

# Deep Learning Approach for Industrial Process Improvement

Sin Yong Teng<sup>a,\*</sup>, Vítězslav Máša<sup>a</sup>, Petr Stehlík<sup>a</sup>, Hon Loong Lam<sup>b</sup>

<sup>a</sup>Brno University of Technology, Institute of Process Engineering & NETME Centre, Technická 2896/2, 616 69 Brno, Czech Republic

<sup>b</sup>Department of Chemical and Environmental Engineering, University of Nottingham, Malaysia.  
 Sin.Yong.Teng@vut.cz

The full operation of an industrial processing facility with artificial intelligence has been the holy grail of Industry 4.0. One of the inherent difficulties is the enumerate and complex nature of processing information within an industrial plant. Hence, such data should be processed efficiently. This paper demonstrates the effectiveness of a deep auto-encoder neural network for the dimensionality reduction of industrial processing data. The deep auto-encoder neural network functions to intake all possible processing data from the processing system by sending it into an encoder neural network. Subsequently, the encoder condenses the data into highly compressed encoded variables. The network is trained in an unsupervised manner, where a decoder neural network simultaneously attempts to revert the encoded variables to their original form. Such a deep learning approach allows data to be highly compressed into lower dimensions. The coded variables retain critical information of the processing system, allowing reconstruction of the full process data. Auto-encoder neural networks are also able to provide noise removal for encoded data. For application, the encoded variable can be utilized as an effective dimension-reduced variable that can be used for plant-wide optimization. This paper also discusses the further applications of encoded variables for industrial process improvements using the Industrial Internet of Things (IIoT) technologies.

## 1. Introduction

Deep learning is a specialized strain of machine learning which utilizes computation models with multiple processing layers to learn representations of data with multiple levels of abstraction (LeCun et al., 2015). One of the core-enabling technology of deep learning is the invention of deep neural networks. Neural networks originate from the fields of neuroscience where McCulloch and Pitts (1943) first formalized the mathematical model of the biological neural activity inside the brain. The first model of this neural graph consists of three main components, which are the dendrites, soma, and axon. This simple model of the neuron is later referred to as the McCulloch-Pitts neuron and has opened the doors to the popular deep learning as it is today. Although behaviors of nervous activity can be explained using McCulloch and Pitts' approach, there was one key dynamic factor that was missing, that is the simulation of neural learning. This piece of irreplaceable technology for the development of deep learning is later implemented using the theory of backpropagation. Backpropagation is based on the method of automatic differentiation which was first published by Seppo (1970) in his master's thesis. Later, Rumelhart et al. (1988) popularized the use of the backpropagation method by applying it to learning representations in a self-organizing neural network. Today, deep learning has demonstrated remarkable success in applications of computer vision, medical diagnosis, phonetic identification, voice recognition, feature coding, natural language processing, robotics, computer games, molecular and drug analysis (Deng and Yu, 2014).

The pivotal factor that differentiates deep learning as a subfield of machine learning is the action of learning intrinsic representations in a hierarchical manner (Bengio et al., 2013). Some research works such as Mhaskar and Poggio (2016) classifies single hidden layer neural networks as shallow networks while neural network with multiple hidden layers is classified as deep neural networks. Poggio et al. (2017) have also reviewed and compiled mathematical proofs and experiments of situations which deep networks perform better than shallow networks. Even for practical applications, deep learning has substantially outperformed other good old-

fashioned Artificial Intelligence (GOFAI) in the aspects of expressibility, efficiency, and learnability (Lin et al., 2017). Although there are occasionally implementations of shallow learning that can attain performance close to deep learning (Ba and Caruana, 2014), the state-of-art record holder for application performances are always either deep, or very deep neural networks (Szegedy et al., 2017).

