NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "HSE and University of London Double Degree Programme in Data Science and Business Analytics"

Software project report

on the topic Neural Networks from Scratch

Student:
group БПАД232

Sign

A.V.Suvorova
Name, Surname, Patronymic

02.02.2025
Date

Supervisor:

D.V.Trushin
Name, Surname, Patronymic

Docent / FCS HSE
Position / Company

12.06.2020	11	
Date	Grade	Sign

Moscow 2025

Contents

Abstract

This report presents a software library designed for constructing fully connected neural networks with customizable architectures, including variable number of layers and their sizes. The library provides tools for training models on datasets and generating predictions.

1 Introduction

"Neural Networks from Scratch" is a software project comprising a library for constructing and training fully connected neural networks, along with a console application demonstrating its functionality. The library supports core features for working with fully connected neural networks, including creating networks of various configurations, training them, and generating predictions on test data. The project is primarily intended for educational purposes, providing students with a practical tool for learning about neural networks.

2 Overview of sources

In order to work with matrices, the EigenRand library [1] is used as a plug-in on the github.

3 Description of functional and non-functional requirements

The program consists of source code for classes implementing a neural network library. The library provides the following header files:

1. ActivationFunction.h: This file contains the ActivationFunction class, which represents the activation function of a fully connected neural network layer. The class supports methods such as a constructor for initializing the activation function and its derivative, evaluate0 for computing the activation function's output, and evaluate1 for calculating the derivative's output.

2. LinearLayer.h: This file provides access to the LinearLayer class, representing a fully connected neural network layer. The class stores a weight matrix, a bias vector, and an initializeWeights function. It includes methods like forward for computing the linear transformation $A*x+b$, and backward for calculating gradients, updating weights and biases, and returning the updated output.

3. LossFunction.h: This file provides the functionality of the LossFunction class, which represents the loss function of a neural network. The class includes the following methods:

   (a) A constructor that accepts two functions: one for calculating the distance between two matrices composed of column vectors, and another for computing the derivative of this distance function.

   (b) The dist method, which returns a real value representing the distance between two matrices of column vectors, based on pairwise distances between corresponding vectors.

   (c) The derivativeDist method, which returns a matrix composed of rows representing the derivatives of the dist function for corresponding column vectors of the input matrices, providing initial gradients for each input column vector.

References

[1] Eigen. Eigen: A c++ template library for linear algebra. 2024.