## Problem Definition

During my undergraduate years I took **METE230-Material Science,** as my restricted elective. In that course professor explained us the aim of the course, as picking a right element for your material which means picking a right element for the condition in which your material will be used. So I decided to solve, for which conditions usage of Iron(Fe) is appropriate. I created a data set which consist of 75 training data and 25 test data. Attributes and values are as provided below:

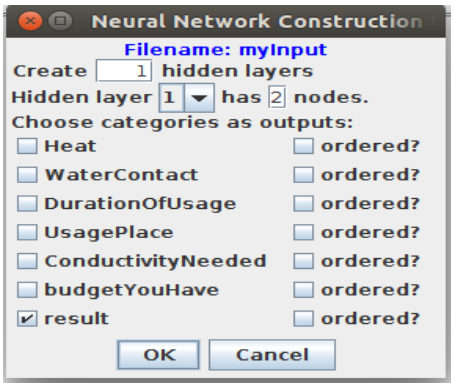
Heat=[low, normal, high]
WaterContact=[yes, no]
DurationOfUsage=[short, normal, long]
UsagePlace=[UnderWater, Land, Space]
ConductivityNeeded=[resistive, semi, high]
budgetYouHave=[high, low]
result=[yes, no] //label

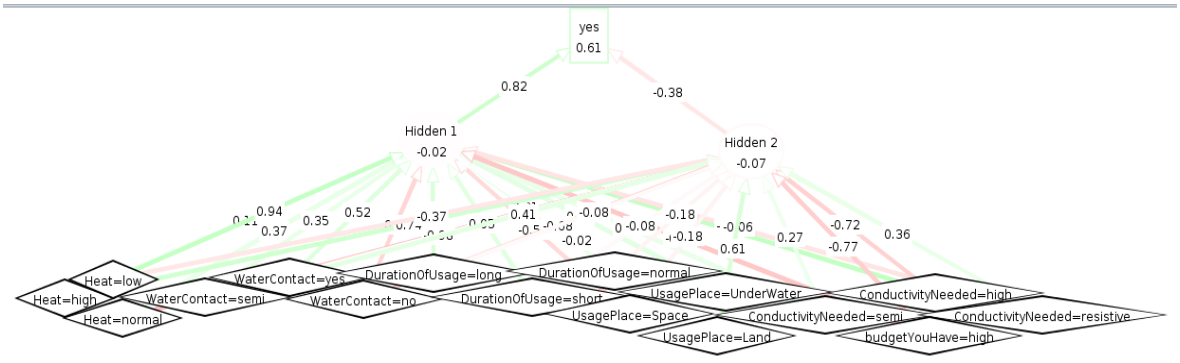
I think attributes are self-explanotary by looking at these attributes I decide whether usage of Fe is appropriate for this material or not. I gave the data set to my friend to label it. He is a 4th year student at **Department of Metallurgical and Materials Engineering.**

I used AISpace Neural applet to solve the problem. The result that I got and how I used this applet is provided below.

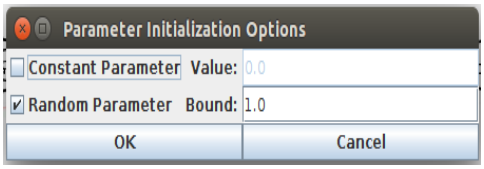
## How I used AISpace Neural Tool.

After uploading the data("myinput") that I provided. This tab will appear as you can see I used 1 hidden layer with 2 nodes. After you click Ok, it will automatically generate for you a network as I provided below. The weights of edges are randomly initialized, you can change it by clicking on **Neural Options** and selecting **Parameter Initialization Options.** I left it as randomly initialized.

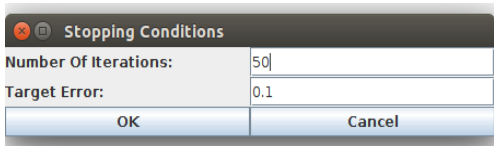


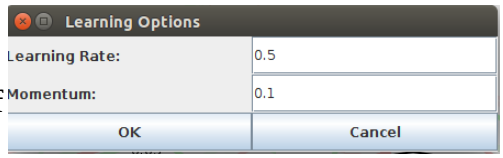
As you can see here I left it as randomly initialized, I also tried **Constant Parameter Value** option in order to get same result when I started everything from scratch but it didn't give satisfactory result. So I left it randomly initialized.



