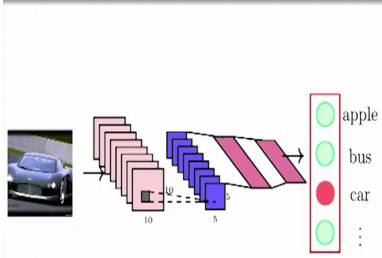


**Deep Learning**  
**Prof. Mitesh M. Khapra**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture – 13**  
**Sequence Learning Problems, Recurrent Neural Networks, Backpropagation Through Time (BPTT), Vanishing and Exploding Gradients, Truncated BPTT**

In this lecture we will talk about Sequence Learning Problems and in particular some neural network architectures which deal with sequences, so recurrent neural networks is what we are going to see. So, we will start with the first module which is on Sequence Learning Problem.

(Refer Slide Time: 00:30)



- In feedforward and convolutional neural networks the size of the input was always fixed
- For example, we fed fixed size ( $32 \times 32$ ) images to convolutional neural networks for image classification
- Similarly in word2vec, we fed a fixed window ( $k$ ) of words to the network
- Further, each input to the network was independent of the previous or future inputs
- For example, the computations, outputs and decisions for two successive images are completely independent of each other

3/41

Mitesh M. Khapra CS7015 (Deep Learning)

So, what are Sequence Learning Problem? So, so far we have dealt with two types of networks one is feedforward neural networks and the other is convolution neural networks and both these networks the input was always of a fixed size.

So, what do I mean by that is, if you take a convolution neural network you are feeding  $32 \times 32$  images to it or  $227 \times 227$  images to it and this size will always fixed. All your training images, all your test images were always scaled or cropped to this particular size, ok. Similarly when we used feedforward neural networks, so one example was word2vec the size of the input was always fixed we had this input of size  $2 \times v$ , right or  $k \times v$  in general; if you are looking at the  $k$  word2vec, right.

So, this input was not varying from one training instance to another training instance or one training instance to the test instance or anything. And secondly, each input to the network was independent to the previous or future inputs. So, I pass an image of an apple I get the prediction apple, then I pass some other image to the network and I get a different prediction. It does not matter whether my previous image was a apple or a car or a mango or whatever it just reads each of these inputs independently there is no dependence between the inputs and the size of the inputs is fixed.

(Refer Slide Time: 01:46)

• In many applications the input is not of a fixed size

• Further successive inputs may not be independent of each other

• For example, consider the task of auto completion

• Given the first character 'd' you want to predict the next character 'e' and so on

Mitesh M. Khapra CS7015 (Deep Learning)

But in many applications the input is not of a fixed size. So, and also successive inputs may not be independent of each other. So, let us understand this with the example of auto completion that all of us are used to while typing SMSs or Whatsapp or other things.

So, given the first character d, I want to predict the next character which is e then once I have predicted e, I want to predict the next character again and so on till I get the full word ok, this is what my task is.

(Refer Slide Time: 02:21)

- Notice a few things
- First, successive inputs are no longer independent (while predicting 'e' you would want to know what the previous input was in addition to the current input)
- Second, the length of the inputs and the number of predictions you need to make is not fixed (for example, "learn", "deep", "machine" have different number of characters)
- Third, each network (orange-blue-green structure) is performing the same task (input : character output : character)

