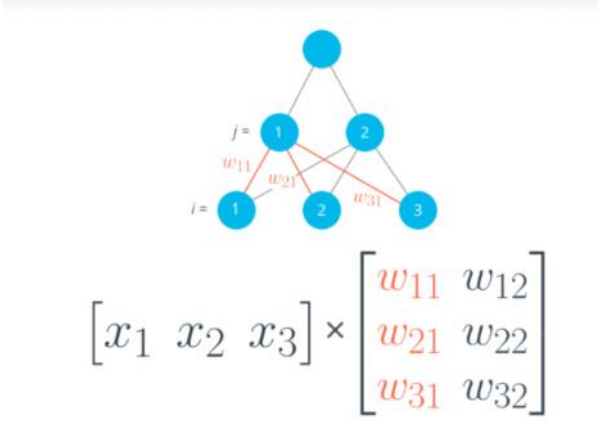
Nanodegree - Lesson 5

Sunday 17 January 2021 19:16



Screen clipping taken: 17/01/2021 19:17

Screen clipping taken: 17/01/2021 19:20

Data Types and Shapes

The most common way to work with numbers in NumPy is through `ndarray` objects. They are similar to Python lists, but can have any number of dimensions. Also, `ndarray` supports fast math operations, which is just what we want.

Since it can store any number of dimensions, you can use `ndarray`s to represent any of the data types we covered before: scalars, vectors, matrices, or tensors.

Screen clipping taken: 17/01/2021 19:23

Scalars

Scalars in NumPy are a bit more involved than in Python. Instead of Python's basic types like `int`, `float`, etc., NumPy lets you specify signed and unsigned types, as well as different sizes. So instead of Python's `int`, you have access to types like `uint8`, `int8`, `uint16`, `int16`, and so on.

Screen clipping taken: 17/01/2021 19:24

If you want to create a NumPy array that holds a scalar, you do so by passing the value to NumPy's `array` function, like so:

```
s = np.array(5)
```

Screen clipping taken: 17/01/2021 19:24

Vectors

To create a vector, you'd pass a Python list to the `array` function, like this:

```
v = np.array([1,2,3])
```

If you check a vector's `shape` attribute, it will return a single number representing the vector's one-dimensional length. In the above example, `v.shape` would return `(3,)`

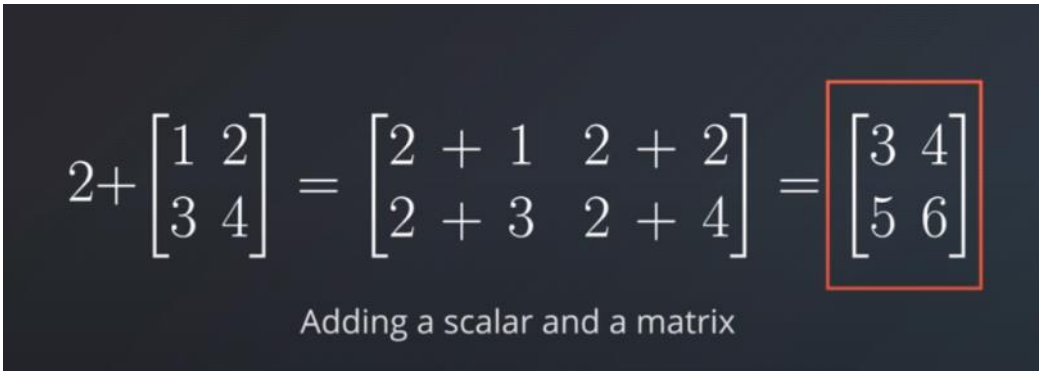
Screen clipping taken: 17/01/2021 19:25

You can access an element within the vector using indices, like this:

