Application of the Attention Mechanism in LSTM Networks to Predict the Parameters of Technological Systems

Maxim Dli[1][a], Andrey Puchkov[1][b], Ekaterina Rysina[1][c] and Andrey Sokolov[1][d]
Branch of the National Research University Moscow Power Engineering Institute in Smolensk,
1 Energeticheskiy Dr., Smolensk, 214013, Russia
midli@mail.ru, putchkov63@mail.ru, lobaneva94@mail.ru, andreisokol98@gmail.com

Keywords: Attention mechanism in deep neural networks, Time series forecasting.

Abstract: The paper develops a method for predicting the parameters of a complex chemical energy-technological system of recycling wastes from apatite-nepheline ore processing. The relevance of the study is explained by the necessity of decreasing environmental damage caused by mining-and-processing integrated works to adjacent areas and by the recyclability of the wastes having a rich chemical and mineralogical composition, which enables them to be viewed as technogenic deposits. The technological system implements the process of yellow phosphorus production from apatite-nepheline ores, the source information about its parameters being presented as a multivariate time series. For forecasting, the method employs a long short-term memory deep recurrent neural network. The network is an autoencoder-based one, with an encoder at the input and a decoder at the output. The novelty of the method lies in the adaptation of the autoencoder architecture to solving the problem of predicting the parameters of a technological system and in the introduction of an attention mechanism into its structure. This allows the decoder to receive information about the intermediate internal states of the encoder and increases the forecast accuracy, as is confirmed by a model experiment performed in Matlab.

1 INTRODUCTION

The application of cyber-physical systems to implement various processes implies using hardware (units, devices) responsible for performing different stages of these processes. A complex chemical energy-technological system (CETS) of phosphorus production from apatite-nepheline ore waste, developed within the concept of circular economy, is an example of such hardware (Puchkov, 2022). It comprises three tandem process units: a granulator, a conveyor-type multi-chamber calcining machine (CTMCM), and an ore thermal furnace (OTF) (Meshalkin, 2019). The CETS is characterized by large amounts of consumed power, about several tens of megawatt. This foregrounds the problem of optimizing the power consumption of the CETS since even a small percentage of energy saving results in large values in absolute terms. Within solving the problem of optimizing the power consumption of the CETS, it is necessary to have data on the current state of the process, as well as the forecast of its changes at various time horizons. The current state is estimated by the information coming from the instrumentation, and the prediction functions are implemented by the information and analysis support of the intelligent monitoring system; the improvement of this support increases the prediction quality due to the application of new methods and algorithms (Solopov, 2021). They are currently being rapidly developed for various applications and subject areas due to great prospects offered by deep learning (DL) as a branch of machine learning (ML) and artificial intelligence (AI). DL is now considered to be the main technology of Industrial Revolution 4.0 (Sarker, 2021), which

[a] https://orcid.org/0000-0002-5657-0892
[b] https://orcid.org/0000-0003-2616-0259
[c] https://orcid.org/0000-0002-0695-5145
[d] https://orcid.org/0000-0003-1357-4807

implies a wide-scale application of cyber-physical systems to meeting human needs and manufacture.

The aim of this study is to improve the CETS parameter prediction method based on the application of a deep artificial neural network (DNN) through the implementation of the attention mechanism (AM). This mechanism allows searching for interrelations of input and output data sequence lengths (Vaswani, 2017).

Attention is a metaphoric term in this context, i.e. AM simply facilitates *focusing* on separate data sequence elements which are the most significant in DNN response production. Initially, neural networks with AM were meant for natural language processing. However, it follows from the summary of the 4th annual report on the state of AI, recently published by N. Benaich, general partner of Air Street Capital, a venture company specializing in AI startups (https://www.stateof.ai/), that AM technologies have turned into general-purpose architecture for ML. General-purpose architecture implies that AM enables one to work with any input data if a neural architecture returning the set of annotation vectors can be selected for them. The Transformer neural network architecture presented by Google Brain researchers in 2017 and based on the attention mechanism became the main trend in AI systems in 2021, and it is used in the most effective state-of-the-art models, such as BERT and GPT.