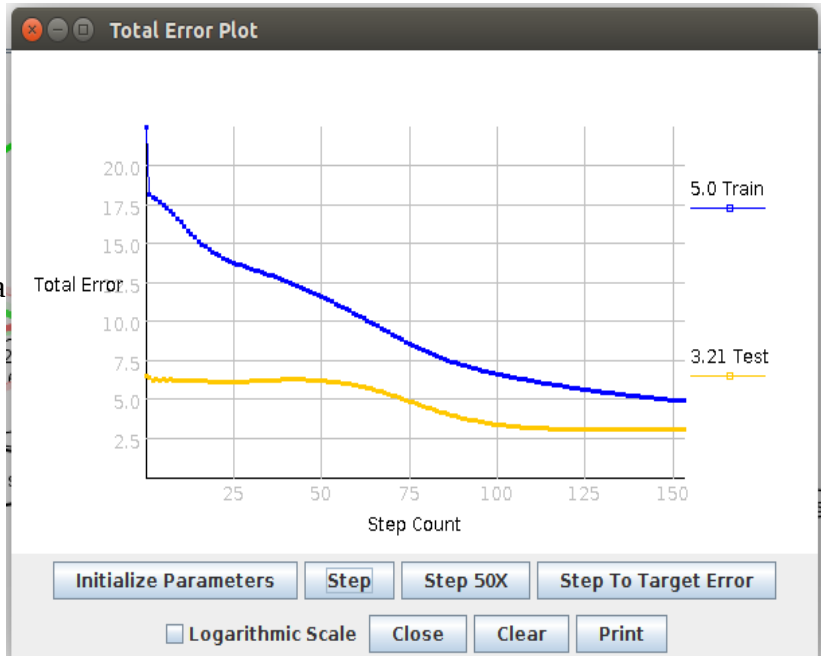

I did not change **Stopping Condition**. I left it as default.



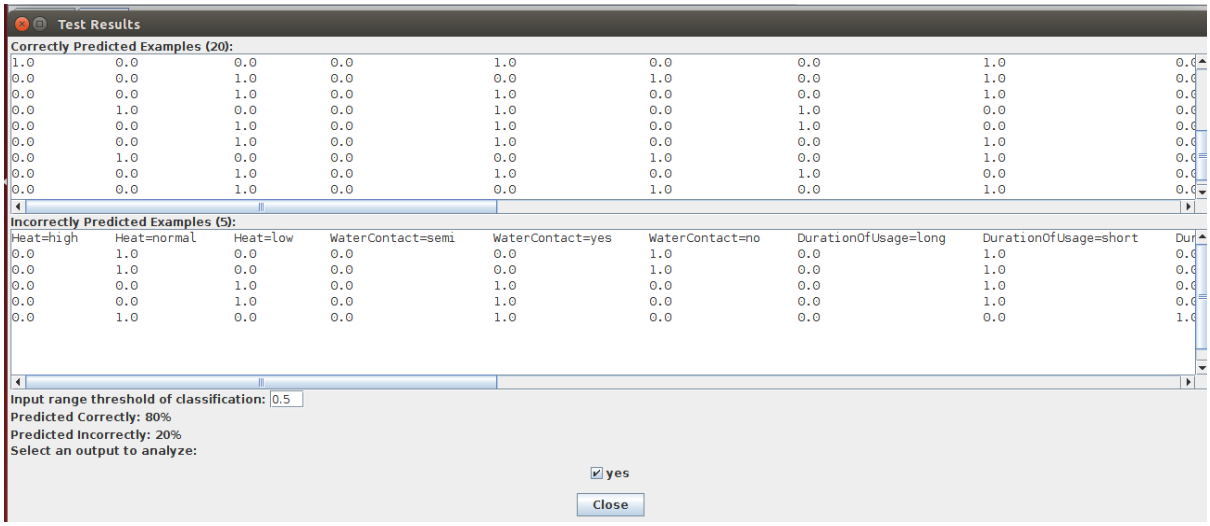

But I changed **Learning Options,** I changed **Learning Rate** from **0.2 to 0.5** of course it gave me sometimes fluctuating results but usually results were more satisfactory than results of 0.2 . Also I made **Momentum 0.1** in order not to stuck in local minima.




During the learning process I observed the error plot of training data and test data. I noticed that at some point training error and test error are stabilized. After continuing a little bit I saw a little increase in test data's error and a little decrease in training data's error. I concluded it as the network started to over-fit the training data so I stopped the process and saved solution in **part3Solution.xml**.




I checked how my network performed on the test data and got the result as provided below




.

After saving the solution I wanted to be sure in my conclusion and continued to run the learning process and I got the plot as provided . As you can see my test data error continued to increase while training data error continued to decrease. Which clearly shows that learning process started to over-fit the training data.