

UDC 621.91.01:004.032.26

O. Sokolova., Student, S. Vysloukh, PhD, Associate Professor
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

MODELING OF TITANIUM ALLOYS PROCESSING PARAMETERS

Annotation. The article discusses the modeling of the efficient cutting tool life in the processing of titanium alloy. To realize this aim, an effective method for processing research results has been selected. The features and algorithm of using the TensorFlow platform for the design, creation and study of models in the field of machine learning are presented.

Key words: machine learning, modeling, neural networks, regression analysis, platform TensorFlow, turning, titanium alloy, efficient cutting tool life.

INTRODUCTION

Titanium alloys are important structural materials that are characterized by high hardness, corrosion resistance and low specific gravity. This determines their using for the manufacture of critical parts in instrument and mechanical engineering. In [1] the results of predicting the efficient cutting tool life are presented, which were obtained by means of using the tools of the program Matlab, namely, the NeuralNetwork Toolbox module. In doing so the already existing neural network was used, the application of which resulted to large deviations of the predicted results from the real ones. This necessitates the search for other ways to solve the problem of modeling technological processes.

CHOICE OF RESEARCH METHOD

An analysis of instrumental methods for solving modeling problems allowed us to pay attention to the Tensorflow platform, an open source software library for numerical computation using data-flow graphs. It was originally developed by the Google Brain Team within Google's Machine Intelligence research organization for machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains as well [2].

TensorFlow is intended for the design, creation and study models of deep learning – an area of machine learning, the algorithms of which are determined by the structure and brain works. TensorFlow can be used to perform numerical calculations. These computations are performed using data-flow graphs. In these graphs, vertices are mathematical operations, and edges are data, which are usually represented as multidimensional arrays or tensors that are connected between these edges [3].

Designing neural networks on TensorFlow becomes easier if using the Keras add-on, which is convenient and easy to prototype. An API add-on is recommended for those who prefer object-oriented programming and layered modeling of neural networks. In a few lines of code, you can create a direct distribution neural network using standard methods such as regularization [4].

RESEARCH RESULTS

The objective of these studies is to predict the efficient cutting tool life equipped with BK4 carbide inserts during turning of BT6 titanium alloy.

The building of model was using TensorFlow, which is an integrated open source platform for machine learning. In this case, the study of some properties of the data set is implemented and the subsequent testing of these properties with respect to another data set. A common practice in machine learning is to evaluate an algorithm by splitting

a data set into two parts. One data selection is a training selection, on which the neural network studies some properties, and the other is a test selection, on which the tested properties are tested.

When conducting the process research, the cutting speed V ranged from 70 to 200 m/min., the feed S – from 1 to 4 mm/min., and the cutting depth t – from 0.08 to 0.4 mm. As a result of the research the efficient cutting tool life T varied from 1.8 min up to 17.1 min.

To process the experimental data, their pre-testing was performed. Since the speed values are very different from the values of other parameters, for the convenience and improvement of training the neural network, their values are divided by 100. Next, the following necessary modules were imported:

```
#import TensorFlow
import tensorflow as tf

#import the module for indicators used to measure
from sklearn import metrics

#import a model Ridge
from sklearn.linear_model import Ridge

#import model RidgeCV
from sklearn.linear_model import RidgeCV

from sklearn.preprocessing import PolynomialFeatures

#import ordinary least squares LinearRegression model
from sklearn.linear_model import LinearRegression
```

Model `Ridge()` solves a regression model where the loss function is the linear least squares function and regularization is given by the l_2 -norm. Also known as Ridge Regression or Tikhonov regularization.

Model `RidgeCV()` is a model Ridge regression with built-in cross-validation. By default, it performs Generalized Cross-Validation, which is a form of efficient Leave-One-Out cross-validation.

Function `PolynomialFeatures()` is a function of generating polynomial and interaction features with degree less than or equal to the specified degree.

`LinearRegression()` fits a linear model with coefficients $w = (w_1, \dots, w_p)$ to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. [5].

At the next step in processing the research results, the model was trained using the built-in `fit()` function. To assess the quality of the selected dependencies (models), the determination coefficient is determined, which shows the ratio of the distribution of points explained by this dependence and, thus, serves as the main quality criterion for selecting the regression line. Determination coefficient provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model [6]. It characterizes the degree of similarity between the source data and the predicted ones.

