Neural networks to solve modern artificial intelligence tasks

D V Gruzenkin¹, A V Sukhanova¹, O S Novikov¹, G V Grishina¹, V N Rutskiy¹, R Y Tsarev¹ and K Y Zhigalov³,⁴

¹ Siberian Federal University, 79 Svobodny pr., Krasnoyarsk, 660074, Russia
² V A Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, 65, Profsoyuznaya street, Moscow, 117997, Russia
³ Moscow Technological Institute, Block 2, 8, Kedrova, Moscow, 117292, Russia

E-mail: gruzenkin.denis@good-look.su

Abstract. Today the information technologies are increasing and improving by leaps and bounds becoming smarter and faster. And artificial intelligence is penetrating deeper and deeper in everyday life. So, artificial intelligence field is becoming more interesting for modern scientists and engineers. Artificial neural networks are used all over the world as one of the approaches that can provide with the high relevance level of resolving results of badly formalized or unformalized tasks. In this article a review of artificial neural networks development tools of different kinds is presented. Also there is a detailed description of all of them and the main features are emphasized. As a result, the table is presented that allows to make a quick compare of described tools and decide if any of them are suitable for a reader or not.

1. Introduction
During the last decade a lot of hardware and software technologies were invented. Technology development covers more and more areas of human activity, and it requires a lot of different resources [1]. So software developers are faced with the task of intelligence resources optimization, that a specially actually in such fields as space and defense industry, banks and finance, underwater and underground research, chemical industries and others [2]. Moreover, the Internet is becoming more and more complex today. It provides users with a wide set of various accessories such as searching systems, online translators, maps and GPS services etc. However, the extent of the World Wide Web is so great and the level of users’ literacy is so low that it results in an increasing number of intelligent information technologies and software. There are a lot of tools, methods and approaches to create an artificial intelligence today. The application of artificial neural networks is one of them.

Artificial neural networks (ANN) look like simplified models of biological neural networks [3]. This technology enables to create an artificial intelligence instrument that allows to resolve badly formalized or unformalized tasks. These kinds of tasks were solved only by human brain up to the second half of the 20th century. These tasks include the recognition of images (for example, pictures and voice speech), deep analysis of any data (for instance, texts), or even drawing pictures. So, generalizing, ANN could solve a lot of various objectives [4].

Artificial neural network is a set of neurons that are connected to each other. Every neuron gets an input signal for processing. All neurons use the same processing algorithm [3]. Herewith, the user of artificial network usually obtains a correct result. It is connected with the adjustment of signals that were sent from one neuron to another. The input data is processed in each neuron in its own way due to the
signal strengthening or signal weakening and the great amount of them minimizes the probability of an error [3].

Some time ago when the artificial neural networks started to appear, developers had to solve not only scientifically oriented tasks that related to the features of the ANN subject area and training networks methods. They also had to solve problems of tools choice, approaches and methods for developing artificial neural networks. Later a considerable experience was accumulated in the sphere of the development approaches. It was accompanied with a multitude of tools (libraries, frameworks, medium development, etc.), intended for a more complete ANN implementation. At present moment it can be difficult for a beginner researcher or developer of artificial neural networks to find a suitable development tool and understand how it works. This circumstance determines the relevance of this article.

The review of the most popular tools for development and creation of artificial neural networks can be useful both for students, graduate students and young scientists who begin the research in this field and for practicing software engineers and software architects.

2. Methods and tools

2.1. TensorFlow

One of the most famous tools for creating ANN is the TensorFlow framework by Google Brain Company. They study and develop artificial intelligence based on deep machine learning. Initially this tool was developed to solve local problems of Google [4], but then it was decided to release TensorFlow as free software which was distributed under Apache 2.0 license. That means free developers support this product in our days. There is a lot of documentation about TensorFlow. It has a vast community, for example on GitHub. The participants of this community release updates and fix bugs regularly.

The TensorFlow API original language is Python. However, there are interfaces for other programming languages such as C++, Java, Go, Swift. TensorFlow is a cross-platform tool that allows to develop neural networks under operating systems of both Unix and Windows families. It is a significant advantage of the framework. In addition, TensorFlow can be executed on a variety of hardware platforms based on different processor types (CPUs, GPUs, TPUs) which allow to use the framework including desktops and mobile devices [5].

