

Selection of pre-training parameters for synthesizing surrogate models of gas turbine units for gas turbine electro power stations

Grigory Kilin¹, Boris Kavalero¹, Artem Suslov^{1*}, and Ilya Tyatenkov¹

¹Perm National Research Polytechnic University, 29, Komsomolsky prospect, Perm, 614990, Russian Federation

Abstract. The article is devoted to the current task of selecting pre-training parameters for the synthesis of surrogate models, which is a key factor in creating high-performance models of complex technological objects. During the study, the authors conduct a systematic analysis of various parameters and their interactions, including determining the optimal number of training iterations, the number of trainable layers, and the number of neurons in these layers. Thanks to this approach, the results of the presented study can significantly improve the accuracy and efficiency of surrogate models, which in turn leads to simplification and acceleration of the process of their development and application in various fields of science and engineering.

1 Introduction

Surrogate models are crucial in a variety of science and technology fields, as they enable the approximation of intricate and resource-intensive processes using more accessible and rapid methods. They are widely utilized in optimization, prediction, and control tasks, where timely results and high precision are essential.

In this article, the pre-training used to obtain surrogate models of gas turbine units (GTUs) in gas turbine electro power stations (GTEPS). One approach to developing surrogate models [1-4] is by employing the theory of artificial neural networks (ANNs) [5-6]. However, a key challenge in creating these models is the prolonged training process. The data for this element was obtained in a computer simulation of a GTPES in different modes of operation and with different configuration of the power system. For computer simulation were used classic models of GTU, synchronous generator (SG), automatic control systems of both GTU and SG.

To address this issue, pre-training [7] based on an autoencoder [8] can be implemented as an effective strategy for accelerating the training procedure. Ensuring high accuracy and efficiency of pre-training requires the appropriate selection of ANN parameters, as these factors have a considerable impact on the training process and the eventual performance of the models. In this article, we delve into a detailed analysis of pre-training parameters,

* Corresponding author: AlSuslov@pstu.ru

focusing on aspects such as the optimal number of training iterations, the ideal number of layers to train, and the suitable quantity of neurons within these layers, with the aim of refining the overall effectiveness of surrogate models.

2 Materials and methods

Pre-training involves training an artificial neural network on a comprehensive dataset before utilizing it for a designated task, enabling the extraction of generalized knowledge and enhancing the model's performance during subsequent fine-tuning on task-specific data. To apply this approach, an extensive dataset consisting of experimental data on the functioning of various GTUs in GTEPS across multiple operating modes and connection configurations was collected, amounting to 971532 data points.

This method necessitates conducting a series of experiments on pre-training an autoencoder [8] while varying the number of neurons in the hidden layers to pinpoint the optimal number of training iterations. Following this, the hidden layers from these autoencoders will be "extracted" and employed in constructing pre-trained ANNs. In the concluding phase, the change in training error between the pre-trained ANNs and the baseline ANN will be assessed. The variables and dependencies required for ANN training are determined in GTEPS computer simulations on classical models.

The advantage of surrogate models over classical ones is that they are built for a specific area of study of the object's operation. Pre-training helps in this by the fact that there is no need for a neuromodel for all modes of operation. Instead, pre-trained neuromodels for specific operating modes and energy power system configurations are used (Figure 1).

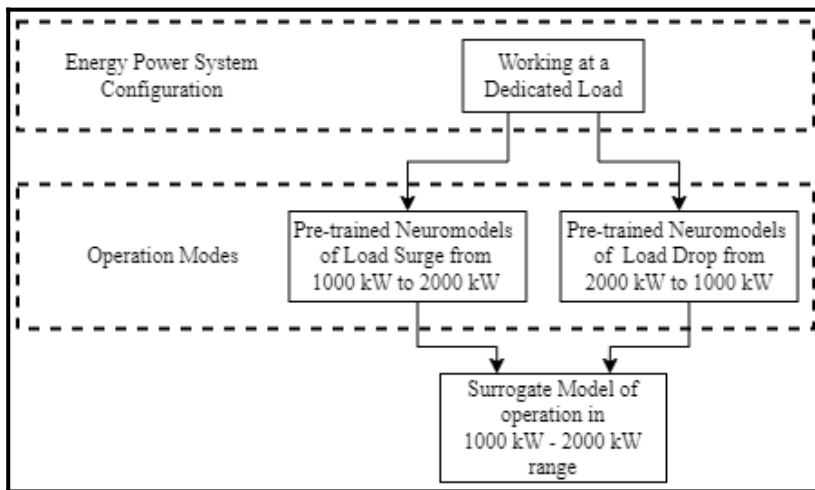


Fig. 1. Diagram of a surrogate model obtaining from pre-trained neuromodels.

