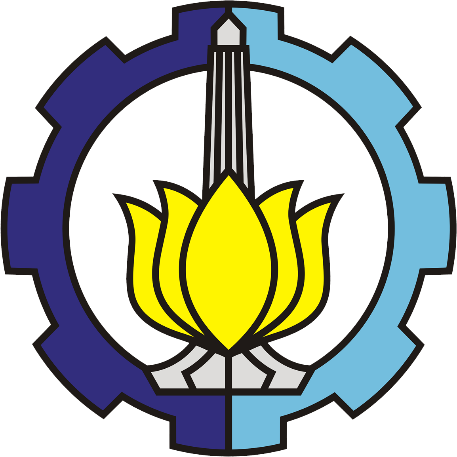
**REPORT**

**MULTY LAYER PERCEPTRON WITH ERROR BACK PROPAGATION DESIGN**

**Course : Fundamental of Intelligent System**

**Lecturer : Dr. Achmad Arifin, ST., M.Eng.**



**Arranged by :**

**Muhammad Yasin**

**07311740000019**

**BIOMEDICAL ENGINEERING DEPARTMENT**

**ELECTICS**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**2020**

**CHAPTER 1**

**THEORETICAL FRAMEWORK**

1. **Intelligent System**

Definitively, Intelligent system is a system with artificial intelligence. In computer science, artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans. Leading AI textbooks define the field as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving".

1. **Artificial Neural Networks**

computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the examples that they process.

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it.

1. **Logic Gates**

A logic gate is an idealized or physical electronic device implementing a Boolean function, a logical operation performed on one or more binary inputs that produces a single binary output. Depending on the context, the term may refer to an ideal logic gate, one that has for instance zero rise time and unlimited fan-out, or it may refer to a non-ideal physical device.

Logic gates are primarily implemented using diodes or transistors acting as electronic switches, but can also be constructed using vacuum tubes, electromagnetic relays (relay logic), fluidic logic, pneumatic logic, optics, molecules, or even mechanical elements. With amplification, logic gates can be cascaded in the same way that Boolean functions can be composed, allowing the construction of a physical model of all of Boolean logic, and therefore, all of the algorithms and mathematics that can be described with Boolean logic. Specifically, the logic gates that will be used in this program are AND, OR and XOR.

1. **Multilayer Perceptron**

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to refer to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation), Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.

An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

1. **Backpropagation**

In machine learning, backpropagation is a widely used algorithm in training feedforward neural networks for supervised learning. Generalizations of backpropagation exist for other artificial neural networks (ANNs), and for functions generally – a class of algorithms referred to generically as "backpropagation". In fitting a neural network, backpropagation computes the gradient of the loss function with respect to the weights of the network for a single input–output example, and does so efficiently, unlike a naive direct computation of the gradient with respect to each weight individually. This efficiency makes it feasible to use gradient methods for training multilayer networks, updating weights to minimize loss; gradient descent, or variants such as stochastic gradient descent, are commonly used. The backpropagation algorithm works by computing the gradient of the loss function with respect to each weight by the chain rule, computing the gradient one layer at a time, iterating backward from the last layer to avoid redundant calculations of intermediate terms in the chain rule; this is an example of dynamic programming.

**CHAPTER 2**

**RESULT AND DISCUSSION**

**Multilayer Perceptron Network Design**

The first part of the program that will be discussed is the steps of designing the said design of the MLP network. The image below displays the overall program display.

