In machine learning, the data often comes in multi-dimensional arrays so matrices are best to handle such inputs. So we need to have some general idea on basic arithmetic operations in matrices. In this article, we will discuss about matrices and matrix arithmetic for machine learning.

Matrix is a two dimensional array where the data is in the form of rows and columns. The each element in the matrix is represented as a[i][j] where i represents the row number and j represents the column number. Each matrix has an order representing the total number of rows (m) and columns (n) in the matrix and it is written as m X n.

Defining a matrix:
In Python, we use numpy module to define a matrix. As we discussed that matrices are two dimensional arrays, let us create a two dimensional array (matrix) using numpy.array( ) method.

```
#importing numpy module
import numpy as np

#defining a matrix using np.array() method
matrix = np.array([[1,2,3],[4,5,6],[7,8,9]])

#printing the matrix
print(matrix)
```

Output:

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

we can see that it is in the form of list of lists which is a two-dimensional array having 3 rows and 3 columns. The order of the above matrix is 3 X 3.

Matrix Arithmetic
Matrix Addition
Addition of matrices can only be done when the two matrices have the same order. It is done by using (+) operator which adds corresponding elements of the matrices. For example, if we have two matrices A and B then the a[i][j] is added with b[i][j] where i is row number and j is column number.

```
import numpy as np
A = np.array([[1,2],[3,4]])
B = np.array([[5,6],[7,8]])

#adding A and B matrices
C = A + B
print(C)
```

Output:

```
[[ 6  8]
 [10 12]]
```

Explanation : In the above example, we have two matrices A and B.

A = [1 ,2] B = [5,6]

[3, 4] [7, 8]

C = [1 + 5 2+6] C = [6 8]

[3+ 7 4+8] ==> [10 12]

We can see that the corresponding elements are added and we get the final matrix with the same order.

Matrix Subtraction

Subtraction of matrices is possible when both the matrices have same order. Subtraction is done by using (-) operator in Python. When we perform subtraction on two matrices then the elements from the first matrix are subtracted with their corresponding elements in second matrix.

```
import numpy as np
A = np.array([[1 , 2],[3 , 4]])
B = np.array([[0 , 1],[2 , 1]])
C = A - B
print(C)
```

Output:

```
[[1 1]
 [1 3]]
```

Explanation: A = [1 ,2] B = [0 , 1]

[3, 4] [2 , 1]

C = [1 - 0 2 - 1] C = [1 1]

[3 -2 4 - 1] ==> [1 3]

The elements of the matrix A are subtracted with elements of matrix B and we got the resultant matrix with order 2 X 2.

Matrix Division

Division of matrices is done by using (/) operator. It divides elements from first matrix with corresponding elements in the second matrix. The matrices should have same dimensions.

```
import numpy as np
A = np.array([[4 , 2],[6 , 8]])
B = np.array([[2 , 2],[3 , 4]])
C = A//B
print(C)
```

Output:

```
[[2 1]
 [2 2]]
```

Explanation : A = [4 ,2] B = [2 , 2]

[6, 8] [3 , 4]

C = [4/2 2 /2] C = [2 1]

[6/3 8/4] ==> [2 2]

Matrix - Matrix Multiplication (Dot product)

Multiplication of matrices can only be done when the columns of first matrix is equal to the rows of the second matrix. Suppose there is a matrix A of order m X n and matrix B of order n X k so here columns of matrix A = rows of matrix B = n and the order of the resulting matrix C = AB is m X k. Hence multiplication is possible for the above matrices.

Note : Matrices do not satisfy commutative property

AB is not equal to BA

```
import numpy as np
A = np.array([[1, 2],[3, 4]])
B = np.array([[5, 6],[7, 8]])
C = A.dot(B)
print(C)
```

Output:

```
[[19 22]
 [43 50]]
```

Explanation : [1 , 2] [5, 6] [1*5 + 2*7 1*6 + 2*8] [19 22]

[3, 4] X [7, 8] ==> [3*5 + 4*7 3*6 + 4*8] ==> [43 50]

The matrix A is of order 2 X 2 and matrix B of order 2 X 2. Since the columns of A = rows of B = 2, the multiplication is possible. The multiplication is done in a way where the 1st row of A is multiplied with 1st and 2nd column of B giving us C[0][0] and C[0][1] and the 2nd row of A is multiplied with 1st and 2nd column of B giving us C[1][0] and C[1] [1].

Matrix - Vector multiplication

A vector is a one dimensional array having either a single row or single column. A matrix having only one row is called as a row vector and matrix having only one column is called as a column vector. While multiplying the matrix with a vector, we need to check the multiplication condition which is the columns of first matrix = rows of the second matrix.

```
import numpy as np
A = np.array([[1, 2],[1 , 1]])
V = np.array([[1] , [1]])
```

```
#matrix - vector multiplication
C = A.dot(V)
print(C)
```

Output:

```
[[3]
 [2]]
```

Explanation : We multplied the matrix A with vector V and we got a column vector C. In the same way we can multiply a matrix with a row vector and the dot product will also be a row vector.

A = [1 , 2] V = [1] [1 * 1 + 2 * 1] ==> A.V = [3]

[1 , 1] [1] ==> A . V = [1 * 1 + 1 * 1] ==> A . V = [2]

Matrix - Scalar multiplication

When we multiply a matrix with a scalar then it is multiplied with each and every element in the matrix. The order remains same even after multiplying the matrix with scalar. We use matrix - scalar multiplications while solving equations in linear algebra.

```
import numpy as np
A = np.array([[1,2],[3,4]])
b = 2
print(A * b)
```

Output:

```
[[2 4]
 [6 8]]
```

Explanation : In the above matrix, we can see that b=2 is multiplied with each element in the matrix.

A*b = [1*2 , 2*2] ==> [2 , 4]

[3*2 , 4*2] ==> [6 , 8]