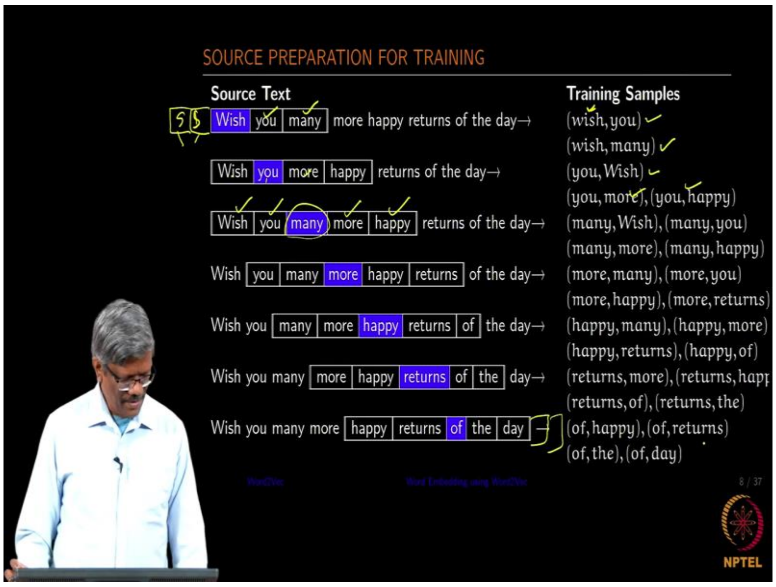
Applied Natural Language Processing

Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute

Lecture - 41
One word learning architecture

(Refer Slide Time: 00:15)



SOURCE PREPARATION FOR TRAINING

Source Text	Training Samples
Wish you many more happy returns of the day→	(wish, you)
	(wish, many)
Wish you more happy returns of the day→	(you, Wish)
	(you, more), (you, happy)
Wish you many more happy returns of the day→	(many, Wish), (many, you)
	(many, more), (many, happy)
Wish you many more happy returns of the day→	(more, many), (more, you)
	(more, happy), (more, returns)
Wish you many more happy returns of the day→	(happy, many), (happy, more)
	(happy, returns), (happy, of)
Wish you many more happy returns of the day→	(returns, more), (returns, hap
	(returns, of), (returns, the)
Wish you many more happy returns of the day→	(of, happy), (of, returns)
	(of, the), (of, day)

8 / 37

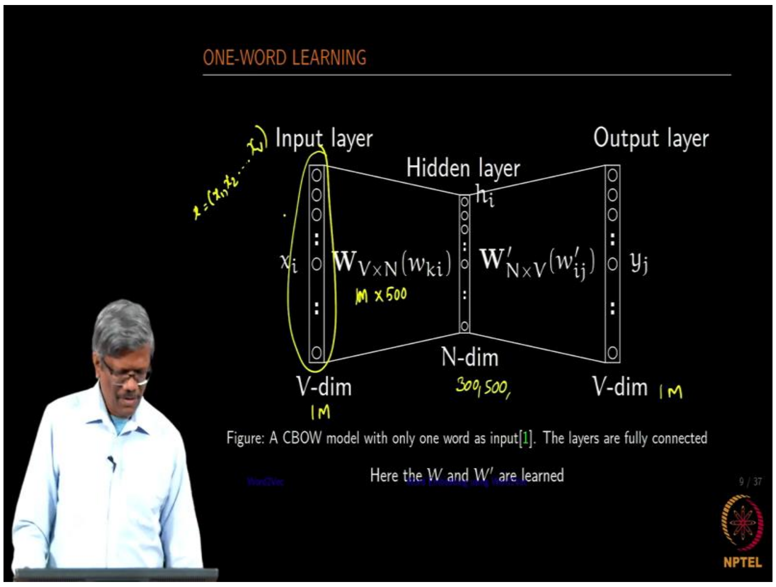
So, now let us see how we can prepare the training data right. This is the most crucial part of any natural language process application right. So, when I give that sentence is given. So, we are given a very cleaned up sentence here. So, there is no need to do a lot of preprocessing with respect to removing certain literals or numbers or whichever are not required for the understanding of the word ok. In this case, we are going to be using a 5 window size and we will be capturing the training samples using that. So, when we start with that 5 gram, so what happens is there is a start symbol in the beginning right, so that when if this becomes the central word. So, there will always be a start symbol and an end symbol like this ok.

So, we start capturing the words that are surrounding the central word ok. So, now, I wish you as one. So, and then we are capturing the bigrams of each of those in this case. And then wish many. So, we have you and many as the context word for the wish. So, we are ignoring these start symbol for the moment here. So, when you move the windows slowly across right, the central word changes and the corresponding context word also change. So, for this case, you are the central word, wish, and you more, you happy correct.

