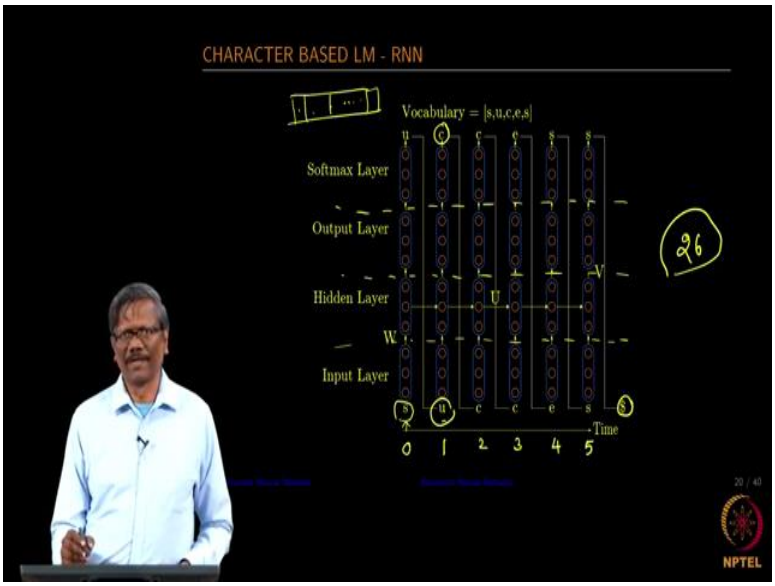
Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute, Madras

Lecture - 56
RNN-Based language model

(Refer Slide Time: 00:15)



So, here in this slide we are going to be talking about the language model based on the character's rights. Let us we discussed earlier in the prediction of word or prediction of a word that you want to find out using this white keyboard or the standard keyboard that you see in many of the recent mobile phones, how do they really capture that and then provide you with a prediction. So, this is one of the simple examples through which you can predict the output.

So, I am going to be providing the input as a character in this case. So, in this case since we are providing the character we are going to be having 26 letters as here vocabulary. So, we are going to be taking the first character, for example, I am going to use the word success ok. I will start with s and then we have all randomly initialized to values initially. So, you compute the activation of the hidden layer, using the one heart vector that you are inputting for s right and then once the activation is computed the output value of y is computed through the softmax layer.

So, the next character that I want to have after ss u this is what I am expecting. So, softmax will give you something values for all the characters. So, when you compute that, you get various values based on this softmax, then since you know that it is not the same as what you expected you send the error backpropagate make the correction, and so on right.

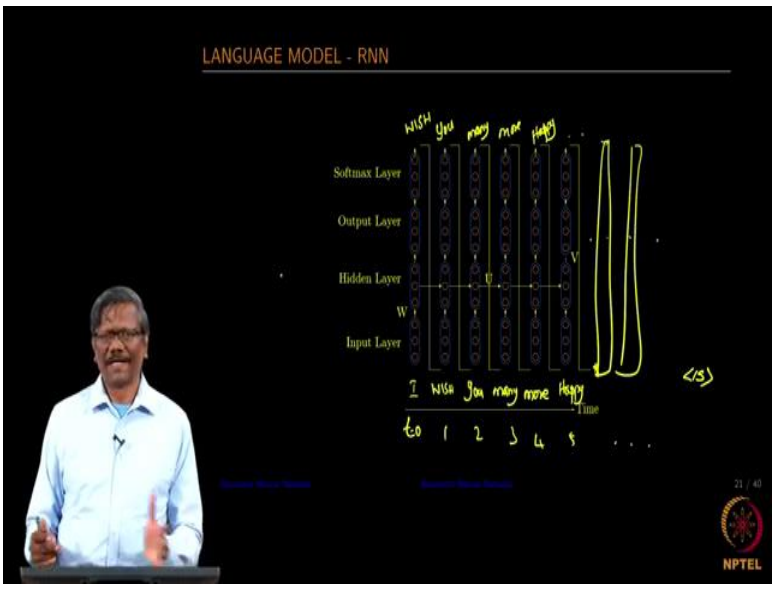
So, the next word is going to be u there is input, the next predicted sorry the next character is going to be u the next predicted character should be u should be c at the output and you keep going in this fashion right.

So, you feed one character at a time. So, these let's say that this is time slice 0 when you encounter some stop symbol you know that you have read the whole word. So, what happens is one this is strained when somebody starts inputting s, it is going to be looking at various there are options with respect to start character as s right.

So, when you type the second one the system can pick up words that start with s and u and then start providing suggestions at the bottom for you to pick up ok. So, in this fashion you will be able to predict the words once the network is at right ok.

This is one model, this we have not spoken about earlier right the character-based model.

(Refer Slide Time: 03:55)



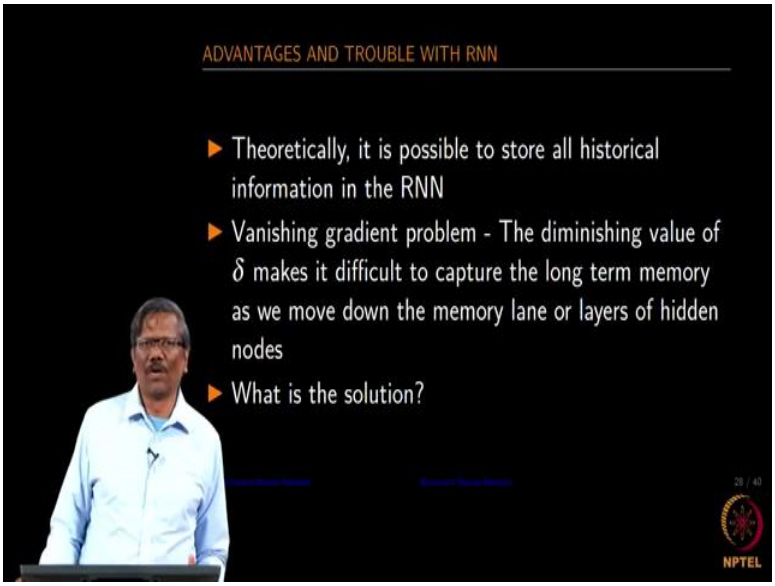
This another one is using the word. So, you can have words like I I am sorry should be more and it goes like this ok. So, we have some more like this then we will have the end of the sentence.

So, we start inputting words in this fashion one word at a time in time slice equal to 0, 1, 2, 3, 4 and so on depending on how many other words that you have in this sentence.

So, in this way it is flexible, you can increase the number of time slices, and then you can make the system learn and finally, you have a model using which you can predict the next word that appears all right. So, this is very similar to what we did with the language model using character ok.

There are a lot of you can see on the internet, there are a lot of examples with respect to generating text using RNN.

(Refer Slide Time: 05:57)



Theoretically we can store all historical information in the RNN know you can theoretically go for a very very long time series. When you go for a long time series the what is a big problem. So, we get into a problem of vanishing gradient and in some cases we also get into the problem of exploding gradient.