The transformation of AM technology into general-purpose architecture can be confirmed by numerous publications describing its various applications in areas not related to word processing, e.g. for information system intrusion detection (Laghrissi, 2021), in visual monitoring systems (Fan, 2021), in pharmacy (Ye, 2021), in seismology, in behavior recognition systems, in medicine (Junlong, 2020), for passenger traffic prediction, etc. All those publications report that AM-based models offer better prediction accuracy as compared to other approaches, including DNN-based ones, not using AM.

The scientific novelty of this study lies in a developed method for predicting CETS parameters, which is based on the idea that data in engineering systems can be processed on the basis of revealing interrelated arrangement in their sequences, similar to natural language. To some extent, this assumption can be based on panpsychism views, the Penrose–Hameroff theory of conscience (Hameroff, 2019), and the extension of the principles of cognitive semantics on the description of processes in inanimate objects. To support the idea, (Ntakouris, 2021) can be mentioned, where the Transformer architecture is applied to timeseries instead of natural language.

2 MATERIALS AND METHODS

The aforementioned idea is supported by the fact that the individual units, namely the granulator, the CTMCM, and the OTF are arranged sequentially in the CETS. The granulator pelletizes the incoming crude ore; the resulting green pellets enter the calcining machine to be devoid of excess moisture, and the dried pellets are melted in the OTF, thus releasing gaseous phosphorus, which is the desired product of this processing procedure. This arrangement makes it so that the output parameters of one unit are the input parameters for the next one; therefore, in the sequences of parameter values there must be an interrelation similar to the predetermined arrangement of words in sentences. This interrelation is expected to be detectable by DNN with AM, which can provide higher accuracy in solving problems of classifying CETS states and regression.

The autoencoder architecture includes an encoder and a decoder (an encoder-decoder model), and the proposed method employs a sequence-to-sequence data model, which reflects the form of the CTMCM parameters as a multivariate time series (. In the proposed architecture, the autoencoder consists of two deep recurrent neural networks (Fig. 1):

– Encoder, which receives the input sequence of CETS parameters;

– Decoder, which generates the output sequence of CETS parameter forecasts and to which AM is added.

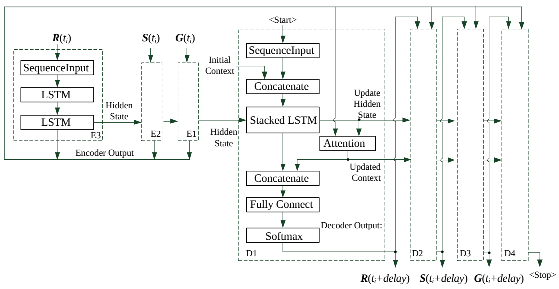

Figure 1: The encoder-decoder recurrent model employing AM.

Figure 1 has the following notations: Sequence Input – the layer receiving the input sequence and preparing it for feeding to the input into the LSTM, i.e. a recurrent layer of long short-term memory; Stacked LSTM – the stack of layers of recurrent DNN LSTM; Concatenate – the layer integrating several input vectors into one; Fully Connect – the fully connected layer; Softmax – the layer converting the input vector containing arbitrary values into the distribution of nonnegative weights whose sum is equal to 1. The input sequence is formed from the vectors $\boldsymbol{G}(t_i)$, $\boldsymbol{S}(t_i)$, and $\boldsymbol{R}(t_i)$ containing the results of measuring the CETS parameters at discrete times t_i. The vector $\boldsymbol{G}$ includes the following: g_1 – water mass flow rate; g_2 – crude ore mass flow rate at the entrance to the granulator; g_3 – the tilt angle of the granulator plates; g_4 – the power supply voltage of the plate drive motor. The vector $\boldsymbol{S}$ includes the following: s_1 – pellet mass flow rate at the entrance to the CTMCM; s_2 – the average temperature of the heat-carrier gas at the entrance to the vacuum chambers; s_3 – the average air temperature at the exit from the vacuum chambers; s_4 – the total airflow rate in the vacuum chambers. The vector $\boldsymbol{R}$ includes the following: r_1 – the average pellet diameter; r_2 – finished pellet mass flow rate; r_3 – phosphorus mass flow rate; r_4 – phosphorus purity degree (Dli, 2022). The composition of the vectors can be changed according to the necessary detailing of the processes in the CETS.

