HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

DEVELOPMENT OF AN INTELLIGENT CONTROL SYSTEM FOR THE ORE GRINDING PROCESS BASED ON AN ARTIFICIAL NEURAL NETWORK

Sharifov Husen Shukhrat ugli

Assistant, department of technological processes and production automation,

Bukhara state technical university

husensharipov1993@gmail.com

Erkinov Shaxboz Namoz o'gli

Assistant, Department of Technological Processes and Production Automation,

Bukhara State Technical University

Doi: https://doi.org/10.5281/zenodo.24039744

Abstract: This article addresses the design of an intelligent controller for ore grinding that is built around an artificial neural network. Two indicators are used to describe the state of a ball mill: the deviation of the mill load from its set value and the deviation of the product particle size from its target. On the basis of these two indicators, a five-layer feedforward network is proposed, with ReLU activation in the hidden layers and Tanh in the output layer. The weighted sums of the neurons, the matrices of synaptic weights and the bias terms are written out in mathematical form. For training, the backpropagation algorithm is combined with the Adam optimizer. The network output yields the control law $U(t)$, which adjusts the ore feed rate.

Keywords: ore grinding, artificial neural network, intelligent control, five-layer network, activation function, synaptic weight, backpropagation, Adam algorithm, ReLU, Tanh, control law.

Introduction

In the mining and metallurgical industry, grinding is one of the stages that most strongly determines the quality of the final product and the efficiency of the whole production chain. Several factors act on the process at once: the physical and mechanical properties of the ore and its hardness, the degree to which the mill is filled, water consumption, rotation speed, and the rheological properties of the pulp. In practice, such processes are usually regulated by PID controllers. The difficulty is that the composition and hardness of the ore vary, the mill load can build up, and the ratio of liquid to solid phases can drift. Each of these changes the dynamic behavior of the plant, so a conventional controller that was tuned once may lose its effectiveness later on.

Muxitdinov and Boybutayev (2023) studied the control of ball mill grinding on the basis of a predictive mathematical model. In their work the mill load, the product particle size, the pulp volume and the control variables were treated as interrelated quantities, and a control structure built on

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

dynamic inversion and a predictive model was proposed. Following this line of research, the present article develops a method for controlling the grinding process with an artificial neural network. A key advantage of neural networks is that they can learn complex nonlinear relationships among process parameters directly from experimental data and then use what they have learned to generate a control action.

The aim of this study is to create a five-layer neural network architecture that accounts for changes in mill load and product particle size, and to use it to form the control law $U(t)$.

Selection of Input and Output Parameters

Ore grinding is a multivariable, nonlinear dynamic plant that operates under constant external disturbances. For the network inputs, two technological indicators that describe the condition of the mill most fully were chosen.

Input vector:

$$X(t)=\begin{bmatrix}x_1(t)\\x_2(t)\end{bmatrix}$$

where:

$x_1(t)$ is the normalized deviation of the mill load;

$x_2(t)$ is the normalized deviation of the product particle size.

Deviation of the mill load:

$$e_1(t)=L(t)-L^* \qquad (1)$$

Deviation of the product particle size:

$$e_2(t)=d_{50}(t)-d_{50}^* \qquad (2)$$

where:

$L(t)$ is the current mill load;

L^* is the set value of the load;

$d_{50}(t)$ is the characteristic particle size of the finished product;

d_{50}^* is the set value of the particle size.

To keep training stable, the input quantities are normalized:

$$x_1(t)=\frac{e_1(t)}{\Delta L_{\max}} \qquad (3)$$

$$x_2(t)=\frac{e_2(t)}{\Delta d_{\max}} \qquad (4)$$

After normalization, any value that falls outside the allowed range is clipped to ±1:

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

$$-1 \le x_1(t), x_2(t) \le 1 \qquad (5)$$

The network output is

$$Y(t) = U(t) \qquad (6)$$

where $U(t)$ is the control signal that adjusts the ore feed rate.

The physical meaning of the control signal is as follows:

$U(t)<0$ means the ore feed rate is reduced;

$U(t)=0$ means the ore feed rate is left unchanged;

$U(t)>0$ means the ore feed rate is increased.

Network Architecture

A five-layer feedforward neural network is proposed for the intelligent controller. Its layers are:

1. Input layer: 2 neurons.
2. First hidden layer: 8 neurons.
3. Second hidden layer: 8 neurons.
4. Third hidden layer: 4 neurons.
5. Output layer: 1 neuron.

