# MACHINE LOAD ESTIMATION VIA STACKED AUTOENCODER REGRESSION

Angshul Majumdar<sup>†</sup>

Tulika Bose\*

Tanushyam Chattopadhyay\*

\* TCS Research and Innovation, Tata Consultancy Services, Kolkata, India
 <sup>†</sup> Indraprastha Institute of Information Technology, Delhi, India

### ABSTRACT

The problem of load estimation from sensor signals holds significance in the field of intelligent manufacturing. The goal of this work is to estimate the axial and spindle load values in a Computer Numerical Control machine from input sensor readings like spindle speed, feed rate, tool positions, etc. This can be viewed as a standard regression problem. Here, we propose a novel deep learning based regression technique that incorporates regression within the stacked autoencoder framework. Unlike the popular heuristic pretraining, fine-tuning approach, we solve all the parameters of the problem jointly. A variable splitting Augmented Lagrangian approach is employed to solve the ensuing optimization problem. Comparisons on standard regression models like linear, Least Absolute Shrinkage and Selection Operator (LASSO), Support Vector Regression and the traditional stacked autoencoder have shown that our technique considerably outperforms them.

*Index Terms*— Asymmetric stacking, deep learning, regression, estimation, intelligent manufacturing

### 1. INTRODUCTION

The intelligent manufacturing domain has been automated to a great extent with the introduction of the Computer Numerical Control (CNC) machines [1]. The entire machining process involves complex interaction of several operating signals obtained from different sensors [2]. Since the tool surfaces undergo continuous wearing with the progress of the operation and often result in tool breakage, tools are replaced manually in a timely manner based on simple rules like the number of workpieces completed and the like [3]. However, in order to make the tool replacement strategy cost-effective as well as to prevent damage due to tool breakage, this process needs to be automated.

The load or power incurred across the axes and the spindle during the actual cutting process, are effective in detecting breakage of the cutting tool [4]. Thus the focus of our paper is to mathematically model and estimate the axial load and the spindle load using sensor signals associated with the machine, including feed rate, surface speed, spindle speed, etc., for better productivity and process control [5,6].

Although ideally a physics based model can be built for the above problem, it is not always possible to develop them due to the complicated dynamics of manufacturing systems. In this work we will go for a completely data driven approach and formulate it as a regression problem. We propose a deep learning architecture based on the stacked autoencoder framework to solve this regression problem as it can learn arbitrary relationships without specification from the user. For such supervised deep learning problems, the standard approach is to follow the pre-training and fine-tuning paradigm [7]. In the pre-training stage, each layer of the stacked autoencoder is learnt greedily. In the fine-tuning stage, the targets are attached at the deepest layer of the encoder and the decoder is detached. The supervised architecture is fine tuned. However, in such a greedy paradigm, there is no relationship between the parameters learnt during the pre-training and the fine-tuning stages.

In this work, we propose to incorporate regression into the stacked autoencoder framework. But instead of learning it greedily, we jointly solve for all the weights in a mathematically more optimal fashion, without using backpropagation. This will be a novel formulation, which will be solved using variable splitting Augmented Lagrangian techniques [8, 9]. The remainder of the paper is structured in the following manner. Section 2 gives a brief review of the prior literature. The proposed regression model is explained in detail in Section 3. This is followed by experimental evaluation and results in Section 4. Finally Section 5 concludes our work.

# 2. LITERATURE REVIEW

### 2.1. Regression for machine automation

There exist several prior-arts for regression models, amongst which some of the popular ones are the linear regression, Generalised Linear Models (GLM) [10], quasi-likelihood estimation [11], Least Absolute Shrinkage and Selection Operator (LASSO) [12], non-parametric Nadaraya-Watson kernel estimator [13, 14] and Support Vector Regression (SVR) [15]. However, since only few of these regression models have been used for the problems of load estimation and machine automation and also due to the limitation of space, we shall refrain from discussing all of these models here. Linear regression models have been explored in the intelligent manufacturing domain for various problems like spindle load estimation [16], error compensation [17] and the like. Given the load  $y_i$  and the signal predictors  $x_i = \{x_{i1}, ..., x_{ip}\}_{i=1}^n$ , the model is given by  $y_i = x_i^T \beta + \epsilon_i$ . Here  $\beta$  is the parameter vector and  $\epsilon$  is the error term or noise.

