Machine Learning for Soil and Crop Management
Professor Somsubhra Chakraborty
Agricultural and Food Engineering Department
Indian Institute of Technology Kharagpur
Lecture 32
ML and DL for Soil and Crop Image Processing

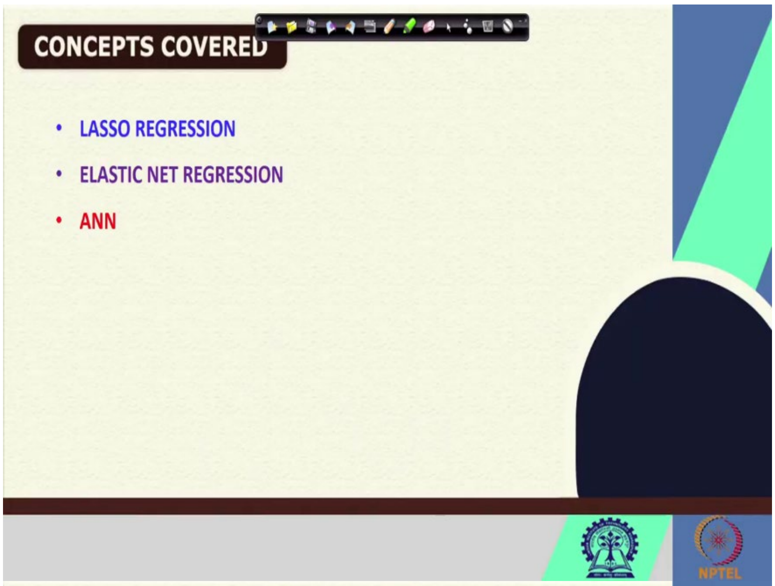
Welcome friends to this second lecture of Week 7 of NPTEL Online Certification Course of Machine Learning for Soil and Crop Management. And in this week, we are talking about machine learning and deep learning for soil and crop image processing. So, now we are going to discuss our second lecture of this week.

And in the first lecture, we have already discussed about different types of calibration and validation method. We have discussed about random holdout and then independent data subset, independent data for unbiased validation, and we have also discussed random holdout then k-fold cross validation, leave-one-out cross validation, we have discussed in details. We have seen their advantages and disadvantages also.

So, now, and also we have seen what is the cost function of a ordinary least square regression. We have also discussed the problem with the overfitting, we have discussed the bias variant trade-off and also, we have started discussing about ridge regression. And while discussing about the ridge regression, we have, I have pointed out that this ridge regression basically improves the overall model performance by reducing the model complexity.

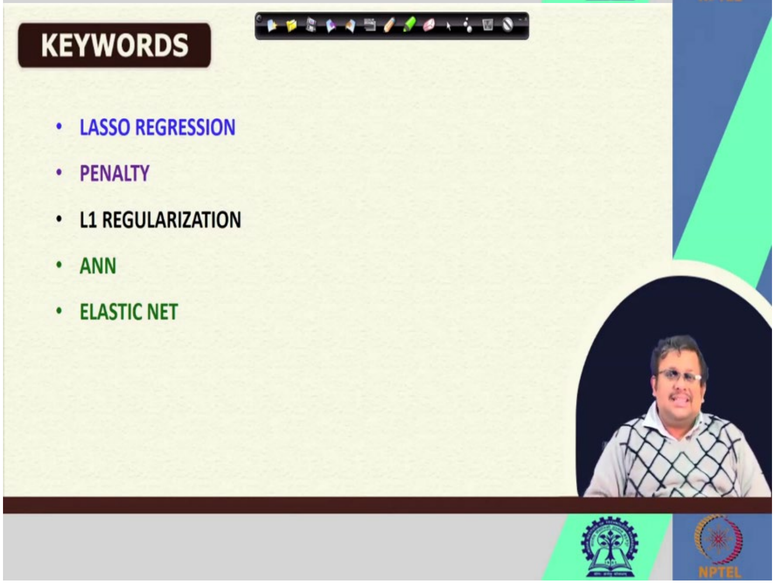
How it reduces the model complexity? By increasing by imposing a penalty parameter by imposing a penalty term, so that the slope becomes lower, and as a result of that, the model becomes less sensitive to the validation data set. And overall performance of the model improves. So, this is the ridge regression. And remember that in the ridge regression this penalty term depends on the square of this beta coefficient.

(Refer Slide Time: 02:29)



CONCEPTS COVERED

- LASSO REGRESSION
- ELASTIC NET REGRESSION
- ANN



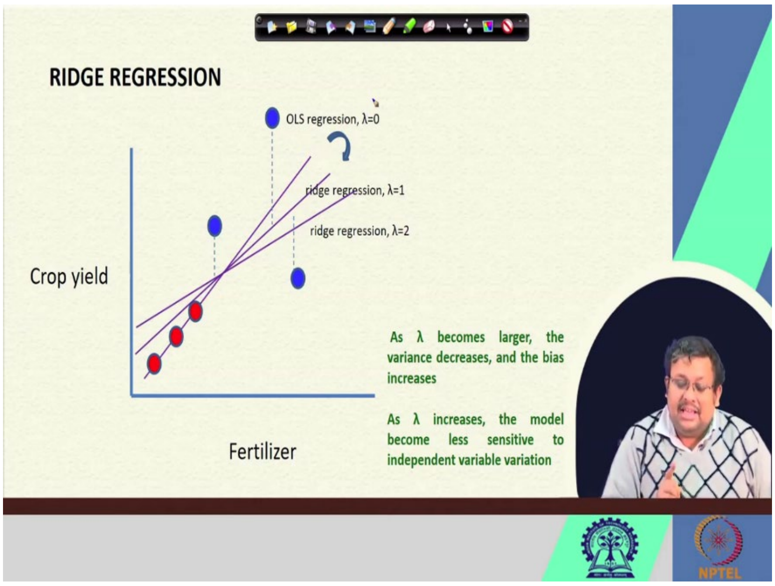
KEYWORDS

- LASSO REGRESSION
- PENALTY
- L1 REGULARIZATION
- ANN
- ELASTIC NET

Now, we today, we are going to finish this ridge regression first and then we will be talking about LASSO regression. Then elastic net regression, and also we will be discussing if time permits artificial neural network.

So, the keywords which we are going to discuss today are LASSO regression, then penalty, then L1 regularization, and then artificial neural network and elastic net. So, we have already discussed L2 regularization. And in case of LASSO we will be discussing L1 regularization.

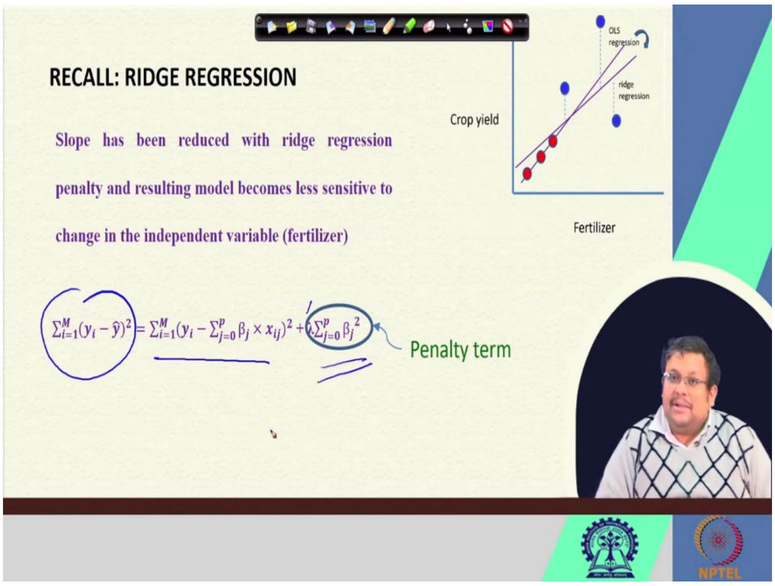
(Refer Slide Time: 03:00)