Therefore, to demonstrate the advantages of pre-training and the correct choice of pre-training parameters, the graphs of the change in error for pre-trained ANNs and baseline ANNs when trained on data from computer simulations of the operation of a gas-turbine electro power station connected to a 1000 kW load (Figure 2) were compared. The simulation mode involved load surge from 1000 kW to 2000 kW.

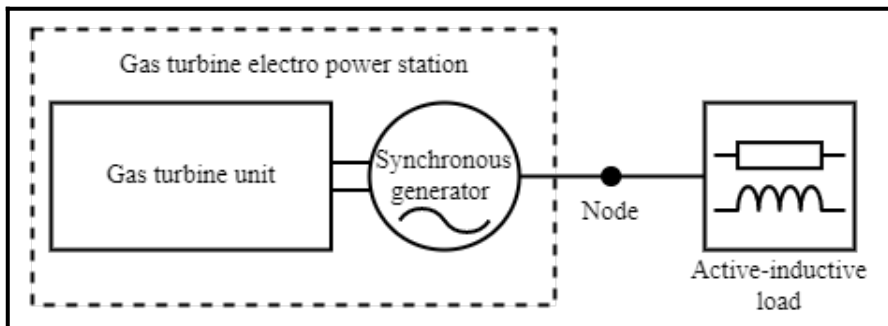


Fig. 2. Diagram of a gas turbine power plant operating at a dedicated load.

It is important to note that the architecture of the baseline ANN is identical to that of the pre-trained ANN, with the exception that the baseline ANN's weights are initialized randomly during its creation. By implementing this method, it is anticipated that the overall effectiveness and efficiency of the ANNs in addressing complex tasks will be significantly improved.

3 Results

Six experiments were conducted, with three experiments for each of the two autoencoder architectures with the following parameters:

- Five neurons in the output layer, ten neurons in the hidden layer, and five neurons in the output layer (architecture 5-10-5).
- Five neurons in the output layer, thirty neurons in the hidden layer, and five neurons in the output layer (architecture 5-30-5). The results are presented in Figure 3.

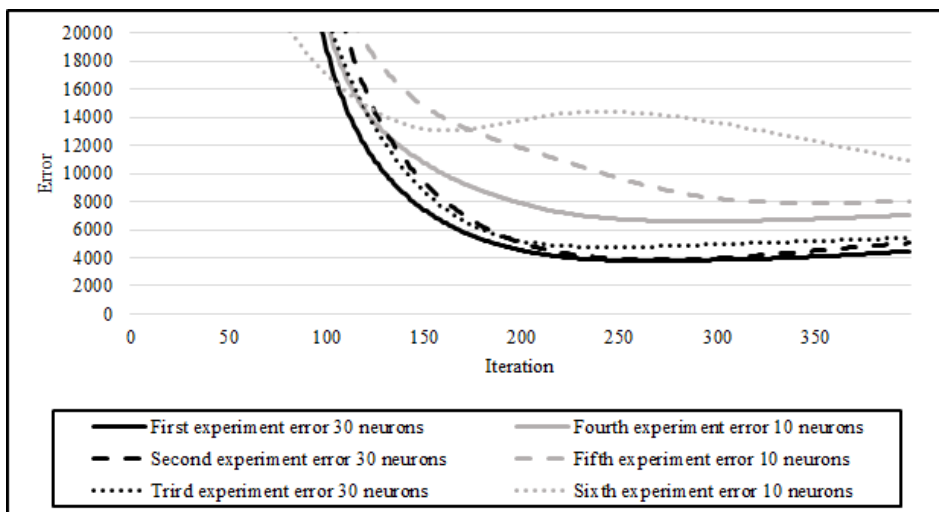


Fig. 3. Autoencoder training error (black lines - experiments with the 5-30-5 neural network architecture; grey lines - experiments with the 5-10-5 neural network architecture).

In Figure 3, it can be seen that experiments with the 5-30-5 neural network architecture showed less training error over 400 iterations compared to experiments with the 5-10-5

neural network architecture. The smallest error value in the first experiment with the 5-30-5 architecture is 42.5% lower than the smallest error value in the fourth experiment with the 5-10-5 architecture (Table 1). Additionally, from the graphs in Figure 3, it can be concluded that 250 training iterations are sufficient for both the 5-30-5 and 5-10-5 autoencoder architectures.

Table 1. Autoencoder training error.

