

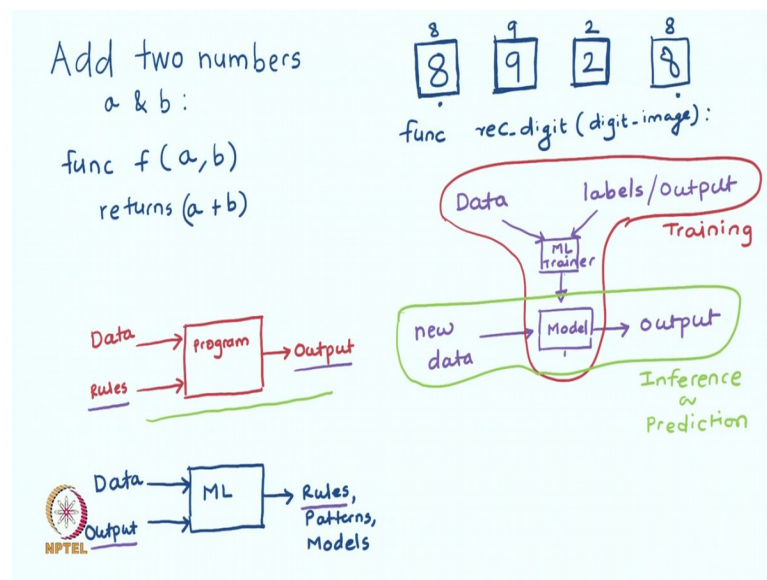
**Practical Machine Learning with TensorFlow**  
**Dr. Ashish Tendulkar Google**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Bombay**

**Lecture – 02**  
**Machine Learning Refresher**

[FL]. In this module we will study the basics of machine learning this is more like a refresher, we assume that all of you have a basic background in machine learning. But, this session is meant to be a refresher of machine learning and we will also understand some of the machine learning concepts using neural network playground as a visualization tool. What is machine learning? Let us try to understand machine learning from a programming perspective.

So, how the programming is different from machine learning? We will try to answer that question first and then slowly go into basic terminologies of machine learning and various different modules of the machine learning systems. So, let's try to understand machine learning again from a programmer's perspective. Let's take two problems.

(Refer Slide Time: 01:37)



The first problem is let's try to let's write a program to add two numbers a and b, most of you will wonder what is a question this is such a basic question probably this particular program is among some of the early programs that all of us have written, yes you are right. So, how do we really write this program? We essentially write a function  $f()$  which

takes two arguments  $a$  and  $b$  and then it returns  $a + b$ . This is a program that all of you are familiar with, we can add two numbers very easily by writing a computer program.

Let us try to solve a slightly different problem with the same technique and we will see whether we can solve it or if we need some more tools in our toolkit. The second problem is let's say, we have a bunch of handwritten digits this is 8 this is 9. So, what I am doing is I am fixing an area in which you can write these digits and now the task is - can you write a program to recognize these digits. Your job is to write a function that recognizes digit given the picture digit image.

So, can you write a program just as you did for the addition of two numbers to recognize handwritten digits, you can think for a couple of minutes and try to give the answer of this particular question. Now, I can imagine that some of you must have started thinking about writing rules for different kind of numbers. Are rules really scalable? What if I write the number in a slightly different orientation or I write a number in a very different style, probably rules will break rules would not be able to cater to all the situations. But as a human being, we are able to recognize these numbers.

What makes us recognize these numbers? We will come to this question in a bit. But before that can we write down the process of recognizing these digits just as we did in the other problem with where we added two numbers. When we were given two numbers  $a$  and  $b$  we immediately came up with a step or we immediately came up with a function to add two numbers which was simply  $a + b$ . But as you can imagine or as you must be facing right now it is incredibly hard to come up with the stepwise process to recognize the digits.

So, how do we really solve this problem? And before getting into solving the problem I would also like you to think what is a difference between these two problems, why am I able to solve the first problem very easily, but the second problem is a bit of a harder problem for me to recognize digits with computers. What are the key differences between these two problems? In the first problem, the formula to add two numbers was known to us. So, given two numbers  $a$  and  $b$  I can simply do  $a + b$  and that gave me the answer.