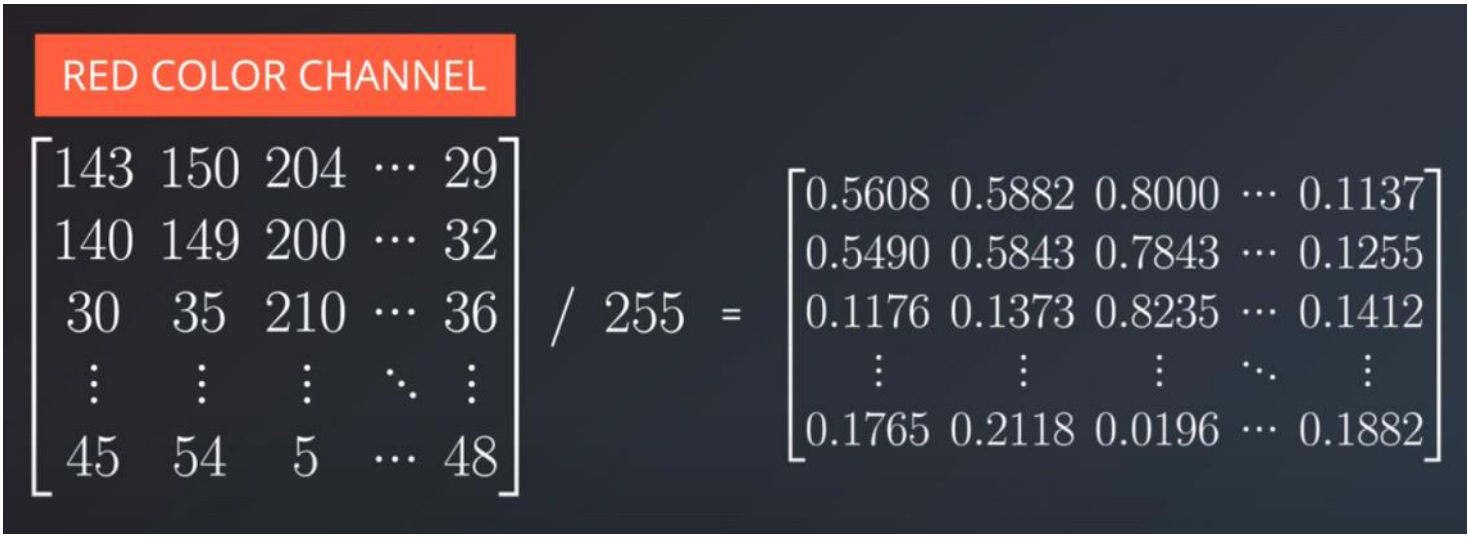
```
x = v[1]
```

Now `x` equals `2`.

