Open IIT

DATA ANALYTICS

INTRODUCTION

Drilling process is selected as the test-bed here. Remaining Useful Life (RUL) is defined as the time-to-failure of a specific machine while Expected life expectancy is the average life of similar components/machines.

Our objective is to predict the RUL of the drill-bit during the machining process by utilizing thrust-force and torque signals captured by a dynamometer during the drilling cycle.Each drill-bit was used until it reached a state of physical failure.All together there are 14 drill bit with varying number of holes drilled.The data is provided for a particular drill and a particular hole.We have to build a prediction model that can predict RUL.So we divided the dataset as first 10 drill bits for training and last 4 for testing.The performans of the model is given by the median RMSE for testing dataset made up of just the last five holes for each drill bit.

The report starts with detailed data description, a brief overview of the exploratory analysis performed during the study, the steps followed for data cleaning to recollect the data in the desirable form, a detailed process and approach involved in feature selection, relevant model selection for predicting RUL

Features

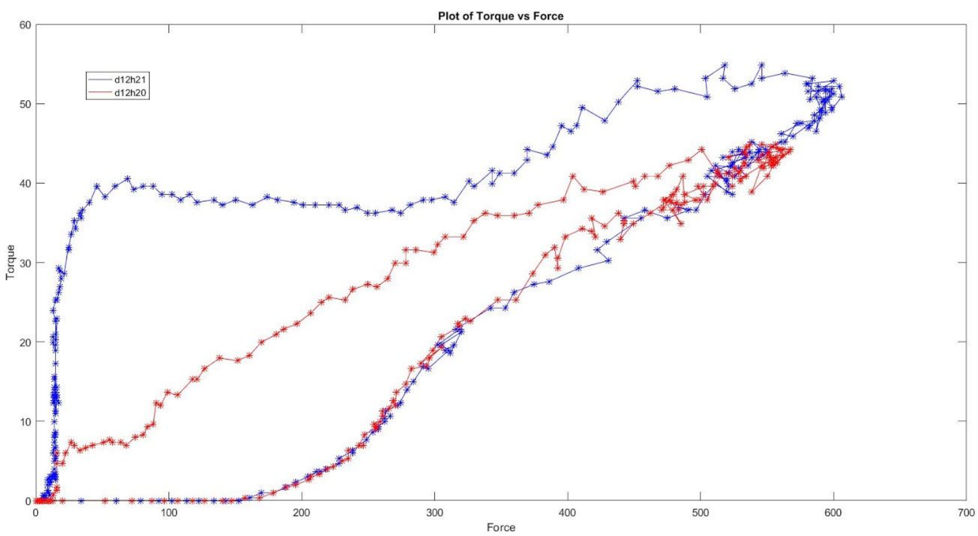
1. Mean_Thrust:- It is the average of all the thrust values for given drill bit and hole no.
2. Mean_Torque:- It is the average of all the torque values for given drill bit and hole no.

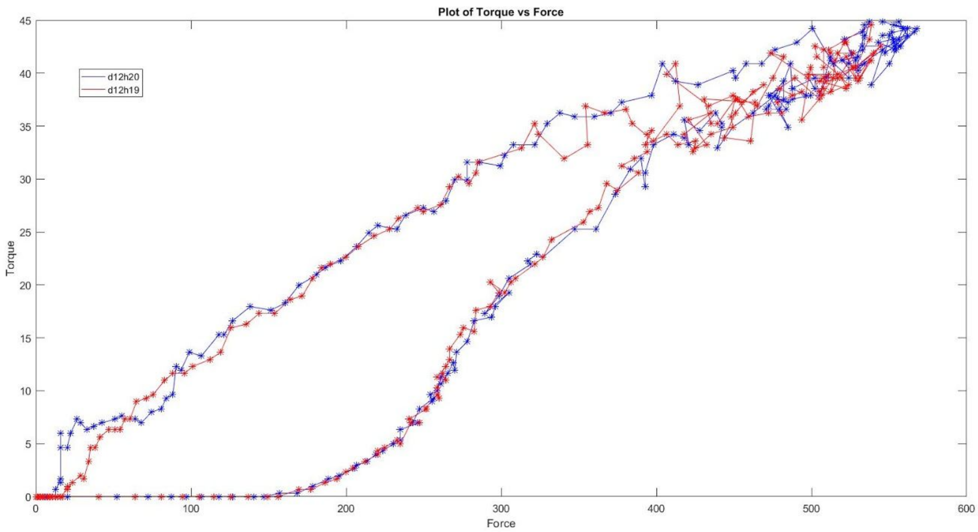
3. Thrust_non_zero:-It is the first non zero thrust value for given drill bit and hole no.
4. Torque_non_zero:-It is the first non zero torque value for given drill bit and hole no.
5. Torque_zero:-It is the last non zero torque value for given drill bit and hole no.
6. Max_Thrust:-It is the maximum of all the thrust values for given drill bit and hole no.
7. Max_Torque:-It is the maximum of all the torque values for given drill bit and hole no
8. Degradation :- Percentage of drill degraded after drilling each hole.