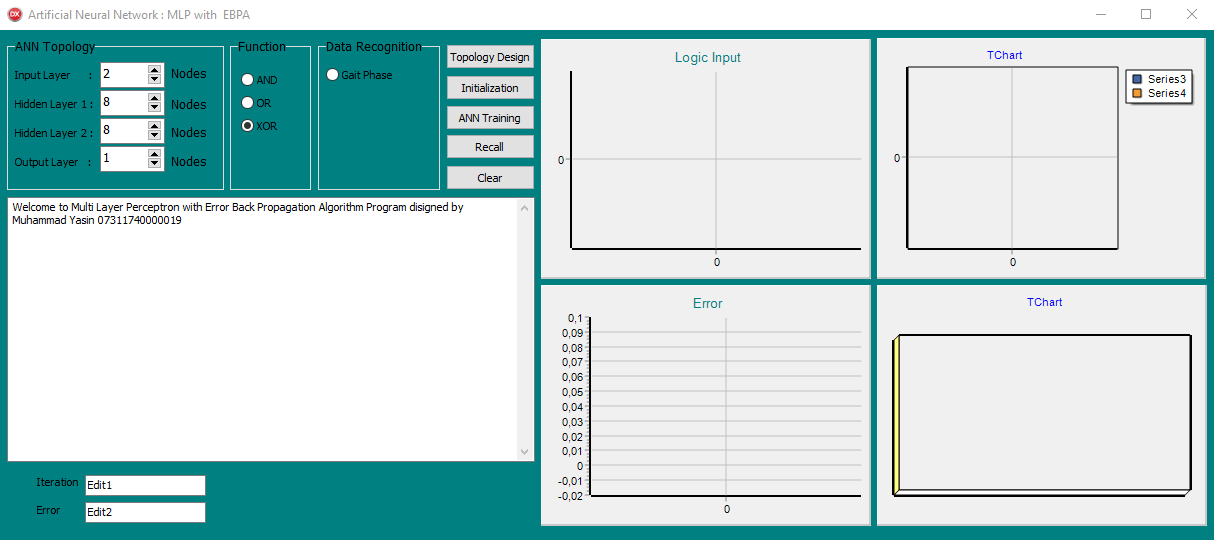
****

Figure : Program Interface

The figure above shows us the interface of the program. The interface shows us some groupboxes that set what process we want to do, such as ANN Topology for determining how many nodes we want to use in this program, function group to choose what logic function to use, and data recognition group to choose the data we want to learn. There is also memo to show us the information about the steps we’ve been done. It shows us the ANN topology information, weight initialization information and some more.

