# Simulation study- How to interpolate and extrapolate a functional dependency with a neural network.

Sub task:

- Literature survey of interpolation and extrapolation
- Determination of a suitable network layout
- Learning how to use MATLAB neural network toolbox
- Various mathematical method for simuation

# Interpolation

- Real world data is noisy. Noise can come in the form of incorrect or missing values. That can affect actual data set value so that interpolation is necessity for finding actual output.
- **Interpolation** - interpolation is a technique for adding new data points within a set of known data points
- The primary reason to interpolate is simply to increase the sampling rate at the output of one system so that another system operating at a higher sampling rate can input the signal.
- Generally intrupt1 command is using in MATLAB to make faster interpolation for bunch ofdata.
- Different interpolation methods like linear, spline, double, etc are using in MATLAB.

# EXTRAPOLATION

The word "**extrapolation**" starts with the word "**extra**" which means "**outside**".so basically extrapolation is where we insert something or some different data values in between twogiven points.

Extrapolation is estimation of a value based on expanding known data or fact beyond the area that is certainly known.

Simply extrapolation is the prediction of any futuristic results by using present data values and past methods which are using for solution but the verified data set is in under observation and try to be the bulky data is similar in on going process.so that we can get predicted outcomes at the end.

An extrapolation may be thought as a hypothesis or a mathematical guess.While extrapolating, one uses data and facts given in the present situation and makes predictions about something that might eventually happen.

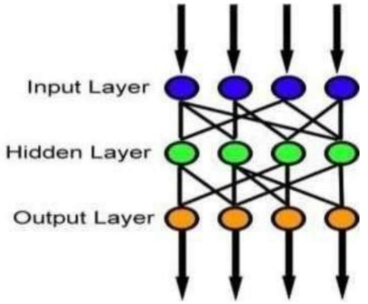
This process is more tough then interpolation.

**For Example :** If we have some information about Sunday, Monday and Tuesday, we might be able to extrapolate about Wednesday or Thursday.

- An example of interpolation may be locating a point between the two given endpoints of a straight line; while extrapolation is the process of locating a point outside the straight line i.e. beyond its two extremes by extending the line in either direction.
- **Extrapolation:** estimating values a = 8 and b = 10 outside the given series 0, 2, 4, 6, a, bin which data values are considered as points such as x1, x2..., xn and then, a value is approximated beyond the given range of the points.
- Application
  - Digital Photography
  - Education Simulation
  - process Security Managment
  - Heat TransferEstimation
  - Finding darivatives from experimental datavalues

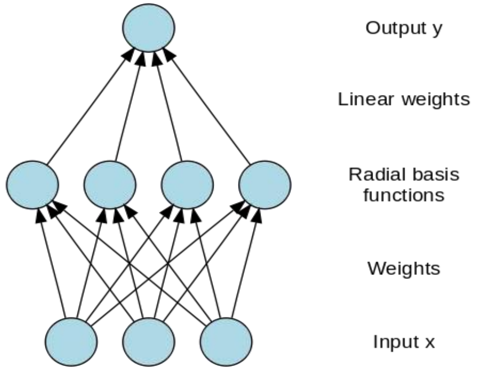
## 1) Feed Forward Neural Network:

This network only moves information in only one direction.  Data only transfer input mode to output mode through  hidden layer.
This is the basic features matlab simulation of any bulky data set.



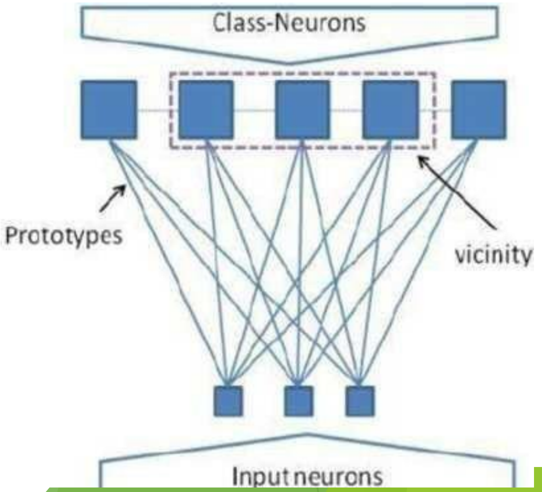

## 2)Radial Basic Function Neural Network :

The RBF neural networks in is the first choice to Interpolating in multi-dimensional space. This is good classification but works bad for continuous values. Continuous process overlapd due to multiple neurons flow in layers.




## 3) KOHONEN Self-organizing Neural Network:

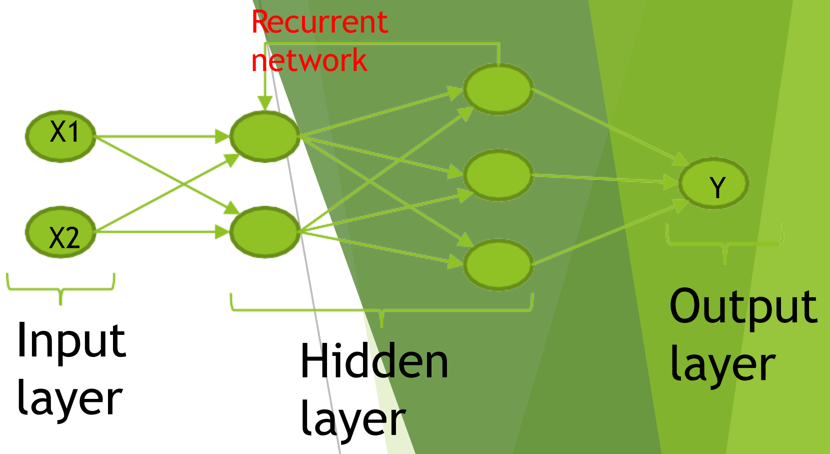
It is Invented by TEUVO KOHONEN. Ideally for the  visualization of lowdimensional data. This network learn  all set of input data.