Training process name	Smallest error, units	Iteration with the smallest error
First experiment 30 neurons	3786	264
Second experiment 30 neurons	3872	271
Third experiment 30 neurons	4753	245
Fourth experiment 10 neurons	6589	289
Fifth experiment 10 neurons	7887	348
Sixth experiment 10 neurons	10923	398

Based on each autoencoder architecture, pre-trained ANNs were formed:

- With architecture parameters of three neurons in the output layer, ten neurons in each of the three hidden layers, and five neurons in the output layer (architecture 3-10-10-10-5).
- With architecture parameters of three neurons in the output layer, thirty neurons in each of the three hidden layers, and five neurons in the output layer (architecture 3-30-30-30-5).

Figure 4 shows a comparison of pre-trained neural networks with the baseline neural networks.

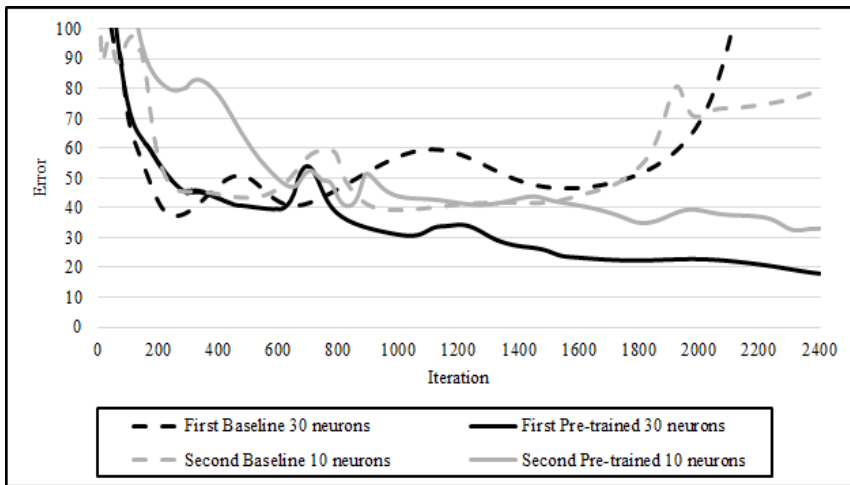


Fig. 4. Training error of baseline and pre-trained neural models with three hidden layers (black lines - experiments with the 3-30-30-30-5 neural network architecture; grey lines - experiments with the 3-10-10-10-5 neural network architecture).

In figure 4, it is evident that the pre-trained ANNs demonstrated the lowest training errors throughout all 2400 iterations. The first experiment, which utilized pre-trained models with 30 neurons in the hidden layers, displayed a reduced error across the entire training process when compared to the second experiment that involved pre-trained ANNs with 10 neurons in each hidden layer. It can also be noted that the pre-trained ANNs continue to learn even after 2400 training iterations. In contrast, the baseline ANNs ceased

learning after merely 200 iterations, as observed by the halted decrease in the ANN's training error.

4 Discussion

Let's elaborate on the findings presented in section 3. Specifically, within the pre-trained models, the experiment featuring 30 neurons demonstrates a significantly smaller error of 17.94 units when compared to the corresponding experiment utilizing 10 neurons, which resulted in an error of 32.42 units (Table 2). This indicates that the 30-neuron experiment boasts a remarkable 51.84% reduction in error compared to the 10-neuron experiment.

Table 2. Training error of baseline and pre-trained neural models with three hidden layers.

Training process name	Smallest error, units	Iteration with the smallest error
First Baseline 30 neurons	37.28	256
Second Baseline 10 neurons	39.04	993
First Pre-trained 30 neurons	17.94	2400
Second Pre-trained 10 neurons	32.42	2326

Moreover, it is worth noting that the smallest error observed in the pre-trained model with 30 neurons is 51.87% lower than the least error detected in the baseline model with 30 neurons. The smaller error in the pre-trained model arises from the minimal training error recorded in the baseline model with 30 neurons, which measured 37.28 units.

In addition, the pre-trained model with 30 neurons displays the smallest error, which is 54.02% less than the least error found in the baseline model with 10 neurons. This contrast in error stems from the minimal training error in the baseline model with 10 neurons, which amounted to 39.04 units. By examining these metrics, we can deduce that pre-trained models with 30 neurons outperform baseline models in terms of minimizing error, ultimately leading to more accurate and reliable predictions.

Moving forward, our next mission focuses on thoroughly investigating the performance of these pre-trained models when confronted with data collected from the operation of GTEPS, taking into account a different power supply arrangement and operational mode. To achieve this, we will delve into the data obtained from GTEPS while it is functioning on a dedicated load and simultaneously operating in parallel with an infinitely capable network, a scenario which provides us with new insights and challenges (Figure 5).

By expanding our analysis in this manner, we will enhance our understanding of the versatility and adaptability of the pre-trained models when faced with varying conditions and configurations, ultimately leading to more robust and reliable neural networks in real-world applications.

