

Optimization of giving employee craft assessment using artificial neural network with Hebb algorithm

Segar Napitupulu¹, Zakarias Situmorang²

¹ Students of Master of Information Engineering, University of Sumatera Utara, Indonesia

² Lecturer at the Faculty of Computer Science, University of Sumatera Utara, Indonesia

¹segarn70cs5@gmail.com, ²zakarias65@yahoo.com

Abstract. One of the most important things in human resource management is employee discipline that can be seen in employee presence. It is very important for companies to set up employee attendance systems properly. In general, benefits - allowances are always associated with employee precession data, so that the results of the calculations became accurate and balanced with employee performance. As is the case with craft benefits, this craft allowance is held for companies to improve employee discipline, especially in presence or presence. Based on the background of the above problems, a problem can be formulated, namely the determination of the model in the process of determining the recipients of employee craft benefits so that the monthly income is optimal and significant. Based on the problem, the authors are interested in redeveloping the final project with the optimization of giving employee craft assessment using the hebb rule neural network method.

1. Introduction

1.1 Problem Background

In addition to basic salary, the company has rules for allowances or income that have been set by the company written on the employment contract. But, the reality is not in accordance with employee expectations of the company and there is employee dissatisfaction with changes in income that are not significant for each month. For this reason, it is very useful if the company implements an artificial neural network model as a reference for companies in providing comprehensive benefits, namely in terms of government regulations and to create a balance, between what employees give to companies, and what the company gives to its employees. Along with the development of technology and the increasingly high demands of work time, the salary calculation process has a lot of payroll systems but the rules and formulations are still not perfect, so that some models can optimize significant monthly wages without losses for employees and companies.

One of the most important things in human resource management is employee discipline that can be seen in employee presence. It is very important for companies to set up employee attendance systems properly. In general, benefits - allowances are always associated with employee precession data, so that the results of the calculations become accurate and balanced with employee performance. As is the case with craft benefits, this craft allowance is held for companies to improve employee discipline, especially in presence.

1.2 Formulation of the problem

Based on the background of the above problems, a problem can be formulated, namely the determination of the model in the process of determining the recipients of employee craft benefits so that the monthly income is optimal and significant.

The formulation of the problem above is limited by the following:

1. Data to be examined only includes attendance data taken from PT. Juma Berlian Exim.
2. The method to be used is an artificial neural network with a hebb rule model.
3. Observation of variables for craft allowances includes permits, absences and disobedience.

The research objectives in this paper can be described to pPredict employees who will receive attendance benefits. So that it can reduce the cost of employee salaries that are less than optimal.

2. Artificial Neural Network

2.1 Definition of Artificial Neural Networks

We can know artificial neural networks like artificial brains in scientific studies. Artificial nerve is able to process data and give conclusions from some information received. This can be done by imitating existing activities and expertise in human biology. Brain surgeons say that activities that occur in the brain are remembering, understanding, storing and recalling. Neurons in the brain can also be retrained on patterns of behavior of the human brain.

The thing that needs attention is that artificial neural networks are not programmed to produce certain outputs. All conclusions drawn by the network are based on their experiences during the learning process. In the learning process, input into the neural network are input and output patterns. Then the network will be taught to provide acceptable answers.

The branch of expert system science is very broad and related to various disciplines. This can be seen from various applications in a combination of various sciences. For example, medical applications that are combined from medical science and artificial intelligence. This is what is part of artificial neural networks, namely adopting the ability of the human brain that is able to provide stimulation / stimulation, carry out processes and provide decisions or output.

2.2 Comparison of Biological Neural Networks with Artificial Neural Networks

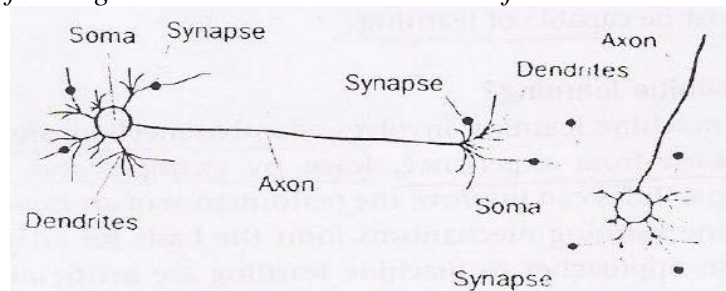


Figure 2.1 Biological Nerve Cells

Before comparing biological neural networks with artificial neural networks, it is very good to first understand what is happening in a biological neural network. Biological nerve tissue is a collection of nerve cells (neurons). Neurons have the task of processing information. The main components of a neuron can be grouped into 3 parts, namely:

1. Dendrite. Dendrit is responsible for receiving information.
2. Cell Body (Soma). Cell body functions as a place of information processing.
3. Axon (Neurit). Axons send impulses to other nerve cells.