The first step to transfer deep learning techniques to process system engineering (PSE) is not to start from the commonly used supervised learning. Statistical methods dominate the field of predictive analysis in PSE, as shown in the works of Boukouvala et al. (2010) where surface response methodology was used to predict missing and noisy data. Furthermore, Teng et al. (2019) demonstrated a novel principal component-aided statistical process optimization (PASPO) method which achieved state-of-art results in a real oil refinery. Although deep learning methodologies often outperform statistical methods (Máša et al., 2018), these statistical methods have already proven their robustness and firmed their roots in the grounds of PSE predictive analysis. Dimension reduction using deep learning (Hinton and Salakhutdinov, 2006) is one of the most promising and implementable directions for PSE. Traditionally, dimension reduction techniques are already applied in PSE, such as the work of Li and Wang (2002), which utilized independent component analysis (ICA) for dimension reduction of dynamic trends. Principal component analysis (PCA) was utilized to evaluate (How and Lam, 2017) and to debottleneck (How and Lam, 2018) integrated biomass supply chain. With the requirements of a leaner and greener manufacturing process (Leong et al., 2019), there is a need for more advanced dimension reduction techniques (Lam et al., 2011).

Main problems of implementing deep and reinforcement learning in the real world are: (i) Rewards in a real-world setting are non-direct, sparse and sometimes imperfect; (ii) Models are often inaccurate or non-robust for the use of model-based method; (iii) Other technical challenges such as model seedings and network architecture (Henderson et al., 2018). Despite these challenges, three main topics within PSE are the most potential areas for the implementation of deep learning techniques, which are Process Integration, process optimization, and process intensification. By insightful speculation, the latter topic is the more difficult area to implement techniques of deep learning.

The contribution of this paper is to extend the ideologies and methods of deep feature learning towards PSE. This work also demonstrates the use of a novel autoencoder for process system engineering which effectively outperformed the engineering state-of-art analysis. The specialty of Evolutionary Deep Autoencoder is that both neural architecture and activation functions are being co-optimized in the training process.

## 2. Methods and theory

The overall framework of this paper is shown in Figure 1. The framework is implemented in a continuous process improvement cycle in order to consistently optimize and debottleneck the processing system of interest. Process Integration, optimization, and intensification focus on the interaction between units, optimality of operational parameters and higher performance density respectively. Objective evaluation ensures that criteria for process improvements are being achieved.

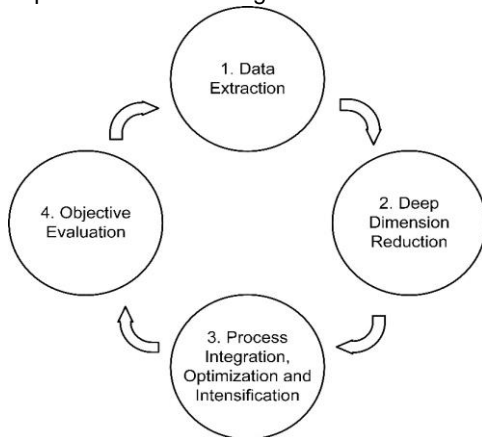


Figure 1: Theoretical framework of Deep process improvement

### 2.1 Learnable Data Extraction in the Processing Industry

The availability of learnable data varies for each industrial facility at a different phase. From the experience of industrial projects, the availability of useful data is maximal during initial data extraction. The most expensive phase for data collection is during the calibration and testing phase as bad implementation can directly damage

processing equipment. This highlights the criticality of front-end loading by doing data analysis using deep learning before calibration and testing (See Figure 2).

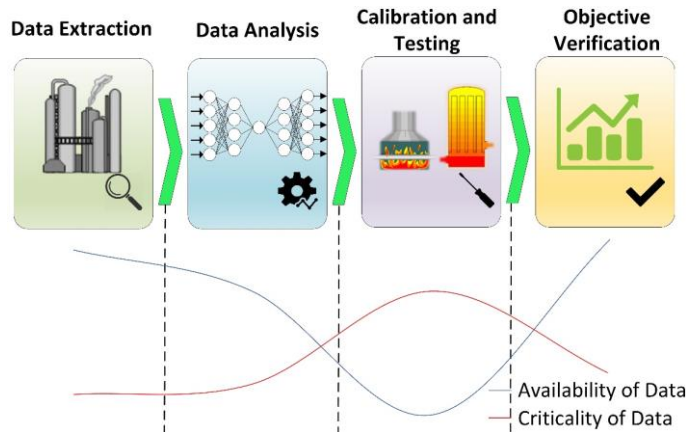


Figure 2: Availability and criticality of data at different phases of industrial process improvement projects

