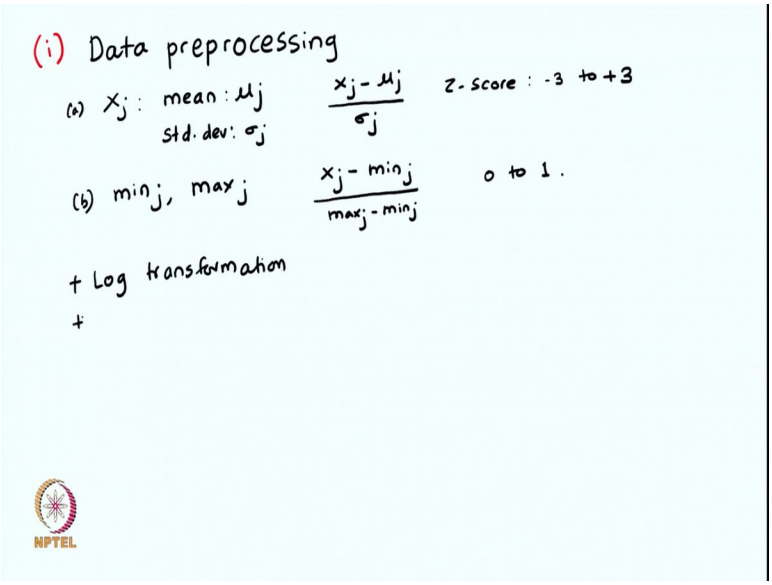
Practical Machine Learning with TensorFlow
Dr. Ashish Tendulkar Google
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Lecture – 03
Steps in Machine Learning Process

In the last session, we studied how traditional programs are different from the machine learning systems or in other words how machine learning algorithms are different from the traditional programs. We also looked at data as an important prerequisite for machine learning and learned various characteristics of data like features, labels and different type of features.

Depending on the type of the label we get different machine learning algorithms and we also studied those types in the previous session. Now that we have data we know what is a machine learning process, let us try to get into the details of the machine learning process in this particular session.

(Refer Slide Time: 01:19)



Data pre-processing: in this case, we take the data that is given to us the data comes from different measuring instruments or through different business processes and since there

are a lot of stakeholders or lot of pipelines generating the data. Some of this pipeline might be erroneous and that leads to some of the data quality issues.

So, indeed the in data pre-processing we try to remove inconsistencies in the data try to look for outliers, which might occur due to some of the erroneous process and try to remove those outliers. The data quality maintaining the data quality and getting high-quality data to the training process is key to building successful machine learning algorithms. Let us try to understand some of the common steps in data pre-processing.

So, often what happens is that we have different features for data and these features are on different scales. For example, in case of housing price prediction, we had an area of the house in square feet which will be in 100s, but the number of bedrooms in the house will be less than 10. You can see that when we try to train machine learning algorithms with features that are on different scales, that causes us some kind of optimality in the training process. The first thing that we do is we normalize features and bring them to exactly the same scale, normalizing helps us in getting better convergence in the training data.

So, how do we exactly normalize the data? We will look at one of the technique for normalization it is called as z-score normalization. So, what happens is that for a particular feature let us say for feature x_j we compute the mean let us call it as x_c and the standard deviation which is σ_j. Then what we do is we calculate the normalized value for a new value: $x_c = \frac{x_j - u_j}{\sigma_j}$

