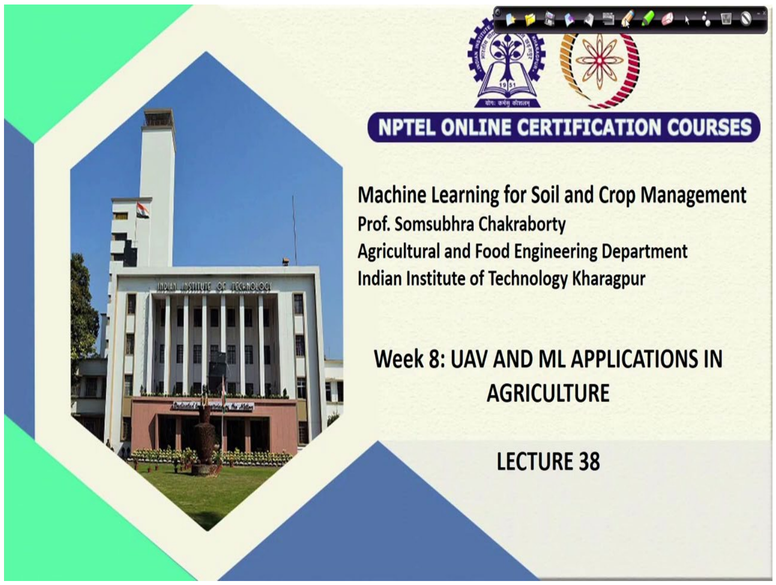
Machine Learning for Soil and Crop Management
Professor Somsubhra Chakraborty
Agricultural and Food Engineering Department
Indian Institute of Technology, Kharagpur
Lecture 38
UAV and ML Applications in Agriculture (Contd.)

(Refer Slide Time: 00:25)

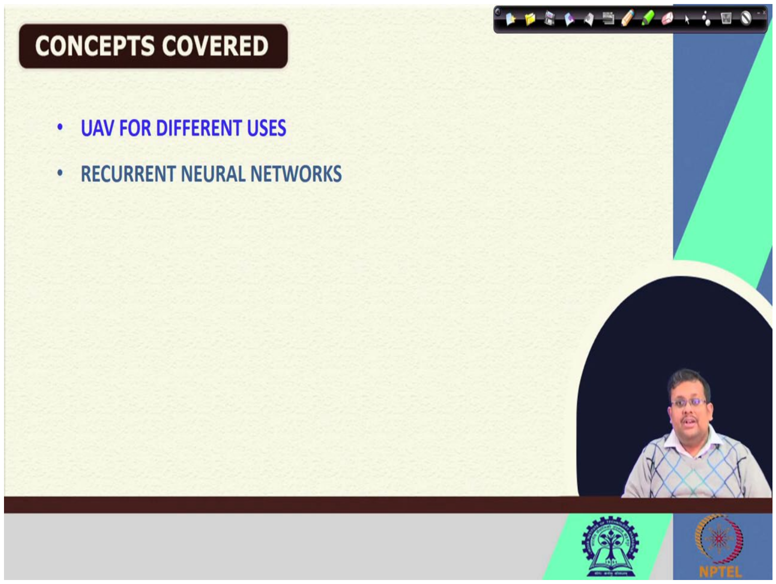


Welcome, friends, to this 3rd lecture of week 8 of this NPTEL online certification course of Machine Learning for Soil and Crop Management. And in this week, our topic is UAV and machine learning applications in agriculture. So, in our first two lectures of this week, we have covered some important aspects. We have covered how we can use the smartphone images for replacing the Munsell soil color chart.

And also we have seen a very good example where Convolutional Neural Network was used along with crop images to identify the weeds in the crop field, or classify the weeds in the crop field. We have seen the details of Convolutional Neural Network and the different types of Convolutional Neural Network like VGC16, ResNet-50, Xception. We have discussed those in details.

And also in the last lecture, we have seen, that is, in the second lecture of this week we have seen the brief overview of UAV's application in agriculture, what are different types of UAVs or drones, what are the different components of drones, what are the different types of sensors, what are the benefits of applying drone in agricultural fields. So, today, in this 3rd lecture, we are going to discuss a very important concept. And these are two important concepts which we are going to cover in this lecture.

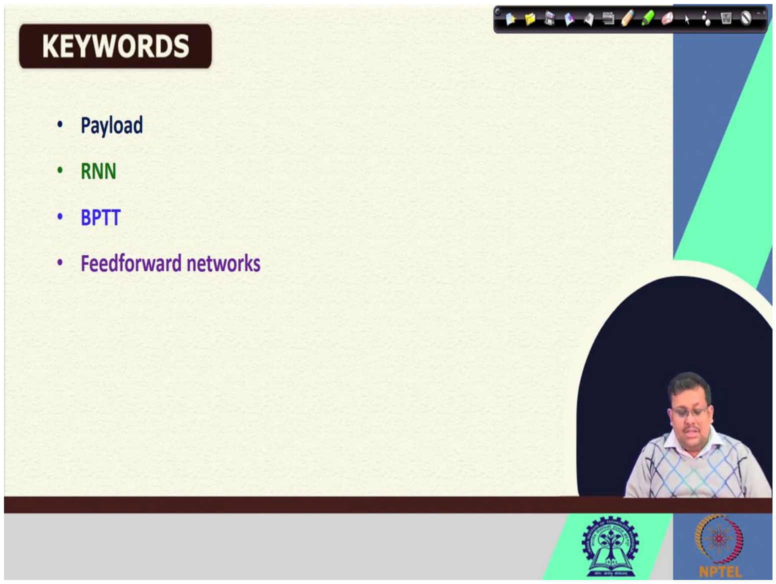
(Refer Slide Time: 0:02:30)



First of all, we are going to use, we are going to see the UAV for different uses, and then we are going to talk about another very important deep learning technique, that is, Recurrent Neural Networks, because this Recurrent, understanding of Recurrent Neural Network will be required for explaining the one of the applications of UAV for agricultural problems.