Degradation = (Sum of observation for one hole in a particular drill)/
(Sum of total observation for that particular drill)

E.g If drill 1 drilled 21 hole. For hole number 1 it has 200 observations(lets say) and the total observations in drill 1(for all holes) is 20000. Then the degradation for Drill1- Hole1 is 200/20000

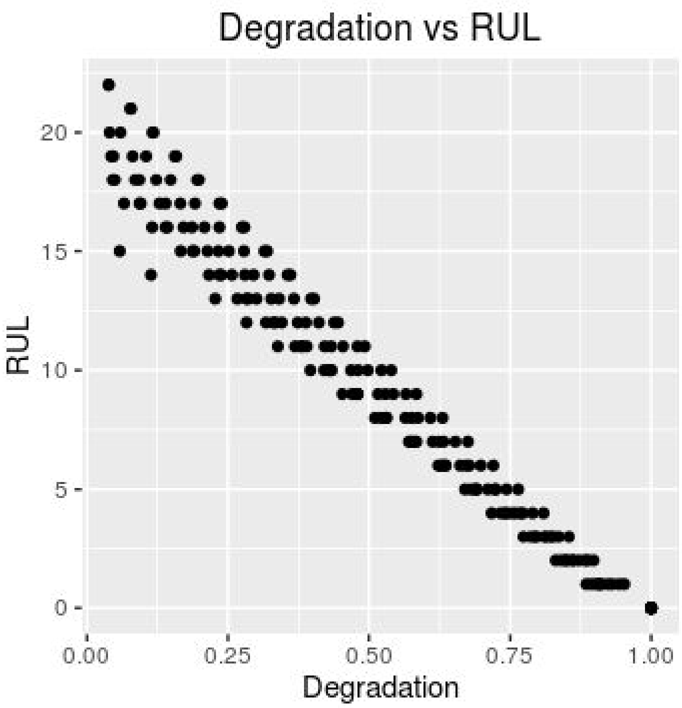
9. Area :- It is defined as the area under the curve of thrust and torque signal for a given drill bit and hole number. We observed increased area under the curve for the last hole for each drill as shown in the figure below.



