CURVE FITTING FOR COARSE DATA USING ARTIFICIAL NEURAL NETWORK

BALASUBRAMANYAM C
Atria Institute of Technology
Department of Mechanical Engineering
001B, DS max, 1st main, Best county-II, MS Palya, Bangalore- 560097
INDIA
balacbs@gmail.com

AJAY M. S, SPANDANA K. R, AMOGH B. SHETTY, SEETHARAMU K.N
PES Institute of Technology
Department of Mechanical Engineering
100ft ring road, BSK III stage, Bangalore-560085
INDIA
ms.ajay5@gmail.com, amoghbshetty@yahoo.com, spandi93@gmail.com
knseetharamu@yahoo.com

Abstract: This paper demonstrates the capability of curve fitting using Artificial Neural Network (ANN), not only for a moderate set of input data but also for a coarse set of input. When appropriate number of neurons is chosen for the training purpose, accurate graphs can be obtained, despite having a coarse data. The effect of number of neurons used for curve fitting and the accuracy obtained is also studied. This aspect of ANN has been illustrated through 2 examples, Weibull distribution and another complex sinusoidal system. This curve fitting technique has been applied to a real world problem i.e. mechanism of a deep drawing press, for both slider displacement and slider velocity.

Key–Words: curve fitting, ANN, accurate, coarse data, best fit, neurons

1 Introduction

Curve fitting is a process of generating a curve which best represents the characteristics of a system using the input data set. It is the basis of any analytical, comparative or growth related statistics. The objective [1] of curve fitting is to select parameter values which minimize the total error over the set of data points being considered. Curve fitting finds its applications in image processing, company growth related graphs, prediction statistics, finance, pattern recognition and many more.

Earlier, due to insufficient methods of exact curve fitting, a system characterization was being reduced to that of a straight line equation for simplicity, using minimum data to develop a plot. For example, an exponential or a power law expression was converted into a semi-log or a log-log graph respectively, to fit an approximate straight line. Polynomial expressions were introduced with various constants to obtain better representations, yet this method could not suffice the accuracy required.

But, in the present day scenario, many curve fitting techniques have come into picture, which are capable of fitting highly complex functions and non-linear plots, out of which some require less inputs and others may need a huge data entry. Some of the attempts include [1 -12]: According to mark and David (1985), Global curve fitting using rational fraction polynomial method [2] can handle very noisy data because it works on the principle of least squared error formulation. But if the parameters are varying by a large amount, accurate results may not be obtained. Genetic algorithm [1] approach is used by by Gulsen et al. (1995) for curve fitting which can solve optimization problems, wherein each solution is encoded as a vector of real valued coefficients. Curve fitting with Bezier cubics [3] by Lejin and Hao (1996) uses piecewise approximation approach consisting of cubic curve pieces, connected end to end, to get a good representation of an image outline. But, this method does not work in fitting the curve when there are many

points of inflexions. Ilaria et al. (2001) Bayesian method [4] uses probability and probability distribution concepts for curve fitting rather than using the binary logic. Aris Spanos [5] discusses the curve-fitting problem using the reliability of inductive inference as a primary criterion for the fittest curve. Frisken [6] proposes a method for fitting a piecewise parametric curve to a sequence of digitized points such as those acquired computer mouse or digitizing pen. The method uses vector distance fields, to represent the intended input path of the digitized points in order to achieve its quality and performance. The local linear regression method [7], by Desmet and Gijbels (2011) discusses the procedure of parameter selection considering target functions with an unknown number of irregularities of unknown type. The method of arbitrage smoothing [8], by Paul and Kees (2009) is an empirical method which gradually imposes arbitrage restrictions in fitting a sequence of yield curves. It controls arbitrage errors that arise while fitting curves.

According to Christopher and Roach (1992) Artificial neural network (ANN) [9] is typically much faster than any other conventional iterative approach. Works of Barbosa et al., (2006) show that ANN is capable of fitting calibration curves to a multivariate system [10] and to describe physical responses of an input to a UAV system [11] with a good detail. In paper [13] the problem with the introduction of a large quantity of wind generators on the electric grid is presented. A method based in artificial neural networks (ANN) is used to predict the average hourly wind speed. Mohammadi et al.'s [14] shows an approach developed for prediction deformation of upsetting processes. The approach combines the finite element method and Neural Network to view the resultant deformation changes in upsetting. The performances of the NARX model [15] are verified for several types of chaotic or fractal time series applied as input for neural network, in relation with the number of neurons, the training algorithms and the dimensions of his embedded memory ANN is a tool used for computing complex relations between the input and the output parameters or entries. Using the input values, the artificial neurons train themselves to produce the initially given output values, thereby developing a pattern. The connection so established, is then applied on another set of interpolate values to give the desired output to the user.

The above methods have succeeded in generating accurate plots, provided the input data is sufficient. But none have discussed the issue of how number of inputs affect curve fitting. The accuracy of the plot has been concentrated upon, whereas, minimizing the input parameters so as to achieve the same, is untouched (has not been worked on).

In this paper, we are showcasing the capability of ANN to fit accurate curves even with coarse data input. And meanwhile, develop a pattern between the number of input entries and the number of neurons to be used. This saves the user from entering large set of data to generate an accurate graph.

2 Methodology

In order to illustrate the above, we have used the Weibull distribution curve (section 2.1) and a complex trigonometric sinusoidal system (section 2.2), and arrived at a limit to the minimum number of input entries i.e. a coarse data that yields an accurate trend. Making a similar application to the mechanism used for deep drawing press, a complex real world problem (section 3), ANN proves to be a simple method of curve fitting.