In the training set, the target values of the output sequence were time-shifted relative to the input sequence by the forecast range (*delay*), and this provided DNN training for generating predicted values $\boldsymbol{G}(t_i+delay)$, $\boldsymbol{S}(t_i+delay)$, and $\boldsymbol{R}(t_i+delay)$. Prediction accuracy increases due to the fact that the AM must provide a richer context from the encoder to the decoder, and the latter identifies what should be paid attention to when predicting each time step of the output sequence. The AM thereby eliminates the encoder-decoder architecture limitation for long sequences and, on the whole, increases the skills of the model in forecasting problems. The proposed method for predicting CETS parameters has been tested by means of the developed program in the Matlab 2021b environment. This environment has a large set of tools for constructing DNNs of various architectures, e.g. multilayer perceptrons, recurrent, convolutional networks, etc., and it offers a large number of implementations.

In order to implement the AM, a separate computational structure was developed, which was incorporated in the encoder-decoder software architecture, which is well known and has been used for a fairly long time (Nakayama, 2017) to solve various problems of data conversion. In the encoder-decoder architecture without the AM, the encoder memorizes the input sequence $\boldsymbol{U}=(\boldsymbol{G}(t_i), \boldsymbol{S}(t_i), \boldsymbol{R}(t_i))$ in one output latent state transmitted to the decoder, which can no longer see the whole input sequence. The application of AM enables the decoder to eliminate this limitation and, during the formation of the output sequence elements, to address to different encoder states based on the input sequence.

In terms of software implementation, AM is the function which, during the formation of an output sequence, takes in one of the latent states of the encoder and the decoder and yields an estimate to characterize how significant it is now for the decoder to look at this latent state of the encoder, so that it can correctly form the current element of the output sequence. Thus, the AM provides the neural network with a mechanism telling the decoder how significant it is for it to pay *attention* to a specific element of the input sequence.

The latent states of the decoder and the encoder are respectively denoted as $H_{\mathrm{d}} = (h_{\mathrm{d}0}, h_{\mathrm{d}1}, h_{\mathrm{d}2}, h_{\mathrm{d}3})$ and $H_{\mathrm{e}} = (h_{\mathrm{e}0}, h_{\mathrm{e}1}, h_{\mathrm{e}2}, h_{\mathrm{e}3})$. At the initial moment the decoder has a state $h_{\mathrm{d}0}$, and it must obtain the next output sequence element at the output (at the initial moment it is $\boldsymbol{G}(t_0+delay)$). The AM passes $h_{\mathrm{d}0}$ and $h_{\mathrm{e}0}$, $h_{\mathrm{e}1}$, $h_{\mathrm{e}2}$, $h_{\mathrm{e}3}$ through itself and yields the estimates $\boldsymbol{W} = (w_0, w_1, w_2, w_3)$ reflecting the *significance* of the input sequence elements (Fig. 2). After the estimates $\boldsymbol{W}$ are obtained, they are fed to the Softmax layer, from whose output we obtain normalized estimates $P = (p_0, p_1, p_2, p_3)$ adding up to 1.

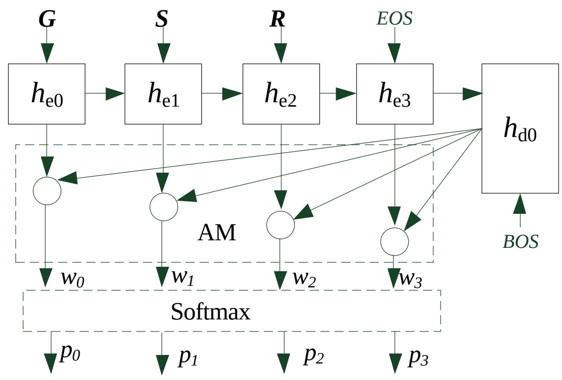

Figure 2: AM structure.

