Inverse Methods in Heat Transfer
Prof. Balaji Srinivasan
Department of Mechanical Engineering
Indian Institute of Technology, Madras

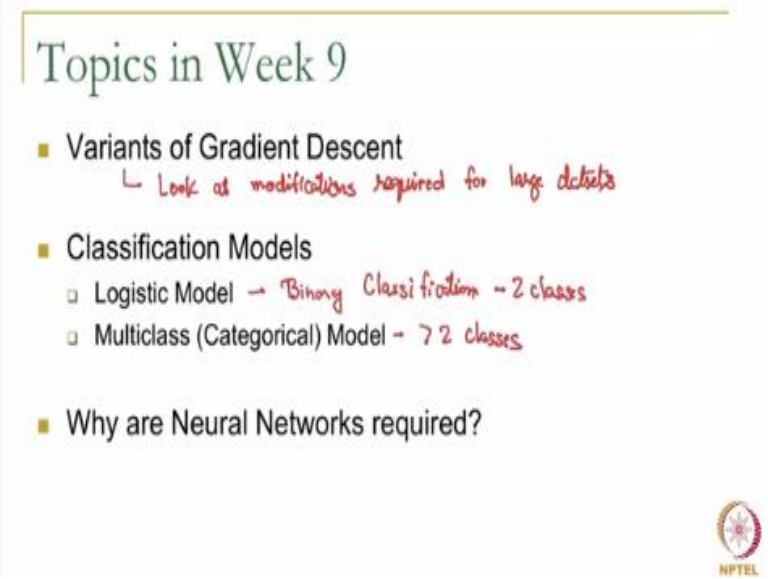
Lecture - 46
Introduction to Week 9 - From Linear Models to Neural Networks

Welcome back. This is week 9 of inverse methods in heat transfer. In the last week we looked at a brief introduction to machine learning and saw how machine learning could be essentially seen as a subset of inverse methods in general. Specifically inverse methods that depend directly on data rather than on the physics. Now so far, all the examples that we have been seeing are examples of what we would call regression.

Now machine learning has an additional thing that it is able to do and we will discuss this in greater detail both this week as well as next week. but this is called classification. So, that is basically the set of algorithms we will be looking at in this week. We will also see this week sort of a transition from what takes us from linear or even simple non-linear models to the models that we will discuss in the next couple of weeks which are known as neural networks.

Now this natural transition comes via something called a classification algorithm. I discussed this briefly last week. So, let us look at the topics that we will be covering in this week and I will also tell you briefly how linear models of the small modification at the start looking like neural networks.

(Refer Slide Time: 01:47)



So, the topics we will be looking at this week are variants of gradient descent. When I mean variance, this has several technical meanings within the machine learning literature. I will be looking at a very small version of this. by variance I mean variants that look at modifications required for large data sets. So, when I say large data sets what I mean is that when you typically look at methods that we use whether it was the normal equations method or whether it was a Gauss, Newton etcetera.

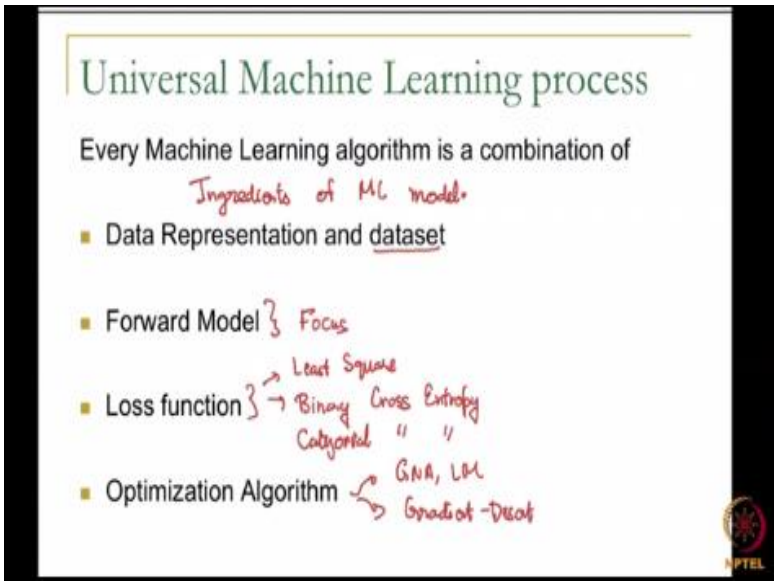
All these methods the examples we did in class during these lectures were all with very few data points five or six, of course that was meant to be for convenience. Even otherwise typically when we build a typical model or an inverse problem the number of sensors is not very high at least within heat transfer. However, when you have machine learning you actually have a large copious amount of data.