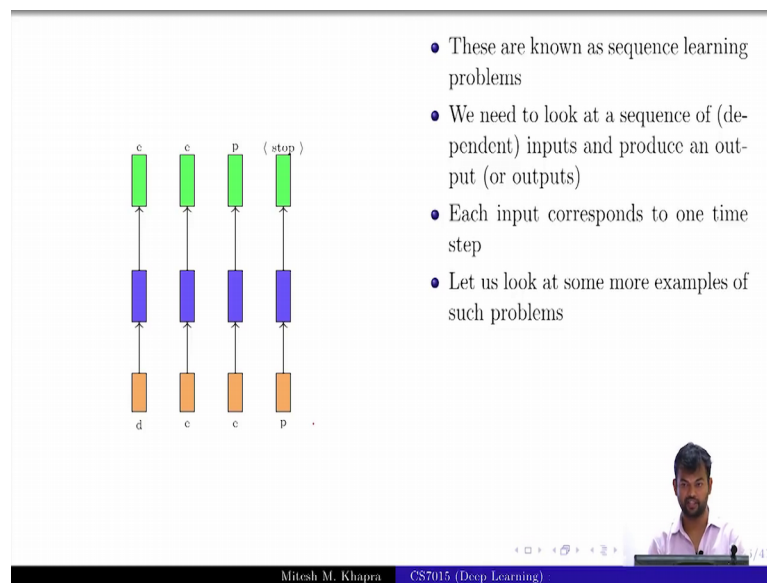
So, let us notice a few things. First successive inputs are no longer independent. If I know that the previous input was d and the correct input is e then I know that only a few things are possible, right. In particular if you know that the previous input was a z and the correct input is a e then most likely the next is going to be a b, right. But if you ignore the previous input which is z, then after e there are many things which can appear, right. So, the inputs are no longer independent of each other.

And the second thing is the length of the input is not fixed because words could be of arbitrary sizes I am trying to type the word deep that is 4 letters or if I am trying to type the learn which is 5 letters machine which is 7 letters and so on, right. So, the input size is no longer fixed and the inputs are now dependent on each other, right; there is some dependence between. So, now, this is very different from what we saw in convolutional neural networks and feedforward neural networks. So, how do we deal with this?

And the third thing here is that. Each network now I am calling this as a network and I will just clarify some notations also soon each network is actually performing the same task. It is taking as input a character and it is producing as output one character. And now just remember that these networks I have drawn them vertically you are used to seeing them as this. So, this is input, this is your hidden layer, and this is your output. So, this is the green part, this is the blue part, and this is the orange input and this is the fully connected layer, right.

So, each of these boxes is actually this network, I have just drawn it more concisely because I need to draw many such networks. So, everyone gets that. Just remember this mind that each of these orange, blue, green, structures is a fully connected network like this.

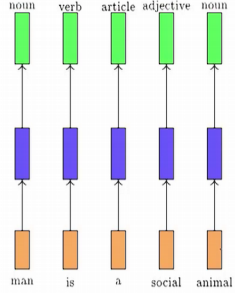
(Refer Slide Time: 04:09)



- These are known as sequence learning problems
- We need to look at a sequence of (dependent) inputs and produce an output (or outputs)
- Each input corresponds to one time step
- Let us look at some more examples of such problems

So, these problems are known as sequence learning problems where you have a sequence of inputs and then you need to produce some outputs. And each input actually corresponds to one time step, so this is the input at time step 1, time step 2, time step 3, time step 4 and so on. So, let us at some more examples of such sequence learning problems.

(Refer Slide Time: 04:29)



The diagram illustrates a recurrent neural network (RNN) processing the sentence "man is a social animal". It shows five time steps. Each time step has an input word (orange box), a hidden state (blue box), and an output part-of-speech tag (green box). The inputs are "man", "is", "a", "social", and "animal". The outputs are "noun", "verb", "article", "adjective", and "noun". Arrows indicate the flow of information from input to hidden state and from hidden state to output. The hidden state is connected to the next time step's hidden state, showing the sequential nature of the network.

- Consider the task of predicting the part of speech tag (noun, adverb, adjective, verb) of each word in a sentence
- Once we see an adjective (social) we are almost sure that the next word should be a noun (man)
- Thus the current output (noun) depends on the current input as well as the previous input
- Further the size of the input is not fixed (sentences could have arbitrary number of words)
- Notice that here we are interested in producing an output at each time step
- Each network is performing the same task (**input** : word, **output** : tag)

Mitesh M. Khapra CS7015 (Deep Learning) 7/41

So, one classic example is the task of predicting the part of speech type of every word in a sentence, right. So, I am given a sentence man is a social animal and for every word I want to predict whether it is a noun or an adverb or an adjective or a verb or any other part of speech type, right and this is how it happens.