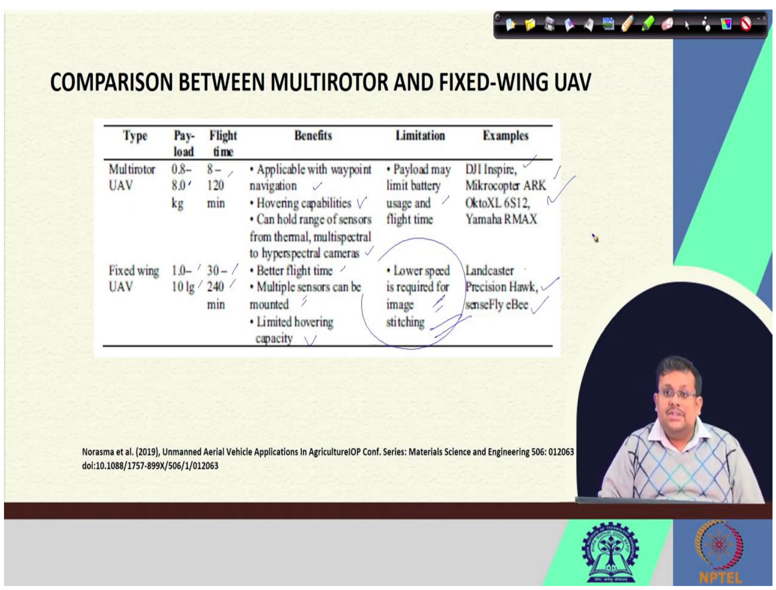
So, and also these Recurrent Neural Networks is one of the most widely used method nowadays, for different types of agricultural applications. So, these two are the major topics for this lecture. And also we are going to cover, So, we are going to cover the theoretical aspects in briefly, for these two points.

(Refer Slide Time: 03:27)



Apart from that, these are some of the keywords for this lecture. We are going to talk about payload, we are going to talk about Recurrent Neural Networks or RNN, then Back Propagation Through Time or BPTT, and then Feedforward networks. So, these are some of the keywords which we are going to discuss in this lecture.

(Refer Slide Time: 03:50)



So, if we start with the comparison between multirotor and fixed-wing UAVs, we have already discussed that what is the difference between a multirotor UAV and fixed-wing wavy UAV in our last lecture. So, remember that these are some of the important points that the payload for multirotor UAVs generally varies from 0.8 to 8 kilos, and whereas in case of fixed-wing UAVs it varies from 1 to 10 kilos.

And then, if we compare the flight time between these two types of UAVs we can see that in case of multirotor UAVs, it goes from 8 to somewhere between 120 minutes, and then for fixed swing it goes from 30 to 240 minutes. Now, each of them has their own benefits.

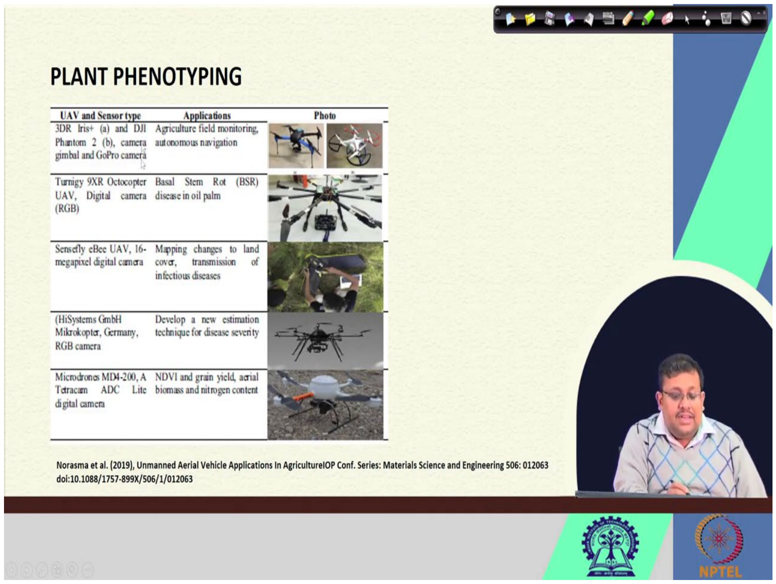
For example, the multirotor UAV is applicable with waypoint navigation. Then, it has hovering capacities, that means it can hover around a certain points of interest, and then, it can hold range of sensors from thermal, multispectral or hyperspectral cameras. So, it, one drone can consist different types of sensor, like thermal sensor, RGB sensor, multispectral sensor, as well as hyper spectral sensors.

However, in case of fixed-wing UAVs they have better flight time. As you can see, the flight time is quite high. And then also here also you can mount multiple sensor but it has limited hovering capacity since it is fixed-wing. Now, these two have their own limitation. First of all the payload in case of multirotor UAV may limit battery usage and flight time.

And in case of fixed-wing UAVs, it has lower speed is required for image stitching. So, image stitching is an important point, important step for UAV based image processing. So, this image stitching is required for subsequent analysis of the acquired images, and of course, for image stitching purpose, it requires lower speed than that of fixed-wing UAV. So, this is one of the limitations.

Some examples of multirotor UAVs which are available in the market are DJI Inspire, then Mikrocopter ARK, OctoXL 6S12, Yamaha RMAX. And if you consider the fixed-wing UAVs, Landcaster, then Precision Hawks, senseFly eBee. So, these are some of the important fixed-wing UAVs which are being used for different agricultural purposes, specifically, for crop monitoring, crop health status identification, and then weed classification and So, on.

