Inversion of Neural Networks:
A Solution to the Problems Encountered by a Steel Corporation

Ashima Dua
Prof. Amar Gupta
Submitted: May 2000
AUP Requirement for BS in 6-3

Acknowledgements

There are many people who helped me on this project. I would like to take the time to thank them for doing so. First Prof. Gupta, my advisor has been a great source of direction by telling me where to look to find my information. Second Ravi Sarma and Ashish Mishra, two student also in the Data Mining Group, have both spent time helping me understand the advanced math which is involved in inversions and matrices as well as the intricacies of matlab. Finally, Bao-Liang Lua a researcher from Japan who's paper was published about inversions was helpful in providing me literature in understanding the problem and solution to the inversion problem. Without these people my project would have never been completed. Thank you all for everything. It was greatly appreciated

Abstract

The productivity of one Steel Corp. is lower than it should be because the Corp. experiences crisises in their blast furnace. This happens when the temperature of the hot metal in its furnace dips lower than the allotted range of 1400 to 1500 degrees Celsius. To remedy this situation the workers of the Steel Plant increase the inputs to the furnace as soon as the crisis begins and wait a few hours until the hot metal temperature increases to resume steel production. Trying to prevent these crisises, the blast furnace of the Steel Corp. has been modeled as a neural network. This paper shows how inverting this network and providing it with a given output/hot metal temperature produces the required inputs/amount of the inputs to the blast furnace which are needed to have that output. Inverting neural networks produces a one to many mapping so the problem must be modeled as an optimization problem, which needs to be minimized. Solving this optimization problem produces the desired inputs, but this paper shows the results are not perfect and there are errors associated with. Therefore the inversion problem is successfully solved when it is minimized but there are other studies, which must be conducted to try to reduce the error associated with the result and even find alternative methods to the problem.

Steel Production

Steel, an alloy of iron and carbon is widely used in the world as a medium for making parts of various objects. From cooking utensils to bike parts, steel has an essential role in the items people use everyday. Ancient people discovered the art of making steel as early as 3000 BC. They found if they heated a mass of iron ore and charcoal in a forge or furnace with a forced draft, the ore was reduced to metallic iron with some slag and charcoal ash. If the metallic iron was removed from the furnace and beaten so that all the slag was driven out they could then use the material to weld into any shape or utensil. This was the prehistoric form of steel. In the 14th century humans one again refined this process of steel making by employing the usage of furnace to smelt the iron. This helped increase the amount of steel, which could be produced and also allowed the draft to the iron to be controlled. Today's modern blast furnaces, which make steel, are just a refinement of the furnaces used then.

These days steel is produced in huge steel mills and at a mass scale. Many companies specialize in this field and focus specifically on it. The most important part of the production is the blast furnace. It is the place where the molten iron more commonly known as Pig Iron is produced. This is the raw material from which steel is produced. The main purpose of the blast furnace is to remove the oxygen from the iron and create the molten material. [3]

The steel making process begins with three basic materials, limestone, iron ore, and coal. The coal is first heated in coke ovens to produce the coke. This process is called carbonization and produces a gas that is used to fuel other parts of the steel plant. Once the coke has gone through this procedure it is pushed out of the oven and placed aside to cool. Simultaneously while this process is taking place, iron ore and limestone are heated in a sinter place. [3] This is a moving belt where the materials are ignited helping them fuse together to form a porous material known as sinter whose main purpose to speed up the process in the blast furnace. After both the sinter and coke are produce the steel making process shifts to the blast furnace.

In the blast furnace, pellets of coke, and iron ore are added to the top by a conveyor belt. From the bottom of the furnace hot air of temperatures over 1400 degrees Celsius is blasted through nozzles that are called tuyeres. The oxygen, which is present in the air combusts with the coke to form carbon monoxide. [10] This process generates a tremendous amount of heat in the furnace. Sometimes oil is also blasted into the furnace with the air to ensure proper combustion. The CO gas, which is produced then, flows through the entire furnace and removes the oxygen from the iron ore as it passes through. What is left behind is pure iron. Next the heat from the furnace assists in melting the iron turning it into a liquid form. On top of this molten iron floats the impurities of coke and iron called slag and is removed at various intervals.