## 4) Recurrent Neural Network

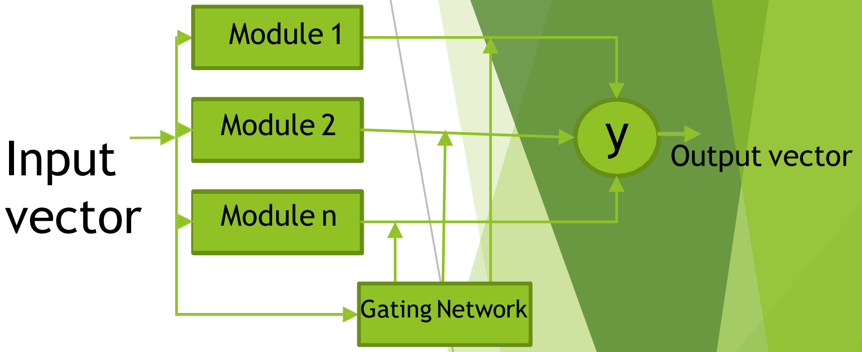
This network is capable to using internal memory to process arbitrary sequence of inputs.

This type of network use for when decision from past interaction or sample can influence current data. This network is similar time delay and distributed delay neural networks which have finite input response.



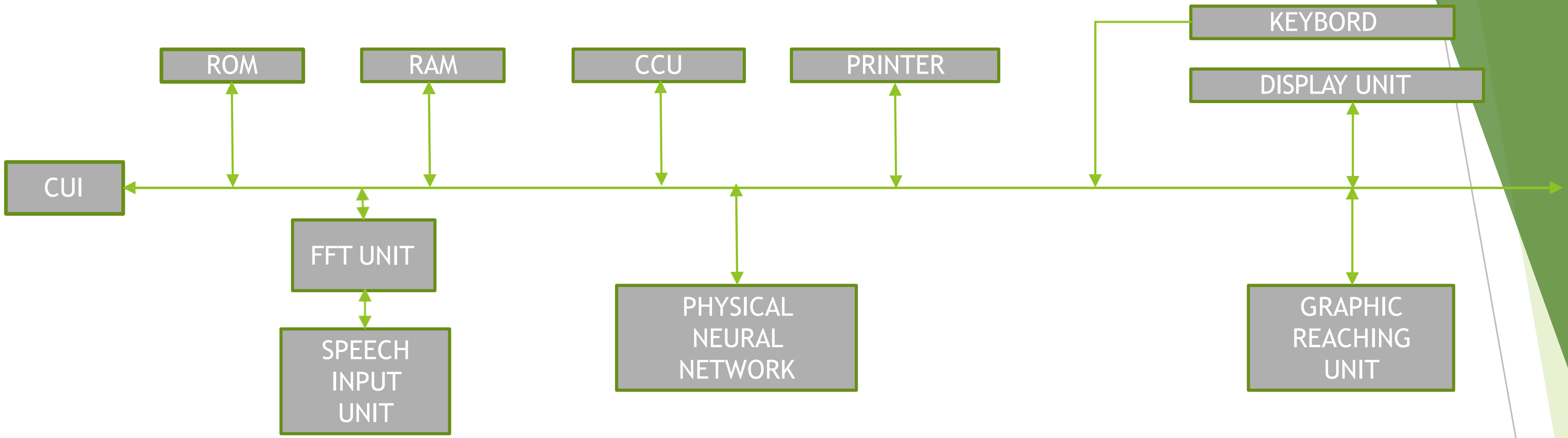

## 5) Modular Neural Network

Multiple independent neural networks that are moderate by intermediary. Every single networks work with single input and every task is going on way and perform like one task.




## 6) Physical Neural Network

This layout give importance or dependent on physical hardware or software when simulatize. Electrically adjustable resistance used for emulating the function of neural synope. It mean physical hardware use for neuron or software emulates the neural networks.