Non-linear regression techniques, especially SVR with polynomial kernels and gaussian kernels, have been used widely in machine automation use-cases. SVRs with these kernels have been applied for tool condition assessment in CNC machine [18], assessment of machine degradation [19], etc.

Besides, feature extraction and training Artificial Neural Networks (ANN) for regression have been performed for load estimation [16], tool wear prediction [20] and the like.

#### 2.2. Autoencoders



Fig. 1. Structure of an Autoencoder

Autoencoders, as shown in Fig. 1, are self-supervised neural networks, i.e. the inputs and the outputs are the same. The input data (X formed by stacking the training samples as columns of a matrix) is projected onto the hidden representation:  $H = \varphi(W_E X)$  by an encoder  $W_E$ ; there is a non-linear activation function  $\varphi$  associated with it. The decoder  $W_D$  reverse maps the representation onto the output (=input) as  $X = W_D H$ . During training, the encoder and the decoder are learnt by minimizing the Euclidean cost function.

$$\underset{W_E,W_D}{\arg\min} \left\| X - W_D \varphi(W_E X) \right\|_F^2 \tag{1}$$

Stacked autoencoders are created by nesting autoencoders one inside the other. Mathematically this is expressed as

$$\underset{W_{E_1}, W_{E_2}...W_{E_L}, W_{D_1}, W_{D_2}...W_{D_L}}{\operatorname{arg\,min}} \|X - g\left(f(X)\right)\|_F^2 \quad (2)$$

where  $g(f(X)) = W_{D_1}\varphi(W_{D_2}...\varphi(W_{D_L}(f(X))))$  and  $f(X) = \varphi(W_{E_L}\varphi(W_{E_{L-1}}...\varphi(W_{E_1}X))).$ 

It is difficult to learn all the parameters in one go by backpropagation owing to the vanishing gradient problem. Therefore, the stacked autoencoder is solved greedily one layer at a time [7] starting from the outermost layer.

However, greedy learning is a sub-optimal approach since the outer layers influence the inner layers but not vice versa. In optimal learning, all the layers should be affecting one another (unlike the case when backpropagation is used).

#### 3. OUR APPROACH

We propose a regression model based on stacked autoencoders with an asymmetric structure, consisting of multiple encoding layers and a single decoding layer. Previous studies on stacked autoencoders had equal number of encoders and decoders owing to the greedy training paradigm. Since we propose to learn all the layers jointly, we have the flexibility to change the architecture. We do not see any reason to have an equal number of encoders and decoders, especially since the decoders do not play any direct role in the analysis. More decoders mean more parameters to learn, which leads to over-fitting with limited training data. In a recent study [21], it was shown that having multiple encoders but only one decoder keeps the robust abstraction capacity of deep learning without the pitfalls of over-fitting. This motivates us to build upon the asymmetric architecture. We have shown the architecture for the two layer model, but one can easily extrapolate it to more layers.



**Fig. 2**. Architecture of the proposed Asymmetric Stacked Autoencoder for Regression

The regression framework, as shown in Fig. 2, consists of an asymmetric stacked autoencoder having two encoding layers and a single decoder. Non-linearity is added at each layer using a non-linear activation function. The autoencoder yields representations of the raw input X at its encoding layers ( $Z_1$ and  $Z_2$  in Fig. 2) associated with weight matrices  $W_{E_1}$  and  $W_{E_2}$  respectively. The single decoding layer, associated with the weight matrix  $W_D$ , strives to yield the raw data from the representation at the deepest encoding layer. In addition to the encoding-decoding paradigm, the estimated or regressed target value y is obtained from the representation at deepest layer ( $Z_2$  in Fig. 2) of the stacked autoencoder, associated with weight vector w.

