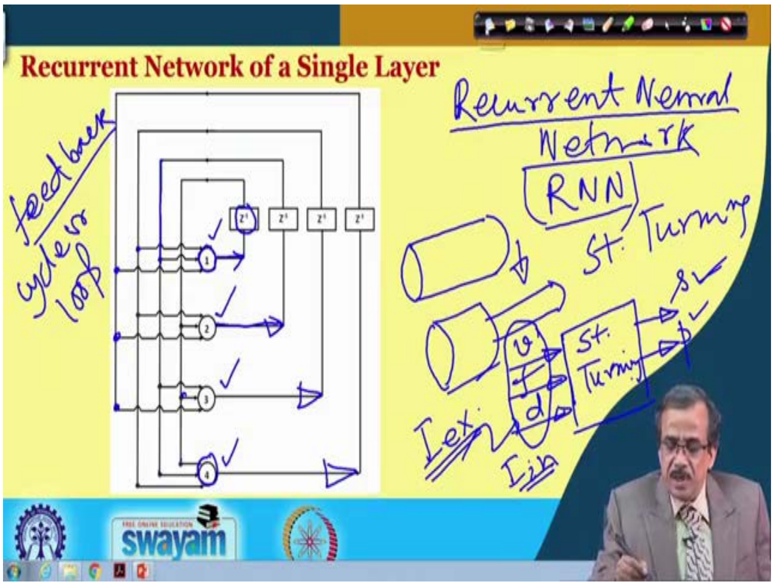
Fuzzy Logic and Neural Networks
Prof. Dilip Kumar Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 27
Some Examples of Neural Networks (Contd.)

(Refer Slide Time: 00:15)



We have discussed the working principle of multi-layered feed-forward networks and radial basis function network. And, we have seen, how to model the input-output relationships of a particular process, but supposing that the process is highly dynamic. Now, for a very dynamic process, these networks may not be able to capture the dynamics of it.

Now, in that case, if you want to make a successful model, we will have to go for another type of network, which is known as the recurrent network. Now, this recurrent network, in short, is known as your RNN. Now, we are going to discuss, in details, the working principle of these particular recurrent neural networks.

Now, before I go for the recurrent neural network, we should try to understand the reason behind going for this type of network in more details. Now, to explain the fact, why should we go for this type of network and let me try to take one very practical example. Now, supposing that I am going to model a process, which is something like straight turning of a cylindrical part. Now, supposing that I have got a cylindrical part something like this, now from this particular cylindrical part, I will have to make one turned cylindrical part and supposing that, I will have to make something like this.

Now, for getting this type of products starting from here, I will have to go for the straight turning. Now, the straight turning is carried out on a lathe. Now, if I just go for this particular straight turning, what I will have to do is, I will have to consider a process which is having three inputs, say cutting speed, feed and depth of cut, say cutting speed is denoted by say v, speed is denoted by f and your depth of cut is a denoted by d.

And, what are the outputs? The outputs are nothing, but the quality of the turned surface and that is measured in terms of say, surface roughness, it is denoted by small s and supposing that I will have to find out what is the power consumption so, that is a denoted by p. Now, let me repeat. So, this is a process, process of say straight turning and this particular straight turning has to be carried out on a lathe and here, there are three inputs like cutting speed, speed and depth of cut and the outputs are the surface roughness and power consumption.

Now, these inputs are selected by the user or the operator and we do the turning and then, as a result, I will be getting some surface roughness and power consumption. Now, the moment we select these input parameters, something else happens inside this particular process. Now, if I call, these input parameters are nothing, but the external input parameters and the moment we select these external input parameters, some of the internal input parameters will be created.

For example, say, there will be some level of vibration, while doing this particular machining. Now, this generated vibration, that is called the internal input, that is I_in. So, this particular internal inputs will have some contributions towards the surface roughness and power consumption.

Now, if I want to model this particular dynamic process in a very efficient way, we will have to consider both the external inputs as well as the internal inputs because both of them are having some contributions towards the output. Now, the network, which have already been discussed like your multi-layered feed-forward network or the radial basis function network cannot be used to capture the whole dynamics of this particular process. Now, if I want to capture the complete dynamics of this particular process, we will have to go for in fact, the Recurrent Neural Networks, that is your RNN.

