Business Analytics & Data Mining Modeling Using R

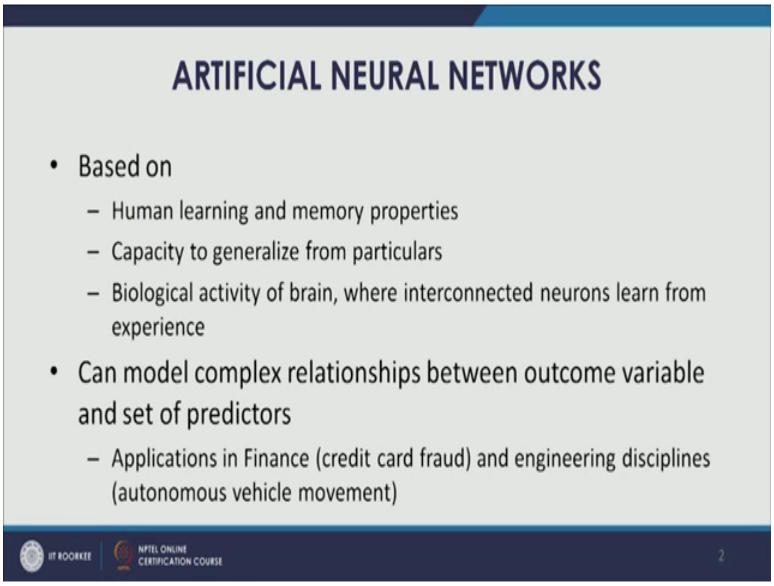
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology, Roorkee

Lecture - 53
Artificial Neural Network-Part I

Welcome to the course Business Analytics and Data Mining Modeling using R. So, in this particular lecture we are going to start our discussion on Artificial Neural Networks. So, let us get this started.

So, let us understand a particular background about background of artificial neural networks how they were conceived, and other things other details. So, first artificial neural networks they are based on human learning and memory properties.

(Refer Slide Time: 00:50)



So, they mimic the process of memorizing and learning of humans, so that is mimicked by these models artificial neural networks model. So, capacity is to generalize from particulars so that is also the in the one capacity of humans, third one biological activity of a brain where interconnected neurons learned from experience.

So, the structure the biological activity and the particular structure that is part of the brain the similar kind of a structure is used in artificial neural network, and the learning a procedure the learning process is also quite similar so directly inspired from human brain activity. So, these particular model neural networks they mimic our human brains learning and learning process and other things.

So, learning and memory properties ability to generalized from a specific things and also the overall network of neurons that is there. So, all these things are quite similar in artificial neural networks. So, typically neural networks model they are used where the relationship between outcome variable, and the set of predictors is quite complex.

So, because of that the applications in finance and some engineer application few examples are give. For example, credit card fraud, fraud as a finance application to detect whether a particular whether there is some fraud in a particular gate card account and then engineering discipline. One example is autonomous vehicle movement, so that whether movement of whether the steering has to be moved left or right, so those kind of things can also be predicted or classified using artificial neural network.

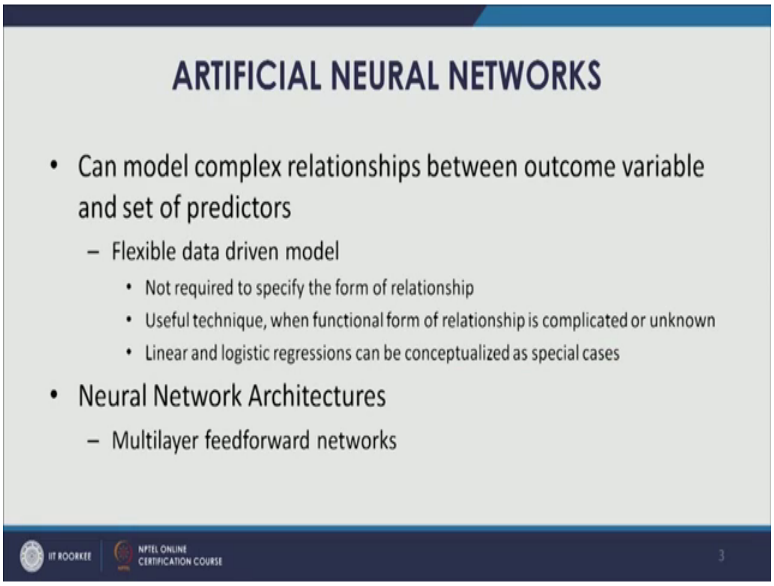
So, typically here we are dealing with complex relationships something for which it is difficult to difficult to use a structured or functional form or actual relationship is unknown, probably artificial neural network can be used. And applications also applications of artificial neural networks also are seen in areas where the main where the main behaviour, or main phenomena or relationship is being determined is being based on how the human things are and therefore, behave.

