ORIGINAL RESEARCH

Multi-block deep regression with reinforced gradients: Modelling the estimation of direct weld parameters in automated welding

Soheil Keshmiri*

Hiroshi Ishiguro Laboratories (HIL), Advanced Telecommunication Research Institute International (ATR), Kyoto, Japan

ABSTRACT

This paper presents a deep regression model to estimation of the weld bead parameters in welding tasks. It is an aggregate of deep regression blocks where number of these blocks is proportional to the cardinality of the weld parameters. These blocks are trained simultaneously and share an identical structure with four-hidden-layer Sigmoid activation functions and a linear transformation at their outputs. Moreover, they incorporate a new meta-parameter, shared by all the hidden layers of a given block, to maintain the quality of the gradients of their respective weight matrices. This allows the model to further reduce the deviation of its estimates from the expected values of the weld parameters to significantly minimize its estimation error. The evaluation of the performance of this approach in contrast to state-of-the-art techniques in the literature shows a significant improvement in estimating these values for different welding processes. Furthermore, the proposed deep regression network is capable of retaining its performance when presented with combined data of different welding techniques. This is a nontrivial result in attaining an scalable model whose quality of estimation is independent of adopted welding techniques.

Key Words: Deep learning, Regression, Neural network, Gradient descent, Weld beads parameters estimation

1. INTRODUCTION

Welding is an indispensable part of virtually every industrial process, from construction of buildings and complex structures to automotive and shipyard production lines. This explains the existence of a wide range of welding techniques from Gas Metal Arc Welding (GMAW)[1] and Submerged Arc Welding (SAW)[2–4] to shielded metal-arc welding (SMAW),[5] Tungsten Insert Gas Welding (TIG),[6] and Gas Tungsten Arc Welding (GTAW).[7]

A welding task, in its common form, comprises of a number of input control signals. Some examples include the voltage, the current, the torch traveling speed, the wire feed rate, and the arc gap. Cook[8] uses the term Indirect Weld Parameters (IWP) to refer to these input control signals. Furthermore, Nunes[9] shows that IWP directly influence the geometrical description of the welding pool *i.e.*, its Direct Weld Parameters (DWP).[8] The depth of penetration, the width of weld beads, and the transverse cross-sectional areas are some of the examples of these parameters. A thorough investigation of the relationship between DWP and IWP is presented by Chandel and Bala.[10]

Research pertaining to welding processes is broadly categorized into two domain of studies, namely the estimation of the weld parameters and their sensitivity analysis. The latter

---
*Correspondence:** Soheil Keshmiri; Email: soheil@atr.jp; Address: Hiroshi Ishiguro Laboratories (HIL), Advanced Telecommunication Research Institute International (ATR), Kyoto, Japan.

refers to the study of the effect of the changes of IWP on the final weld quality. Whereas, the estimation problem attempts to predict the weld parameters, given the IWP. This article focuses on the estimation problem.

Rosenthal[11] presents an early static model based on the heat flow where the welding contour is expressed as a function of electrode velocity, heat input, and the material properties. In addition, many researchers apply sophisticated mathematical models to capture this relationship. Some examples include factorial design,[12–14] linear regression,[1, 15] multiple regression analysis,[16] response surface methodology,[17, 18] Taguchi method,[19, 20] and the combination of genetic algorithm and artificial neural network.[21]

In particular, application of the artificial neural network (ANN) in estimation of the weld bead parameters bring a remarkable progress in recent years. These models are successfully deployed in investigation of the modeling, the process control, and the estimation of the quality of the weld beads.[22] Anderson *et al.*[7] use ANN to model the arc welding process. Their model achieves satisfactory results on its prediction accuracy as compared to the conventional control systems. Tang *et al.*[23] apply back-propagation and counter-propagation to analyze the relationship between IWP and the geometry of the weld bead in TIG welding. Polte *et al.*[24] consider the application of the self-organizing maps (SOM) in monitoring and quality evaluation of the GMAW welding. Nagesh and Datta[5] apply back-propagation on modeling and estimation of the weld bead parameters in SMAW. Benyounisa and Olabi[25] present a comprehensive survey on optimization approaches to estimation of direct weld parameters.

One challenging aspect of this estimation problem is the dependency of DWP on the same input control signals or IWP. For instance, the same set of values of voltage, current, and torch traveling speed (*i.e.*, IWP) produces the different values that pertain to the depth of penetration and the width of weld beads (*i.e.*, DWP). Therefore, it is of no value to combine these parameters in a single output vector to estimate their values. This is mainly due to the pseudo-redundancy caused by the overlapping values of input feature vectors for different output data. This is particularly problematic when applying backpropagation[26] due to the vanishing gradients exhibited by the activation functions in neural networks.[27] In addition, the weak correlation among DWP (please refer to Figure 3) does not allow the inference of the value of one (*e.g.*, depth of penetration) from the other (*e.g.*, width). Moreover, the IWP are inherently low-dimensional feature vectors with voltage, current, and torch traveling speed among the most common values. This limits the ability of estimation

models to establish a well-generalized mapping of different DWP onto the same set of IWP values. Although some models attempt to compensate for such shortcomings through introduction of polynomial features, such polynomial degrees do not necessarily guarantee a better estimation of DWP values.[14, 28]

Another challenging aspect of this problem is due to the wide range of application-specific welding techniques (*e.g.*, GMAW, SAW, TIG, *etc.* Please visit `https://en.wikiped ia.org/wiki/List_of_welding_processes` for comprehensive list of welding techniques). More specifically, such a diversity results in development of estimation techniques that solely model a particular welding domain. It is apparent that formulation of an scalable model to address the estimation of the welding parameters and regardless of the adopted welding techniques is highly desirable.

