Paper No: (??)– 37th MATADOR Conference

Modelling of Direct Metal Laser Sintering of EOS DM20 Bronze Using Neural Networks and Genetic Algorithms

A. Singh[1], D.E. Cooper[2], N.J. Blundell[3], G.J. Gibbons[4], D.K. Pratihar[5]

[1] Mr. A. Singh, 08mf1025@iitkgp.ernet.in, Dept of Mechanical Engineering, IIT Kharagpur, West Bengal, India.
[2] Mr. D.E. Cooper, d.e.cooper@warwick.ac.uk, WMG, University of Warwick, Coventry, CV4 7AL, UK.
[3] Mr. N.J. Blundell, N.J.Blundell@warwick.ac.uk, WMG, University of Warwick, Coventry, CV4 7AL, UK.
[4] Dr. G.J. Gibbons, g.j.gibbons@warwick.ac.uk, WMG, University of Warwick, Coventry, CV4 7AL, UK.
[5] Dr. D.K. Pratihar, dkpra@mech.iitkgp.ernet.in, Dept of Mechanical Engineering, IIT Kharagpur, West Bengal, India

Abstract: An attempt was made to predict the density and micro-hardness of a component produced by Laser Sintering of EOS DM20 Bronze material for a given set of process parameters. Neural networks were used for process-based-modelling, and results compared with a Taguchi analysis. Samples were produced using a powder-bed type ALM (Additive Layer Manufacturing)-system, with laser power, scan speed and hatch distance as the input parameters, with values equally spaced according to a factorial design of experiments. Optical Microscopy was used to measure cross-sectional porosity of samples; Micro-indentation to measure the corresponding Vickers' hardness.

Two different designs of neural networks were used - Counter Propagation (CPNN) and Feed-Forward Back-Propagation (BPNN) and their prediction capabilities were compared. For BPNN network, a Genetic Algorithm (GA) was later applied to enhance the prediction accuracy by altering its topology. Using neural network toolbox in MATLAB, BPNN was trained using 12 training algorithms. The most effective MATLAB training algorithm and the effect of GA-based optimization on the prediction capability of neural networks were both identified.

.

Keywords: Direct Metal Laser Sintering, Genetic Algorithms, Neural Networks.

1. Introduction

Direct Metal Laser Sintering (DMLS) is an Additive Manufacturing technique, capable of constructing metallic components by depositing and selectively melting successive layers of metal powder [15]. Fig 1 shows the principle of the process, with raw material fed into a processing area by a re-coating mechanism.

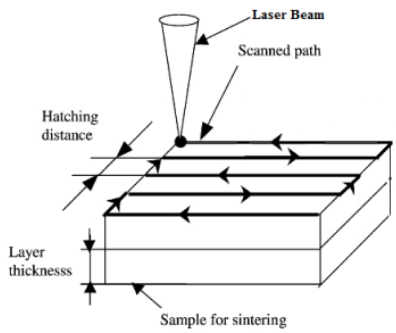


Fig.1. Schematic of DMLS Parameters [8]

In recent years, neural networks have become very useful tool in the modelling of input–output relationships of some complicated systems [1]. They have excellent ability to learn and generalize (interpolate) the complicated relationships between input and output variables. There are different training schemes for these neural networks [2]. Counter Propagation Neural Network (CPNN) and Back Propagation Neural Network (BPNN) are two designs of neural network, with the approximation efficiency of each varying with the type of data used [11],[16]. Radial basis function network [5] was also used to check and compare the accuracy of modelling, but it did not yield appreciable results. Margaris et al. [3] discussed the implementation of CPNNs. Network optimization concerns the technique used to achieve the optimum number of hidden neurons in a CPNN [4]. BPNNs have been used for a variety of modelling tasks for complex systems [6].

The properties of components produced by DMLS depend heavily on fine control of the input parameters, so identifying the precise effect of each parameter is crucial. Wang et al. [7] explored the part shrinkage of samples manufactured by Selective Laser Sintering (SLS), by varying seven process parameters. An experimental design approach was used towards SLS of low carbon steel by Chatterjee et al. [8], where the parameters used were layer thickness and hatching distance, to consider the effects of density, hardness and porosity of sintered components. Ning et al. [9] and Wang et al. [7] used models to intelligently select the parameters for modelling the DMLS process. One of the notable studies related to the application of soft computing towards laser sintering included the estimation of build time [10]. Comparisons between the applicability of BPNN and CPNN towards manufacturing process (TIG welding) have been demonstrated by Juang et al. [11]. Apart from SLS, multiple designs of neural networks have been used in the past to model different aspects of various other manufacturing processes. Lu et al. [12] worked on modelling the Laser Engineered Net Shaping (LENS) process, where BPNN-based models were applied to control the deposition height of the prototype. For laser welding, Lim and Gweon [13] investigated the application of neural networks in estimating joint strength for pulsed laser spot welding. Balasubramanian et al. [14] discussed about the performance of BPNN for the modelling of stainless steel butt joints.

