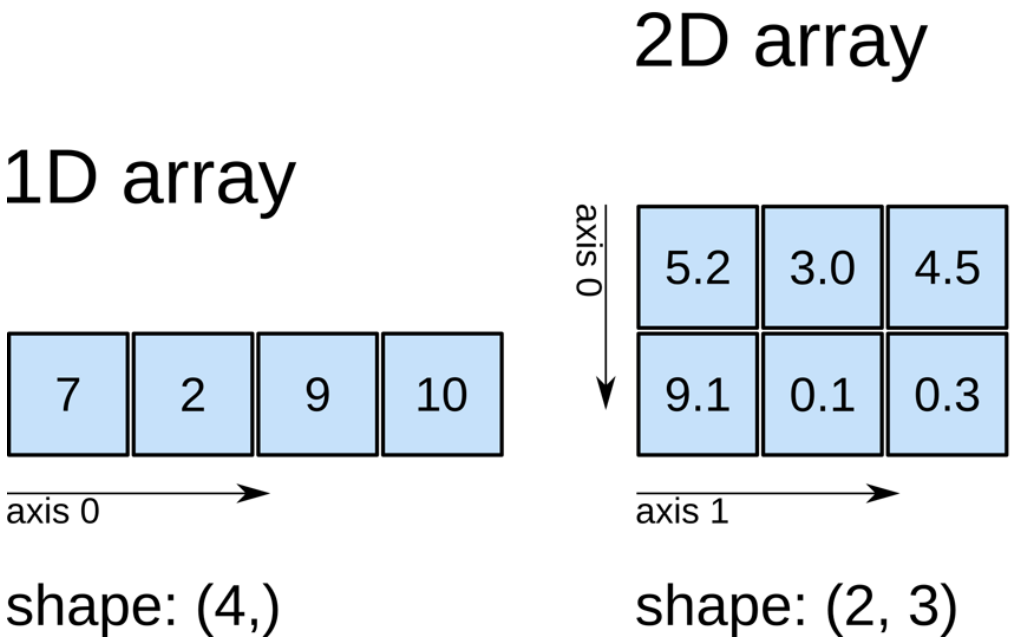
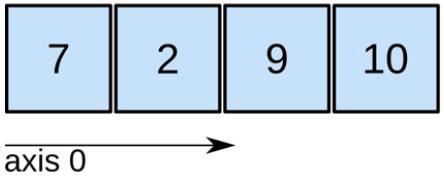
Numpy

Visual Representation

Dimensions

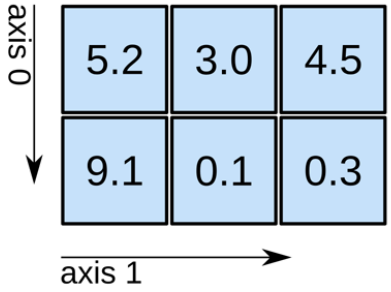

1D array
7
2
9
10
axis 0
shape: (4,)
2D array
axis 0
5.2
3.0
4.5
9.1
0.1
0.3
axis 1
shape: (2, 3)

1D array



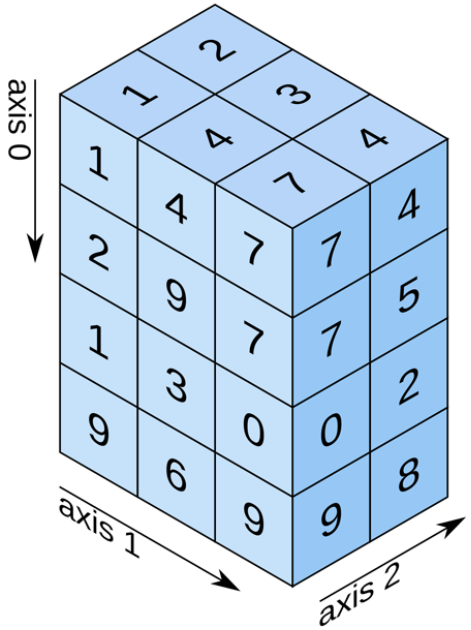
shape: (4,)

2D array



shape: (2, 3)

