβλήματος και στην κάθε στήλη (δεδομένο) θα πρέπει να υπάρχει μια μονάδα στην κλάση στην οποία ανήκει πραγματικά το συγκεκριμένο δεδομένο και μηδέν σε όλες τις υπόλοιπες κλάσεις.

yp: πίνακας με τις προβλέψεις (τα δεδομένα σε στήλες και οι μεταβλητές σε σειρές)