But in case of the second problem where I am when I am trying to recognize digits, I am able to recognize it with my vision but I am unable to come up with steps that I can code

up in the computer so that computer can also start recognizing digits. So, we need to do something else: machine learning. Let us take a step back and try to understand why we are able to recognize these digits you can think that we have we are seeing these kinds of digits right from our childhood. When you started our formal education we are introduced to these digits and when and we have also observed many people writing these digits.

So, somehow our brain is trained to recognize these digits even if they are written in a slightly different style or in a slightly different orientation. I can easily recognize that this particular number is 8 and this number is also 8 even though they are written differently. Can we try to mimic the training that we provided to a brain, can we give the same training to a computer? Let's try to explore that. This is the question that machine learning tries to explore. So, let us write down the key difference between the programming the traditional programming paradigm and the machine learning.

This is our traditional programming world, where we have a program we give some data as an input and we also input the rules rather we code these rules in the program and then pass the data into this program, the rules get applied on the data and we get the output. We did exactly the same thing while adding two numbers when you sort the numbers we also give step by step instructions to the computer as to how to sort these numbers.

Now, let us look at how machine learning operates, remember the handwritten digit recognition examples example and we see that we have data, but we do not have rules. We cannot write a traditional computer program, but we can actually provide lots of examples of handwritten digits along with the corresponding digit. For example, I can say that this is the image and 8 is the digit corresponding to this particular image, 9 is the digit corresponding to this particular image, 2 is the digit corresponding to this particular image and this is 8. We have lots of examples where we have images of handwritten digits along with their actual labels, which are nothing but the numbers that are there in the handwritten digit.

We have data and we also provide the intended output as input to ML and machine learning comes up with rules or sometimes you also collect patterns or models got it. You can now see a clear difference here let us highlight that you can see that the rule is

on the left-hand side here, the rule is on the right-hand side here and the output which was on the right-hand side had moved to the left-hand side has moved to the input side.

The traditional program takes data and rules as input, the rules are applied to the input data to produce the output in case of traditional programs. In the case of machine learning, we have data and the output as the input given to the machine learning and machine learning comes up with rules or patterns or models that it sees in the input data. We learn details of this particular process as we progress in this lesson, but this is the key difference between the traditional programming paradigm and machine learning.

We will write down the steps in the machine learning process here. So, we have data and we have labels. Input them to machine learning trainer. The trainer looks at the input data and corresponding labels or you have mentioned labels also as outputs earlier. So, let us call it as output to make it consistent with the earlier representation and this gives us a model or rules; the model is nothing but the mapping of input to the output.

Once we get this particular model what we do is, so we can take the new data and pass it through the model to get the output. You can see that once you get the model the process is exactly the same as the programming world, because once I know the model I know exactly the formula to map the input to the output. The process or the work that we do in machine learning training is to take the data and desired output and use machine learning trainer to come up with a model and once we have modelled we can use that model to get output on the new data.

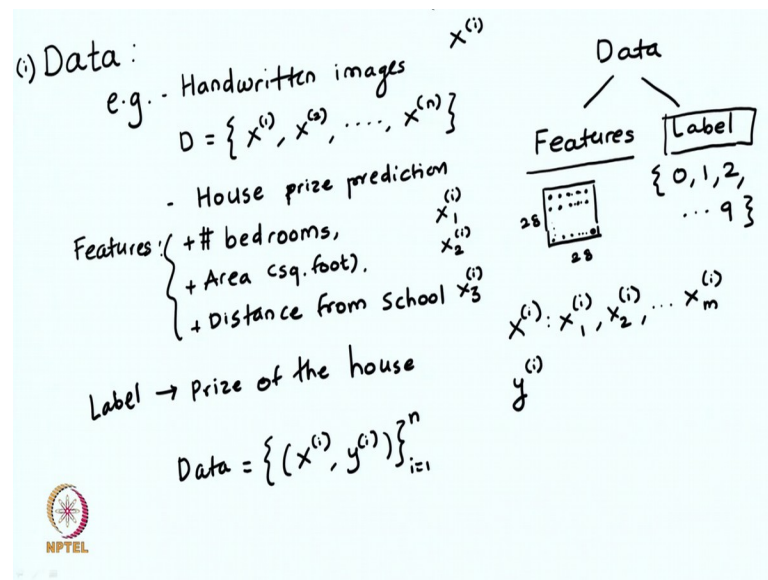
Now you can see that, so there are two stages in the machine learning process. This particular stage where we had data and we got to model. This particular stage is called as a training phase. There are two phases one is called training and there is another phase where we take the model to take this new data and get the output. This particular phase is called as inference or prediction.