RIDGE REGRESSION

As λ becomes larger, the variance decreases, and the bias increases

As λ increases, the model become less sensitive to independent variable variation

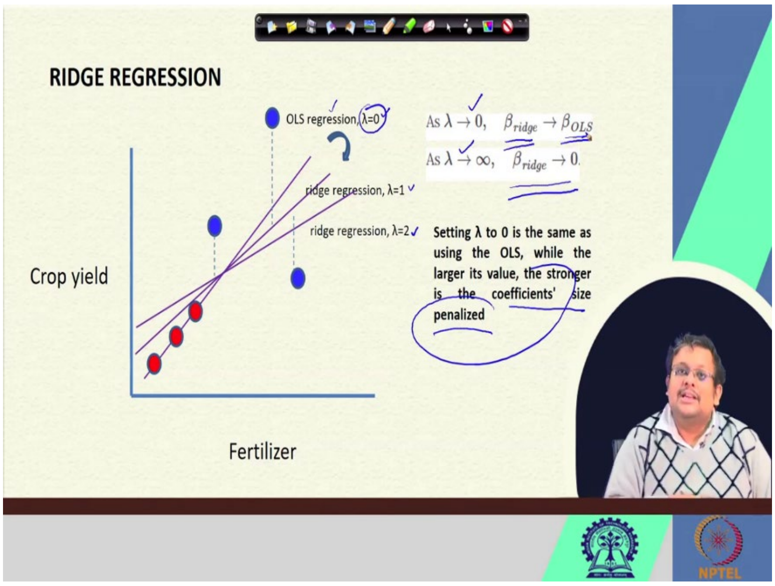


RECALL: RIDGE REGRESSION

Slope has been reduced with ridge regression penalty and resulting model becomes less sensitive to change in the independent variable (fertilizer)

$$\sum_{i=1}^{M}(y_i - \hat{y})^2 = \sum_{i=1}^{M}(y_i - \sum_{j=0}^{p}\beta_j \times x_{ij})^2 + \lambda\sum_{j=0}^{p}\beta_j^{\,2}$$

Penalty term



RIDGE REGRESSION

As $\lambda \to 0, \quad \beta_{ridge} \to \beta_{OLS}$

As $\lambda \to \infty, \quad \beta_{ridge} \to 0.$

Setting λ to 0 is the same as using the OLS, while the larger its value, the stronger is the coefficients' size penalized

So, recall that from our previous lecture that ridge in the region regression, we basically improve the model performance by rotating the line by increasing the value of lambda. In other words, when we are increasing the value of lambda that reduces the slope and when it reduces the slope, that means, it becomes a model becomes less sensitive to the change of this independent variable.

So, in case of ridge regression, this is the cost function, model cost function, which is similar to that ordinary least squares, but at the same time, we have these added penalty term, which is basically depends on the square of the coefficient value. Now, how it happens generally, to give you more pictorial view you can see here, this is the original OLS curve and when we move, when you rotate these OLS curve. So, remember one important thing.

In case of OLS regression, the same equation, which I have showed you in our previous slide holds good when the lambda is 0. So, of course, when the lambda is 0 the whole terms become 0. So, that means, the when the lambda is 0, the ridge regression assumes the ordinary least squares regression. But here you can see in case of ordinary least squares regression the lambda value is 0.

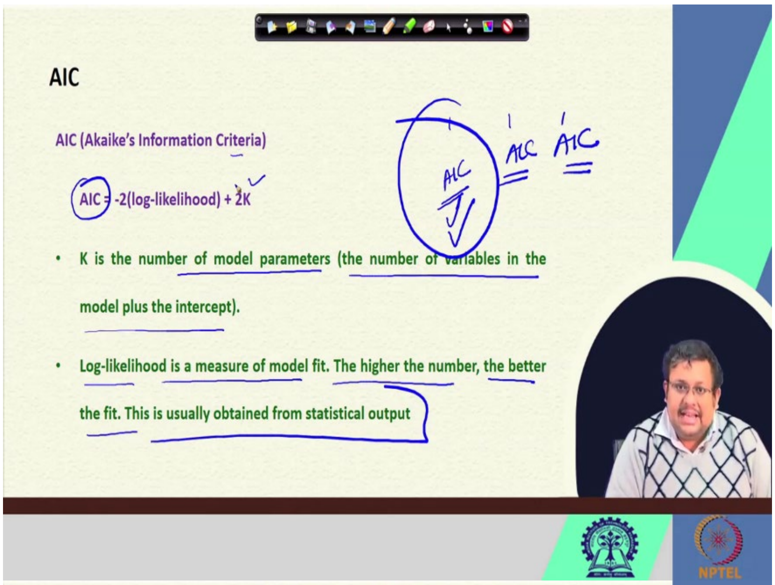
But in the next instance, we are applying the ridge regression and we are rotating that model. So, we are getting ridge regression where lambda equal to 1 another instance we are getting ridge regression where lambda equal to 2. Remember one thing, when we are increasing the lambda our ridge or beta ridge will be or the coefficient will go down to 0 that means, we are shrinking the coefficient. And in this way, we can help identify the and also remove the multicollinearity effect or the overfitting problem.

And remember that when lambda is equal to 0 here you can see lambda is equal to 0, then the slope of the ridge regression is equal to slope of the ordinary least squares regression. In other case, where lambda A equal to infinity that means, infinity, when it is quite high, then we can see the beta or the coefficient of ridge coefficient will be equal to 0.

So, setting this lambda to 0 is the same as using the ordinary least squares, while the larger its value the stronger is the coefficient size penalized. So, basically, by adding by increasing the value of this lambda parameter we are penalizing the coefficient size, we are penalizing the coefficient size as you can see, the coefficients are changing.

And as a result of that, we are overall improving the model performance by inducing some of the bias. So, this is called ridge regression. Remember, I showed you a couple of application of the ridge regression in our previous week. So, this is how this rigid regression works.

(Refer Slide Time: 06:34)

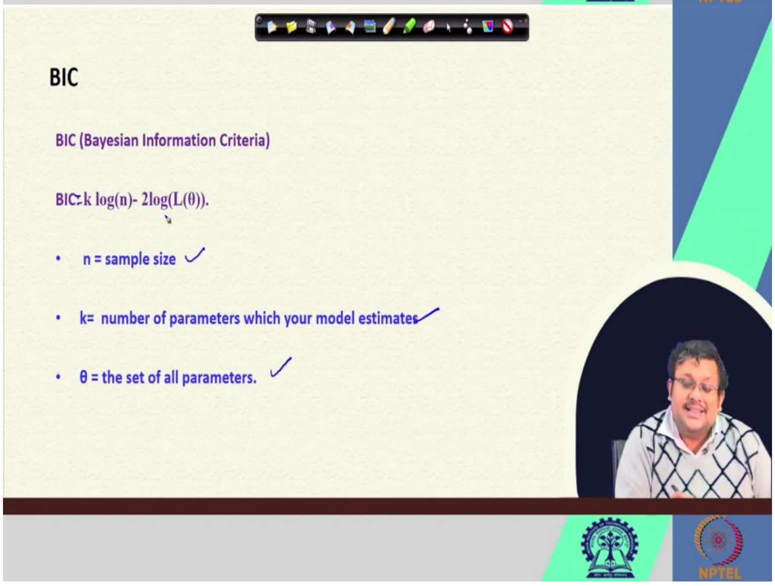


AIC

AIC (Akaike's Information Criteria)

