

Indicizzazione Lineare delle Matrici (Linear Indexing).

È possibile accedere agli elementi di una matrice in matlab utilizzando un'indicizzazione lineare. Questo dipende dal fatto in Matlab, una matrice viene rappresentata come un vettore, anche se all'utente questo fatto non è trasparente: il vettore è formato dal concatenamento delle colonne della matrice.

Vediamo un esempio:

A=

```
0.9501  0.4860  0.4565  0.4447
0.2311  0.8913  0.0185  0.6154
0.6068  0.7621  0.8214  0.7919
```

ecco come viene eseguito il concatenamento

```
0.9501
0.2311
0.6068
0.4860
0.8913
0.7621
0.4565
0.0185
0.8214
0.4447
0.6154
0.7919
```

Data un matrice di dimensioni $n \times m$, l'elemento (h,k) viene individuato dall'indice lineare

$$(k-1)*n+h \quad (1)$$

Esempio:

Indicizzazione normale

```
>> A(2,4)
```

```
ans =
```

```
0.6154
```

Indicizzazione lineare

```
>> A(11)
```

```
ans =
```

```
0.6154
```

Oltre che con la formula (1) l'indice lineare può essere calcolato a partire dagli indici riga-colonna (h,k) tramite la funzione sub2ind():

```
sub2ind(size(A),2,4)
```

```
ans =
```

```
11
```

Si possono ottenere gli indici riga-colonna a partire dall'indice lineare

```
>> [h k]=ind2sub(size(A),11)
```

```
h =
```

```
2
```

```
k =
```

```
4
```

Array multidimensionali

Anche nel caso di array multidimensionali matlab è possibile usare l'indicizzazione lineare. Consideriamo il seguente array

```
>> A=rand(4,5,3,2)
```

```
A(:,:,1,1) =
```

```
0.9218 0.9355 0.0579 0.1389 0.2722  
0.7382 0.9169 0.3529 0.2028 0.1988  
0.1763 0.4103 0.8132 0.1987 0.0153  
0.4057 0.8936 0.0099 0.6038 0.7468
```

```
A(:,:,2,1) =
```

```
0.4451 0.8462 0.8381 0.8318 0.3046  
0.9318 0.5252 0.0196 0.5028 0.1897  
0.4660 0.2026 0.6813 0.7095 0.1934  
0.4186 0.6721 0.3795 0.4289 0.6822
```

```
A(:,:,3,1) =
```

```
0.3028 0.3784 0.4966 0.8180 0.3412  
0.5417 0.8600 0.8998 0.6602 0.5341  
0.1509 0.8537 0.8216 0.3420 0.7271  
0.6979 0.5936 0.6449 0.2897 0.3093
```

```
A(:,:,1,2) =
```

```
0.8385 0.5466 0.7948 0.1730 0.8757
0.5681 0.4449 0.9568 0.9797 0.7373
0.3704 0.6946 0.5226 0.2714 0.1365
0.7027 0.6213 0.8801 0.2523 0.0118
```

A(:,:,2,2) =

```
0.8939 0.2844 0.5828 0.4329 0.5298
0.1991 0.4692 0.4235 0.2259 0.6405
0.2987 0.0648 0.5155 0.5798 0.2091
0.6614 0.9883 0.3340 0.7604 0.3798
```

A(:,:,3,2) =

```
0.7833 0.7942 0.4154 0.7680 0.4387
0.6808 0.0592 0.3050 0.9708 0.4983
0.4611 0.6029 0.8744 0.9901 0.2140
0.5678 0.0503 0.0150 0.7889 0.6435
```

Ogni matrice A(:,:,h,k) viene indicata come un a “pagina” (questa è la dizione usata nella documentazione). La memorizzazione interna viene ottenuta concatenando le colonne delle matrici di ogni pagina, seguendo l’ordine delle pagine.

Anche in questo caso è possibile utilizzare la funzione sub2ind(). Come esempio si calcola l’indice lineare dell’elemento (3,4,2,2)

```
>> sub2ind(size(A),3,4,2,2)
```

```
ans =
```

```
95
```

```
>> A(95)
```

```
ans =
```

```
0.5798
```

Reshape di array multidimensionali

Tramite l’operazione di reshape è possibile inserire gli elementi di un array all’interno di un array con dimensioni differenti: l’unica condizione affinché questo sia possibile è che il numero degli elementi dei due array sia lo stesso.

Questa funzione opera lungo le colonne degli array.

Esempio.

```
>> A=rand(3,4)
```

```
A =
```

```
0.6992  0.5548  0.7159  0.2548
0.7275  0.1210  0.8928  0.8656
0.4784  0.4508  0.2731  0.2324
```

```
>> B=reshape(A,3,2,2)
```

```
B(:,:,1) =
```

```
0.6992  0.5548
0.7275  0.1210
0.4784  0.4508
```

```
B(:,:,2) =
```

```
0.7159  0.2548
0.8928  0.8656
0.2731  0.2324
```

Esempio di applicazione alla matrice di dati fMRI.

Consideriamo dati da un esame di risonanza magnetica funzionale composti da una sequenza di 100 immagini di 64x64 pixel. Questo tipo di dato può essere memorizzato in un array `im_serie` di dimensione 64x64x100.

L'immagine *k*-esima può essere estratta con il comando

```
im_k=im_serie(:,:,100);
```

La serie temporale che descrive l'andamento nel tempo del pixel (x,y) dell'immagine si può ottenere come

```
serie_t_x_y=im_serie(x,y,100);
```

Un altro modo di memorizzare questo tipo di dato, modo che risulta utile in alcune applicazioni, è quello che rappresenta ogni immagine della serie come una riga.

Un modo potrebbe essere quello di rappresentare i dati come una matrice 100x4096. Una matrice con i dati in questa forma è stata fornita nel materiale di supporto alla didattica anno 2011/2012 nel file *fetta_20_007_V6.mat*.

E' possibile passare dalla matrice 100x4096 ad una 64x64x100 in questo modo

```
im_serie=reshape(matrix',64,64,100);
```

Si fa notare come sia stata necessaria la trasposizione di `matrix`, per permettere che ogni riga di `matrix` (colonna di `matrix'`) formi un'immagine della serie. In caso contrario i dati delle immagini risulterebbero mescolati.

Le serie temporali di ogni pixel sono memorizzate nelle colonne di `matrix`.

L'indice lineare che individua la colonna della matrice `matrix`, contenente la serie temporale relativa al pixel (x,y) di ogni immagine, si può trovare utilizzando la conversione da indice riga colonna a indice lineare. L'indice di colonna *k* si ottiene come `k=sub2ind([64 64],x,y)`; per cui

```
serie_t_lin=matrix(:,k);
```