(Refer Slide Time: 07:11)



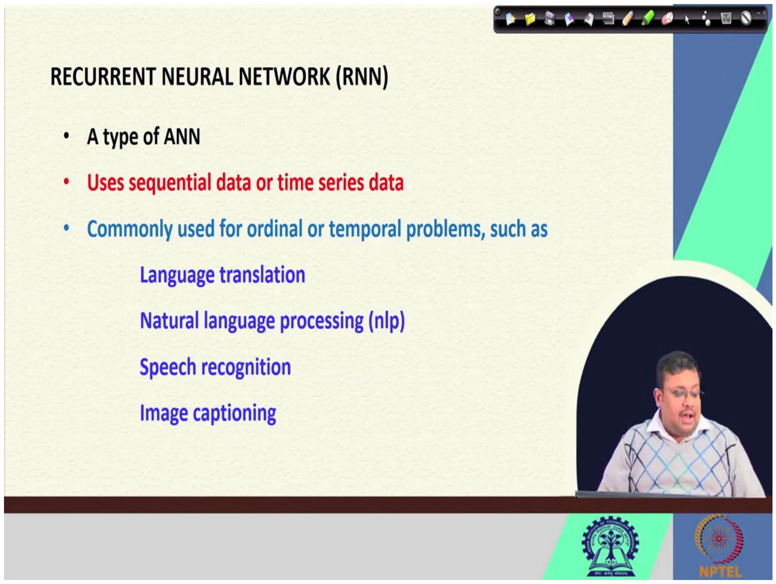
So, let us see some examples of UAV. One of the major example of UAV is plant phenotyping. So, for example some examples are given here. If you consider these 3DR Iris4, and then DJI Phantom 2, with a camera gimbal and GoPro camera. So, these type of UAV and sensor combinations are useful for agriculture film monitoring and autonomous navigation. So, some the photographs are given here.

And then, Turnigy 9XR Octocopter UAVs with the digital camera. They have been, this combination was used for identification of the Basal Stem Rot disease in oil palm. This is the, this is the drone. And then Sensefly eBee UAV with 16 megapixel digital camera was used for mapping changes to land cover, transmission of infected diseases. This is the fixed-wing.

And then these HiSystems GmbH Mikrokopter along with the RGB camera was used to develop a new estimation technique for disease severity. This is the picture of this Mikrokopter. And then, Microdrones MD4-200 with a Tetracam ADC Lite digital camera was used for NDVI and grain yield, we have discussed about the NDVI in last lecture, and also for measuring the aerial biomass and the nitrogen content.

So, you can see some examples of these type of drones for plant phenotyping, and there are other applications are being developed in the agricultural domain using drone based images and subsequent machine learning.

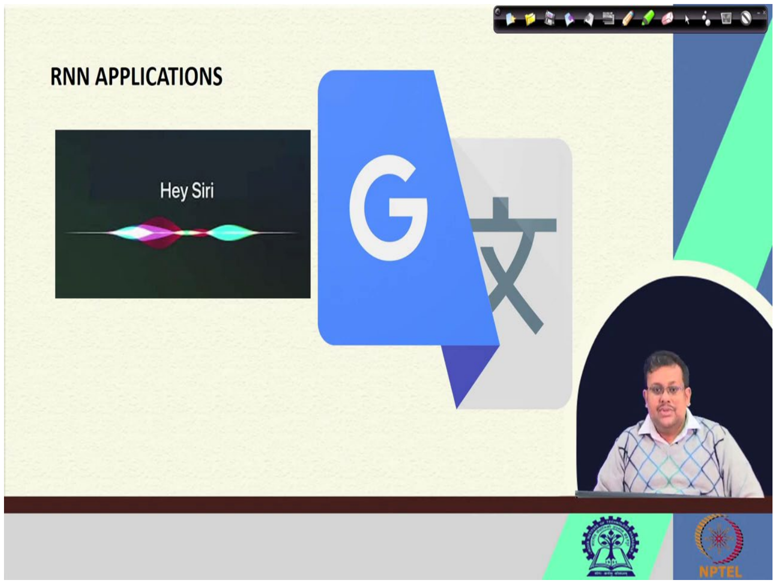
(Refer Slide Time: 09:11)



Now, let us discuss one of the very important and very popular neural network methodology that is called Recurrent Neural Network, or in short form, we call it RNN. So, this RNN is basically a type of artificial neural networks, and it uses sequential data or time series data. So, the feature or the specialized feature of this type of artificial neural networks or RNN is, it uses sequential data or time series data.

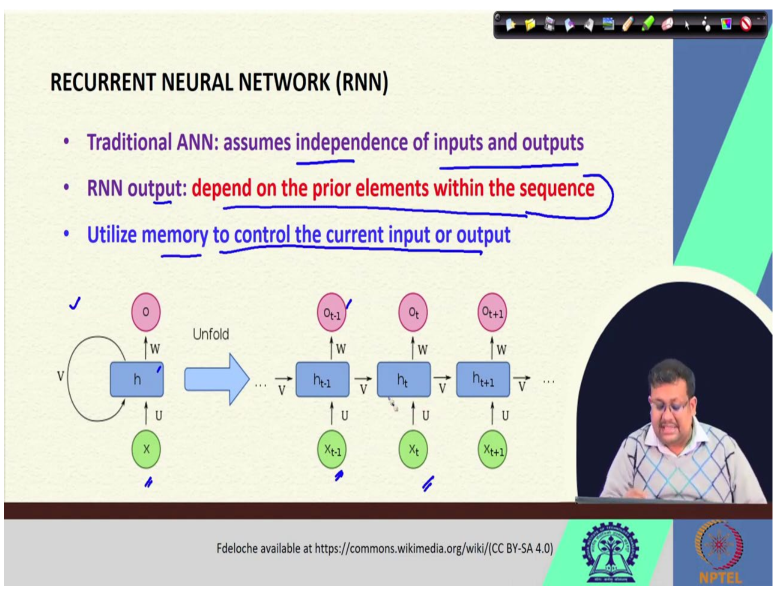
And what are the applications? So, generally, this RNN is being used for solving the temporal problem, that means, the problems which are having some time relationship, that means, some phenomena which changes with time can be solved by using this Recurrent Neural Network. For example, language translation, then natural language processing, speech recognition, image captioning. So, these are some of the popular application with the RNN.