## 2.2 Deep Dimension Reduction

This paper utilizes a deep autoencoder to process large and sparse data collected from industrial facilities. An autoencoder is a special type of neural network that is used for feature learning in an unsupervised manner. Information from a high dimensional input can be hierarchically encoded into network layers of lesser neurons (See figure 3). The layer with the least number of neurons is called the bottleneck layer, in which information is forced through a small-dimensional latent space. To ensure that the latent variables retain a large representation of the data, a decoder neural network attempts to map the latent variables to the original input.

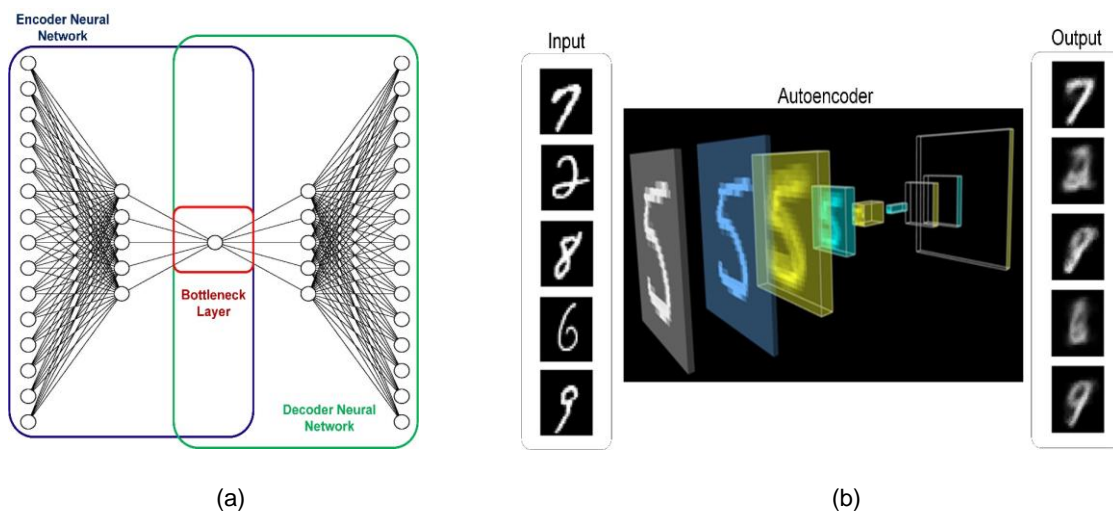


Figure 3: (a) Structure of autoencoder (b) Autoencoder retaining information under dimension reduction (Toy problem using 60k MNIST dataset)

A visual example has been created using JavaScript language to visualize the mechanisms of an autoencoder can be observed in Figure 3(b). A convolutional autoencoder is trained on the MNIST handwritten digit benchmark dataset. Handwritten digits from the input can be compressed into a small latent space, and then be dimensionally expanded to retain critical information of the input. Although the reconstruction of information is not perfect, the autoencoder learns critical features of the full data set and generalizes the learning output. In this work, an evolutionary autoencoder (also known as EvoAE) is implemented using Elitist Strategy Genetic Algorithm (ESGA) with adaptive learning rate optimization algorithm (Adam) to reduce the dimension of the processing system. The uniqueness of evolutionary autoencoders is such that both the activation functions and neural architecture are also hyper-optimized in the learning process. All computation work was implemented in

Python and C/C++ language using custom codes and libraries such as Sklearn and Tensorflow. Prior to dimension reduction, all data were normalized using a min-max normalization routine to standardize data between 0 and 1. Mean absolute error (MAE) was chosen as the benchmark criteria for dimension reduction accuracy because it gives a proportionate indication to the variability of data in the processing system.