$\text{AIC} = -2(\text{log-likelihood}) + 2K$

- K is the number of model parameters (the number of variables in the model plus the intercept).
- Log-likelihood is a measure of model fit. The higher the number, the better the fit. This is usually obtained from statistical output

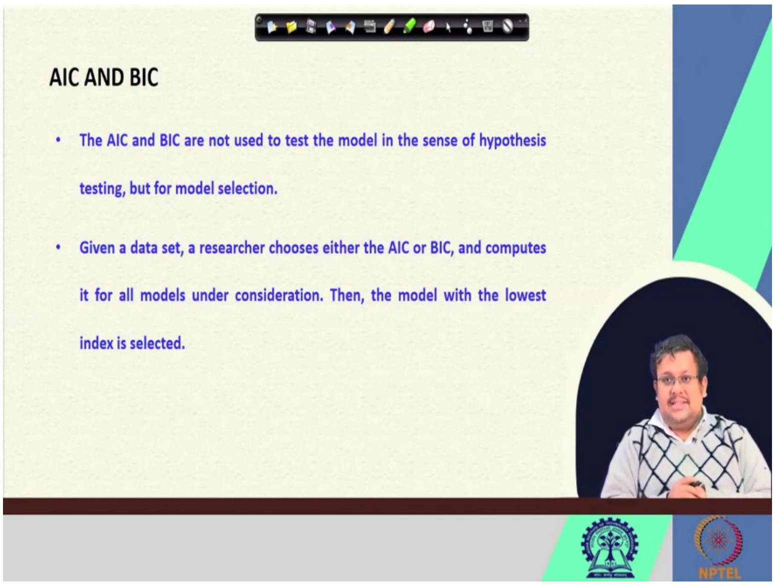


BIC

BIC (Bayesian Information Criteria)

$\text{BIC} = k \log(n) - 2\log(L(\theta))$.

- n = sample size
- k= number of parameters which your model estimates
- θ = the set of all parameters.



AIC AND BIC

- The AIC and BIC are not used to test the model in the sense of hypothesis testing, but for model selection.
- Given a data set, a researcher chooses either the AIC or BIC, and computes it for all models under consideration. Then, the model with the lowest index is selected.

Now, how we can select? The question comes to our mind that, how we select the value of lambda? There are two ways of selecting the value of lambda. But before that, we are going to discuss two important model selection criteria. One is called AIC another is called BIC. So, what is the AIC, AIC stands for Akaike's Information Criteria, full name is Akaike's Information Criteria.

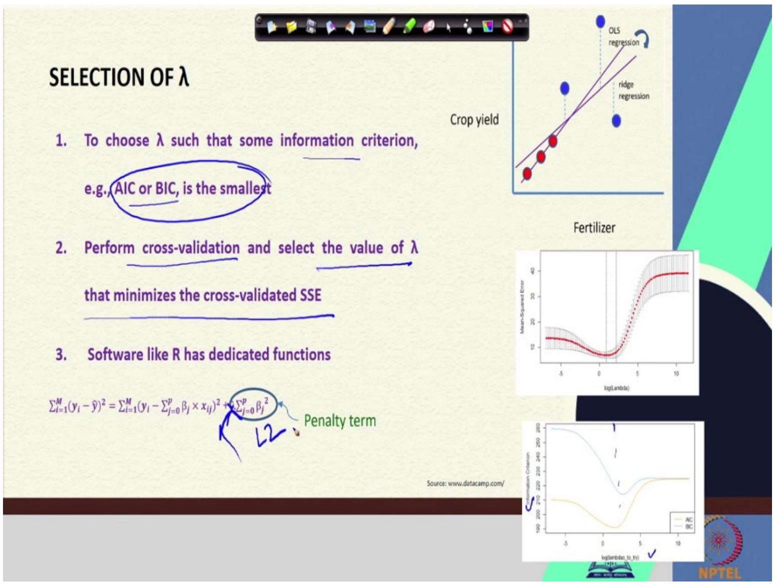
And so, the formula is AIC equal to minus 2 log likelihood plus 2K where K is the number of model parameters that is the number of variables in the model plus its intercept. All the betas plus the intercept and log likelihood is the measure of the model fit. So, the higher the number, the better is a fit, so this is usually obtained from the statistical output. So, based on when you input all these values, and then calculate the AIC.

Suppose there are 1, 2, 3 models, and for all of them we are calculating these AIC values. And the model, which will give you with using the same data set. So, the model for which we are getting the lowest value of AIC will be considered, so that model is preferred, so it is a feature selection. I would say it is a model, model selection criteria, and we generally select the model which shows the comparatively lowest AIC value.

Similar things goes to BIC which is the short form of Bayesian Information Criteria, the full, the value of these BIC is basically k log n minus 2 log L theta, where n is the sample size k is the number of parameters which your model estimates and finally, theta is the set of all the parameters. So, when you put all these parameters, then you get the BIC, similar to AIC you select the model with the least BIC value.

Now, the question comes, as I have pointed out that the AIC and BIC are not used to test the model, in the same sense as hypothesis testing, but for model selection. We just use these AIC and BIC for model selection only. Given a data set, a researcher generally chooses either the AIC or BIC and computes it for all the models under computation or consideration and then the model with the lowest index is selected. So, this is how these index are being used for AIC and BIC.

(Refer Slide Time: 09:22)



SELECTION OF λ

1. To choose λ such that some information criterion, e.g., AIC or BIC, is the smallest
2. Perform cross-validation and select the value of λ that minimizes the cross-validated SSE
3. Software like R has dedicated functions

$\sum_{i=1}^{M}(y_i - \hat{y})^2 = \sum_{i=1}^{M}(y_i - \sum_{j=0}^{p}\beta_j \times x_{ij})^2 + \sum_{j=0}^{p}\beta_j^2$

Penalty term

L2

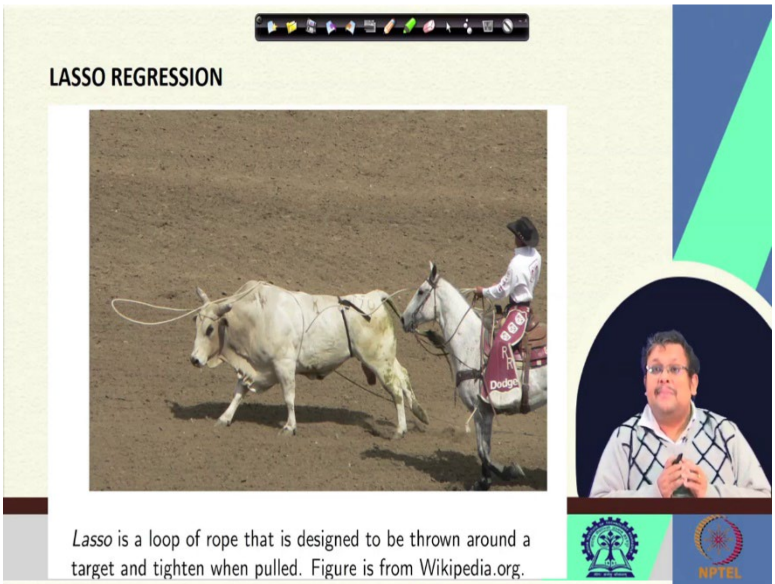
Source: www.datacamp.com/

Now, the reason I am talking about these is how these are important. Why we are talking about AIC and BIC, because selection of lambda somewhat depends on these AIC and BIC. So, the choose to choose this, the lambda, there are two general important methods. One is to choose the lambda. You can go with AIC or BIC and select the model with the, which produce the lowest AIC or BIC as you can see here, we are plotting log of lambda and then, these information criteria. And as you can see, for different values of lambda, the AIC and BIC are plotted and then you can select the one for which you are getting the lowest value of BIC and AIC.