(Refer Slide Time: 10:33)



Now, some of the popular apps or popular programs which we use which uses this RNN methodology is Siri, you know, and also in case of Apple, there is a Siri, and also Google Translate. So, these are some of the good RNN applications nowadays, which we use, almost all the day.

(Refer Slide Time: 11:00)



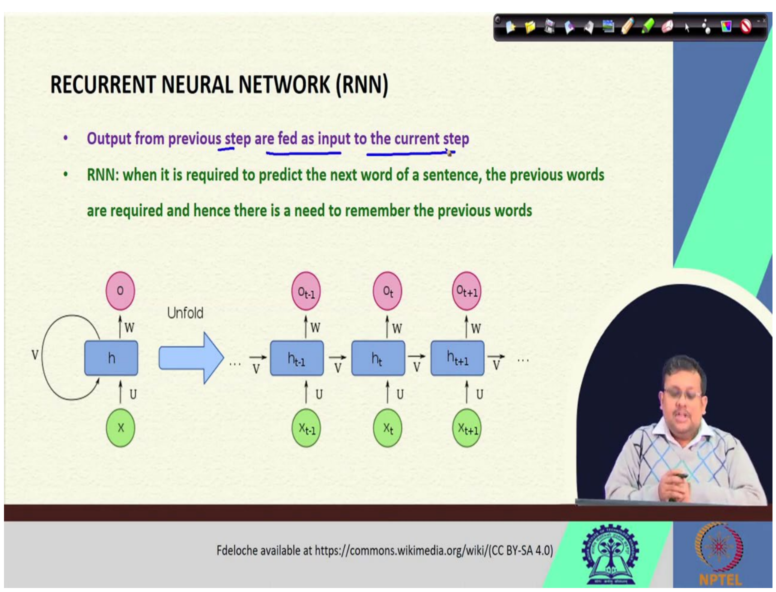
Now, what is the difference? If you see the trend, if we consider the fundamental difference between the artificial neural networks and Recurrent Neural Networks, So, what is the difference between these two types of deep learning methods? So, if we consider, this, just artificial neural network, So, in case of traditional artificial neural network, it assumes independence of inputs and outputs.

So, in case of artificial neural network, normal artificial neural networks, you assume that the inputs or the outputs are independent to the inputs. There is no direct relationship. However, in case of RNN or Recurrent Neural Network the output depends on the prior elements within the sequence. So, the sequence of events, in other words the sequence of events will impact the output of the Recurrent Neural Network.

Because, the Recurrent Neural Network utilizes memory to control the current input or output. In other words, the output from a Recurrent Neural Network from the, the output is dependent on the input and also the present input is dependent on the output from the previous step. So, here, two representations are given. One is called the compress representation, this one, another is called the unfolded representation.

Now in the compressed representation, you can see, this X is the inputs, and this O is output, and these u and w are the weights. So, and this is the hidden layer which is denoted by h. So, in case of compressed representation of this Recurrent Neural Network, we can see that this basically can be represented in this way. However, if we unfold this, we can see that the inputs are influencing the output. Not only, the output is also influencing the next layer. So, we are going to discuss this in details.

(Refer Slide Time: 13:55)

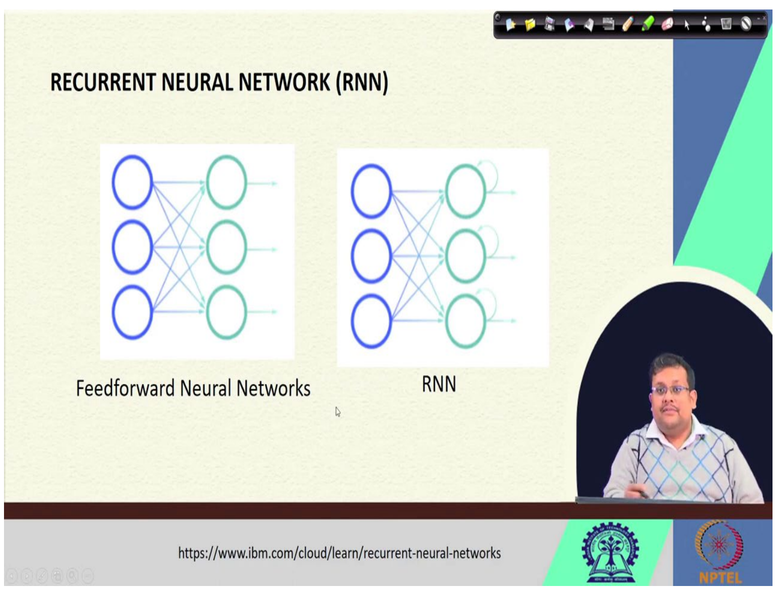


So, if we see, the output from this, this compressed as well as unfolded view, we can see that, output from previous step are fed as inputs to the current step. As I have told you, that it utilizes its memory to determine the input. So, that means it utilizes the output from the previous step to feed as an input in the current step.

So, RNN, in case of RNN, when it is required to predict the next word, suppose there is a sentence, and using the RNN you want to predict the next word in a sentence. Sometime we found that while we are typing any text in our mobile or somewhere in the Google, we can see some suggestions are coming, because they are predicting what should be the upcoming word, most probable upcoming word.