So, for example, driving so one example that we have given here driving. So, driving is essentially how a person how a driver thinks about moving their car in a congested in traffic environment. How you know people plan people plan and execute credit car, related frauds so that is also how human brain thinks and plan things.

So, therefore, those things are difficult to put it in a functional form; however, they can be modelled using neural network, because neural network also mimic the properties of human brain. So, as discussed they can model complex relationship, so they are flexible data driven model.

So, we are not required to specify the form of relationship so like in linear regression we assume that the relationship is going to be linear the relationship between outcome variable, and set of predictors that is going to be linear so that is the exemption. And logistic regression we assume that you know we use that logistic function as the form of relationship.

(Refer Slide Time: 04:58)



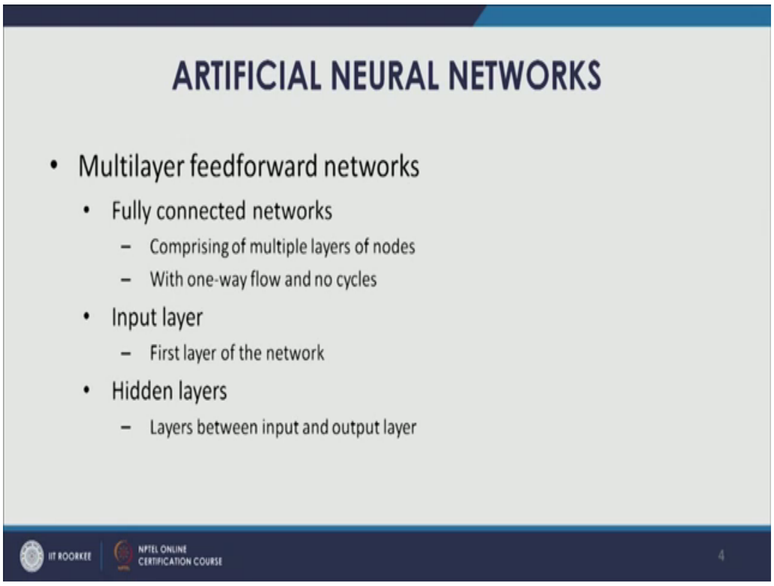
And in this particular case neural network we are not required to specify any such form linear or logistic or other forms; So, a useful technique especially useful technique when functional form of relationship is complicated or unknown.

So, if we do not know the actual relationship or it is quite complicated, so in such situation artificial neural networks are quite useful as I have talked about any you know specific areas where you know human activity the way human things becomes the core part of that big activity for example, driving or you know some of the these frauds the financial frauds that happen. So, there the applications of neural networks have been quite useful.

Another important aspect of neural network is that a linear and logistic regression these can be conceptualized as a special cases of neural network which we will see that later in the lecture. Now, let us move to next point that is neural network architectures.

So, there are various neural network architectures that are being used; however, the most popular one that is typically used in data mining is multi layer feed forward networks. So, this is the architecture that we would be using for our discussions, and exercises modelling exercises. So, multi layer feed forward network this is the neural network architecture that is used in data mining.

