

Machine Learning For Engineering and Science Application
Professor Dr. Balaji Srinivasan
Department of Mechanical Engineering
Indian Institute of Technology Madras
Linear Regression Least Square Gradient Descent

In this video we will be looking at some details of the linear regression we had seen a simple plot obtained through MATLAB for a linear fit, a quadratic fit and a cubic fit the last video we will look at some details of how to do this please pay attention to the process that is been shown here because this is essentially the process that we will be repeating for almost all of deep learning specially for the deep learning module as I said the paradigm is set by what we do for a simple linear fit and we will just continuing that for quadratic cubic and then neural network etcetera, and even for classification problems.

(Refer Slide Time: 00:52)

Regression example

Our example was as follows

Table 1 Coefficient of thermal expansion versus temperature for steel.

Temperature, T °F	Coefficient of thermal expansion, α $\times 10^{-6}$
95	6.470
80	6.360
60	6.240
40	6.120
20	6.000
0	5.880
-20	5.720
-40	5.580
-60	5.430
-80	5.280
-100	5.090
-120	4.910
-140	4.720
-160	4.520
-180	4.300
-200	4.080
-240	3.830
-280	3.580
-300	3.370
-320	2.960
-340	2.450

Ground Truth

We will discuss how to come up with these in the coming video(s)

So here is the example the we saw last time, last time we looked at temperature versus coefficient of thermal expansion and we had all this data on the X axis let us call this X and we have all this data on the Y axis let us call this Y so this Y as I said earlier is call the ground truth this is basically the experimental truth or reality that is available to us,

What we would like to know is what happens in between that is the classic regression problem we would like a fit for this data and we saw three different kinds of fit last time one of the fit is

was a linear fit you see that actually this line which we can call \hat{Y} which is a function of X called the hypothesis function of X in fact had hypothesis this two W_0 plus $W_1 X$.


So this is \hat{Y} versus this is Y for the same X you have the real prediction and you also have the hypothesized prediction so we saw that there is a difference between the two but non the less overall trend is captured by the hypothesis so that us one of the thing that we saw last time we also saw that if you put a quadratic fit in this case let say this is \hat{Y} so quadratic shown in red here that a little bit better than linear in fact it is reasonably better then linear and cubic which is merely better then quadratic almost in distinguishable so we had all this different fit is that we had for the same set of data why do all this fit differ because our H of X or module for what Y is like which we call \hat{Y} is actually different for each of this cases so we had linear quadratic cubic so for example for quadratic we has $W_1 X$ plus $W_2 X$ square so on and so forth.

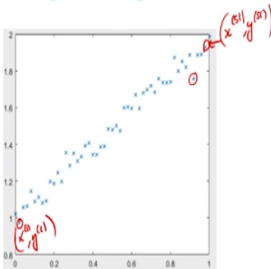
So what we will see in the next video is how do you actually come up with the coefficients so I just use some in build MATLAB function now we are going to do it from scratch in the coming videos.

(Refer Slide Time: 03:09)

The general univariate linear regression problem

Input (x)	Output (y)
$x^{(1)}$	$y^{(1)}$
$x^{(2)}$	$y^{(2)}$
...	...
$x^{(m)}$	$y^{(m)}$






We start with the case where there is a single input and a single output

$(x^{(i)}, y^{(i)})$: i^{th} "example" of (input,output) set

m : Number of examples or data points



So let us look at this general problem just like the previous problem you have some X and you have some Y you have some scalar X and you have some scalar Y and each of this data points is we can lobular task data point 1 data point 2 so and so forth, and let us say there are M such data points so we have single input like the temperature and single out put like the thermal coefficient

so let say we take this data pack $X_i Y_i$ and we called it the i^{th} example, why example because later on we will see for images I will say this is an example of cat, this is an example of dog, so each images also call an example this is simply machine learning terminology you can called it data point.

So i^{th} example simply means i^{th} data point so in this figure let say I think there are approximately 51 more points so you could start with $X_1 Y_1$ and go up till something like $X_{51} Y_{51}$, something that is see have all this points and what we you would to see is which hypothesis function fit is this the best, remember this terminology will be using a lot of letters and within our course M simply means the number of data points or number of example that you have please remember this.