Another approach for selection of lambda is you perform a cross-validation and select the value of lambda that minimizes the cross-validation sum square error. You know, the cross-validation we have already seen, so you plot their mean squared error for each cross-validation or you can go with the leave-one-out cross validation also, and then you plot against the log of lambda and the select the one which is producing the lowest mean squared error.

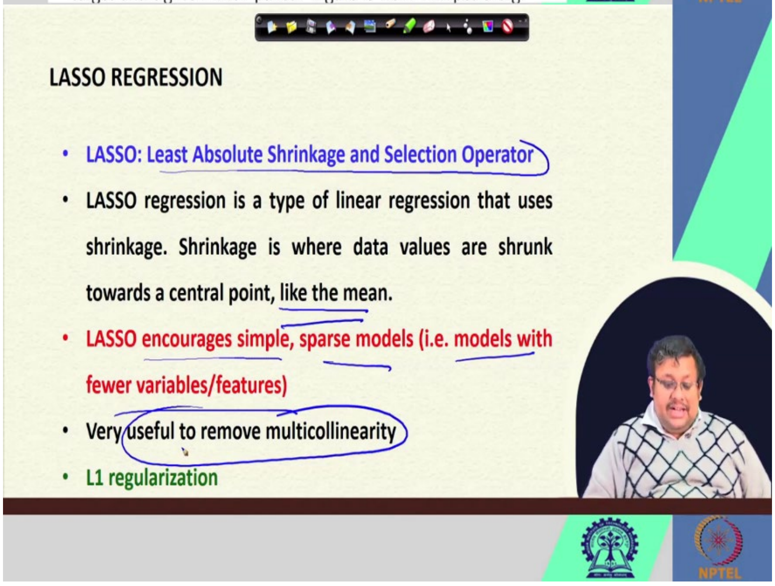
So, this gives you the idea about which value of lambda has to select, and you input that value of the lambda as a penalty term in this regularization L2 regularization, in case of ridge regression. So, this is how these values of lambda is being selected.

(Refer Slide Time: 11:11)



LASSO REGRESSION

Lasso is a loop of rope that is designed to be thrown around a target and tighten when pulled. Figure is from Wikipedia.org.



LASSO REGRESSION

- LASSO: Least Absolute Shrinkage and Selection Operator
- LASSO regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean.
- LASSO encourages simple, sparse models (i.e. models with fewer variables/features)
- Very useful to remove multicollinearity
- L1 regularization

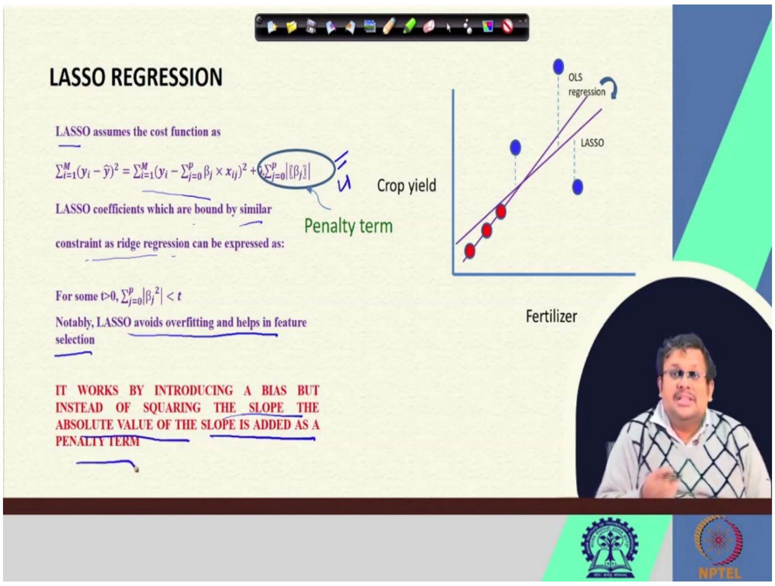
Now, another method which is important is the Lasso regression. Now, Lasso, remember it has some, although the full name of Lasso regression is somewhat different, but it has some similarity with the Lasso, which is a loop of rope that is designed to be thrown around a target and tighten when pulled. So, similar to these phenomena, you can see this is the Lasso, which is being used by these by these Matador had this you know bullfighting, but you can see here one important thing in case of our Lasso regression, it works in all in somewhat similar way.

It tries to throw you tries to make a loop of rope or kind of constrict the coefficient, so that we can selectively remove some of the useless coefficient from the or useless parameters from the model. So, let us see how it works.

So, LASSO, the full name is Least Absolute Shrinkage and Selection Operator. So, this is called Lasso, so Lasso regression is a type of linear regression that uses the shrinkage term. Now, shrinkage is where data values are shrunk to towards the central point like the mean. So, this is you see that resemble, so resemblance. So, you are when you are using the Lasso regression, it is basically shrinking the coefficients, where the data values are shrunk together to a central point like the mean.

So, remember Lasso always encourages a simple sparse model that means, the model which with the fewer variables and features, so it is very useful for removing the multicollinearity because multicollinearity when it happens, because multicollinearity happens, when there are excessive numbers of features, which are not useful and also these features may be also related to each other, so that is called multicollinearity. And when this multicollinearity happens, then Lasso regression comes into play. And the Lasso implies a L1 regularization, so, let us see what is L1 regularization.

(Refer Slide Time: 13:48)



LASSO REGRESSION

LASSO assumes the cost function as

$\sum_{i=1}^{M}(y_i-\hat{y})^2 = \sum_{i=1}^{M}(y_i-\sum_{j=0}^{p}\beta_j \times x_{ij})^2 + \lambda\sum_{j=0}^{p}|[\beta_j]|$

Penalty term

LASSO coefficients which are bound by similar constraint as ridge regression can be expressed as:

For some t>0, $\sum_{j=0}^{p}|\beta_j^2| < t$

Notably, LASSO avoids overfitting and helps in feature selection

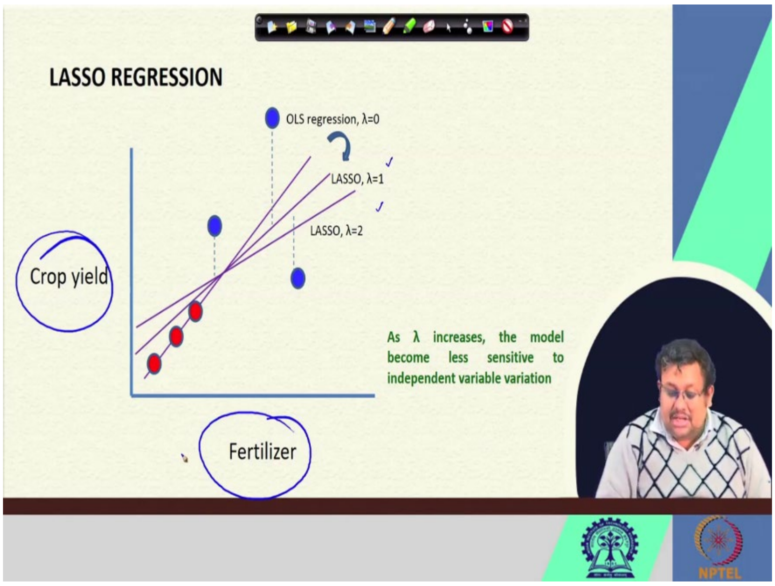
IT WORKS BY INTRODUCING A BIAS BUT INSTEAD OF SQUARING THE SLOPE THE ABSOLUTE VALUE OF THE SLOPE IS ADDED AS A PENALTY TERM

We have seen in case of in case of ridge regression, this is the penalty term. So, similarly so we have already seen in case of ridge regression, they are imposing a penalized term in the cost function of linear regression, but, in case of Lasso cost function, it is similar, but here instead of a square of these coefficient, we are taking the absolute value.