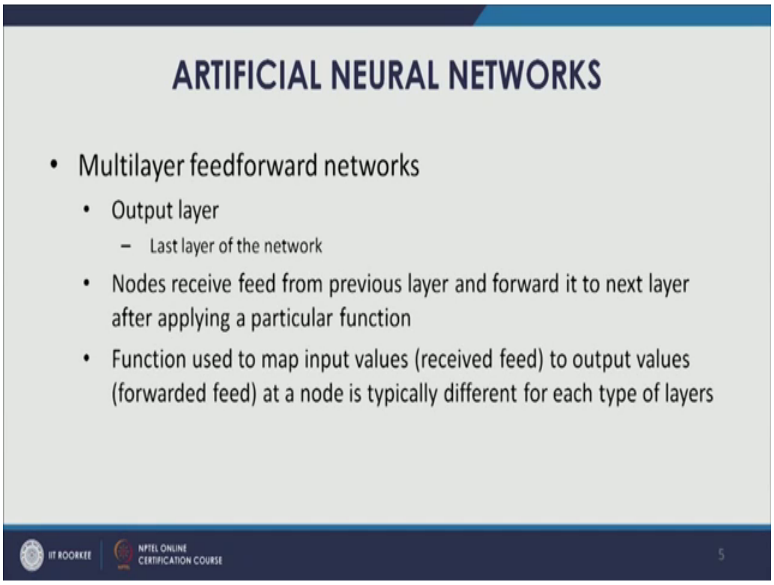
(Refer Slide Time: 06:44)



So, let us understand the multi layer the architecture of multi layer feed forward networks, so they are fully connected networks, so comprising of multiple layers of nodes. So, there are multiple layers of nodes and there is a one way flow and no cycles we will see this through a diagram.

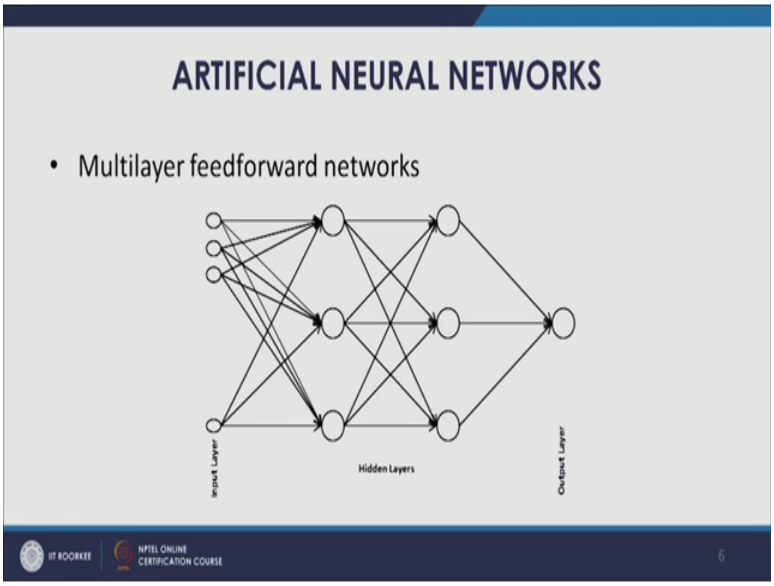
Then we have typically we have 3 types of layer in multi layer feed forward networks, first one is first layer is input layer, then we have a series of hidden layers and finally, we have output layers.

(Refer Slide Time: 07:21)



So, a typical neuron network diagram is something like this, so, you can see first we have input layer. So, all these circles they are actually nodes this whole diagram is called a network a specifically neural network, and these circles they represent nodes and we have 3 types of layer.

(Refer Slide Time: 07:24)



As discuss input layers so we can see some you know nodes here, and these nodes are connected to you know next layer of node which are called hidden layer, so we might have one or more hidden layers. So, input layer nodes are going to be connected to you know to to these hidden layer nodes in one directional sense and then these hidden layer nodes would finally, the last hidden layer nodes would finally, be connected to nodes of output layer right.