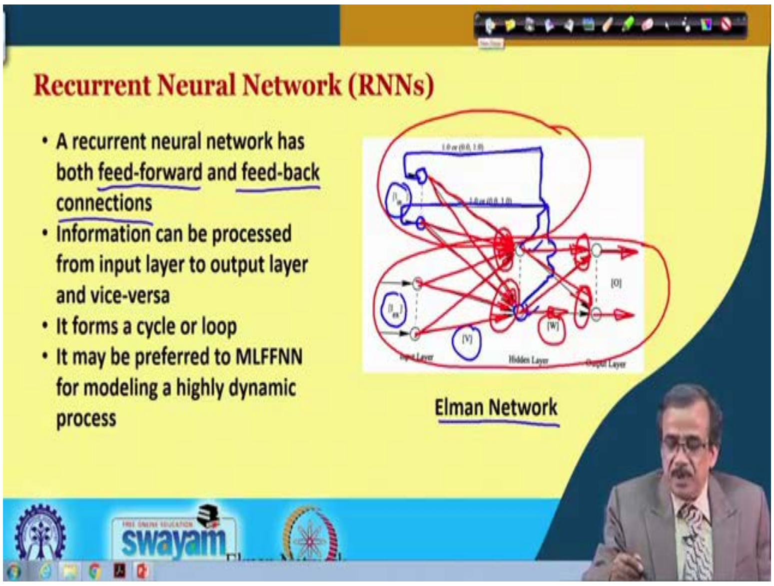
Now, before we discuss the working principle of this particular RNN, let me try to concentrate on how to capture the dynamics of a particular layer of neurons, now let us try to concentrate on this. Now, for simplicity, let me consider that on a layer of a neural network, say I have got only 4 neurons, say 1, 2, 3 and 4. Now, what do you do is, to find out the output of this particular the first neuron, the inputs should come from the second, third and fourth. So, input is coming from this particular neuron, then it is coming from your this particular neuron and it is coming from this. So, the inputs are coming from these three and we will be getting some output here.

Now, this particular output will be fedback. So, this particular symbol z inverse indicates the feedback. So, this will be fedback actually to the 2nd neuron, then comes your 3rd neuron and it will also go to the 4th neuron. Now, let us concentrate on the 2nd neuron. The inputs of the 2nd neurons should come from the 1st one then comes your the 3rd one and the 4th one and supposing that this is the output, we are getting. Now this output will be feed-back and this will come here and this particular output will enter the 1st neuron, then comes, it will enter your the 3rd neuron and it will also enter the 4th neuron.

Now let us concentrate on the 3rd one, now the output which you are getting from the 3rd will be feed-back and that will enter to the 1st neuron and then, it will also enter to the 2nd neuron and this will also go to the 4th one and let us concentrate on the 4th neuron. So, the output of the 4th neuron that will be feed-back and that will enter in fact, the 1st neuron, then it will enter the 2nd neuron and then, it will enter the 3rd neuron. Now, this is the way actually, we try to capture the dynamics of this particular the layer of neural network and this particular layer, for simplicity, we have assumed that it consists of only a few neurons.

Now, to summarize let me mention that in this type of recurrent network, what we do is, we take the help of some sort of feedback. Now, this feed-back is actually going to help us to capture the dynamics of this particular process and here, there will be a cycle or a loop in this type of network. So, this is actually, the way we can capture the dynamics of a single layer of this particular neuron.

(Refer Slide Time: 08:46)



Now, we are going to concentrate on the network; that means, it has got say three layers like input layer, hidden layer and output layer. Now, here if you see the literature for this particular recurrent neural networks, we have got a few very popular models.

For example, say we have got Elman model, then comes we have got the Jordan network and the combined Elman and Jordan network. Now, let be concentrate first on this particular the Elman network, let us see, how does it work. Now, in this type of network, in fact, we have got two circuits; one is your feed-forward and another is called your the feed-back circuit.