There are two steps one is training. Training is nothing but given data and the output comes up with the model or the formula and once we have the model we apply that model in the new data to get output. You can see that inference and prediction are very similar to the traditional programming paradigm, while this training is something new to all of you if you have a programming background and we will try to understand this particular process training as well as inference in detail as we progress in this course.

Now that you have understood how machine learning algorithms are different from the traditional programming world and you have also understood two broad steps in the machine learning pipeline. It is a time to go through some of the terminologies and understand them in a bit more detail.

What are the key components?

(Refer Slide Time: 15:57)



So, first is the first component is data; data is an important prerequisite of machine learning. You must have heard a term called data is new oil and data is indeed very very important if you want to train machine learning models, if you do not have data we probably would not be able to train machine learning models.

So, data is first and the most important aspect or important input or important prerequisite for a machine learning model. So, what are the some of the examples of the data? So, in the example that we saw the handwritten images are example of the data. There are multiple images we use normally  $x^{(i)}$  to represent  $i^{\text{th}}$  data point. So, let us say  $D$  is the data and in this data we will have lots of images  $x^1, x^2$  all the way up to let us say if there are  $n$  images  $x^n$ . These are all the images in the data, so this is an example of handwritten images.

If you are trying to predict the price of the house based on some of the attributes of the house, so in that case, the house is a data point. So, we will have a bunch of houses over

here. What happens is that data has two important components one is called features and second is called label. Features are nothing but the attribute of an item, in case of the handwritten digit, what could be the features?

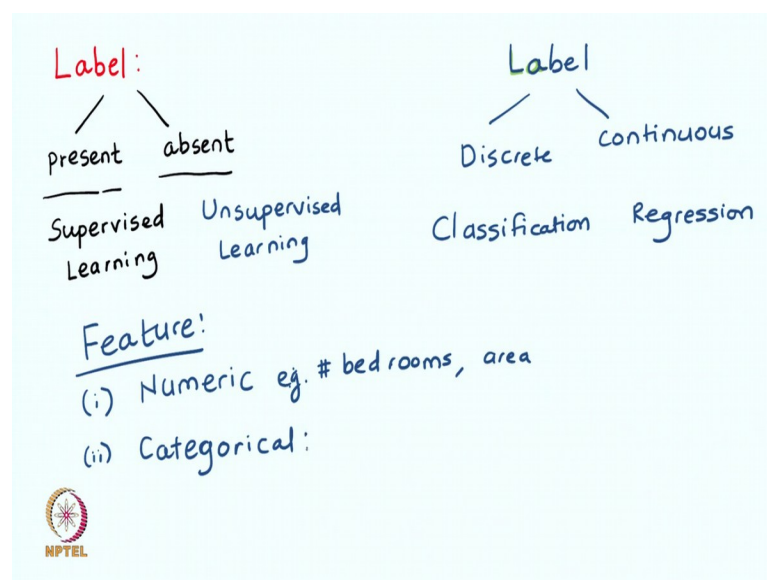
So, if let us say our handwritten digit image is in 28 by 28 grid, the value of each of the pixel is a feature for this handwritten digit problem. In case of housing price prediction house price prediction, what could be the features? The features could be the number of bedrooms area in square foot, let us say a distance from school and there could be many more such kinds of features, there are some of the features that I am denoting. Those are the features in case of housing price prediction.

Now, the label is the thing that we are interested in predicting; in case of handwritten digits the label is the digit between 0 to 9, because our task is to take an image and predict one of the 10 labels. So, labels in case of handwritten digit could be 0, 1, 2 all the way up to 9. There are 10 possible labels in case of handwritten digit recognition.

In case of housing price prediction these are all features and the label is the price of the house; price of the house is the label. We denote each feature using subscript, so we can say that this is feature  $x_1$  this is feature  $x_2$  this is feature  $x_3$  and we use superscript inside bracket to denote the index of this particular data point in the data matrix.

What we have in data concretely is a feature and a label and features for  $i^{\text{th}}$  data point are there are  $n$  features all right  $x_1, x_2$  all the way up to  $x_m$ . And then there is a label which we denote with letter  $y$  and we also use the same superscript to denote that this is the label for  $i^{\text{th}}$  item or  $i^{\text{th}}$  data item. You can think of this as we have pairs, we have features and we have labels and we have  $n$  such objects or  $n$  such items in the data. This is the basic information about data. Now, let us first focus on labels and then we will come back to the features. Now, what happens is depending on the label we get different types of machine learning problems.