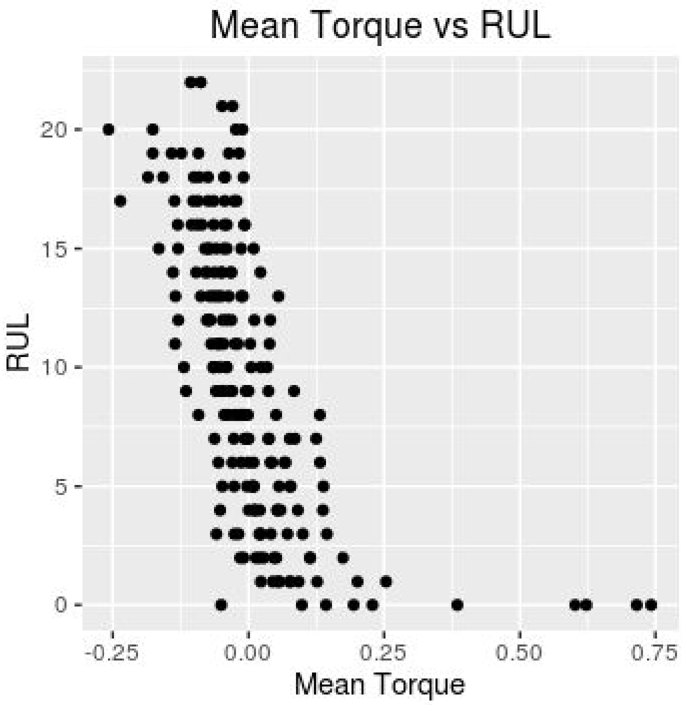


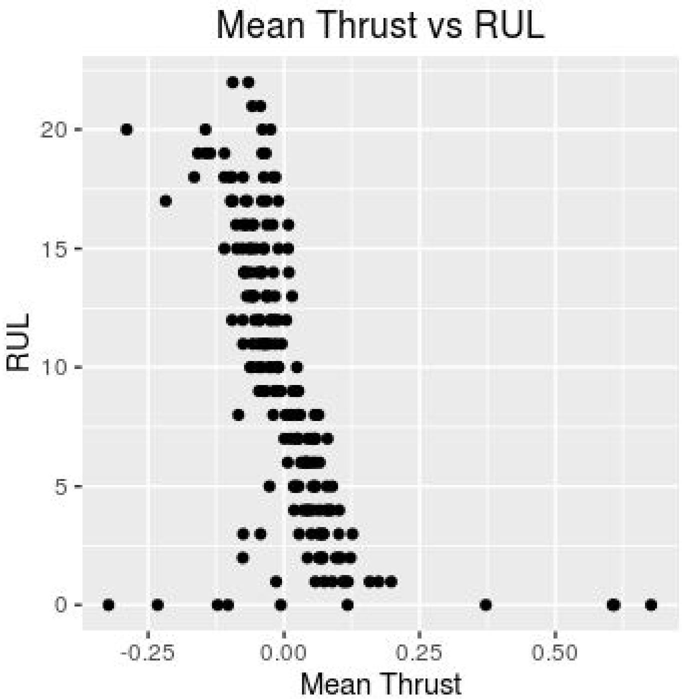
Area feature is the area under this curve.