Artificial neural networks are arranged with the same assumptions as biological neural networks:

1. Information processing occurs in elements process (neurons).
2. The signal between two neurons is transmitted through connection links.
3. Each connection link has associated weights.
4. Each neuron applies an activation function to the network input (number of weighted input signals). The aim is to determine the output signal. The activation function used is usually a nonlinear function.

The way to learn artificial neural networks is as follows: input into artificial neural networks information that has previously been known to output. Inputting this information is done through nodes or input units. Interconnection weights in an architecture are given initial values and then artificial neural networks are carried out. Table 2.1 shows analogies between artificial neural networks and biological neural networks.

Table 2.1 Analysis of artificial neural networks with biological neural networks	
Artificial Neural Network	Biological Neural Networks
Node or Unit	Body Cell (soma)
Input	Dendrit
Output	Akson
Weight	Sinapsis

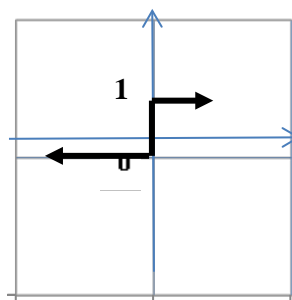
What you want to achieve by training / teaching artificial neural networks is to achieve a balance between memorization and generalization capabilities. What is meant by memorization ability is the ability of artificial neural networks to call back perfectly a pattern learned.

2.5 Activation Function

There are several types of activation functions in neural networks, namely:

1. Hard Limit Function

Single layer networks often use step functions to convert a variable whose value is continuous to a binary output (0 or 1)

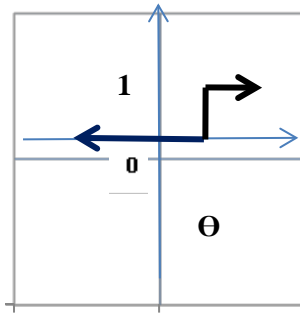


$$y = \begin{cases} 0, & \text{if } x \leq 0 \\ 1, & \text{if } x > 0 \end{cases}$$

Figure 2.2 Hard Limit Function

2. Threshold Function

The threshold function is the hard limit function using the threshold value. Often referred to as the heaviside function.

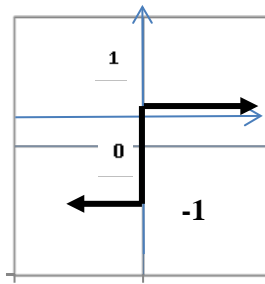


$$y = \begin{cases} 0, & \text{if } x < \theta \\ 1, & \text{if } x \geq \theta \end{cases}$$

Figure 2.3 *Threshold Function*

3. *Bipolar Function*

This function can be said to be almost the same as binary functions (hard limit and threshold), only the output produced by the bipolar function is 1.0 or -1.

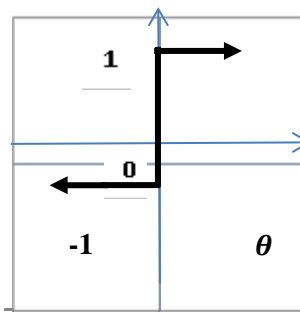


$$y = \begin{cases} 1, & \text{jika } x > 0 \\ 0, & \text{jika } x = 0 \\ -1, & \text{jika } x < 0 \end{cases}$$

Figure 2.4 *Bipolar Function*

4. *Bipolar With Treshold Function*

Bipolar function is almost the same as the threshold function, except that the output produced consists of 1.0 or -1. This function determines as a threshold value.

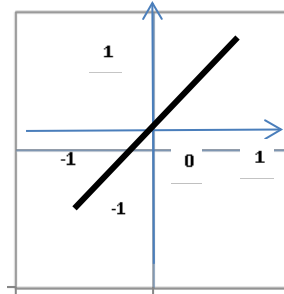


$$y = \begin{cases} 1, & \text{jika } x \geq \theta \\ -1, & \text{jika } x < \theta \end{cases}$$

Figure 2.5 *Bipolar With (Threshold) Function*

5. Linear Function

The Linear function has an output value that is equal to the input value.