(Refer Slide Time: 04:31)

The general univariate linear regression problem

Input (x)	Output (y)	Model (\hat{y})
$x^{(1)}$	$y^{(1)}$	$\hat{y}^{(1)}$
$x^{(2)}$	$y^{(2)}$	$\hat{y}^{(2)}$
...
$x^{(m)}$	$y^{(m)}$	$\hat{y}^{(m)}$

We now introduce our first model hypothesis – Linear Model

$$\hat{y} = h(x) = w_0 + w_1 x \quad \leftarrow \text{Form}$$

There are infinite w_0, w_1 possibilities. Which do we choose?

For this, we define a cost function $J = \frac{1}{2m} \sum_i (y^{(i)} - \hat{y}^{(i)})^2$ Least Squares

Optimal w is the one that minimizes the above cost function

Called the **least squares** fit. The cost function is called **least mean square (LMS)**

Now let us look at how do you do the fit here the input output is actually the given data and \hat{Y} is our model above and beyond that the data is given so you have X_1 and Y_1 but your guessing what \hat{Y} should be for the given X remember like the example we gave X was the temperature Y was the thermal expansion coefficient this is the actual and you will guess something else, we are going to introduce our first very first for this course model hypothesis this is a simple very very trivial linear model but it is enormously powerful as you will shortly see in the next couple of videos.

Now there is only one question we have already fix the form of the function and as i said in the previous video we still have the parameters, the parameters are unknown what W_0 and W_1 should I fit now obviously for different choices of W_0W_1 even in each case even though in each case your hypothesis will look like a line it is going to look like a different line depending on what W_0W_1 you fixed.

So suppose somebody randomly give's some value of W_0W_1 here is the original data here is your hypothesis this is the data this is your hypothesis or the model based on the model parameter some model parameter (θ) (6:03) now it doesn't look like a very good fit intuitively it doesn't look like a good fit so somebody else another person gives us a slightly different model and this looks slightly better than this again we have an intuitive notion of what is better we will formalize this and this very slight so this looks better than this because all that has change here is W_0 and W_1 remember all three are still lines.

Now this one looks really good this one looks much better than this also much better than this so the question is, is there any way in which we can formalize or quantify remember this word in machine learning we are always looking at quantitative things we are looking at number so the machine only recognized numbers so is there any way in which we can quantify why this is better than this or this and the idea goes back to an old idea which we have which is a cost function.

So what is a cost function for this it is simply one simple number that will tell you how good the fit is, so how is it going to say that now for each point, let say I take this X there also Y there also a \hat{Y} so there is a difference between the two what do I do, I take difference between the two square them and then add it, so let us say this was Y_1 and this was \hat{Y}_1 this is $Y_1 - \hat{Y}_1$ and this is $(Y_1 - \hat{Y}_1)^2$ so for the same X you find out the Y and the corresponding \hat{Y} square it add all of them how many example do you have, you have Y equal to one two M .

So you have 51 such examples these some of all this squares this is basically some of the square of the errors and you say which ever line or whichever choice of W or W_1 minimizes this total error I am happy with that notice that no line is going to fit all of this perfectly you cannot drive this Z_0 because no line will fit all points but overall, you know overall little sort of kind of split the data so that you do not go too far away from the line.

So this is how we achieve our optimal W , so we say that the optimal W , so you can now notice it has now become optimization problem and optimal W_1 is the one which minimizes this net cost function now couple of things I have put by M here this is arbitrary even if you remove this 1 by $2M$ the minimum will be the same but this M is often used because you would like mean of squared error for several reason 1 is of course to avoid some kind of over flow errors you some time just take a mean another thing is this two is also arbitrary, but it is put there just show that when it differentiate this function the two and this two will cancelled out, this fit is called the least squares fit so the W 's that we get at end of the process will be called least square coefficients and this cost function is sometime it is called the least mean square cost function or the mean square cost function some time it is called LMS.

(Refer Slide Time: 09:20)

Finding the linear regression coefficients Using gradient descent

We have m data points $(x^{(i)}, y^{(i)}) \quad i = 1, 2, \dots, m$

Guess $w = [w_0, w_1]$

For any given guess of w , we have the corresponding output

$$\hat{y}^{(i)} = w_0 + w_1 x^{(i)}$$

Calculate $J = \frac{1}{2m} \sum_{i=1}^m (y^{(i)} - w_0 - w_1 x^{(i)})^2$

Improve w by using

$$w = w - \alpha \nabla_w J$$

Stop when the stopping criterion is met

The final set of $w = [w_0, w_1]$ obtained are the regression coeffs.