This article addresses these issues through introduction of a deep regression model. Its contributions are:

(1) It resolves the issue of the vanishing gradients in backpropagaton through introduction of a new meta-parameter, referred to as *reinforced gradient coefficient*, to enable this model to maintain the quality of its computed gradients. This approach is comparable to Gulcehre *et al.*[27] Whereas, they apply an additive noise to the input or output layers, the multiplicative nature of reinforced gradient coefficient which is directly applied to the calculated gradients magnifies the exponent of the activation function in next iteration, thereby sharpening its transition. In addition, this article presents a deep neural regression model as opposed to a deep neural classifier in their study.

(2) It shows a high scalability on combination of the datasets of different welding techniques. In particular, the change in its estimation error due to combination of data of different welding techniques is statistically insignificant. This is a non-trivial result that challenges the general belief in this field of research. However, this result is, by no means, a definitive solution but an early attempt to formulation of one such scalable model.

It is worth noting that the effect of low-dimensionality of IWP is tamed via readjusting the number of neurons of the hidden layers of the respective blocks of this model at runtime and in consecutive training epochs. This enables the model to better capture the interrelation between individual elements of the input feature vectors (*i.e.*, IWP), thereby eliminating the needs for further polynomial features.[14, 18] Additionally, the comparative study of present model in contrast to state-of-the-art techniques in the literature shows a

significant improvement in estimation of direct weld parameters.

The remainder of this article is organized as follows. Section 3 explains the overall architecture of the deep regression model along with its formulation. Results of performance of deep regression network are presented in Section 3. Section 4 elaborates on scalability of this model on combined data of different welding techniques. Conclusion and some insights on future direction of this research are presented in Section 5.

2. NEURAL ARCHITECTURE

The neural architecture of proposed model is an aggregate of several independent, four-hidden-layer regression blocks as depicted in Figure 1. Number of these blocks is proportional to cardinality of DWP. For instance, it is a two-block regression model if DWP are depth of penetration and width of weld beads. These blocks are trained simultaneously and share an identical structure in which the neurons of every preceding layer are fully connected to those of the succeeding layer as shown in Figure 2.
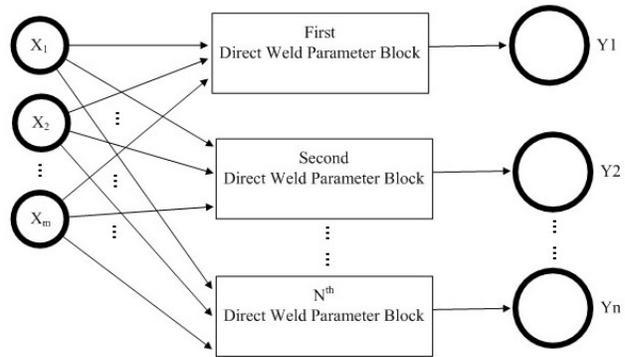
Figure 1. Overall structure of proposed deep regression model. Each rectangular block represents an independent four-hidden-layer structure whose set of meta-parameters is independent of the other blocks. Furthermore, each block learns to estimate the value of a unique direct weld parameter such as depth of penetration or width of weld beads. However, they all utilize the same set of IWP (*i.e.*, input control signals) during their training.
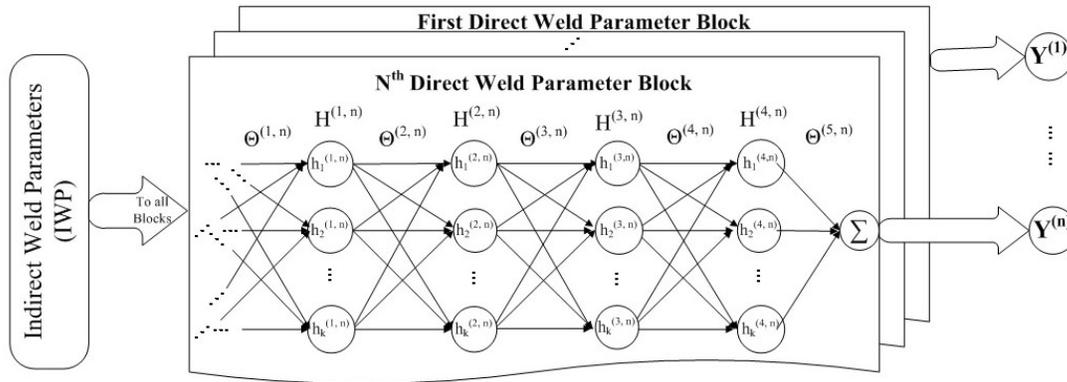
Figure 2. Block-level deep regression structure. Each block has its own four-hidden-layer regression (represented by $H^{(l,n)} = \{h_1^{(l,n)}, \dots, h_k^{(l,n)}\}$, $l = 1, \dots 4$) to estimate its designated direct weld parameter $Y^{(i)}$. In this figure, n is the total number of blocks (i.e., cardinality of DWP) and k indicates the number of neurons in a given layer. The weight matrices $\Theta^{(i,n)}$, $i = 1, \dots, 5$ are unique to their respective blocks. Each block utilizes its own independent set of meta-parameters. IWP form the input features to the entire model.

The first three hidden layers of these blocks utilize Sigmoid activation function to produce their respective intermediate outputs. As a result, the output of these layers are bounded within $[0, 1]$ interval. On the other hand, their last hidden layers employ linear transformation to generate their corresponding final estimates, thereby enabling each block to act as an independent deep non-linear regression model. Initially, each layer is assigned with two neural computation units. During the training, the model modifies the size of these hidden layers if such an update results in further improvement of the estimated values of the direct weld parameter of a given block.

Each of these deep regression block is trained using feedforward with back-propagation. In addition, they incorporate a new meta-parameter, referred to as *reinforced gradient coefficient*, in calculation of the gradients of their respective weight matrices. In essence, a *reinforced gradient coefficient* is a multiplicative factor that fine-tunes the value of the

learning rate during the training to maintain stronger gradients. This enables the model to further reduce the deviation of its estimated DWP from their expected values, thereby significantly minimizing its estimation error.