So, this is only the difference. Almost same, only the difference is here Bj square is not used only the absolute values of Bj is used or basically the absolute value of this regression coefficient is used. So, Lasso coefficient is which is bound by similar constant as ridge regression also can be, they can be bound by these similar constraints. And remember that Lasso avoids overfitting and helps in feature selection.

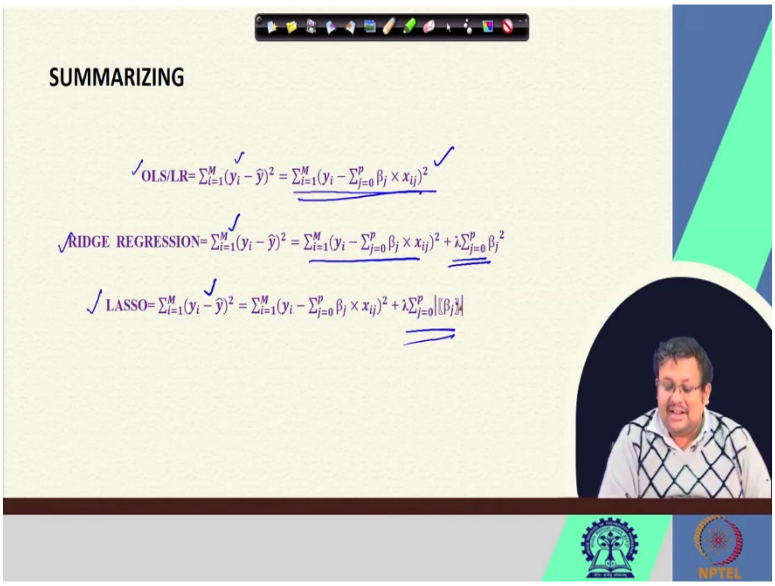
So, similarly, like ridge regression, it also helps in rotating the regression line so that it can reduce the bias, it can reduce the variance by imposing some bias, and this is called the L1 regularization. And it works by introducing a bias, but instead of squaring the slope, the absolute value of the slope is added as a penalty term. So, this is how this Lasso works, Lasso regression works almost similar to ridge regression, but how these penalty can be calculated is somewhat different.

(Refer Slide Time: 15:28)



LASSO REGRESSION

As λ increases, the model become less sensitive to independent variable variation



SUMMARIZING

$$\text{OLS/LR}= \sum_{i=1}^{M}(y_i - \hat{y})^2 = \underline{\sum_{i=1}^{M}(y_i - \sum_{j=0}^{p} \beta_j \times x_{ij})^2}$$

$$\text{RIDGE REGRESSION}= \sum_{i=1}^{M}(y_i - \hat{y})^2 = \underline{\sum_{i=1}^{M}(y_i - \sum_{j=0}^{p} \beta_j \times x_{ij})^2} + \lambda\underline{\sum_{j=0}^{p}} \beta_j^{\,2}$$

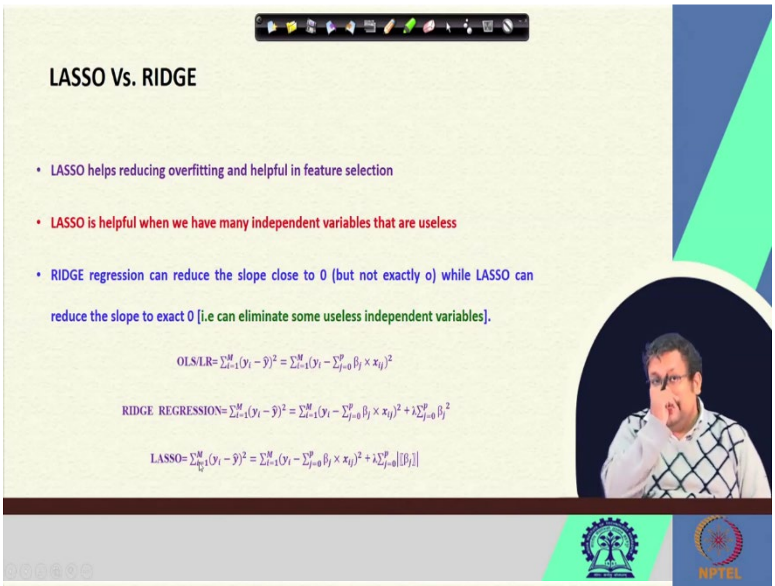
$$\text{LASSO}= \sum_{i=1}^{M}(y_i - \hat{y})^2 = \sum_{i=1}^{M}(y_i - \sum_{j=0}^{p} \beta_j \times x_{ij})^2 + \lambda\sum_{j=0}^{p}|\beta_j|$$

So, you can see here in case of Lasso regression also when we are increasing the value of lambda just like in case of the ridge regression, we are getting the variation of the model coefficient and at the same time we are getting less sensitive changes in the target parameter in our case it is crop yield within, with each unit change of fertilizer. So, as lambda increases, we can see the model becomes less sensitive to independent variable variation.

So, let us summarize we can see these are the three cost functions for ordinary least square regression, ridge digression and Lasso regression. Remember, this is basically all these three cases we try to reduce the sum square of error, so this is nothing but the sum square of error and in all these three cases, this is also known as cost function. So, in any type of regression the cost function is to reduce or minimize the sum square of error or residuals. And in this ordinary least squares or linear regression, this is a cost function where we are not introducing a penalizing term.

Here a penalty parameter, here we are this ridge regression we are introducing a penalty term. However, in case of Lasso, we are introducing the penalty term just like ridge regression. However, how we calculate this penalty term is somewhat different, instead of taking the square value, we are taking the absolute value of the coefficient. I hope this is now clear to all of you.

(Refer Slide Time: 17:18)



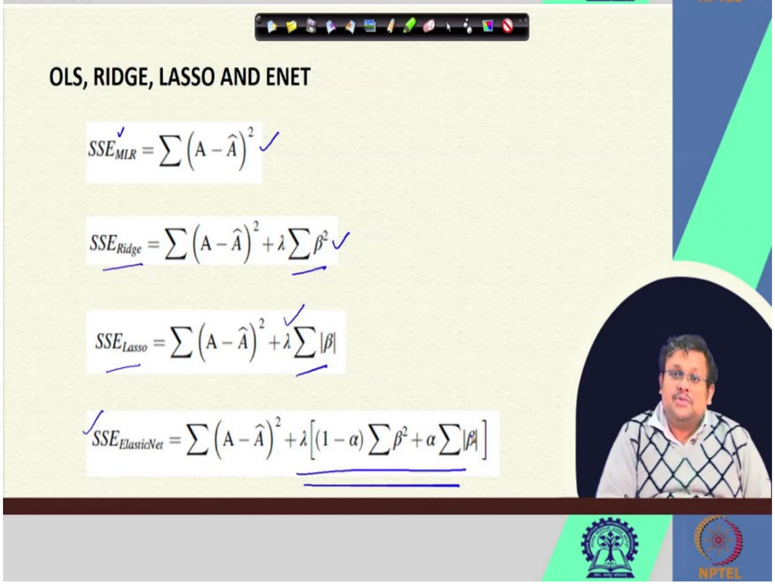
LASSO Vs. RIDGE

- LASSO helps reducing overfitting and helpful in feature selection
- LASSO is helpful when we have many independent variables that are useless
- RIDGE regression can reduce the slope close to 0 (but not exactly o) while LASSO can reduce the slope to exact 0 [i.e can eliminate some useless independent variables].