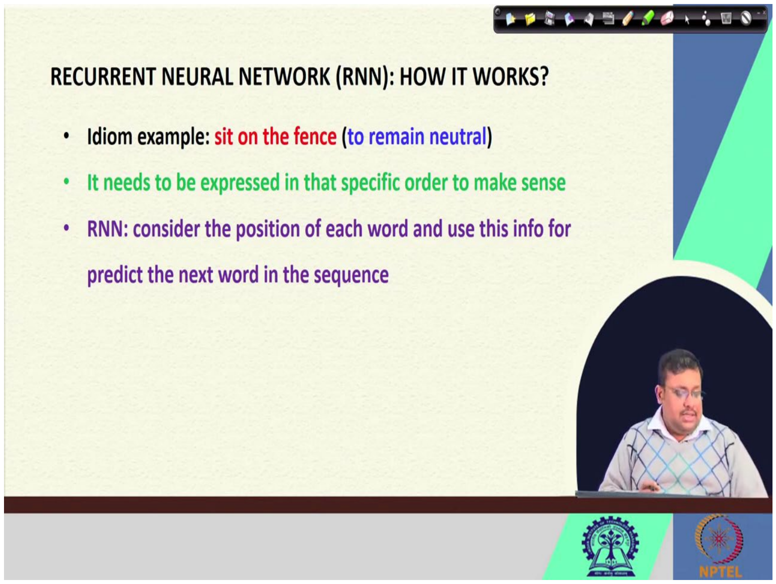
So, this basically uses the RNN where the next word in the sentence is determined by the previous words. So, the previous words are required for determining the next word, possible next word in the sentence, and hence there is a need to remember the previous words. So, these new network basically remembers the previous word So, that it can predict the upcoming next word. So, we are going to discuss this in details.

(Refer Slide Time: 15:35)



So, if you see, this is the difference between a Recurrent Neural Network and Artificial Neural Network. So, in this is the feedforward neural network, we know that the input generally goes in one direction. However, in case of RNN, since we are applying the same weight in all the layers, or same, I would say we are applying the same weightage to all the inputs, So, here the Recurrent Neural Network will be assuming this type of this type of structure.

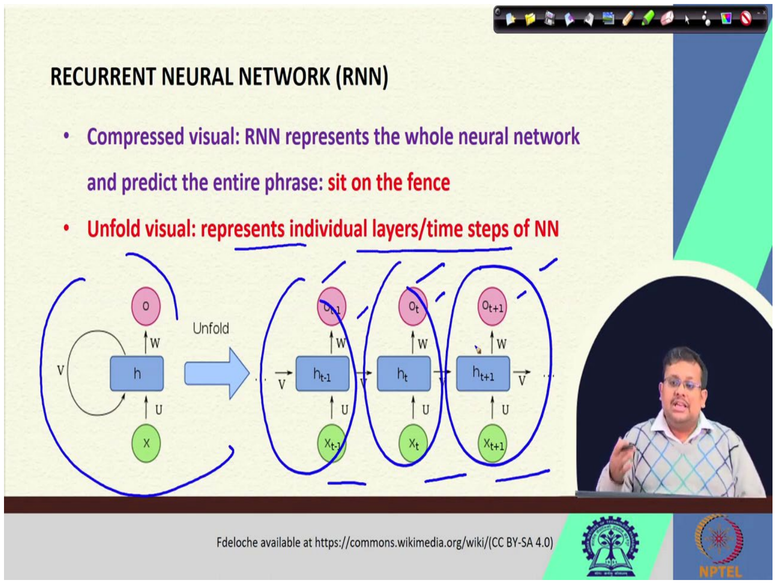
(Refer Slide Time: 16:17)



So, let us see how this Recurrent Neural Network or RNN works. So, let us consider an idiom example, for example. So, there is an idiom very commonly used in English language, that is, sit on the fence. So, literally, it means to remain neutral. Now, this it, needs to be expressed in that specific order to make sense. So, of course, this has to be expressed as sit on the fence. So, in this order, this idiom needs to be expressed so, that it makes sense.

So, in case of RNN, so, it follows an order, so, in case of RNN, it consider the position of it consider the position of each word, and use this information for predicting the next word in the sentence.

(Refer Slide Time: 17:13)



So, if you see, this Recurrent Neural Network structure, as I have told you, that this is the folded structure or compressed structure, where this is input, this is the output, and u, v, w are the weights of the Recurrent Neural Network, and this h is the hidden layer. So, here we can see that, in the unfolded diagram, here, the output of, or input at any particular stage is dependent on the output of the previous step. And here, u, v, w are all the weights.

And we can see, that since in all these layers, these weights are same. u, v, w, u, v, w, u, v, w. That is why, this whole compressed the, compress representation can be utilized in this way, because here we are not assuming that there are different inputs with different weightage. Remember, in case of traditional artificial neural network, we assume that the inputs are having different weightage.

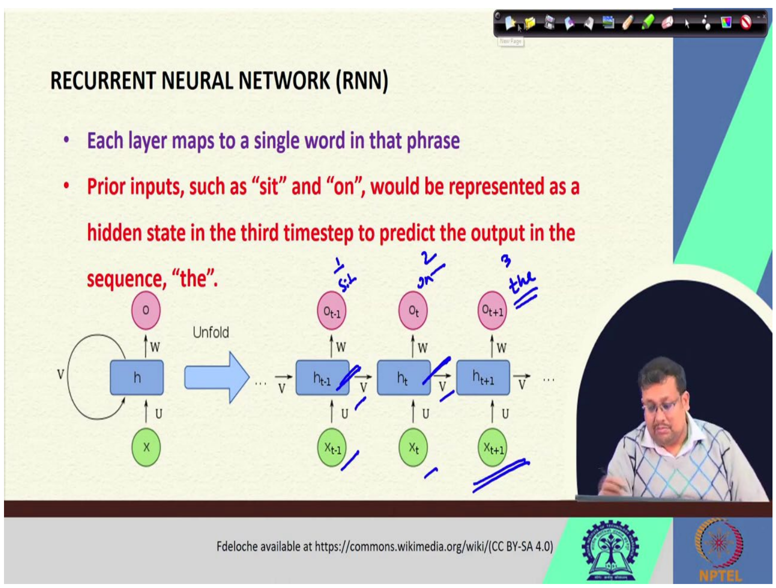
However, here, since all the weightage are same in all these layers, So, we can represent in this fashion. So, here, the idea is, in case of Recurrent Neural Network we can see that this is the compressed representation. And when we unfold, you can see the outputs are from this each, each stage we are getting the output. And this output is dependent on the previous, or we are getting the output from the each stage, and the output from each stage will influence the input of the next stage.