Let X denote the set of feature vectors (*i.e.*, IWP) to this model. Furthermore, let $Y^{(k)}$ be the set of direct weld parameter in training data that corresponds to the kth regression block. In addition, let $H^{(l,k)}$, $l = 1, \ldots, 4$, represent the lth hidden layer of the kth block of this architecture. Moreover, let $\Theta^{(i,k)}$, $i = 1, \ldots, 5$, be the weight matrix of the ith layer of the kth block that connect every neural unit of the preceding layer to those of its successor. This includes the connection between the input layer X to $H^{(1,k)}$ and the hidden layer $H^{(4,k)}$ to the output layer $Y^{(k)}$ at the kth block. There are three steps involved in formulation of the deep regression architecture, namely minimization of the cost function, feed-forward computation, and weight updates through back-propagation. These steps are explained in the following subsections.

2.1 Cost function

During the training process, we are interested in weight matrices that minimize the discrepancy between the estimated values of DWP in contrast to their actual values in training data. Moreover, it is important to prevent the training process from overfitting on training data. This increases the accuracy of the model in predicting the new DWP that do not come from the training set, thereby enabling it to scale well in novel scenarios. Furthermore, it is crucial to take into account the real-valued output of the last hidden layers of the independent regression blocks of this architecture. Therefore, the cost function of each block is formulated as a regularized sum of the squared error of its output to its corresponding direct weld parameter in training data:

$$J^{(k)} = \frac{1}{2m} \times [(Y^{(k)} - Y'^{(k)})^2 +$$
$$\lambda^{(k)} \sum_{l=1}^{L-1} \sum_{i=1}^{p^{(l,k)}} \sum_{j=1}^{q^{(l,k)}} (\Theta_{ji}^{(l,k)})^2] \tag{1}$$

where $Y'^{(k)}$ is the estimated values of the direct weld parameter that are calculated by the kth block. m is the total number of training data and k reflects the cardinality of the DWP. The second term in equation (1) is the regularization term that incorporates the sum of the squared of the weight matrices of all the layers in the kth block. This helps prevent the deep regression model of a given block from overfitting on its corresponding direct weld parameter in training data. $\lambda^{(k)}$ is the regularization factor that corresponds to the independent set of meta-parameters of the kth block. L, $p^{(l,k)}$,

and $q^{(l,k)}$ refer to the total number of hidden layers along with the row and the column dimensions of the weight matrix of the kth block at its lth layer, respectively.

2.2 Feed-forward

Values that are generated at the hidden layers are:

$$z^{(l,k)} = (H^{(l-1,k)})^T \times \Theta^{(l,k)} \tag{2}$$

$$H^{(l,k)} = \frac{1}{1 + e^{-z^{(l,k)}}}, \; l = 1, \ldots, 4 \tag{3}$$

with $H^{(0,k)} = X$ for all values of k in equation (2). These values correspond to the application of the Sigmoid transformation on the input as we move forward through layers of the networks of regression blocks and towards their respective output layers. It is worth noting that an extra column with all whose entries equals to 1 are added to the input and the hidden layers to count for the bias term. This is why the bias term is not explicitly included in equation (2). This also explains the starting indices of the last two summation operations from 1 in equation (1). The final outputs of these blocks are the linear transformation of their calculated values at the last hidden layer of their corresponding network *i.e.*, $H^{(4,k)}$:

$$Y'^{(k)} = (H^{(4,k)})^T \times \Theta^{(5,k)} \tag{4}$$

where $\Theta^{(5,k)}$ is the weight matrix that transforms the calculated values of the hidden layers of the kth block onto its final estimates.

2.3 Back-propagation and weights update

Equation (4) shows the final estimates of the values of the DWP by deep regression architecture. The discrepancies between these generated outputs and the actual DWP in training data are:

$$\delta^{(Y^{(k)})} = Y^{(k)} - Y'^{(k)}, k = 1, \ldots n \tag{5}$$

These values are used to update $\Theta^{(5,k)}$:

$$\Delta^{(5,k)} = \left[(\delta^{(Y^{(k)})})^T \times H^{(4,k)} \right] \times \gamma^{(k)} \tag{6}$$

$$\Theta^{(5,k)} = \Theta^{(5,k)} + \frac{1}{m}(\alpha^{(k)} \times \Delta^{(5,k)} + \lambda^{(k)} \sum_{i=1}^{p^{(5,k)}} \sum_{j=1}^{q^{(5,k)}} \Theta_{ji}^{(5,k)}) \tag{7}$$

where m is the total number of training data and $k = 1, \ldots, n$ reflects the total number of blocks (*i.e.*, cardinality of DWP).

$p^{(5,k)}$ and $q^{(5,k)}$ refer to the dimensions of the weight matrix of the output layer of kth block. Equation (6) is the gradient of this output layer and $\gamma^{(k)}$ is the *reinforced gradient coefficient* of the kth block. In essence, $\gamma^{(k)}$ is a standard normal multiplicative factor (*i.e.*, $\mu + \sigma^2 \times \gamma^{(k)}$, $\mu = 0, \sigma^2 = 1$) that is learned during the training to maintain stronger gradients. This allows the deep regression model to further reduce its deviation from the expected DWP to significantly minimize its estimation error. Furthermore, $\alpha^{(k)}$ and $\lambda^{(k)}$ are the learning rate and the regularization factor of the corresponding independent set of meta-parameters of kth block, respectively. Similarly, the updates of the weight matrices $\Theta^{(4,k)}$, $\Theta^{(3,k)}$, and $\Theta^{(2,k)}$ are:

$$\delta^{(i,k)} = (\Theta^{(i,k)})^T \times \delta^{(i+1,k)} \times H^{(i,k)} \times \left[1 - H^{(i,k)}\right],$$
$$k = 1, \ldots, n$$