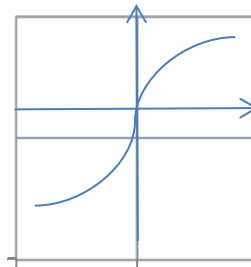
$$y = x$$

Figure 2.6 Linear Function

6. Binary Sigmoid Function

This sigmoid binary function is used for neural networks that are trained with the back propagation method. Has a range from 0 to 1. Therefore, this function is often used for neural networks that require output values in range 0 to 1.

However, this function can also be used by networks whose output values are 0 or 1.



$$y = f(x) = \frac{1}{1 + e^{-ax}}$$

Figure 2.7 Binary Sigmoid Function

2.4 Hebbian

Hebbian Algorithm (Hebb 1994) is a type of forward / forward learning with an unsupervised learning paradigm. The learning signal r is equal to output 0 and initialization of weight w is done randomly with an initial value of 0.

$$r = f(w_i^t x)$$

Changes in weight w after the learning stage is equal to Δw_i ,

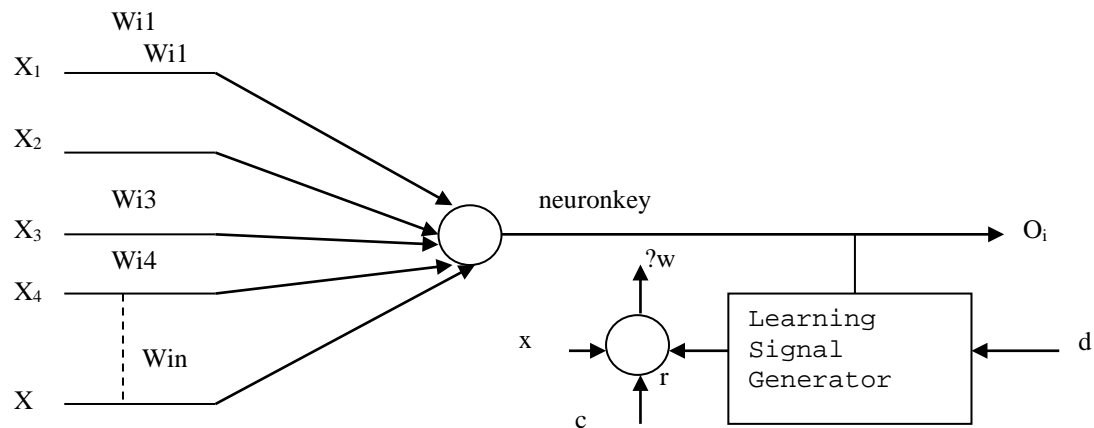
$$\Delta w_i = c f(w_i^t x) x$$

Example:

Simple artificial neural network that only consists of one neuron with the function of activation (net) in the form of steps:

$$O = \text{sgn}(\text{net}) = +1 ; \text{net} > 0$$

$$O = \text{sgn}(\text{net}) = -1 ; \text{net} < 0$$



Weakness of the McCulloch-Pitts model: determining line weights and analytic bias. For complex problems, this will be very difficult to do. In 1949, Donald Hebb introduced a method of calculating weights and biases iteratively. The Hebb model is the oldest model that uses supervision rules.

The basis of the Hebb algorithm is the fact that if two neurons which are simultaneously simultaneous (the same as the strength of the syntax increases. The reverse is not synchronous (one of the values of the strength of the synapses will weaken.

Because of that, in each iteration, the synaptic and biased weights are changed based on multiplication of neurons-2 on both sides. For a single screen network with 1 output unit where all input units x_i are connected directly to the output unit y , then the weight changes are made based on the equation: $w_i(\text{new}) = w_i(\text{old}) + x_i y$

Modeling steps in the hebb algorithm:

Step0 Initialize all weights:

$w_i = 0$ ($i = 1$ to n)

Step1 For each input-target pair

($s:t$),

do Steps 2-4:

Step2 Fill in the input with training data:

$x_i = s_i$ ($i = 1$ to n)

Step3 Fill in the output with the target data:

$y = t$

Step4 Weight modification:

$w_i(\text{new}) = w_i(\text{old}) + x_i y$ ($i=1$ to n)

Modified bias :