2.1 Weibull Distribution Curve

Weibull Distribution is a continuous probability distribution, found most likely in the field of probability theory and statistics. It is far more likely to be discussed and used in works dealing with experimental results, particularly reliability. It is a chameleon distribution, asymmetrical, containing a good approximation of both normal as well as an exact representation of exponential distribution. There are 2 forms of Weibull, the three- parameter and two-parameter($x_0 = 0$) distributions. We have used the three- parameter Weibull for our example. The expression is:

$$R(x) = e^{-(x-x_0)/(t-x_0)^b}, x > x_0,$$

where,

$x_0 =$ guaranteed value of x ($x_0 > 0$)
$t = a$ characteristic value ($t > x_0$)
$b = a$ shape parameter ($b > 0$)

The variate serves a role similar to the mean and represents a value of x below which lie below 63.2 % of the observations. The shape parameter b controls the skewness of the distribution. In the range $3.3 < b < 3.5$, approximate symmetry is obtained along with a good approximation to the normal distribution.

To find the probability function, we note that

$$F(x) = [1 - R(x)]$$

Differentiating the above with respect to x and solving it results in the final expression:

$$\frac{\partial F}{\partial x} = \frac{b}{(t-x_0)}\left[\frac{(x-x_0)}{(t-x_0)}\right]^{(b-1)} e^{-\frac{(x-x_0)}{(t-x_0)^b}}$$

The nature of the plot for the above expression varies with the value b. For larger values of b, the curve skews to the right and for smaller values it skews to the left. In our analysis, the value assumed for the different constants are:

$$x_0 = 0.98; \qquad t = 2.53; \qquad b = 1.7;$$

For the above assumed values, we have developed a curve and used the concept of ANN to fit a curve for a given set of coarse input data(x).

2.1.1 ANN Graph for coarse data

In the Figure 1, the curve is obtained as a result of the entered coarse set of points (15 points). The data is as follows: (x, y)

X axis:	1.00	1.08	1.15	1.46	1.55
Y axis:	0.0522	0.1595	0.2281	0.4213	0.4536
X axis:	2.05	2.20	2.6	3.10	3.5
Y axis:	0.4968	0.4767	0.3844	0.2487	0.1569
X axis:	3.84	4.0	4.40	4.68	5.0
Y axis:	0.0991	0.0782	0.0410	0.025	0.0136

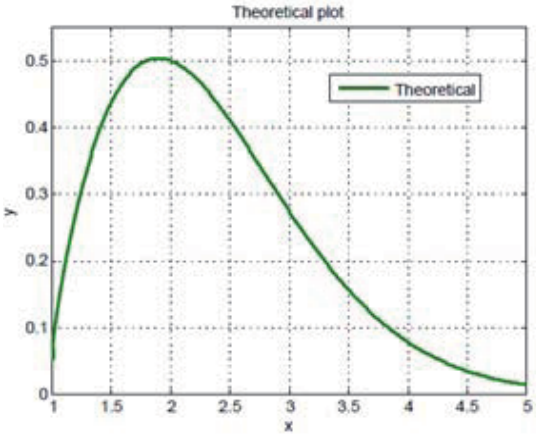

Figure 1: Theoretical plot for Weibull curve

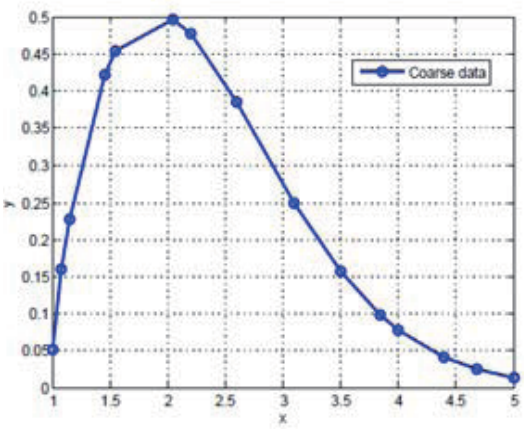

Figure 2: Graph for Weibull curve with coarse input

As it can be seen from Figure 2, values at the peaks are insufficient to plot a perfect Weibull curve, corresponding to the given conditions. In order to obtain an accurate plot through ANN, selection of number of neurons is an important factor. For a particular range of data entered, there exists an optimum number of neurons which best fits the same. This has been shown through the following set of graphs for different number of neurons (4-6 neurons), keeping the peak value and the average error of the ANN graph as parameters of comparison. Since the initial rise and dip in the graph are more or less traceable, predicting the peak accurately in this curve holds the main determining factor for finding the appropriate number of neurons. Graphs for different number of neurons has been shown below which also indicate the dependence of the same on accuracy of the plot.