It is important to note that this framework requires Anaconda to support GPU architecture (the Python programming language distribution) and CUDA Toolkit (the development environment for the GPU with CUDA support). The framework allows to realize data flow graphs under condition that pieces of data can be processed by a set of algorithms which were defined in a graph [6]. This tool is applied for intellectualization of many Google services, for instance such products as Google Cloud Translation and Google Photos. TensorFlow is useful for the development of different kinds of diagnostic systems which include human health diagnostic systems [7]. In addition, the framework simplifies the implementation of some artificial intelligence functions such as voice processing, computer vision, classification problems resolving, machine translation etc. for existing software. As a result, TensorFlow is suitable for a wide range of tasks [6].

In addition, TensorFlow is equipped with its own logging system and visualizer. Linear algebra together with other internal processes are transparent, i.e. user has an opportunity to track calculation execution without any additional tools. All these features of the framework reflect its advantages. However, it should be noted that TensorFlow can be quite difficult to understand and therefore less suitable for scientists and developers who are the beginners at machine learning [5].

2.2. Caffe

Another framework is called Caffe. It is mostly focused on commercial use and is distributed under the BSD 2-Clause license [8]. Similar to the previous framework, it has a community on GitHub and quite an extensive documentation. It was developed at the Berkeley Vision and Learning Center (BVLC) with the participation of independent developers. The installation requires the CUDA Toolkit and the
NVIDIA cuDNN-library for neural networks training and the NVIDIA NCCL-library for running Caffe on multiple GPUs [9]. In comparison to Tensor this framework is designed to solve a smaller amount of tasks. Generally these tasks are aimed to image classification and segmentation. It explains its use in the areas related to computer vision, multimedia, and less often to speech and voice recognition. Caffe can be run on both CPU and GPU. This tool is released for Linux, macOS and Windows operating systems. It has an architecture that allows an instantaneuous deployment of a network which provides ease of use and enhanced functionality [8]. In addition, Caffe has some training models integrated into the system. It contributes to the development of new algorithms for third-party developers. It is important to note that the Caffe code is extensible. This technology also supports a Python interface. Caffe can be recommended for developers who have an advanced level of the C++ language [10].

The advantage of this framework is in its high speed. As for the drawbacks it has the lack of an intuitive interface which makes the creation of a new layer of the neural network quite time-consuming. Caffe also has a minimum number of opportunities for modeling of recurrent neural networks [10].

2.3. OpenNN
Open Neural Network Library (OpenNN) is an open-source library for deep machine learning, distributed under the GNU LGPL license and is opened for revision by third-party developers. It was developed by Spanish company Artelnics in C++ [11]. This choice was made due to the most efficient use of memory and higher performance [11]. OpenNN has an extensive documentation in English and a community on GitHub. The library is suitable for computers both CPU-based and GPU-based. And as a result it requires OpenMP (an API for programming multithreaded applications on multiprocessor systems with shared memory) or CUDA Toolkit to be installed. The library is a cross-platform service [11]. The Open architecture is designed in such a way that allows to create an unlimited number of nonlinear processing layers in the neural network. It creates some universal approximation properties [11].

For the comfortable work with the library OpenNN, software Neural Design was also developed. It has a user-friendly graphical interface and it greatly simplifies data entry and interpretation of the results. There is also an Arsenal of advanced Analytics tools in the form of automatically generated tables, charts and tools to work with them. Neural networks developed by OpenNN can be applied to data mining [11]. For example, this technology has already been used in the fields of engineering, energy and chemistry [11]. But OpenNN has its drawbacks. The main problem for the users who have just started the work is the entry threshold. It happens because of the unusual terminology and the general organization of work with the model. Also the training parameters themselves may seem unusual even to the users who have already had some experience in machine learning [12].