(8)

$$\Delta^{(i-1,k)} = ((\delta^{(i,k)})^T \times H^{(i-1,k)}) \times \gamma^{(k)} \tag{9}$$

$$\Theta^{(i-1,k)} = \Theta^{(i-1,k)} + \frac{1}{m}(\alpha^{(k)} \times \Delta^{(i-1,k)}$$
$$+ \lambda^{(k)} \sum_{i=1}^{p^{(i-1,k)}} \sum_{j=1}^{q^{(i-1,k)}} \Theta_{ji}^{(i-1,k)}) \tag{10}$$

where $(H^{(i,k)} \times (1 - H^{(i,k)}))$ in equation (8) is the gradient of the Sigmoid activation function of the kth block at its ith hidden layer. The weight update for $\Theta^{(1,k)}$ closely follows equations (8-10) except that $H^{(i-1,k)}$ is replaced by X in equation (9):

$$\delta^{(2,k)} = (\Theta^{(2,k)})^T \times \delta^{(3,k)} \times \left[H^{(2,k)} \times (1 - H^{(2,k)})\right] \tag{11}$$

$$\Delta^{(1,k)} = \left[(\delta^{(2,k)})^T \times X\right] \times \gamma^{(k)} \tag{12}$$

$$\Theta^{(1,k)} = \Theta^{(1,k)} + \frac{1}{m}(\alpha^{(k)} \times \Delta^{(1,k)} + \lambda^{(k)} \sum_{i=1}^{p^{(1,k)}} \sum_{j=1}^{q^{(1,k)}} \Theta_{ji}^{(1,k)})$$

(13)

Training process starts with feed-forward calculation of the estimated outputs of the blocks. This is followed by the updates of their respective weight matrices through back-propagation. Next, it calculates the cost functions of these blocks using their latest outputs and their corresponding up-dated weight matrices. This procedure continues until a given number of iterations for a specific training epoch is met.

3. CASE STUDY

The performance of the deep regression model is compared in contrast to a number of selected studies from the literature namely, Anderson *et al.*,[7] Chandrasekhar and Vasudevan,[21] and Karaoglu and Secgin.[4] These articles form a reliable representatives of the trends of research in this field over the past two and a half decades. More specifically, Anderson *et al.*[7] is the seminal paper that introduces the application of multi-layer neural networks in estimation of the direct weld parameters. Furthermore, Karaoglu and Secgin[4] apply a curvilinear approach to estimation of DWP. Moreover, Chandrasekhar and Vasudevan[21] propose a combination of single hidden layer neural network and genetic algorithm for this purpose.
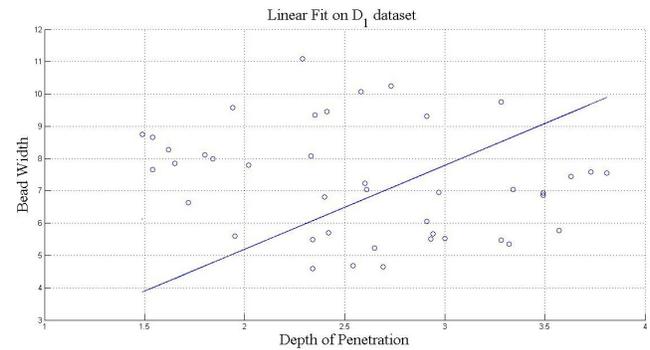
Figure 3. Regression fit for depth of penetration and width of weld beads in Anderson *et al.*[2] dataset of size 42. The Pearson, Spearman, and Kendall correlation ratios for this dataset are -0.24, -0.31 -0.17, respectively, indicating a weak inverse proportionality of the values of depth of penetration and width of weld beads. Furthermore, their *P*-values are 0.12, 0.05 and 0.09, rejecting the null hypothesis of presence of correlation between these parameters (with a very weak exception for Spearman) between these DWP.

Additionally, the welding data of these studies are publicly available through their respective articles. They pertain to three main welding techniques, namely arc welding,[7] submerged arc welding,[4] and A-TIG weldings on 304LN and 316LN stainless steel.[21] Throughout this Section, the following abbreviations are used to refer to each of these datasets:

(1) D_1: Anderson *et al.*[7] dataset of size 42 on arc welding process.
(2) D_2: Karaoglu and Secgin[4] dataset of size 28 on submerge arc welding process.

(3) D_{31} and D_{32}: Chandrasekhar and Vasudevan[21] datasets of sizes 120 each for A-TIG welding on 304LN and 316LN stainless steel, respectively.

Figure 3 shows the regression fit to the depth of penetration and width of weld beads in D_1. Their Pearson correlation ratio along with its *P*-value is $\{-0.24, 0.12\}$. Furthermore, those of Spearman and Kendall for these DWP are $\{-0.31, 0.05\}$ and $\{-0.17, 0.09\}$, respectively. These correlation ratios show a weak inverse proportionality between the values of these DWP. Furthermore, their *P*-values reject the null hypothesis of presence of any correlation between these DWP, with a very weak exception for Spearman *P*-value. This is in accordance with the claim in Section 1 of this article. Moreover, this trend of absence or weak correlation between DWP is exhibited by data in D_2, D_{31}, and D_{32}.

3.1 Training setup

Eighty percent of data of each of these datasets are reserved for training purpose. The remaining 20% is used for testing. The values that fall in train and test data are selected at random. Moreover, the current, the voltage, and the torch travelling speed are used as input features (*i.e.*, IWP) for estimating the depth of penetration and the width of weld beads as direct weld parameters (*i.e.*, DWP). These choices of input and output parameters make the comparative analysis of the performance of the proposed deep regression model in contrast to the above literature plausible.

The training process comprises of a number of training phases with varying iterations. More specifically, 1,000 through 12,000 iterations are adopted where each setting of the number of iterations corresponds to a different training phase. This helps optimize the meta-parameters of each of the deep regression blocks, thereby achieving better estimates of DWP. These meta-parameters include the learning rate α, the reinforced gradient coefficient γ, the regularization factor λ, and the number of neural units at the hidden layers of these blocks.