$$\text{OLS/LR}=\sum_{i=1}^{M}(y_i-\hat{y})^2=\sum_{i=1}^{M}(y_i-\sum_{j=0}^{p}\beta_j\times x_{ij})^2$$

$$\text{RIDGE REGRESSION}=\sum_{i=1}^{M}(y_i-\hat{y})^2=\sum_{i=1}^{M}(y_i-\sum_{j=0}^{p}\beta_j\times x_{ij})^2+\lambda\sum_{j=0}^{p}\beta_j^{\,2}$$

$$\text{LASSO}=\sum_{i=1}^{M}(y_i-\hat{y})^2=\sum_{i=1}^{M}(y_i-\sum_{j=0}^{p}\beta_j\times x_{ij})^2+\lambda\sum_{j=0}^{p}|[\beta_j]|$$



OLS, RIDGE, LASSO AND ENET

$$SSE_{MLR}=\sum\left(A-\hat{A}\right)^2$$

$$SSE_{Ridge}=\sum\left(A-\hat{A}\right)^2+\lambda\sum\beta^2$$

$$SSE_{Lasso}=\sum\left(A-\hat{A}\right)^2+\lambda\sum|\beta|$$

$$SSE_{ElasticNet}=\sum\left(A-\hat{A}\right)^2+\lambda\left[(1-\alpha)\sum\beta^2+\alpha\sum|\beta|\right]$$

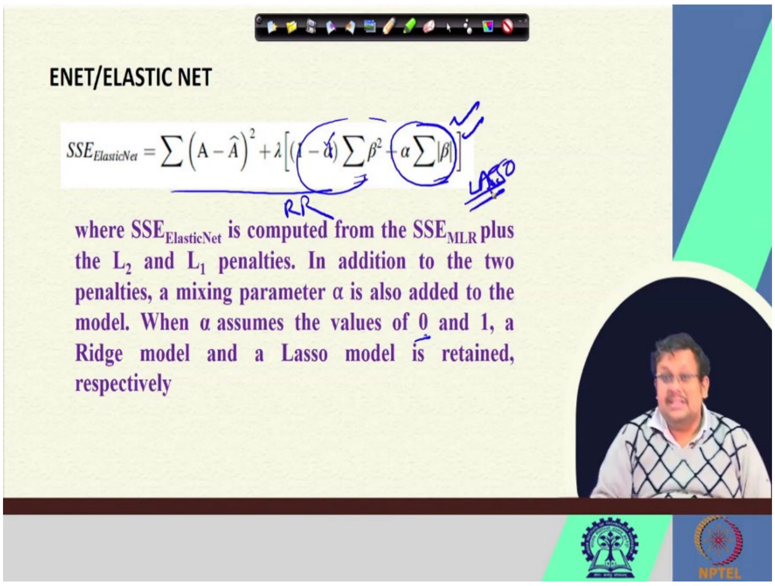
Now, what is the difference between how these Lasso and ridge contrast between each other. I mean, how they are different from each other. So, remember that Lasso helps reducing overfitting and it is also helpful in feature selection. And it is helpful when we have many independent variables that are useless.

So, some features are there which are useless to remove those features we use the Lasso regression, but ridge regression can reduce the slope close to 0, but not exactly 0, while Lasso can reduce the slope to absolute 0 or exact 0, that is, it can eliminate some useless independent variables. It can literally eliminate because it can reduce the slope to exact 0. So, you can see this is the difference between Lasso and ridge regression.

So, in other way, we can say that sum square of error in case of MLR is basically can be considered as the, the sum square of error which is predicted observed minus predicted the sum square and then, this is SSE ridge where we are putting these lambda beta square and then summation of beta square and then SSE Lasso is basically using the absolute values of b or beta and then some squared error of elastic net you can see here this this is another model.

So, instead of using these L1 and L2 regularization, it uses a mix of both L1 and L2 liberalization which is denoted by this term that is lambda multiplied by 1 minus alpha and then summation of beta square plus alpha summation of absolute value of beta. Now, let us see what are these?

(Refer Slide Time: 19:31)



ENET/ELASTIC NET

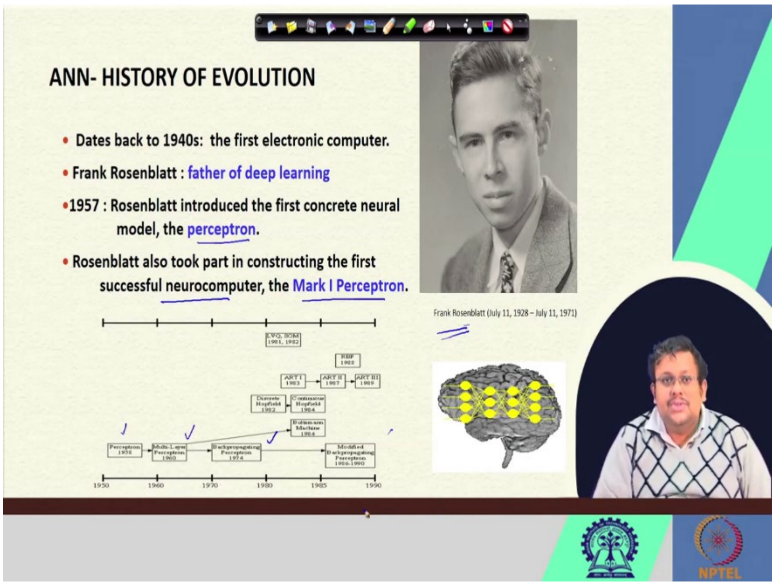
$$SSE_{ElasticNet} = \sum \left(A - \hat{A}\right)^2 + \lambda\left[(1-\alpha)\sum \beta^2 + \alpha \sum |\beta|\right]$$

where $SSE_{ElasticNet}$ is computed from the SSE_{MLR} plus the L_2 and L_1 penalties. In addition to the two penalties, a mixing parameter α is also added to the model. When α assumes the values of 0 and 1, a Ridge model and a Lasso model is retained, respectively

So, in this equation, where we are getting a mix of both L2 and L1 penalties or L1 and L2 regularization, we are getting here the mix while calculating the cost function of elastic net. So, here you can get the L2 regularization from ridge this is L1 regularization from the Lasso. In addition to these two penalties a mixing parameter of alpha is also added to the model.

Now, when the alpha assumes a value of 0 and 1 two extremes, a ridge model and a Lasso model is retained respectively. Of course, when the value is 0 that means, this term becomes 0. So, it assumes say the ridge regression, and when the alpha becomes 1, so, this becomes 0. So, that means this elastic net assumes the Lasso regression. I hope it is all clear to you.

(Refer Slide Time: 20:43)



ANN- HISTORY OF EVOLUTION

- Dates back to 1940s: the first electronic computer.
- Frank Rosenblatt : father of deep learning
- 1957 : Rosenblatt introduced the first concrete neural model, the perceptron.
- Rosenblatt also took part in constructing the first successful neurocomputer, the Mark I Perceptron.