2.4. Apache MXNet
Another solution for the development of neural networks is Apache MXNet. It is a system with API for machine learning and obtaining logical conclusions [13]. It was developed by a Distributed (Deep) Machine Learning Community, and it is distributed under the Apache 2.0 license [13]. This also means that the authors of the system are expecting a revision from third-party developers. The cross-platform of MXNet and the ability to work on peripheral devices, mobile applications and even in the cloud are some specific features and the advantages of MXNet. It is possible due to the use of the Gulo interface. To get started with MXNet, you need to install a platform to deploy machine learning models, such as Amazon SageMaker, as well as the images to create your own AWS Deep Learning AMI environment [13]. The system is released for Linux, macOS, Windows and Android, iOS [14]. It is widely spread among Apache, Microsoft, Apple and Samsung developers. It allows to perform symbolic calculations, moreover the system assembly provides tools and components for the images classification on default. There is a support for the CUDA architecture. Due to the distribution of loads between several graphic processors, the system stands out with its high performance. Also it is possible to obtain inferences on the basis of serverless and batch computing [13]. Despite the fact that MXNet compiles in C++, it
supports work in a number of other programming languages, such as Python, R, JavaScript or Go. It has pre-trained models [13].

2.5. AML
Amazon Machine Learning (AML) is a managed service for building machine learning models and creating predictions. It enables to develop a variety of scalable, intelligent applications. It was developed by Amazon [15]. The machine learning algorithms implemented in this system are used in a number of local company systems. There is an extensive documentation in English as well as a community on GitHub. It is distributed under the Service Offerings License which assigns all rights to the source code directly to the developer and their licensors [15]. In fact, Amazon Machine Learning is a full-fledged framework that allows to work on both CPU and GPU. Released for Windows, however, it can be run under other operating systems based on Windows PowerShell Core which in its turn is an extensible open-source automation tool from Microsoft consisting of a shell with a command line interface and a related scripting language [15]. The problems of a binary classification, multi-class categorization and regression from the source data to create the model can be solved by means of AML. A variety of neural networks are created as well. They can be used in a number of branches of modern life. The best evidence of it is the numerous developments of Amazon (such as Prime Air or Amazon Go). The main advantage of this service is a large number of tools for Analytics. In addition, AML has an intuitive interface, so this framework is the most suitable for both a beginner and an experienced specialist.

2.6. CNTK
The next neural network development tool is the Microsoft Cognitive Toolkit (CNTK), a tool for deep learning. Like many analogues, it is an open-source product [16]. The tool was developed by Microsoft. Distributed under the MIT License [16]. It has an extensive documentation and a community on GitHub, thus the project receives constant support from both the developer corporation and third-party developers. This tool can be implemented in computing machines both CPU-based and GPU-based [17]. Moreover, this framework describes the neural network as a series of computational steps combined by means of a connected graph. It allows to create and combine all of the popular types of neural network models such as recurrent networks, convolutional and direct neural networks. Microsoft Cognitive Toolkit implements the stochastic gradient descent (Stochastic Gradient Descent, SGD). It is a method of the real-time training when the data comes gradually so that the neural network does not need the complete amount of data to solve the problems of classification and regression analysis [18]. CNTK can be connected to programs implemented in the Python, C++, C# languages as a library, and it is used as a standalone machine learning tool that functions in its own descriptive language BrainScript [16]. The tool is designed to work under Linux and Windows operating systems. One of the advantages of the Microsoft Cognitive Tool is that it can implement its algorithms on several GPUs using the algorithm of parallel computing and increasing significantly the speed of calculation [16]. Built-in tools allow to configure carefully the neural network and adjust the process of its training even more carefully. With the help of the Microsoft Cognitive Toolkit, the user can develop neural networks to solve any type of tasks. So, the range of its application in human life is practically unlimited. Moreover, the CNTK supports ONNX (Open Neural Network Exchange). It provides the compatibility of the developed neural network models with Caffe, MXNet and other popular machine learning tools [16].