So, we have got feed-forward and feed-back circuits. Now, if I concentrate on this particular input layer, the external inputs, that will be passed through the network through this input layer. So, this is the input layer, now we have got the connecting weights between the input layer and the hidden layer and that is denoted by v and on the hidden layer, these neurons are having some transfer functions and depending on this particular transfer function, I will be getting some output here; I will be getting some output here.

Now, here actually, what we do is. So, these outputs of the hidden neurons or the hidden layers are not directly passed to the output, instead those outputs of the hidden neurons are taken as the feedbacks. So, the output here, for this particular hidden neuron is actually taken back, as your the feed-back and what they do is, the output which are getting, either we consider the 100 percent of that or say might be said 50 percent or 30 percent or 40 percent of that as feed-back and we just keep it, here.

Now, similarly, the output which you are getting on these particular hidden neurons, we take as actually the feed-back and once again, we just copy and we put it here. Now, these are actually the feedbacks, which are nothing, but the internal inputs of this particular process denoted by I_im. So, here, we pass the external inputs and these internal inputs will be generated inside this particular process and now, actually what we do is, we try to pass actually all such inputs once again to the network and what you do is, this particular the feedback, that will be allowed to come here and this particular feed-back will be coming here and of course, we are having this particular circuits also and here, we will be getting actually the combined input for this hidden neuron.

Now, similarly, here also I will be getting. So, this particular feed-back is coming from here, this particular feed-back is coming from here and of course, this will come, this will also come. So, here, I will be getting this combined input further on the hidden neuron. And, these particular combined inputs will be passed through the hidden neurons and then, I will be getting these particular outputs, here I will be getting this particular output.

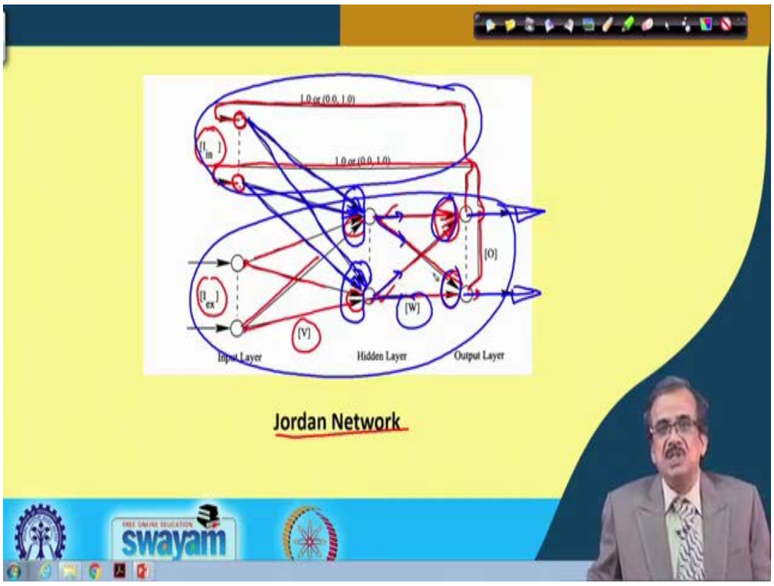
So, whatever output we get here, we multiply by the connecting weights, that is denoted by your w and here, we will be getting the combined input for the output layer and the output layer neurons are having some transfer functions. So, I will be getting this particular final output, that completes actually, one iteration of this particular network.

Now, here, to summarize actually what we do is, we take the feed-back from this particular hidden layer. And, either we consider 100 percent of this output of the hidden layer or slightly less than 100 percent as feed-back and this feed-back will be actually considered as input to the hidden layer once again and then, it is passed and ultimately, we will be getting this particular output.

Now, if you see, this particular network consists of, in fact, two such components, one is called the feed-forward component. So, this is nothing but the feed-forward component and this particular component is nothing but the feed-back component. So, this network has got both the components and it will be able to capture the dynamics and this shows the working principle of this Elman network.

Now, as I have already mentioned several times that it will be preferred to the multi layered feed-forward network, if we want to model a highly dynamic process; so, this is the working principle.