So, again move the windows slide the window to the next word. Now, many are your central word and you have many you, many wish, many more, many happy. So, in this way you keep moving the window across the entire corpus and then form the bigrams of all those words and then create a training sample ok. So, do not worry about the repetition of certain bigrams in this case ok, there will be repetitions, when you start processing a large corpus. So, we do not need to really worry about this point in time.

So, in this case, what you are going to do is we know which one is the central word when we take it, the first one is always the central word right. So, this is going to be input into the neural network. Let us say if you are looking at the CBOW model, so I will use which I will take the corresponding one hot vector related to this and then use that as the input for the network. So, you understand how this is created right, how this training data is created ok.

(Refer Slide Time: 03:29)


ONE-WORD LEARNING

$x = (x_1, x_2 \ldots x_V)$
Input layer
Hidden layer
h_i
Output layer
x_i
$\mathbf{W}_{V \times N}(w_{ki})$
1M x 500
$\mathbf{W}'_{N \times V}(w'_{ij})$
y_j
V-dim
1M
N-dim
300, 500,
V-dim
1M
NPTEL

Figure: A CBOW model with only one word as input[1]. The layers are fully connected

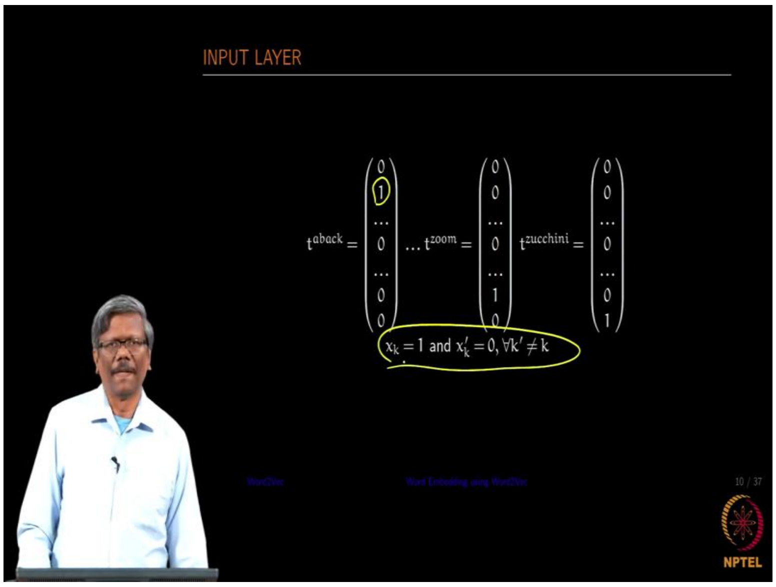
Here the W and W' are learned

So, again I am just giving the same network model, but in a different style so we have not this is a bowtie model. This is the input layer of size V. So, x I would be x equal to x 1, x 2 to x v that would be the size of the input. Supposing, if you have 1 million words as the vocabulary, then the number of elements in the input layer is 1 million ok. Hidden layer size is smaller than the input layer size, we can go up to the value of 300 or if you want still further finite a distinction, you can go up to 500 OKs, so that means, a hidden layer will have 500 elements and we will be connecting 1 million to 300 or 500, so that many weight that you will see so that means, this is 1 million into 500 that means, we will have you will have 500 million connections going.

In the same fashion, since the input layer and the output layer size is going to be the same, both will have the same V dimension. If you are using 1 million, this also would be 1 million. Again the number of elements in the matrix W dash here would be 500 by 1 million OKs, so that is the size of that. Let us not worry about the size at this point in time, let us only worry about how we can really make this neural net functional ok.

As I mentioned earlier there will be only one input we will be provided at a time. So, one word will be input and the weights are initially randomly created and then the hidden layer is computed by using a dot product of the input layer and the weight vector. And then later the weight the output values are computed by again doing the dot product of the hidden layer elements and the weight vectors on the output layer side. So, this is the simple representation of the one word learning neural network using the bag of words ok.

(Refer Slide Time: 06:21)



INPUT LAYER

$$t^{aback} = \begin{pmatrix} 0 \\ 1 \\ \cdots \\ 0 \\ \cdots \\ 0 \\ 0 \end{pmatrix} \ldots t^{zoom} = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ \cdots \\ 1 \\ 0 \end{pmatrix} \; t^{zucchini} = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ \cdots \\ 0 \\ 1 \end{pmatrix}$$

$x_k = 1$ and $x'_k = 0, \forall k' \neq k$

10 / 37

So, as I mentioned earlier, so we are going to be using a one-hot going to be using a one-hot vector as the input element. And then you know well that the values in the one-hot vector would be equal to 1 only once in the entire vector for a given word ok, the rest of the values would be 0. So, it also gives you the index of that word in the vocabulary list. So, this defines how the one-hot vector is given.