Frank Rosenblatt (July 11, 1928 – July 11, 1971)

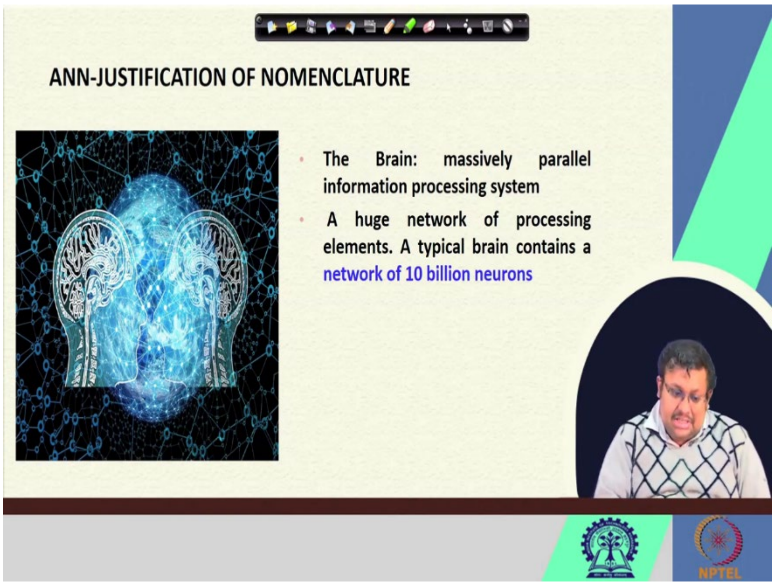
So, we have discussed the Lasso regression, we have discussed the ridge regression, we have also discussed the elastic net regression. Now, we will discuss one of the most important aspect of deep learning method that is called artificial neural networks. And we will first discuss the history of evolution of neural network, and how they modified and also then we will talk about their features and general overview.

So, the first use of neural networks you can date back to 1940s when the first electronic computer was introduced, but the father of this deep learning is considered, this American psychologist whose name is Frank Rosenblatt, he first introduced the first concrete neural model, which is known as the perceptron.

So, he was the one, he first introduced this concrete neural model called perceptron. And also took part in constructing the first successful neural computer that is Mark 1 Perceptron. So, you can see that the first, the perceptron or perceptron was built in 1957 and then the multilayer perceptron was developed in 60s, and then from there different advancement, where came into picture. And you can see back propagation perceptron came in 1974 and so forth also modified back propagation, propagating perceptron came in 19 in between 1986 to 1990.

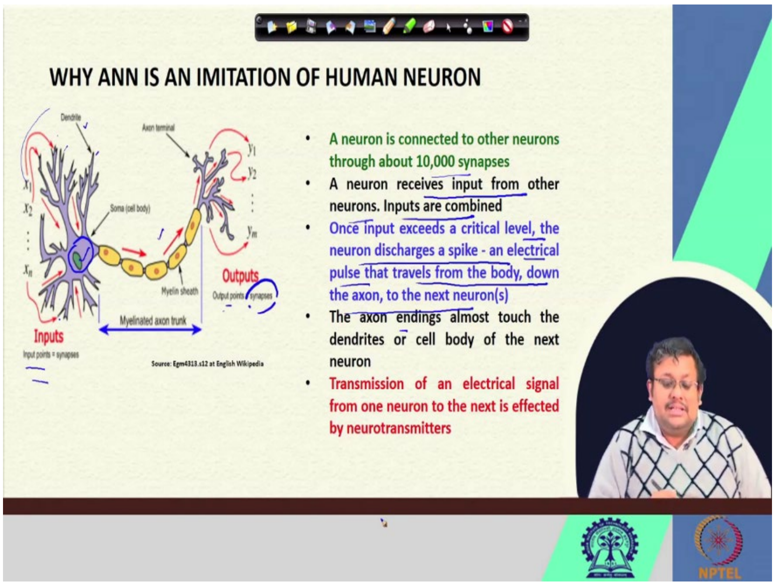
So, why we call it artificial neural network? Because it resembles the working or mode of operation of our brain, which is basically mediated by the network of neurons. So, that is why we call it artificial neural networks. So, this is one of the most widely used deep learning method, which we are going to discuss in our upcoming lectures. So, let us see more details about this ANN.

(Refer Slide Time: 23:26)



ANN-JUSTIFICATION OF NOMENCLATURE

- The Brain: massively parallel information processing system
- A huge network of processing elements. A typical brain contains a network of 10 billion neurons



WHY ANN IS AN IMITATION OF HUMAN NEURON

Source: Egm4313.s12 at English Wikipedia

- A neuron is connected to other neurons through about 10,000 synapses
- A neuron receives input from other neurons. Inputs are combined
- Once input exceeds a critical level, the neuron discharges a spike - an electrical pulse that travels from the body, down the axon, to the next neuron(s)
- The axon endings almost touch the dendrites or cell body of the next neuron
- Transmission of an electrical signal from one neuron to the next is effected by neurotransmitters

So, what is the justification? You can ask, okay sir, what is the justification of this nomenclature? Why we call it artificial neural network? Now, remember the justification comes from the human brain, because the human brain is basically a massively parallel information processing system, and we can consider these as a huge network of processing elements.

Remember that a typical human brain contains a network of 10 billion neurons. So, it is a intricate connection between the neurons and these neuron passes the information in synaptic way, just like as this mathematical model works, so, that is why we call it artificial neural network.

Now, why an ANN is considered Artificial Neural Network is considered as an imitation of human neuron. So, let us see what our human neurons actually does in the biological system. So, this is a picture of neuron, you can see different parts of the neuron. Here, you can see these are called the dendrites and these dendrites are getting information from different sources different other sources and then this is the, this is called the axon trunk and this is called the axon terminal.

So, this is these are the dendrites, this is the soma or cell body, this is the axon trunk and finally, which is the axon terminal. So, a neuron is connected to another neuron through about 10,000 synapses. What are the synapses? So, you can see that a neuron is connected to another neuron and these connections are mediated by these points, so these are known as synapses.

So, a neuron receives inputs from other neuron and inputs are combined. So, inputs are combined. So, here one input is given from one neuron and other input is given, so here we are getting signals from different neurons, which are connected to these dendrites through synaptic way. So, here you can see that x1 is a signal x2 is a signal x3, xn is in another signal, so these input points are known as the synapses and these input points are getting the signal from the other neurons and which are getting processed in this cell body.

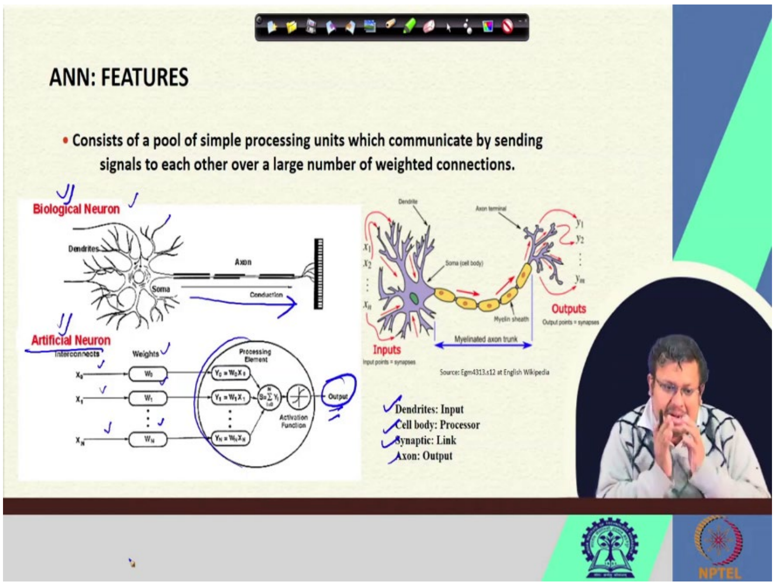
So, once inputs exceeds a critical level the neuron discharges spike that is an electrical pulse that travels from the body down the axon to the next neuron. So, the processing when these inputs are, when these from these inputs points or synapses these inputs or signals are processed in the cell body and exceeds a critical level, then these neuron discharges spike or in another electrical pulse that travels through these body and then down to these axon.