In this study, a unique comparison of CPNN and BPNN had been carried out in the context of modelling the sample hardness and cross-sectional porosity. To have a better estimation of predictive capability of the two designs, they were trained and tested with three unique data sets. Using MATLAB, several BPNN training algorithms were tested. The effect of a binary-coded GA (Genetic Algorithm) was also studied towards enhancing the predictive capability of a BPNN.

2. Methodology

The following steps were followed to carry out the experiments:-

- Sample production by ALM.
- Metallographic sectioning & polishing.
- Visual examination and hardness tests to obtain the desired output values to be fed into the Neural Network.

The values of the input parameters were Laser Power (kW): 0.75, 1.00, 1.25, 1.50, 1.75; Laser Scan Speed (m/min): 5.0, 6.5, 8.0, 9.5, 11.0; Hatch Distance (mm): 0.25, 0.50, 0.75, 1.00.

2.1 Tools and Techniques used, and Developed Approaches

After the values for the various experimental runs were obtained, the entire data was assembled in input-output pairs. A total of 99 such pairs were obtained, which were divided into three sets of 89 training values and 10 testing values. These values were then used in the neural network-based modelling task. An artificial neural network (ANN) is a mathematical or computational model that is inspired by the structure and/or functional aspects of biological neural networks. The CPNN is a hybrid network, consisting of an outstar network and competitive filter network. The hidden layer is a Kohonen network, which categorizes the pattern that was input. The output layer is an outstar array, which reproduces the correct output pattern for the category. The second kind of neural network used in the study was a BPNN, the topology of which is shown in Fig 2. The numbers of nodes in the input and output layers are N_i and N_o, respectively. The use of a larger number of hidden nodes can potentially improve the accuracy and convergence of the back-propagation (BP) algorithm at the cost of computational processing time [2].

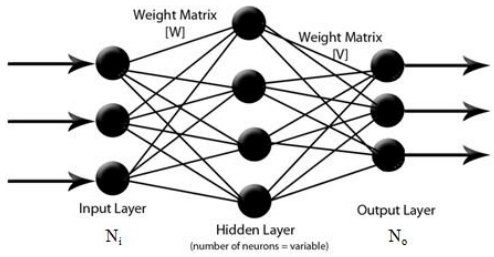


Fig. 2. BPNN Architecture

In the tests carried out, $N_i = 3$ and $N_o = 2$, while the number of neurons in the hidden layer was varied and tested with the help of a Genetic Algorithm (GA). Juang et al. [11], Goh et al. [6] and Margaris et al. [3] discussed the network structures and training schemes in detail. Pratihar [2] discussed the training schemes of BPNN in detail.

A binary-coded GA was used to optimize the topology of the network. Of the two types of errors described above, the GA tends to minimize the training error by choosing the best combination of network parameters, such as number of neurons of the hidden layer 'n_h', coefficient of transfer function of the hidden layer 'a_h', coefficient of transfer function of the output layer 'a_o' (Pratihar [2]). The following steps were used in GA implementation:-

- Creation of random population (size of 100). Each chromosome in the population represents a certain combination of 'n_h', 'a_h' and 'a_o'.

- Fitness evaluation of each chromosome using the mean square error (MSE) of the BPNN [2], after 10000 iterations, keeping the topology represented by that chromosome into account.
- Tournament-based selection [2] was used to select the pool of better chromosomes.
- Single point crossover and Mutation with respective probabilities of 0.9 and 0.09, forming a new pool of 100 chromosomes and indicating the completion of a generation.
- The process was repeated for 100 generations, and the fittest chromosome was finally chosen.

The modelling was conducted in C++, where codes were written for GA optimized BPNN and CPNN. Using neural network toolbox in MATLAB, analysis of 12 training algorithms was carried out for feed-forward network. C++ coding was performed on a GCC compiler (version: Dec 20 1999 15:39:08). Minitab v16 was used to perform the Taguchi L9 analysis.