2.7. FANN
Fast Artificial Neural Network (FANN) is another deep learning library supporting the development of direct-propagation neural networks. Developed by Steven Nissen, it was first mentioned in 2003 [19]. It is distributed under the GNU LGPL [20]. So, it refers to open-source development. It has a community on GitHub, and it is very well documented. It is suitable for both CPU-based and GPU-based computers. Since this development tool is distributed in the form of a library it is considered as a full cross-platform. It is a significant advantage. FANN provides a framework that allows to manage easily the learning process of neural networks on different data sets. In addition, this library provides API to work in more than 20 programming languages including the most popular ones, such as C# and Python [20]. To ensure
even better convenience a lot of graphical interfaces were developed for this library, such as FANNTool, Agiel Neural Network, Neural View, FannExplorer, sfann and others. All of them are designed to simplify and speed up the process of neural networks deep learning. It supports the users who are not familiar with programming. As a result, the library determines the low entry threshold for users. So, it can be concluded that FANN is useful for both the beginners and the professionals. Training of neural networks takes place on the basis of the reverse propagation method. This method implies the calculating of the gradient that is necessary for the calculation of weights to be used in the network [21]. Internal functions for neural network training are optimized to reduce training time as much as possible [20]. Despite all these advantages, the future of the project is unknown, because its support is suspended in 2015, and GitHub users report that they can no longer contact the author.

3. Results and discussions
To generalize all the information mentioned above all comparative results are presented in the table 1.

Table 1. Comparison of the tools.

| Parameter          | TensorFlow | Open Neural Network | Apache MXNet | Caffee | Amazon Machine Learning | Microsoft Cognitive Toolkit | FANN |
|--------------------|------------|---------------------|--------------|--------|------------------------|-----------------------------|------|
| Creator            | Google     | Artelnics           | Distributed (Deep) Machine Learning Community | Berkeley Vision and Learning Center (BVLC) | Amazon | Microsoft | Steffen Nissen |
| License            | Apache 2.0 | GNU LGPL            | Apache 2.0 | BSD 2-Clause License | Amazon Software License | MIT License | GNU LGPL |
| Languages          | Python, C++, Java, Go, Swift | C++ | C++, Python, R, JavaScript, Go and others | C++, Python, MATLAB | Amazon, .NET languages, JavaScript | Python, C++, C#, BrainScript | Python, Ruby, C#, PHP, JavaScript, Prolog and others |
| OS                 | Windows, MacOS Linux | Cross Platform | Linux, iOS, Windows, Android | Linux, MacOS, Windows | Linux, MacOS, Windows | Linux, Windows | Cross Platform |
| Processor          | CPU, GPU | CPU, GPU            | CPU, GPU | CPU, GPU | CPU, GPU | CPU, GPU | CPU, GPU |
| Execution on PC    | +         | +                   | +            | +       | +                   | +                     | +   |
| Execution on mobile device | –        | +                   | +            | –       | –                   | –                     | +   |
| Suitable for a beginner | –       | –                   | +            | +       | –                   | –                     | +   |

– it is distributed as a source code that allows to build software under preferable operation system.

Table 1 demonstrates some main features of the tools described in the article. The tools have a lot of similarities. It is possible because of a similar aim. All of them create deep machine learning systems but in different fields. The important criteria for the users of these tools are technical peculiarities, usability, easy installation and quick starting. Software tools of this kind could be used for modeling complex objects in research, analytic and engineering practice and learning activities in natural, social and humanitarian studies [22].
4. Conclusion
The main purpose of this article was to present a short review of existing artificial neural networks development tools. It is impossible to review all existing tools, so we could describe only seven the most popular items. However, the presented tool set is very diverse by both fields of application and technical and usability characteristics. This article is useful not only for beginner-developers but it may be beneficial for senior engineers and scientists as well, as they can find alternative tools for their projects implementation. These tools are a base of previously accumulated knowledge. It is possible to use them to simplify further work with neural networks. That knowledge base is becoming more and more complicated every year, so new advanced tools will appear for neural networks creation and training and the range of their application will expand. However, the given list of tools is up to date.

Acknowledgments
This work is supported by Grants Council of the Russian Federation President according to the research project CII-1292.2019.5.