So, in such cases the simple gradient descent method that I covered earlier on in week three or four of this course, that is not sufficient. You have to do some variations within that and we will see these small variations within this in the next video. The next and the main topic of this week would be classification models. There is something called the logistic model which is for binary classification.

What is meant by that is, in case you want to decide something like cancer or non-cancer, is this chip going to blow up or not depending on what about the amount of power we provide is, such things are called binary classifications this or that basically it means two classes whereas multi-class or a categorical model means more than two classes. So, we will discuss these cases in greater detail later this week.

These applications of machine learning and I will also mention how once again this is an example of inverse modelling. Now finally we will talk about why are neural networks required and why we cannot stop with the basic simple regression models we looked at so far and that is the plan for this week.

(Refer Slide Time: 04:14)



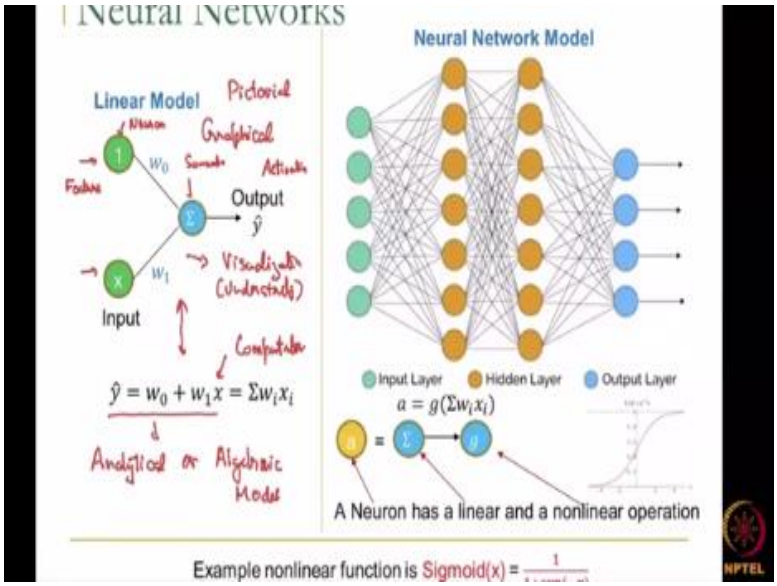
Now there is a universal machine learning process that you will notice within everything that we discussed this week. I had briefly mentioned this in the previous week also. but the basic ingredients of any ML model are these four. So, you get new machine learning algorithms by simply changing various portions of this recipe. So, for the first is data representation which I talked about a little bit in the last week which is if you have an image how do you represent it.

If you have a number a set of numbers, how do you represent it and of course the data set that we collected. This is one important ingredient of an ML algorithm, how you do this is an important thing. Next of course is a forward model where we will focus on. Next as you will see this time is

loss function typically, we used to use least square so far, for our objective function. But it will turn out that there are other loss functions like binary cross entropy is a loss and function that we discussed this week.

Or a general cross entropy or what can be called categorical cross entropy, we will discuss that this week. Finally, of course the optimization algorithm that we use, so, we have seen at least a couple, we have seen Gauss Newton, we have seen Levenberg Marquardt, we have also seen gradient descent. There are a few others that exist commonly within a machine learning algorithm. But we will keep our focus on these because we are good enough for most of our purposes.

(Refer Slide Time: 06:05)



Now just a brief look ahead on to what I mean by a neural network and how it is related to the linear model that we have already seen. So, you might already be sick and tired of the model on the left, but I want to show a new perspective on this model. So, this is something that we had seen already. We have this simple model of $\hat{y} = w_0 + w_1 x$. But this model is essentially an analytical or algebraic model.