Exploratory data analysis

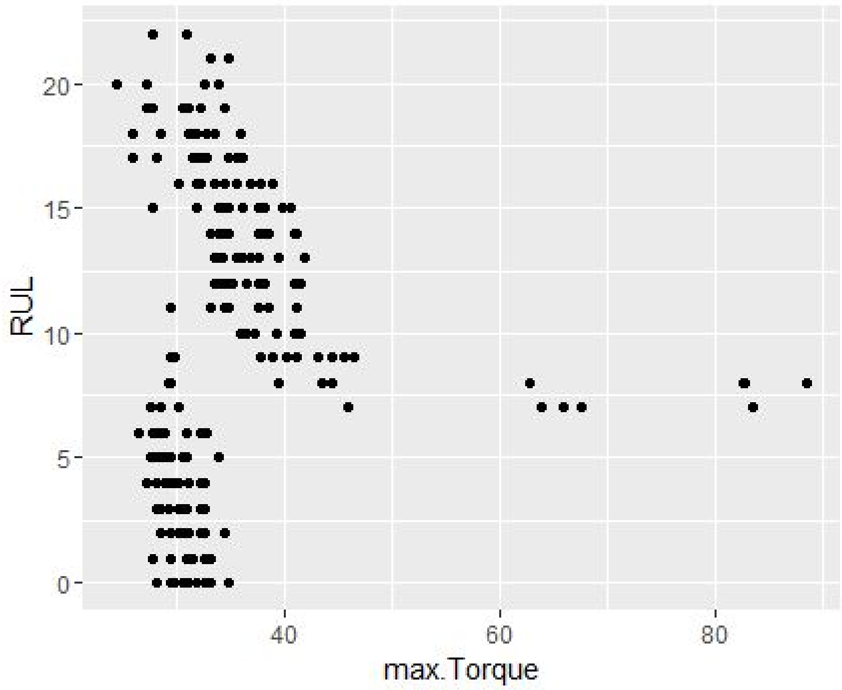


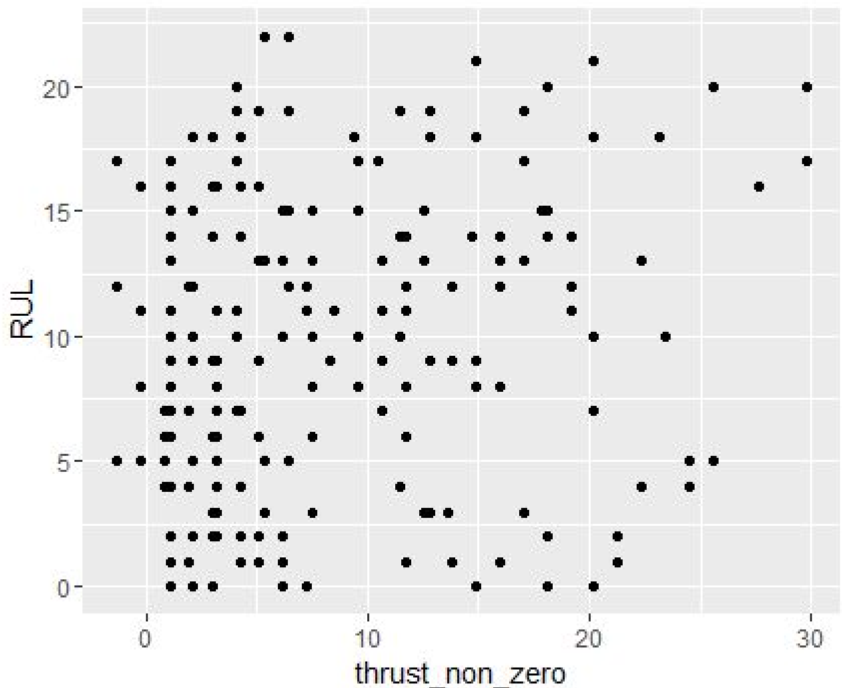
It is seen that degradation is showing a linear relationship with RUL and hence is an important feature for our model.