(Refer Slide Time: 14:24)



And, now, I am just going for another recurrent network, that is called the Jordan network. Now, in Jordan network, actually what we do is, we take the feed-back from the output layer, but we do not take the feed-back from your the hidden layer now once again let me start. So, I am passing one set of external inputs denoted by say I_ex and here, we have got the connecting weights. So, whatever outputs we are getting here, those things will be multiplied by these connecting weights, these are the connecting weights v and those things will be summed up here. So, I will be getting some input; I will be getting some input here.

And, initially we assume that this particular feed-back circuit is not present and now, I will be getting the inputs and we will pass it through the neurons of this hidden layer. So, depending on this particular transfer function, I will be getting some outputs here and similarly, here also, for this particular hidden neuron I will be getting some outputs. Now, these outputs will be multiplied by the connecting weights, and these things will be summed up and this will be considered as input of the output layer. Similarly, this will be multiplied by this connecting weight and these things are summed up and these are nothing, but the input for the output neuron.

And depending on the transfer function so, we will be getting some your output here and we will be getting some output here, but these outputs are not actually the final output. Now what we do is instead of going for here. So, these output actually we feed-back and this will be used as your some sort of feed-back to this particular the network and similarly whatever output we are getting here.

So, these particular things will be fed-back and it will be used as the internal inputs. Now, here we will be getting some internal inputs, now those internal inputs will be passed through this particular network and that means, it will be getting some feedbacks.

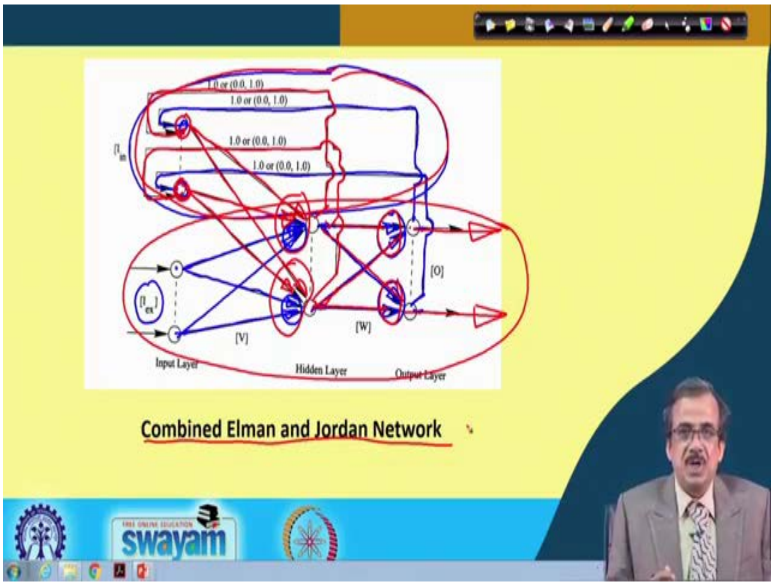
So, this feed-back will come here. So, this particular feed-back will go and I will be getting the feed-back here; I will be getting the feed-back here. Now, we are in a position to determine what should be the combined input for these particular the hidden neurons and we will be able to find out, what should be the combined input for this particular hidden neuron.

Now, depending on the transfer function, I will be getting the output; and once again, we use these particular connecting weights and then, we will be getting the combined input for this particular output neuron and then, using the transfer function, I will be getting the final output; I will be getting the final output here. So, that completes actually one cycle or one loop or one iteration of this particular network.

Now, to summarize, in Jordan network, we take the feed-back from the output layer and as I have already mentioned, this part is nothing but the feed-back circuit and this is the feed-forward circuit and in this type of RNN, we consider both the feed-forward circuit as well the feed-back circuit. And, ultimately for a set of external inputs, I will be getting the final output and of course, the system is going to generate this type of your internal inputs.

Now, for the purpose of calculating the outputs, feed-back and all such things, exactly the same procedure, which we have discussed for the multi-layered feed-forward network (like how to find out the inputs for a particular layer, how to find that output and all such things exactly) the same principle, we can follow.