References
[1] Evdokimov I V, Khaluimov A S, Sokolov N V and Golokhvastov S E 2017 The Information Community of the Arctic in Russia: Evaluation of the Expenses for the IT Projects Development, Characteristics of the Labour Costs Calculating Arctic Yearbook ed Heininen L, Exner-Pirot H et al (Iceland: Northern Research Forum) pp 1-15
[2] Tsarev R Yu, Gruzenkin D V, Kovalev I V, Prokopenko A V and Knyazkov A N 2016 Evaluation of availability of cluster distributed disaster tolerant systems for control and information processing based on a cluster quorum IOP Conf. Ser. Mater. Sci. Eng. 155 (1) 012041
[3] Yegnanarayana B 2006 Artificial Neural Networks (New Delhi: Prentice-Hall of India Pvt Ltd)
[4] Evdokimov I V, Markomenko A S, Konstantinov V I, Zagrebin V A and Maksimov A E 2017 The Decision of the Traveling Salesman Problem in Two Different Ways: "Hungarian Method" and "The Method of Branches and Borders" Integration of the Scientific Community to the Global Challenges of Our Time: materials of the 2nd int. scientific-practical conf. vol 1 (Osaka: Regional Academy of Management) pp 294 – 8
[5] Witten I H, Frank E, Hall M A and Pal C J 2017 Data Mining: Practical Machine Learning Tools and Techniques (Cambridge: Morgan Kaufman)
[6] Abadi M et al 2015 TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems Preprint arXiv:1603.04467
[7] Abadi M et al 2016 TensorFlow: A system for large-scale machine learning 12th USENIX Symp. on Operating Systems Design and Implementation (Berkeley: USENIX Association) pp 265-83
[8] Rampseck L and Goldenberg A 2016 TensorFlow: Biology’s Gateway to Deep Learning? Cell Systems 2 12-4
[9] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S and Darrell T 2014 Caffe: Convolutional Architecture for Fast Feature Embedding Proc. of the 22nd ACM Int. Conf. on Multimedia (New-York: Association for Computing Machinery) pp 675-8
[10] Ahmed A, Jones M and Marks T K 2015 An Improved Deep Learning Architecture for Person Re-Identification IEEE Conference on Computer Vision and Pattern Recognition (Boston) (Piscataway: IEEE) pp 3908-3916
[11] Lopez R 2010 OpenNN: Open Neural Networks Library (Salamanca: Artificial Intelligence Techniques, Ltd)
[12] Kumar T and Aneja P 2018 Tools and Techniques of Machine Learning paradigm: A Study and Analysis Int. J. Electron. Comput. Sci. Eng. pp 152-5
[13] Cyphers S et al 2018 Intel nGraph: An Intermediate Representation, Compiler, and Executor for Deep Learning Preprint arXiv:1801.08058
[14] Tripathi G, Singh K and Vishwakarma D K 2018 Convolutional neural networks for crowd
behaviour analysis: a survey Vis. Comput. (Berlin: Springer) pp 1-24
[15] Rodriguez J I, Blanco M and Gonzalez K 2018 Proposal of a Supply Chain Architecture Immersed in the Industry 4.0 Proc. of the Int. Conf. on Information Technology & Systems vol 721 ed Rocha A and Guarda T (Basel: Springer International Publishing) pp 677-87
[16] Seide F 2017 The computer science behind the Microsoft Cognitive Toolkit: An open source large-scale deep learning toolkit for Windows and Linux Int. Symp. on Code Generation and Optimization (Austin) (Piscataway: IEEE)
[17] Pathak S, He P and Darling W 2017 Scalable Deep Document / Sequence Reasoning with Cognitive Toolkit Proc. of the 26th Int. Conf. on World Wide Web Companion (Perth: Australia/ Int. World Wide Web Conferences Steering Committee) pp 931-934
[18] Awan A A, Chu Ch-H, Subramoni H and Panda D K 2017 Optimized Broadcast for Deep Learning Workloads on Dense-GPU InfiniBand Clusters: MPI or NCCL? Preprint arXiv:1707.09414
[19] Nissen S 2003 Implementation of a Fast Artificial Neural Network Library (FANN) Dept. of Computer Science University of Copenhagen Report
[20] Deperlioglu O and Kose U 2011 An educational tool for artificial neural networks Comput. Electr. Eng. 37 (3) 392-402
[21] Hecht-Nielsen R 1989 Theory of the backpropagation neural network International Neural Network Society 1988 First Annual Meeting vol 1 (1 SUPPL) (Boston: MA, USA/HNC Inc.) p 445
[22] Semenenko M G, Kniazeva I V, Beckel L S, Rutskii V N, Tsarev R Yu, Yamskikh T N and Kartsan I N 2019 How to use neural network and web technologies in modeling complex technical systems IOP Conf. Ser. Mater. Sci. Eng. 537 (3) 032095