PSYC40012 Models of Psychological Processes

Tutorial Video 3

In this video, I will demonstrate some common functions and operations applied to matrices.

First, let's create a few variables:

```
A = [1 2; 3 4; 5 6]

B = [11 12; 13 14; 15 16]

C = [1 1; 2 2]
```

Matrix multiplication

Elementwise multiplication

There are a few ways in which we can multiply matrices together. One way is to multiply each element of the matrix together. To do *elementwise* multiplication, we need to use the .* operator

```
A .* B %  multiplies each element
```

Inner product

The other method for multiplying matrices takes advantage of special linear algebra operation known as the *inner product*

The inner product computes the sum of the products of each of the elements in a vector. For instance, if you have two vectors such as the following:

```
a = [2, -1, 0]

b = [1, 2, -1]
```

The inner product is computed as follows:

```
a(1) * b(1) + a(2) * b(2) + a(3) * b(3)
```

This is computed easily in Matlab by using the * operator and multiplying a row vector times a column vector with the same number of elements.

So if we type the following:

```
a * b'
```

Matlab returns the inner product. The key thing to remember is that the number of entries in the row vector has to match the number of entries in the column vector.

For matrices, we can compute the inner product as follows:

```
X = [2, -1, 0; -1 2 -1; 0 -3 4];
```

```
y = [1; 2; -1];

X * y
```

This returns the inner product between the matrix X and the column vector y. The key is that the number of columns in X has to match the number of rows in y. What is returned is the inner product between each row of X and y.

We can also find the inner product between two matrices (so long as the number of columns in the first matrix is equal to the number of rows in the second matrix).

```
A * C
```

The inner product is very useful when computing the activation of units in a neural network model. We'll discuss this further in class.

Outer product

Notice that in computing the inner product, we multiply a row vector (or a matrix comprised of multiple rows) times a column vector (or a matrix comprised of multiple columns). We can also multiply a column vector times a row vector, but this returns a different result. This operation finds the *outer product.*

```
a = [2, -1, 0]

b = [1, 2, -1]

a' * b
```

The outer product returns a matrix in which the row entries are the product of the first element in the leading column vector (i.e., a') times each of the elements in the row vector (i.e., b). I won't go into further detail about the outer product, but it is useful to know that there are several ways to multiply matrices and each gives different results.

Applying functions to matrices

There are number of different ways we might want to transform matrices or evaluate matrices

```
A = [1 2; 3 4; 5 6]

1./A  % Find the inverse of a matrix

log(A) % Log function

exp(A) % Exponential function

abs([-1; 2;-4]) % Absolute value

-A % Negative of a matrix
```

```
V = [1 2 3]'
V + ones(length(v), 1)% Add one to all entries of the vector v
V + 1 % Add one to all entries of the vector v
```

```
a = [1 15 2 .5]
max(a)
[a, idx] = max(a)
max(a) % maximum across the rows (i.e., in each column)

a < 3 % returns true for any element in which a is less than 3
find (a < 3) % returns the index for any true element
a(a < 3) % returns only those true elements

A = magic(3) % magic squares
[r, c] = find(A >= 7)
help find
```

```
sum(a)
prod(a)
floor(a)
ceil(a)
round(a)
```

```
rand(3)
max(A, [], 1) – max of each column
max(A, [], 2) – max of each row
max(max(A)) – overall max
max(A(:)) – overall max
```