But how do we calculate $\nabla_w J$?

Input (x)	Output (y)	Model (y-hat)
$x^{(1)}$	$y^{(1)}$	$\hat{y}^{(1)}$
$x^{(2)}$	$y^{(2)}$	$\hat{y}^{(2)}$
...
$x^{(m)}$	$y^{(m)}$	$\hat{y}^{(m)}$

Now that we have reduced our fitting problem to an optimization problem can we use gradient descent, which we discussed in the previous week so gradient descent we used as a general one box algorithm in order to find out minima and it turns out we can use gradient descent, how do we do it very simple idea again to start with an X some data point for a temperature that is given for an example guess some W , this W is remember true for all X run it through the hypothesis our Y hat was linear function W_0 plus $W_1 X$ the ground truth is already available, we just got a hypothesis because we guess the W .

Now there is going to be gap between Y and Y hat square it sum it that is going to give you the net cost of the coefficients that you have chosen remember this net cost is because we have chosen some W0 and W1 then find out gradient and improve your W by using gradient descent, so let see this again you have M data points let say 51 just is an example now for each of this data points you can get a corresponding Y hat provided I give you some guess for W0 and W1.

You have the corresponding output then you calculate the net cost function which is Y minus Y hat I, so YI minus Y hat I square will give you the net cost function and then you improve W by using gradient descent when do we stop you keep on doing this you has some stopping criteria, I gave you three different types of stopping criteria and we will see at least two of them and then an example shortly so if you used your stopping criteria it will stop the final results that you obtain for your W are actually your regression coefficient, so you can carry out this whole process but theoretically you have only one small catch how do you calculate this gradient of J with respect to W, So let us see that.

(Refer Slide Time: 11:48)

Calculating the least squares gradient

$$J = \frac{1}{2m} \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)})^2$$


$$= \frac{1}{2m} \sum_{i=1}^m (y^{(i)} - w_0 - w_1 x^{(i)})^2$$

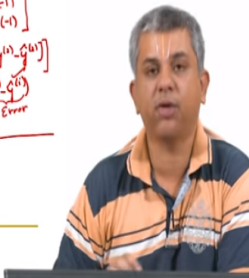
$\frac{\partial J}{\partial w_j} = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)}) x_j^{(i)}$

Claim: $\frac{\partial J}{\partial w_j} = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)}) x_j^{(i)}$

Proof: $\frac{\partial J}{\partial w_0} = \frac{1}{2m} \sum_{i=1}^m 2(y^{(i)} - \hat{y}^{(i)})(-1) = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)})$

$\frac{\partial J}{\partial w_1} = \frac{1}{2m} \sum_{i=1}^m 2(y^{(i)} - \hat{y}^{(i)})(-x^{(i)}) = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)}) x^{(i)}$





So our J as you saw on the last slide is submission of Y minus Y hat square, Y hat was W0 plus W1X so you get Y minus W not minus W1X square, and we want gradient of W we want gradient of J with respect W, notice that W is a vector so W is basically in our case W not W1, I will avoid that transposes similarly gradient of J with respect to W not which is nothing but del J del W not and del J del Y, I am just writing out this both the components of this vector equation

$\frac{\partial J}{\partial W}$ is $\frac{\partial J}{\partial W} - \alpha \frac{\partial J}{\partial W}$, now we want these two expressions, I will show you that they can be written in a compact form like this please do not pay too much attention to this before the derivation that the derivation is extremely straight forward actually, so let us take the simple case, let us take the case that M is equal to two.

I will just do the derivation for that and you can see that it easily extends to any number of N , so let us give a proof of this statement so suppose I want $\frac{\partial J}{\partial W}$ for the case M is equal to 2, I will keep the 1 by $2M$ here now what term I will have, I will have $Y_1 - W X_1$ square plus $Y_2 - W X_2$ square these are the only two terms that exist and now I have to take $\frac{\partial}{\partial W}$ of this, since it is a partial with respect to W not only these terms are actually dependent on W , so how do we do this 1 by $2M$ take this term this is the same as two times $Y_1 - W X_1$ multiplied by derivative of this term with respect to W which is -1 plus two times $Y_2 - W X_2$, the two's cancelled out which is why we had two in the definition in the first case we are going to get 1 by M , I will take the minus out you will see Y_1 minus this is nothing but our hypothesis function \hat{Y}_1 plus Y_2 minus \hat{Y}_2 .

