Chap 3 - NumPy Matrix Multiplication

March 24, 2018

0.1 NumPy Matrix Multiplication

0.1.1 Element-wise Multiplication

This can be accomplished by the **multiply** function or the * operator.

```
In [1]: import numpy as np

        m = np.array([[1,2,3],[4,5,6]]) # 2x3 matrix
        print('m:\n', m)

        n = m * 0.25 # multiply with a scalar
        print('n\n', n)

        # multiply 2 matrices in the element-wise manner
        # method#1
        mn1 = m * n
        print('m * n:\n', mn1)
        # method#2
        mn2 = np.multiply(m, n)
        print('np.multiply(m,n):\n', mn2)
```

```
m:
 [[1 2 3]
 [4 5 6]]
n
 [[0.25 0.5  0.75]
 [1.   1.25 1.5 ]]
m * n:
 [[0.25 1.   2.25]
 [4.   6.25 9.  ]]
np.multiply(m,n):
 [[0.25 1.   2.25]
 [4.   6.25 9.  ]]
```

0.1.2 Matrix Product

This can be accomplished by NumPy's **matmul** function. However, the function expects the matrices have compatible shapes.

```
In [2]: a = np.array([[1,2,3],[1,2,3]]) # 2x3
        print('a:\n', a)
        print('a.shape:', a.shape)

        b = np.array([[1],[10],[100]]) # 3x1
        print('b:\n', b)
        print('b.shape:', b.shape)

        # Compute the matrix product
        c = np.matmul(a,b)
        print('c:\n', c)
        print('c.shape:', c.shape)
```

```
a:
 [[1 2 3]
 [1 2 3]]
a.shape: (2, 3)
b:
 [[  1]
 [ 10]
 [100]]
b.shape: (3, 1)
c:
 [[321]
 [321]]
c.shape: (2, 1)
```

0.1.3 NumPy's `dot` function

If the matrices are two dimensions, the results of `dot` and `matmul` are the same.

```
In [3]: a = np.array([[1,0,0],[0,10,0],[0,0,100]]) # 3x3
        print('a:\n', a)

        # dot product
        b = np.dot(a,a)
        print('np.dot(a,a):\n', b)

        # matmul
        c = np.matmul(a,a)
        print('np.matmul(a,a):\n', c)
```

```
a:
 [[  1   0   0]
```

```
 [  0  10   0]
 [  0   0 100]]
np.dot(a,a):
 [[    1     0     0]
 [    0   100     0]
 [    0     0 10000]]
np.matmul(a,a):
 [[    1     0     0]
 [    0   100     0]
 [    0     0 10000]]
```