**Topology Design**

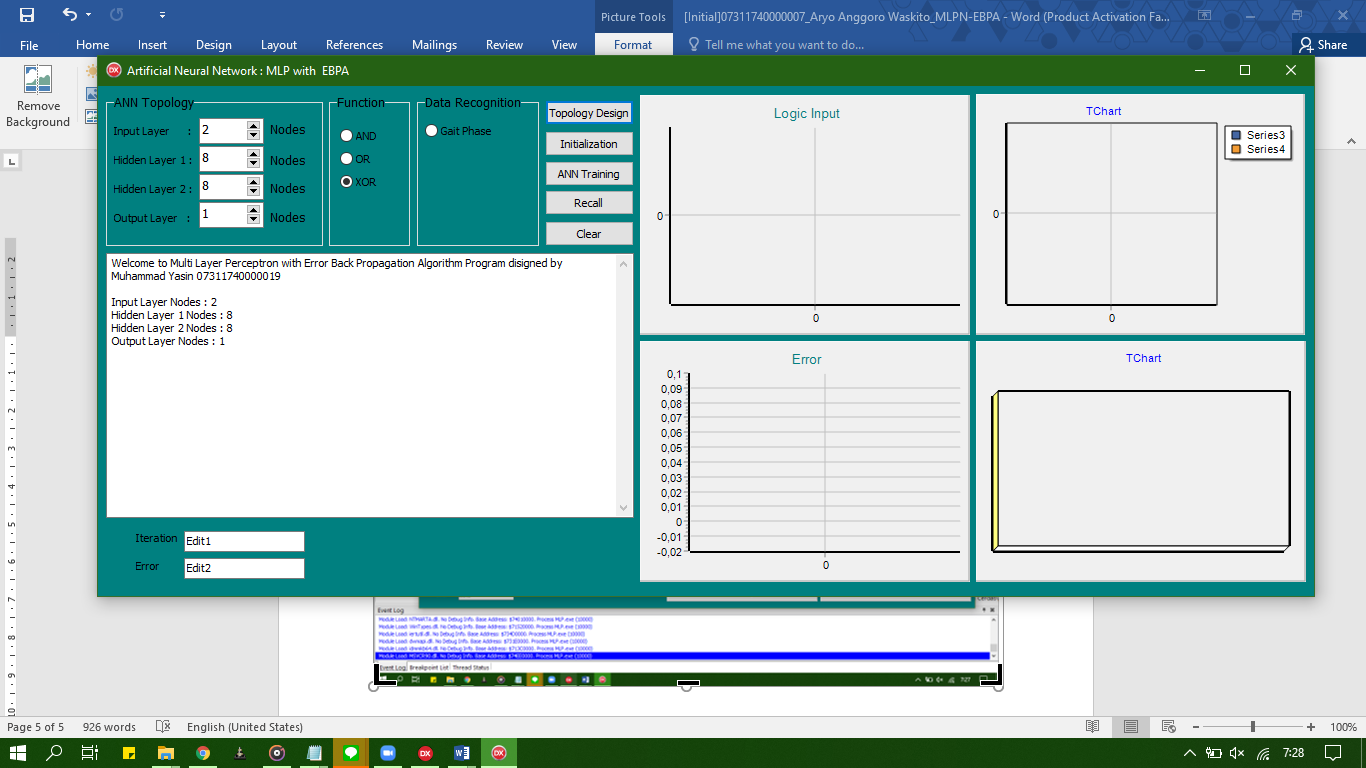


Figure : Memo showing the topology design information

after clicking the topology design button, the memo will show us the ANN Topology Design based on the nodes we inputed.