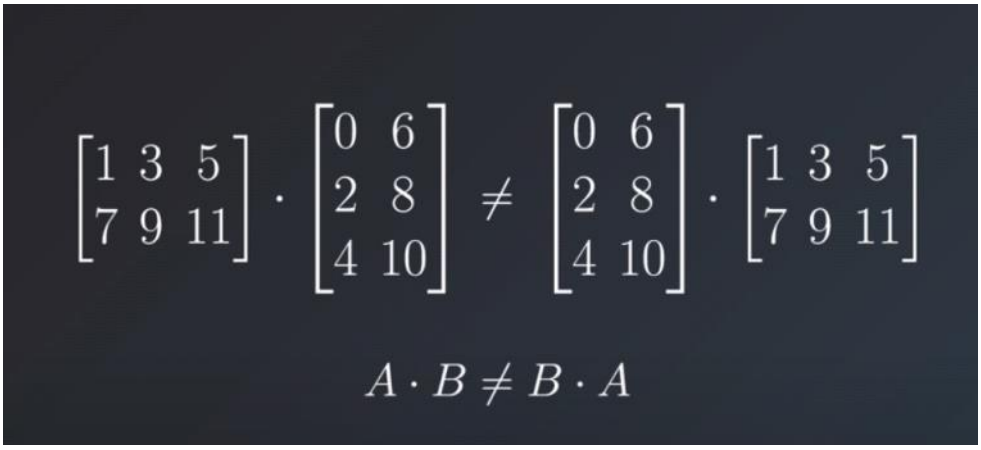
Screen clipping taken: 17/01/2021 19:25



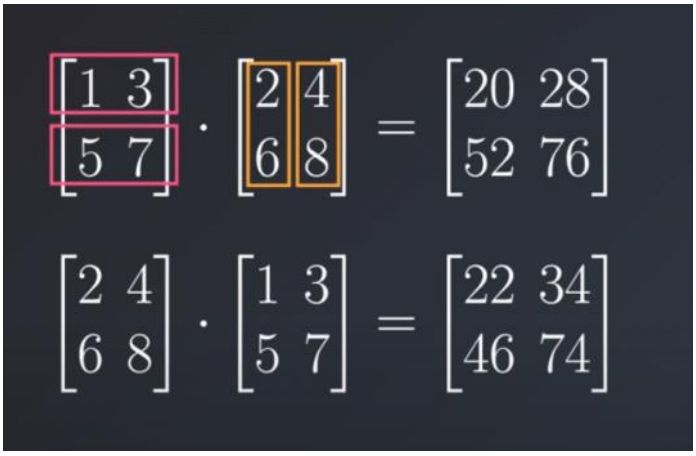
Screen clipping taken: 17/01/2021 20:11



Screen clipping taken: 17/01/2021 20:12



Screen clipping taken: 18/01/2021 20:30

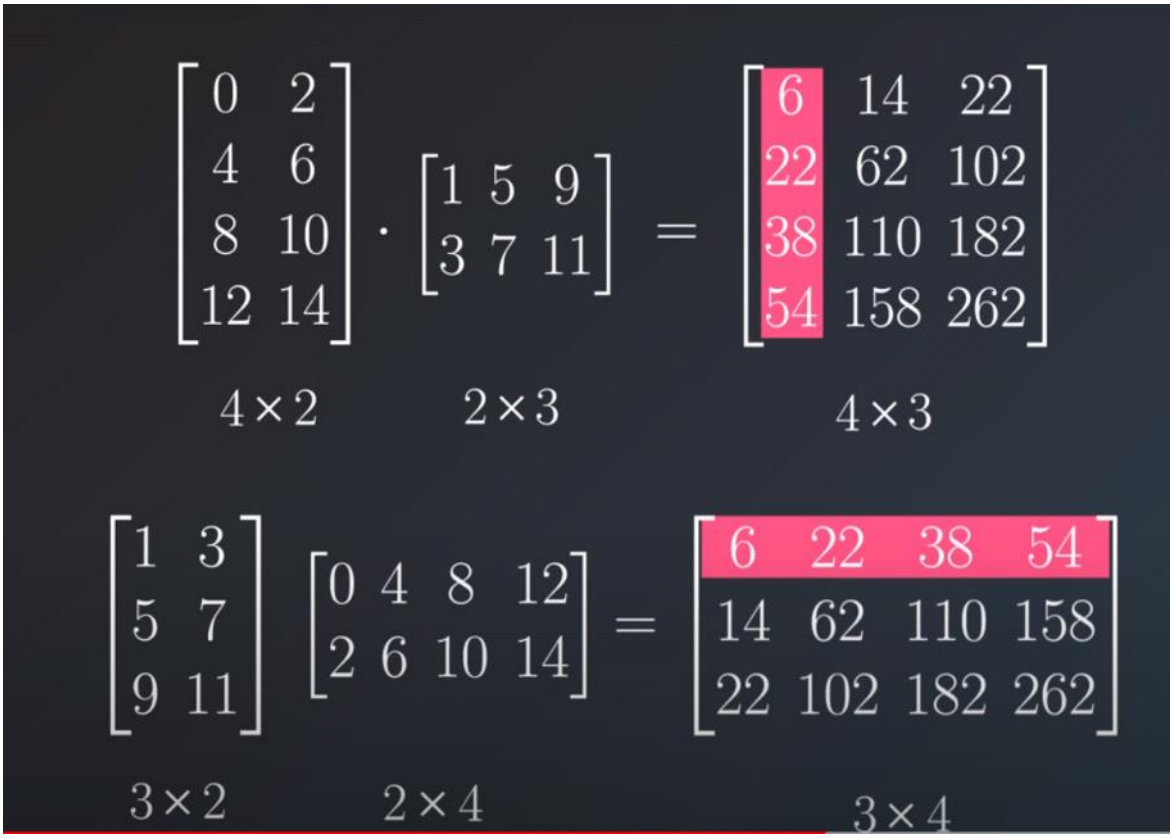


The determinant of those two matrix is the same det(A * B) = det (B * A) if A*B is square matrix