The context vector $C = H_{\mathrm{e}} \circ P$ is formed based on H_{e} and P, where $\circ$ denotes the Hadamard matrix product. The context vector is used to form the decoder output $D = f([y_1,C], h_{\mathrm{d}0},)$, where the brackets denote vector concatenation and y_1 is the decoder input (originally, it is BOS – begin of sequence), for it to generate the first element of the output sequence. As a result, the probability distribution at the decoder output takes into account not only $h_{\mathrm{d}0}$ and y_1, but also the context vector C. The latter provides the decoder with information that allows it to calculate the first output sequence element in the best possible way, the vector $\boldsymbol{G}(t_i+delay)$ in this case.

With the entrance of a new element of the sequence $\boldsymbol{U}$, a new context vector C is formed, the previous output D and the new context C are fed to the decoder input, etc. As a result, the decoder will gradually generate the whole sequence: $\boldsymbol{G}(t_i+delay)$, $\boldsymbol{S}(t_i+delay)$, and $\boldsymbol{R}(t_i+delay)$.

The function $Att(h_{\mathrm{ei}}, h_{\mathrm{dj}})$ implementing the AM can be different, e.g., it can be matrix multiplication, and for h_{d0} it will be represented by a row vector ($< h_{\mathrm{e0}}, h_{\mathrm{d0}} >$, $< h_{\mathrm{e1}}, h_{\mathrm{d0}} >$,$< h_{\mathrm{e2}}, h_{\mathrm{d0}} >$,$< h_{\mathrm{e3}}, h_{\mathrm{d0}} >$). In the proposed recurrent architecture, a fully connected single-layer neural network is used for $Att(h_{\mathrm{ei}}, h_{\mathrm{dj}})$, which was created with the use of fully Connected Layer from Matlab Deep Network Designer. This network takes in encoder and decoder state concatenations and gives normalized estimates: $P = Att(h_{\mathrm{ei}}, h_{\mathrm{dj}})$ = fully Connected Layer($h_{\mathrm{ei}}, h_{\mathrm{dj}}$). After the joint training of the encoder, the decoder, and the fully connected network, the elements P will represent the significance level of this or that input sequence element in the formation of the output sequence element.

The encoder, the decoder, and the AM in Matlab software are described by the following functions:

- model Encoder receives the input sequence $\boldsymbol{U}$ and returns the latent states;

- model Decoder receives the encoder output data, the context vector, and the initial latent state and returns the updated context vector, decoder output, the updated state, and AM weights;

- attention returns AM weight estimates at each time step.

To prevent the sharp increase or disappearance of activation outputs during DNN training, various layer weight initialization methods presented in Matlab were applied. The initialize Glorot method implementing Glorot initialization was applied for the AM.

3 INTERPRETATION AND DISCUSSION OF RESEARCH RESULTS

To generate training data, a model was created in Simulink, whose structure is shown in Fig. 3. Since the mathematical description of the CETS as a multidimensional controlled object is an aggregate of a large number of partial integro-differential equations, it is a separate complex problem to obtain the values of the parameters for them to be used in DNN training. Therefore, a simplified approach was used in the model experiment, considering dynamic models only in one channel reflecting the material balance of the process in the CETS in terms of incoming crude ore and the yield of the finished product (phosphorus).

Approximation of the acceleration characteristics of individual units was used for this channel, namely the granulator $s_1(g_2)$, the calcining machine $r_2(s_1)$, and the ore thermal furnace $r_3(r_2)$, in order to obtain their transfer functions, which were utilized to obtain training samples. DNN training in the model experiment with the developed software for predicting CETS parameters was based on a generated sample containing 30000 pairs $\{\boldsymbol{U}(t_i), \boldsymbol{U}(t_i+delay)\}$, 22000 of which were allocated to the training set, 4000 were assigned to the testing sample, and 4000 to the validation one.

