**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **16-07-2020** | **Name:** | **Kavya M M** |
| **Course:** | **Coursera** | **USN:** | **4AL17EC040** |
| **Topic:** | **Mathematics for machine learning: Linear algebra** | **Semester & Section:** | **6th A** |
| **Github Repository:** | **Kavya\_ECE040** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |
| **Einstein summation convention and the symmetry of the dot product:**  Now, there’s an important other way to write matrix transformations down. It's  Called the Einstein's Summation Convention. And that writes down what the actual Operations are on the elements of a matrix, which is useful when you're coding or programming. It also lets us see something neat about the dot product that want to show you. And it lets us deal with non-square matrices. When we started, we said that multiplying a matrix by a vector or with another matrix is a process of taking every element in each row in turn, multiplied with corresponding element in each column in the other matrix, and adding the mall up and putting them in place. So, let's write that down just to make that concrete.  So, I'm going to write down a matrix A here, and I'm going to give it elements.  A's Ann by n matrix. I'm going to give it elements a11, a21, all the way down to an1.And then a12, all the way across to a1n. And then I'll have a22 here all the way across, all the Way down until fill it all in and I've gotten down here. So, the first suffix on this matrix, first Suffix on all of these elements in the matrix is the row number, and these condone is the Column number. Now, if I want to multiply A by another matrix B, and that's also going to be An n by n matrix, and that will have elements b11, b12 across to b1n, and down to bn1 and Across to bnn, dot.  If I multiply these together, I'm going to get another matrix, which I'll call AB, and  Then what I'm going to do is I'm going to take arow of A multiplied by the elements of a Column of B and put those in the corresponding place. So let's do an example. So if I want an element, let's say ab, element two, three. I'm going to get that by taking grow two of A, multiply by column three of B. So I'm going to take row two of A, that's going to be a21, a22, and all the other sup to a2n, and I'm going to multiply it by column three of B. So that's b13, b23, all the way to bn3. And I'm going to add all those up. And I'll have a dot, dot, dot in between. So that's going to be this element, row two, column three of AB. Now, in Einstein's convention, what you do, is you say, well okay, this is the sum over some elements jof aij, bike. So, if I add these up over all the possible j's, I'm going to get a11, b11 plus a12, b21, and  soon, and soon, and that's for I and k as well.  I'm going to then go around all the possible i's and k's. So, what Einstein then  says, well okay, if I've got are peated index, I won't bother with the sum and I'll just write that down as being aij, bjk. And that's equal to this the product ab ik. So abik is equal to ai1,b1k, plus ai2, b2k, plus ai3, b3k and soon and soon, until you've done all the possible j's, and then you do that for all the possible i's and k's, and that will give you your whole matrix for AB, for the product. Now, this is quite nice. If you are coding, you just run three loops over i, j And k, and the nuse an accumulator on the j's here to find the elements of the product matrix AB. So the summation convention gives you a quick way of coding up these sorts of operations. Now, we haven't talked about this so far but now we can see it. There's no reason, so long as them articles have the same number of entries in j, then we can multiply the m together even if they're not the same shape. So, we can multiply a two by three matrix, something with two rows and three columns. So, one, two, three, one, two, three, by a three by four matrices, three there and four there. So, it's got one, two, three, four times. And when I multiply these together, I'm going to go that row times that column. |