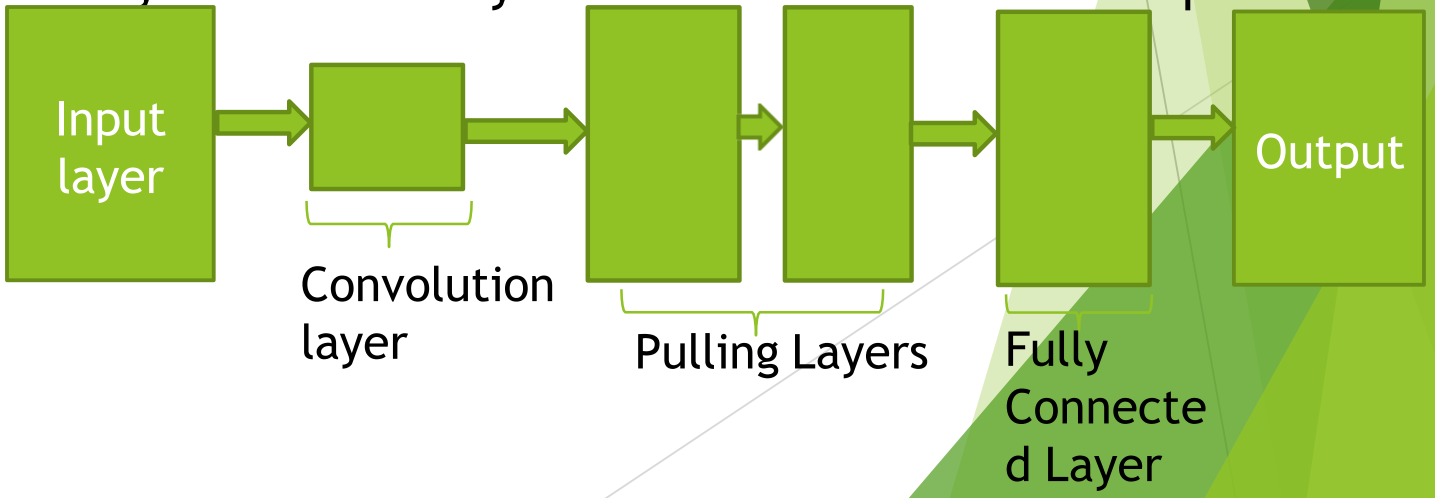

- **7) Convolutation Neural Network (CNN)**
- This neural network is known for folding neural network in artificial networks world.
- basically this network is a biological-inspired concept in the field of machine learning.
- Best neural network for any interpolation and extrapolation. There are three layers
- In the processing any data value or image and etc.
- **1. convolution layer**
- Wherein all activity of neurons can be calculated by discrete convolution. If the data set
- is in small convolution matrix then its moved step by step over the input.
- **2. Pooling layer**
- In the next step, superfluous information is discarded. In pooling, There are different

The activity of the remaining neurons is discarded. Despite the data reduction 75% or more.

- Reduced space requirements and increased calculation speed
- The resulting ability to create deeper networks that can solve more complex tasks
- Automatic growth of the size of the receptive fields in deeper Convolutional Layers (without explicitly increasing the size of the folding matrices)
- Prevention measure against overfitting.

3. **Fully-connected Layer**

when the convolution and pooling layer process is completed then after our network is terminated and made one fully connected layer which is look like multiple layer of perceptron's.



## ❖ Learning how to use Matlab neural network tool box

- The Matlab neural network toolbox provides a complete set of functions and a graphical user interface for the design , implementation, visualization and simulation of neural networks.
- It supports the most commonly used supervised and unsupervised network architectures and a comprehensive set of training and learning functions.
- Graphical user interface for creating, training and simulating your neural networks.

- GENERAL CREATION OF NETWORK

  net = network net= network(numInputs , numLayers , biasConnect , inputConnect , layerConnect, outputConnect, targetConnect)

## ❑ NEURAL NETWORK TOOLBOX GUI

1. The graphical user interface (GUI) is designed to be simple and user friendly. This tool lets you import potentially large and complex data sets.

2. The GUI also enables you to create, initialize, train, simulate, and manage the networks.
It has the GUI Network/Data Manager window.

3. The window has its own work area, separate from the more familiar command line
workspace. Thus, when using the GUI, one might "export" the GUI results to the (command line) workspace. Similarly to "import" results from the command line workspace to the GUI.

4.Once the Network/Data Manager is up and running, create a network, view it, train it,
simulate it and export the final results to the workspace. Similarly, import data from the