Once the hot metal is heated to a correct consistency it flows into torpedo ladles, which are, specially constructed railway containers used to transport the iron still in its liquid form to the steel furnace. At the steel furnace the molten material is passed through a caster and cooled into slabs or rolled into sheets. This is the finished steel and can then be shipped to manufacturing plants to be made into anything. What is important to note in this production process is that it should not be stopped. In other words the blast furnace must always stay at the consistent temperature and not be allowed to cool. If it were, this would cause damage to its lining and thus cause impurities in the molten iron. [3]

Figure 1 below outlines the entire steel making process.

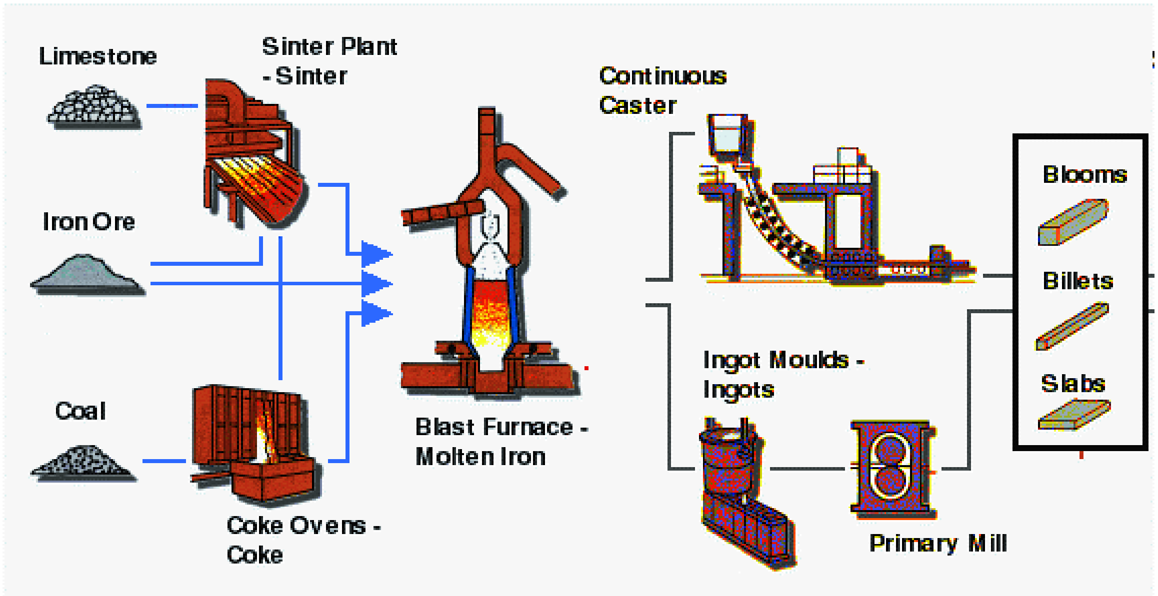

Figure 1: Steel Production.

Steel Corp.

Steel production is a huge industry with many competitors. One such company located in South Asia is the biggest private sector company of its country. What originally began as a modest company with two small furnaces in 1907 has now expanded to include a 3.5 million ton steel plant. When its owner founded it, the company was the first step in the direction towards the country's dream of self-sufficiency. [4] The Steel Corp's goals have always been rooted in the interest of the people and therefore it constantly strives to achieve greater heights and bigger achievements. In recent years, representatives from this Steel Corporation have been in contact with Prof. Amar Gupta of the Sloan Business School at MIT in hopes of increasing their productivity. Prof. Gupta is Co-Director of the PROFIT Program at Sloan and specializes in research projects in the fields of knowledge acquisition, discovery, and dissemination. Currently he heads one research group focused on data mining, which analyzes the problems of different companies. And hopes to provide them quick and reliable solutions. [1]

Data Mining

Data Mining is a branch of Artificial Intelligence which enables companies to discover hidden knowledge which is present in their databases. It employees the usage of AI techniques such as neural networks, fuzzy logic, and genetic algorithms upon mass quantities of data to try to find and understand various hidden trends or relationships within this information. [1] Once these patterns are discovered they can be used to predict such things as management of inventory or detection of fraud. Thus data mining can greatly enhance the product output of any company. It has been estimated that most company databases are never used for anything even though they contain so much vital information about the company's management and productivity. Therefore the upcoming field of data mining will help change that and enable a company to gain better insight behind the reasons for its pitfalls and find suitable solutions to eliminate them.