Unlike the traditional pre-training and fine tuning approach, the proposed method jointly learns regression weights along with the encoding-decoding weights. This ensures a more optimal solution owing to the fact that in the former, the shallower layers influence the deeper layers but the opposite influence does not take place, which results in sub-optimality. When they are jointly learnt, all the layers influence one another as should be the case. Moreover, since backpropagation can only be used for training unidirectional structures, i.e. one input and one output, it cannot be applied to the proposed structure that has two outputs. Thus the regression problem is addressed by solving the ensuing optimization problem. Here, a variable splitting Augmented Lagrangian paradigm is followed to solve the same [8,9].

### 3.1. Joint-learning for regression

In contrast to the traditional two phase training model, the ensuing joint optimization problem for regression that needs to be solved in the training phase can be formulated as below:

$$\underset{W_{E_1}, W_{E_2}, W_D, w}{\operatorname{arg\,min}} \left( \left\| \left( X - W_D \varphi \left( W_{E_2} \varphi \left( W_{E_1} X \right) \right) \right) \right\|_F^2 + \lambda \left\| y - w^T \varphi \left( W_{E_2} \varphi \left( W_{E_1} X \right) \right) \right\|_F^2 \right)$$

$$(3)$$

The symbols used here are described previously. Here  $\lambda$  controls the weight of regression and  $\varphi$  is the non-linear activation function. This is a non-convex joint optimization problem.

We introduce two proxy variables viz.  $Z_2 = \varphi(W_{E_2}\varphi(W_{E_1}X))$  and  $Z_1 = \varphi(W_{E_1}X)$  [9]. The corresponding Augmented Lagrangian formulation is:

$$\underset{W_{E_{1}}, W_{E_{2}}, W_{D}, w, Z_{1}, Z_{2}}{\arg\min} \left( \left\| X - W_{D} Z_{2} \right\|_{F}^{2} + \lambda \left\| y - w^{T} Z_{2} \right\|_{F}^{2} \right. \\ \left. + \mu_{2} \left\| Z_{2} - \varphi(W_{E_{2}} Z_{1}) \right\|_{F}^{2} \\ \left. + \mu_{1} \left\| Z_{1} - \varphi(W_{E_{1}} X) \right\|_{F}^{2} \right);$$

$$(4)$$

where  $Z_1$  and  $Z_2$  are the representations of the input X at the encoding layers. The parameter  $\lambda$  and the hyper-parameters  $\mu_1$  and  $\mu_2$  need to be tuned as per the application. Breaking equation (4) into smaller pieces using Alternating Direction Method of Multipliers (ADMM) [22], we end up with 6 subproblems as given below:

P1: 
$$W_D \leftarrow \underset{W_D}{\operatorname{arg\,min}} \|X - W_D Z_2\|_F^2$$

P2: 
$$w \leftarrow \operatorname*{arg\,min}_{w} \left\| y - w^T Z_2 \right\|_F^2$$

P3:

$$W_{E_1} \leftarrow \underset{W_{E_1}}{\operatorname{arg\,min}} \|Z_1 - \varphi (W_{E_1}X)\|_F^2$$
$$\Rightarrow W_{E_1} \leftarrow \underset{W_{E_1}}{\operatorname{arg\,min}} \|\varphi^{-1}Z_1 - W_{E_1}X\|_F^2$$

$$W_{E_{2}} \leftarrow \underset{W_{E_{2}}}{\operatorname{arg\,min}} \|Z_{2} - \varphi (W_{E_{2}}Z_{1})\|_{F}$$
$$\Rightarrow W_{E_{2}} \leftarrow \underset{W_{E_{2}}}{\operatorname{arg\,min}} \|\varphi^{-1}Z_{2} - W_{E_{2}}Z_{1}\|_{F}^{2}$$