3D array



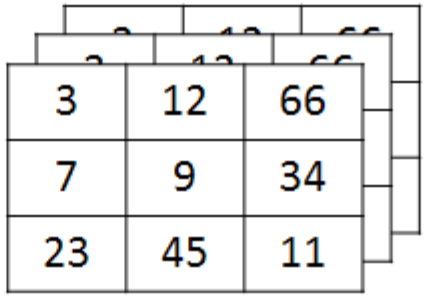
shape: (4, 3, 2)

1	5	18	23

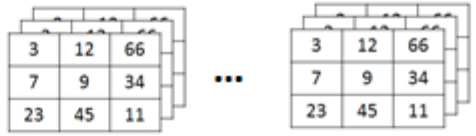
Vector (1D array)
Dimension = 1
(1 index required)

3	12	66
7	9	34
23	45	11

Matrix (2D array)
Dimension = 2
(2 indexes required)

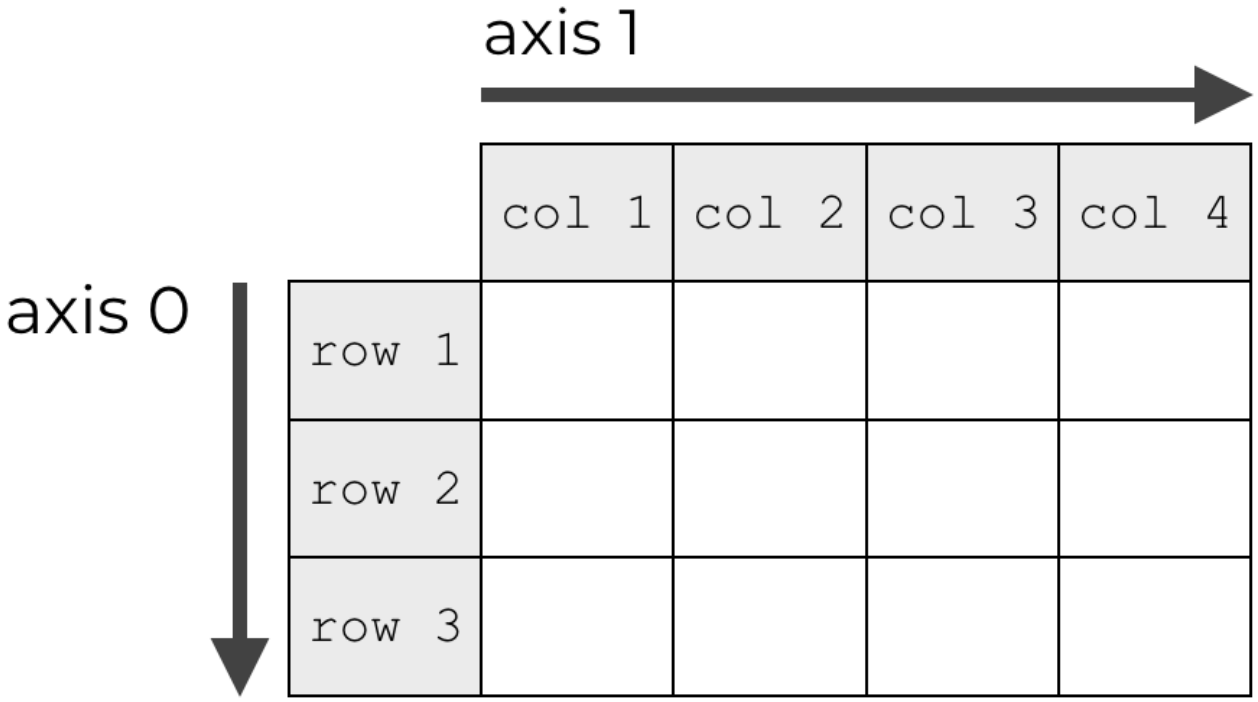


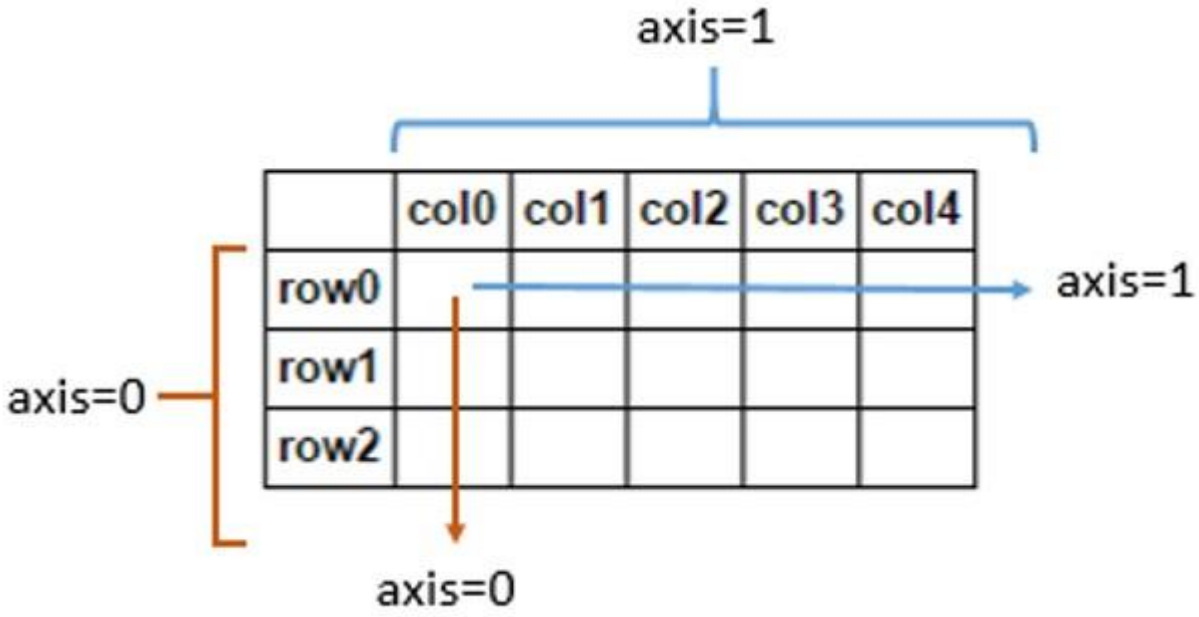
3D array (3rd order Tensor)
Dimension = 3
(3 indexes required)