The Problem

Like any company looking to improve upon its weaknesses, this Steel Corp is greatly concerned about the problems it encounters at its steel manufacturing plants. Therefore with the help of Prof. Gupta research groups they are attempting to remedy many of them. One serious problem, which they face on a regular basis, is the sudden drop of hot metal temperature in their blast furnace. Because of different reasons, every so often the temperature of the blast furnace drops below 1400 degrees Celsius causing a stop in production of liquid iron until the temperature can be brought up again.

The hot metal or liquid iron, which is produced in the furnace, must be of top quality to ensure the proper production of carbon steel. If the hot metal is not at the correct temperature, it will not melt properly and impurities will result in the steel, which is produced, from it. Therefore it is highly important that the temperature of the blast furnace remain at a temperature between the ranges of 1400 to 1500 degrees Celsius. This ensures the iron is properly melted into hot metal and is of the right consistency to make top quality steel. During situations when the temperature of the furnace drops below 1400 degrees Celsius production must be stopped until the temperature of the furnace is once again between the above ranges. To remedy this situation, the workers at the plants must increase the inputs to the furnace to quickly increase its temperature. However this increase of substance must be monitored until the correct temperature in the furnace is achieved. Once this is reached, the production of hot metal can once again continue. One can note this stop of production and monitoring of temperature until the proper range is reached wastes time, which could be used to produce more products. It causes time to waste and takes away from production of steel. In addition simply adding the raw materials to the blast furnace does not automatically just increase the temperature of the furnace. Since the additional coke added to the furnace must be combusted to form CO gas which heats the iron, there is a delay time associated with the addition of materials and the point when proper temperature is once again reached in the furnace. Therefore if there was a way to prevent these crisis's of temperature dropping from happening, our Steel Corp would be able to not only increase their production rates at their steel plants but also prevent the hassle and stress which comes when quickly trying to remedy a crisis situation

Proposed solution

To prevent this phenomenon from happening it has been suggested that the blast steel furnace be modeled as a neural network with the inputs of the network being the inputs to the blast furnace and the output of the network corresponding to the temperature of the hot metal in the furnace. After this is achieved, it is proposed that the entire network be inverted such that a given output or hot metal temperature can predict the inputs, which must be in place for it. Therefore as a means to prevent hot metal temperatures from dropping, if one knows the exact inputs, which are required to maintain a certain temperature of hot metal, one can just monitor the input amounts and make sure they are sufficient. In this way no input will ever go below its recommended dose and as a result the temperature of the hot metal will not drop resulting in greater productivity and less time waste.

Neural Networks

To implement such a solution, one must have a good understanding of neural nets. Neural networks are parallel machines, which model mathematical functions between inputs and outputs. They have the capacity to learn which allows them to gain knowledge through relationships in the training data. Instead of having to program them to perform a certain way, one can just subject them to various data sets and let them induce the relationships between them. [11] Although there are many different types of neural networks the problem outlined above is most easily represented by a feed forward neural network. A feed forward network provides for a mapping between the input space and the output space and has the ability to model complex non-linear functions. This mapping is called forward mapping and provides for a one to many mapping because each output can correspond to a variety of inputs. Shown below is an example of what a feed forward network looks like.