Table 1. Architecture of the neural network

Layer	Number of neurons	Activation function	Role
Layer 1	2	Linear	Receiving the input parameters
Layer 2	8	ReLU	Extracting primary features
Layer 3	8	ReLU	Learning nonlinear relationships
Layer 4	4	ReLU	Forming the key features
Layer 5	1	Tanh	Producing the control signal

The general mathematical model of the network is

$$A^{(l)} = f^{(l)}\left(W^{(l)} A^{(l-1)} + b^{(l)}\right) \qquad (7)$$

where:

$A^{(l)}$ is the output vector of layer l;

$W^{(l)}$ is the matrix of synaptic weights between adjacent layers;

$b^{(l)}$ is the vector of bias coefficients;

$f^{(l)}$ is the activation function;

l is the layer index.

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

Each neuron multiplies its incoming signals by the corresponding synaptic weights and adds the bias to the result, which gives a weighted sum:

$$S_j^{(l)} = \sum_{i=1}^{n_l} W_{ji}^{(l)} A_i^{(l-1)} + b_j^{(l)} \qquad (8)$$

where:

- $S_j^{(l)}$ is the total input signal of neuron j;
- $W_{ji}^{(l)}$ is the synaptic weight on the connection from input i to neuron j;
- $A_i^{(l-1)}$ is the output signal of a neuron in the previous layer;
- $b_j^{(l)}$ is the bias value;
- n_l is the number of neurons in the previous layer.

The output signal of the neuron is

$$A_j^{(l)} = f\left(S_j^{(l)}\right) \qquad (9)$$

The hidden layers use ReLU as the activation function:

$$f(S) = \mathrm{ReLU}(S) = \max(0, S) \qquad (10)$$

For negative arguments this function returns zero, and for positive arguments it passes the input through unchanged. Its main advantage is how cheap it is to compute.

The derivative of ReLU is

$$f'(S) = \begin{cases} 0, & S \le 0 \\ 1, & S > 0 \end{cases} \qquad (11)$$

The output layer, by contrast, uses the hyperbolic tangent:

$$f(S) = \tanh(S) = \frac{e^{S} - e^{-S}}{e^{S} + e^{-S}} \qquad (12)$$

Tanh keeps the output within the interval $[-1;1]$, which guarantees that the control signal stays bounded.

Layer-by-Layer Computation

First layer: the input layer

$$A^{(1)} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Second layer: 8 neurons

$$S^{(2)} = W^{(2)} A^{(1)} + b^{(2)}$$

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

$$A^{(2)}=\mathrm{ReLU}\left(S^{(2)}\right) \tag{13}$$

The weight matrix of this layer is

$$W^{(2)}\in R^{8\times 2}$$

Third layer: 8 neurons

$$S^{(3)}=W^{(3)}A^{(2)}+b^{(3)}$$

$$A^{(3)}=\mathrm{ReLU}\left(S^{(3)}\right) \tag{14}$$

Its weight matrix is

$$W^{(3)}\in R^{8\times 8}$$

Fourth layer: 4 neurons

$$S^{(4)}=W^{(4)}A^{(3)}+b^{(4)}$$

$$A^{(4)}=\mathrm{ReLU}\left(S^{(4)}\right) \tag{15}$$

Its weight matrix is

$$W^{(4)}\in R^{4\times 8}$$

Fifth layer: the output neuron

$$S^{(5)}=W^{(5)}A^{(4)}+b^{(5)}$$

$$\boxed{U(t)=\tanh\left(S^{(5)}\right)} \tag{16}$$

Its weight matrix is

$$W^{(5)}\in R^{1\times 4}$$

In total, the network therefore has 137 trainable parameters: 116 synaptic weights and 21 bias coefficients.

Initialization of Weights

The quality of the network depends to a large extent on how its synaptic weights are chosen and trained. The weight matrices have the following dimensions:

$$W^{(2)}=\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ \vdots & \vdots \\ w_{81} & w_{82} \end{bmatrix} \tag{17}$$

$$W^{(3)}\in R^{8\times 8}$$

$$W^{(4)}\in R^{4\times 8}$$

$$W^{(5)}=\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \end{bmatrix} \tag{18}$$

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

For the initial weights, the He method (He et al., 2015) is recommended, since it suits hidden layers with ReLU activation:

$$W_{ij} \sim N\left(0, \frac{2}{n_{in}}\right) \tag{19}$$

where n_{in} is the number of neurons feeding into the layer.

For the initial setting of the output-layer weights, the Xavier (Glorot) method can be used (Glorot & Bengio, 2010).

Training of the Network

During training, the weights are updated automatically in response to the error. The network is trained by supervised learning on a data set of the form

$$D=\{\left(X_k, U_k^*\right)\}_{k=1}^{N} \tag{20}$$

where:

$X_k=[x_{1k}, x_{2k}]^T$ are the input parameters;

U_k^* is the reference control signal;

N is the number of training samples.

