

1) Stringe
2) Matrici di celle
3) Strutture
4) File
5) Grafici

Ivlin.Stoianov@gmail.com

Stringhe

stringa / testo = un **vettore-riga di lettere** (rinchiuso in '')

```
S='The quick lazy fox';
```

```
S(1)
```

```
T
```

```
S(1:3)
```

```
The
```

```
S(2,1:21)='La veloce pigra volpe'
```

```
The quick lazy fox
```

```
La veloce pigra volpe
```

Ivlin.Stoianov@gmail.com

Operazioni su stringhe

- `[str1 str2]` concatena due stringhe
- `str(i:j)` sotto-stinga
- `str(i:end)` -- // --
- `I=findstr(s1,s2)` cerca dove si trova **s1** in **s2**
- `strcmp(s1,s2)` verifica se $s1 = s2$
- `N=str2num(str)` converte stringa con numero in numeri
- `S=num2str(N)` converte una matrice numerica in str.
- `str{+ - * /}` operazioni algebriche su stringhe:
le lettere vengono convertite in numeri
(codice ASCII) e poi si applica la operazione
- `char(v)` converte una matrice numerica
(codice ASCII) in una stringa di lettere

lvilin.Stoianov@gmail.com

Stringhe - Esempi

Compito: che cosa fanno questi espressioni ?

```
N=7; ['Soggetto numero ' num2str(N)]
```

```
S='13.14'; num2str(S(2:end))
```

```
'matrix'+1
```

```
char('matrix'+1)
```

```
char('lettere->numeri->lettere'+10-10)
```

`repmat(A,m,n)` replica una matrice
(*m* volte in verticale / *n* volte in orizzontale)

```
tril(repmat('matrix',6,1))
```

lvilin.Stoianov@gmail.com

Matrici di celle

[25; 'testo'; [1 2 3]] -> ERR

Cella {X}: un contenitore di dati di diversi tipo (scalari, matrici, stringhe ecc.) interpretati come un singolo elemento.

C1={'testo'};

C2={25};

C3=[1 2 3];

Matrice di celle: insieme di contenitori di dati **riferiti per indice**.

Costruzione

- M = [C1; C2; C3]; -> OK
- M = {'testo';25;[1 2 3]};

Riferimento

- M(1) = {'testo'}; (cella)
- M{1} = 'testo'; (il contenuto nella cella)

lvilin.Stoianov@gmail.com

Celle - manipolazioni

- Visto che una matrice di celle di solito contiene elementi di diversi tipi, in generale non ci sono operazioni matriciali su matrici di celle.
- Quindi, i dati in una matrice di celle vengono **elaborati in cicli**.
- Se invece i dati sono di un solo tipo (numeri o testi), si possono convertire in matrici "normali" per poter elaborarli in modo matriciale

• **num2cell**(M) matrice di numeri → matrice di celle

• **cell2mat**(C) matrice di celle → matrice di numeri

• **cellstr**(S) matrice di lettere S → matrice di celle;
ogni riga (cioè, stringa) va inclusa in una cella.

• **char**(C) matrice di celle → matrice di stringhe

lvilin.Stoianov@gmail.com

Matrici di Celle – Esercizio

(A) Creare una matrice di celle che contiene:

- in prima colonna, ma in separate righe, i nomi di 3 funzioni di matlab;
- in seconda colonna, breve descrizione di queste funzioni
- in terza colonna, esempi di variabili d'ingresso (parametri)
- in quarta colonna, esempi di risultato

(B) Visualizza, con un ciclo, l'informazione relativa ad ogni funzione descritta

```
Doc(1,1:4)={'eye','matrice diagonale',4,eye(4)};
Doc(2,1:4)=[{'sum'},{'somma degli elementi'},{[1 2 3]},{sum([1 2 3])}];

for i=1:size(Doc,1)
    fprintf('\nFunzione %s\n',Doc{i,1});
    fprintf('%s\n',Doc{i,2});
    fprintf('Esempio: %s(%s)=\n',Doc{i,1},sprintf('%d ',Doc{i,3}));
    disp(Doc{i,4});
end
```

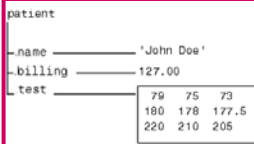
lvilin.Stoianov@gmail.com

Strutture

- **Contenitore** di **dati** diversi tipi (anche strutture!)
riferiti per nome (**campi**)

<Nome-della-struttura> . <nome-dell-campo>

- S.Nsubj=2;
- S.Name= 'John' ;
- S.Score=[4.5, 9.0];



| | | | | | | | | | | |
|----------|--|-------|----|----|-----|-----|-------|-----|-----|-----|
| patient | | | | | | | | | | |
| -name | 'John Doe' | | | | | | | | | |
| -billing | 127.00 | | | | | | | | | |
| -test | <table border="1"><tr><td>79</td><td>75</td><td>73</td></tr><tr><td>180</td><td>178</td><td>177.5</td></tr><tr><td>220</td><td>210</td><td>205</td></tr></table> | 79 | 75 | 73 | 180 | 178 | 177.5 | 220 | 210 | 205 |
| 79 | 75 | 73 | | | | | | | | |
| 180 | 178 | 177.5 | | | | | | | | |
| 220 | 210 | 205 | | | | | | | | |