Remember, this axon is again combined with other neurons, where the dendrites of other neurons through a synapses. So, the out these are the output points from these output points, we are getting outputs. So, these are the output points and output points are also connected through these dendrites of the next several hundreds or thousands of neurons through synapses.

So, you can see all these are connected and the information is flowing from the input layer to the output layer and then it is being transformed to the next layer or next neuron in the system. So, the axon ending so, these are the axon ending almost touch the dendrites of the or the cell body of the next neuron. So, this is the dendrite. So, another cell is there, where their dendrites are also touching with this axon.

So, transmission of an electrical signal from one neuron to the next is affected by these neurotransmitters. So, this is how these artificial these human neuron our nervous system works, and the same principle is being utilized in case of these artificial neural network.

(Refer Slide Time: 28:22)



ANN: FEATURES

- Consists of a pool of simple processing units which communicate by sending signals to each other over a large number of weighted connections.

Source: Egm4313.s12 at English Wikipedia

Dendrites: Input
Cell body: Processor
Synaptic: Link
Axon: Output

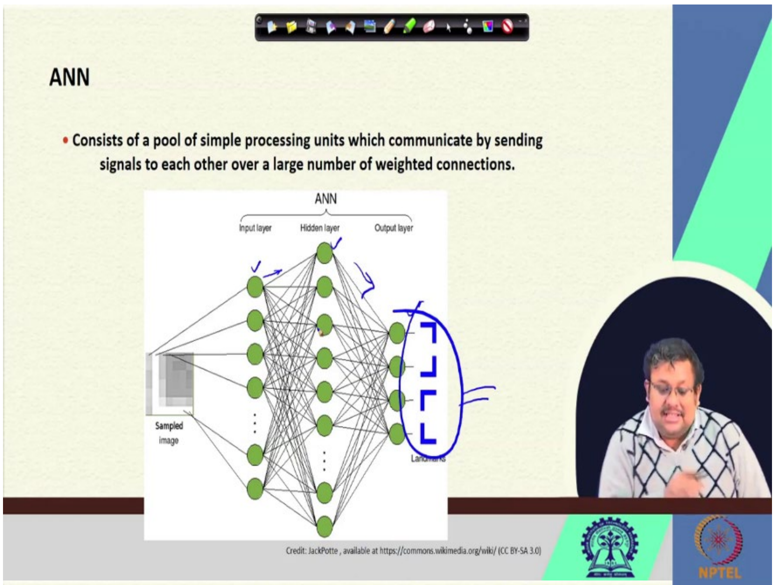
So, if you see the features of artificial neural network, it consists of a pools of simple processing unit, which communicate by sending signals to each other over a large number of weighted connections. So, you can see the biological neuron we have just described, they have dendrites, which are getting the which are the entry point of the input signals, and then they are being processed in the soma and then the resulting output is being transferred by this axon to these axon ending which is also linked to these next, the dendrites of the next neuron.

So, similarly, in case of artificial so, here the dendrites can be considered as input points, cell body is the processor, synaptic these are the links between the different neurons and axon is basically the outputs and they content the outputs.

So, similarly, in case of the artificial neural network, we are getting the inputs from different sources. So, these can be considered as the dendrites and then we are having they have different weights and then they are being processed in the cell body, which are also known as our in our case it will be processor and then there will be an activation function, which can give you the output so, you can consider this output as the axon.

These synapses are the connections and these dendrites are basically these points from where these input signals are coming and the cell body is the processor where these signals are being processed. So, you see that this biological neuron is being imitated by this mathematical model, so we call it artificial neural. And since these artificial neural, neurons are making an interconnected web to transform the information from one layer to another layer to get the output to predict output, that is why we call it artificial neural networks. So, I hope now, it is clear to you why we call it artificial neural network.

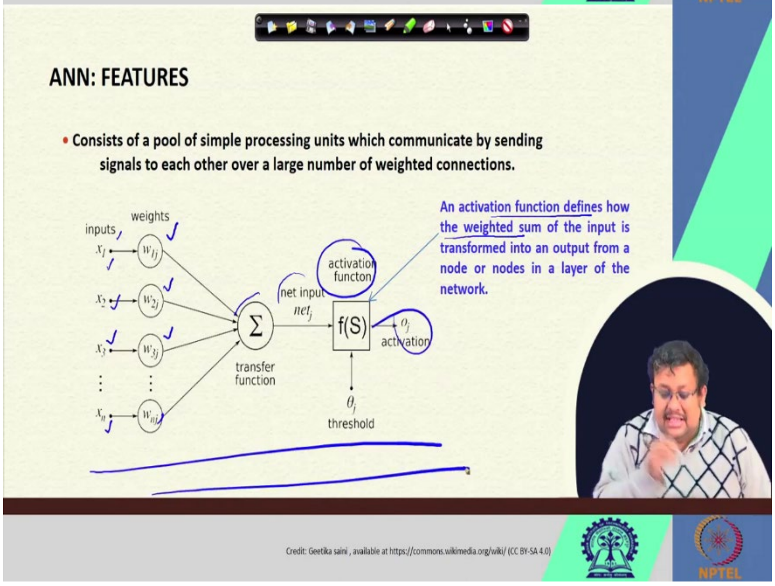
(Refer Slide Time: 30:39)



ANN

- Consists of a pool of simple processing units which communicate by sending signals to each other over a large number of weighted connections.

Credit: JackPotte , available at https://commons.wikimedia.org/wiki/ (CC BY-SA 3.0)



ANN: FEATURES

- Consists of a pool of simple processing units which communicate by sending signals to each other over a large number of weighted connections.

An activation function defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network.

Credit: Geetika saini , available at https://commons.wikimedia.org/wiki/ (CC BY-SA 4.0)

Now, in artificial neural network mathematically it consists of a pool of simple processing units, which communicate by sending signals to each other over a large number of weighted connections. So, you can see in the artificial neural network there is an input layer, there is some hidden intermediate layer and output layer.

And all these are connected by giving some weighted connection, so these are all connected. So, information generally pass from these input layers to these hidden layers to these output layers and ultimately we get the final output in this output layer. So, this is how these artificial neural networks works in a mathematical way.

So, the artificial neural network consists of pool of simple, so you can see here this is a mathematical representation. So, there are these are the inputs x1, x2, x3, xn these are the inputs and each of them have their assigned weights and then they have a transfer function they goes to the processor there is an activation function.

What is an activation function? So, before going to the activation function, remember, the net input is being calculated. So, first we get the inputs and their associated weights are being used to calculate these net input and they are being influenced by this transfer function. Ultimately, there is an activation function the activation function defines how the weighted sum of these inputs is transformed into an output from the node or nodes in a layer of the network.

So, this is basically considered this basically considers how this information will flow from one layer to the output layer, and ultimately, it will gives us the activation. So, this is how these ANN works. And this is the in a nutshell, this is how this ANN works. We will discuss more about these ANN in our next upcoming lecture, but I hope that you have now and basic understanding of artificial neural network. Why we call it artificial neural network? And then you have also gained knowledge about these linear regularization method.

So, I hope that you have got some good information in these two lectures, in these lectures of this of this weeks, which we have already completed. In our upcoming lectures we will talk more about these artificial neural network, and how we can use these neural network. What are the other variants and how can we use these neural networks for the processing of the data for soil and crop images. So, we learn in our upcoming lectures. So, thank you let us meet in our next lecture.