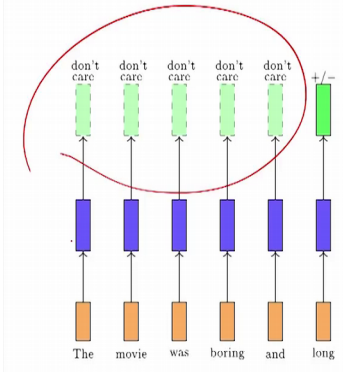
Now, notice that once we see an adjective in this case social we are almost sure that the next what is going to be a noun or at least we are sure that the next word cannot be an article or most likely it will not be a verb, right. There is a very high prior that the next word is going to be a noun. So, that is why these inputs are actually dependent on each other.

So, the current output not only depends on the current input it is also actually depends on the previous input, right. Unlike the case of convolutional neural networks where I feeded an apple it is no dependence on whether the previous input that I pass to the network was an apple or a car or what not. And the size of the input is not fixed because these sentences could be of arbitrary lengths I could have sentences as small as 3 to 4 words or as long as 25 to 30 words, right. So, average Wikipedia sentence for example, is 25 words, roughly 25 words.

And notice that and this case we are interested in producing an output at every time step because for every input I want an output. And each network again this orange, blue,

green structure is performing the same task, its taking as input a word and its producing an output. What is producing? It is a output part of speech tag.

(Refer Slide Time: 06:01)



- Sometimes we may not be interested in producing an output at every stage
- Instead we would look at the full sequence and then produce an output
- For example, consider the task of predicting the polarity of a movie review
- The prediction clearly does not depend only on the last word but also on some words which appear before
- Here again we could think that the network is performing the same task at each step (input : word, output : +/-) but it's just that we don't care about intermediate outputs

Mitesh M. Khapra CS7015 (Deep Learning) 8/11

So, here the two examples that we saw we were having an input and every time step and an output at every time step. But there could also be cases we are interested in producing the output only at sometime steps or at the final time step. So, let us consider task of predicting the polarity of a movie review, right sentiment analysis.

So, I am given a movie review and after I have read the entire review I should give a prediction, right otherwise it would be incomplete. I cannot actually look at only this word and give a prediction does not make sense and it is also not make sense to make a prediction here at this point because there could have been the movie was boring, but I still loved it or but the action was amazing or something like that it because it could have already flipped after that. So, I need to look at the entire sentence and make a prediction. But you are not interested and prediction as these intermediate time steps.

Even in this case you can actually assume that every network is performing the same task its taking a word as an input and it is producing some output it just that till the end you do not care about you outputs you care about the output only produced at the final step. You do not care about what the outputs are at this time step, right that is one way of looking at it.

So, again at every time step we have the same network, but you are only interested in some time steps of the network, ok.

(Refer Slide Time: 07:14)

- Sequences could be composed of anything (not just words)
- For example, a video could be treated as a sequence of images

Finally, it is not always necessary that sequences are composed of only words. What other kinds of sequences are you familiar popular sequences? Speech is one, video is another.

So, a video could be treated as a sequence of images and now you could have a video where someone is performing Surya Namaskar. And as you can understand that I need to look at the entire sequence and only then be able to make a prediction, right. If I stop at this point if I only consider this, this is only Namaskar, no Surya Namaskar, right. So, you have to look at the entire sequence and then decide what the output is and you do not care about the intermediate outputs. I do not care what is the prediction till this point. This of course, is again some aasan but I do not care about that, I care about the full sequence that I am dealing with. It is just to motivate that sequences can be of all types.

And I apologize to the speech people, I do not really understand much of speech processing, so I never give speech examples. But video is something I understand, so I can give examples on that.