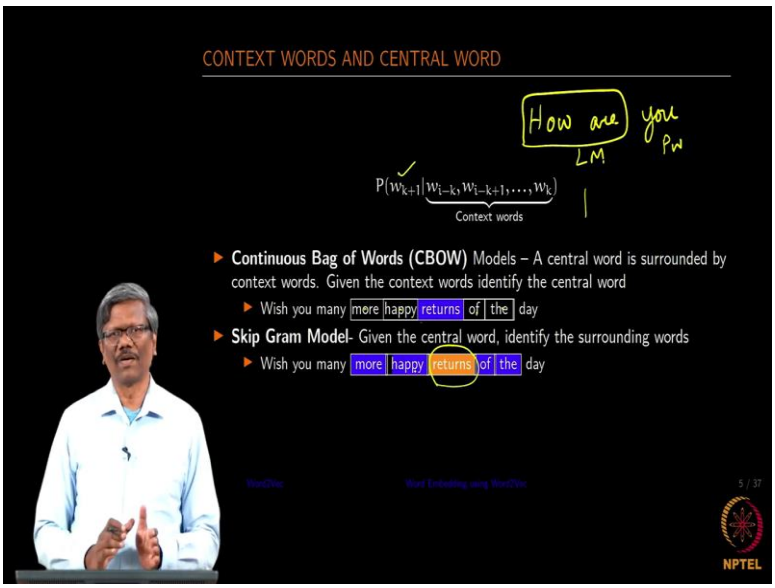
Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute

Lecture – 40
What are CBOW and Skip-Gram Models?

(Refer Slide Time: 00:15)



$$P((w_{k+1} | w_{i-k}, w_{i-k+1} \ldots . w_k)$$

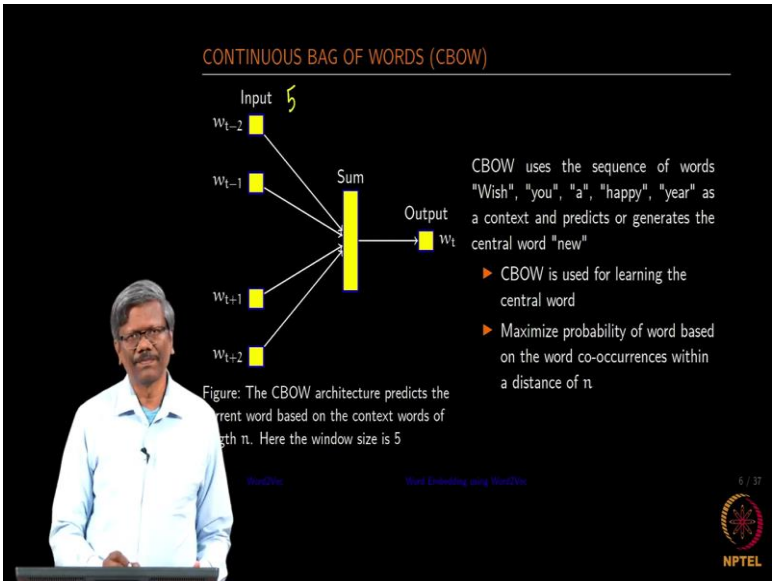
So, now I also we will try to refresh your memory in terms of what are context words and so on, and then describe them with respect to the context word, what is a continuous bag of words, and what is a skip-gram and so on. So, if you look at the context words, and you remember the conditional probability that we used in terms of identifying or predicting the next word in the language model.

So, in the language model the next word is the last word in the context of words correct; that means, when a context of words is given, the next one to the context text is the word that you want to predict. So, in the case of a continuous bag of words, a central word is surrounded by context words.

So, in the case of the language model, let us say so this is your context word in the case of the language model right. And this is what the word you want, this is the word you want to predict correctly. So, now, in the case of a continuous bag of words, we have an example here at the bottom, wish you many more happy returns of the day, this is the sentence that I want to be able to process. Here I am taking the window size as 5 1, 2, 3, 4, 5. And then the central word is the one, which I want to identify given the context words of more happy of the ok. So, given those context words, I want to be able to find out what my central word is ok.

In the case of this skip-gram model, given the central word, I want to be able to predict the context words, surrounding the central word, so that is the difference ok. Let us; so in the case of the continuous bag of words, given the context of the surrounding words, I want to find the central word; in the case of the skip-gram model, given the central word, I want to find the context.