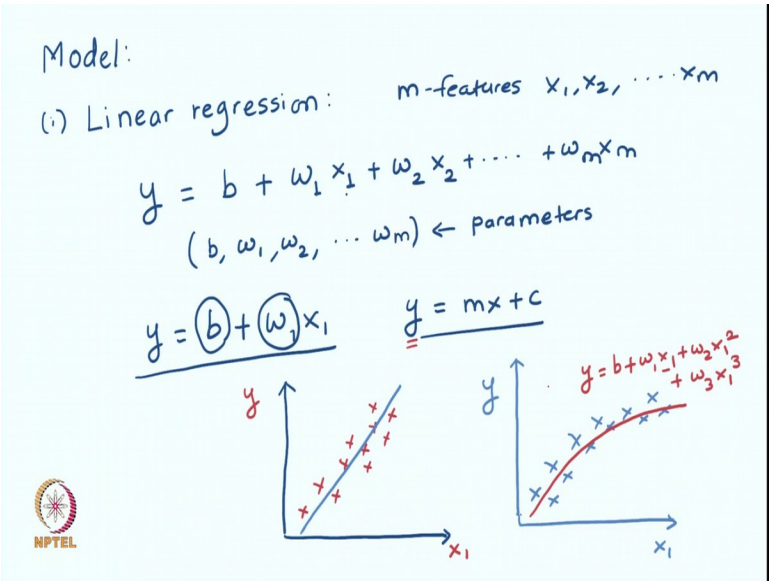
This particular formula gives us the distance of the point with respect to the mean in terms of the standard deviation. This is called as z score normalization and this brings the feature x_j in the range approximately -3 to +3. This is one of the possible ways of normalizing the data, other way could also be we find out what are the minimum values let us call it as min_j and max_j; max_j is the maximum value of the feature min_j is the minimum value of the feature and what we do is, $x_c = \frac{x_j - min_j}{max_j - min_j}$. This also helps us in getting all the features in the range 0 to 1.

These two are the techniques that are used quite often in normalization. Apart from normalization sometimes we use log transformation in the features to get them to a new domain. So, we also use what is called as log transformation or we can also apply it is a square root transformation.

So, all these transformations along with normalization, so all these constitute to what is called as data pre-processing in the domain of machine learning. After pre-processing data, we also encourage you to visualize the data and explore it and try to understand the relationship between the features and the labels.

So, now that after data pre-processing we have a data ready for our next step. So, once the data is ready now what is the next task in machine learning? The next task is to build a model. What is the model? Model is nothing but a mapping from the features to the labels.

(Refer Slide Time: 06:57)



So, second step is a model building step. The simplest model is a linear regression model, who specified these models. You can see that when we were talking initially about difference between machine learning and traditional programming. We said that machine learning model essentially maps input to the output. It is some kind of a mapping or functioning other words and you can see that there are infinite function classes that are possible. But we take some leap of faith and assume some kind of a function class. When we are selecting this function class we also take into account some

of the domain knowledge as well as some knowledge that we have gathered while practicing machine learning model building.

We generally begin with simpler models, the simplest model is a linear regression model which has the following form y is the feature, y is essentially $b + w_1x_1 + w_2x_2 + \ldots\ldots w_mx_m$. You can see that in this model b, w_1, w_2 all the way up to w_m are the parameters, y is the label and x_1, x_2 up to x_m are the features. So, we have a setting where we have m features and we come up with a very simple mapping between the features to the label and we hypothesize that there is a linear relationship. Geometrically this represent an equation of a hyper plane.

Let us try to understand this with respect to a single feature, just to build an intuition about linear regression let us assume that we just have a single feature. Then we have a very simple model y is equal to $b + w_1 x_1$ this is a familiar equation to each one of you. We studied this in high school this is very similar to the equation that we studied in high school which is $y = mx + c$. Do you remember this equation? This is an equation of line and in this case there are two parameters b and w_1 are 2 parameters that we want to learn. And b is called as bias, but it is really a y intercept and w_1 is the slope of the line.

