UDC 004.8

Andrii Kobyliuk, Artem Volokyta.

ARTIFICIAL INTELLIGENCE-BASED SCHEDULING

The paper considers the algorithm for scheduling jobs in a computer system using artificial intelligence. The two-tier convolutional neural network chose the best heuristic algorithm. For training the NN is used genetic algorithm. The model is trained on the sets of random generated jobs.

Key words: artificial intelligence, neural network, scheduling, task, genetic algorithm.

Fig.: 1. Tabl.: 1, Tabl.: 2.

Target setting Lack of an optimal scheduling algorithm that would work equally effectively for different data sets.

Actual scientific researches and issues analysis. Nowadays the field of artificial intelligence is developing rapidly, each year there are a dozen articles and methodologies to speed up or replace existing algorithms. Scientists D. Gupta and E. Tunk have already written about the use of neural networks to solve the problem of planning in their article [1].

Uninvestigated parts of general matters defining. Despite a huge number of works devoted to applying of neural networks for the scheduling, the problem of applying a new algorithms for this purpose remains little investigated. The task of optimization of the scheduling problems usually deals with NP-class of algorithms, so a deterministic algorithm can`t be created to resolve these problems.

The research objective. The purpose of this paper is to research the application of the artificial intelligence in combination with job scheduling using A two-tier convolutional neural network. There are different types of schedulers (long-term | short-term | medium-term), this work is focus on long-term scheduling because at this level users can abstract from the details inside the hardware and just focus on how jobs interact with each other. There is also a need to be independent from job`s size and to build neural network with such requirements. As a solution, the article will focus on creating a model that generates schedule on a given topic using the above-mentioned structures and shows that the suggested way is quite productive and effective.

The statement of basic materials.

$N = \{1 \dots n\}$ of n jobs is to be to be computed in m parallel machines. If job j is completed before its due date, it is called *early*, otherwise it is considered *tardy*. Let's assume that totally have r heuristic algorithms, presented by a series R_i. $n_t(j)$ is the whole amount of tardy jobs of the scheduling from using algorithm $j \in R$. Define the following:

$$\alpha \ = \max\nolimits_{j \,\in\, R}\{n_t(j)\} \tag{1}$$

and for each $j \ \in R_i$, define

$$\beta_j = n_t(j)/\alpha \tag{2}$$

the minimum value of j calculated by using (1) and (2), the minimum total number of tardy jobs of a schedule obtained by heuristic j. Actual values of $\beta = (\beta_1, \dots \beta_r)$ can be used to deduce which algorithm results in the smallest number of tardy jobs and accordingly is the most appropriate heuristic to resolve for a given N.

General model structure. The input layer of the neural network, included the number of jobs n and the output layer included the amount of machines m (see Fig. 1). The NN configuration uses three layers. The number of nodes in the hidden layer is made via experimentation.