It is also worth noting that lack of cross validation set due to insufficient data is an important issue. However, this shortcoming is compensated by adapting the following strategy: Given a dataset, the training starts on a randomly selected 80% of available data, optimizing the meta-parameters that correspond to respective DWP blocks of the network. Next, these learned parameters are used in training and estimation in 100 independent episodes with randomly initialized weight matrices of layers at the commencement of each episode. Finally, the averages of these RMSEs and PEs are calculated as the final performance result of the deep regression network. Although this is not a highly desirable substitute for valida-

tion set, the convergence of these averaged RMSEs/PEs over the randomly initialized weight matrices in these episodes is an indicative of stability of this architecture.

Table 1. Best set of meta-parameters for depth of penetration regression block

Dataset	No. Neurons	Deg	α	γ	λ	Iters
D_1	4	0	40.0	0.9	0	1,000
D_2	4	0	0.009	1,000	0.001	7,000
D_{31}	2	0	0.003	3,500	0	7,000
D_{32}	8	0	0.009	3,500	0.001	9,000

Table 2. Best set of meta-parameters for width of weld beads regression block

Dataset	No. Neurons	Deg	α	γ	λ	Iters
D_1	2	0	0.01	2,500	0	9,000
D_2	3	0	0.009	1,000	0.001	7,000
D_{31}	3	0	0.009	1,000	0	7,000
D_{32}	2	0	0.009	3,500	0.001	7,000

Table 1 and Table 2 show the final values of the best meta-parameters for depth of penetration and width of weld beads blocks of deep regression network. These tables reveal the overall simplicity of the architecture of the proposed deep regression model. More specifically, the cumulative number of neural units of the two blocks of this model (selected from the range $\{2, \ldots, 100\}$) does not exceed 5 neurons on average *i.e.*, 20 in total (with mean and standard deviation of (4.5, 2.516) and (2.5, 0.577) for depth of penetration and width of weld beads). Furthermore, the most populated block corresponds to the depth of penetration on D_{32} with 32 neurons in each case. It is interesting to note that the best results reported by Anderson *et al.*[7] on multi-layer neural networks correspond to a 2×18 networks (*i.e.*, 36 neurons in total). Moreover, this model does not require any polynomial features (examined in the range $\{0, \ldots, 6\}$), as indicated by the zeros in *"Deg"* entries of these tables. This significantly reduces the amount of time spent on training the model. Additionally, the regularization parameters of these blocks are considerably small, with most of whose entries zeros. This indicates that the proposed deep regression network is not susceptible to bias on training data. In other words, it does not suffer from overfitting in general.

Moreover, these tables reveal the inverse proportionality of the values of the learning rate α and the reinforced gradient coefficient γ. This implies that in all these cases the readjustment of the gradients of the weight matrices results in better compensation for the variance that is exhibited by the input feature matrix (*i.e.*, IWP), thereby inducing a smaller

value of α while updating the weight matrices in equation (7), equation (10), and equation (13). However, the variation in number of iterations during the training (with mean and standard deviation of (6,000, 3,464.1) and (7,500, 1,000) for depth of penetration and width of weld beads) does not exhibit any particular pattern or correspondence with the size of given datasets and/or the DWP. As a result, any conclusion on the effect of the size of the dataset or type of direct weld parameter on training process is not warranted. However, it takes longer, on average, to converge on width of weld beads as compared to depth of penetration.

Table 3. RMSE and PE results - deep regression model (DRM) *vs.* D_1

	Penetration		Width	
	RMSE	PE	RMSE	PE
DRM	0.081 mm	5.57%	0.093 mm	2.37%
D_1	—	19.58%	—	5.68%

3.2 Results

Tables 3-6 show the results of the performance of the deep regression model (DRM in these tables) on the test data in comparison with results reported in D_1, D_2, D_{31}, and D_{32}, respectively. Root Mean Squared Error (RMSE) and the percentage of Prediction Error (PE) are reported as measures of accuracy of estimated values. PE is calculated as:

$$PE = \frac{|Y^{(k)} - Y'^{(k)}|}{Y^{(k)}} \times 100 \qquad (14)$$

where $Y^{(k)}$ and $Y'^{(k)}$ are the actual and estimated direct weld parameter of the kth block and $|.|$ gives the absolute value of its argument. RMSE and PE are chosen to allow the comparative analysis of these results with respect to the selected literature. More specifically, Anderson *et al.*[7] report on PE. Whereas, Karaoglu and Secgin[4] and Chandrasekhar and Vasudevan[21] provide RMSE as a measure of accuracy of their models.

Table 4. RMSE and PE results - deep regression model (DRM) *vs.* D_2

	Penetration		Width	
	RMSE	PE	RMSE	PE
DRM	0.031 mm	1.01%	0.022 mm	0.62%
D_2	0.292 mm	—	0.353 mm	—

A comparison between the PE values in Table 3 reveals that the standard deviation of the differences between these values is above one standard deviation (9.91 and 2.34, respectively). As a result, the reduction of the percentage of the prediction error by the proposed model on this dataset is significant. Additionally, the RMSE values that are calculated by DRM

are considerably small (in the scale of millimeter). However, their significance cannot be confirmed since the RMSE values of the multi-layer network in[7] are not reported.

Table 5. RMSE and PE results - deep regression model (DRM) *vs.* D_{31}

	Penetration		Width	
	RMSE	PE	RMSE	PE
DRM	0.113 mm	4.26%	0.054 mm	1.84%
D_{31}	0.148 mm	—	0.205 mm	—

Table 6. RMSE and PE results - deep regression model (DRM) *vs.* D_{32}

	Penetration		Width	
	RMSE	PE	RMSE	PE
DRM	0.043 mm	2.03%	0.056 mm	1.96%
D_{32}	0.124 mm	—	0.156 mm	—

We notice the same trend of improvement on the entries of Tables 4-6. More specifically, Table 4 shows 9.42 and 16.05 times improvements on the RMSE values of the proposed deep regression model on these DWP, as compared to D_2. Moreover, the PE values reported in this table are significantly small. Similarly, Table 5 and Table 6 report the improvements of the RMSE values that range between 1.31 to 3.80 times, compared to results obtained in D_{31} and D_{32}, respectively. Although the reported improvements in Table 5 are considerably smaller, they present the direct results of the performance of the proposed deep regression model and without application of any post optimization processing (*e.g.*, genetic algorithm, as reported by Chandrasekhar and Vasudevan[21]).