So, let us call it that now this picture is very central to what we will do later on in not the next week, but the week after that which is called physics informed neural networks. So, I want to introduce you to this concept right now and then we will build upon this over the next couple of

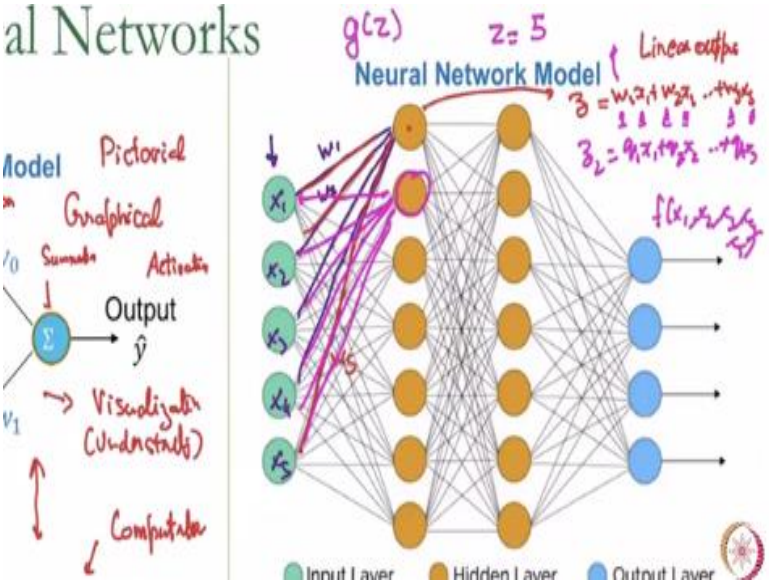
weeks. So, this is an algebraic or an analytical model. This on the other hand the model that you see on the top is the same model.

But this is a graphical or let us say a pictorial representation of this model. So, the important thing to remember is, when I draw a picture, it is exactly the same as writing an algebraic expression, except this is easier for us to visualize, this is easier for us to compute. So, this is what we use for computation, this is typically what we use for visualization or understanding. So, let us look at this model in a little bit of more detail.

So, when I say $w_0 + w_1 x$, I can say that one is one feature, of course it is a constant feature and that feature or this is I will call this so-called neuron. Neuron nothing is nothing but a computational unit, there is a number sitting there in this case one. Now this one is multiplied by w_0 and it comes here, similarly this x is multiplied by w_1 and comes here and you sum it up, this simply is a summation.

And the output which we will call an activation or you can call it output in this case is $\hat{y}$. So, $\hat{y} = w_0 + w_1 x$ that is all this feature is same. But notice that is exactly what this expression is also saying. So, the picture and this are one to one equivalent so these are equivalent. Now let us come to a more complex picture like this.

(Refer Slide Time: 09:06)



This as it turns out is a neural network model and as we will see from later on this week, all models can be written in this pictorial format. All models that we will be using can be written in this pictorial model pictorial format and all models essentially that we will be discussing in the next few weeks are all special cases of neural networks. So, one can think of a linear model as a special case of a neural network model.

So, let us come back here and let us say we have some numbers here x_1, x_2 let us say I had five inputs or I had an input with of vector of size 5. Now I can now think of this region feeding into some neuron what this neuron means etcetera we will discuss later. But for now, just think of a calculation where x_1 is multiplied by w_1, x_2 is multiplied by w_2 so on and so forth and x_5 is let us say multiplied by w_5 this one.

Now this here I will say something like $z = w_1x_1 + w_2x_2 + \cdots + w_5x_5$ and that would be a linear output very simple. So, you take all these inputs multiply them with w_1, w_2 etcetera and you get a linear output at the end. Now that looks like a very simple model now, I could simply similarly say something like this and let us call this z_2 would be now you have another weight. So, I will call it something else let us say $q_1x_1 + q_2x_2 + \cdots + q_5x_5$ that would be the output of this neuron.

So, similarly you can now think of outputs from each one of these multiplied by different weights coming out from each one here. Repeat that whatever output you get here multiplied by some other weights you get some output. Here as you can see at the end you are going to get some function of x_1, x_2, x_3, x_4, x_5, but there will be a lot of parameters w_1, w_2, w_3, w_4, w_5 etcetera. Now all these basically are examples of neural networks.