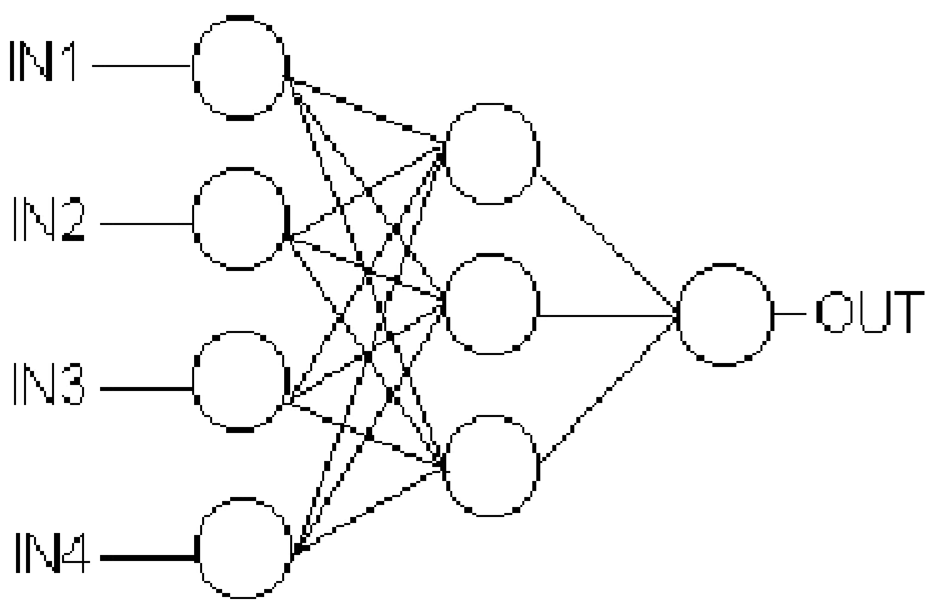

Figure 2: Feed Forward Neural Network

A typical feed forward network consists of inputs, weights and an output. The output of any feed forward network is the sum of the inputs multiplied by the weights. In the picture above, the feed forward network has three layers. An input layer, which has the inputs being fed into it, a hidden layer in the middle and the output layer. The relationship between the input and hidden layer is determined by the weights of the network. As can be seen, a line connects each node in the hidden layer with a node in the input layer. This is because the value of each node in the hidden layer is determined by all of the values of the input layer. Each line connecting an input node with a node in the hidden layer has an associated weight with it. Therefore to determine the value of the topmost node for example, one would multiply all the inputs one by one with their weights, and then add their sums. So the value of the node would be equal to IN1 * Weight 1 + IN2 * Weight 2 + IN3 * Weight3 + IN4 * Weight4. Similarly the values of the other nodes can be found. In the same fashion, multiplying the hidden nodes with their weights and adding them together find the output.

Although not shown in the example above, some neural networks also have what is called a bias, which is a number between 0 and 1. This is associated with a particular layer and is added to the sum of inputs and weights to reach the output. So if the example above if the input layer had a bias, the value of the topmost node in the hidden layer would be IN1 * Weight 1 + IN2 * Weight 2 + IN3 * Weight3 + IN4 * Weight4 + bias.

Figure 2 shows a neural network with 1 hidden layer. Sometimes the network contains many hidden layers and sometimes it contains none. Regardless of how many layers the network contains, the relationship from layer to layer is always the same, each value of the a node in a layer is equal to the nodes of the previous layer multiplied by the weights with the addition of a bias if one is present. Therefore in general we can express the output (y) of the forward mapping of the network as a function of its weights (w) and input (x). as $y = f(w,x)$.

Inverting the Neural Network

As already stated inverting such a neural network would entail figuring out the inputs x' which correspond to a given output y'. This means the output space will now be mapped to the input space instead of being mapped from the input space. However the problem associated with this representation is that this mapping will result in a one to many mapping between the output and inputs. This is because in neural networks, different inputs can yield the same output. Therefore there is no close-formed expression for the inverse mapping of such neural networks.