So, in case of unfold visual, we can see that the represents the individual layers or time steps of neural networks. So, this is the individual layers. We can see this is another individual layer, this is another individual layer, or time steps of neural network. So, these will occur step by step sequentially, So, you can see h t minus 1, h t and h t plus 1. So, it will occur temporarily, and that is why Recurrent Neural Network is more suitable for this temporal problems.

I hope now it is clear. So, again, the Recurrent Neural Network is a neural network where we are assigning the same weightage to each individual layer. So, that means the inputs are having the, we are assigning the same weightage in the neural network, and here, the output from one individual layer will sequentially influence the input of the next layer. So, in other words, the RNN basically predicts the next word based on the previous output.

Or, so, this is how it remembers. It has a memory. And that is why it is sequentially solve a problem. Now, we know how this neural network or RNN works.

(Refer Slide Time: 20:46)



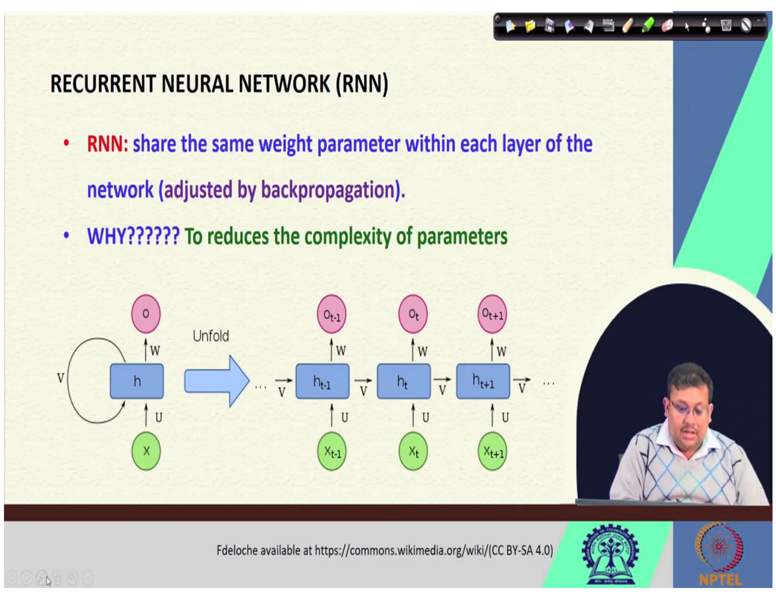
Now let us go ahead and we see that each layer, when this type of layers are there with the individual or uniform weight of u, v, w, we can see that each layer mapped to a single word in the phrase. So, here, you can see the first word. Suppose sit on the fence. So, this first layer will map to sit, the second layer will map to on, the third layer will map to the. So, you can see that the prior inputs, like sit, suppose here we are getting sit, and here we are getting on. So, this prior input, So, Input 1 and Input 2 would be represented as a hidden states. So, these are the hidden states.

You can see these hidden states are influencing this, the next hidden state. So, these hidden states are influencing the next hidden state in the time, third time step. So, this is the third time steps. So, in the third time steps this first hidden step and the second hidden step is influencing to predict the output in the sequence that will be the. So, this is the third output. So, you can see, the first output, second output will help to predict the third output in the whole sentence.

And here, we are assigning the same weight, and here u, v, w. Here, inputs are different, and then we are getting the different outputs in individual layers, but they are predicting sequentially based on the previous output. So, I hope now it is clear to all of you. So, let us move ahead and see that what is the difference between feedforward network and in case of RNN.

Now, in case of feedforward network, if you remember, that they have different weights across each node. So, individual node has different types of weightage, different weights. So, but in case of RNN, share the same weight parameter within each layer of the network, which is of course adjusted by backpropagation. Although they are same, but they are still adjusted by backpropagation. What is backpropagation? We have already seen.

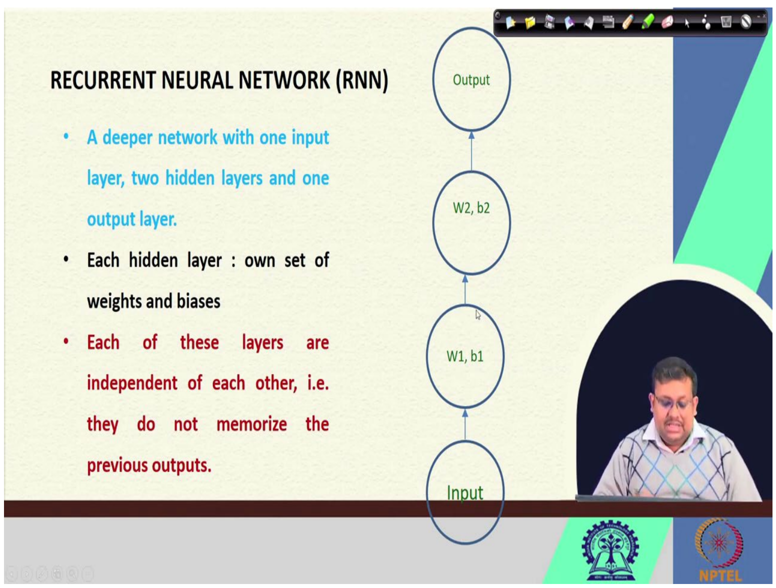
(Refer Slide Time: 23:20)



Now, if we just move ahead and see that in case of RNN, remember, it share the same weight parameter as I have told you, and it has the same weight parameter within each layer of the network, which is, of course adjusted by backpropagation. But why we require the adjustment? Because to reduce the complexity of the parameters. To reduce the complexity of the parameters, they use the same weight parameters within each layer of the network.