A.) **4 NEURONS**: See Figure 3 and Table 1

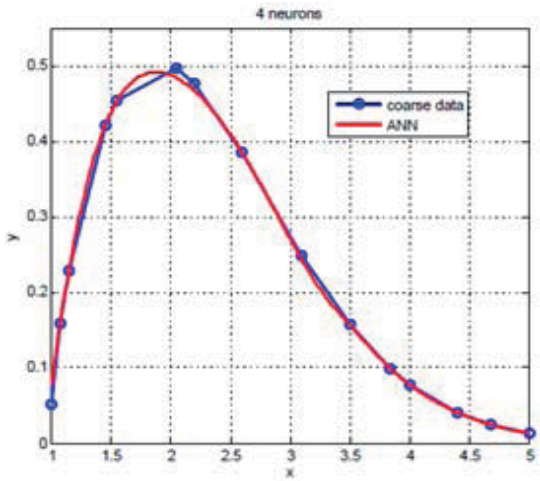

Figure 3: ANN predicted curve for 4 neurons

Table 1: ANN predictions and errors with 4 neurons

x	y(ANN)	y(theoretical)	Error (%)
1.00	0.0758	0.0621	-22.0399
1.26	0.3153	0.3135	-0.5742
1.40	0.3973	0.3944	-0.7353
1.60	0.4645	0.4678	0.7054
1.70	0.4813	0.4887	1.5142
1.90	0.4915	0.5042	2.5188
2.10	0.4792	0.4913	2.4629
2.20	0.4668	0.4767	2.0768
2.40	0.4316	0.4358	0.9637
2.60	0.3850	0.3849	-0.0260
3.00	0.2713	0.2750	1.3455
3.20	0.2165	0.2236	3.1753
3.40	0.1746	0.1776	1.6892
3.80	0.1054	0.1049	-0.4766

B.) **5 NEURONS**: See Figure 4 and Table 2

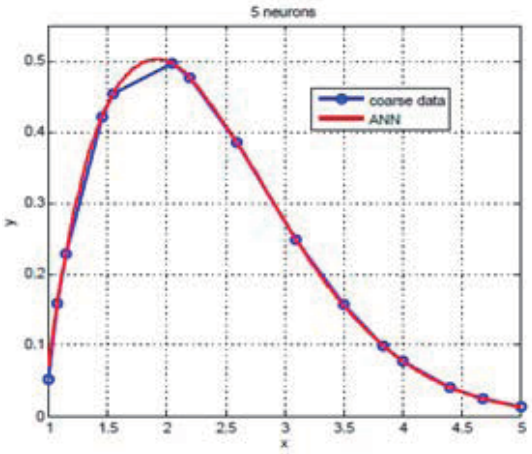

Figure 4: ANN predicted curve for 5 neurons

Table 2: ANN predictions and errors with 5 neurons

x	y(ANN)	y(theoretical)	Error (%)
1.00	0.0651	0.0621	-4.7297
1.26	0.3163	0.3135	-0.8931
1.40	0.3976	0.3944	-0.8114
1.60	0.4667	0.4678	0.2351
1.70	0.4859	0.4887	0.5729
1.90	0.5019	0.5042	0.4562
2.10	0.4917	0.4913	-0.0814
2.20	0.4767	0.4767	0.0000
2.40	0.4338	0.4358	0.4589
2.60	0.3849	0.3849	0.0000
3.00	0.2756	0.2750	-0.2182
3.20	0.2232	0.2236	0.1789
3.40	0.1771	0.1776	0.2815
3.80	0.1049	0.1049	0.0000

C.) **6 NEURONS**: See Figure 5 and Table 3

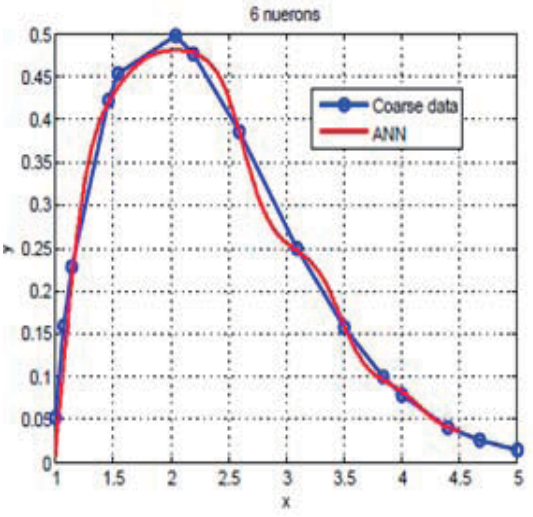

Figure 5: ANN predicted curve for 6 neurons

Table 3: ANN predictions and errors with 6 neurons

x	y(ANN)	y(theoretical)	Error (%)
1.00	0.0604	0.0621	2.8314
1.26	0.3270	0.3135	-4.3062
1.40	0.4123	0.3944	-4.5385
1.60	0.4467	0.4678	4.5105
1.70	0.4613	0.4887	5.6067
1.90	0.4777	0.5042	5.2559
2.10	0.4803	0.4913	2.2390
2.20	0.4776	0.4767	-0.1888
2.40	0.4553	0.4358	-4.4745
2.60	0.3840	0.3849	0.2338
3.00	0.2412	0.2750	12.2909
3.20	0.2360	0.2236	-5.5456
3.40	0.1928	0.1776	-8.5586
3.80	0.0994	0.1049	5.2431

RESULT:

Table 4: Effect of number of neurons on ANN predictions

No of neurons	**4**	**5**	**6**
Average error (%)	-0.53	**-0.3**	0.75
Maximum error (%)	-22.04	**-4.72**	12.3
Peak value (Theoretical)	0.5042	**0.5042**	0.5042
Peak value (ANN)	0.4915	**0.5019**	0.4777

Table 4 shows the effect of number of neurons on the predictions using ANN. We can see that the average error for all the curves is considerably low, but the actual crux of the problem lies in obtaining the accuracy of the plot at sensitive parts(e.g. peaks, valleys etc) which can otherwise be easily missed while plotting. Taking the above into consideration, the results have been consolidated. When the number of neurons is 4, the peak value predicted is only 0.4915 (indicating 2.518% error) and the maximum error in the prediction is -22.04%. When 5 neurons are used for predictions, the peak value is 0.5019 (0.4562% error) and the maximum error is -4.73%. When the number of neurons is increased to 6, the peak value is 0.4777 (5.255% error) and the maximum error is 12.29%.