The initial training set can be generated from the Fuzzy Logic model that the author developed earlier: a base of 25 IF–THEN rules together with the Mamdani algorithm computes a reference control signal for every pair of inputs. Later, the set can be extended with real production data and the network retrained.

The error is the difference between the reference value and the network output:

$$e_k=U_k^* - U_k \tag{21}$$

The objective function for training is the mean squared error (MSE):

$$\boxed{\text{MSE}=\frac{1}{N}\sum_{k=1}^{N}\left(U_k^* - U_k\right)^2} \tag{22}$$

A falling MSE indicates that the network output is approaching the reference control signal.

Weights and biases are updated by propagating the error backward through the network (Rumelhart et al., 1986). The gradient descent rule is

$$W_{new}=W_{old}-\eta\frac{\partial\,\text{MSE}}{\partial W} \tag{23}$$

$$b_{new}=b_{old}-\eta\frac{\partial\,\text{MSE}}{\partial b} \tag{24}$$

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

where η is the learning rate.

The propagation of the error into the hidden layers is computed with the chain rule.

At the output layer:

$$\delta^{(5)}=\left(A^{(5)}-U^{*}\right)f'\left(S^{(5)}\right) \tag{25}$$

At the hidden layers:

$$\delta^{(l)}=\left(W^{(l+1)T}\delta^{(l+1)}\right)\odot f'\left(S^{(l)}\right) \tag{26}$$

where $\odot$ denotes element-wise multiplication.

The Adam Optimization Algorithm

Training with the Adam algorithm is proposed (Kingma & Ba, 2015). Adam takes the first and second moments of the gradient into account and adapts the step size for each parameter individually.

$$m_t=\beta_1 m_{t-1}+(1-\beta_1)g_t \tag{27}$$

$$v_t=\beta_2 v_{t-1}+(1-\beta_2)g_t^2 \tag{28}$$

The moment estimates are then corrected for bias:

$$\hat{m}_t=\frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t=\frac{v_t}{1-\beta_2^t} \tag{29}$$

The weights are updated as

$$\boxed{W_t=W_{t-1}-\eta\frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\varepsilon}} \tag{30}$$

where:

- g_t is the current gradient;
- m_t is the first moment of the gradient;
- v_t is the second moment of the gradient;
- β_1,β_2 are the decay coefficients used to compute the moments;
- ε is a small constant that prevents division by zero.

In this way, every weight is updated with an adaptive step that reflects its own gradient history.

The training procedure consists of the following steps:

1. Collect measurement data on mill load and particle size.
2. Preprocess and normalize the measurements.
3. Generate the reference control signals U^{*} that correspond to the inputs.

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

4. Split the data into training, validation and test sets.
5. Choose the initial weights of the network.
6. Run the forward pass to obtain $U(t)$.
7. Compute the MSE.
8. Determine the gradients by backpropagation.
9. Update the weights with the Adam algorithm.
10. Stop training according to the validation error.
11. Evaluate the model on the test set.

As a starting point, the data can be divided into 70% for training, 15% for validation and 15% for testing. Because the data form a time series, it is better to split them in chronological order than to shuffle them at random.

Control Law and Operating Sequence

Once trained, the network output is used to adjust the ore feed rate. The overall control law of the network is

$$\boxed{U(t)=\tanh\left[W^{(5)}\mathrm{ReLU}\left(W^{(4)}\mathrm{ReLU}\left(W^{(3)}\mathrm{ReLU}\left(W^{(2)}X(t)+b^{(2)}\right)+b^{(3)}\right)+b^{(4)}\right)+b^{(5)}\right]} \quad (31)$$

This expression captures the entire computation of the five-layer network, from input to output, in a single equation.

The normalized control signal is mapped to the real technological range as

$$\Delta F_r(t)=U(t)\cdot \Delta F_{\max} \quad (32)$$

The new set point for the ore feed is

$$\boxed{F_r^*(t)=\mathrm{sat}\left[F_r^*(t-1)+\Delta F_r(t)\right]} \quad (33)$$

where:

$F_r^*(t)$ is the new set point of the ore feed;

$\Delta F_r(t)$ is the correction applied to the ore feed rate;

$\Delta F_{\max}$ is the largest change of the control action in one step;

sat is the limiting operator that enforces the technological constraints.

In other words, the network analyzes the deviations of mill load and product particle size and, from them, produces a control signal that changes the ore feed rate.

The system operates in the following sequence:

1. Mill load and product particle size are measured by the appropriate sensors.
2. The measured values are sent to the control system.
3. The deviations between the current and set values are calculated.