So, this you can see one node is there in this particular output layer, so this is a typical neural network diagram, and specifically for this architecture multi layer feed forward network. So, let us go get back to our discussion. So, as I said in the diagram that we saw typical neural network diagrams we saw that all the nodes were fully connected.

So, comprising of multiple layers of nodes so there were multiple layers input layer then series of hidden layers and output layer and we saw that there was one way flow, and there were no cycles, so there were no feedback loops, there were no cycles.

So input layer that was the first layer of the network. So, typically the predictors the variables that are part of the modelling exercise they corresponding, so corresponding to each of those predators typically we have a node in input layer. So, we have p predictors, typically we p nodes in input layer and then we have hidden layers. So, there can be one there can be there can be any number of hidden layer, hidden layer nodes; however, typically single hidden layer node is sufficient to model even the highly complex situations, highly complex relationships.

So, these hidden layers are between input and output layer and finally, we have the output layer which is the last layer of the network and we have nodes. So, corresponding to the output variable that is outcome variable of interest. So, depending on the outcome variable we might have you know nodes there.

So, one outcome very typically we have typically we have one node there, if the outcome variable is categorical in nature. So, typically you know for binary you know variable binary variable we might still have just one node, but if the categorical variable is having m classes, then we might have m nodes. So, output layer typically of a prediction there might be a single node for binary categorical they might be just single node and for a categorical variable m classes there might be m nodes so that is going to be the last layer.

And so this is about the structure or architecture of multi layer feed forward networks. Let us understand a few more details about these networks, so nodes as we saw in the diagram nodes receive feed from previous layer and forward it to the next layer after applying a particular function.

So, let us go back to this diagram so we can see that for example, first layer is input layer. So, we do not have any previous layer in this case. So, we just receive the input values and then they are fed to a next layer. Now when we talk about the first hidden layer you would see that they receive the feeds from the previous layer nodes.

So, you can see nodes receive feed from previous layer and forward it to next layer after applying a particular function so these nodes first hidden layers, so these nodes. So, they receive feeds these connected arrows. So, you know from every node in input layer every node in input layer is going to be connected to every node in the next layer that is the first hidden layer.

So, you can see 3 arrows emanating from this particular node and they connecting to all 3 nodes of first hidden layer then the second input node and again you can see 3 arrows connecting to 3 hidden layer nodes, and similarly for the next node and similarly for the last input node.

So, similarly if we look at other layers so first hidden layer, and second hidden layer as you can see from first hidden layer nodes you can see 3 arrows connecting to 3 nodes in the next hidden layer that a second hidden and second and last hidden layer. And we can see similarly from all the nodes in the last hidden layer we can see arrows connecting to a output node in the just single output node in the output layer.

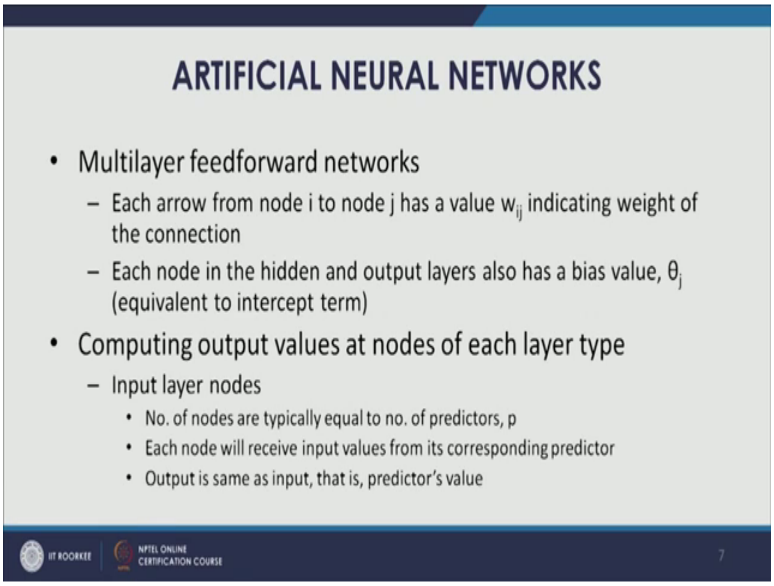
So, for example, first hidden layer we can see that it is receiving feed from the previous layer nodes that is input layer nodes, and then it will apply certain function you know that certain function on that on that feed. And then the output is going to be forwarded to next layer node and for next layer node it will become the input.