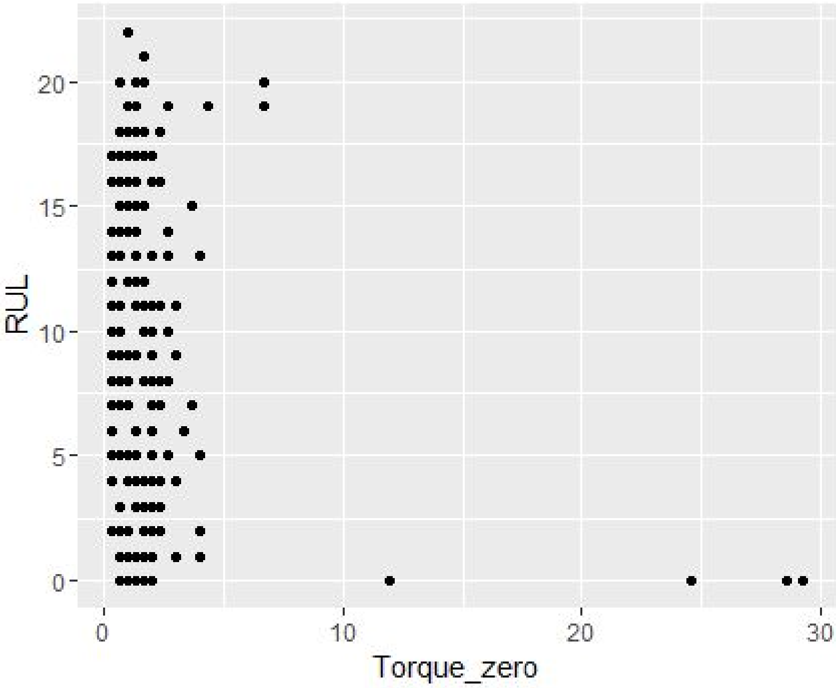




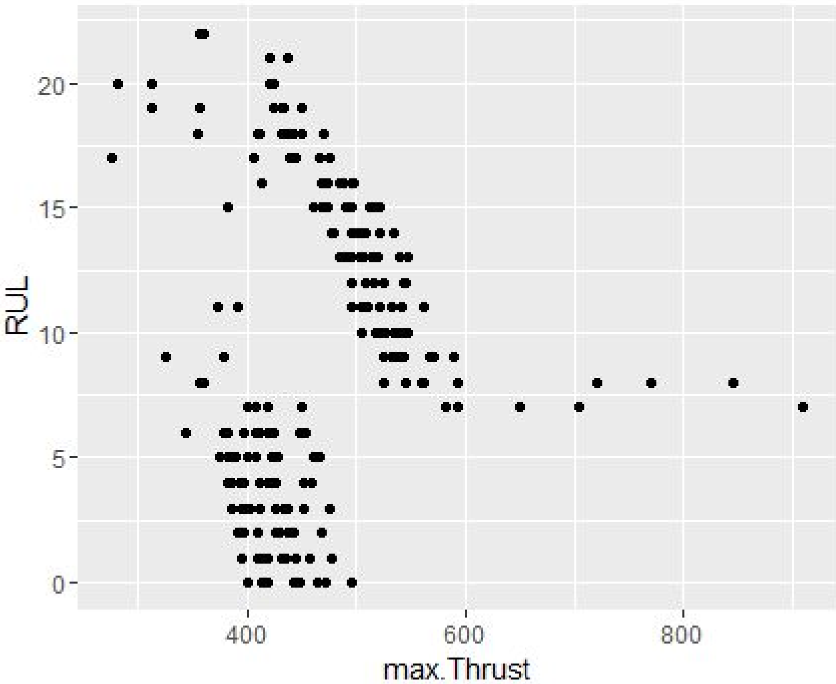
These two graphs show that RUL has a relatively weak relationship with the mean of these two values but it can be seen that after a particular mean value, the RUL becomes zero.


RUL
max.Torque
0
5
10
15
20
40
60
80


RUL
thrust_non_zero
0
5
10
15
20
0
10
20
30



z



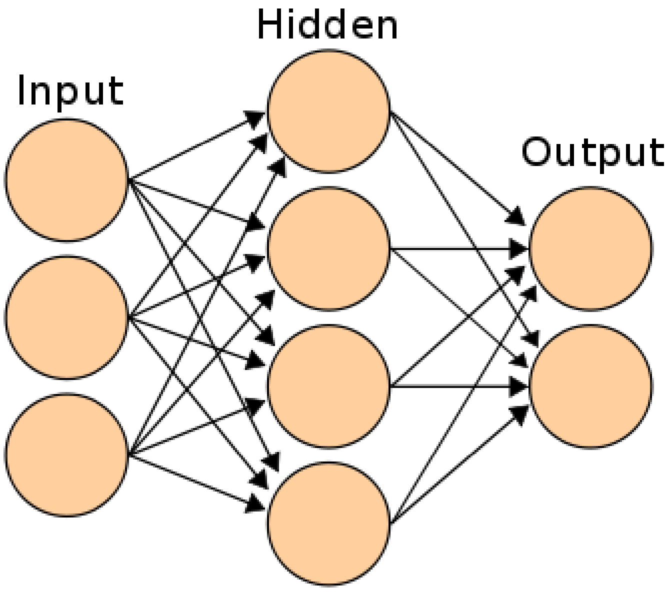
Data Description

Drilling process, one of the most commonly used machining processes, is selected here as a test-bed for validating the proposed autonomous diagnostics framework. The diagnostics objective is to assess the health or well-being of the drill-bit during the machining process by utilizing thrustforce and torque signals captured by a dynamometer during the drilling cycle (constituting a logical sensor signal segment). Tests were conducted on HAAS VF-1 CNC Machining Center with Kistler 9257Bpiezo-dynamometer (sampled at 250Hz) to drill holes in ¼ inch stainless steel bars. High-speed twist drill-bits with two flutes were operated at feed rate of 4.5 inch/minute and spindle-speed at 800 rpm without coolant. Each drill-bit was used until it reached a state of physical failure either due to excessive wear or due to gross plastic deformation of the tool tip due to excessive temperature (resulting from excessive wear). There are total 14 drills in our dataset with the shortest drill-bit lasted 16 holes and the longest 45 holes.