$b(\text{new}) = b(\text{old}) + y$

Or it can be simplified by:

$\mathbf{W}(\text{new}) = \mathbf{W}(\text{old}) + \Delta \mathbf{W}$

$\Delta \mathbf{W} = \mathbf{X} \mathbf{Y}$

$\Delta w_1 = x_1 t$ ($t=\text{target}$)

$\Delta w_2 = x_2 t$

3. Analysis

Table 3.1 Precession Data of PT. Juma Berlian Exim in March

NO	NIK	Name	Position	March			ALLOWANCES CRAFT
				Permission	Absent	Late	
1	160078012	ANITA	STAFF FINANCE	EVER	NEVER	OFTEN	GET
2	160078013	ASRI	CORPORATESALES	EVER	EVER	NOT OFTEN	NOT GET
3	160078014	AYU SILABAN	STAFF DOCUMEN	NEVER	EVER	OFTEN	NOT GET
4	160078015	BAMBANG GINTING	STAFFPURCHASING	EVER	EVER	OFTEN	NOT GET
5	160078016	DAVID ADITYA	STAFF DESIGN	EVER	EVER	NOT OFTEN	GET
6	160078017	DEWANTARA GINTING	STAFF BEACUKAI	EVER	EVER	NOT OFTEN	GET
7	160078018	EBENEZER	CORPORATESALES	EVER	EVER	NOT OFTEN	GET
8	160078019	ERWIN HAERIAN TO	MANAGER	NEVER	EVER	NOT OFTEN	NOT GET
9	160078020	FAHMI SARAGIH	RAZALI KABAG GUDANG	NEVER	EVER	NOT OFTEN	NOT GET
10	160078021	FITRI	STAFF DOCUMEN	NEVER	EVER	OFTEN	NOT GET
11	160078001	GENTA SINULINGGA	DRIVER	EVER	NEVER	NOT OFTEN	GET
12	160078002	HENDRA	STAFF OPERASIONAL	EVER	EVER	OFTEN	NOT GET
13	160078003	HEVY REHULINA	HRD	NEVER	EVER	OFTEN	NOT GET
14	160078004	INDRA PRATAMA	STAFF PERSONALIA	NEVER	EVER	OFTEN	NOT GET
15	160078005	LISMAYANI SIREGAR	STAFF HRD	NEVER	EVER	NOT OFTEN	NOT GET
16	160078006	LUSIANA SIMANJUNTAK	STAFF ADMINISTRASI	NEVER	NEVER	NOT OFTEN	NOT GET

The Heb training algorithm with the input vectors s_i and target units is as follows:

Initialize all weights, $w_i = 0$ ($i = 1, \dots, n$)

For all input vectors s and target units t , do the following:

a) Set activation of input unit $x_i = s_i$ ($i = 1, \dots, n$)

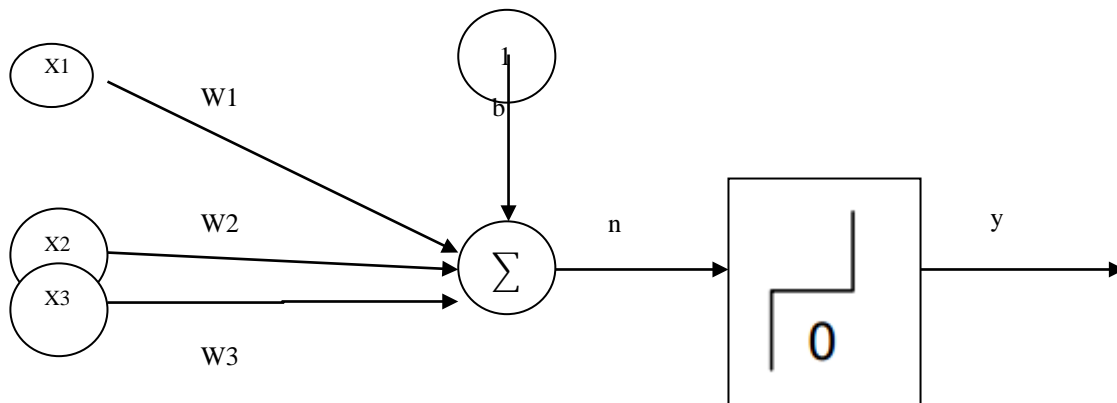
b) Set output activation $y = t$

c) Fix the weight according to the equation: $w_i (\text{new}) = w_i (\text{old}) + \Delta w$ ($i = 1, \dots, n$)
with $\Delta w = x_i y$

d) Fix bias according to equation $b (\text{new}) = b (\text{old}) + \Delta b$; with $\Delta b = 1 * t = 1 * y$

The basic notes of the Hebb algorithm are:

Improvements can be treated the same as weights Hebb network architecture is the same as McCulloch-Pitts network. Some input units are connected directly to an output unit, plus a bias.