- A graphical user interface can thus be used to

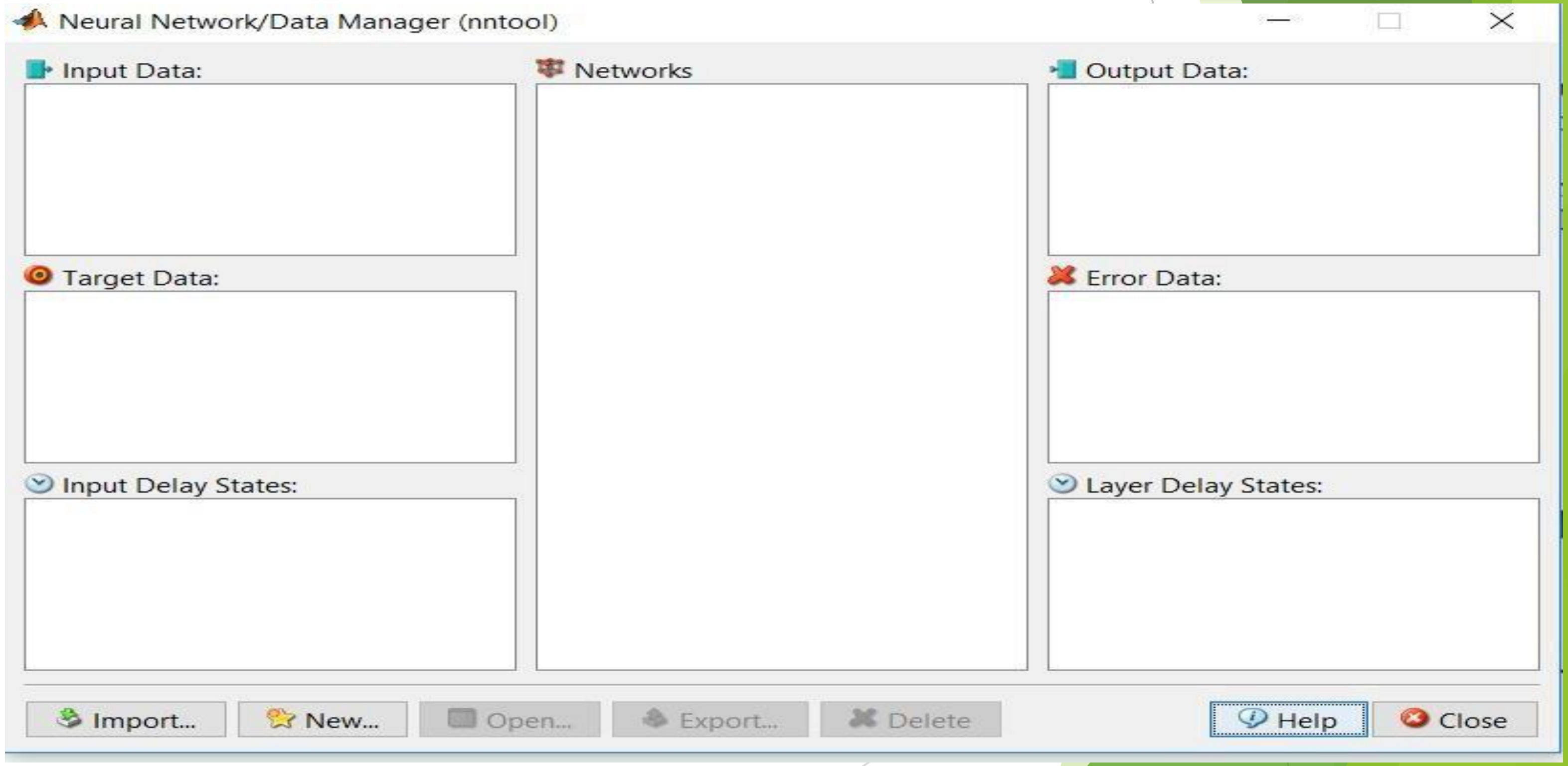
1. Create networks
2. Create data
3. Train the networks
4. Export the networks
5. Export the data to the command line workspace

- To define a fitting problem for the toolbox, arrange a set of Q input vectors as columns in a matrix. Then, arrange another set of Q target vectors (the correct output vectors for each of the input vectors) into a second matrix . For example, you can define the fitting problem for a Boolean AND gate with four sets of two element input vectors and one-element targets as follows:

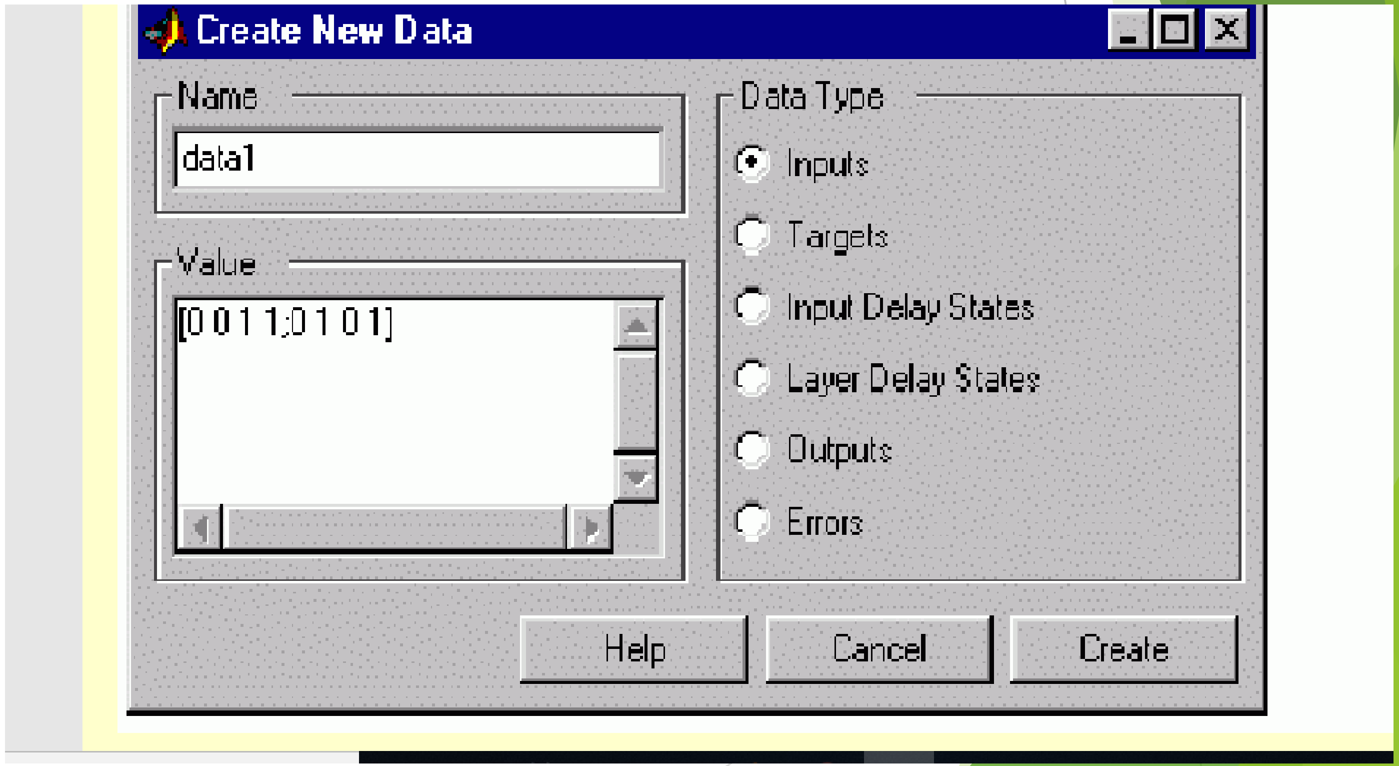
  inputs = [0 1 0 1; 0 0 1 1]; targets = [0 0 0 1];