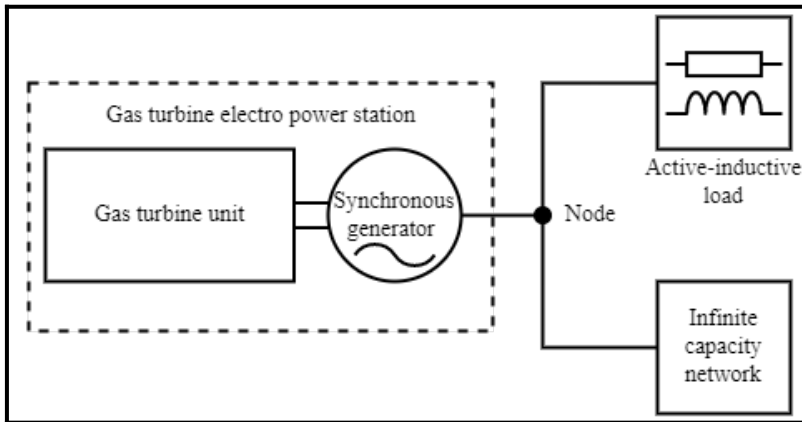


Fig. 5. GTEPS operating on a dedicated load parallel to the infinite capacity network.

5 Conclusions

The experimental results clearly showed that it is sufficient to use 250 training iterations to obtain pre-trained neural networks, the advantages of which are visible in Figure 4. However, the situation is slightly different when it comes to the number of hidden layers and the number of neurons in these layers. The main advantage of pre-training is that it is performed before the direct synthesis of the neural network to solve a specific task (for example, modeling of GTU or GTEPS [9-10]). To address the need for efficient and effective neural network training, a strategy of pre-training a vast number of hidden layers of autoencoders has emerged. This process results in the creation of a bank of pre-trained layers (BPL) consisting of multiple autoencoders, each with a different number of neurons in their respective hidden layers. The benefit of constructing a BPL is that it provides a ready-made solution to tackle specific tasks, as the pre-trained neural network can be assembled from the hidden layers of autoencoders within the BPL, taking into account the requisite volume of experimental data necessary for training.

This approach streamlines the process of neural network creation, reducing the need for extensive training and enabling faster model convergence. By leveraging pre-trained layers, neural networks can be quickly assembled and fine-tuned to adapt to new data and environments. Overall, the use of BPL and pre-trained neural networks can accelerate the deployment of machine learning solutions, enhancing the efficiency and effectiveness of decision-making processes across a range of domains.

The findings from our research have significant implications for the advancement and optimization of surrogate models pertaining to GTU and GTEPS. As we move forward with our investigations, we intend to delve deeper into understanding the impact of pre-training parameters on the quality of surrogate models. Additionally, we aim to devise strategies for the automated selection of optimal parameters, which will be tailored to the unique requirements of specific tasks and application domains. By doing so, we aspire to contribute to the improvement and fine-tuning of surrogate models, ultimately enhancing their accuracy and efficiency in a diverse range of contexts.

Acknowledgements

The research was carried out with the support of the Russian Science Foundation grant No. 23-29-00381, <https://rscf.ru/project/23-29-00381/>.

References

1. S. Grihon, S. Alestra, D. Bettebghor, E. Burnaev, P. Prikhodko, *Comparison of different techniques for surrogate modeling of stability constraints for composite structures*, in Proceedings of 1st International Conference on Composite Dynamics, DYNACOMP, 22-24 May 2012, Archanon, France (2012)
2. M. Asher, B. Croke, A. Jakeman, L. Peeters, *Water Resources Research*, **51**, 5957 (2015)
3. G. Sun, S. Wang, *J. Aerospace E.*, **233**, 5863 (2019)
4. A. Forrester, A. Sobester, A. Keane, *Engineering design via surrogate modelling: a practical guide* (John Wiley & Sons, New York City, 2008)
5. S. Haykin, *Neural networks: a complete course* (Williams Publishing House, Moscow, 2006)
6. K. Robert, *Basic search neural networks* (Williams Publishing House, Moscow, 2001)
7. D. Erhan, Y. Bengio, A. Courville, P. Vincent, P. Manzagol, S. Bengio, *J. Machine Learning Research*, **11**, 625 (2010)
8. C.Y. Liou, C.W. Cheng, J.W. Liou, D.R. Liou, *Intelligent Science and Intelligent Data Engineering*, **7751**, 458 (2013)
9. H. Asgari, X.Q. Chen, R. Sainudiin, *I.J. of Modeling, Identification and Control*, **20**, 253 (2013)
10. A.I. Abdalnagimov, G.K. Ageev, *Bulletin of the Ufa State Aviation Technical University*, **23**, 86 (2019)