If the target is in the form of binary data, then the input and target values are as shown in the following table:

Settlement:

Based on the data above, can be resolved with conversion data like the following:

The picture above is the Hebb network architecture to express the AND function. The following table is the input table and the target is represented binary (input and output are all worth 1 or 0. Based on the above data, the following pattern arrangement is obtained:

(X1) Permission	(X2) Absent	(X3) Late
1	1	1
1	1	-1
1	-1	1
1	-1	-1
-1	1	1
-1	1	-1
-1	-1	1
-1	-1	-1

Initially all weights and bias are given a value = 0. For each input and target data, changes in weights are calculated from the multiplication of input data and targets

$$\Delta w1 = x1 \cdot t; \Delta w2 = x2 \cdot t; \Delta w3 = x3 \cdot t; \Delta b = 1 \cdot t = t$$

Weight w_i (new) = w_i (old) + Δw_i ($i = 1,2,3$) The results of the weight iteration using the formula appear in the following table:

Input				Target	Weight changes				New Weight			
X1	X2	X3	B	t	$\Delta w1$	$\Delta w2$	$\Delta w3$	Δb	$\Delta w1$	$\Delta w2$	$\Delta w3$	Δb
Initialization									0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	-1	1	-1	-1	-1	1	-1	0	0	2	0
1	-1	1	1	-1	-1	1	-1	-1	-1	1	1	-1

1	-1	-1	1	-1	-1	1	1	-1	-2	2	2	-2
-1	1	1	1	-1	1	-1	-1	-1	-1	1	1	-3
-1	1	-1	1	-1	1	-1	1	-1	0	0	2	-4
-1	-1	1	1	-1	1	1	-1	-1	1	1	1	-5
-1	-1	-1	1	-1	1	1	1	-1	2	2	2	-6

So according to the interaction table above, the final network weight is

w1 = 2, w2 = 2, w3 = 2, dan b = -6

$$net = \sum_{i=1}^3 W_i X_i + b = 1 * X_1 + 1 * X_2 + 1 * X_3 + 1 = X_1 + X_2 + X_3 + -6$$

If tested on all input data, the results will be obtained as shown in the following table.

X1	X2	X3	Target (X1*W1) + (Xn*Wn)... +b					$f_{net} \begin{cases} t \geq 0, then 1 \\ < 0, then -1 \end{cases}$
			W1	W2	W3	b		
1	1	1	2	2	2	-6	0	1
1	1	-1	2	2	2	-6	-4	-1
1	-1	1	2	2	2	-6	-4	-1
1	-1	-1	2	2	2	-6	-8	-1
-1	1	1	2	2	2	-6	-4	-1
-1	1	-1	2	2	2	-6	-8	-1
-1	-1	1	2	2	2	-6	-8	-1
-1	-1	-1	2	2	2	-6	-12	-1

It appears that f (net) is the same as the target referred to as the AND function.

MEAN: the network can understand the intended pattern. Then, if the target result 1 (receives allowances) and -1 (not receiving allowances)

4. References

- [1].Puspitaningrum, Diah. 2006. Introduction to Artificial Neural Networks (Yogyakarta:Andi)
- [2].Budiharto, Widodo. Suhartono, Derwin. 2014. Artificial Intelligence Concept and Application (Yogyakarta: Andi)
- [3].Muis ,Saludin. 2017. Artificial Neural Network Techniques (Yogyakarta: Graha Ilmu)
- [4].Simamora, Rizka Yana. Huzaeni.Rizka , Muhammad. 2016. Detection of Hull Disturbance Through the Image of Iris Eyes Using Neural Network Method Imitation Hebb Rule (Banda Aceh:Jurnal Infomedia)
- [5].Krisnawan Komang ,Gafar. Widadga Agung, Gusti. Ngurah suarbawa, Komang. 2014. Human Voice Recognition Using the Hebb Artificial Neural Network Method (Bandung:Udayanan University)