### 2.3 Case Study

For the purpose of demonstration, a case study which consists of an area of the oil refinery plant is simulated. The process consists of oil being heated and then flashed in a flash tower with a recycle loop. The vapor top product is cooled into a stream of light oil and is collected, while the bottom oil product is stored for further processing (See Figure 4).

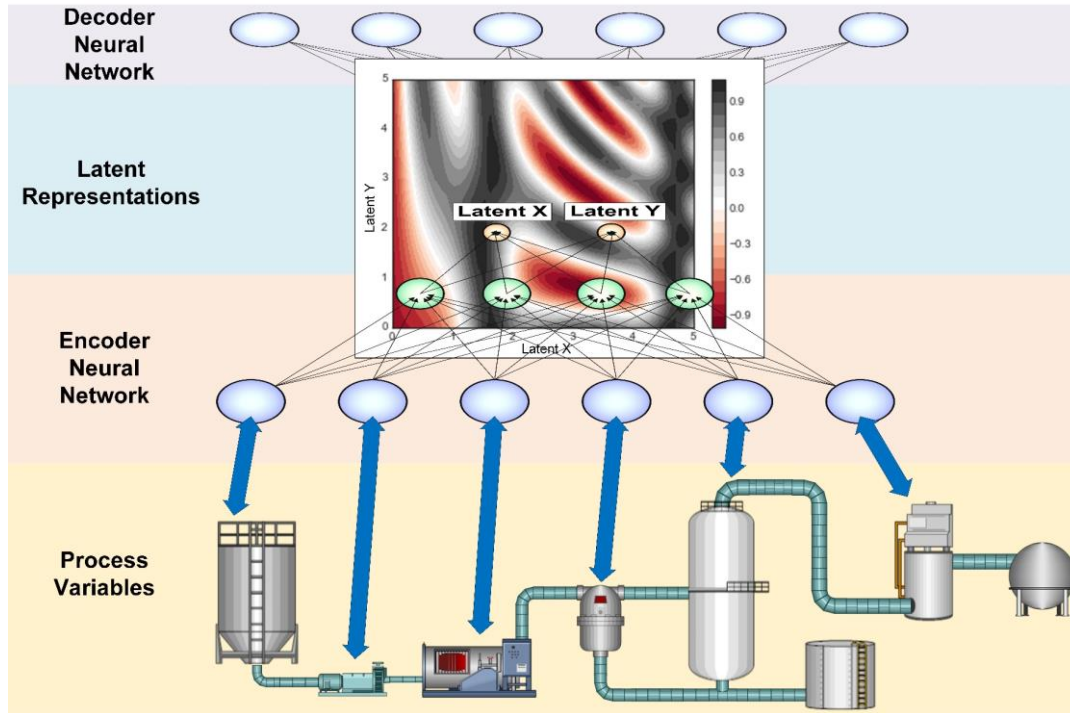


Figure 4: Case study and usage of autoencoder for process improvement

Monte Carlo simulation was implemented on the simulation model with Gaussian noises to stochastically generate process variables (See Eq(1) and Eq(2)). A total of 13,000 data points was simulated using this approach. 10,000 data points were allocated for training, while the remaining 3,000 data points were used for validation. 15 dimensions of process variables were recorded, which includes temperature, pressure, flow rate, density and viscosity of processing fluid at different parts of the process.

$$y(s) = m(s) + \varepsilon \quad (1)$$

$$\varepsilon \sim N(0, \sigma^2) \quad (2)$$

Where  $y(s)$  is the stochastic process variables at a discrete state “s”,  $m$  is the stochastic ground truth at discrete state “s”, and  $\varepsilon$  is Gaussian noise which follows a normal distribution at 99% confidence interval.

### 3. Results and discussion

By using the processing information dataset that contains noise and stochasticity, an autoencoder was trained using the neuro-evolution approach allowing variability of neural architectures. A latent dimension of 2 was specified to enable for effective interpretability of reduced variables. Neuro-evolution has shown no significant improvements in performance at the 100<sup>th</sup> generation with a structure of [15, 25, 10, 2, 2, 13, 15] and training was stopped. The detailed architecture of the neural network is shown in Figure 5.