So, that is why you can see, they are represented in this fashion where the same u, v, w are revolving in this hidden layer to get the final output. And these are not varying from one layer to another layer. So, this is how this compressed representation is just like this. And again, this RNN assigns the similar weight parameter within each layer of the network just to reduce the complexity of the parameter.

(Refer Slide Time: 24:35)

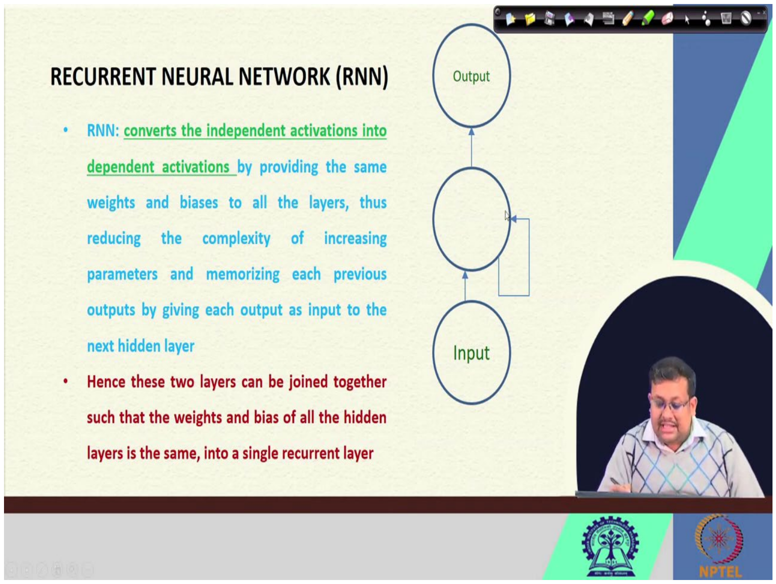


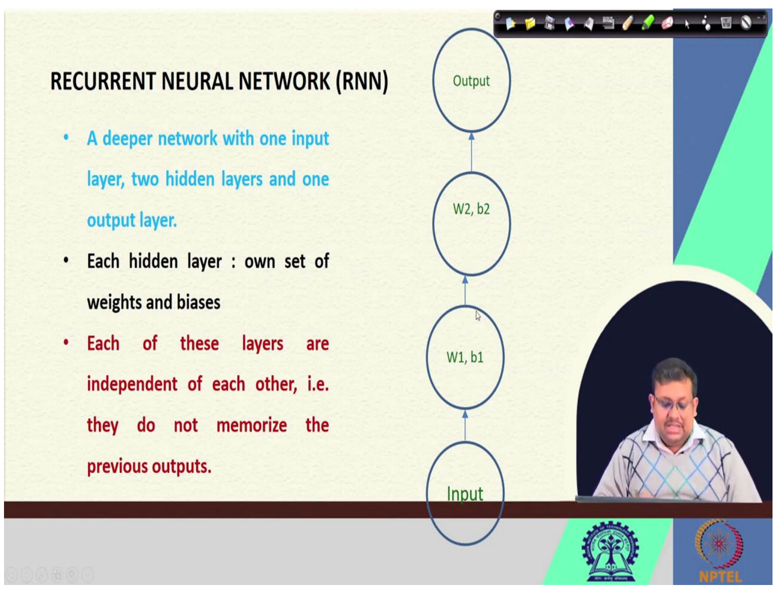
Now, if we see, a RNN structure, of course, here, you can see, in case of a general neural network, or a deep neural network you can see there are inputs and there are hidden layers where we can see the weights and the bias. So, suppose here we are having the weights and the bias in this first node, in the second node, we are having another weight, that is, W2 and B2, and finally the output.

So, we can see that in case of a deeper network, with one input layer, two hidden layers of the output layer. So, here, we can see in this network, deep layer network, we have two hidden layer. This is one, this is another. And this is the output, and this is the input layer. So, each hidden layer is having their own set of weights and bias. Well, as I have already covered this thing while discussing the neural network.

But remember that each of these layers are independent to each other. That is, they do not memorize the previous outputs.

(Refer Slide Time: 25:42)



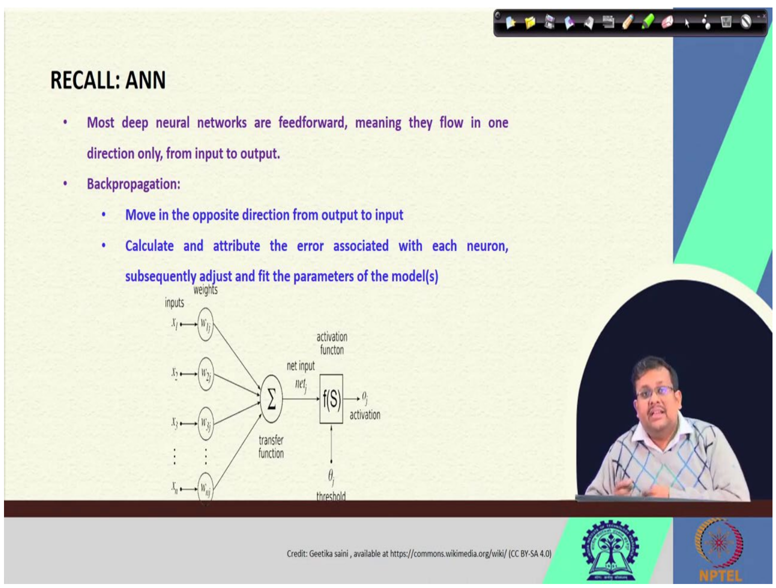


However, in case of Recurrent Neural Network, it converts the independent activations into dependent activations. So, this independent activation function is converted into dependent activation by providing the same weights and biases in all the layers of this Recurrent Neural Network thus, reducing the complexity of increasing parameters and memorizing each previous outputs by giving each output as input to the next hidden layer.