And then again next layer node will apply a certain function to the feed that they are receiving from previous layer, and then again the output would be computed and this output would be forwarded to the next layer that is in this case output layer.

So, in this fashion the feed is computed the input values so they are used and the output is produced. So, you can see next point is function used to map input values, that is received feed to output values that is the forwarded feed at a node is typically different for each type of layers, to the function that are going to be used that is going to be used in input layer, you know you know that is going to be applied on input values that is predictors values.

So, this particular function is going to be different from hidden layer and output layer a hidden layer the function that is going to be applied on the received feed is going to be different from the other two types of layers. Each type of layer will have its own function. So, this function typically is called transfer function as we will see later on.

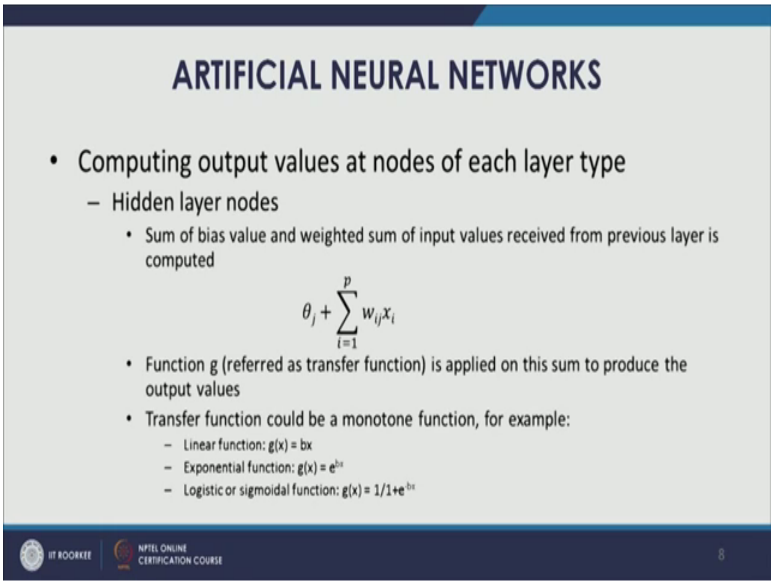
(Refer Slide Time: 14:16)



So, a few more details about the these networks. So, each row from node i to node j has a value w ij that is the weight indicating weight of the connection so in this particular neural network. So, if this node is connected to this particular node if this is the node i and this is node j.

Then the arrow this is connecting these two nodes this will represent that this will have a weight this will have a weight called w ij, and it will represent the weight of this that particular connection is strength of that particular connection. Each node in the hidden and output layer layers also has a bias value that is theta j equivalent to intercept term.

(Refer Slide Time: 15:00)



So, we will see that these nodes hidden layer nodes first hidden layer, second hidden layer and then the output layer. So, these particular nodes they will also have a bias value right, so all these nodes will also have a bias value and which is going to be equivalent.

So, you can consider the scenario in this fashion like these connected arrows they are they are bringing the feed to this particular node and the output level of the output that is going to be produced that is being controlled by these biased values which is equivalent to what we have intercept term in linear regression. So, in this fashion these values are controlled, and the feed is forwarded to the next layer.

So, next important aspect is computing output values at nodes of each layer type. So, how do these values are computed at you know each layer and nodes at each layer each type of layer. So, let us first discuss the input layer nodes so typically as I talked about number of nodes in input layer are typically equal to the number of predictors.

So, if they are p predictors that we are using in our modelling exercise then we will have to in a typically we will have p number of nodes in input layer corresponding to each of the predictors. Because all the predictors values would be fed to the input layer nodes and then the output would be produced that would feed into the first hidden layer nodes.