The performance of the evolutionary deep autoencoder is compared with state-of-art dimension reduction methodologies and was shown to out-perform all prior process system engineering methods (See Table 1).

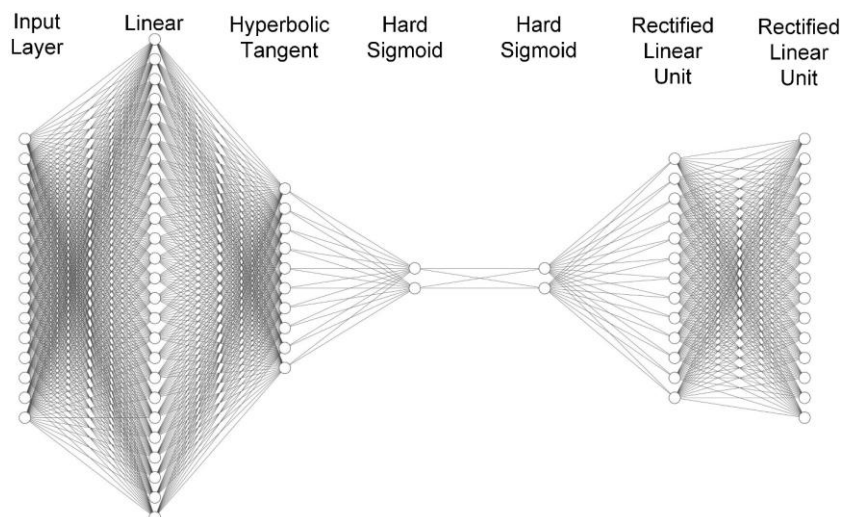


Figure 5: Optimal structure of evolutionary deep autoencoder

Table 1: Comparison between state-of-art dimension reduction methods using 2 latent dimensions

Dimension Reduction Method	Code Implementation Reference	Validation Mean Absolute Error (%)
Principal Component Analysis (PCA)	Tipping and Bishop (1999)	12.07
Linear Kernel PCA	Schölkopf et al. (1997)	12.07
RBF Kernel PCA	Schölkopf et al. (1997)	11.93
Polynomial Kernel PCA	Schölkopf et al. (1997)	11.52
Sigmoid Kernel PCA	Schölkopf et al. (1997)	14.03
Independent Component Analysis (ICA)	Hyvärinen and Oja. (2000)	12.07
Non-negative Matrix Factorization (NMF)	Févotte and Idier (2011)	16.52
Evolutionary Deep Autoencoder	This work	9.54
Benchmark Autoencoder	See footnote*	16.75

\*The autoencoder architecture is [15,11,7,2,7,11,15] and activation functions are all sigmoid. This is a conventional heuristic which approximately equalizes changes in neurons of each hidden layer.

### 3.1 Applications

Dimension reduction using evolutionary deep autoencoder enables more efficient consideration of variables and a compressed search space for process optimization, integration and intensification. Using such techniques, the most critical aspects of the processing system can be targeted and improved. The bandwidth of transferring processing information can also be reduced using this approach to enable low latency protocols such as LongRange (LoRa) in the Industrial Internet of Things (IIoT) settings for rapid industrial improvement.

### 3.2 Limitations and future works

The main limitation of this approach is the availability and reliability of real data. In future works, extended use of the Evolutionary Deep Autoencoder to efficiently carry out fully autonomous plant-wide optimization. A more detailed implementation guideline for objective evaluation will also be provided in future works.

## 4. Conclusions

This work has demonstrated the use of novel evolutionary deep autoencoder for the dimension reduction of industrial processing systems. The autoencoder neural network was applied to reduce process information within an oil refinery processing system and achieved a Mean Absolute Error of 9.54 % with only two latent dimensions. By comparing with methods such as Independent Component Analysis (ICA), Non-negative Matrix Factorization (NMF) and various types of Principal Component Analysis (PCA), the proposed approach significantly outperformed the state-of-art methods in process system engineering.