# ❑Create a Perceptron Network (nntool)

We create a perceptron network to perform the AND function in this example.
It has an input vector p= [0 0 1 1;0 1 0 1] and a target vector t=[0 0 0 1]. We call the network ANDNet. Once created, the network will be trained. We can then save the network, its output, etc., by “exporting” it to the command line. Input and target To start, type nntool. The following window appears.

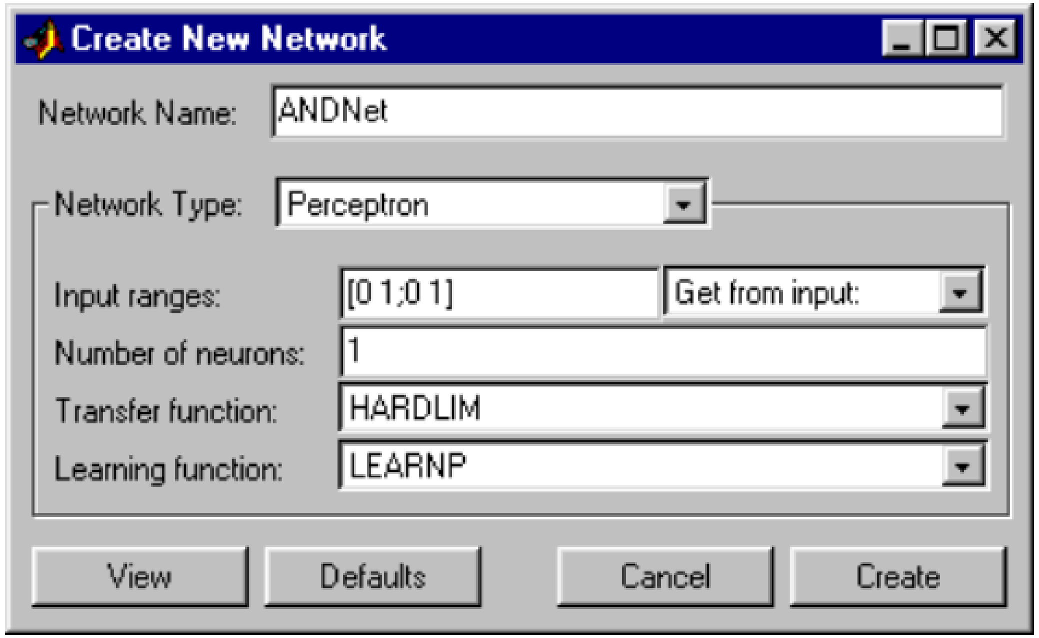


- Click on Help to get started on a new problem and to see descriptions of the buttons and lists. First, we want to define the network input, which we call p, as having the particular value [0 0 1 1;0 1 0 1]. Thus, the network had a two-element input and four sets of such two-element vectors are presented to it in training. To define this data, click on New Data, and a new window, Create New Data appears. Set the Name to p, the Value to [0 0 1 1;0 1 0 1], and make sure that Data Type is set to Inputs .The Create New Data window will then look like this:



- Now click Create to actually create an input file p. The Network/Data Manager window comes up and p shows as an input. Next we create a network target. Click on New Data again, and this time enter the variable name t, specify the value [0 0 0 1], and click on Target under data type. Again click on Create and you will see in the resulting Network/Data Manager window that you now have t as a target as well as the previous p as an input. Create Network Now we want to create a new network, which we will call ANDNet . To do this, click on New Network, and a Create New Network window appears. Enter AND Net under Network Name. Set the Network Type to Perceptron, for that is the kind of network we want to create.

- Next you might be get final network after train network and than export the network.



## OVERVIEW OF ONE DIMENSIONAL TECHNIQUES

This sections deals with the techniques used to find the basis1 function which represents all the measured data. A function is an algebraic expression that relates the dependent variable to the independent one. Using these forms, an analytical function is modelled which is used while interpolation and extrapolation depending on the application. Among the functional forms, the most common one dimensional techniques are discussed here.