Therefore, we observe that for a specified range of inputs, there is an optimum value for number of neurons which yields the correct graph. Deviating from this number, leads to distortions in the plot. Thus, it is clear that, for the coarse input (15 points), accurate plot is obtained for 5 neurons (Fig. 4).

2.2 Complex curve with varying Amplitude. [$y = x * sin(3x - 2)$]:

The same concept as performed above has been extended to another example- a complex trigonometric curve having varying amplitude. This complex curve consists of various inflexion points and continuously changing amplitude, which may not be plotted accurately in case of missing points at the peaks and valleys. By applying the concept of ANN, we can show the ability of the same to fit a curve for the given coarse data in spite of the complexity of the curve.

2.2.1 ANN Graph for coarse data

Following the same approach, as that of Weibull, we find similar connections between the number of neurons required and the input data entered.

X axis:	0.25	0.50	0.75	1.00	1.25
Y axis:	-0.2372	-0.2397	0.1856	0.8415	1.2300

X axis:	1.50	1.75	2.00	2.50	2.75
Y axis:	0.8977	-0.1893	-1.5140	-1.7640	-0.0912

X axis:	3.00	3.25	3.50	3.75	4.00
Y axis:	1.9710	3.2320	2.7950	0.6521	-2.1760

As we can see from Fig.7, a lot of points are missing at the peaks and valleys which are important characteristics of the curve. Depending on the selection of the suitable number of neurons, an accurate fit for the above incomplete curve can be obtained. Here the parameter of comparison between the graphs is the accuracy of prediction of the inflexion points in the different plots corresponding to different number of neurons.

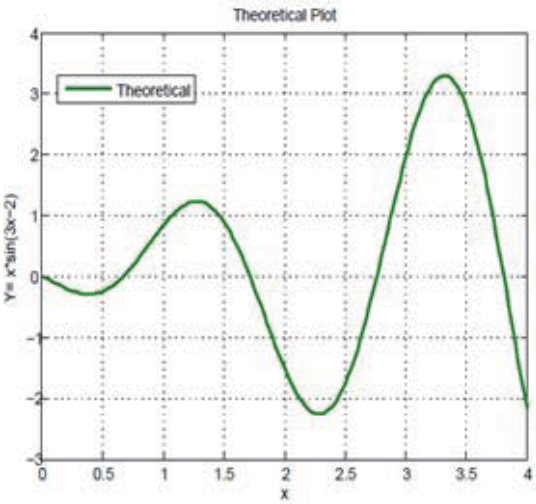

Figure 6: Theoretical plot for complex curve

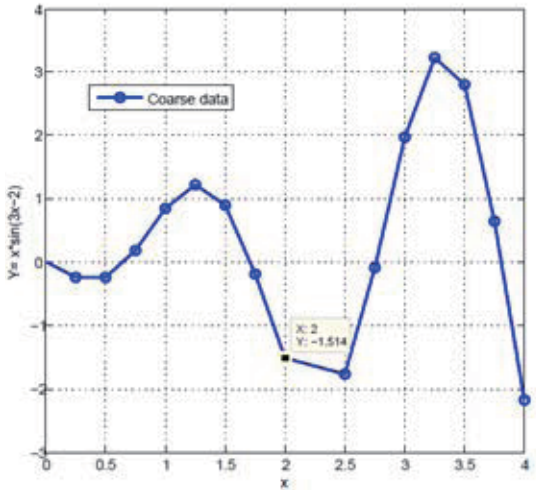

Figure 7: Graph for complex curve with coarse input

A.) **4 NEURONS**: See Figure 8 and Table 5

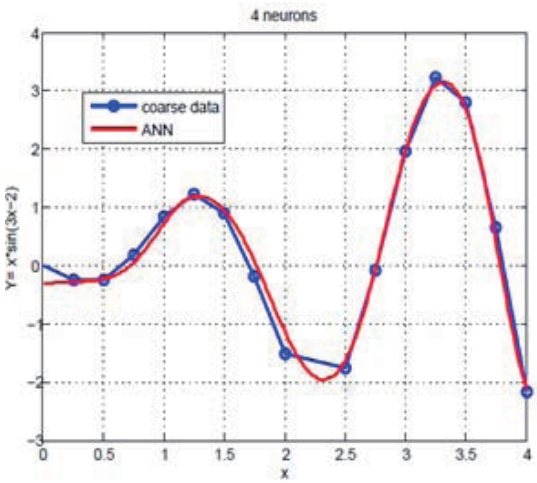

Figure 8: ANN predicted curve for 4 neurons

Table 5: ANN predictions and errors with 4 neurons

x	y(ANN)	y(theoretical)	Error (%)
0.20	-0.2932	-0.1971	-48.7570
0.60	-0.1593	-0.1192	-33.6409
0.80	0.1535	0.3115	50.7223
1.05	0.8608	0.9584	10.1836
1.30	1.1940	1.2300	2.9268
1.65	0.4895	0.3142	-55.7925
2.00	-1.1360	-1.5140	24.9670
2.30	-1.9580	-2.2600	13.3628
2.70	-0.4724	-0.4918	3.9447
3.00	1.9880	1.9710	-0.8625
3.30	3.1460	3.2970	4.5799
3.50	2.6750	2.7950	4.2934
3.70	0.9598	1.1810	18.7299
3.90	-1.2850	-1.0600	-21.2264