P5: 
$$Z_{2} \leftarrow \underset{Z_{2}}{\arg\min} \left( \|X - W_{D}Z_{2}\|_{F}^{2} + \lambda \|y - w^{T}Z_{2}\|_{F}^{2} + u_{2} \|Z_{2} - \wp(W_{E}, Z_{1})\|_{T}^{2} \right)$$

$$\Rightarrow \underset{Z_2}{\operatorname{arg\,min}} \left\| \begin{pmatrix} X \\ \sqrt{\lambda}y \\ \sqrt{\mu_2}\varphi \left( W_{E_2}Z_1 \right) \end{pmatrix} - \begin{pmatrix} W_D \\ \sqrt{\lambda}w^T \\ \sqrt{\mu_2}I \end{pmatrix} Z_2 \right\|_F^2$$

P6:

 $\Rightarrow$ 

5: 
$$Z_{1} + \mu_{1} \left\| Z_{1} - \varphi(W_{E_{1}}X) \right\|_{F}^{2} \right)$$
$$\arg\min_{Z_{1}} \left\| \left( \begin{array}{c} \sqrt{\mu_{2}}\varphi^{-1}(Z_{2}) \\ \sqrt{\mu_{1}}\varphi(W_{E_{1}}X) \end{array} \right) - \left( \begin{array}{c} \sqrt{\mu_{2}}W_{E_{2}} \\ \sqrt{\mu_{1}}I \end{array} \right) Z_{1} \right\|_{F}^{2}$$

 $Z_1 \leftarrow \arg\min\left(\mu_2 \left\| Z_2 - \varphi(W_{E_2} Z_1) \right\|_F^2$ 

P1, P2 and P5 are standard linear least square problems having closed form solutions. They can also be solved using conjugate gradient. P3, P4 and P6 can be easily converted to their equivament forms since the activation functions are unitary and easily invertible. Hence, P3, P4 and P6 turn out to be linear least square problems. This concludes the training process.

Finally, during testing, the unknown output  $\tilde{y}$  for test data  $X_{test}$  can be estimated using the learned weights by solving

$$\tilde{y} = w^T \varphi \left( W_{E_2} \varphi \left( W_{E_1} X_{test} \right) \right) \tag{5}$$

### 4. EXPERIMENTAL EVALUATION

We have performed validation over smart factory real-world data involving 2 axis Horizontal and 4 axis Vertical CNC Turning machines. Since we have worked on proprietary data. we are unable to share the same publicly. The data consist of different sensor signals associated with the machine. The axial load and spindle load, incurred while machining specific components, have been measured in units of mega-watt (Mw) and recorded as ground truth. Data samples were collected per second throughout a day over a period of 1 month during the actual operating condition. The captured signals consist of absolute tool position across all the axes, amount of distance to go along the controlled axes, spindle speed value related to constant surface speed control on CNC, surface speed value, feed rate and the spindle motor speed. Our goal is to estimate load values at different axes as well as the spindle load using these captured signals.

 
 Table 1. Parameter values considered for the proposed regression model

Axes names	$\lambda$	$\mu_1$	$\mu_2$
Axis 1	1.6	1.3	1.3
Axis 2	0.5	1.7	1.7
Axis 3	0.6	1.6	1.6
Axis 4	1.1	1.5	1.5
Spindle	0.7	1.6	1.6

We have added non-linearity at each layer of the model using the hyperbolic tangent function. The number of nodes at hidden layers were adjusted experimentally. The proposed regression model was implemented in Matlab. The parameter values considered for our model, after performing grid search, are listed in Table 1. Table 2 presents the results obtained across 3 exemplary days using the proposed model. Though similar results were obtained for all the other days, they could