However that is not to say that there is no way to solve this problem. Infect many methods have been employed to solve the problem at hand. One of the first methods of inverting feed forward neural networks was discovered separately by R. L. Williams and A. Linder and J. Kinderman. Both of these people developed an iterative algorithm for inverting the feed forward network. In this algorithm, the inversion problem is set up as an unconstrained optimization problem and is solved by a gradient descent method, which is very similar to the back propagation algorithm. (The back propagation algorithm of neural networks has two main parts, the sigmoid function and the gradient descent using error propagation. The main idea behind back propagation is to make a large change to a particular weight of the net if the change leads to a large reduction in the errors observed at the output nodes. For each sample input you consider each output's desired value, its actual value and the influence of a particular weight. A big change to the weight makes sense if that change can reduce a large output error and if the size of that reduction is substantial.) [6]

Next Jordan and Rumelhart were able to discover an inversion approach which helped solve inversion problems for kinematics involving redundant manipulators. Their method involved two steps. In the first the network to be inverted is trained so that is approximated the forward mapping. After achieving this network is then connected to another network in series so that a particular inverse solution is obtained by an identity mapping across the entire network. [7] [9]

A third solution to the inversion problem was discovered by Lee and Kil. They proposed a method for computing inverse mapping by combining a local update rule using the Lypunov function and the relocation rule using the predefined information on forward mapping. [5]

But perhaps the easiest and most straightforward method to inverting feed forward networks is through the use of mathematical programming techniques. Bao Liang Lu, Hajime Kita and Yoshikazu Nishikawa have been working on implementing such methods during their research in Japan. Their idea consists of formulating the inverse problem as a non-linear programming problem, a separable problem or a linear programming problem depending on the structure of the neural network. One should note the advantage of employing this method as opposed to the others is that many times network inversions for multi-layer perceptrons or radial basis functions can be obtained by solving the separable programming problem with the easier method used for linear programming problems. [7] This paper will utilize the non-linear programming approach to converge upon a solution for the inversion problem.

Methodology of Implementation

We know there are many different inputs x which correspond to a particular output y1. Therefore since there is no closed expression for inverse mapping, it does not make sense to simply find all of these inversions which would satisfy the output. This would be very time consuming and perhaps impossible depending on the number of solutions. Therefore it is necessary to restrict the solution to only specific solutions. This can be accomplished by solving an optimization problem by non-linear programming methods. Specifically the non-linear programming inversion problem can be stated as below:

$$
\begin{aligned}
&\text{Minimize } p(x) \\
&\text{subject to} \quad f(w, x) - y1 = 0 \\
&\qquad\qquad\quad\; a < x < b
\end{aligned}
$$

Here f(w, x) is the representation of the forward mapping of the feed forward network, x refers to the inputs and w to the weights. y1 is the given output of the inversion problem and a and b refer to the range the inputs must fall between. p(x) is called the objective function and it is the function which must be minimized. x refers to the vector which satieties all the constraints that are stated and provides a solution to the NLP problem.

The purpose of p(x) is to express what kind of inversions are to be computed. For example p(x) can be set such that the inputs which satisfy the inversion and those which are the lowest of all the possible solutions. In this case the answer to the inversion would be equal to xmin. In addition the inversion can be found which is closest to corresponding inputs at the present time. This inversion is called the xnear inversion where p(x) will be set to ||x - c||^2. C refers to a reference point already part of the present inputs. It can be the average of all the inputs for example. Finally, p(x) maybe set to xi or -xi in which case we will want to find the inversion in which xi is the value of the ith element and is the smallest among all the inversions which match to the output y1.

In this paper, the when picking our objective function we must keep in mind that the point to our solution is to keep the hot metal temperature within a specified range without wasting too much time or resources it makes sense to determine the inputs which would sustain that temperature and are closest to the actual inputs in place right now so that the least amount of resources are utilized saving money and overall overhead.