B.) **5 NEURONS**: See Figure 9 and Table 6

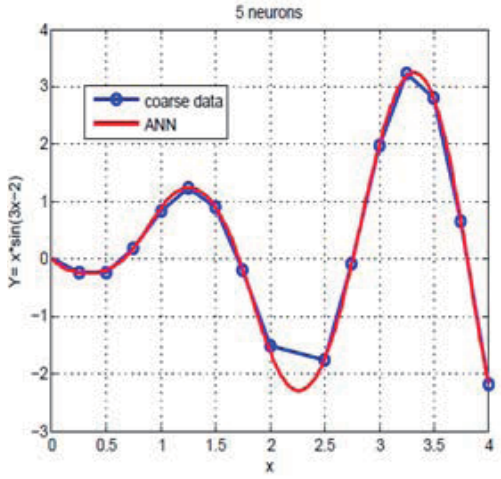

Figure 9: ANN predicted curve for 5 neurons

Table 6: ANN predictions and errors with 5 neurons

x	y(ANN)	y(theoretical)	Error (%)
0.20	-0.2179	-0.1971	-10.5530
0.60	-0.0991	-0.1192	16.8624
0.80	0.3145	0.3115	-0.9631
1.05	1.0060	0.9584	-4.9666
1.30	1.2210	1.2300	0.7317
1.65	0.3425	0.3142	-9.0070
2.00	-1.6090	-1.5140	-6.2748
2.30	-2.2790	-2.2600	-0.8407
2.70	-0.4803	-0.4918	2.3383
3.00	1.9770	1.9710	-0.3044
3.30	3.2400	3.2970	1.7288
3.50	2.8010	2.7950	-0.2147
3.70	1.2080	1.1810	-2.2862
3.90	-1.0860	-1.0600	-2.4528

C.) **6 NEURONS**: See Figure 10 and Table 7

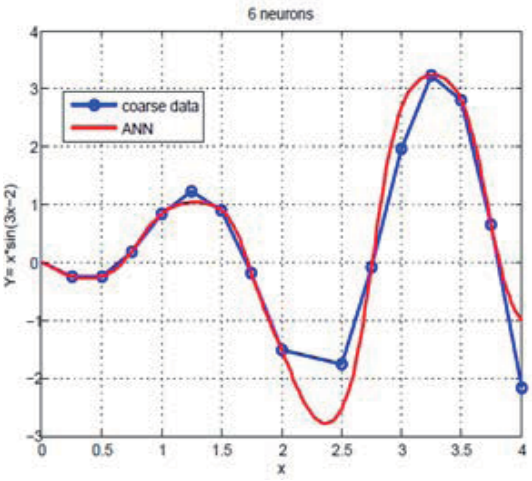

Figure 10: ANN predicted curve for 6 neurons

Table 7: ANN predictions and errors with 6 neurons

x	y(ANN)	y(theoretical)	Error (%)
0.20	-0.2027	-0.1971	-2.8412
0.60	-0.1394	-0.1192	-16.9463
0.80	0.3332	0.3115	-6.9663
1.05	0.9135	0.9584	4.6849
1.30	1.0350	1.2300	15.8537
1.65	0.3986	0.3142	-26.8619
2.00	-1.5140	-1.5140	0.0000
2.30	-2.7280	-2.2600	-20.7080
2.70	-0.7961	-0.4918	-61.8747
3.00	2.6700	1.9710	-35.4642
3.30	3.2250	3.2970	2.1838
3.50	2.7950	2.7950	0.0000
3.70	1.1870	1.1810	-0.5080
3.90	-0.5968	-1.0600	43.6981

From the above plots, though there are many points of inflexion in the graph, for means of comparison we have chosen a point at $x = 2.3$. The trough value predicted for the corresponding abscissa is noted for different plots given by different neurons and results are tabulated to determine the best fit.

RESULT:

Table 8: Effect of number of neurons on ANN predictions

No of neurons	4	5	6
Average Error (%)	-1.89	**-1.15**	-7.55
Maximum Error (%)	-55.79	**-10.55**	-61.87
Trough value (Theoretical)	-2.26	**-2.26**	-2.26
Trough value (ANN)	-1.95	**-2.27**	-2.72

From Table.8 we can observe that for the chosen abscissa ($x = 2.3$), the plots corresponding to 4 and 6 neurons show a large deviation from the theoretical value ($y = -2.26$), whereas the graph produced by 5 neurons proves to be the required best fit. Also comparing the overall error, we see that the plot produced by 4 neurons has a value of -1.8977% and that of 6 neurons has a value of -7.5536%. Whereas the plot for 5 neurons contains the minimum deviation from the theoretical plot with a value of -1.1572%. Also the maximum error posed is very high for 4 and 6 neurons as compared to that of 5 neurons. Here too we observe that for a specific number of neurons (5 in this case), an accurate fit for any complex curve can be obtained.

Similarly, by applying the above concept to several other examples we have arrived at a relation be-

tween the number of inputs used and the corresponding number of neurons required to obtain the required best fit. We observe that, depending on the complexity of the curve, for a higher number of inputs, the number of neurons required for the best fit also increases. The final relation has been consolidated in conclusion section.

The above examples and results shown validate our idea of simplicity of curve fitting through ANN. So, this approach can now be applied to real world problems.