Day	Axis	Perf. metrics	Linear Regression	LASSO Regression	SVR (Polynomial)	SVR (Gaussian)	Traditional SAE	Proposed model
4 Apr	Axis 3	NMSE	0.1208	0.1231	0.1307	0.1308	0.1204	0.1162
		RMSE	5.0349	5.1313	7.0871	7.0864	5.0163	4.8417
4 Apr	Axis 4	NMSE	0.8018	0.8026	0.9593	0.9717	0.9800	0.6579
		RMSE	4.9956	5.0007	6.1880	6.1863	6.1061	4.0995
4 Apr	Spindle	NMSE	0.9406	0.9421	0.9955	0.9965	0.9745	0.8891
		RMSE	6.6704	6.6810	7.0711	7.0722	6.9114	6.3053
7 Apr	Axis 1	NMSE	0.6343	0.6353	0.6841	0.6852	0.7132	0.5842
		RMSE	5.1879	5.1959	6.1761	6.1635	5.8331	4.4397
7 Apr	Axis 3	NMSE	0.1039	0.1042	0.1132	0.1147	0.1002	0.0980
		RMSE	4.3305	4.3438	6.3841	6.3579	4.1767	4.0834
7 Apr	Spindle	NMSE	0.8675	0.8806	0.9978	0.9990	0.9809	0.8141
		RMSE	7.1641	7.2726	8.2515	8.2523	8.1010	6.7231
12 Apr	Axis 2	NMSE	0.6434	0.6477	0.7850	0.7857	0.5945	0.5587
		RMSE	7.7262	7.7779	9.7801	9.7791	7.1389	6.7093
12 Apr	Axis 3	NMSE	0.1082	0.1082	0.1168	0.1185	0.1026	0.1001
		RMSE	4.4554	4.4554	6.0906	6.0639	4.2260	4.1199
12 Apr	Axis 4	NMSE	0.7401	0.7591	0.9601	0.9806	0.9862	0.5372
		RMSE	4.8051	4.9286	6.4675	6.4629	6.4030	3.4877

 Table 2. Comparison of performance metrics using Linear, LASSO, SVR, traditional SAE and proposed regression model

not be presented here due to space constraints. We compare our results with that obtained via linear regression, LASSO, SVR using both the polynomial and gaussian kernels and the traditional stacked autoencoders (SAE) with two phase learning using backpropagation. We have reported the best results obtained using different parametric settings for the regression techniques mentioned in Table 2.

Normalised Mean Square Error (NMSE) and the Root Mean Square Error (RMSE) are used as performance metrics as they are widely used for making comparisons in the manufacturing domain regression techniques [20, 23]. They are given by:  $NMSE = \frac{\|y - \tilde{y}\|_2}{\|y\|_2}$  and RMSE =

 $\sqrt{\frac{\sum_{i=1}^{N} (y_i - \tilde{y}_i)^2}{N}}$  where y is the vector of actual load values measured from the axes,  $\tilde{y}$  is the estimated load vector using regression analysis and N = length(y).

It is quite obvious from the Table 2 that when the proposed regression model is applied, there is a consistent improvement in performance in terms of both the metrics over the traditional regression techniques. Since as per the results, the closest competitor was observed to be the linear regression in maximum of the cases and also for maintaining visual clarity, we have only compared against it in Fig. 3. Both (a) and (b) of Fig. 3 display portions of the load signals, pertaining to a different day, regressed using the proposed method as well as using linear regression along with the corresponding metric values. It can be seen clearly that the signal peaks could be better estimated using the proposed method.

## 5. CONCLUSION

In this work, we proposed a stacked autoencoder based novel regression framework. The focus of this paper is the problem



**Fig. 3**. Part of axial load values regressed using both linear regression and the proposed method

of load estimation in a real time smart machine using captured sensor signals. We compared our regression framework with the existing popular methods like linear regression, LASSO, SVR as well as traditional stacked autoencoder and observed that our method outperforms them.

Although the proposed model is applied in the manufacturing domain, the same can be applied to any kind of regression problems, which makes the solution generic. For instance, the model can be applied on problems like prediction of household electricity consumption, blood pressure estimation using PPG and ECG signals and the like.