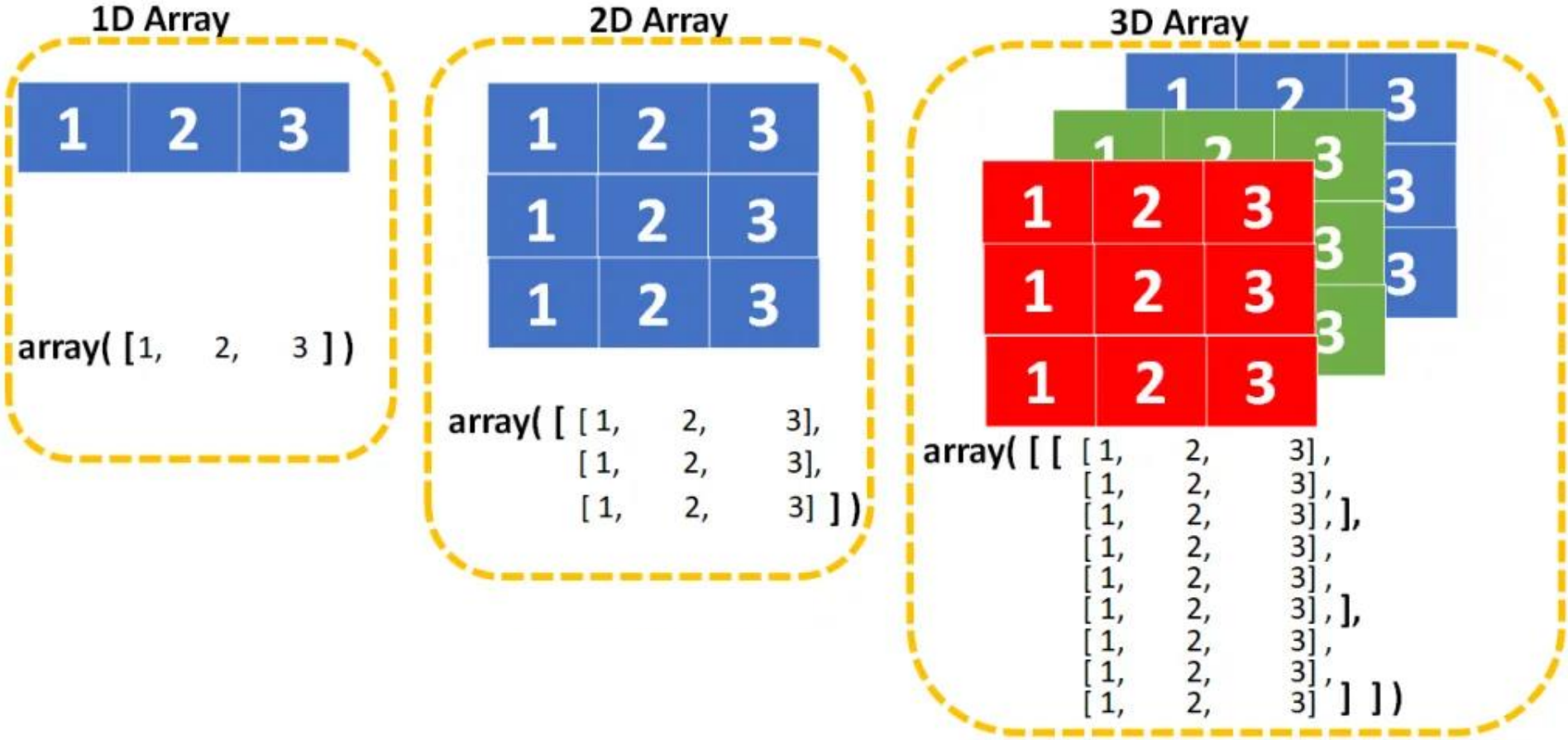


ND array
Dimension = N
(N indexes required)

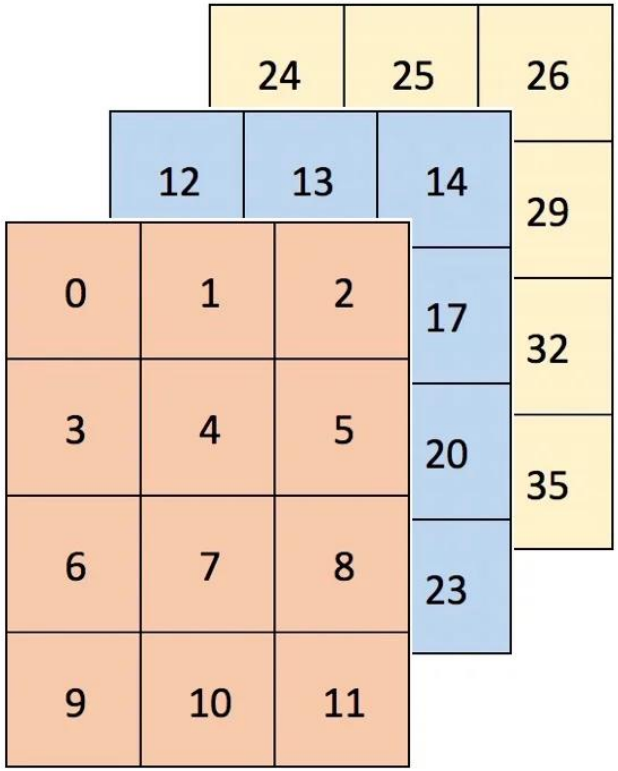
axis




axis=1
col0
col1
col2
col3
col4
row0
row1
row2
axis=1
axis=0
axis=0


1D Array
1 2 3
array([1, 2, 3])
2D Array
1 2 3
1 2 3
1 2 3
array([[1, 2, 3],
[1, 2, 3],
[1, 2, 3]])
3D Array
1 2 3
1 2 3
1 2 3
array([[[1, 2, 3] ,
[1, 2, 3] ,
[1, 2, 3] ,],
[1, 2, 3] ,
[1, 2, 3] ,
[1, 2, 3] ,],
[1, 2, 3] ,
[1, 2, 3] ,
[1, 2, 3]]])

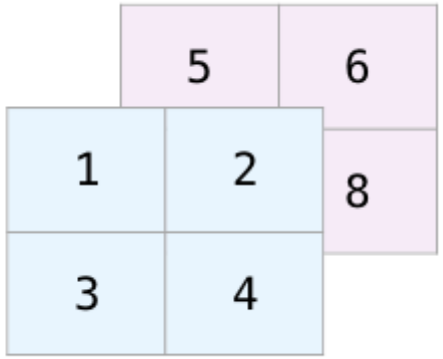
3-Dimensions



arr.shape
(3, 4, 3)
A container of three 4x3 grids (or a rectangular prism)