3 Application of ANN to Real Life Problem:

DEEP DRAWING PRESS and its Mechanism: Deep drawing press is a metal forming operation in which blank is radially drawn into forming die by the mechanical action of punch. Deep drawing is a shape transformation process with material retention. The mechanism used for this, comprises of three 4 bar chains which link the crank to the sliding press. The technological process requires that the die approaches the work piece with medium-high velocity and moves with almost constant low velocity during the operation and thereafter returns quickly [12]. The methodology of curve fitting through ANN has been applied to the link mechanism of a deep drawing press (Fig 11) to generate plots of slider displacement and velocity using coarse data.

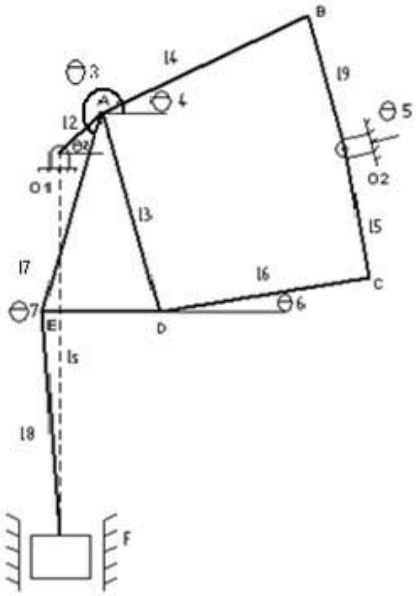

Figure 11: Mechanism used for deep-drawing press

3.1 Displacement graphs:

By applying loop equations to the 4 bar chains and reducing them, we arrive at a final relation between slider displacement (l_s) and crank angle (θ_2) which helps in generating the displacement graph. The table below shows the coarse input data collected for displacement for various crank angles.

X axis:	0.00	60.0	90.0	120	150
Y axis:	46.902	53.787	58.527	61.453	63.302
X axis:	180	220	270	313	320
Y axis:	65.824	68.436	64.256	53.727	51.918

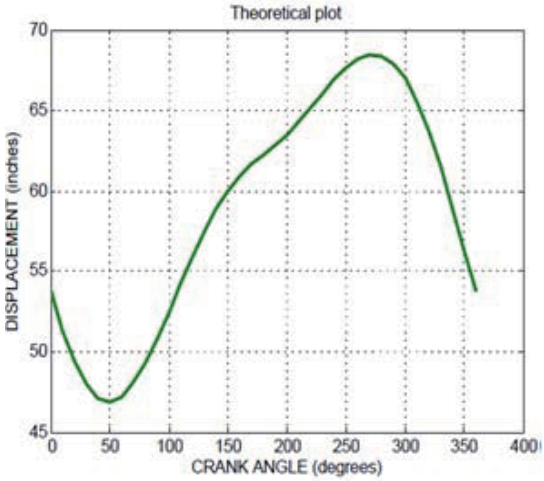

Figure 12: Theoretical Graph of displacement v/s crank angle

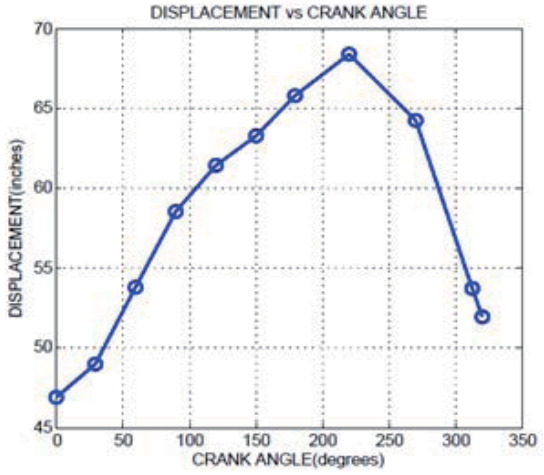

Figure 13: Graph of displacement v/s crank angle (coarse input)

Generating the required graph through ANN:

A.) **4 NEURONS**: See Figure 14

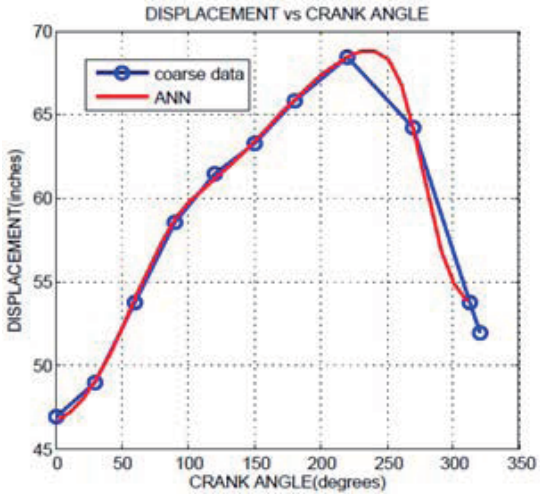

Figure 14: ANN predicted curve for 4 neurons

B.) **5 NEURONS**: See Figure 15

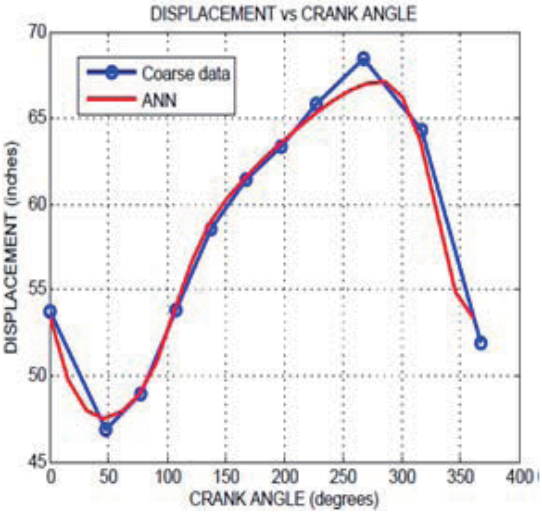

Figure 15: ANN predicted curve for 4 neurons

RESULT:

In accordance with the relation obtained between the number of inputs and the number of neurons from the previous examples and extending the same to the displacement characteristics, we observe that for 4 neurons, an accurate graph is obtained. The number of inputs used in this case is only 11.