Model Description

Model1 - Neural Networks

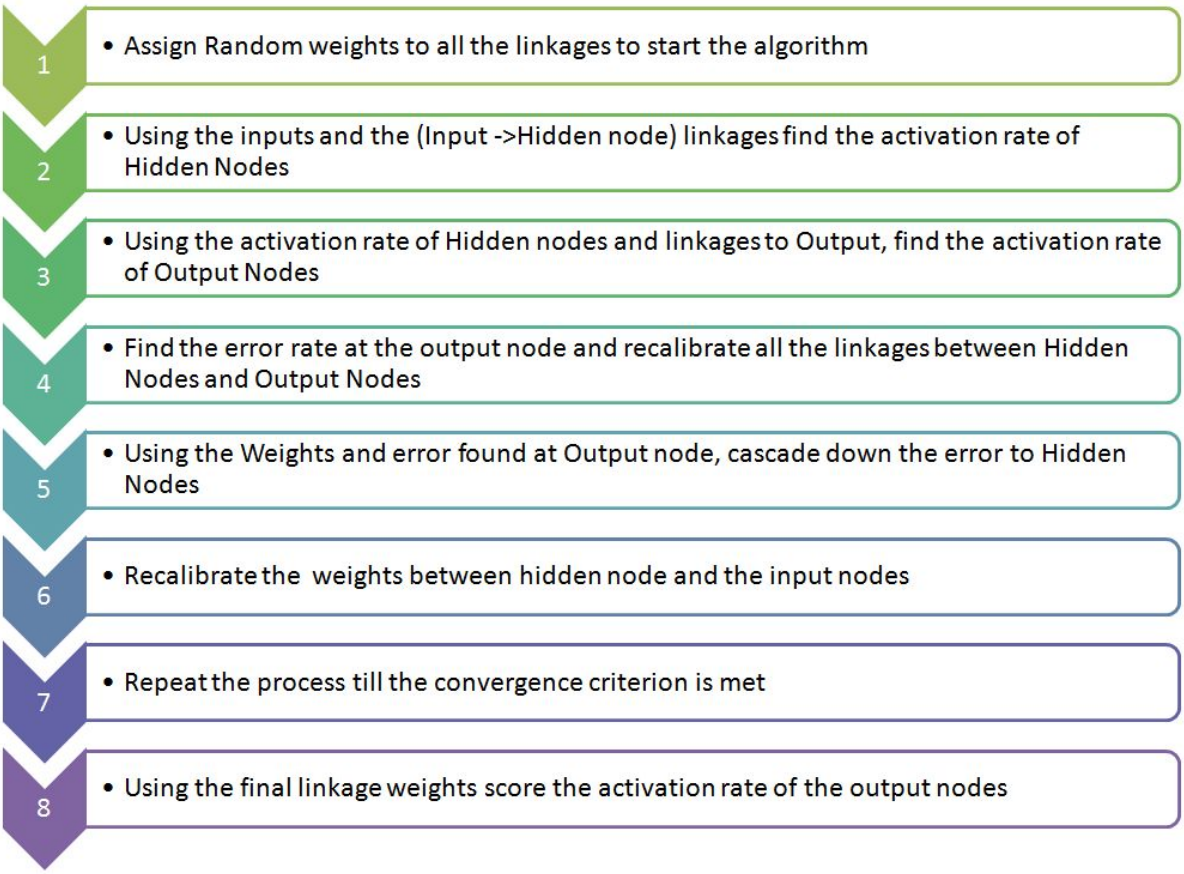
We will start with understanding formulation of a simple hidden layer neural network. A simple neural network can be represented as shown in the figure below:



The linkages between nodes are the most crucial finding in an ANN. We will get back to “how to find the weight of each linkage” after discussing the broad framework. The only known values in the above diagram are the inputs. Lets call the inputs as I1, I2 and I3, Hidden states as H1,H2.H3 and H4, Outputs as O1 and O2. The weights of the linkages can be denoted with following notation:

W(I1H1) is the weight of linkage between I1 and H1 nodes.