So, that is why they are continuously revolving within the hidden layer, and there is a common temporal linkage between the output and the next input. So, it revolves it goes through sequentially from one layer to another layer, and that is how it basically, this is how it works. So, hence, these two layers can be joined together.

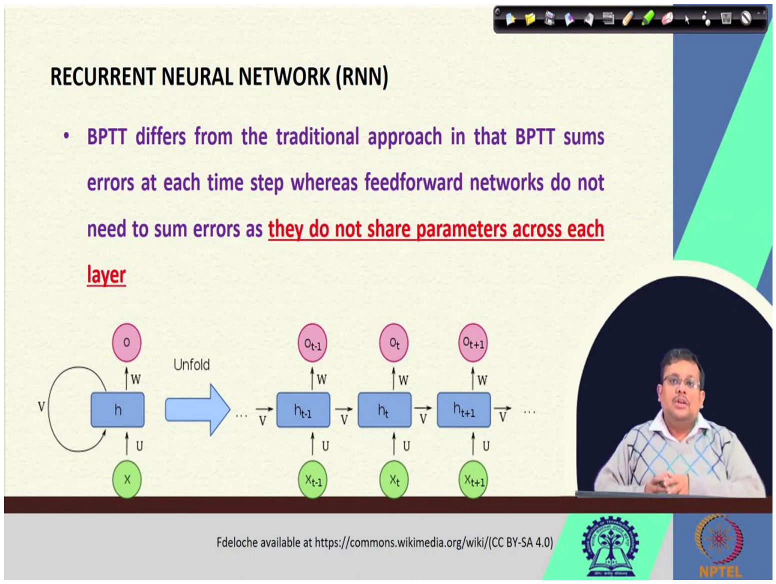
So, here you can see there are two different hidden layers, but here, we are joining these two hidden layers together such that the weights and the biases of all the hidden layers are same into a single recurrent layer. Now, why we are giving this symbol? Because it is recurrent. So, that is why we are using this type of symbol to just to express the Recurrent Neural Network.

(Refer Slide Time: 27:13)



Now, if you recall, that most deep neural networks are feedforward, meaning, they have their flow in one direction only. And there is a term called backpropagation that moves in the opposite direction from output to input. So, in the backpropagation, the weights are adjusted by moving from output to input. So, basically they calculate and attribute the error associated with each neuron, subsequently adjusted feed the parameters in the model.

(Refer Slide Time: 27:47)

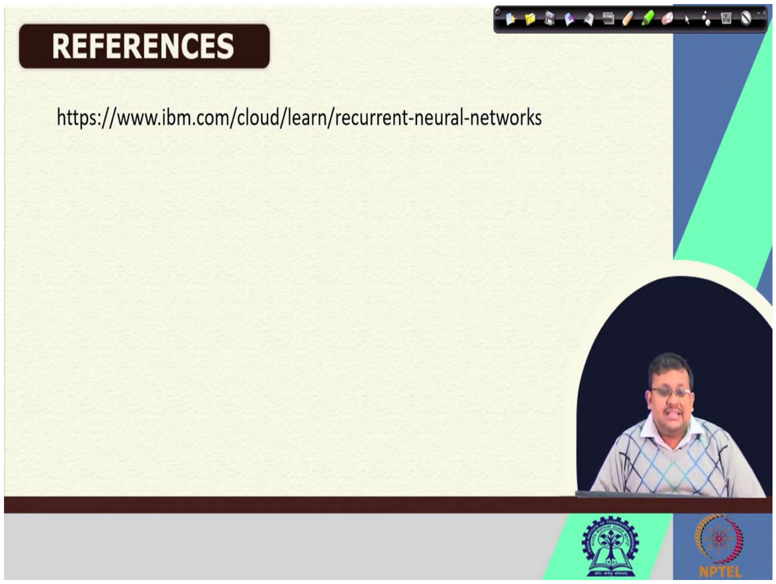


However, in case of Recurrent Neural Network, it leverage back propagation through time. So, there is a term called BPTT, that is, back propagation through time algorithm to determine this gradient. Now, it is slightly different that common artificial neural network because it is specific to sequence data. So, this is how this Recurrent Neural Network adjusts its weights.

Now this BPTT differs from the traditional approach in that, that this backpropagation, this back propagation through time sums error at each time step, whereas in case of normal feedforward method, they do not need to sum error cycle, because they do not have same parameters across each layer.

Since these RNN are having same parameters and same weight in the same, in individual layer so, that is why in each step, we have to calculate the back propagation through time sums of error. But in case of normal artificial neural network, we do not, or feedforward network, they do not need the sum errors to adjust their weights.

(Refer Slide Time: 29:04)



So, guys, this is the reference. And I hope that you have learnt something new in this lecture. And if you are interested, please go ahead and see some more literature regarding the Recurrent Neural Network. I hope I am able to make you understand about the basics of the Recurrent Neural Network.

We will start from here in our next lecture, and we will discuss that what are the different types of Recurrent Neural Network, and then we will start the application of machine learning for UAV based image processing for agriculture. So, let us meet in our next lecture. Thank you.