3. Results and Discussion

Inputs of the neural network were normalized in the scale of 0 to 1. In neural network toolbox of MATLAB, feed-forward networks were developed using 12 different training algorithms, namely *traingd* (Gradient descent), *traingdm* (Gradient descent with momentum), *traingdx* (Gradient descent momentum with an adaptive learning rate), *trainrp* (Resilient BP algorithm), *traincgf* (Conjugate gradient BP with Fletcher-Reeves updates), *traincgp* (Conjugate gradient BP with Polak-Ribiere updates), *traincgb* (Conjugate gradient BP with Powell-Beale restarts), *trainscg* (Scaled conjugate gradient method), *trainbfg* (BFGS quasi-Newton method), *trainoss* (One step secant method), *trainlm* (Levenberg-Marquardt optimization) and *trainbr* (Levenberg-Marquardt optimization with Bayesian regularization). Three combinations of 89 training and 10 testing cases have hereby been referred to as Set-1, Set-2 & Set-3. The number of neurons in the hidden layer was varied from 2 to 17, keeping the number of unknowns (5 × number of neurons in the hidden layer) lower than the number of equations (89 training cases). Tests were conducted for 'a_h' and 'a_o' by individually varying the ranges, and the optimum range was found to be (0.2 to 15.95, in steps of 0.25) and (0.2 to 3.35, in steps of 0.05) for a_h and a_o, respectively. Table-1 shows the optimum values of the parameters: n_h, a_h and a_o.

Table-1: BPNN Results

Set	Optimum Parameters			MGE
	n_h	a_h	a_o	
Set-1	15	8.75	2.10	0.1076
Set-2	15	8.50	2.25	0.1300
Set-3	16	8.75	2.10	0.1676

The Mean Generalization Error (MGE) represents the mean absolute difference between the normalized values of computed and actual porosity and hardness. The training algorithm shown is the same as the *traingdm* algorithm discussed later in this section in the MATLAB results. For the CPNN, the number of neurons in the competition layer was varied from 2 to 89. The network was allowed to train as long as the MSE was converging towards 0. The loop was terminated the moment the MSE started diverging. Upon analysing the final MSE before divergence occurred, the best network topology was chosen. Upon increasing the number of hidden neurons, up to a certain number the pattern was uniform. After that, there were indications of improper (over/under)-training. For all the three cases, the best values for MGE were obtained between 25 to 30 neurons. The BPNN took 10000 iterations (for NN weight modification) converge to the specified MGE, while the CPNN took only 4 iterations. Table-2 shows the best CPNN configuration (lowest MSE-based analysis).

Table-2: CPNN Results

Set	No. of hidden neurons	MGE
Set-1	30	0.1110
Set-2	28	0.1393
Set-3	29	0.1320

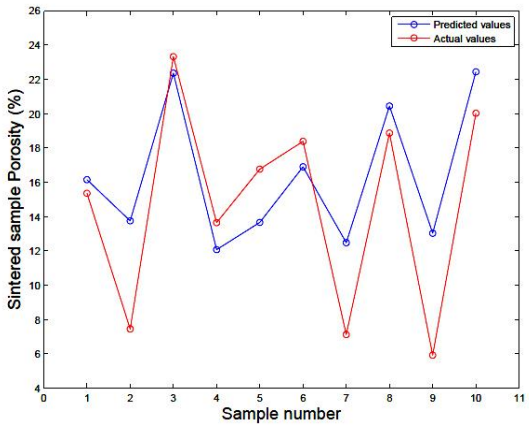
Feed-forward back-propagation neural network was trained using 12 different training algorithms; the results have been compared in this section. As training parameters, the number of iterations/epochs was set to a maximum of 10000. The performance goal, based on the MSE, was set to 0 (zero) and rest all other training parameters were at their default values. For all the networks, the one hidden layer with a *tansig* activation function and one output layer with *purelin* activation function were used. The value of 'n_h' was varied from 2 to 17. The best approximation was identified as the network topology (i.e., the number of neuron in the hidden layer) with the least MSE at the completion of the training process. Table-3 displays the values of MGE for various training algorithms. The number of neurons of the hidden layer for the same network has been indicated in the bracket. The MATLAB training algorithm named *traingdx* turned out to be the most accurate one with an average MGE of 0.1139 on the normalized scale.

Using the Taguchi L9 analysis, laser scan speed was found to have the maximum effect on both the outputs, while laser power and hatch distance were found to have the minimum effect on porosity and mean hardness, respectively. Fig. 3 overleaf, compares the predicted values of the two outputs with their respective actual values for the test cases (using *trainrp* for Set-1)

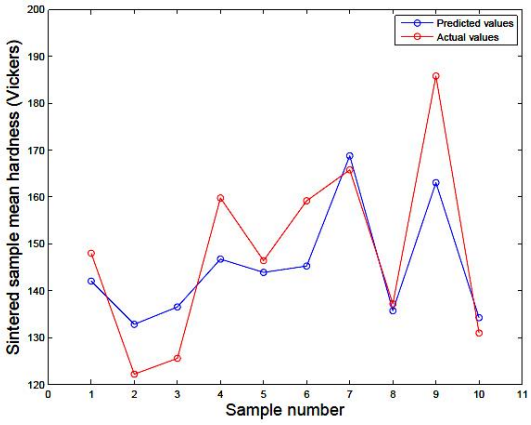
Table-3: MATLAB Results

Algorithm	Set-1	Set-2	Set-3
traingd	0.1334 (17)	0.1304 (14)	0.1798 (8)