Next, the data was tested using the predict() function and an array of data for testing and displaying them on the screen. For linear regression, it looks like this:

```
#make predictions on the testing set
out_pred = linreg.predict(in_test)
print(out_pred)
```

Variable in_test is variable that contains input data for testing models. Variable out_pred is variable that contains output test data predicted by LinearRegression() model.

Based on the test results, the prediction accuracy was estimated, which showed that the best model for solving this problem is LinearRegression().

Tables 1 and 2 present the target results on which models were trained and tested, and also the data predicted by LinearRegression().

Table 1. Training data

<i>V, m/min</i>	<i>S, mm/min</i>	<i>t, mm</i>	<i>T (target), min</i>	<i>T (predicted), min</i>
100	0.4	0.1	1.8	1.89
70	0.4	0.1	7.5	7.13
120	0.3	1.0	2.7	2.88
80	0.3	1.0	8.1	9.86
160	0.2	1.0	3.0	2.08
140	0.2	1.0	4.5	5.56
100	0.2	1.0	10.8	12.54
90	0.2	1.0	16.2	14.28
200	0.08	1.0	2.9	2.505
120	0.08	1.0	14.9	16.46
120	0.2	4.0	2.5	3.39
100	0.2	4.0	7.8	6.88
150	0.2	2.0	2.3	1.94
120	0.2	2.0	7.0	7.16
160	0.2	1.0	3.0	2.07
140	0.2	1.0	4.4	5.56
90	0.2	1.0	17.1	14.283

Table 2. Testing data

<i>V, m/min</i>	<i>S, mm/min</i>	<i>t, mm</i>	<i>T (target), min</i>	<i>T (predicted), min</i>
90	0,4	0,1	4,2	3.64
100	0,3	1,0	5,4	6.37
100	0,2	2,0	11,0	10.66

The print on the screen of the coefficients of the obtained linear regression model is implemented using two lines of code:

```
print('Intercept: ', linreg.intercept_)
print('Coefficient: ', linreg.coef_)
```

Attribute intercept_ is independent term in the linear model. Attribute coef_ is coefficients for the linear regression problem.

As a result of the researches and processing of their results, the model is obtained that is represented by the regression equation:

$$T = 44.2 - 17.438 * V * 10^3 - 61.692 * S - 1.885 * t \quad (1)$$

To assess the regression model, three common indicators were determined – the mean absolute error of the MAE, the mean-square error of the MSE, and the root-mean-square error of the RMSE.

All of this is implemented by using the built-in functions:

```
# calculate MAE using scikit-learn
print("MAE =", metrics.mean_absolute_error(out_test, out_pred))
# calculate MSE using scikit-learn
print("MSE =", metrics.mean_squared_error(out_test, out_pred))
# calculate RMSE using scikit-learn
print("RMSE =", np.sqrt(metrics.mean_squared_error(out_test, out_pred)))
```

Variable `out_test` is variable that contains target (output) test data.

The verification of the adequacy of the obtained regression equation was performed according to the Fisher test, which showed its adequacy with a confidence probability of 0.95.

CONCLUSIONS

The solution to the problem of modeling the efficient cutting tool life made of BK4 carbide during turning of BT6 titanium alloy showed that the TensorFlow platform is an effective tool for designing, creating and studying models in the field of machine learning.

LIST OF USED REFERENCES

- [1] Соколова О. А. Моделирование периода стойкости твердосплавных резцов при обработке титановых сплавов. / О.А. Соколова, С.П. Вислоух // *Збірник наукових праць X Всеукр. науково-технічної конференції з міжнародною участю “Процеси механічної обробки, верстати та інструмент”*, 6-9 листопада 2019 р. – Житомир: «Житомирська політехніка». – 2019. – С. 181–182.
- [2] What is the TensorFlow machine intelligence platform? Режим доступа: <https://opensource.com/article/17/11/intro-tensorflow> – 09.11.2017 г.
- [3] TensorFlow туториал. Часть 1: тензоры и векторы. Режим доступа: www.URL:https://neurohive.io/ru/tutorial/tensorflow-tutorial-tenzory-i-vektory/ – 20.09.2018 г.
- [4] 9 вещей, которые нужно знать о TensorFlow. Режим доступа: www.URL:https://dev.by/news/9-veshei-kotorye-nuzhno-znat-o-tensorflow – 18.12.2018 г.
- [5] `sklearn.linear_model.LinearRegression`. Режим доступа: www.URL:https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- [6] Coefficient of determination Режим доступа: www.URL:https://en.wikipedia.org/wiki/Coefficient_of_determination – 14.04.2020 г.

Academic adviser – PhD, associate professor S. Vysloukh