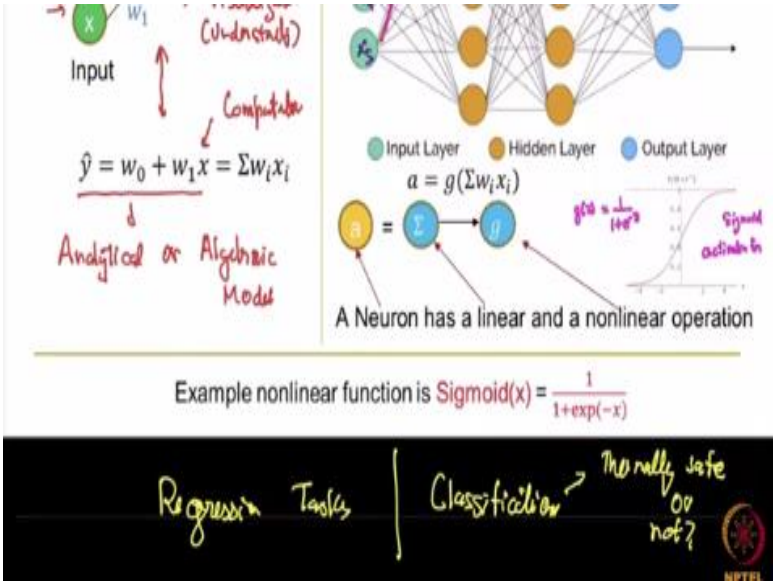
Though proper neural networks have one more twist. I will show you that but as of now all you need to imagine is that if I give some x_1, x_2, x_3, x_4, x_5, even though there is a picture here you know that if I give you all these parameters in the middle you can get some number out from each one of these four. Now this thing here in a neural network is called an input layer as marked here. So, this is an input limit this thing here is called a hidden layer and this thing here is called an output layer.

This hidden layer we do not know what it means and I will discuss this in more detail in the next week. But we really do not know what is what the meaning of these terms are, but we know these are just calculations. They are just internal calculations that we are doing. Remember we always think of a neural network as having some input and having some output and this thing here is a black box, it does some calculation and we are just writing what this calculation is.

In a neural network however unlike a linear model where all you do is some things up you add one more thing. So, for example a neuron that is the output here has two steps, it has a linear step which is exactly the same as in linear model, but after that you run it through a non-linearity. What is this non-linearity? It is just when non-linear operation. So, suppose at the output of this step, you got something like five, so let us say w_1, w_2, w_3, w_4, w_5 are all equal to 1 and x_1, x_2, x_3, x_4, x_5 are also equal to 1.

So, the output here will be Z = 5 and then what you give is not just five but a function of 5 for a function of Z so you will give out $g(5)$. What is this function?

(Refer Slide Time: 13:53)



A typical function is a function of this sort. So, something like $g(z) = \frac{1}{1+e^{-z}}$ that is a complicated looking function this is called a sigmoid function. I will briefly explain the reason for the sigmoid, but we will see this in greater detail later this week also. So, the reason for the sigmoid function is

very simple. It was supposed to simulate what happens in the brain. So, the idea is this; imagine that you are seeing or you are sleeping.

So, you are sleeping and somebody is shaking you to wake up. So, you are you know well lost within your dreams and your brain or at least a few neurons which are supposed to wake you up are asleep. Now the person shakes, shakes, shakes. Now one model of you waking up is like this and it is right here that is asleep, awake. Now this is not a continuous or a smooth function and later on as you can see for as we have been doing for gradients we need to differentiate.

So, we have a smoother model, the smoother model is like this 0, 1. So, that is this model. It is as if somebody shook, shook, shook and after a certain amount of activation you actually woke up. So, this is also called an activation function. But basically, mathematically all that activation function is on the simple linear model. Remember if you had multiple inputs all those would have added up to on the simple linear model all you do is you put a function of this sort that is it.

So, you take a linear combination, so add all these up, come here, put some non-linear functions. Suppose that non-linear function was Z square then g of Z would have become 25. So, 25 will come out rather than just 5 from here. Similarly, this could be another non-linear function, this could be another non-linear function, put all these together, then again repeat. So, all a neural network does is you take some input linear combination non-linearity.

Linear combination non-linearity, linear combination non-linearity and you then have an output. Remarkably this simple idea of chaining together linear operations or non-linearity is good enough to do not only one task but any task. So, a typical type of task so is making a prediction. so that would be a regression task but these we have seen. So, far, if temperature is so such and such thing at these locations what is the temperature in the middle, what is the heat flux in the middle, that is our recreational task.

But another task is a classification task. So, if a neural network is good enough to do anything it should be good enough to do classification also. Classification says cancer or no cancer, spam or not spam, is this thermally safe or not. So, you are designing a mobile or you are designing a

chipset or will ask is this thermally safe or not. So, such tasks are called classification tasks and we will see that atom as a prototype, you will start needing this sigmoid function in order to achieve this task.

And a simple linear function is generally not sufficient for it. So, we will start looking at these ideas later on this week. The primary purpose of this week is to naturally segue, to naturally move and smoothly move from a linear model to a neural network model. In the next video, I will show you why we need a different variance of gradient descent in order for optimization and after that we will start looking at this idea of a sigmoid. So, I will see you in the next video, thank you.