Now, next point is each node will receive input values from its corresponding predictor as I talked about now output is same as input that is predictors value. So, typically the function that we talked about the transfer function that you know it is called transfer function, so that we talk about typically it is the identity function or linear function in that is used an input layer; that means, whatever input is received the same input is transferred same input is fed to the next layer.

So, the function is linear so there is you know no effect is no factor is applied on the input values. So, now let us discuss the next type of a layer that is hidden layer hidden layer nodes, and how values are computed. So, hidden layer nodes what we do is we take some of bias value and weighted some of input values received from previous layer so that is computed.

So, you can see here theta j so that is the equivalent of intercept term. So, this is the bias value and then we add weighted sum of input values that we received from previous layer. So, let us go back to the diagram and let us try and understand this thing again, so if we look at this particular node this node in the first hidden layer. So, this particular node is receiving is receiving values from these nodes input layer node.

So, if there are p nodes so it will receive p values now each of these connections each of these arrows will have their own weights. Now, these weights would be multiplied with the values that is being received from these input nodes.

So, each of these input nodes the values of that is being received from each of these nodes would be multiplied by the weights. And then that would be summed up summation would be performed and then the intercept term that is biased value would also be added to this sum, so that is what you see here in this particular expression.

So, you see that x i is corresponding to you can see that this summation is starts from i is equal to 1 to p so; that means, for each predictor we will have a value and so that value would be fed to the node that node that we just saw when a first hidden layer and node and for that matter any other nodes in the hidden layers or output layer.

So, so this value then is the multiplied by the weight of the connecting arrow, and then we take summation and then bias value is also added to this. So, this bias value will as you can see in a way is controlling the level of the overall value of hidden layer nodes just like it.

So, this some similarity as we can see to the linear equations linear line that we you know linear modelling that we see beta 0, so theta theta j here is equivalent of beta 0 and then these weights multiplied to greatest values are equivalent of, but we have you know beta 1 x 1 and beta 2 x 2 that in linear regression. So, this formulation is equivalent to that.

So, function g as we have been talking about that each layer will have its own function that is going to be the applied on the input values that are received at a particular node. And using that function will get the output values that is going to be forwarded to next layer.

So, this function g is called referred as transfer function and is applied on this sum, so for hidden layer nodes we have to compute this term. So, all the predictors values that we are receiving from p nodes they will be multiplied by their respective weights and then bias value would be added then this will have the value from you know using this expression and this value would be then passed to this transfer function and we will get the output, so this output is going to be the output of that hidden layer node.

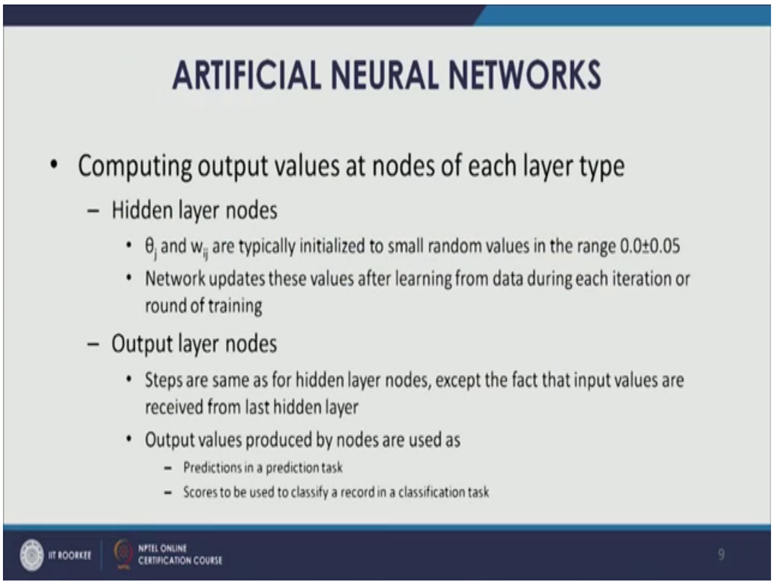
So, what could be the different alternatives different options for our transfer function, so transfer function could be a monotone function and these are few examples for example, linear function g x is equal to v x. So, this is the function that is used in input layer where the input is transferred as is as output re right.

