Report

Curve Fitting on V-T Data

February 22, 2021

Problem Statement

To explore curve fitting (best fit) to each volume-time curve and extract coefficients for each curve.

Data

We have Volume data corresponding to 150 time points(1,2,..150) of 100 patients. If we plot the V-T curve for all 100 patients, then it will look like the following -

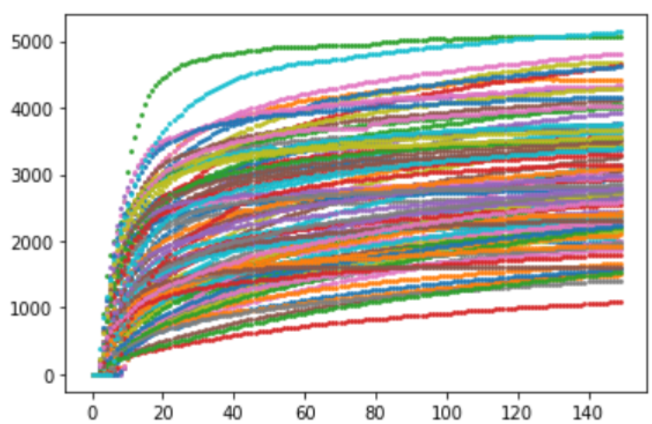

Figure 1: V-T Curve of 100 Patients

Visualizing the data helps us to understand which type of function may fit on the data.

Curve Fitting using $f(x) = a\ln(|x^2+bx+c|) + \frac{2(d-ab)}{\sqrt{4c-b^2}} tan^{-1}\frac{2x+b}{\sqrt{4c-b^2}} + e$

From the plot we can say that a logistic function, $g(x) = \frac{L}{1+e^{-k(x-x_0)}}$ may work. But the logistic function didn't work well even after normalizing the data. It is also clear from the plot that the original function is an increasing function. So to know how the function is increasing and to get an idea about the curve, the first difference(derivative) of V-T data of different patients was plotted. The plots look like the shape of the function ($f'(x)$) mentioned later. One example of how the derivative function fits on the first difference of V-T data of one patient is shown in Figure 2. So after guessing the derivative function from the plot of first difference of V-T data, we integrate it to get the function which may work on the actual data.

Derivative Function - $f'(x) = \frac{ax+d}{x^2+bx+c}$

By integrating the function we get -

$\int f'(x)dx = f(x) = a\ln(|x^2+bx+c|) + \frac{2(d-ab)}{\sqrt{4c-b^2}} tan^{-1}\frac{2x+b}{\sqrt{4c-b^2}} + e$

The function $f(x)$ worked well for most of the patients but didn't fit on some patients' data. So taking derivative can help us to identify the curve. If we still can't guess the function then we can keep taking the derivative few times to get a shape of some known function. Then we have to integrate it till we get the original function. This is one of the approach which can be used.

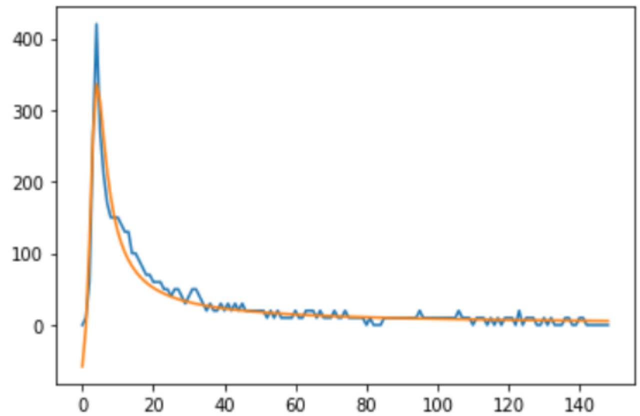

Figure 2: First difference of V-T data of one patient fitted with $f'(x)$

Suppose the data can be fit by a polynomial of degree 3. Then if we keep on taking derivative, after 3rd derivative we will get a shape which will look like a constant function. In our case if we keep taking differences and plot them, after 6-7 times the plot looks like a constant function. So Polynomial regression can be a good choice.

Curve Fitting using Polynomial Regression

We can see this curve fitting problem as a regression problem where we have a dependent variable, Volume and an independent variable, Time-point. Here our target is to generate a curve which can capture the pattern of the data.From the graphs we can see that the relationship between these two variables is not linear. So here we can not use linear regression Model.

$$y = w_0 + w_1 x$$

So to capture the non-linearity of data, we add higher power of the dependent variable as new features in the model. So the model looks like the following -

$$y = w_0 + w_1 x + w_2 x^2 + ... + w_n x^n$$

Here n is the degree of the polynomial. Polynomial of degree 7 is used to fit the curves in the V-T data.

Curve Fitting using Neural Network

We know that Neural Networks can be used as Function Approximation Algorithms as it can learn any complicated function. Here the Architechture of the network is important. We can use as many layers and as many nodes in a layer as we want but it will increase the number of parameters which will be learned during backpropagation. We are interested on these learnable parameters.

Here we want the best posssible curve fit but at the same time using complicated architecture(same as involving many paramters) may effect our goal. So it is preferable to use Vanilla Neural Network Architecture which won't involve large no of parameters but will be able to approximate the curve. We need to train the model for large no of epochs for the parameter values to converge. Even if we use slightly complicated Neural Network architecture, we need to do some analysis on the learned coefficients so that any V-T data can be differentiated based on the coefficients.

Architecture of the Network:

Input Layer : No of Nodes - 1
1 Hidden Layer : No of Nodes - 5
Output Layer : No of Nodes - 1
Activation Function in Hidden Layer : Sigmoid
Loss Function : Mean Square Error
Optimizer : Adam

Results:

Average MAE using the function f(x) - 14.45
Average MAE using Polynomial Regression - 22.96
Average MAE using Vanilla NN - 24.29
Variance of errors for the 3 methods are 193.20, 245.04, 205.99 respectively.

Though the average and variance of Mean Absolute Errors are the least in case of first method, the function f(x) didn't fit on some V-T data. The average MAE and variance of MAEs are little high in case of other two methods compared to the first method but both methods(Method 2 & 3) worked for all V-T data. Average MAE is almost similar in both these methods but variance of errors is slightly smaller in case of Neural Network. But again the no of coefficients involved in case of Neural Network is 16 whereas it is 8 in case of Polynomial Regression. Looking at the coefficient plot for all data, we can say that if we want to differentiate the curve based on the coefficients then it would be less difficult in case of polynomial regression whereas some analysis need to be done on the coefficients in case of Neural Network.