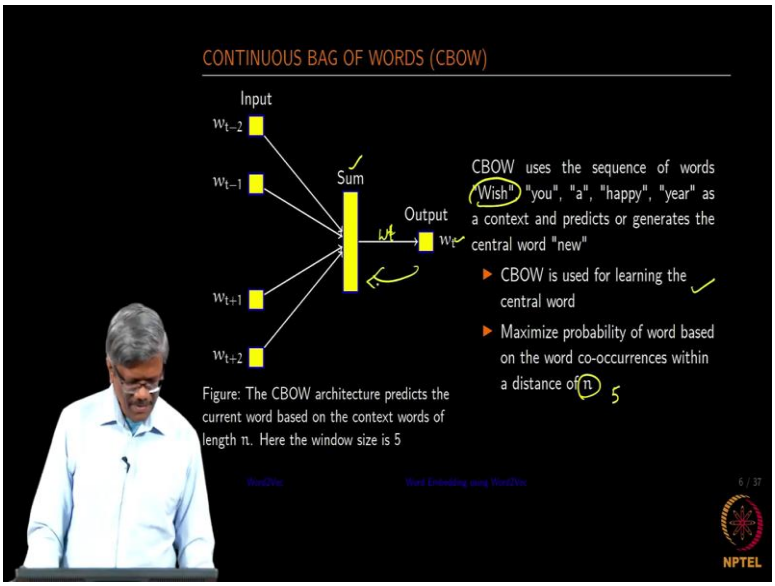
(Refer Slide Time: 02:47)



So, in this case, we have an input layer ok, and then we have a neuron that sums all the incoming weights, and then we have an output that predicts the central word ok. So, in the case of CBOW, we have in this case it is a 5 window word. So, we have w t minus 2, w t minus 1, w t plus 1, w of t plus 2 ok. So, every time we will be inputting one of these to the network, and then the linear sum is taken and then the output word.

So, based on the understanding of the continued continuous bag of words, so we will now define a neural network. So, in the neural network, you are going to be having a context word coming in as the input, and then it is linearly summed using the input and the weights, and then the output is generated using the weights here.

(Refer Slide Time: 03:53)

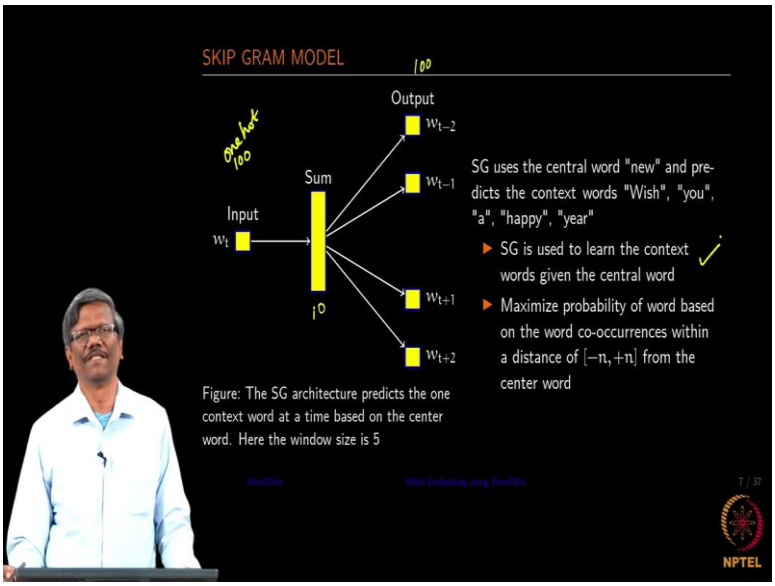


And the hidden layer values. And then finally, we take a softmax to find out what is the right word for us. So, in this case, since we are going to be using the one-hot vector, I will not be able to input all the words together in one shot. So, I will be inputting only one word at a time in this case.

For example, if I take wish as my first word, I look at the one-hot vector related to the wish, and then that will be input here ok. So, as I mentioned earlier CBOW is used for learning the central world. So, we want to maximize the probability of the word based on the word co-occurrence within a distance of n.

So, in this case, it is 5 ok. If the input size is 100, then the output size also would be 100. So, it will be 100 by 1, input also would be 100 by 1. So, they have to match that is when, when you do the soft match, all the 100 values would be output in the output layer, and then you pick up which one you want to really take as the predicted output and then you start doing the backpropagation and back and forth you adjust the weights and calculate the summation, predict the output.

(Refer Slide Time: 05:23)



In the same passion, we would be training this skip-gram model as well. So, in this case, the input is going to be only one right that is the central word. Central word is going to be finding the context surrounding that central word ok. So, the process is the same. Again we will be using a one-hot vector as the input.

And then if our vocabulary size is 100 and there will be 100 elements in this one hot vector, and then we can decide the size of the neurons in the hidden layer let us say 10 OKs, and then the output would be also of the same size as the input. Even though I have given all those things all those context words in this case, we will be computing the context word one at a time, and then the size of the output would also be 100 will be 100 neurons in this case.

Again, in this case, we will be applying a softmax, so that the values are distributed across all the 100 elements, and then we try to maximize the probability of obtaining the right word in the output ok, or we will try to for formulate an error function, and try to minimize this error function and do the backpropagation so that the error is minimized after certain (Refer Time: 06:50).

So, in this case, as I mentioned SG is used to learn the context word given the central word. So, we try to maximize the probability of word based on the word occurrence within the distance of minus n plus n from the central word.