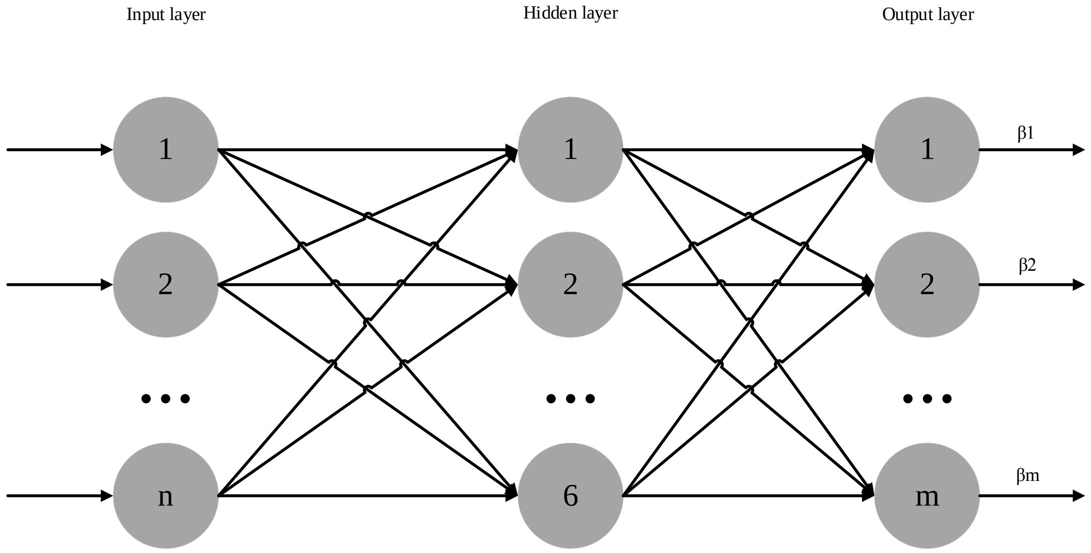

Fig. 1. One-tier NN

Even that results are not bad, but the one-tier neural network may not work very well with respect to predicting the best algorithm. For improving the NN's performance was designed the two-tier neural network. This is a three-layer neural network, similar to the previous one, with m input nodes, six hidden nodes and m output nodes (see Fig. 2).

Experiments. Two groups of test data were generated. The processing on the first stage generated is the test set which randomly generate integers values from the next uniform distributions:

- First data set $(1, 100)$ – non-dominant;
- Second data set $(1, 100/m)$ – dominant.

The amount of jobs, n in range from twenty to one hundred with step of twenty jobs.

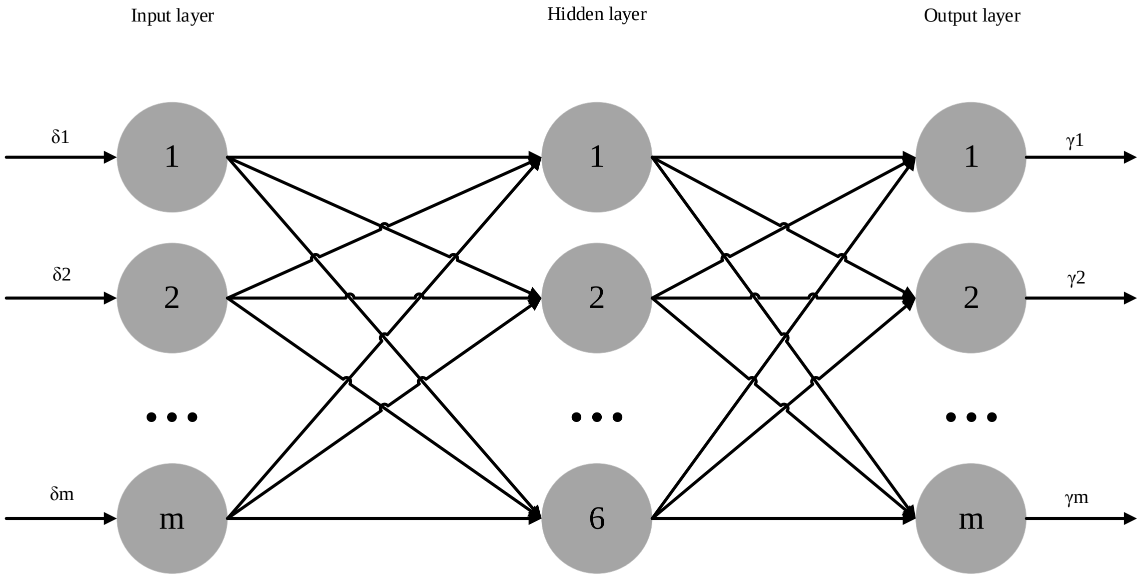

Fig. 2. Two-tier NN

The amount of symmetric parallel machines was m = 2; 3, or 5 at the second stage.