4. Scalability on combined data of different welding techniques

This section elaborates on performance of the proposed deeep regression network on combination of D_1 through D_{32} (referred to as D_{all}) to show its scalability on different welding techniques. Same ratio of 80% and 20% are used for train and test data. Moreover, same precedure for training and averaging described in Section 3.1 are followed on D_{all}. Test data is selected at random once all datasets are combined. Furthermore, the result of performance of the proposed deep regression model (DRM) is reported in contrast to normal equation regression (NER),[15] multiple curvilinear regression (MCR),[4,29] and support vector regression (SVR).[30,31] The NER, using the pseudoinverse matrix of input feature vectors is:[32]

$$\Theta = (X^T X)^{-1} X^T Y \qquad (15)$$

where X, Y, and Θ are the IWP, DWP, and the weight matrix corresponding to all direct weld parameters, respectively. It is apparent that Θ is of $m \times \|DWP\|$ dimension with m and $\|DWP\|$ being the size of train data and the cardinality of direct weld parameters. In addition, Python scikit-learn package is used for modeling MCR and SVR.[33]

Table 7. Best set of meta-parameters for depth of penetration for normal equation regression (NER), multiple curvilinear regression (MCR), and support vector regression (SVR) on D_{all}

	α	γ	Deg	Kernel	C	Γ	Iters
NER	-	-	3	-	-	-	-
MCR	0.3	0.09	3	-	-	-	7,000
SVR	-	-	3	poly	0.009	0.001	5,000

Table 8. Best set of meta-parameters for width of weld beads for normal equation regression (NER), multiple curvilinear regression (MCR), and support vector regression (SVR) on D_{all}

	α	γ	Deg	Kernel	C	Γ	Iters
NER	-	-	2	-	-	-	-
MCR	0.3	0.09	2	-	-	-	4,000
SVR	-	-	0	rbf	300	0.03	1,000

Table 7 and Table 8 show the best set of meta-parameters for NER, MCR, and SVR to calculate depth of penetration and width of weld beads, respectively. Hyphenated entries of these tables imply that the corresponding meta-parameters are not used by the given technique. In these tables, *"Deg"* refers to the number of additional polynomial degrees to yield the best RMSEs of the DWP by a given model. It is selected from the range $\{0, 1, \dots, 9\}$. Similarly, the *kernel*, *C*, and Γ entries refer to the kernel function, the penalty parameter of the error term (equivalent to $\frac{1}{\alpha}$ in logistic regression), and the kernel coefficient that are associated with the SVR. The best kernel function is selected from possible choices of kernel for SVR in.[33] They are *rbf, ploy, sigmoid* and *linear* kernels where *rbf* and *poly* stand for radial basis and polynomial functions, respectively. The kernel coefficient Γ is associated with the first three of these functions.

Table 9. Best sets of meta-parameters for depth of penetration and width of weld beads blocks - deep regression model on D_{all}

Dataset	No. Neurons	Deg	α	γ	λ	Iters
Penetration	9	0	20	1,000	0.003	6,000
Width	7	0	7	1,500	0.003	4,000

The best sets of meta-parameters for DRM on D_{all} is shown in Table 9. This table indicates that the polynomial degrees

of the model remains zero, as its comparison to Table 1 and Table 2 suggests. However, there is an increase in the number of neural units in the hidden layers of its DWP blocks, compared to their numbers, on average, in Table 1 and Table 2 (*i.e.*, 4.5 and 2.5, respectively). Moreover, the inverse proportionality of the values of learning rate α and reinforced gradient coefficient γ is, comparably, more subtle in Table 9. These observations imply a higher variation in input feature vectors (*i.e.*, IWP) to the model. Such changes on meta-parameters are expected as D_{all} pertains to the combination of IWP of different welding techniques. However, the regularization λ retains its low value, indicating that DRM is less susceptible to overfitting and bias on training data on D_{all} as well. Furthermore, this observation holds true as we compare *DRM* with *MCR* and *SVR*. This is evident in its smaller λ as compared to C and λ in Table 7 and Table 8.

Table 10. RMSE and PE results - deep regression model (DRM) *vs.* normal equation regression (NER), multiple curvilinear regression (MCR), and support vector regression (SVR) on D_{all}

	Penetration		Width	
	RMSE	PE	RMSE	PE
DRM	0.183 mm	3.86%	0.187 mm	2.58%
NER	0.343 mm	6.88%	0.895 mm	11.20%
MCR	0.322 mm	6.09%	0.873 mm	11.42%
SVR	1.586 mm	31.81%	0.400 mm	8.02%

Table 11. 95% Confidence interval and the Coefficient of variation of RMSEs of width of weld beads - deep regression model (DRM) *vs.* normal equation regression (NER), multiple curvilinear regression (MCR), and support vector regression (SVR) on D_{all}.

	Confidence interval	Coefficient of variation (in %)
DRM	(0.121, 0.245)	4.87
NER	(0.281, 0.405)	9.13
MCR	(0.260, 0.384)	8.57
SVR	(1.524, 1.648)	42.21

Such adjustments on meta-parameters that are associated with the updates of weight matrices have positive influence on estimation of DWP as shown in Table 10. The RMSE entries of this table indicate that DRM achieves 1.75 - 8.67 and *2.13 - 4.78* times improvements on estimation of the depth of penetration and width of weld beads, as compared to other models. Furthermore, an analysis of the PE values in Table 10 indicates that these values are not within one standard deviation from each other (*11.40 and 3.57 for depth of penetration and width of weld beads, respectively). Moreover, the non-overlapping confidence interval of DRM with those of *NER*, *MCR*, and *SVR* in Table 11 and Table 12

support the above observations. These analyses imply that the improvements exhibited by the proposed deep regression network is statistically significant. Table 11 and Table 12 provide the coefficient of variation of the RMSE values of these models for comparison.