- Una struttura è anche matrice:

```
S(1).Name='John';
S(2).Name='Marry';
```

lvilin.Stoianov@gmail.com

Strutture - utilizzo

- Archiviare dati (*database*)
`S(i).Name = 'John';`
(anche le matrici di celle si usano come *database*)
`C{i,1} = 'John';`
- *Incapsulare* le variabili che descrivono un concetto (oggetto in OOP)
`Exp.Subjects = {'John', 'Marry'};`
`Exp.Method.ITI = 2500;`
`Exp.Method.SOA = 120;`

Ivilin.Stoianov@gmail.com

Strutture/Celle - Esempio

I dati di un esperimento

| | |
|---------------------------|--|
| <code>Exp.nSubj</code> | numero N di soggetti |
| <code>Exp.SubjName</code> | I nomi dei soggetti (matrice di celle) |
| <code>Exp.Gender</code> | il sesso dei soggetti (vettore-colonna di lettere) |
| <code>Exp.RT</code> | tempi di reazioni in 4 condizioni – una matrice con 4 colonne |

... messi insieme in una matrice di celle

```
>> C=[Exp.SubjName cellstr(Exp.Gender) num2cell(Exp.RT)];
```

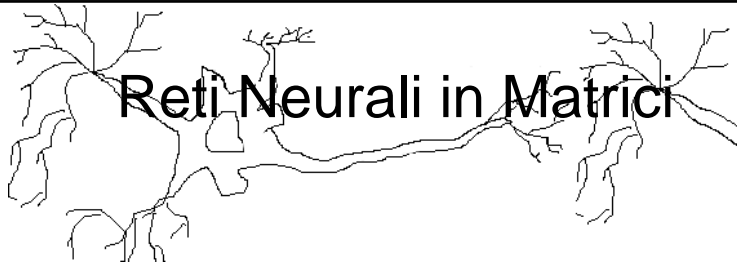
... ed esportati in una cartella d'Excel

```
>> xlswrite('MyExp.xls',C,'RT');
```

La funzione che esporta una matrice di celle in Excel:

```
xlswrite(FileName,CellArray,SheetName,Range)
```

Ivilin.Stoianov@gmail.com



Reti Neurali in Matrici

- **Neuroni** (rate-coding) strutturati in strati (vettori)
 $X^{N_{inp}} \quad Y^{N_{hidd}} \quad \dots \quad Z^{N_{out}}$
 X: ingresso(sensori) → Y: neuroni interni → Z: uscita(effettori)
- **Connessioni** ponderati (grandezza dei sinapsi) in matrici
 W_{XY}, W_{YY}, W_{YZ}
- **Attivazione** di una rete:
 - attivazione dei sensori assegnare valori a X
 - attivare i neuroni interni $Y = \text{sigm}(X * W_{XY} + Y * W_{YY});$
 - attivare gli effettori $Z = \text{sigm}(Y * W_{YZ});$

lvilin.Stoianov@gmail.com

```
function y=sigm(x)
y = 1.0 ./ (exp(-x)+1);
```

Strutture - Esempio

Una rete neurale

- Net.X
- Net.Y
- Net.Z
- Net.Wxy
- Net.Wyy
- Net.Wyz

```
function Z=NetActivation(X)
global Net;
Net.X=X;
Net.Y=sigm(Net.X*Net.Wxy+Net.Y*Net.
Wyy);
Net.Z=sigm(Net.Y*Net.Wyz);
Z=Net.Z;
```

Qui la struttura *Net* incapsula tutte le variabili che descrivono una rete neurale.

lvilin.Stoianov@gmail.com

Strutture Dati - Esercizio

- (A) Progettare la struttura di dati di un vostro problema scientifico a scelta.
- (B) Inserire gli informazioni di un caso d'esempio.

lvilin.Stoianov@gmail.com

File – Codifica in formato proprio

- `save(fname)` `save <fname>`
- `save(fname, 'V1', 'V2')` `save <fname> V1,V2,...`

- `load(fname)` `load <fname>`
- `load(fname, 'V1', 'V2')` `load <fname> V1,V2,...`

- `pwd`
- `filesep '\\'`
[filePath filesep fileName]

Esercizio:

```
>> x=1;  
>> save('test',x);  
>> clear all;  
>> load('test');  
>> whos  
>> disp(x);
```

lvilin.Stoianov@gmail.com

Utilizzare altri tipi di *File*

EXCEL

- `xlswrite(FileName, CellArray, SheetName, Range)`
- `[N, T, C]=xlsread(FileName, SheetName, Range)`

