ARTIFICIAL INTELLIGENCE FOR COMPUTING AND ITS APPLICATIONS
IN THE FIELD OF NATURAL RESOURCES AND ENVIRONMENT
Lesson 3. Single-layer neural networks and network training.

Dr. Nguyen Du Khang

Summary:

In the previous article, the author presented some core functions and parameters used in artificial neural networks. In this article, the author will present single-layer artificial neural networks and network training algorithms, along with specific examples based on the functions and parameters introduced. Single-layer neural networks, being simple networks, are usually only applied when classifying objects with clearly distinct characteristics. However, this knowledge is important for readers to approach multi-layer artificial neural networks, which will be presented in later articles, serving specific applications that each interested reader can apply to their own professional field.

Keywords: *Perceptron, delta, gradient .*

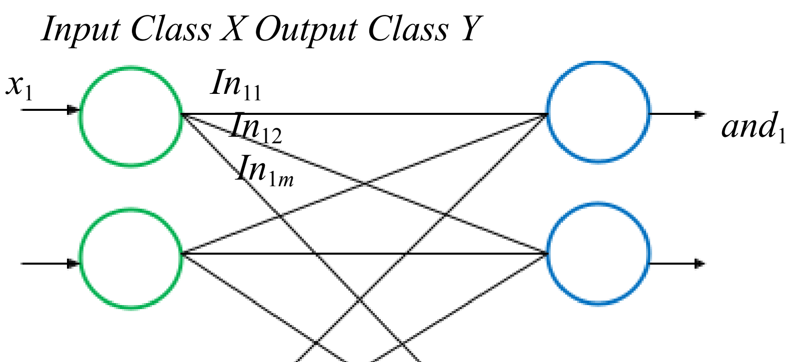
1. Single-layer neural network

A single-layer neural network is the simplest form of artificial neural networks; it consists only of an input and output layer, with no hidden layers. Because the input layer of the network does not perform computation (input = output), the network is called a single layer.

There are many single-layer neural network models such as Hopfield network (1982), network Adaptive bidirectional memory type ABAM (Adaptive Bidirectional Associative Memory Neural Network), Kohonen network (1989), Perceptron network, etc. This section presents...***Perceptron network***A class, proposed by F. Rosenblatt in 1960 and widely used in classification problems.

A single-layer perceptron network is a feedforward network with multiple outputs, often used when the output layer ensures linear separability. Conversely, a multi-layer neural network is used (discussed in subsequent issues).

Single-layer neural network structure:



x_2 In_{21}

In_{22} and_2

In_{2m}

⋮ ⋮

In_{n1}

x_n In_{n2}

In_{nm} and_m

Figure 1. Single-layer neural network

The input vector is denoted as X = $(x_1, x_2, \ldots, x_n)^T$, n- is the number of input signals, corresponding to $n$neurons of the input layer.

Input vector X corresponds to output vector Y = $(and_1, and_2,\ldots, and_m)^T$, m- is the number of neurons in the radiating layer.

Notation for the input weight vector of the second neuron.$j$of the class output is $W_j = (In_{1j}, In_{2j},\ldots, In_{nj})$, $j = 1 \div m$The set of input weight vectors of the output layer forms a weight matrix W = $(W_1, IN_2,\ldots, IN_m)^T$:

$$IN = \begin{pmatrix} In_{11} & In_{21} & \ldots & In_{n1} \\ In_{12} & In_{22} & \ldots & In_{n2} \\ \ldots & & & \\ In_{1m}, & In_{2m} & \ldots & In_{nm} \end{pmatrix}. \quad (1)$$

Then, the input vector of the output layer is calculated using the formula:

$$I = (I_1, I_2,\ldots, I_m)^T = W.X . \quad (2)$$

The output vector of the output layer is Y = $(and_1, and_2,\ldots, and_m)^T$, in there:

$$AND_j = F (I_j), j = 1 \div m, \quad (3)$$

F is the activation function [1].

2. Network training algorithm

Let's assume that***sample pair*** (X_s, AND_s) consists of input vector $X_s = (x_1, x_2,\ldots, x_n)^T$and output vector $Y_s = (and_1, and_2,\ldots, and_m)^T$, n- is the number of input signals,m- is the number of neurons in the radiating layer.

Training a network is the process of determining the weight matrix W (formula 1), such that the input vector X_sDetermine the output vector O relative to Y_s.Satisfied within the allowed error margin.