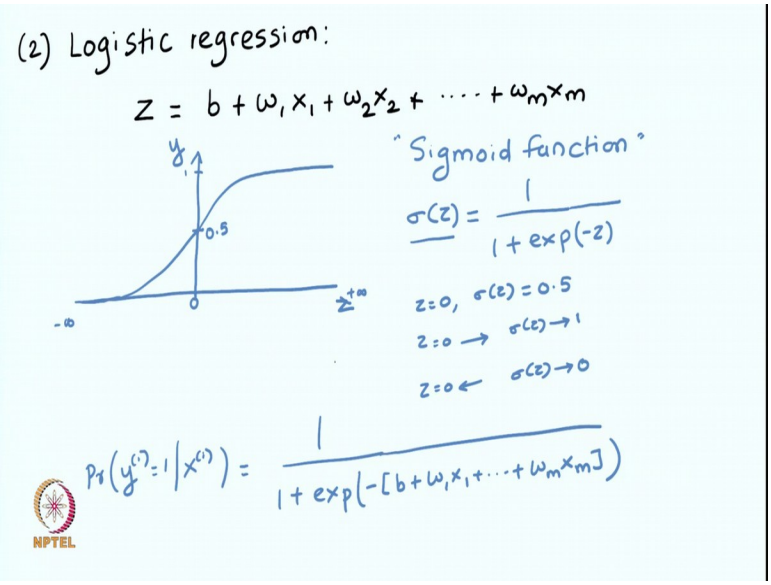
Let us try to represent this line geometrically, let us say these are some of the points we have x_1 and y as an output x_1 is the feature y is the corresponding label, this is one such line this line passes x passes the y-axis at some negative number and it has got some slope. This is an example of a simplest model that we can use to map the input feature x_1 to the variable y. We can also come up with some complex models where let us say we have data which is distributed like this.

Let us say this is feature x_1 this is y, obviously in your model is probably not a good choice. So, data is slightly data is represented by a slightly higher order model. So, here we can say that this is essentially we can use some kind of a polynomial model, then we say that this is $b + w_1 x_1 + w_2 x_1^2 + w_3 x_1^3$. You can see that we raise the power of the input to the second order and the third order and this is a polynomial regression model that we are using to map the input x_1 to the output y.

This particular approach works very well when the output y is the real number, because by solving this particular equation we will get a real number. What kind of models can

be used for the classification task, where *y* is a discrete quantity as in case of handwritten digit recognition.

(Refer Slide Time: 13:23)



Logistic regression: So, instead of predicting a real number we are interested in predicting some kind of a discrete quantity. Let us represent the real number that we got out of linear regression using *z*; let us say *z* is $b + w_1 x_1 + w_2 x_2 \ldots w_m x_m$; given m features we can perform the multiplication of the parameter with the feature value and then we add all these things together this is called as a linear combination. The linear combination of features and a parameter values we get *z* and our job is to take this particular real number and convert that into a discrete quantity.

Let us say you want to predict whether a particular house will be sold or not. So, now here the label that we are interested in is whether house will be sold: which is which will be represented by number 1 and if house is not sold we call it as label 0 and we have all the features of the house and based on that we want to predict whether house will be sold or not. So, what we can possibly do is, we can create a linear combination of the features and the parameters and we get this intermediate representation which is *z* which is the real number and we want to convert this real number into a discrete quantity between 0 to 1.

So, we have a special function called logistic function for that. Logistic function has *z* let us say going from -∞ to +∞ and let us say this is your 0 and this is the *y* that we are

trying to predict *y* goes between 0 to 1. So, what happens is this particular function takes the real number and squashes it between 0 to 1; it crosses the y-axis exactly at 0.5 and this *S* shape function is called as a sigmoid function.

