Neural Network Software Technology Trainable on the Random Search Principles

Victor Krasnoproshin and Vadim Matskevich
Belarusian State University
Belarus, Minsk, Nezavisimosti av. 4, 220030
Email: krasnoproshin@bsu.by, matskevich1997@gmail.com

Abstract **—The paper deals with a state-of-art neural technology programmed implementation problem in which the training process is based on random search algorithms. *T*raining neural networks is a typical optimization problem. At the initial stage of neural network technologies development, various variants of gradient methods were traditionally used to solve such problems. Such methods, as a rule, met the requirements for the problem in terms of quality and speed of training. However, with the appearing of a new class of applied problems, the situation has changed. The traditional approach to training using gradient methods did not always meet the requirements of the applied problem in terms of the resulting solution quality. *T*he paper proposes one of the options for the software implementation of neural network technology (in the form of a framework) according to the ostis 2021 standard, in which random search algorithms are used to train neural networks.**
***K*eywords—framework, neural network, training, random search algorithms, annealing method.**

I. Introduction

In modern society, digital data processing technologies based on artificial intelligence methods are rapidly developing. In particular, neural network technologies based on various artificial neural networks architectures have become widespread.

Due to their high flexibility and the ability to tune to the subject area, they are actively used to solve a wide class of applied problems. However, neural network tuning for the problem being solved (training) is a time-consuming process.

Automation of the training process is effectively solved within the existing frameworks [1]. They allow us to simplify the neural networks training process by using already implemented training algorithms. The use of gradient optimizers inside such frameworks is quite justified. Gradient methods have a high convergence rate and in practice provide an acceptable solution quality obtained. When developing the first automated neural networks training systems, there wasn't a wide variety of computing devices, so they didn't have cross-platform property. However, with the computing technology development, more and more calculations are transferred from the central processor to connected computing devices. This allows us to use simultaneously a large number of devices and significantly increase the efficiency of computing. Moreover, modern frameworks have become cross-platform. However, as digital technologies develop, the class of applied problems for which the solution quality obtained is critical is constantly expanding.

It should be noted here that many modern frameworks use gradient optimization methods, which do not always guarantee the optimal solution achievement. Consequently, when solving such applied problems, they are not effective enough, which makes the problem of developing a software package with alternative training methods up to date.

The paper proposes a framework's software implementation variant, in which random search algorithms are used to train neural networks.

II. Problem analysis

Currently, a wide range of applied problems is solved using neural network technologies implemented in the form of frameworks. This technology is a set of software and algorithmic tools that implement the architectures of various types of neural networks focused on solving various classes of applied problems.

Today, there are a number of frameworks for solving machine learning problems. Among the most popular, in particular, the following can be distinguished.

MXNet is a high-performance and cross-platform framework that is widely used in solving applied problems. However, this framework has certain drawbacks. This is not a very convenient user interface compared to simpler frameworks and a rather meager range of optimization algorithms. It only supports some modifications of gradient descent. This framework completely lacks support for random search methods.

Tensorflow 2. is cross-platform and has a simple user interface. Currently, it is the most common framework for applied problems solving. Supports learning with various gradient methods and genetic algorithm. The disadvantages include insufficiently high performance, since it contains the costs of high-level programming languages and a poor variety of non-directional optimization algorithms.

Caffe 2 is a high performance cross platform framework. However, it lacks support for recurrent neural networks