Mat multiplication

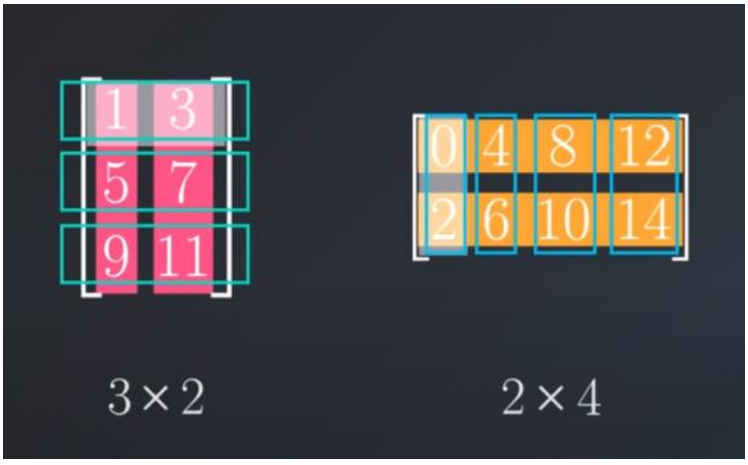
C = A x B => Cij = SUM(Ai * Bj)

I j k m
A - 3 x 4 x 5 x 6
B - 5 x 4 x 6 x 3

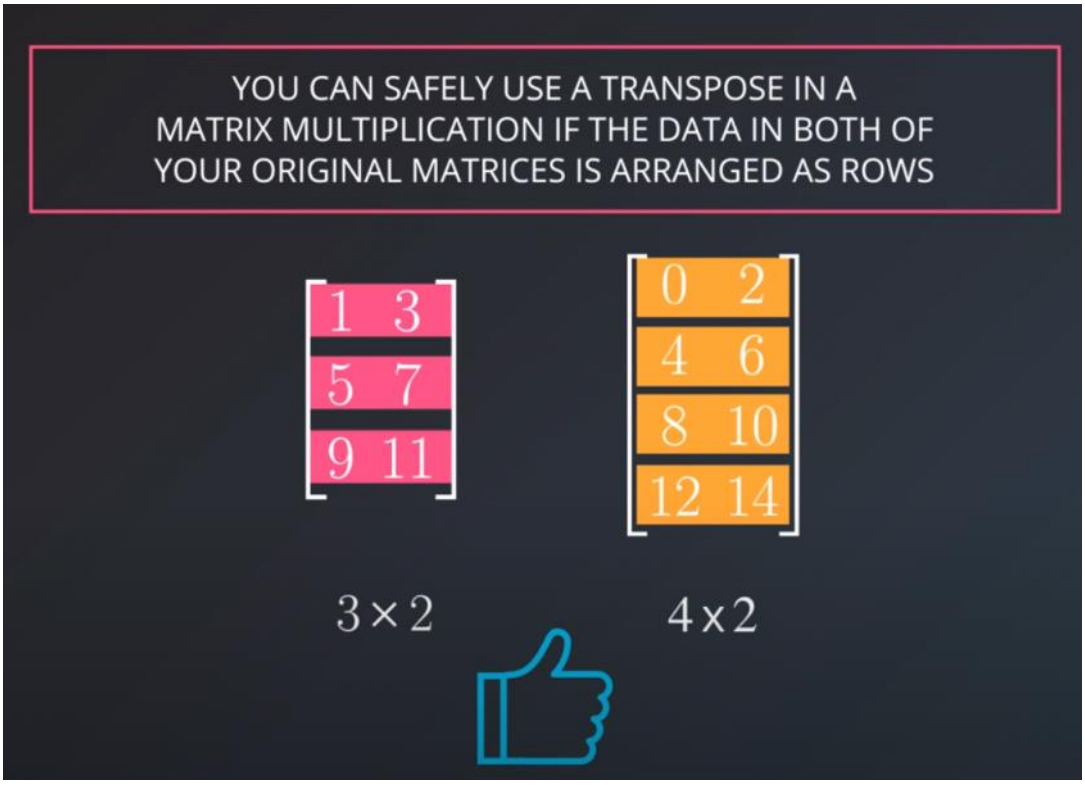


Screen clipping taken: 18/01/2021 21:02

Something that works mathematically, it does not necessary makes sense semantically. Let's say the two matrix below store information in the columns (each column is one item of data - ie. A person, a book, etc). If you do the matrix multiplication you mix up the data, you take data from two different data item and you do not want this. You want to deal with data coming from each entity.



Screen clipping taken: 18/01/2021 21:03



Screen clipping taken: 18/01/2021 21:08