A. **Linear Interpolation Technique** Two data points in coordinate frame which are given as $(x0, y0)$ and $(x1, y1)$respectively. The equation of a straight line is given by:

$f(x) = y = mx + c$ Here, $m$ is gradient of the line and $c$ is y-intercept of the equation, which is the value of $y$ at $x = 0$,and are given by the following formula: $m = \frac{(y_1 - y_0)}{(x_1 - x_0)}$

$c = y_1 - mx_1$

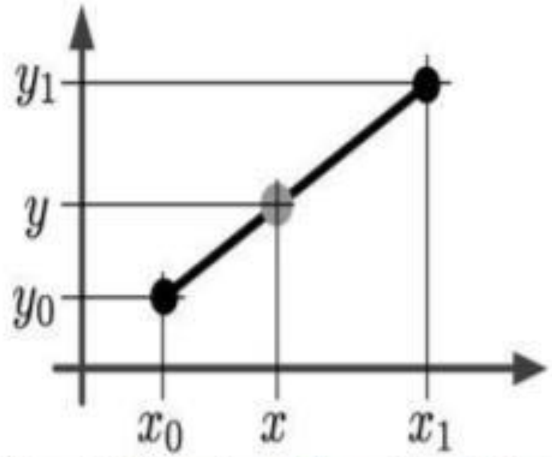
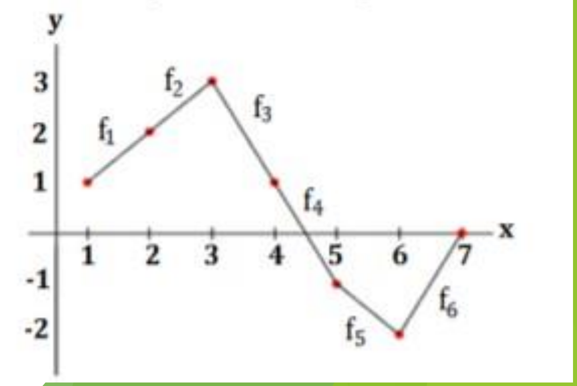


Figure 4. Illustration of linear interpolation.



## B. Piecewise Interpolation Technique

Piecewise interpolation fits separate functions to cover a part of data range rather to fit a single function to all of the data points. Consider $N$ data points and each pair of successive data points define the interval and for each interval, a specific function $f_k$(x) is assigned, where $k = N - 1$and represents the number of function which is needed for $N$ data points. To interpolate, find the interval in which desired value of $x$ lies and use that corresponding function to interpolate. Linearly interpolated estimate for $y$ is a straight line function that can be expressed as $y = f_k(x) = y_k + \frac{x - x_k}{x_{k+1} - x_k}(y_{k+a} - y_k)$

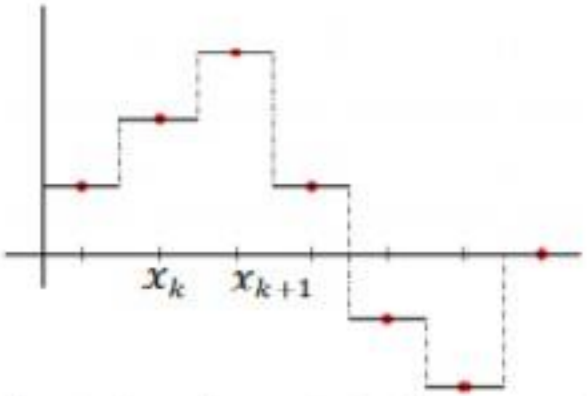


Data points interpolated with Piecewise Linear Interpolation

## C. Nearest Neighbor Interpolation Technique

This method sets the value of an interpolated point to the value of the nearest existing data point. Thus, the interpolation looks like a series of plateaus, which can be thought of as zero-order polynomials. Consider two successive data points as $x_k$ and $x_{k+1}$ , this method first find the mid value using these data points. Then, values of $x$ which is smaller than the mid value corresponds to the value of of $y_k$ and values greater than the mind value corresponds to the value of $y_{k+1}$. . The function $f_k(x)$ which define nearest neighbor interpolation technique is given by,

$$f_k(x) = \begin{cases} y_k, & x \leq \frac{1}{2}(x_k + x_{k+1}) \\ y_{k+1}, & x > \frac{1}{2}(x_k + x_{k+1}) \end{cases}$$