Table 9:ANN predictions and errors with 4 neurons

X (Deg)	Y disp(in)	Actual disp(in)	Error (%)
1	53.26	53.477	0.400
16	47.97	50.122	4.293
31	46.30	47.855	3.250
46	46.59	46.921	0.705
61	47.80	47.295	-1.067
76	49.44	48.754	-1.407
91	51.26	50.960	-0.588
106	53.15	53.525	-0.700
121	55.06	56.080	1.818
136	56.92	58.327	2.412
151	58.72	60.088	2.276
166	60.42	61.349	1.514
181	62.00	62.284	0.456
196	63.45	63.200	-0.395
211	64.76	64.340	-0.652
226	65.92	65.686	-0.356
241	66.94	67.007	-0.099
256	67.80	68.084	0.329
271	68.41	68.480	0.102
286	68.56	68.151	-0.600
301	67.65	66.865	-1.174
316	64.69	64.545	-0.224
331	59.78	61.283	2.452
346	55.37	57.409	3.551

Average error= 0.77%

3.2 Velocity graphs:

Assuming a specific RPM for the crank, values of velocity can be found for corresponding crank angles to obtain the velocity plot.

The following table shows the coarse input:

X axis:	10	15	20	25	30
Y axis:	0.918	1.298	1.710	2.072	2.363
X axis:	70	75	80	85	90
Y axis:	3.017	2.920	2.763	2.585	2.382
X axis:	100	110	120	130	140
Y axis:	2.070	1.638	1.133	1.180	1.064
X axis:	150	160	170	180	190
Y axis:	1.212	1.360	1.650	1.670	1.593
X axis:	230	235	240	245	250
Y axis:	0.664	0.683	1.070	1.472	2.145
X axis:	270	311	312	314	316
Y axis:	2.936	4.522	4.486	4.436	4.327
X axis:	318	320	325	330	335
Y axis:	4.079	4.050	3.785	3.345	2.860
X axis:	340	345	350	360	

Y axis: 2.230 1.905 1.236 0.000

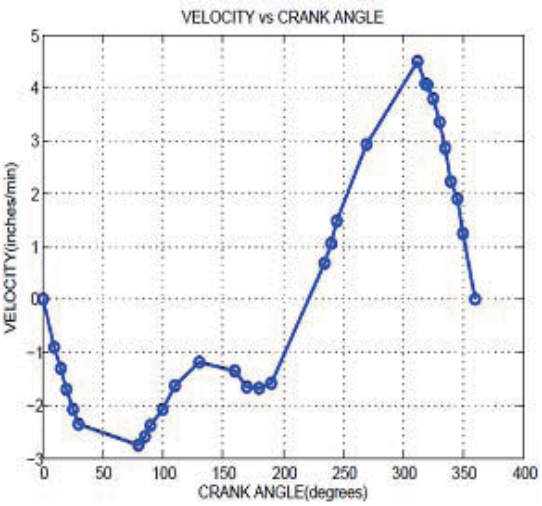

Figure 16: Graph of velocity v/s crank angle for coarse data

Generating the required graph through ANN:

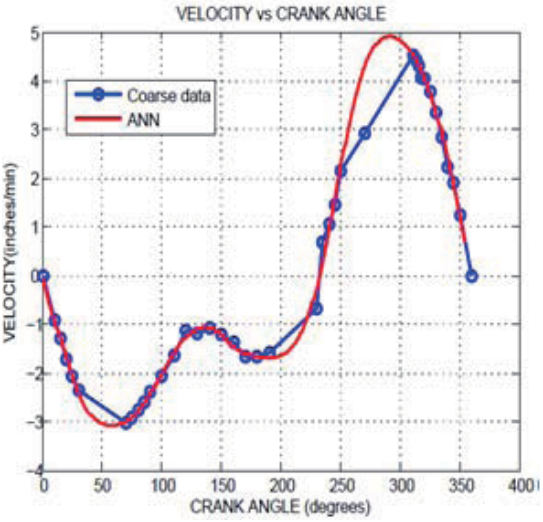

Figure 17: ANN predicted curve for 8 neurons

RESULT

Accurate graphs are obtained for multiple points of inflexion for both displacement and velocity plots corresponding to correct number of neurons (here, correct number of neurons is 8).

Table 10: ANN predictions and errors with 8 neurons

X (deg)	Vel (ANN)	Vel (Actual)	Error (%)
4.5	-0.4402	-0.4000	-10.0500
28.5	-2.2900	-2.2630	-1.1931
52.5	-2.9170	-3.1180	6.4464
76.5	-2.8340	-2.8590	0.8744
100.5	-2.0030	-1.9700	-1.6751
124.5	-1.1760	-1.1380	-3.3392
148.5	-1.2040	-1.2110	0.5780
172.5	-1.5930	-1.6440	3.1022
196.5	-1.6430	-1.5032	-9.3002
220.5	-1.1490	-1.4345	19.9024
244.5	1.5960	1.4220	-12.2363
268.5	3.7720	3.1150	-21.0915
292.5	4.4270	4.6000	3.7609
316.5	4.2330	4.2800	1.0981
340.5	2.2940	2.2900	-0.1747

Average Error= 1.5532%

Though a maximum error of 21% is obtained at a point, we can see that the average error is low and thereby succeeds in tracing the curve quite accurately at almost all of the intervals. It is also successful in predicting the peaks quite effectively.