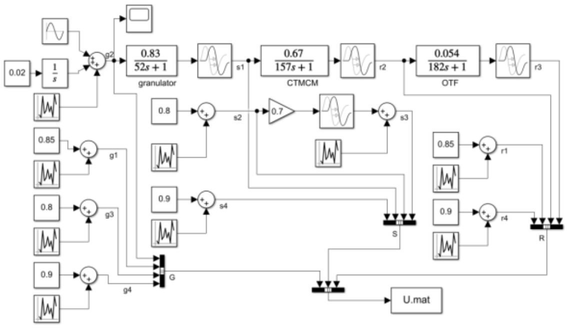

Figure 3: Simulink model for training data generation.

The transfer functions were obtained as inertia links with a delay of the form $W(p)=ke^{-\tau p}/(1+pT)$, where p is the Laplace variable and k, T, and τ are parameters determined by the acceleration characteristic of a corresponding object. Simple proportional models were used for the other parameters in the sequences $\boldsymbol{U}(t_i)$. Besides, the proportional model in the channel $s_3(s_2)$ took into account the heat-carrier transit time in the vacuum chambers, and this is represented by using the link of transportation delay by the time τ from the Simulink library, $s_3=k \cdot s_2 \cdot 1(t-\tau)$, where k is the object transmission factor.

The input parameter g_2 was represented as a linearly increasing signal with a harmonic component, and the parameters g_1, g_3, and g_4 were set constant. All these data were normalized to the range [0; 1]. To simulate the measurement noise, a uniformly distributed random signal from the range [−ξ; ξ] was additively impressed on all the parameters.

The DNN was trained within 80 epochs in a notebook with an AMD Ryzen 7 3750H processor (2.3 GHz) and an NVIDIA GeForce GTX 1650 video card. Figure 4 shows a change in the metric loss of the DNN in one of the experiments, which represents the behavior of this metric, typical of all the experiments.

The quality of the proposed DNN model was tested on the validation sample based on the calculation of the root-mean-square error (RMSE). Fig. 5 shows the *delay* dependences of the RMSE for three values of ξ.

The visual analysis of the dependences shows that the DNN with the AM provides a lower error than the DNN without the AM, this being particularly noticeable on large prediction intervals. This can be an experimental verification of the stated proposition that the attention mechanism in neural networks is able to reveal the interrelated organization in the sequences of process parameter values, which are presented as multivariate time series, and to provide higher prediction accuracy than a DNN without the AM.

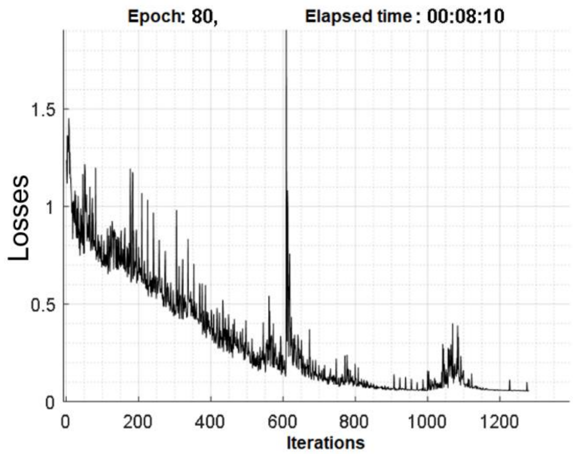

Figure 4: Loss metric variation during training.

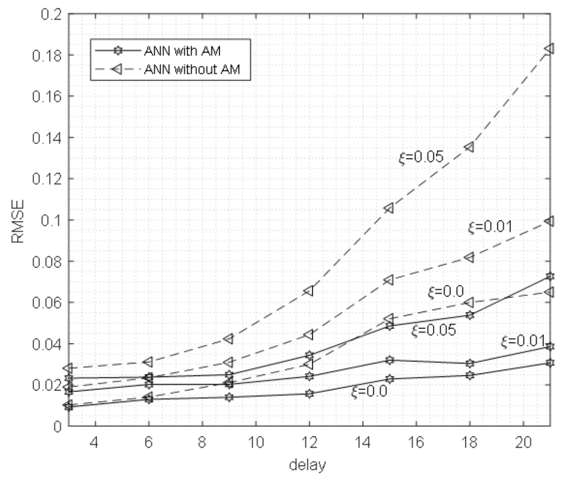

Figure 5: The delay and ξ dependences of the prediction error.