Data points (red) are interpolated with nearest neighbor technique

## D. Polynomial Interpolation Technique

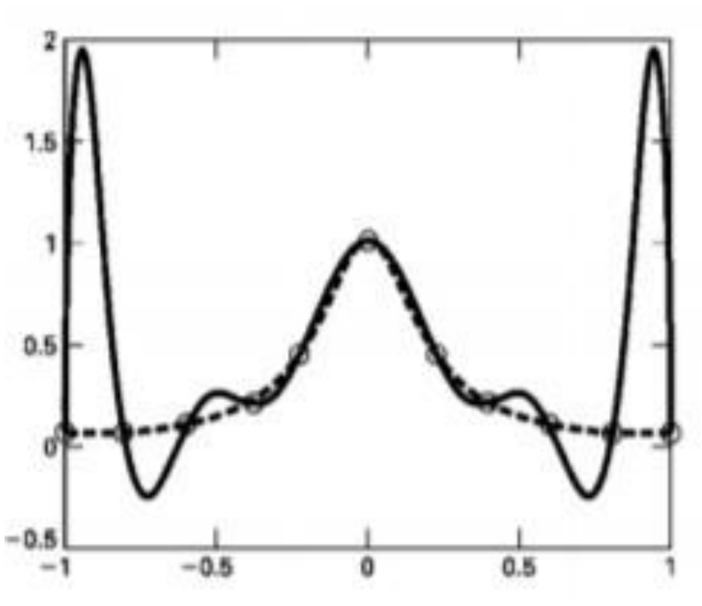
Linear interpolation is also the polynomial interpolation of degree 1 which is obtained using two data points. For 3 data points a quadratic equation of degree 2 polynomial is formed. The form of quadratic equation is $y = ax^2 + bx + c$ where $a$, b and c are coefficients. To find which quadratic function will pass through all these data points, the values of coefficient should be found. Similarly for $N$ points, degree $N - 1$ polynomial is needed. Then the function $f(x)$ becomes $f\ x = c_0 + c_1x + c_2x^2 + \cdots + c_{N-1}x^{N-1}$. In polynomial interpolation, when increasing the number of data points, the number of parameters and equations increases that results in complex mathematical equations. There exists a more practical form known as Lagrange form for the polynomial interpolating N data points and it is given as

$$f(x) = \sum_{k=1}^{n} \left( \prod_{\substack{j=1 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x} \right) y_k$$

Lagrange formula can be seen as sum of $N$ different polynomials and each is a polynomial of degree $N - 1$and is weighted by the measurement values$yk$ for $k$ = 1, … , $N$.

This method has some limitations as the increasing number of points are computationally more costly to perform this method. Also, higher order of polynomial will give rise to the round off error and hence it will deviate from our desired function, and it can introduce oscillations to our interpolated function where it should not as shown in the figure 7. The dashed line represents a function, the circles represent samples of the function, and the solid line represents the results of a polynomial interpolation in the figure.

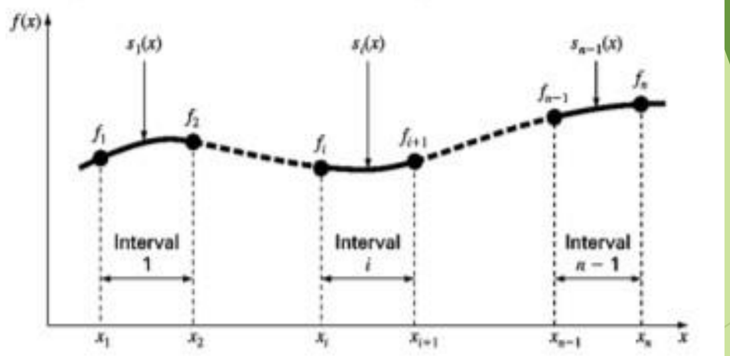
Neville‟s algorithm2 is the most efficient way of computing Lagrange Formula.



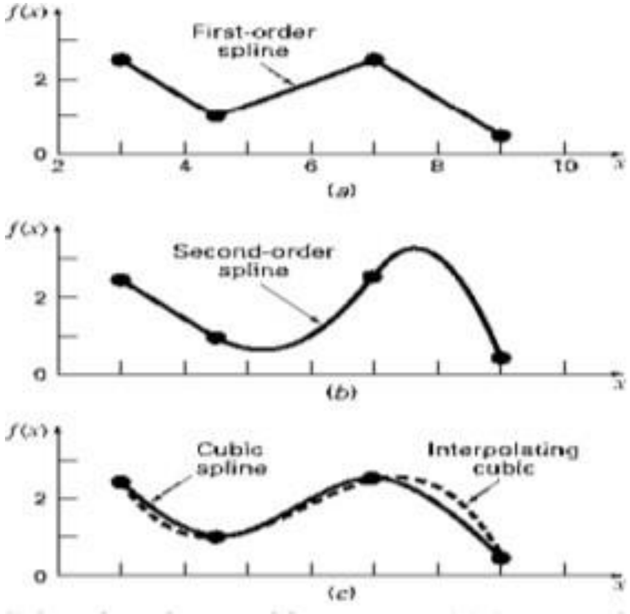

Tenth-order polynomial fit to 11 points[8]

E. Spline Development

The $(N - 1)th$ order polynomial can lead to erroneous results because of round-off error and oscillations as shown in the above figure. To avoid this an alternative approach is to apply lower order polynomials in a piecewise fashion to subsets of data points. These connecting polynomials are called spline functions. For $N$ data points, $(i = 1, \dots , N)$ there are $N - 1$ intervals, and each interval has its own spline function $si\ x$ , shown in figure 8, and the number of data points used in each splice function depends on the order of the spline function




Spline are simplest representation with the appearance of smoothness and without the problems of higher order polynomials. Linear splines have discontinuous first derivatives, whereas quadratic splines have discontinuous second derivatives as shown in the figure 9. Higher-order splines tend to exhibit ill-conditioning or oscillations.