### 6. REFERENCES

- R. Ramesh, S. Jyothirmai, and K. Lavanya, "Intelligent automation of design and manufacturing in machine tools using an open architecture motion controller," *Journal of Manufacturing Systems*, vol. 32, no. 1, pp. 248–259, 2013.
- [2] A. Simoneau and J. Meehan, "The impact of machining parameters on peak power and energy consumption in cnc endmilling," *Energy and Power*, vol. 3, no. 5, pp. 85–90, 2013.
- [3] A.K. Sahu, H.K. Narang, A.K. Sahu, and N.K. Sahu, "Machine economic life estimation based on depreciation-replacement model," *Cogent Engineering*, vol. 3, no. 1, 2016.
- [4] D. Cuppini, G. D'errico, and G. Rutelli, "Tool wear monitoring based on cutting power measurement," *Wear*, vol. 139, pp. 303–311, 1990.
- [5] N.K. Mandal, M. Mondal, and N.K. Singh, "Modelling and optimisation of a sustainable manufacturing process with cnc turning centre," *International Journal of Applied Environmental Sciences*, vol. 12, no. 6, pp. 1101– 1116, 2017.
- [6] D.Y. Pimenov, "Mathematical modeling of power spent in face milling taking into consideration tool wear," *Journal of Friction and Wear*, vol. 36, no. 1, pp. 45–48, 2015.
- [7] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in Advances in neural information processing systems, 2007, pp. 153–160.
- [8] K.O. Jörnsten, M. Näsberg, and P.A. Smeds, Variable splitting: A new Lagrangean relaxation approach to some mathematical programming models, University of Linköping, Department of Mathematics, 1985.
- [9] P.L. Combettes and J. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212. Springer, 2011.
- [10] J.A. Nelder and R.W.M. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society*. *Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972.
- [11] R.W.M. Wedderburn, "Quasi-likelihood functions, generalized linear models, and the gauss-newton method," *Biometrika*, vol. 61, no. 3, pp. 439–447, 1974.
- [12] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

- [13] E.A. Nadaraya, "On estimating regression," *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.
- [14] G.S. Watson, "Smooth regression analysis," Sankhyā: The Indian Journal of Statistics, Series A, pp. 359–372, 1964.
- [15] H. Drucker, C.J.C. Burges, L. Kaufman, A.J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155–161.
- [16] R.S. Kadu, G.K. Awari, and C.N. Sakhale, "Determination and analysis of spindle load in boring machining operation on cast iron using mathematical models," *International Journal of Engineering Development and Research*, vol. 2, no. 1, pp. 1002–1014, 2014.
- [17] Q. Yongliang, "Research on real time compensation of thermal errors of cnc lathe based on linear regression theory," in *Machinery, Materials Engineering, Chemical Engineering and Biotechnology, 2015 2nd International Conference on*, 2016, Advances in Engineering Research.
- [18] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak, "Health assessment and life prediction of cutting tools based on support vector regression," *Journal of Intelligent Manufacturing*, vol. 26, no. 2, pp. 213– 223, 2015.
- [19] V.T. Tran, H.T. Pham, B. Yang, and T.T. Nguyen, "Machine performance degradation assessment and remaining useful life prediction using proportional hazard model and support vector machine," *Mechanical Systems and Signal Processing*, vol. 32, pp. 320–330, 2012.
- [20] P. Palanisamy, I. Rajendran, and S. Shanmugasundaram, "Prediction of tool wear using regression and ann models in end-milling operation," *The International Journal* of Advanced Manufacturing Technology, vol. 37, no. 1, pp. 29–41, 2008.
- [21] A. Majumdar and A. Tripathi, "Asymmetric stacked autoencoder," in *Neural Networks (IJCNN)*, 2017 International Joint Conference on. IEEE, 2017, pp. 911–918.
- [22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends*(R) *in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [23] A. E. Ouafi and N. Barka, "A multisensor approach for error compensation in cnc machine tools," *Journal* of Automation and Control Engineering, vol. 1, no. 3, 2013.