In the proposed version, the AM decoder takes into account only the latent states of the encoder; however, more complex implementations of this mechanism are also possible, e.g. Self-Attention and Multi-Head Attention, which also provide higher quality of time-series prediction as compared to other methods (Gao, 2022). Note in addition that various AM variants find application in analysis of images to provide more complete acquisition of a vast amount of semantic information contained in them, thus significantly increasing the effectiveness of recommendation systems (Zhong, 2020).

4 CONCLUSION

The study has yielded a method for predicting the process parameters of phosphorus production from apatite-nepheline ores, which implies the application of an autoencoder architecture using one of the variants of the attention mechanism. In the presented architecture, information on the intermediate internal states of the encoder is accessible to the decoder, and the latter uses it to form an output sequence of the predicted values of the process parameters.

The results of a model experiment in the Matlab environment have shown a higher accuracy of prediction based on the proposed method for various time horizons as compared to an autoencoder without an attention mechanism. The proposed method for predicting process parameters can find application in know are and software for recommendation systems and intellectual monitoring systems for various applications after taking into account their specificity, particularly, due to the adaptation of the dimensions of the sequences to be processed.

ACKNOWLEDGEMENTS

This study was performed under the state assignment, project No. FSWF-2023-0012 and grant of the Russian Science Foundation No. 22-11-00335.

REFERENCES

Puchkov, A., Lobaneva, E., Kultygin, O., 2022. Algorithm for predicting the parameters of a system for processing waste apatite-nepheline ores.*Journal of Applied Informatics*. vol. 17. no. 1 (97). pp. 55-68.

Meshalkin, V.P., Puchkov, A.Y., Dli, M.I., et al., 2019 Generalized Model for Engineering and Controlling a Complex Multistage Chemical Energotechnological System for Processing Apatite-Nepheline Ore Wastes. *Theor Found Chem Eng*, V.53, 463–471.

Solopov R.V., Samulchenkov A.S., Ziryukin V.I., 2021. Genetic algorithm as a tool for modeling calculations of electric power systems. *Journal of Applied Informatics*. vol. 16. no. 6 (96). pp. 43-53.

Sarker, I.H., 2021. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN COMPUT. SCI.* 2, 420.

Vaswani A., Shazeer N., Parmar N. et al., 2017. Attention Is All You Need. last revised 6 Dec.

Laghrissi, F., Douzi, S., Douzi, K., et al., 2021. IDS-attention: an efficient algorithm for intrusion detection systems using attention mechanism. *J Big Data* 8, 149.

Fan, S., Wang, R., Wu, Z., et al., 2021. High-speed tracking based on multi-CF filters and attention mechanism. *SIViP* 15, 663–671.

Ye, Y., Wang, J., Xu, Y., et al., 2021. MATHLA: a robust framework for HLA-peptide binding prediction integrating bidirectional LSTM and multiple head attention mechanism. *BMC Bioinformatics*, 22, 7.

Cheng, J., Tian, S., Yu, L., et al., 2020. Multi-Attention Mechanism Medical Image Segmentation Combined with Word Embedding Technology. *Aut. Control Comp. Sci.* 54, 560–571.

Hameroff, S., 2019. Consciousness and Quantum State Reduction—Which Comes First?. *Act Nerv Super* 61, 31–40.

Ntakouris, T., 2021. Timeseries classification with a Transformer model. *Keras documentation*.

Dli, M., Sinyavsky, Yu., Rysina E., Vasilkova M., 2022., A method for classifying mixing devices using deep neural networks with an expanded receptive field. *Journal of Applied Informatics*. vol. 17. no. 5 (101). pp. 51-61.

Nakayama, H., Nishida, N., 2017. Zero-resource machine translation by multimodal encoder–decoder network with multimedia pivot. *Machine Translation* 31, 49–64.

Gao, C., Zhang, N., Li, Y. et al., 2022. Self-attention-based time-variant neural networks for multi-step time series forecasting. *Neural Comput & Applic*.

Zhong, T., Zhang, S., Zhou, F. et al., 2020. Hybrid graph convolutional networks with multi-head attention for location recommendation. World Wide Web 23, 3125–3151.