So, this is the function that is used in input layer and another function that could be used is exponential function. So, e to the power bx and then another function that is that would be used is logistic or 6 model function that is 1 divided by 1 plus e to the power minus v x, so this could also be used.

So, a logistic function is the typically the most you know used transfer function in multi layer feed forward networks, in neural networks in general. So, logistic regression is typically used for hidden layer nodes, and as we will discuss even for output layer nodes also this logistic function is typically used most use function.

Now, these hidden layer nodes few more points these biased values and weights theta j and w ij's they are typically initialized to small random values in this range 0 to 0.05 0 to plus minus 0.05.

(Refer Slide Time: 22:40)



So, typically for the first iteration as we will discuss the; you know training process neural network training process. So, typically these values are initialized in this range, and then the network will learn through different iterations. So, as you can see the next point network updates these values.

So, first they would be randomly initialized with these small values, and in next iterations the network will update these values after learning from data during each iteration or round off training. So, these values theta j's theta theta j's which are biased values which are going to be you know each for each hidden layer or and output layer nodes and then weights which are going to be there for each of the connection from input layer nodes to output hidden layer nodes to output layer node.

So, each of those connection will have these weights they would be in initialised to some values for first iteration and after that these particular values theta j's and w ij's they would be continuously updated after each iteration, after each pass through the network. So, for each observation in the training partition all the or the sample that is these weights, these values are going to be updated. So, let us move forward.

So, now let us talk about the next type of layer that is output layer, so output layer nodes as you can see in the first point is steps are quite similar same as we just discussed for hidden layer nodes except that the fact that input values are received from last hidden layer right.

So, when we talked about hidden layers to the previous layer whether it is the hidden layer or input layer the values were received. And then those fields then they were using that expression those that value was computed weighted value weighted value was computed and that was then used in the function transfer function. A similar process is used in output for output layer nodes as well.

However, the values that we receive they come from the last hidden layer and using the same expression that we used for hidden layer nodes, the weighted value is going to be computed and then that is going to be used in transfer function and the output value is going to be produced.

So, typically the transfer function that is used in the hidden layer nodes the same transfer function is used in output layer nodes as well. So, typically the same function same transfer function is used. So, as I talked about that most used function most used transfer function is logistic function, but that is mainly for out hidden layer node hidden layer nodes and output layer nodes. So, typically logistic function is used for hidden layer nodes, and output layer nodes. And linear or identity function used for input layer nodes.

Now, output values that are produced by output layer nodes they are used as predictions, because that is the final layer last layer that we have in a network. So, the output that is produced out of these nodes output layer nodes, so they are going to be used as a predictions and in a in a in a prediction task. And also they are going to be used as this course for example probabilities values and that are going to be used to classify record in a classification task. So, this is how the computations happen computations are performed in a typical neural network architecture.

Now, if we talk about the neural network training process so a steps that we have discussed for computation for whether it was for input layer, or hidden layer, or for the output layer. So, these steps are going to be repeated for all the records in the training partition. So, for each so each whenever I know for each observation when these particular steps are repeated that is called one iteration.

So, each observation will go through this neural network and all those computations in input layer, and then hidden layer and then the output layer are going to be performed and for each record this process will continue. Now, for each observation therefore, will have a predicted value and therefore, we will have you know predicted error.

Now, these errors are later on going later on are going to be used to learn for example, each error as we will discuss later that in coming lecture that these errors for each observation these errors are then used to update the neural network to update the weights w ij's and theta j's. And from each observation that is you know each observation and the computations that are performed in the neural network that duration that happens are going to be used as a part of learning process.

So, more about this process we will discuss through an example in a next lecture.

Thank you.