Table 1

Accurate prediction by the one-tier, %

Generic Data Set	**No. of Job**	**In-sample**			**Out-sample**		
		2 m	**3 m**	**5 m**	**2 m**	**3 m**	**5m**
First set of data	20 n	81	93	95	75	93	97
	40 n	95	99	99	97	95	99
	60 n	87	93	93	95	93	99
	80 n	93	87	95	97	95	93
	100 n	93	89	79	95	91	91
		Average = 91.40			Average = 94.67		
Second set of data	20 n	65	69	83	71	59	65
	40 n	79	71	77	81	69	79
	60 n	93	97	91	99	97	91
	80 n	95	95	99	99	99	95
	100 n	97	99	99	93	97	95
		Average = 87.27			Average = 81.93		
Overall		Average = 89.33			Average = 89.80		

These experiments with the one-tier NN show (see Tabl.1) that the prediction of the best heuristic algorithm which used to resolve the out sample accurately in approximately 89.9% of the cases. Also, the possibility to predict the best heuristic algorithm to be used in the second data set decreased to approximately 81.93% of the cases.

Table 2

Accurate prediction by the two-tier network , %

Generic Data Set	No. of Job	In-sample			Out-sample		
		2 m	**3 m**	**5 m**	**2 m**	**3 m**	**5m**
First set of data	20 n	82	94	96	96	98	96
	40 n	96	100	100	100	100	100
	60 n	96	96	98	100	98	100
	80 n	98	98	100	100	96	100
	100 n	100	98	100	100	100	100
		Average = 96.80			Average = 98.93		
Second set of data	20 n	92	72	62	72	70	66
	40 n	90	98	92	92	86	76
	60 n	94	98	96	100	98	96
	80 n	96	96	100	100	100	96
	100 n	98	100	100	94	98	94
		Average = 92.27			Average = 89.20		
Overall		Average = 94.53			Average = 93.40		

On these experiments with the two-tier NN show (see Tabl.2) was seen the improvements comparing to the previous experiment. The best algorithm which used to resolve the out sample choose in up to 93.40% from 89.80% of the cases . For the second set of data, the two-tier NNs correctly predicted the algorithm in 89.2% of the cases, as compared to 81.93% for the one-tier NNs.

Conclusions. This paper has considered the problem of using artificial intelligence to resolving a scheduling problem. It can be effectively solved using a neural-network by simply predictions of the best heuristic algorithm. The designed neural-network system may correctly predict more than 93.4% of the cases. In the future there are several ways to deeper research in this area. Firstly, discovering and set up such configuration which will allow to cover more input data with better approximation. Finally, development more complicated and multi-tier NNs will expand their application domain to solve other problems of computer science and AI.

References

1. J. N. D., Gupta, E. A., Tunc (1998). Minimizing Tardy Jobs in a Two-stage Hybrid Flowshop, International Journal of Production Research 36 (pp. 2397 – 2417).

2. Błażewicz, Jacek, Ecker, K. H., Pesch, E., Schmidt, G., Weglarz, J. (2001). Scheduling computer and manufacturing processes (2 ed.).

3. L. I., Burke, J. P., Ignizio (1992). Neural Networks and Operations Research: An Overview, Computers and Operations Research 19 (pp. 179 -189).

4. Schmidhuber, Jürgen (2015). "Deep Learning". Scholarpedia (pp. 85–117).

5. Jatinder, N. D., Gupta, Randall, S., Sexton, Enar, A., Tunc (2000, January). Selecting Scheduling Heuristics Using Neural Networks (pp. 2-22).

6. D., Tuzun, M. A., Magent, L. I., Burke (1997). Selection of Vehicle Routing Heuristics Using Heuristic Networks, International Transactions on Operations Research 4 (pp. 211-221).

AUTHORS

Andrii Kobyliuk – student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

Artem Volokyta (supervisor) – associate professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: artem.volokita@kpi.ua