3-Dimensions

```
np.array([ [[1,2],[3,4]],
           [[5,6],[7,8]] ])
```



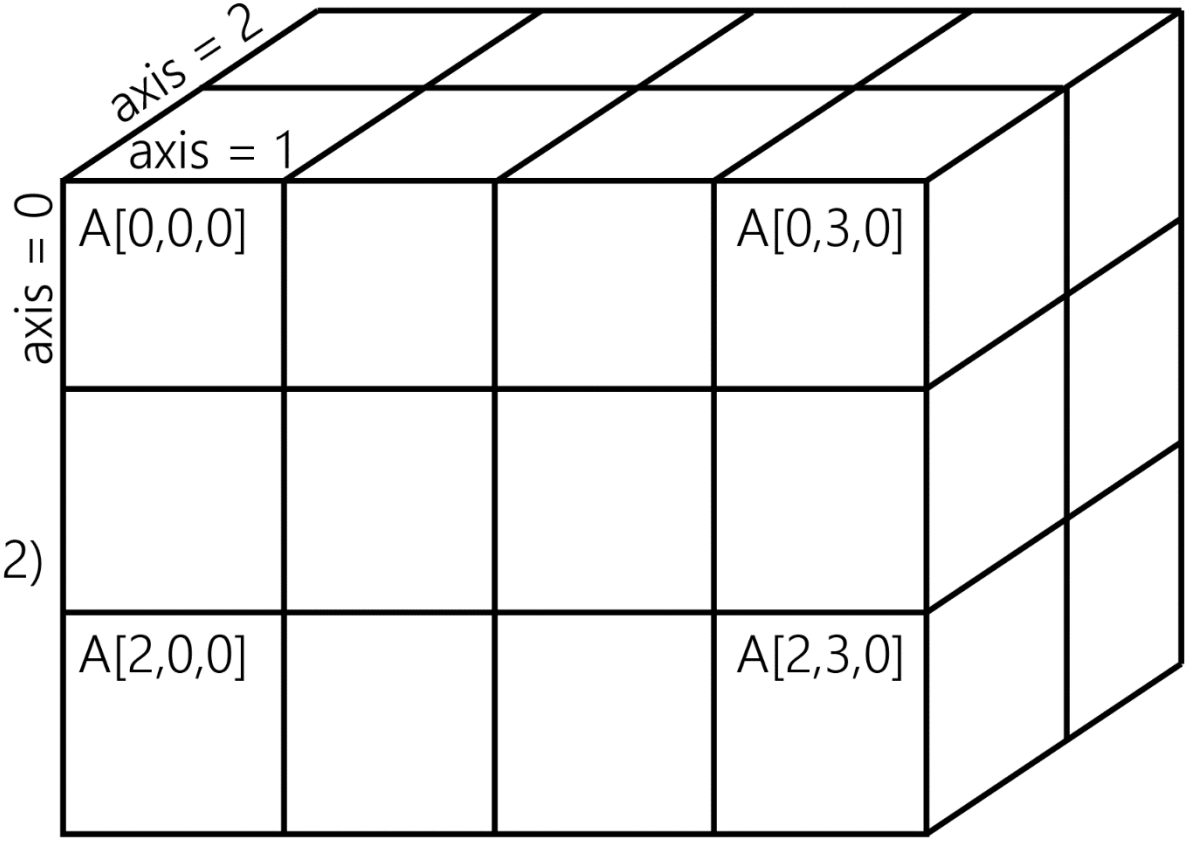

```
[[[1 2]
  [3 4]]

 [[5 6]
  [7 8]]]
```