(Refer Slide Time: 18:56)



Now, actually, we are going to discuss another network, which is the combination of this particular Elman network and Jordan network. Now, here in the combined Elman and Jordan network, we take the feed-back both from the hidden layer as well as the output layer and let us see, how does it work? Now, let me assume that initially, this particular feed-back is not there. So, let me assume that this particular feed-back is not there and let me concentrate on the feed-forward circuit first.

Now, I am passing the set of the inputs that is nothing, but your external inputs exactly the same way I discussed. So, here, on the input layer, we have got the neurons, they are having their transfer functions and based on that you will be getting some output here. And, once you have got this particular output, now these outputs, will be multiplied with a connecting weights. So, I am here, this will be multiplied with a connecting weight.

So, I am here, similarly this will be multiplied by the connecting weight, this will be multiplied by the connecting weight. So, I will be getting some input for this hidden layer I will be getting some input for this particular hidden layer. Now, I will pass these inputs to the hidden layer. So, I will be getting some outputs here and now, what we do is. So, this particular output has got two applications; one is, it will be allowed to pass to the output layer and this particular output will also be allowed to enter; this particular feed-back circuit.

Now, let us see like what happens if it goes to the feed-forward circuit. So, this output will be multiplied by the connecting weight, this output will be multiplied by connecting weight. So, I will be getting some combined input here. So, this particular output will be multiplied by your the connecting weights.

So, this will be multiplied with a connecting weight and those are summed up and these inputs are going to pass through the output neurons and here, we have got the transfer function. So, I will be getting some output here. Now, what we do is, this particular output we consider either 100 percent or less than 100 percent of that as feedback. So, this will be used as feedback. So, this is the feed-back and it will be stored here.

Now, similarly, this particular output will be considered as feed-back and it will be stored here and let us concentrate on the output of the hidden layer. Now, if you concentrate on the output of the hidden layer, you can see that we have got the output of the hidden layer, which is nothing, but this. So, this will be used as feedback. So, I am just going to use feed-back and it will be stored here. Similarly, the output of these particular hidden neuron. So, this will be allowed to pass through like this and we will be able to collect here in this particular neuron.

So, this is the way actually, we collect the combined feedbacks both from this hidden layer as well as the output layer and here, we will be getting that particular internal inputs. Now, these internal inputs will be allowed to pass through these particular hidden layer. So, I am just going to pass it here, I am just going to pass it here and similarly, this will be allowed to enter, this particular hidden neuron and this will be allowed to enter this particular hidden neuron.

So, what happens? Here, I will be getting the combined input for the hidden layer, I will be getting the combined input for this hidden layer and these combined inputs will be passed to the transfer function here. So, I will be getting the output here.

Now, the output, which you are getting, that will be multiplied by the connecting weight; multiplied with a connecting weight, this output will be multiplied with a connecting weight and so, these things, we are going to collect here. So, these inputs for the output neuron, we are going to collect and these inputs will pass through the transfer function and ultimately, we will be getting the final output of these particular the network. And, this completes actually one cycle of this particular combined Elman and Jordan network.

Now, this is the way actually, we capture the dynamics of this particular process, as I mentioned. So, this is the feed-back circuit and this is the feed-forward circuit and both the things will be working together and we will be getting this combined Elman and Jordan network. Now, if you see the performance; the performance of this particular network is found to be very reliable, like if you want to model the dynamics of a highly complex or highly dynamic process. Now, as it is having this feed-back circuit, there is almost a guarantee that it will be able to capture the dynamics of this highly dynamic process.

Now, if you see the computational complexity of these networks, here, as it is having both feed-back and feed-forward circuits, compared to your multi-layered feed-forward network, there will be more computations.

So, computationally, it could be a little bit more complex and moreover, as we have discussed the Elman network, Jordan network and the combined Elman and Jordan network, if you compared their computational complexity. The computational complexity of the combined Elman and Jordan network is obviously, becoming more compared to that of only Elman network and only Jordan network, but supposing that we have got a very complex process, very dynamic process. So, its better to go for this type of combined Elman and Jordan network.

So, the working principle of the recurrent network, we have discussed and principle-wise, it is very simple and it has been reported that this network can perform very well particularly to capture the dynamics of the highly dynamic process.

Thank you.