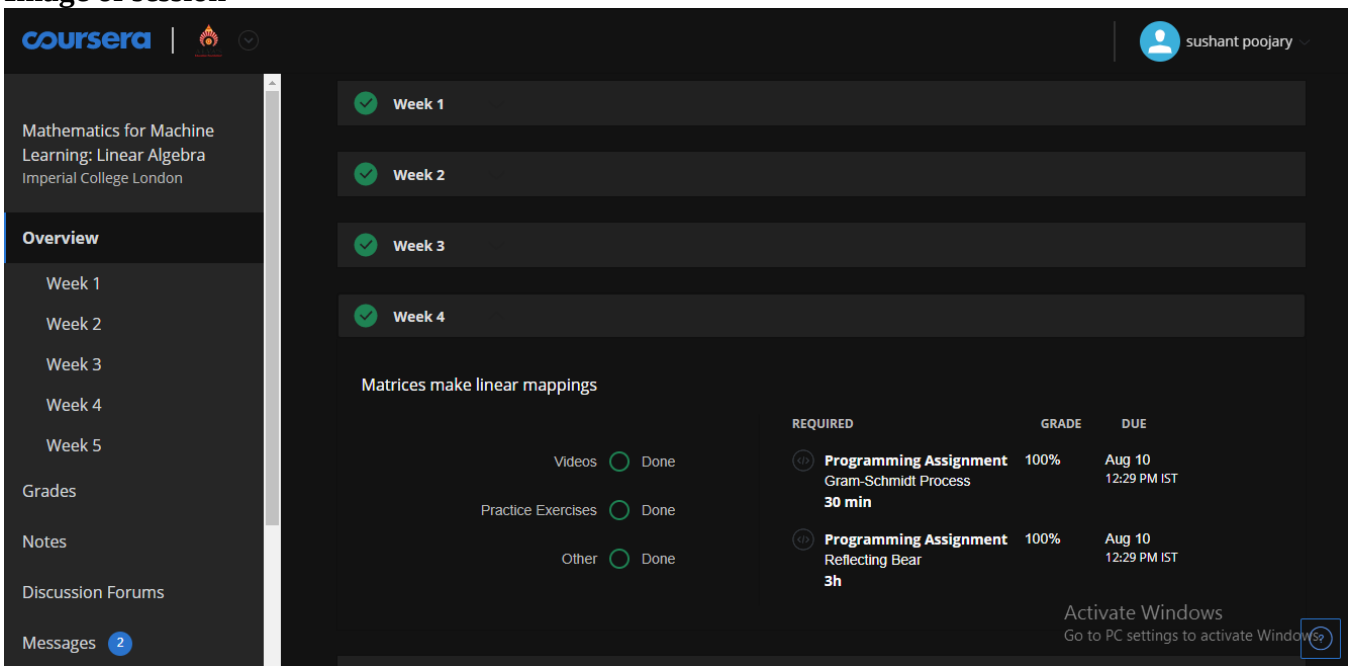
DAILY ASSESSMENT FORMAT

Date:	**16th July 2020**	**Name:**	**Poojary Sushant**
Course:	**Coursera**	**USN:**	**4AL18EC400**
Topic:	**Mathematics for Machine Learning**	**Semester & Section:**	**6th sem 'B"**
Github Repository:	**Sushant7026**		

FORENOON SESSION DETAILS

Image of session



Report –

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 I 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 them all 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 an n 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 I fill it all in and I've got ann down here. So the first suffix on this matrix, first suffix on all of these elements in the matrix is the row number, and the second one 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, dot, dot, dot, dot, dot, dot, dot, dot, 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 a row 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 row 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 others up 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 j of aij, bjk. So if I add these up over all the possible j's, I'm going to get a11, b11 plus a12, b21, and so on, and so on, 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 a repeated 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 abik. So abik is equal to ai1, b1k, plus ai2, b2k, plus ai3, b3k and so on and so on, 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 then use 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 the matrices have the same number of entries in j, then we can multiply them 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 matrix, 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. I've got the same number of j's in each case, so and then I'm going to be able to do that for all of the possible columns, so I'm going to get something with four columns. And I'm going to be able to do that for the two rows here. I'm going to be able to do that row times that one, is going to get a two by four matrix out. So it's going to have one, two, three, four, one, two, three, four. So I can multiply together these non-square matrices if I want to, and I'll get, in the general case, some other non-square matrix. I'm going to have the number of rows of the one on the left and the number of columns of the one on the right.

Now, all sorts of matrix properties that you might want, inverses and so on, determinants, all start to get messy and mucky, and you somehow can't even compute them when you're doing this sort of thing. But there are times when you want to do it. And the Einstein summation convention makes it very easy to see how you do it, and how it's going to work. As long as you got the same number of j's, you're good, you can multiply them together. Now, let's revisit the dot product in light of the summation convention.

So if we've got two vectors, let's call them u and v, and we'll say, u is a column vector having elements ui and v is another column vector having elements vi. And when we dot them together, what we're doing is we're multiplying u1 by v1, adding u2, v2, all the way up. So in the summation convention, that's just ui, vi, while we repeat over all the i's and add. But this is just like writing u as a row matrix pushing u over from being a vector to being a matrix with elements u1, u2, all the way up to un, and multiplying it by another matrix v1, v2, all the way up to vn. That's to say that matrix multiplication is the same thing as the dot product. I'll just push the u vector over, and then my dot product is just like doing a matrix multiplication, which is sort of neat. There's some equivalence between a matrix transformation, a matrix multiplication, and the dot product. So let's look at that. Now, if I take a unit vector here, let's call him u hat, with components u1 and u2. And let's imagine what happens if I dot him with the axis vector. So if I've got an axis here e1 hat, which would be one, zero. And I've got another axis here e2 hat, which will be zero, one. Now, let's think about what happens when I dot u hat with e1. When I do the projection of u hat onto e1. So when I drop u hat down onto the axis here, when I do the projection of u hat onto e1, I'm going to get the length here just of u1, just of the x-axis element of u hat. Now, what happens if I drop, project e1 onto u hat. Well, I'm then going to get this projection. I'm going to get a length here, this projected length here. Now, the fun thing is, we can actually draw a line of symmetry here through these two projections where they cross. And this little triangle and this little triangle are actually the same. You can go and do a bit of geometry and prove to yourself that that's true. So this length here, this projection, that projection is the same length as that projection, which is implied by the dot product. If I dot e1 with u hat, when I do this multiplication here, it's symmetric, I can flip around and I get the same answer. So this show geometrically why that's true. And if we repeat this with the other axes, with e2 here or any other axes there are, then we'll also get the same results. So this is why the projection is symmetric and the dot product is symmetric and why projection is the dot product. So there is this connection between this numerical thing, matrix multiplication, and this geometric thing, projection. Which is quite beautiful and mind blowing really. And that's why we talk about a matrix multiplication with a vector as being the projection of that vector onto the vectors composing the matrix, the columns of the matrix. So what we've done in this video is look at the summation convention, which is a compact and computationally useful, but not very visual way to write down matrix operations. And that's opened up looking at funny shaped matrices, and that's opened up re-examining the dot product here. So that's really nice.