## Acknowledgements

The research leading to these results has received funding from the Ministry of Education, Youth and Sports of the Czech Republic under OP RDE grant number CZ.02.1.01/0.0/0.0/16\_026/0008413 "Strategic Partnership for Environmental Technologies and Energy Production".

## References

- Ba, J., Caruana, R., 2014. Do deep nets really need to be deep? NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, Canada.
- Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798-1828.
- Boukouvala, F., Muzzio, F.J., Ierapetritou, M.G., 2010. Predictive modelling of pharmaceutical processes with missing and noisy data, *AIChE Journal*, 56(11), 2860-2872.
- Deng, L., Yu, D., 2014. Deep learning: methods and applications, *Foundations and Trends® in Signal Processing*, 7(3-4), 197-387.
- Févotte, C., Idier, J., 2011. Algorithms for nonnegative matrix factorization with the  $\beta$ -divergence, *Neural Computation*, 23(9), 2421-2456.
- Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., Meger, D., 2018. Deep reinforcement learning that matters, In *Thirty-Second AAAI Conference on Artificial Intelligence*, Louisiana, USA.
- Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks, *Science*, 313(5786), 504-507.
- How B.S., Lam H.L., 2017. Novel evaluation approach for biomass supply chain: an extended application of PCA, *Chemical Engineering Transactions*, 61, 1591-1596.
- How, B.S., Lam, H.L., 2018. PCA method for debottlenecking of sustainability performance in integrated biomass supply chain, *Process Integration and Optimization for Sustainability*, 3(1), 43-64.
- Hyvärinen, A., Oja, E., 2000. Independent component analysis: algorithms and applications, *Neural networks*, 13(4-5), 411-430.
- Lam, H.L., Klemeš, J.J., Kravanja, Z., 2011. Model-size reduction techniques for large-scale biomass production and supply networks, *Energy*, 36(8), 4599-4608.
- LeCun, Y., Bengio, Y., Hinton, G. 2015. Deep learning, *Nature*, 521(7553), 436-444.
- Leong, W.D., Lam, H.L., Ng, W.P.Q., Lim, C.H., Tan, C.P., Ponnambalam, S.G., 2019. Lean and Green Manufacturing—a Review on its Applications and Impacts, *Process Integration and Optimization for Sustainability*, 3(1), 5-23.
- Li, R.F., Wang, X.Z., 2002. Dimension reduction of process dynamic trends using independent component analysis, *Computers & Chemical Engineering*, 26(3), 467-473.
- Lin, H.W., Tegmark, M., Rolnick, D., 2017. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6), 1223-1247.
- Máša, V., Stehlík, P., Touš, M., Vondra, M., 2018. Key pillars of successful energy saving projects in small and medium industrial enterprises, *Energy*, 158, 293-304.
- McCulloch, W., Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity, *The Bulletin of Mathematical Biophysics*, 5(4), 115-133.
- Mhaskar, H.N., Poggio, T., 2016. Deep vs. shallow networks: An approximation theory perspective, *Analysis and Applications*, 14(6), 829-848.
- Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., Liao, Q., 2017. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review, *International Journal of Automation and Computing*, 14(5), 503-519.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1988. Learning representations by back-propagating errors, *Cognitive modelling*, 5(3), 1.
- Schölkopf, B., Smola, A., Müller, K.R., 1997. Kernel principal component analysis. In *International conference on artificial neural networks*, Springer, Berlin, Heidelberg.
- Seppo, L., 1970. The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors, Master's thesis, University of Helsinki, Helsinki, Finland.
- Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning, In *Thirty-First AAAI Conference on Artificial Intelligence*, California, USA.
- Teng, S., How, B., Leong, W., Teoh, J., Siang Cheah, A., Motavasel, Z., Lam, H. (2019). Principal component analysis-aided statistical process optimization (PASPO) for process improvement in industrial refineries, *Journal of Cleaner Production*, 225, 359-375.
- Tipping, M.E., Bishop, C.M., 1999. Probabilistic principal component analysis, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3), 611-622.