Spline fits of a set of four points. (a) Linear spline, (b) quadratic spline, and (c) cubic spline

1) Cubic Spline Interpolation Cubic

spline interpolation is sufficiently a smooth piecewise interpolant due to the reason that a cubic polynomial has continuous first and second derivatives and it is the smoothest method of interpolation discussed in this paper. The cubic spline function for the $i\ th$ interval can be written as:

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

For $n$ data points, there are $(n - 1)$ intervals and thus $4(n - 1)$ unknowns to evaluate to solve all the spline function coefficients.

Following are the Conditions to determine the spline coefficients.

- The first condition : The spline function goes through the first and last point of the interval; this leads to 2(n-1) equations:

$$s_i x_i = f_i \Longrightarrow a_i = f_i$$

$$s_i(x_{i+1}) = f_i$$

- The second condition: The first derivative should be continuous at each interior point; this leads to (n-2) equations:

$$s_i{}'\, x_{i+1} = s_{i+1}{}'\,(x_{i+1})$$

- The third condition: The second derivative should be continuous at each interior point; this leads to (n-2) equations:

$$s_i{}''(x_{i+1}) = s_{i+1}{}''\,(x_{i+1})$$

Now there are $(4n - 6)$ equations; and $(4n - 4)$ equations are required to find the all the coefficients of the spline function.

# MULTIDIMENSIONAL INTERPOLATION TECHNIQUES

The interpolation and extrapolation methods for onedimensional problems can be extended to multi-dimensional interpolation. In this, each measurement location requires more than one coordinate to specify its position. For example if interpolating function is a function of two variables, then is known as multidimensional interpolation. This is also known as bilinear interpolation. In this case, data points can be specified by $x_i$ and $y_i$ coordinates and the measured value at the $i\ th$ data point can be defined by $z_i$. Now interpolating function is a function of two variables now and can be written as $z = f(x, y)$.

A. Bilinear Interpolation Technique

The simplest interpolation in 2-D is bilinear interpolation on the grid square. In this approach, the main idea is to interpolate in one direction and then to interpolate in the other remaining direction. It is bounded by four points then, two in one direction and two in other direction. Hence, the problem is divided into the combinations of linear sub problem. This can also be seen from the figure.

Following the linear interpolation in which the interpolated estimate at an arbitrary location $x_i$ depends upon a specific interval or a pair of data points in which this point lies. Once that pair of data points has been identified say $x_1$ , $x_2$ as shown in the figure, then the interpolation uses these values at the end points of the interval $y_1$ , $y_2$ to calculate the desired value of function as $z = f(x, y)$.

From the figure, four data points are $Q_{11} = (x_1 , y_1)$, $Q_{12} = (x_2 , y_2)$, $Q_{21} = (x_2 , y_1)$, and $Q22 = (x_2 , y_2)$.

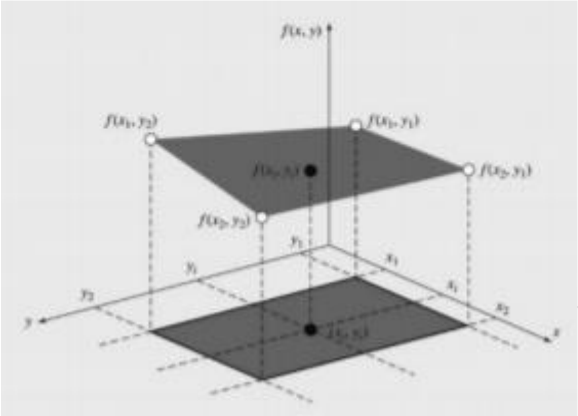
First interpolate in $x$-direction, which is given by the following formula,

$$f\ (x,y_1) \approx \frac{x_2-x}{x_2-x_1}\ f(Q_{11}) + \frac{x-x_1}{x_2-x_1}\ f(Q_{21})$$

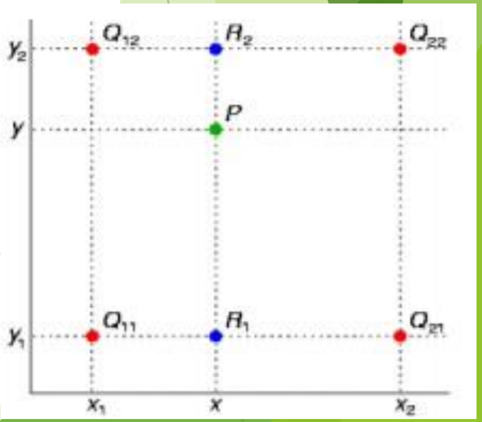
$$f\ (x,y_2) \approx \frac{x_2-x}{x_2-x_1}\ f(Q_{12}) + \frac{x-x_1}{x_2-x_1}\ f(Q_{22})$$

And then in y-direction to find our desired function,

$$f\ (x,y) \approx \frac{y_2-y}{y_2-y_1}\ f(x,y_1) + \frac{y-\ {}_1}{y_2-y_1}\ f(x, y_2)$$



Graphical depiction of two-dimensional bilinear interpolation where an intermediate value (filled circle) is estimated based on four given values (open circles).




. Two-dimensional bilinear interpolation can be implemented by applying one-dimensional linear interpolation

# Thank you