traingdm	0.1442 (16)	0.1336 (12)	0.1392 (5)
traingdx	0.1006 (10)	0.1387 (11)	0.1024 (16)
trainrp	0.0843 (10)	0.1255 (13)	0.1920 (10)
traincgf	0.0896 (10)	0.1066 (13)	0.1605 (9)
traincgp	0.1016 (17)	0.1404 (14)	0.1636 (13)
traincgb	0.1473 (16)	0.1170 (12)	0.1845 (14)
trainscg	0.1044 (11)	0.1329 (13)	0.1539 (11)
trainbfg	0.1056 (16)	0.1357 (14)	0.1577 (15)
trainoss	0.1249 (16)	0.1238 (14)	0.0964 (14)
trainlm	0.1298 (13)	0.1457 (7)	0.1659 (12)
trainbr	0.0990 (9)	0.1399 (14)	0.1594 (13)



(a)



(b)

Fig. 3. Actual vs. Predicted values of (a) porosity and (b) mean hardness for the test cases

4. Conclusion

The MATLAB training algorithm named *traingdx* was found to be the most accurate one, with an average MGE of 0.1139 on the normalized (0 to 1) scale. CPNN and GA-optimized-BPNN had average MGE values of 0.1274 and 0.1350, respectively. The training time for CPNN was much less than that for BPNN. Laser Scan Speed was seen to have the greatest influence on both the outputs, using the Taguchi method.

The use of GA-based optimization successfully reduced the MGE for BPNN trained by the gradient descent with momentum algorithm from 0.1350 to 0.1274.

References

[1] J.A. Freeman, D.M. Skapura, Neural Networks: Algorithms, Application and Programming Techniques, Addison–Wesley, New York, 1991

[2] D.K. Pratihar, Soft Computing, Narosa Publishing House, New Delhi, 2008

[3] A. Margaris, S. Souravlas, E. Kotsialos, M. Roumeliotis, Design and Implementation of Parallel Counterpropagation Networks Using MP, Informatica, 2007, 18, 79-102

[4] Y. Dong, C. Sun, X. Tai, An Adaptive Counter Propagation Network, Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (IEEE Transactions), Qingdao, 2007, 695 – 700

[5] J.S. Park, I.W. Sandberg, Universal Approximation using Radial-Basis-Function Networks, Neural Computation 3 (MIT Press), 1991, 246-257

[6] A.T.C. Goh, Back-propagation neural networks for modelling complex systems, Artificial Intelligence in Engineering 9, 1995, 143-151

[7] R.J. Wang, L. Wang, L. Zhao, Z. Liu, Influence of process parameters on part shrinkage in SLS, International Journal of Advanced Manufacturing Technology 33, 2007, 498 – 504

[8] A.N. Chatterjee, S. Kumar, P. Saha, P.K. Mishra, A. Roy Choudhury, An Experimental design approach to selective laser sintering of low carbon steel, Journal of Materials Processing Technology 136, 2003, 151 – 157

[9] Y. Ning, J.Y.H. Fuh, Y.S. Wong, H.T. Loh, An Intelligent parameter selection system for the direct metal laser sintering process, International Journal of Production Research 42, 2004, 183 – 199

[10] J. Mungunia, J. Ciurana, C. Riba, Neural-network-based model for build-time estimation in selective laser sintering, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2009, 995 – 1002

[11] S.C. Juang, Y.S. Tarng, H.R. Lii, A comparison between the back-propagation and counter-propagation networks in the modelling of the TIG welding process, Journal of Materials Processing Technology 75, 1998, 54 – 62

[12] Z.L. Lu, D.C. Li, B.H. Lu, A.F. Zhang, G.X. Zhu, G. Pi, The prediction of the building precision in the Laser Engineered Net Shaping process using advanced network, Optics and Lasers in Engineering 48, 2010, 519-525

[13] D.C. Lim, D.G. Gweon, In-Process Joint Strength Estimation in Pulsed Laser Spot Welding Using Artificial Neural Networks, Journal of Manufacturing Processes 1, 1999, 31 – 42

[14] K.R. Balasubramanian, G. Buvanashekaran, K. Sankaranarayanasamy, Modelling of laser beam welding of stainless steel sheet butt joint using neural networks, CIRP Journal of Manufacturing Science and Technology 3, 2010, 80 – 84

[15] S. Das, M. Wohlert, J.J. Beaman, D.L. Bourell, Producing Metal Parts with Selective Laser Sintering/Hot Isostatic Pressing, JOM 50 (12), 1998, 17 – 20

[16] A. Dharia, H. Adeli, Neural Network model for rapid forecasting of freeway link travel time, Engineering Applications of Artificial Intelligence 16, 2003, 607 – 613