Methods

Actually implementation of the solution to this inversion problem was done using mat lab. Since the involvement of the data-mining group with this Steel Corp has been an ongoing one, neural networks have already been set up which model the blast furnace of the steel plant. As the time when the inputs are added and the time when they affect the output varies there are different neural networks that have been set up to take into account this delay. Some neural networks have a four-hour delay from the time additional input is added and the effect is seen. Others may have a two or eight hour delay. Therefore by utilizing these networks, it was possible to solve the NLP problem as stated above. Since the blast furnace is constantly being fed with different materials it made sense to model the NLP problem such that the inversions found would be those closest to the actual inputs at the present time. Therefore it was decided to solve the xnear inversion problem where p(x) was set to ||x-c||^2. (c refers to the value of the input at the present time). Although when initially the project was undertaken it was assumed this minimization this optimization problem involving inversions would have to be calculated either by writing a program or utilizing software packages, it was determined after research and partial implementation that mat lab contains the capabilities to perform this task.

Therefore the actual inversion problem, which was solved, was:

minimize p(x) where p(x) = ||x – avg of actual_inputs_of_neural_net||^2
subject to f(w, x) - y1 = 0
$0 < x < 1$

Lets take a look at an example problem and see how that would work.

Examples & Results

One of the networks which was created to model the blast furnace was once which involved a four hour delay between the addition of the inputs and the actual effect in the temperature of the hot metal. If this network is loaded up into mat lab, one can feed it some inputs and see what the output of those inputs would be four hours from now. Please note all the inputs have been normalized between 0 and 1 as has the output hot metal temperature. Therefore we are not dealing with the actual quantities of the input but rather a measure of them relative to the other inputs. These inputs correspond to 18 inputs, which are placed into the blast furnace to create the liquid iron. These include the coke, iron ore, limestone, as well as other materials, which are necessary to created the liquid iron that is made in the blast furnace. Let us assume an example of the sample inputs, which are placed into the blast furnace, correspond to those listed below.

Input number	Input Value
Input 1	.1
Input 2	.2
Input 3	.1
Input 4	.1
Input 5	.1
Input 6	.1
Input 7	.4
Input 8	.6
Input 9	.3
Input 10	.2
Input 11	.1
Input 12	.5
Input 13	.6
Input 14	.9
Input 15	.6
Input 16	.1
Input 17	.3
Input 18	.4
Input 18	.2

Table 1: Sample inputs into the four-hour delay neural network

The output of these inputs is .3670 which corresponds to a temperature between 1400 and 1500 Celsius inside of the blast furnace and is the temperature which is read when the hot metal is tapped every so often as it is being produced.

Now let us assume we want the temperature to change four hours from now so that its normalized value is .6622. We know we must adjust the inputs but we are not sure what exactly needs to be adjusted or how much adjustment must take place. Therefore we solve the inversion problem as stated above in matlab. The inversion problem is solved using the command fmincon which is part of the optimization toolbox of matlab and is designed to assist in such problems. We should also keep in mind we want the inputs to be as close to the actual values which they are at the present time so that we know how much we must add to the blast furnace to achieve such results. Shown below are the results which correspond to the output .6622 of the neural network.

Input number	Input Value
Input 1	.2
Input 2	.4
Input 3	.2
Input 4	.3
Input 5	.2
Input 6	.3
Input 7	.8
Input 8	.9
Input 9	.3
Input 10	.1
Input 11	.2
Input 12	.9
Input 13	.9
Input 14	.9
Input 15	.4
Input 16	.2
Input 17	.5
Input 18	.8
Input 18	.2

Table 2: Sample inversions corresponding to the output temperature of .6622

Thus we see if we want the blast furnace to have the desired temperature which is stated above we must increase the inputs to the amount shown in Table 2. Simply subtracting the values in Table 1 from their corresponding values in Table 2 will tell the Steel Corp engineers how much of each input to add to the blast furnace. Such a process can be repeated for any of the neural nets in the Steel Corp's environment.

What now needs to be checked is if these inputs are actually fed into the network what the output hot metal temperature will be. Thus the neural network is simulated so that these are the inputs. What results as the output is not the number .6622 but rather .8301. We see our inversion has not produced the desirable results but rather has an **error value of .1679** associated with it where the desired output and actual output differ.

Second Example

Suppose now we want the output of the blast furnace to be .5439 after a duration of four hours. Solving the optimization problem with the new requirement the results obtained are shown below.