So this means that $\frac{\partial J}{\partial W}$ is $-\frac{1}{M} \sum (Y_i - \hat{Y}_i)$, I equal to 1 to M you can see easily that this will continue for any M , Now what is this term, this is nothing but the error, so what it tells us is that the first component of the gradient is simply the sum or the mean error it is not means square error it is simply mean error also notice you can write \hat{Y} equal to $W X$ as $W X_0 + W_1 X_1$ and you surely we will call this X_0 as X not, you will see the power of this notation later because of this when I said $\frac{\partial J}{\partial W}$ I am going to say X not I , where X not is nothing but 1 , it just lets me write this whole expression compactly.

Let us look at $\frac{\partial J}{\partial W}$, I will derive this in slightly different fashion just to give you another tool near tool set 1 by $2M$, $\frac{\partial}{\partial W} \sum_{i=1}^M (Y_i - \hat{Y}_i)^2$, let us deal with this in this form it is self and see if we can do some algebra quickly, so this is equal to $\frac{1}{2M} \sum_{i=1}^M 2(Y_i - \hat{Y}_i) \cdot (-1)$ multiplied by $\frac{\partial \hat{Y}_i}{\partial W}$.

So what did I do here instead of expanding \hat{Y} fully into W not plus $W1X$ which I did last time I am just differentiating this directly so derivative of this term with respect to $W1$ is nothing but 2 times this term multiplied by derivative of this term simple chain grown, so I am just going to use this here, so the 2 and 2 cancelled out once again we are going to have a minus 1 over M sigma I equal to 1 to M , YI minus \hat{Y} I multiplied by $\text{del } \hat{Y} \text{ hat } I$ by $\text{del } W1$ now what is $\text{del } \hat{Y} \text{ hat } I$ by $\text{del } W1$, remember \hat{Y} I is W not plus $W1 X$ I which means $\text{del } \hat{Y} \text{ hat } I$ by $\text{del } W1$ is simply equal to X I so this gives us minus 1 , I equal to 1 to M , YI minus \hat{Y} I times X I .

So let us go back to this term here so you see this J here and J here what does this denote $\text{del } J$ $\text{del } W1$ is equal to YI correct minus \hat{Y} I correct times X I is also correct accept I will have X I I here X I is nothing but X power 1 which is equal to X , that simply denotation similarly if I take $\text{del } J$ $\text{del } W$ not, I will get YI is correct minus \hat{Y} I which is also correct multiplied by X not I , X not I is simply 1, so it is just compact notation if you are not comfortable with this that is fine, you can simply write both this terms individually and say that $\text{del } J$ $\text{del } W$ not is minus 1 by M times sigma of YI minus \hat{Y} I similarly $\text{del } J$ $\text{del } W1$ is minus 1 over M sigma of Y minus \hat{Y} I hat multiplied by X .

(Refer Slide Time: 20:04)

Steps of the linear regression procedure

1. Decide on α, ϵ and stopping criterion
2. Make an initial guess for the weight vector $w = w^{(0)}$
3. Calculate $w^{(k+1)} = w^{(k)} - \frac{\alpha}{2m} \sum_i (y^{(i)} - \hat{y}^{(i)}) x^{(i)}$
4. Calculate stopping criterion
 1. If condition satisfied, stop
 2. If not satisfied, go to Step 3

Let us now see a code to implement this



So what are the steps of the linear regression procedure we first decide on our learning rate remember learning rate is required for gradient descent and we also have to decide on what our stopping criterion is we will make an initial guess for the weight vector then systematically you

will calculate the next iteration, now that you have one weight vector you have one guess for W not and W_1 we will make another guess, how do you guess this by using the formula that we just derived and once you update your W you calculate your stopping criteria if this works out you stop, if it is not satisfied you go back here you calculate once more you keep on stepping through the radius.

So in the next video we will actually see a code to implement this and hopefully all of this things will come together very nicely this is another reasons why we insist on doing the code because in theory you might understand something it is only when you actually see it practically implemented into a code that things become clearer we will be using as we had declared earlier will be using an example through a MATLAB code all of you are welcome to use whatever programming language that you would like to see but MATLAB is usually the easiest to explain things as well as visualize things nicely so we will see that in the next video, thank you.