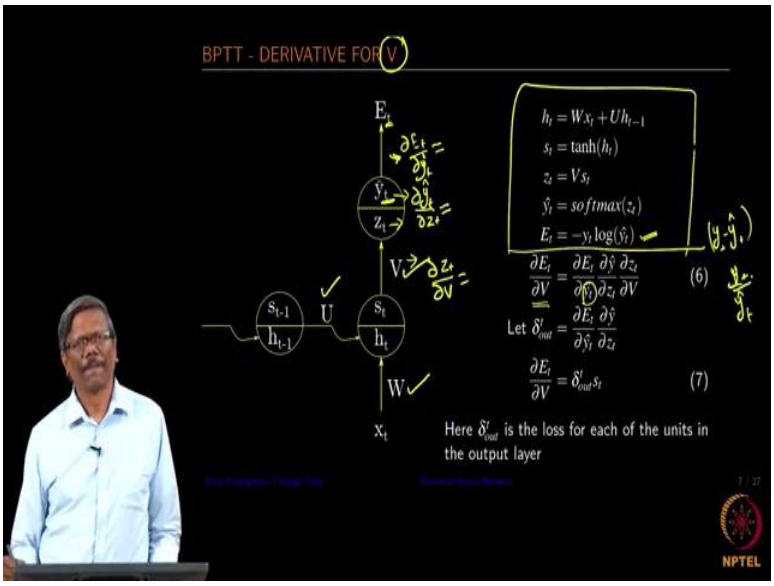
Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute, Madras

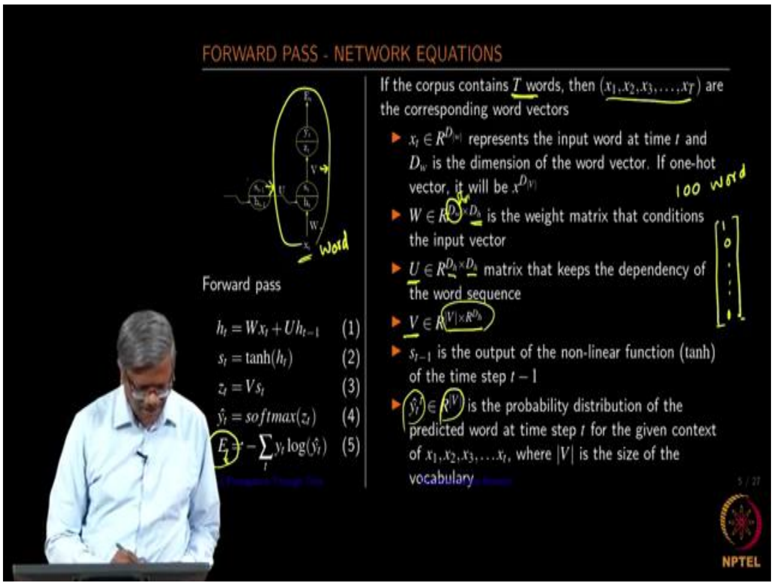
Lecture - 58
BPTT - Derivatives for W, V and U

(Refer Slide Time: 00:15)



So, let us look at the derivatives now.

(Refer Slide Time: 00:20)



So, what we want to find out is to minimize this error right this should be E_t. So, in order for is to find these smallest values of the derivative of E_t vbc , we have to do the derivation for all the parameters that we are looking at DV rather V W and U. So, let us start from the first parameter ok. So, we want to find out what is the change that we want to make when there is a error that coming from the output layer ok. So, I have listed this for our convenience.

Let us look at you E / douV. So, we are taking these errors at state t right. So, we have to do E_t/douv which would be equal to. So, we are going to be finding the error with respect to yt right. So, we have dou E/ dou yt and then we are going to be finding this yt with respect to E zt and then we are going to be finding the change with respect to V.

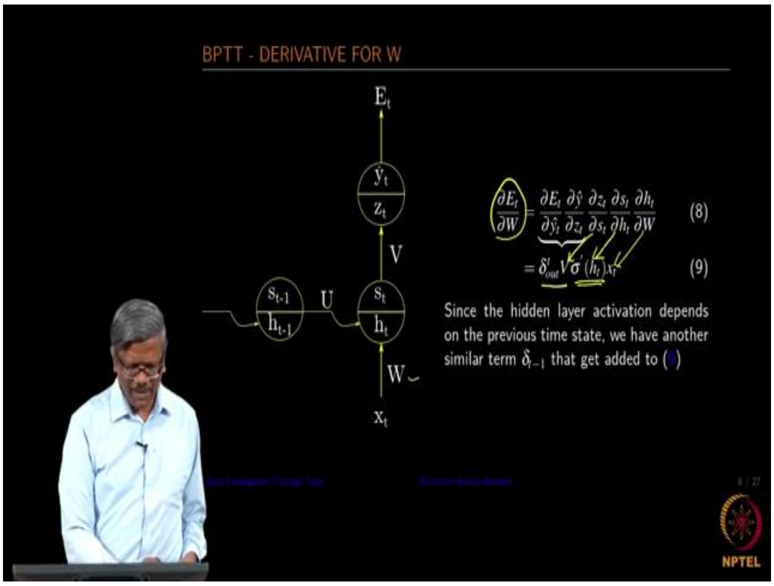
So, it is a chain rule that you can apply wherein you will have this dou E there and then here you will write you y that by dou zt and then here you will have dou zt / dou V.