Input number	Input Value
Input 1	.3
Input 2	.4
Input 3	.1
Input 4	.3
Input 5	.1
Input 6	.1
Input 7	.3
Input 8	.4
Input 9	.5
Input 10	.2
Input 11	.4
Input 12	.3
Input 13	.6
Input 14	.3
Input 15	.6
Input 16	.1
Input 17	.2
Input 18	.3
Input 18	.2

Table 3: Sample inversions corresponding to the output temperature of .5439

Once again to see what the results of such inputs to the blast furnace would cause the hot metal temperature of the furnace to be, these inputs are simulated into the neural network and the resulting temperature is once again not .5439 but instead .7903, resulting in an **error value of .1536.**

Interpretation & Comparison of Results

What do the results above mean? It has been shown that to obtain a certain temperature it is necessary to change the value of the inputs to the blast furnace and then wait until the desired temperature is reached, but the inversions are not completely accurate. They do not fully reach the desired temperature which is sought after. In one case, example 1, the temperature becomes hotter than what was originally wanted, and in the second case, example 2, the same phenomenon is observed.

In order to see if the results produced should be disregarded or taken as the best results which can be obtained, it is necessary to see what the error ranges are for others in the industry trying to invert neural networks. Since the method employed in this paper is similar to the method used by Bao Liang Lu of Japan it makes most sense to compare the results here with the results he obtained.

When working with a 6- layer neural network, Lu attempted to invert the network and obtain inputs for different desired outputs. Based on those trials, there were different errors associated with the actual output desired and the output produced by the inversions. The **lowest error** he experience **was .066** and the **highest error recorded was .455**. Therefore the errors obtained above of .1679 and .1536 as examples are within the range experienced by Lu.

It must be noted however that the errors of Lu quoted above were for a 6 layer neural network while the ones cited for the work done above are for a two layer network. Thus in comparison the errors above are slightly worse than should be expected because one would assume it is easier to invert a less complex neural network. Therefore although the error falls in the lower end of the error scale which Lu experienced in his research, it must be noted that in proper comparison they are less desirable than the results reached by Lu.

According to the error analysis done above the results are still within range in which the inversions obtained can be considered correct. Therefore there is no need to disregard the results but instead the focus should be upon improving them.

Problems Encountered & Resolved

The inversion problem had many obstacles which had to be overcome in order to reach a suitable answer. First understanding how a neural network works is essential to figuring out how to invert it. The initial research involved figuring out how a neural network is set up and what it entails to train it and calculate appropriate inputs from its output. Once that was properly understood it was necessary to tackle the inversion problem. Although this seems like a logical solution to make predictions easier it is not a very widely studied topic and thus it was hard to obtain papers on it. However after spending some effort it was possible to obtain some thus the implementation process outlined in them was begun. Initially it was attempted to try and invert the weights of the neural network to see if that would work in providing accurate results but it was soon discovered this was the wrong approach to take. The results were completely off.

Thus it was back to researching more papers on how to do this and finding other methods of tackling the problem. Many of these papers discussed gradient methods which were not fully understand so it was necessary to locate someone who knew a little more about what was involved with them. In the meantime the search for much more simpler solutions to the inversion problem was continued. Finally a paper which explained how to invert feed forward networks using linear as well as non linear programming was found. This seemed like a big break because the paper was fairly thorough in the explanation of the methods involved.

Although the paper was fully understood it was a starting point to the solution of the inversion problem. With the help of Ravi Sarma and Ashish Mishra I was able to understand the theoretical math background which was needed to fully comprehend the statements made in the paper. It was advised by both of them to begin implementation in matlab. However not knowing the environment or set up of matlab very well it was hard to know what exactly to do. Thus it was necessary to be brought up to speed with the project by Ravi Sarma who has been developing neural networks for the Data mining Group for the past year. With this newly gained knowledge it was possible could finally test out the method of inversion and obtain the results shown above.

Results of Other Data Mining Members

As stated earlier Prof. Gupta's entire Data Mining group has been involved with different aspects of providing a solution to the Steel Corp. Some of the work done by other team members is relevant to the work outlined in this paper.