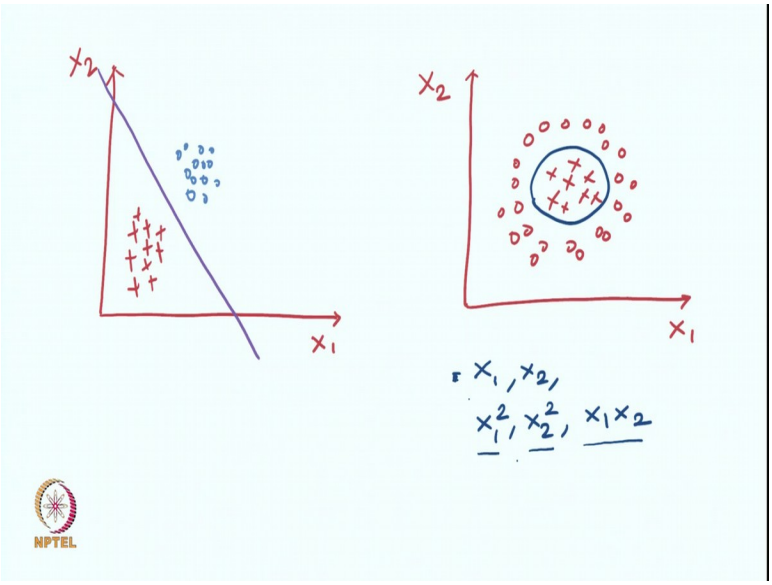
The sigmoid function is represented by *σ(z)* is nothing but 1 over 1 plus exponential of minus z and all that it does is it does linear combination followed by a sigmoid which is this particular function and *σ(0) = 0.5*. As we go away from 0 we get value closer and closer to 1 we go from 0 on the right hand side this tends to 1. If we go away from *z* this will tends to 0 0 at 0 it has got 0.5.

If we go away from the mean in the positive direction sigmoid value tend to 1, if we go to the left in the negative direction of z we get σ value closer to 0. This is a sigmoid function if we apply the sigmoid function what we get is we get a number between 0 to 1. So, which we can conveniently interpret as a probability. So, we say that the

$$Pr\left(y^{(i)}=1/x^{(i)}\right)=\frac{1}{1+e^{-(b+w_c+.....+w_m x_m)}}$$

This is the sigmoid function it predicts the probability that the data item will take table one given it is feature. Now, you will be wondering what kind of decision boundary, because we are trying to essentially divide the area into positive class and a negative class.

(Refer Slide Time: 18:37)



So, what we do is let us try to see what kind of decision boundary logistic regression gives us. There are let us say two different kind of classes: crosses and circles. This could be one of the potential decision boundary between two classes circles and crosses. Logistic regression is also linear classifier because, we get some kind of a linear classification boundary in the most basic case.

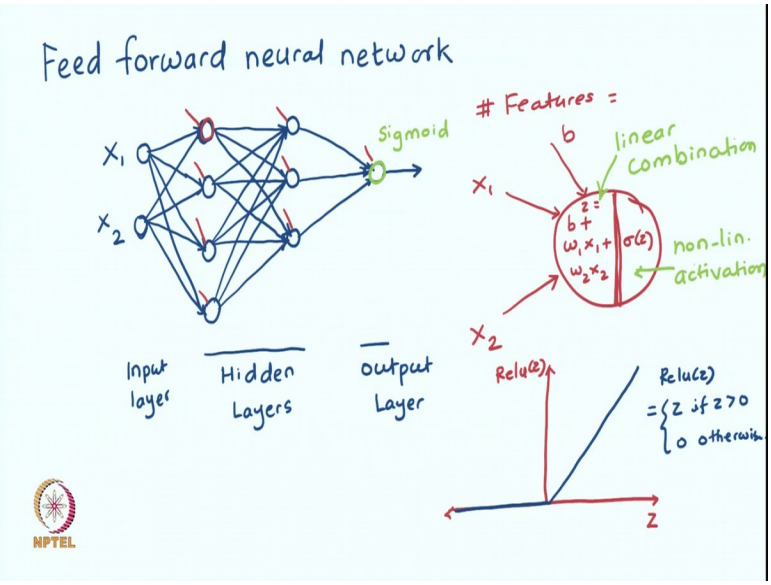
You must be wondering what if two classes are not linearly separable. Let us take a situation where we have cross inside all the circles, we forgot to name the these are features x_1, x_2 feature x_1 and this is the feature x_2. So, here what we can do is we can use polynomial features to get a decision moderate like this. So, instead of just using feature x_1x_2 you also introduce features like x_1^2 x_2^2 , x_1x_2 which is an interaction feature made up of two individual features.