We, therefore, conclude from Table 10 that, even for such a highly non-linear plot containing ridges and valleys at different parts of the graph, ANN is very much capable of providing the best fit, in spite of a coarse data, corresponding to the optimum number of neurons.

4 Educational Value:

While conducting experiments, results are recorded at particular intervals. Graphs are plotted with these records to analyze the system. Hence, it is necessary that the plot obtained is accurate enough. But the data is insufficient to characterize the system between the intervals, especially in indicating the points of inflexions. In such a situation, ANN succeeds in generating the best representation of the intermediate (missing) data. This methodology can be utilized widely in understanding the behavior of any system. Therefore, it can be incorporated as a part of the experimental procedures in laboratories, to get better and accurate results.

5 Conclusion:

An attempt has been made in this paper to fit curve for coarse data using ANN. The effect of number of neurons on the accuracy of the prediction is illustrated for 2 examples of Weibull distribution curve and a complex trigonometric sinusoidal curve. This methodology is successfully extended to an 8-bar mechanism

used in deep drawing press, for obtaining both displacement and velocity plots.

From the analysis, the following can be concluded for relating the number of inputs to the number of neurons for obtaining the required best fit.

- For a coarse input set consisting of up to 10 to 15 entries, the number of neurons required to get the desired graph is 3 or 4.
- For a relatively bigger set of inputs of around 30-40 entries, optimum number of neurons is 7 or 8 (as shown in the velocity plot of the deep drawing press mechanism).

  -Use of more than 10 neurons is undesirable, as the graph generated tends to distort and fails to predict correctly the trends of the system.
- For a very large input data there is no need to use ANN to fit a curve as the input points joined together itself proves sufficient to plot the required graph.

Therefore we are able to show that in experiments/ studies requiring a huge number of inputs and calculations to be entered in order to produce an appropriate graph, ANN reduces that burden and proves to be a simple tool for efficient curve fitting even for insufficient or coarse data.

Acknowledgements: We would like to thank Mr. T S Ashwin for helping us with the Artificial Neural Network (ANN) tool in the MATLAB software. This work is supported in part by the TEQIP 1.2.1 research grant (World Bank), for the Centre of Excellence in Knowledge Analytic and Ontological Engineering (KAnOE) at PES Institute of Technology, Bangalore-560085, India.

References:

[1] M. Gulsen, A. E. Smith and D. M. Tate, A genetic algorithm approach to curve fitting, *International Journal of Production Research*, No.33, 1995, pp.1911-1923.

[2] Mark H. Richardson and David L. Formenti, Global curve fitting of frequency response measurements using the rational fraction polynomial method, *3rd IMAC Conference*,1985.

[3] Lejun Shao and Hao Zhou, Curve fitting with Bezier cubics, *Graphical models and image processing*, No. 58, 1996, pp.223-232

[4] T. Kato, Non–stationary flows of viscous and ideal fluids in $\mathbb{R}^3$, *J. Func. Anal.* 9, 1972, pp. 296–305.

[5] Ilaria Dimatteo, Christopher R Genovese and Robert E Kass, Bayesian curve fitting with free-knot splines, *Biometrika*, No.88,Vol. 4, 2001, pp.1055-107.

[6] Aris Spanos, Curve-fitting- the Reliability of inductive inference and the Error-Statistical Approach, *Philosophy of Science*, 2007

[7] Sarah F. Frisken, Efficient curve fitting, *Journal of Graphic tools*,No.13, 2008, pp.37-54.

[8] L. Desmet and I. Gijbels, Curve fitting under jump and peak irregularities using local linear regression, *Communications in statistics- Theory and methods*, No. 40, 2011, pp.4001 -4020.

[9] Paul. A. Bekker and Kees. E. Bouwman, Arbitrage smoothing in fitting a sequence of yield curves, *International Journal of Theoretical and Applied Finance*, Vol. 12, No. 5, pp.577- 588.

[10] Christopher M. Bishop and C. M. Roach, Fast curve fitting using neural networks,*Microsoft research, Cambridge*, 1992

[11] I. M. Barbosa, E. del Moral Hernandez, M. L. C. C. Reis, O. A. F. Mello, Measurement uncertainty contribution to the calibration curve fitting of an aerodynamic external balance using MLP artificial neural network,*XVIII Imeko World Congress, Metrology for Sustainable Development*, 2006

[12] Paul Theodosis, Neural network: Interpolation and curve fitting, *(SOURCE- internet)*, 2007.

[13] Y. Krishnamurthy, C. Balasubramanyam and A. K. Gangopadhyay, Optimization of link drive mechanism, *12th AIMTDR Conference*, 1986

[14] P. M. Fonte, Gonalo xufre silva, J. C. Quadrado, Wind Speed Prediction using Artificial Neural Networks, *6th WSEAS International Conference on Neural Networks, Lisbon, Portugal*, June 16-18, 2005, pp. 134-139

[15] H. Mohammadi Majd, M. Poursina, K. H. Shirazi, Determination of barreling curve in upsetting process by artificial neural networks, *9th WSEAS International Conference on Simulation, Modelling and Optimization*

[16] Eugen Diaconescu, The use of NARX Neural Networks to predict Chaotic Time Series, *WSEAS Transaction on Computer Research* Issue 3, Volume 3, 2008, pp.182-191