Ravi Sarma, Ashish Mishra and Michael George are all focusing on the prediction side of the project and working on trying to make the network more useful in providing and accurate measure of the results. This project does not directly deal with those issues so there is no real overlap in terms of work, however, the networks used in this paper are made by people from this group. In addition the inputs which are relevant and must be increased to change the temperature of the blast furnace is determined by these three people.

Mu'taz Qubaj and Danny Lai have been working on other control strategies of the network similar to this paper. They are also interested in finding the inputs of the network when given a specific output and have been following different approached to achieve that result Mu'taz has been focusing his work on decision trees and hoping to find a solution for the problem in that way, while Danny has been researching genetic algorithms to use one as a controller which when given a future time and specific hot metal temperature will give the required inputs to achieve that.

Danny has not been able to reach conclusive results yet with his algorithm. He suggests a more powerful set of genetic algorithm tools be utilized if this project was to conducted again. Although his research shows genetic algorithms are a very good candidate for a solution to the crisis situation experienced his implementation did not show complete results. Therefore more research needs to be conducted in this field to fully understand the capacity of genetic algorithms in respect to this situation.

Future Research

Although a method to solving the inversion problem was discussed here, there must be other methods to solve the problem. It would be interesting what they are and apply them. Perhaps there is still an easier way to obtain the solution to the above crisis. In addition if the Steel Corp actually uses these numbers and applies them to their blast furnaces it is necessary for them to understand there are errors associated in this method and they can not rely on it blindly to fix their crisis problems. If they do the Steel Corp will once again find themselves experiencing delays in production.

In terms of the results shown above, it has been determined there are errors associated with the inversions produced. Therefore as a future expansion of this project, one could try to minimize such results. Perhaps there are other software packages other than matlab which are better at solving optimization problems such as the one outlined above and assist in providing a solution closer to the one desired.

In addition, it is necessary to observe how the inversion results would correspond to other types of problems. For example. it would be interesting to see how the optimizations behave with other types of neural networks and multilayered perceptrons.

Conclusion

A Steel Corp in South Asia has been experiencing crisises in their blast furnaces. Specifically from time to time the hot metal temperature of their blast furnace suddenly drops causing a stop in production until that temperature is once again brought up to the standard range of 1400 to 1500 degrees Celsius. Sometimes this process can take hours because there is a delay between the time inputs is increased in the blast furnace and the time it affects the hot metal temperature. Therefore this causes a delay in production and wastage of money and time. To prevent this from happening it is hoped one can predict the inputs which are required to maintain a desired temperature a certain number of hours from now so that they can be added to the furnace right now and prevent the crisis from happening.

To achieve this the blast furnace must be modeled as a neural network. Then to predict the inputs, the network must be inverted. Solving the optimization problem listed above and explained will result in the inputs which will satisfy that particular output and if the inputs are increased to that value the crisis may be prevented. However the results are not 100% accurate and therefore it will still be necessary to monitor the blast furnace to make sure the hot metal temperature of the furnace does not fall below the recommended range.

References

1) http://scanner-group.mit.edu/htdocs/

2) http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml

3) http://www.steel.org.uk/makstl.html

4) http://www.tatasteel.com

5) S. Lee and R. M. Kill, "Inverse mapping of continuous functions using local and global information" c. 1989

6) A. Linden and J. Kindermann, "Inversion of multilayer nets" c. 1989

7) Bao-Liang Lu, "Inverting Feedforward Neural Networks using Linear and Nonlinear Programming" c. 1990

8) Lida, Osamu and Ushijima, Yuichi and Toshiro, Sawada. "Application of AI techniques to blast furnace operations." October 1992.

9) R.J. Williams, "Inverting a conncetionist network mapping by backpropagation error" c. 1986

10) Sarma, Ravindra K. "Using Neural Networks to Predict the Hot Metal Temperature of the "G" Blast Furnace at Tata Steel" (MIT, 1999)

11) Winston, Patrick Henry, Artificial Intelligence: Third Edition, c. 1992