Menu: *File*→*Import Data* importare dati

IMAGES

- `I=imread(fname) imwrite(I, fname, fmt)`
`imformats`

AUDIO

- `[A, Fs, N]=wavread(fname)`
`wavwrite(y, Fs, N, fname)`

VIDEO

- `M=aviread(fname)` lvilin.Stoianov@gmail.com `MovieReader(fname)`

Grafici

- `figure(n)` crea una figura vuota (numero *n*)
- `plot(Y)` visualizza il *vettore* *Y* come una serie di punti (*i, Y_i*) collegati con linee.
- `plot(X, Y)` X-Y (scatter) plot, dove *X* & *Y* sono vettori
- `plot(X, <formato>)` visualizzazione con formato
 <formato>: '*<colore><punti><linea>*'
 Esempi: 'r.' punti rossi
 'r.' linea rossa tratteggiata
 'r-*' stelle collegate con linea tratteggiata
- `plot(M)` visualizza **le colonne** in *M* con **linee separate**
- `semilogx(X)` plot su scala logaritmica di *x* e lineare di *y*
- `loglog(X)` plot su scala logaritmica di entrambi *x* e *y*
- `line(X, Y)` crea linee collegate definite dai coordinati *X* e *Y*

lvilin.Stoianov@gmail.com

Assi

- `axis([xmin xmax ymin ymax])` pone i limiti delle asse
- `V=axis` riporta i limiti attuali
- `axis auto` limiti secondo i dati
- `axis xy` l'origine: in basso a sx
- `axis ij` l'origine: in alto a sx
- `axis equal` dx=dy
- `axis image` dx=dy; stretto
- `axis square` fa il grafico quadrato
- `axis off` fa sparire le asse
- `axis on` fa apparire le asse

Ivilin.Stoianov@gmail.com

Titoli

- `title(<titolo>)` titolo di un (sotto)grafico
- `xlabel(<titolo>)` etichetta della asse X
- `ylabel(<titolo>)` etichetta della asse Y
- `text(x,y,'testo')` mostra un testo alla posiz. (x,y)
- `legend(S)` legenda degli elementi grafici.
S: matr. di celle con stringhe
- `colorbar` una bara con il mapping colori - numeri visualizzati

Ivilin.Stoianov@gmail.com

Grafici – Esempi

```
T1=[0:.05:10];% tempo a passo lento
Y1=sin(T1); % una senoide
Y2=cos(T1); %
figure(1); % fare una figura vuota
plot(T1,Y1); % grafico della senoide

T2=[0:.5:10];% tempo a passo veloce
plot(T,Y1,'o');
title('Sin(x)');% aggiungere titolo del grafico
xlabel('t'); % indicare che cosa significano x e y
ylabel('y');
```

lvilin.Stoianov@gmail.com

Grafici speciali

- `pcolor(M)` visualizza con colori i valori di una matrice M
- `surface(M)` visualizza M come una superficie 3D
- `mesh(M)` visualizza M come una griglia 3D
- `image(I)` visualizza come immagine una matrice I di dimensioni (m,n) o $(m,n,3)$
- `hist(X,n)` istogramma con n bin dei dati in X .
Se X è matrice, un istogr. per colonna.
- `cellplot(S)` visualizzazione grafica di una matrice di celle

lvilin.Stoianov@gmail.com

Figure complesse

- `subplot(r,c,i)` crea $r \times c$ sotto-grafici e seleziona il grafico numero i per i plot successivi.
- `axes('position',[X,Y,Xsize,Ysize])`
crea un grafico al posto indicato;
le misure sono relative [0 - 1]
- `hold on/off` mantenere **si** o **no** gli elementi grafici creati fino a questo momento
- `cla` pulizia del plot attuale
- `clf` pulizia della figura attuale
- `h=gcf` indice (handle) della figura attuale
- `h=gca` indice (handle) del plot attuale

lvilin.Stoianov@gmail.com

Grafici – Esempi

- scaricare da internet un'immagine jpeg
- caricare l'immagine con `l=imread('xxx.jpg');`
- visualizzarla in un subplot: `subplot(1,3,1);`
- visualizzare la istogramma di ogni suo colore
`lr=single(l(:,:,1));`
`subplot(1,3,2);`
`hist(lr(:,100));`
- fare elaborazione
- visualizzarla di nuovo in un altro subplot

lvilin.Stoianov@gmail.com

Progetto per il prossimo incontro

1) Quale problema risolvere?

Esempi:

- Pre-elaborazione dati per un successivo analisi statistico
- Simulazione con reti neurali
- Preparazione dei stimoli per un esperimento

2) Progettare la struttura generale dell'elaborazione da fare

- Individuare sotto-problemi
- Individuare variabili/strutture per la codifica dei dati
- Progettare l'interfaccia con l'utente

lvilin.Stoianov@gmail.com