(Refer Slide Time: 22:29)



We just saw that there are two types of label, in case of handwritten digit recognition we had labels which were discrete quantities labels were one of the ten digits 0 to 9. In case of housing price prediction the label was more of a continuous quantity. For example housing price the price can be any real number. So, in housing price prediction we had label that was a continuous quantity or continuous number and in case of handwritten digit recognition we had a discrete quantity.

We can have before even getting into the type of label, we first check whether the label is present or not. Label can either represent or absent. If label is present we call the corresponding machine learning algorithm or technique as supervised learning algorithm. If the label is absent we have unsupervised learning techniques or unsupervised learning models.

What are some of the examples of supervised learning? Handwritten digit recognition where the input has the images as well as labels is an example of supervised learning. Housings by housing price prediction where we have a triple of the house and the price of the house is also an example of supervised learning. What are some of the examples of unsupervised learning, can you think of some of them? Ok, one of them could be if I want to group students based on their attributes I do not really know what are the classes of the students good student, bad student, average student. I do not have any of those

ideas. So, label is essentially absent so all that I want to do is I want to group students based on some of their attribute. This is an example of unsupervised learning.

If label is present now we have further classification of the supervised learning models. So, let us use different colour, so labels can either be discrete or continuous. If your discrete label we call that supervised learning problem as a classification problem, you have classification problem and if label is a continuous number we call those supervised learning problems as regression problems. We have seen they we have seen the examples of a classification problem and regression problem. The handwritten digit recognition that you are trying to solve is an example of a classification problem, whereas housing price prediction is an example of a regression problem.

Now, that we know different type of machine learning algorithms based on availability and non-availability of labels, we go back and try to understand another component of the data which is a feature. We can also have features of different types, the simplest features to handle are numeric features. Numeric features we essentially have numbers. For example, we had a number of bedrooms in the housing price prediction problem and we also had area of the house in square feet that is also an example of a numeric feature, number of bedrooms, area these are examples of numeric features.

We also have another type of feature which is called as categorical feature. In this case, we normally have values coming from some finite set. For example the name of the city we do not have the numeric representation of name of the city, but we get multiple strings in the categorical attribute called name of the city Mumbai, Chennai, Pune, Bangalore can some of the examples of the city and city makes a categorical attribute.


The second example of categorical attribute could be the colour red, orange, green. These are also categorical attributes because we cannot represent them numerically and these attributes come from some kind of a finite set. We have to understand that when we are doing machine learning and when we are trying to build models we only can input the numbers. So now, what we need to do is, we need to take these categorical attributes and convert them into numbers. Let us see how we can achieve it.



(Refer Slide Time: 28:35)

City : {"Mumbai", "Delhi", "Chennai"}

	$f\_Mumbai$	$f\_Delhi$	$f\_Chennai$	
Mumbai	1	0	0	One-hot-encoding
Delhi	0	1	0	
Chennai	0	0	1	



And for simplicity let us say there are only three cities Mumbai, Delhi and Chennai in our data set. So, one way in which we can represent these categorical attributes is in form of one hot encoding, let us try to understand what one hot encoding is. So, instead of using city as a single feature we say that there are three features; one corresponding to Mumbai it is called is as  $f\_Mumbai$ , one corresponding to Delhi  $f\_Delhi$  and one corresponds to Chennai  $f\_Chennai$ .

So, instead of having a single feature on the city we converted that into a representation where we have three features and whenever the city is Mumbai we switch on the corresponding features to the city over here. So, in this case we put 1 corresponding the feature for Mumbai which is  $f\_Mumbai$  and 0 in other cities. Similarly, if we get Delhi as the city we put 1 in the column of Delhi and everything else will be 0. Similarly, if we have Chennai as a city we put 1 only in case of Chennai.

You can see that this is called as one hot encoding. One hot encoding is one way of encoding the categorical features. The other way of encoding categorical features could be based on hashing or embedding. We are going to look at this advanced way of encoding in later in the course.

So now, that you know the basic terminology of machine learning around data features labels and you also know how machine learning is different from writing traditional

computer programs. We will continue this exploration and understand more machine learning terminologies like model, training and lot more other terms.