**Weight Initialization**

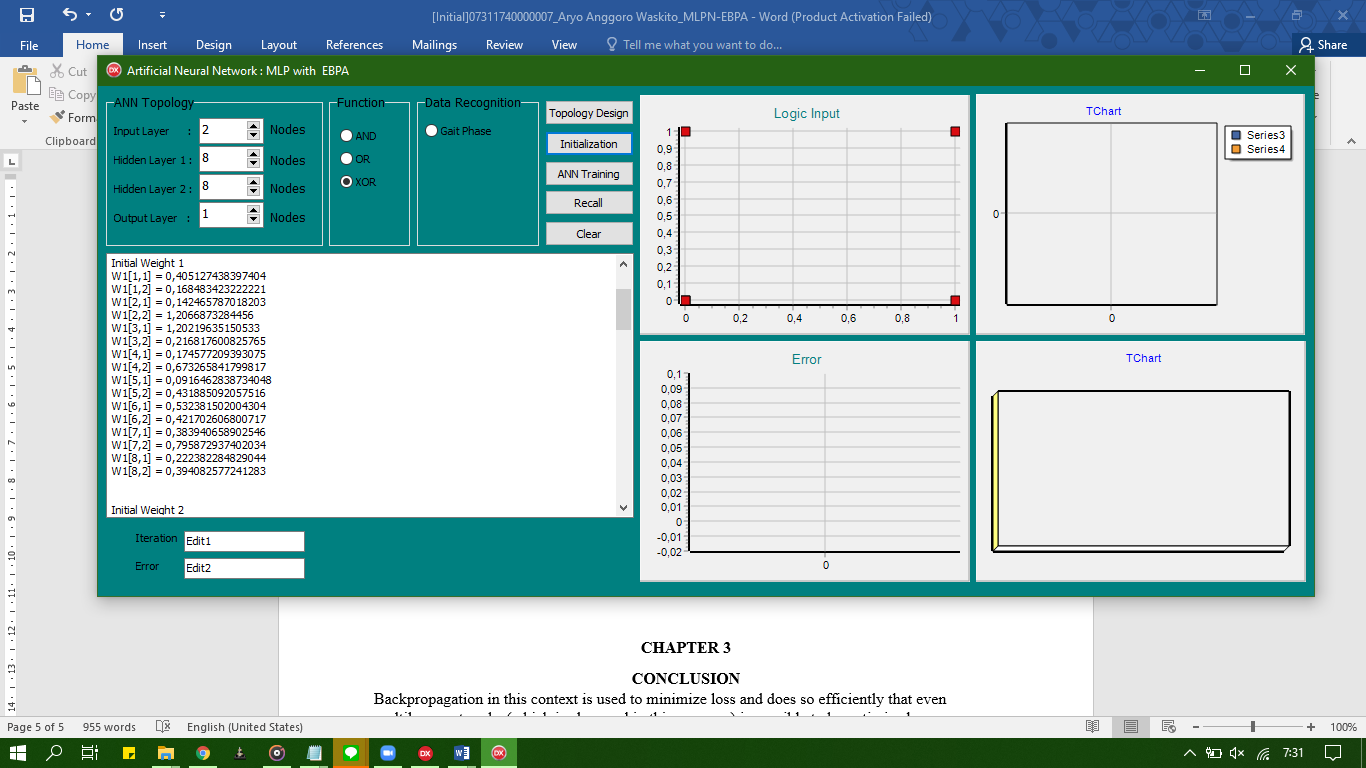
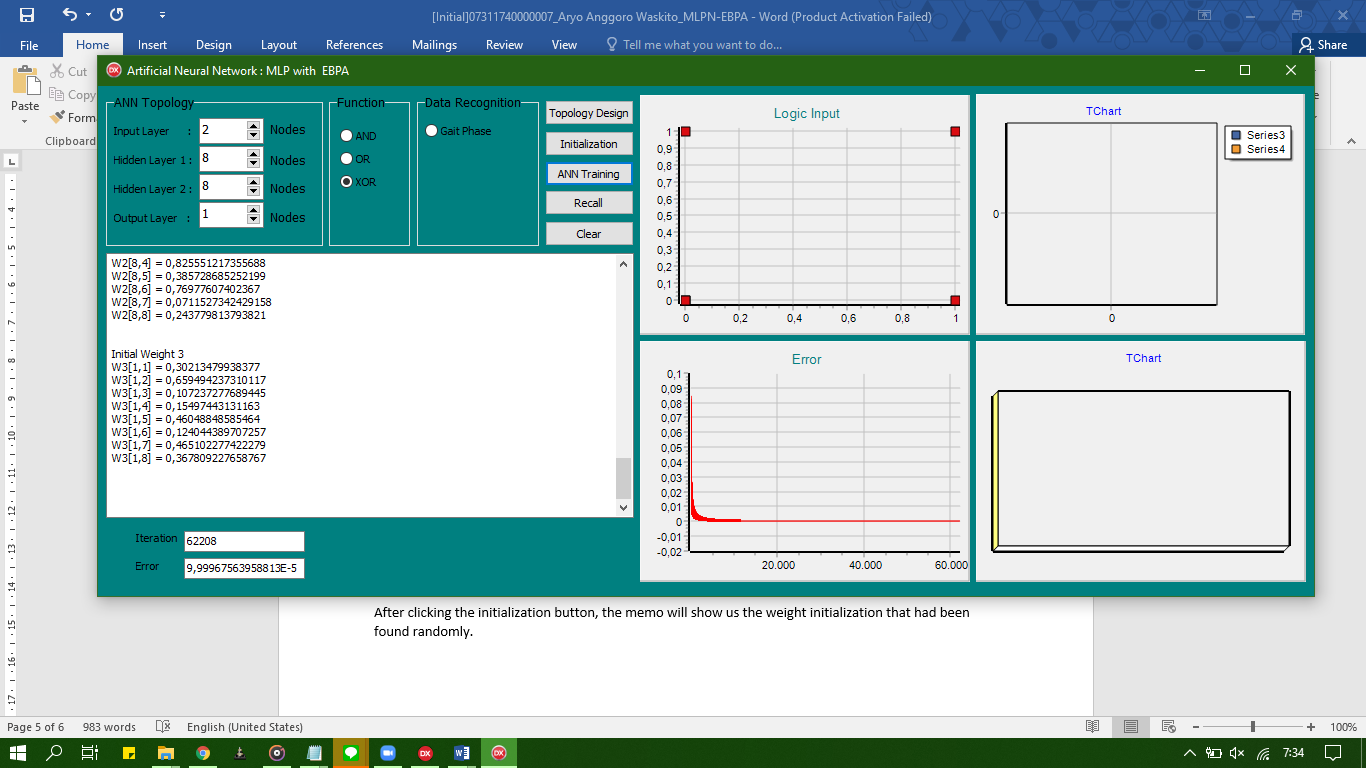