Table 12. 95% Confidence interval and the Coefficient of variation of RMSEs of width of weld beads - deep regression model (DRM) *vs.* normal equation regression (NER), multiple curvilinear regression (MCR), and support vector regression (SVR) on D_{all}

	Confidence interval	Coefficient of variation (in %)
DRM	(0.125, 0.249)	4.98
NER	(0.833, 0.957)	23.82
MCR	(0.811, 0.935)	23.23
SVR	(0.338, 0.462)	10.65

Additionally, a comparison between the performance of DRM on D_{all} in contrast to Chandrasekhar and Vasudevan,[21] and Karaoglu and Secgin[4] in Tables 4-6 indicates that the difference between these RMSE values are insignificant. More specifically, these RMSE values are within one standard deviation of one another (0.064 and 0.076 for depth of penetration and width of weld beads, respectively). Furthermore, this observation is supported by the overlapping confidence interval of DRM and the intervals associated with these techniques in Table 13. In other words, the estimates of DWP based on combined data of different welding techniques is statistically as good as the results reported.[4,21] However, no conclusion on the difference of performance of DRM on D_{all} and D_1 can be drawn since Anderson *et al.*[7] do not provide the RMSEs of their model on D_1.

Table 13. 95% Confidence interval of the RMSEs of DWP - deep regression model (DRM) on D_{all} *vs.* Chandrasekhar and Vasudevan[21] and Karaoglu and Secgin[4] on D_2, D_{31}, and D_{32}

	Depth of penetration	Width of weld beads
D_{all}	(0.121, 0.245)	(0.125, 0.249)
D_2	(0.230, 0.354)	(0.246, 0.415)
D_{31}	(0.086, 0.211)	(0.143, 0.267)
D_{32}	(0.062, 0.186)	(0.094, 0.218)

Table 14. 95% Confidence interval of the RMSEs of DWP - deep regression model (DRM) performance on D_{all}, D_1, D_2, D_{31}, and D_{32}

	Depth of penetration	Width of weld beads
D_{all}	(0.121, 0.245)	(0.125, 0.249)
D_1	(0.019, 0.143)	(0.031, 0.155)
D_2	(-0.031, 0.102)	(-0.040, 0.084)
D_{31}	(0.051, 0.175)	(-0.008, 0.120)
D_{32}	(-0.019, 0.110)	(-0.006, 0.118)

However, the quality of the calculated RMSE values of DRM degrades when its performance on D_{all} is compared in contrast to its results on D_1 through D_{32}. More specifically, the PE values for its depth of penetration regression block are above one standard deviation from each other (it is 1.62). Although, its corresponding PE values for the width of weld beads regression block is within one standard deviation (*i.e.*, 0.65), the non-overlapping confidence interval of D_{all} with most intervals associated with D_1 through D_{32} in Table 14 indicates that the change in performance of DRM is significant (exceptions include D_1 and depth of penetration for D_{31}).

5. CONCLUSION

This article presents a learning architecture based on an aggregate of number of independent deep regression blocks where number of these blocks is proportional to the cardinality of the direct weld parameters (DWP). The proposed deep regression network shows a significant improvement in estimating these parameters as compared to state-of-the-art techniques in the literature. Furthermore, it demonstrates its ability to retain its performance when presented with combined data of different welding techniques as compared to results obtained by the selected articles. This is a nontrivial result in attaining an scalable model whose quality of estimation is independent of adopted welding techniques.

Although the adapted sets of meta-parameters of the DWP blocks of this model indicates its ability to realize the high variation of the input feature vectors (IWP), a comparison between its performance on combined data of different welding techniques versus its results on separate datasets reveals a significant decay in its performance. The future of this research pertains to identification of the cause(s) of such a degradation in its performance along with evaluation of the potential solutions

In addition, the domain of welding techniques that are investigated in this study will be broaden to include more welding data to evaluate the effect of the size of the training data on the quality as well as scalability of this model.

Moreover, most welding tasks are multi-layer welding processes whose structures consist of several layers of welds as opposed to a single path welding. This results in every preceding layer of welded material to affect the geometry of the beads of the succeeding layer due to a number of factors such as heat, their surface layout, and so on. Therefore, it is crucial to analyze the performance of this model on data that corresponds to such multi-layer welding operations to realize the potential that the multi-block deep regression model can offer to the solution concept of such multi-layer welding operations.

REFERENCES

[1] Son KJ, Yang YS, Kim IS, et al. Sensitivity analysis for process parameters in GMA welding processes using a factorial design method. Int. J. Mach. Tools Manuf. 2003; 43: 763-9. `http://dx.doi.org/10.1016/S0890-6955(03)00054-3`

[2] Dhas JER, Satheesh M. Sensitivity analysis of submerged arc welding parameters for low ally steel weldment. Indian J. of Eng. & Materials Sci. 2013; 20: 425-34.