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

4. The deviations are normalized.
5. The normalized signals are fed to the input layer of the network.
6. In the second, third and fourth layers, ReLU performs the nonlinear transformation.
7. In the fifth layer, Tanh produces the control signal $U(t)$.
8. The signal is limited by the technological constraints.
9. Through a frequency converter, the signal is passed to the motor of the ore feed conveyor.
10. The new state of the mill is measured again, and the cycle repeats.

Safety limits must be enforced independently of the neural network, at a separate protection and control level.

Conclusion

This article has developed a mathematical model for the intelligent control of ore grinding on the basis of an artificial neural network. The deviations of mill load and product particle size were chosen as the inputs of the control system, and the control signal $U(t)$, which adjusts the ore feed rate, was defined as the output. A five-layer feedforward network with a 2–8–8–4–1 architecture was proposed, using ReLU in the hidden layers and Tanh in the output layer. The summation functions, weight matrices and bias coefficients of the network were expressed mathematically. The architecture contains 116 synaptic weights and 21 biases, that is, 137 trainable parameters in total. Backpropagation and the Adam algorithm were recommended for training, and it was shown that the first reference values can be taken from the control signals of the previously created Fuzzy Logic model. The proposed model provides a mathematical basis for assessing the state of the mill and for intelligent regulation of the ore feed rate. To confirm its practical effectiveness, the next stage should include simulation in MATLAB or Python and experiments with production data.

References

1. Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 9, 249–256. **https://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf**
2. Haykin, S. (2009). Neural networks and learning machines (3rd ed.). Pearson.
3. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 1026–1034).
4. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR).
5. Muxitdinov, D. P., & Boybutayev, S. B. (2023). Управление процессом измельчения на основе прогностической модели шаровой мельницы [Control of the grinding process based on a predictive model of a ball mill]. O'zbekiston konchilik xabarnomasi, (2), 93.

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

6. Pomerleau, A., Hodouin, D., Desbiens, A., & Gagnon, E. (2000). A survey of grinding circuit control methods: From decentralized PID controllers to multivariable predictive controllers. Powder Technology, 108(2–3), 103–115.

7. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323, 533–536. **https://doi.org/10.1038/323533a0**

8. Kalandarov, P. I., Hayitov, A. N., & Sharifov, H. Sh. (2023). Matematik modellashtirish va donni maydalash jarayonini tizimli tahlili. Xorazm Ma'mun akademiyasi axborotnomasi, 8(2), 199–205.
9. Kalandarov, P. I., Sharifov, H. Sh., & Hayitov, A. N. (2023). Avtomatlashtirish obyekti sifatida donni qayta ishlash mashinalarining diagnostikasi. Irrigatsiya va melioratsiya, 3(33), 47–52.
10. Kalandarov, P. I., Hayitov, A., Ubaydullayeva, S. R., Nikolov, N. N., & Sharifov, H. (2023). Technological machines diagnostics of grain processing enterprises as the object of automation. 2023 International Conference Automatics and Informatics (ICAI), 252–256.
11. Kalandarov, P. I., Sharifov, H. Sh., & Hayitov, A. N. (2023). Буғдой донини майдалаш ускуналарининг вибродиагностика тизими, қурилмалари ва дастури [Vibrodiagnostic system, devices, and program of wheat grain crushing equipment]. International Scientific Journal Science and Innovation, Special Issue, 35–39.
12. Hayitov, A. N., Sharifov, X. Sh., & Kalandarov, P. I. (2023). Омухта ем ишлаб чиқариш жараёнини автоматлаштириш. Educational Research in Universal Sciences, 2(Special Issue 9), 377–381.
13. Kalandarov, P. I., Khayitov, A. N., & Sharifov, H. S. (2024). The electrophysical method of grain grinding is based on mathematical models. In A. A. Radionov & V. R. Gasiyarov (Eds.), Proceedings of the 10th International Conference on Industrial Engineering: ICIE 2024 (pp. 490–501). Springer.
14. Kalandarov, P. I., Sharifov, H. S., & Khayitov, A. N. (2024). Diagnostics of technological machines of grain processing enterprises. In A. A. Radionov & V. R. Gasiyarov (Eds.), Proceedings of the 10th International Conference on Industrial Engineering: ICIE 2024 (pp. 521–531). Springer.
15. Kalandarov, P. I., Abdullaev, Kh. Kh., Khaitov, A. N., Sharifov, Kh. S., & Murodova, G. F. (2024). Modeling of biotechnological objects. BIO Web of Conferences, 108, 25005.
16. Хайитов, А. Н., & Шарифов, Х. Ш. (2024). Математическое моделирование процесса измельчения зерна. In Актуальные проблемы современной науки, техники и образования: Тезисы докладов 82-й международной научно-технической конференции (Т. 2, с. 104). Магнитогорский государственный технический университет им. Г. И. Носова.