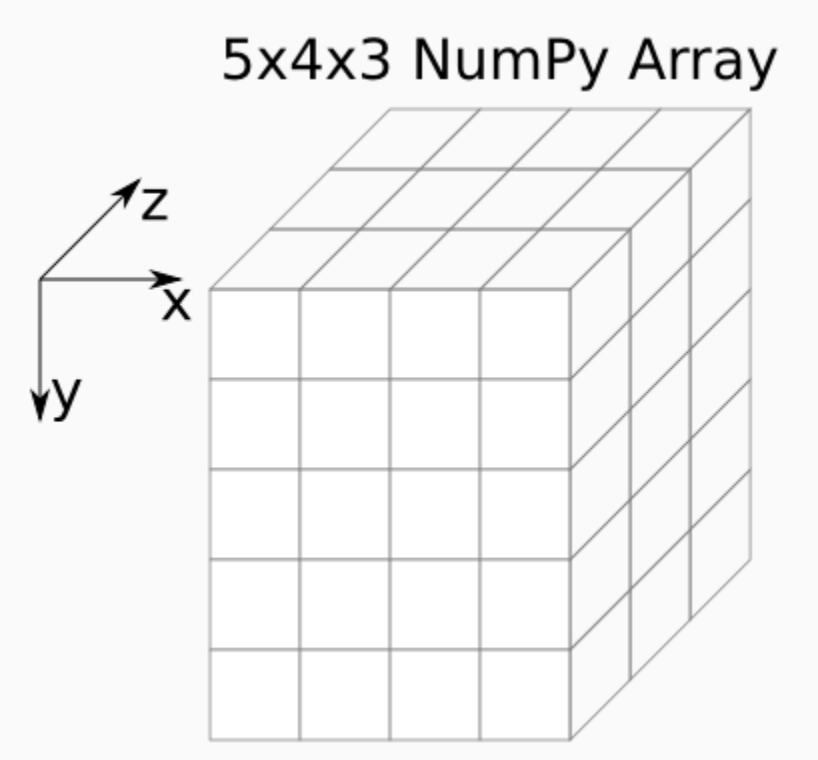
3
(2, 2, 2)
(Depth/layers,Row,Col)

Numpy -3d

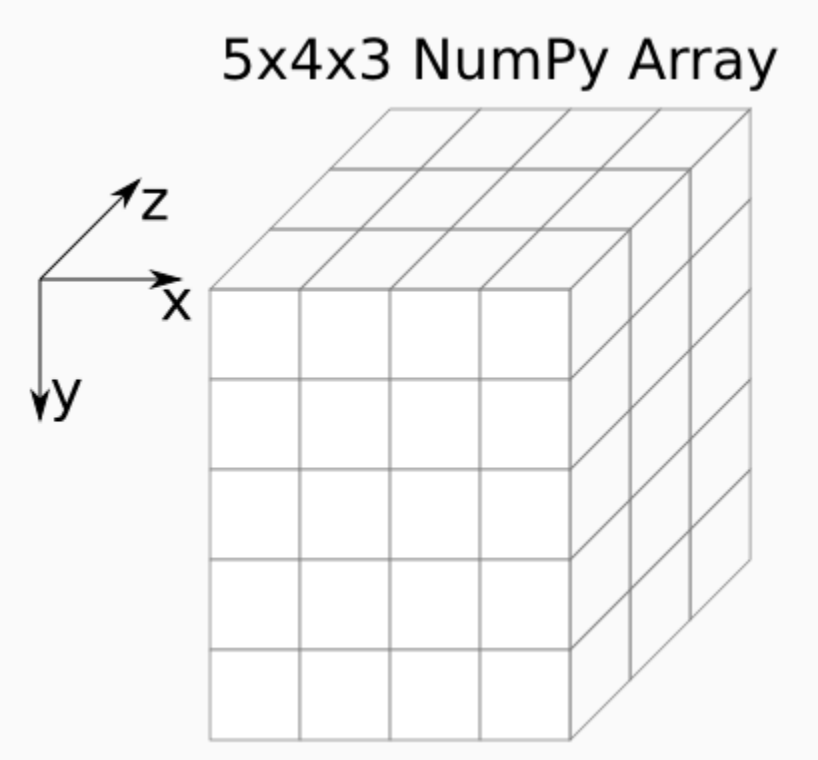
ndarray
ndim = 3
shape = (3, 4, 2)