So, from two features we will get these five features and see if we can find a decision boundary that is able to separate two classes ok. So, we saw that in order to separate this particular non-linear nonlinearly separable classes, where the circles and crosses are separated by a circle or a circular boundary. We what we did is we took the feature original feature x_1 and x_2 and constructed more features by calculating the feature cross.

The feature cross gave us three more feature which is x_1^2 x_2^2 and x_1x_2 by using all these five features we are able to construct such a decision boundary. However, if we have a very large number of input features constructing such kind of feature crosses by hand can be very expensive. Let us look at a technique called neural network or a feed forward neural networks that will help us to free up some of our task of generating this feature crosses and neural network does this feature crosses automatically.

Let us see how neural network does it, we are anyway going to study neural networks in far more detail in the next section, but let us try to understand neural network from basic perspective. So, what neural network does is? So, we are going to look at a specific neural network called Feed-forward neural network.

(Refer Slide Time: 22:23)



So, we can think of a neural network as a mechanism to construct complex functions by taking simpler functions. So, we are essentially going to construct complex function by composing simpler functions. Let us try to understand that in more detail. In the context of this particular example we had two features x_1 and x_2 they are the inputs to the neural network. Then we can have a toy neural network which is to begin with we can have maybe another layer with three hidden units and then we have an output layer with a single unit.

And this output layer will output one of the two classes. It will output the probability of the class one or the positive class and what happens is that we are going to connect x_1, we are going to send x_1 to all units in the next layer this is called a dense architecture. where the unit over here is getting input from all the units in the previous layer. In the same manner we are going to send this particular input to each node in the next layer and each unit from second layer we are sending it to the next layer.

Now, you can see that the unit so let us get the terminology right. This is this layer is called as the input layer of the neural network, these two layers are called as hidden layers and this is an output layer. And on we also have we have we also have a bias unit to each of the nodes here. You can see that from just two features which is x_1x_2 we have constructed a complex representation containing far more features. So, how many features are there in all in this neural network.

The number of features is equal to we can see that. So, we have two features to begin with we take these two features and make four more features out of this and then there are three more features over here the effect of this is that we get a model with far more capacity or in other words which has got a very large number of parameters. How many parameters are in this model? The number of parameters are equal to the number of the number of connections. That is there is one to one relationship between number of parameters and connections, each connection or each connection represent one particular parameter and each unit over here has got two parts to it.

Let us concentrate on this particular unit. So, it has got two inputs x_1 and x_2 it also has an additional unit called bias unit. This particular unit what it does is it first does linear combination which is $z = b + w_1x_1 + w_2x_2$. and this is passed through some kind of a non-linear activation. It is a common practice to use Relu as an activation function here the Relu function looks like this.

So, for it is x axis is the value of a linear combination and y axis is a non-linear activation.

$$Relu(z) = max(0, z)$$

Since we are interested in having two classes or a binary output we use instead of Relu we use a sigmoid activation. So there were two steps one is linear combination followed by non-linear activation.

We will see in more detail why we use non-linear activation in the next stage in the next class or in the next session. This to you know complete the point it is important to use non-linear activations. Here non-linear activations help us to build models corresponds to a non-linear surface. For example, if you want to build a model correspond corresponding to the circular boundary, non-linear activations make that particular thing possible alright.

So, neural network is one such model that as I said earlier helps us to compose help us to you know break down the complex function into simpler functions and we combine the output of this simpler functions to get us you know far more powerful models in terms of their capacity or number of parameters. In the next session we will study how to train these models, hope you enjoyed this session. See you in the next session [FL].