So, we have to find these values individually and then multiply that you will get you Et /dou V ok. So, we know that when we differentiate with respect to this ok. So, this is nothing, but change in this value right. So, if we use a mean square value this is going to be y minus y hat or it will be 1/y if you are is going to using a cross-entropy model this becomes 1 because we are going be using a 1 hat vector if we use one hat vector, this will become 1 otherwise it will be y t ok.

So, let us not really worry about what we are going to be taking on the right side. So, we will just have the notation for that and then use it in the computation of the derivative for V all right. So, what we have done here is. So, we are going to be taking these two and then calling it as delta out t. So, this is going to be the loss for each of the units in the output layer here ok.

For each of the units in the output layer. So, we call it as delta out t and then when you differentiate this when you do the partial differentiation of d y t by d z t what you get? Oh, I am sorry. So, this is what we are calling it as delta tout. So, we are going to be doing the partial derivative of dou z t with respect to V. So, what you get is s t. So, the error with respect to the value here is delta tout into s t all right. So, this is the derivative for V.

(Refer Slide Time: 04:19)

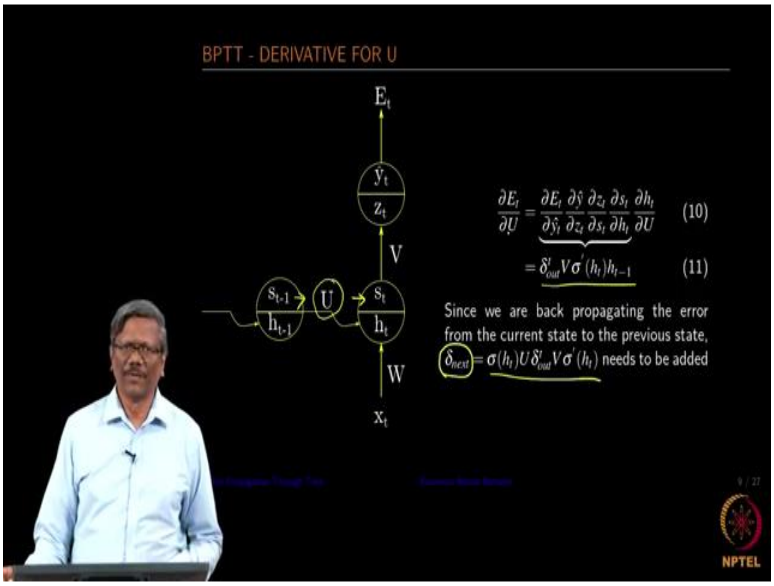


And then now let us look at the derivative for W, it is the similar fashion we are going to be using the chain rule to find out what is the change that we require. So, that we can update W with that particular change and this is the change that you want to find out with using which we will update the W.

Again use the chain rule when I am going to leave this to you as an exercise to find out how these values are obtained. So, in this case I am not completely changing these values I am just using a sigma dash this is a derivative I am just leaving it as is. So, if you can actually expand.

When you want to write your computer application using this ok. So, dou E t by dou W is obtained using the chain rule and then the value would be delta tout V, sigma dash of ht and next is the x t comes from here and this is what you get from dou s t by dou s t what you get from here. So, you may want to expand this and then see what the actual value is ok.

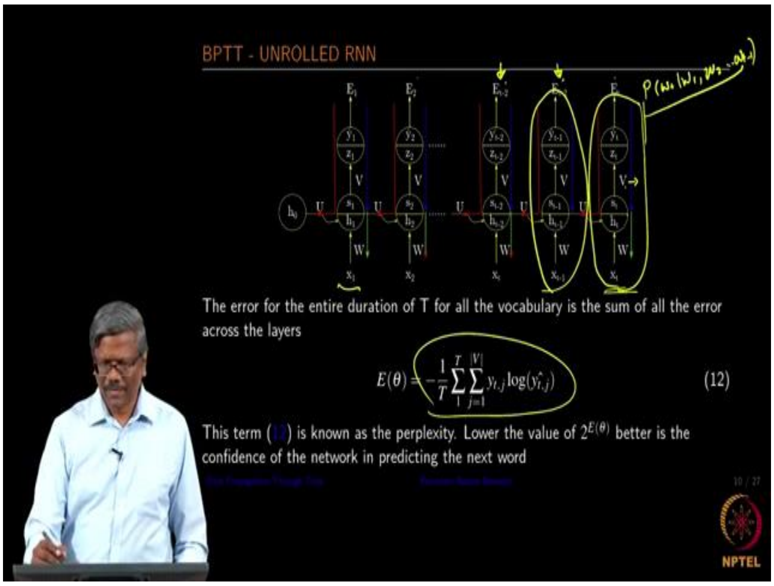
(Refer Slide Time: 06:03)