Figure : memo showing the weight initialization

After clicking the initialization button, the memo will show us the weight initialization that had been found randomly.

**Training Process**



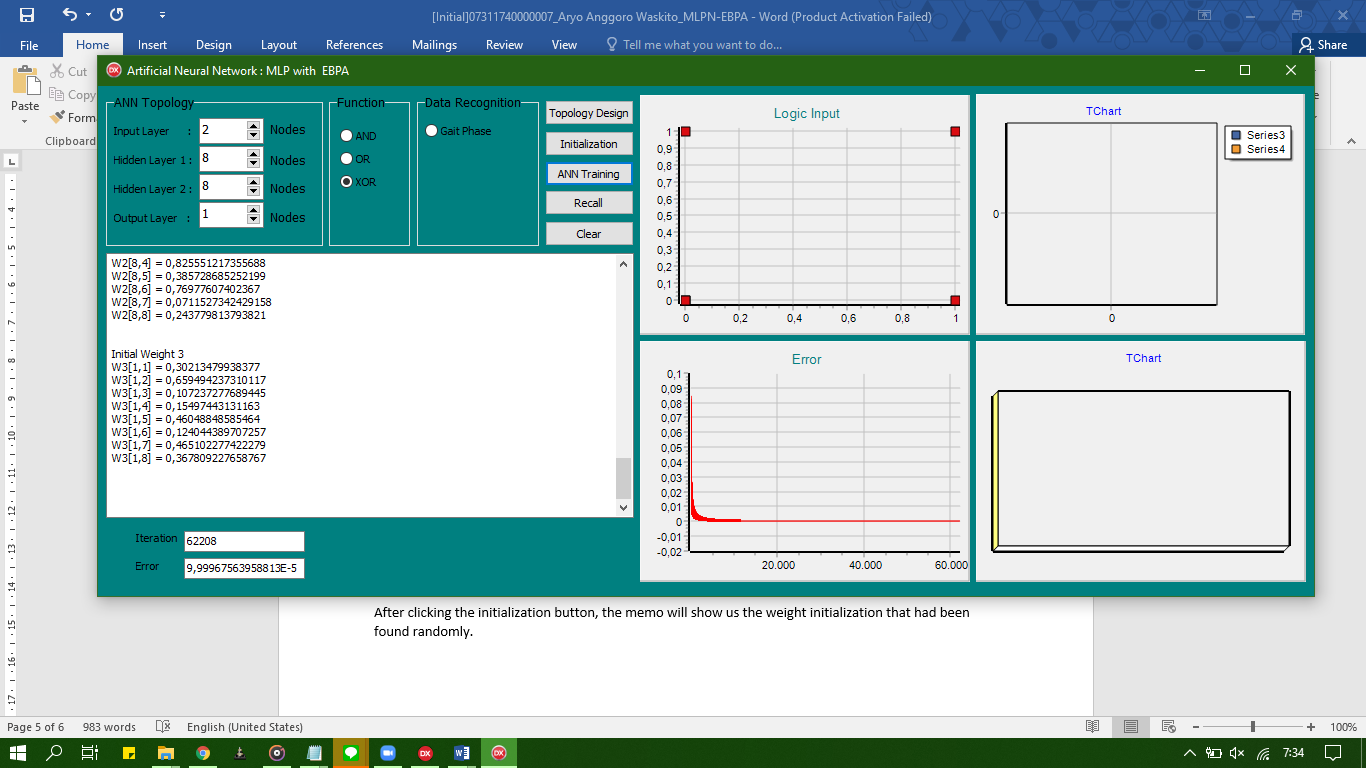


Figure : Logic input and Error graph

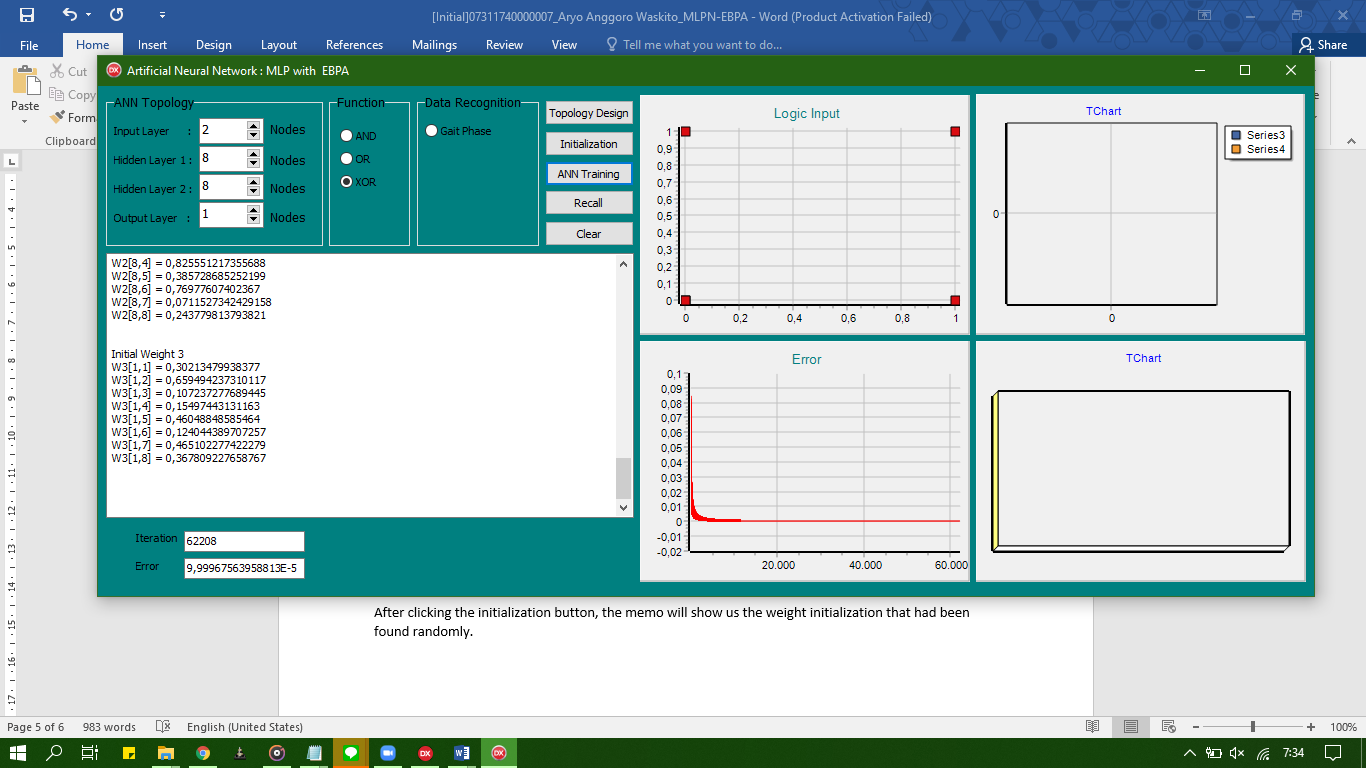


Figure : iteration and final error value

After clicking the ANN Training button, the program will do the training process. This process takes seconds to do because of doing so many iterations. The final iteration is 62208 and the final error value is 9,99967563958813E-5.

**CHAPTER 3**

**CONCLUSION**

Backpropagation in this context is used to minimize loss and does so efficiently that even multilayer networks (which is also used in this program) is possible to be optimized.