

# Deep learning: Neural Network from scratch

## *training a neural network to solve the mnist dataset*

Adrien PELFRESNE - Alexis VAPAILLE

May 24, 2024

## Contents

1	Abstract	1
2	Motivation	2
3	Neural network categories	2
4	Sequential neural network	2
5	The learning process	2
6	Activation	2
7	Optimizers	2
8	Convolutional neural network	2
9	Implementation	2
10	Results	2
10.1	Multi-layer perceptron . . . . .	2
10.2	Convolutional multi-layer perceptron . . . . .	2
11	Discussion	2
12	Conclusion	2

## 1 Abstract

In this report, we are going to review, and present our work of creating a deep neural network "library" from scratch, in rust, without the use of any machine learning or deep learning library.

The goal of this project was to solve the **mnist** dataset of handwritten digits, and to provide performances comparasion between two multi-layer-perceptron, with and without convolution.

Our library let you create your own sequential neural network, with a simple and declarative api and is widely open to extension. Our implementation currently suport

- Dense Layer
- Activation Layer with various activation functions
- Convolutional Layer
- Rehshape Layer
- Optimizer (Adam and Stochastic gradient descent)
- Various parameters initialization methods

## 2 Motivation

The objective with this *from scratch* approach was to learn in details how neural network is really working. because high level library like keras in python and the incredible level of abstraction that come with it, doesn't teach us anything on the underlying process of learning. nevertheless, this underlying work that those library are doing is passionating, we thought it was a shame not to fully understand the whole deep learning process. The only helper library that we have used is ndarray which is similar to *ndarray* from the python library 'numpy'. This allowed us to create a relativazly performant implementation of neural networks, without goind to low level.

## 3 Neural network categories

## 4 Sequential neural network

## 5 The learning process

## 6 Activation

## 7 Optimizers

## 8 Convolutional neural network

## 9 Implementation

## 10 Results

### 10.1 Multi-layer perceptron

### 10.2 Convolutional multi-layer perceptron

## 11 Discussion

## 12 Conclusion