Let us move on to the next one. So, now, we have to find the derivative with respect to U so, that we can update are the matrix U here. Again apply the chain rule. So, we are going to be getting values similar to this when you are doing the backpropagation you will notice that this particular state depends on the previous state right. So, there is some contribution that is coming from the previous state. So, we need to make sure that the value of the previous state also is included in this particular computation.

So, that is what I write here as delta next this value will turn out to be and this needs to be added to dou E t by dou U all right.

(Refer Slide Time: 06:59)



So, when I unroll this we have done only for one small piece in the RNN. So, now, we unroll it. So, now, this is how the entire network looks ok. So, we have from the timestamp of 1 to time slice t ok. So, we have all the states now available.

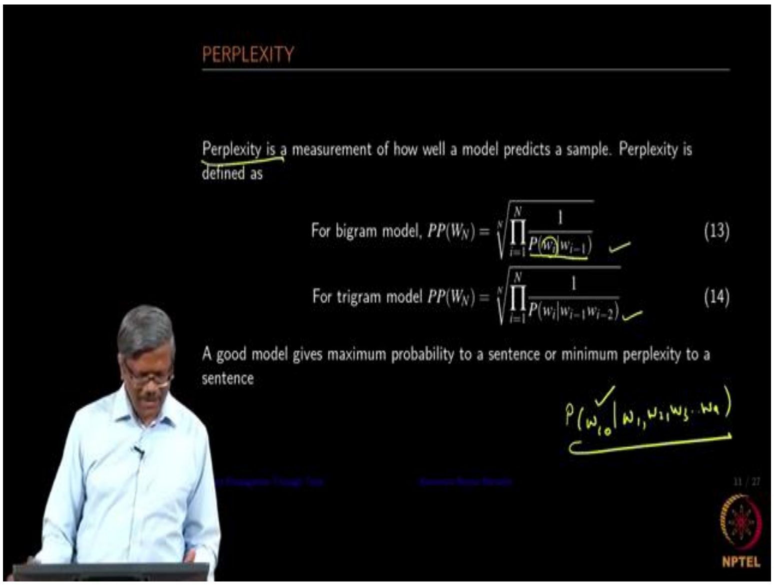
So, how do you do the backpropagation in this case? We have done it for one now we have to see how can be extended for the entire talk. So, since they are unrolled now when they are enrolled you do not see any of those, but when you when they are not enrolled you see like this, when they are unrolled you see what you are seeing on the screen right like this and then every unit or every state you should do the operation of what we are done earlier the error for the entire duration T is obtained using this relationship ok.

So, that means, the error is summed across. So, every one that you find here they are summed and then the error with respect to those parameters should be minimized ok. So, that is the actual aim of this backpropagation through time all right. So, this has to be continued until we reach a state where there cannot be any more updates possible we are able to do the backpropagation through time in this fashion.

Once in neural networks settles we will stop or when the error becomes very small we stop or when the error does not change beyond a point we will stop again if you look at this parameter when this become smaller and smaller the confusion in terms of what is the pattern that I want to identify you know within the network, I am talking about network looking at the patterns right. The confusion for it is going away when the value of E becomes smaller and smaller ok.

So, that is what we call it as perplexity. So, when this becomes smaller we know that network now has a lot of confidence in terms of identifying or predicting the next word if you are using it for natural language processing application ok.

(Refer Slide Time: 09:41)



So, let us text look at what this is in terms of probability. See if I am sure you remember this right probability of predicting this word I given I minus 1 or this is for the bigram right and this is for the trigram and then if you use a character-based model that we are looking at right now, let us say I am going to be looking at the 10th character and I want to predict the 10th character. So, in this model what we are actually computing is this correct this is what we are to computing.

So, if in order for you to compute this and then identify that the machine has a lot of confidence in terms of protecting the right value, we need some kind of a mechanism to measure that right. So, and that what we use here. So, it is very similar to what we are here finding is nothing but right this is what we are trying to find out. So, once the prediction is right we move on to the next job correct. So, the perplexity is computed using this formula the lower the perplexity better it is all right.