The training steps are as follows:

<u>*Step 1*</u>Enter data:

- Number of input signals$n$and the number of neurons in the radix layer$m$;
- Sample pair $(X)_s$, AND_s): $X_s = (x_1, x_2,\ldots, x_n)^T$; $AND_s = (and_1, and_2,\ldots, and_m)^T$;
- Network training speed$a$ ($0<a<1$); moment b,training error threshold$i$and loop limits*Epoch*.
- The weight matrix W is randomly selected, usually in the range [-0.5, 0.5]. Give $l = 0$.

<u>*Step 2*</u>Calculate$l = l+ 1$, if$l > Epoch$, move to*Step 3.*

- Determine the input vector of the output layer according to formula (2):

  $I = (I_1, I_2,\ldots, I_m)^T = W. X_s.$
- Determine the output vector $O = (O_1 The_2,\ldots, O_m)^T$:

  $THE_j = F(I_j),\ j = 1 \div m,$

F is the activation function.

- Calculate the error using formula [1]:

$$MSE =(\sum_{j=1}^{m} (and_j - THE_j)^2)/m,$$

if $MSE \le i$, move to*Step 3*.

- Adjust the weight matrix $W = (W_1, IN_2,\ldots, IN_m)^T$:

+ Determine the delta of the output layer using formula [2]:

$$d_j= (O_j - and_j).THE_j.(1- O_j),\ j = 1 \div m$$

Determine the gradient of the synaptic connection between neurons.i ($i = 1\div n$) of the input layer and neuron$j$ ($j = 1\div m$) of the output layer [2]:

$\text{Degree}_{ij} = \text{d}_j \text{The}_i$,

in there, THE_i is the neuron output i of the class: $\text{THE}_i = x_i$, $i = 1 \div n$ (For the input layer, the signals are transmitted directly without processing).

+ Determine the correction increment [2]:

$$\text{D}In_{ij} = - \text{a}.\text{Degree}_{ij} + b.\ \text{D}In_{ij}^{(l-1)},\ i = 1 \div n, j = 1 \div m \quad ,$$

in there, $\text{D}In_{ij}^{(l-1)}$ – is the increment for adjusting the weight of the previous loop.

+ Determine the weight vectors:

$$In_{ij} = In_{ij} + \text{D}In_{ij},\ i = 1 \div n,\ j = 1 \div m.$$

Come back *Step 2*.

Step 3 Network training has ended.

If the network is trained using multiple pairs of samples, we obtain a set of weight matrices. Each weight matrix corresponds to a pair of samples. Typically, the number of sample pairs equals the number of neurons in the output layer of the network.

3. Example of training a single-layer neural network

Let's take a simple one-layer neural network as an example, consisting of an input layer with 2 neurons and an output layer with 2 neurons, as follows:

X Y

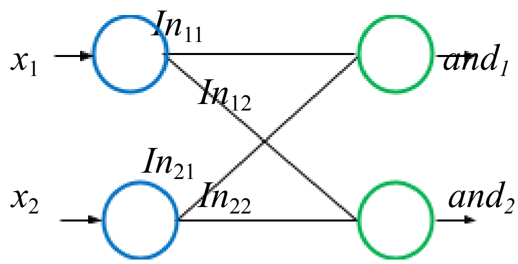


Figure 2. Example of a single-layer neural network.

Train the above single-layer neural network for the sample pair (X, Y) with the following elements: $x_1 = 1$; $x_2 = 0,5$; $and_1 = 1$; $and_2 = 0$.

For network training speed $\text{a} = 20$; moment $\text{b} = 0,3$

Take the weights of the joints: $In_{11} = 0,5$; $In_{12} = 0,5$; $In_{21} = 0,5$; $In_{22} = 0,5$.

	First time	***Second time***	***Third time***
I1	0.75	2.49763	3.15543
I2	0.75	-2.9497	-4.0949
O1	0.67918	0.92398	0.95912
O2	0.67918	0.04975	0.01638
delta1	0.06991	-0.0053	-0.0016
delta2	-0.148	0.00235	0.00026
grad11	0.06991	-0.0053	-0.0016
grad12	-0.148	0.00235	0.00026
grad21	0.03495	-0.0027	-0.0008
grad22	-0.074	0.00118	0.00013
dw11	1.39811	0.52624	0.03205
dw12	-2.9598	-0.935	-0.0053
dw21	0.69905	0.26312	0.01603
dw22	-1.4799	-0.4205	0.00264
w11	1.89811	2.42434	2.4564
w12	-2.4598	-3.3948	-3.4
w21	1.19905	1.46217	1.4782
w22	-0.9799	-1.4003	-1.3977
MSE	0.28211	0.00413	0.00097

In the following paper, the author will introduce multilayer artificial neural networks and training algorithms along with specific examples to solve several problems in the field of natural resources and environment.