[3] Yang WH, Tarng YS, Juang SC. The use of fuzzy logic in the Taguchi method for the optimization of the submerged arc welding process. Int. J. Adv. Manuf. Tech. 2000; 16: 688-94. `http://dx.doi.org/10.1007/s001700070040`

[4] Karaoglu S, Secgin A. Sensitivity analysis of submerged arc welding process parameters. J. Mater. Processing Technology. 2008; 202: 500-7. `http://dx.doi.org/10.1016/j.jmatprotec.2007.10.035`

[5] Nagesh DS, Datta GL. Prediction of weld bead geometry and penetration in shielded metal-arc welding using artificial neural networks. J. Mater. Process. Technol. 2002; 123: 303-12. `http://dx.doi.org/10.1016/S0924-0136(02)00101-2`

[6] Dong ZB, Wei YH, Xu YL, et al. Marangoni convection and weld shape variation in ATIG welding process. Theoretical and applied fracture mechanics. 2007; 48(2): 178-86. `http://dx.doi.org/10.1016/j.tafmec.2007.05.004`

[7] Anderson K, Cook GE, Karsai G, et al. Artificial neural networks applied to arc welding process modeling and control. IEEE Trans. on Industry Applications. 1990; 26(5): 824-30. `http://dx.doi.org/10.1109/28.60056`

[8] Cook GE. Feedback and adaptive control in automated arc welding systems. Metal Construction. 2002; 13(9): 551-6.

[9] Nunes AC. An extended Rosenthal weld model. Welding. 1983: 165-70.

[10] Chandel RS, Bala S. Relationship between submerged arc welding parameters and weld bead size. Schweissen Schneiden. 1998; 40(2): 28-31 (88-91).

[11] Rosenthal D. Mathematical theory of heat distribution during welding and cutting. Welding J. 1941; 20(5): 220-34.

[12] Raveendra J, Parmar RS. Mathematical models to predict weld bead geometry for ux cored arc welding. J. Metal Construct. 1987; 19(2): 31R-5R.

[13] Gupta VK, Parmar RS. Fractional factorial technique to predict dimensions of the weld bead in automatic submerged arc welding. IE (I) J-MC. 1989 November; 70: 67-75.

[14] Murugan N, Parmar RS. Effects of MIG process parameters on the geometry of the bead in the automatic surfacing of stainless steel. Material Proccessing Technology. 1994; 41: 381-98. `http://dx.doi.org/10.1016/0924-0136(94)90003-5`

[15] Bibby MJ, Yang LJ, Chandel RS. Linear regression equations for modeling the submerged-arc welding process. J. Mater. Process Technology. 1993; 39(1-2): 33-42. `http://dx.doi.org/10.1016/0924-0136(93)90006-R`

[16] Muralidharan C, Rajakumar S, Balasubramanian V. Statistical analysis to predict grain size and hardness of the weld nugget of friction-stir-welded AA6061-T6 aluminum alloy joints. Int. J. of Advanced Manufacturing Technology. 2011; 57(1–4): 151-65.

[17] Murugan N, Parmar RS. Effects of welding conditions on microstructure and properties of tupe 316L stainless steel submerged arc welding cladding. Weld J. 1997; 76(5): 210-20.

[18] Gunaraj N, Murugan V. Prediction and control of weld bead geometry and shape relationships in submerged arc welding of pipes. J. Mater Process Technology. 2005; 168: 478-87. `http://dx.doi.org/10.1016/j.jmatprotec.2005.03.001`

[19] Juang SC, Tarng YS. Process parameters selection for optimizing the weld pool geometry in the tungsten inert gas welding of stainless steel. J. Mater. Process Technol. 2002; 122: 33-7. `http://dx.doi.org/10.1016/S0924-0136(02)00021-3`

[20] Han HS, Son KJ, Lee HK, et al. Optimization of Nd-YAG laser welding parameters for sealing small titanium tube ends. J. Mater. Sci. Eng. 2006; A415: 149-55.

[21] Chandrasekhar N, Vasudevan M. Intelligent modeling for optimization of a A-TIG welding process. Mater. and Manufacturing Processes. 2010; 25: 1341-50. `http://dx.doi.org/10.1080/10426914.2010.529584`

[22] Barnett RJ, Anderson K, Cook GE, et al. Weld modeling and control using artificial neural networks. IEEE Trans. Indus. Appl. 1995; 31(6): 1484-91. `http://dx.doi.org/10.1109/28.475745`

[23] Tarng YS, Juang SC, Lii HR. A comparison between the back propagation and counter-propagation networks in the modeling of the TIG welding process. J. Mater. Process. Technol. 1998; 75: 54-62. `http://dx.doi.org/10.1016/S0924-0136(97)00292-6`

[24] Polte T, Wu CS, Rehfeldt D. Gas metal arc welding process monitoring and quality evaluation using neural network. Sci. Technol. Welding Joining. 2000; 5(5): 324-8. `http://dx.doi.org/10.1179/136217100101538380`

[25] Benyounisa KY, Olabi AG. Optimization of different welding processes using statistical and numerical approaches a reference guide. Advances in Engineering Software. 2008; 39(6): 483-96. `http://dx.doi.org/10.1016/j.advengsoft.2007.03.012`

[26] Bottou L, Orr G, Muller B, et al. Efficient backprop. Neural Networks: tricks of the trade; 1998.

[27] Moczulskiy M, Denily M, Gulcehrey C, et al. Noisy activation functions arxiv preprint. arXiv:1603.00391v2. 2016; 2: 1-4.

[28] Murugan N, Gunaraj V. Prediction and control of weld bead geometry and shape relationships in submerged arc welding of pipes. Journal of Materials Processing Technology 2005; 168: 478-87. `http://dx.doi.org/10.1016/j.jmatprotec.2005.03.001`

[29] Pedryczb W, Bargielaa A, Nakashima T. Multiple regression with fuzzy data. Fuzzy Sets and Systems. 2007; 158(19): 2169-88. `http://dx.doi.org/10.1016/j.fss.2007.04.011`

[30] Shinde VB, Gadakh VS, Khemnar NS. Optimization of welding process parameters using moora method. The International Journal of Advanced Manufacturing Technology. 2013; 69(9): 2031-9.

[31] Xingweia T, Shudea J, Zhitaob Y, et al. Prediction of mechanical properties of welded joints based on support vector regression. Procedia Engineering. 2012; 29: 471-1475.

[32] Cormen TH, Stein C, Rivest R, et al. Introduction to Algorithms. MIT Press; 1990.

[33] Available from: `http://scikit-learn.org/stable/`