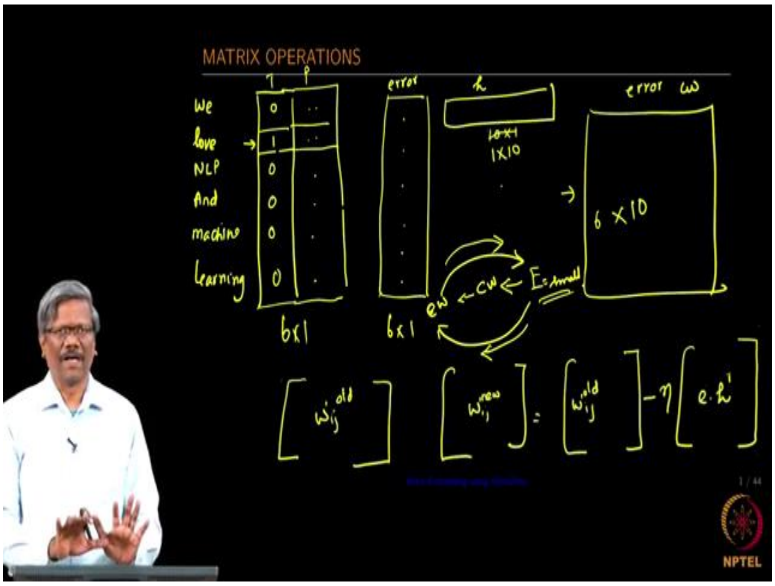
Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute

Lecture – 43
Matrix Operations Explained

(Refer Slide Time: 00:15)



MATRIX OPERATIONS

Alright, in the previous slide, we saw how the softmax was computed. So, now, we see how the difference is computed through the matrix form and then see how that difference in the target versus the computed value is taken through the computations of our weights in the embedding as well as that as the context layers right ok. So, let me redraw that one more time. So, I am going to be using we love NLP and ok; so, we will just first get the target value, the target value you are going to be, this is what we have used as the context. So, the target is going to be this right.

So, this value should be equal to 1 in the real situation ok. So, this is our target and then through softmax, we have some values associated with each of these ok. So now, what we do is, we have to find the difference in the now, we need to find the difference between the target and the predicted value. So, this is our error right. So, the error is also going to have the same size as the target and the predicted value. So, we have 1 2 3 4 5 and 6 values. So now, having computed the error, we have to update the context weight first right.

So, how do you get the difference in the context weight? Just taking the value as we have computed earlier so, we have to do this. So, we take to let me add some space here. So, this is our h right and this is of size 10 ×1 correct. So, when you compute this, what you are going to have is the error that you want to propagate for context weight alright. So, when you do that you get another matrix of size 6 ×10 right. So, and this is what you are going to be using to really update the context weight that you have.

So, we are starting to remember, we have error first and then from there based on the error, we are going to update the context weight and then again based on the error, we are going to compute the embedded weights ok. So, this keeps going until, the error reaches the minimum value correct. So, that is what is our backpropagation approach. So, we do the backpropagation, do the forward pass and then keep going and going and going ok, until this becomes small alright.

So, in the matrix form how do you really get this? So, this is my let, I am going to only write for the context matrix ok, this is our old one and now, we have the difference. So, to get the new one so what we do is; we have w_{1j} a new equal to so, we have the w I j old minus you have a learning parameter and then the difference that you have here right here e doth. So, once you do this, now you have updated the context weight correct and then taking the error back to the embedding layer, we can update the embedding layer weights in the same fashion. So, once it is updated so you keep going back and forth until e becomes small.

I keep repeating this so that you know it gets into the minds clearly ok. So, for many people you know, the equations are good enough to understand this for some folks matrix operations. You know in this form that I have shown is good enough for others the programs are important. If you show them through the typical computer program neither in C or python they grasp it a lot better. So, I am going to be giving the second third flavor as well in this exercise after this ok.

I hope you understand this right. So, this is a very simple operation that you perform to identify the embedding layer weight and also finding what exactly is the target, that we are looking for. The only problem with this is when the size of the vocabulary increases, the size of these matrices are going to increase and it is only a computational problem