Following is the framework in which artificial neural networks (ANN) work:

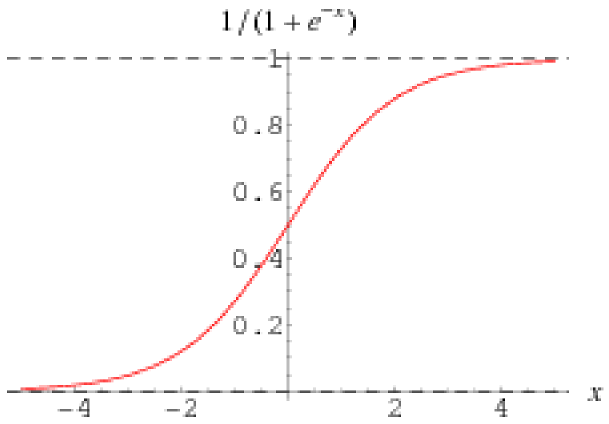


Few statistical details about the framework

Every linkage calculation in an Artificial Neural Network (ANN) is similar. In general, we assume a sigmoid relationship between the input variables and the activation rate of hidden nodes or between the hidden nodes and the activation rate of output nodes. Let's prepare the equation to find activation rate of H1.

Logit (H1) = W(I1H1) * I1 + W(I2H1) * I2 + W(I3H1) * I3 + Constant = f = > P(H1) = 1/(1+e^(-f))

Following is how the sigmoid relationship looks like :



Model 2 - Perceptrons(Single-layer Neural Network)

The perceptorn is an algorithm for supervised learning of binary classifier. It is a type of Linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.

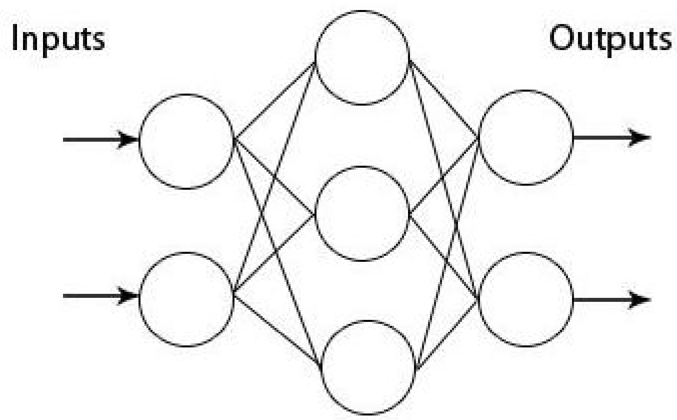
Input nodes (or units) are connected (typically fully) to a node (or multiple nodes) in the next layer. A node in the next layer takes a weighted sum of all its inputs.

Advantages:

- Guarunteee convergence when linearly separable.
- Very fast on test data.

The perceptron algorithm is a gradient descent method, but dosen't get stuck in local maxima -- either converges to global optimum or never converges.

Multilayer feed forward conventionally looks like:



Here there are 3 types of layer;

1. The input layer. Each input is passed through a linear activation function (f(x) = x) to the first hidden layer.
2. The hidden layers. There can be any number of hidden layers with any number of neurons in each layer. Each of these layers performs as the Perceptron described above does, passing a weighted sum of their inputs through some activation function. Typically the activation function is Sigmoid, Hyperbolic Tangent, etc. as this allows each hidden layer neuron to output some non-linear function of its inputs.
3. The output layer. These are neurons that take the inputs from the last hidden layer and combine them by weighted sum to produce some output. If you are solving a classification problem, there could be a neuron for each output class using 1-to-N encoding of the output - then by using a Sigmoid activation function [0, 1] the MLP can output a certain class when that class's output neuron outputs ~1. For regression, a linear activation function can be used for each output variable, allowing the MLP to predict outputs in the same range as the real data.

Model Parameters

In perceptrons hidden layers is one, ensemble =11, monotone=1, bag= TRUE

Results

The csv file submitted has dataset made up of just the last five holes for each drill bit. The RMSE values of all 4 drills are written

Model comparison

The two models used are Artificial Neural networks and Single-layer Neural Networks(Perceptrons)

In case of perceptrons RMSE of last five holes for the testing set
Drill11 = 0.788
Drill12= 0.304
Drill13 = 0.8173
Drill14= 0.6589

In case of ANN RMSE of last five holes for the testing set

Drill11 = 14.15
Drill12= 15.23
Drill13 = 15.66
Drill14= 13.37

We find the median RMSE for Artificial neural network for is 14.6943

And the median RMSE for Perceptrons is 0.7237591

We compared these two models which are based on same mathematical background and we find out that the RMSE for perceptrons is lower by a large margin. Hence the model finally used is Perceptrons.