Numpy 3d



Numpy 3d



Loading Data

1

Load 2-d wine data

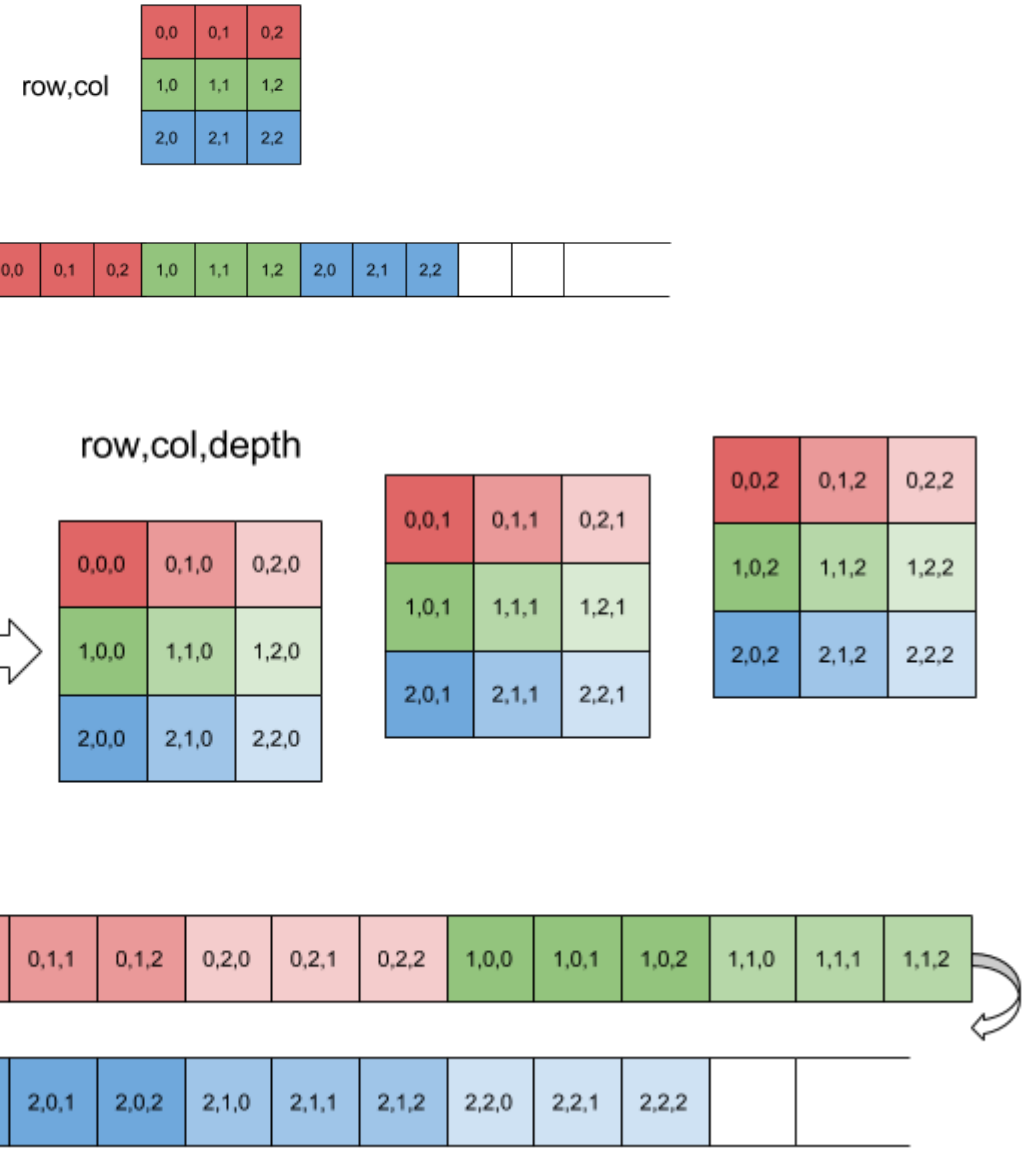
Loading Data

2

- PythonNumpyHands-On-Training
  - Load the image data, 3D data

3

- classifying-movie-reviews
  - Sentiment analysis
  - Load the image data, 3D data


row,col
0,0 0,1 0,2
1,0 1,1 1,2
2,0 2,1 2,2
0,0 0,1 0,2 1,0 1,1 1,2 2,0 2,1 2,2
row,col,depth
0,0,0 0,1,0 0,2,0
1,0,0 1,1,0 1,2,0
2,0,0 2,1,0 2,2,0
0,0,1 0,1,1 0,2,1
1,0,1 1,1,1 1,2,1
2,0,1 2,1,1 2,2,1
0,0,2 0,1,2 0,2,2
1,0,2 1,1,2 1,2,2
2,0,2 2,1,2 2,2,2
0,0,0 0,0,1 0,0,2 0,1,0 0,1,1 0,1,2 0,2,0 0,2,1 0,2,2 1,0,0 1,0,1 1,0,2 1,1,0 1,1,1 1,1,2
1,2,0 1,2,1 1,2,2 2,0,0 2,0,1 2,0,2 2,1,0 2,1,1 2,1,2 2,2,0 2,2,1 2,2,2

End