CHAPTER 4
ANALYSIS AND DESIGN

4.1 Analysis

The problem in this project is the insurance company can't determine how much they will give to a person that have submitted an insurance claim.They can make an insurance program that will not only help reassure the people to join their insurance program but also make the company get profit from it if cost can be predicted beforehand. This program will make an insurance company can predict how much people spend on medical cost and how much they need to cover from an small injury to big injury.

Table 4.1: Analysis Data Table

Age	Gender	Body Mass Index	Smoker	Medical Cost
18	male	28.31	no	11272
18	male	39.14	no	12890
18	male	26.125	no	1709
18	female	31.92	no	2206
18	female	36.85	no	1630
19	female	34.7	yes	36398
19	female	20.6	no	1732
19	male	34.1	no	1261
19	male	29.07	yes	17353
19	male	20.9	no	1832
19	male	34.8	yes	34780
20	male	33	no	1980
20	female	21.8	yes	20167
20	female	29	no	2457
21	female	25.8	no	2008
22	female	28.05	no	2156
22	male	25.175	no	2046
23	female	36.67	yes	38512
24	male	35.86	no	1987
24	male	28.5	no	3538
25	male	25.74	no	2138

The data shown in the table above is a portion of training data that have been sorted. Can be seen from the table above even if the age is the same their medical cost differ from one to another. It means the cost can't be determined by only one factor. For such data it need a method that can calculate the complex requirement for the result. Complex requirement means they need more than one factor to calculate the medical cost. To calculate the medical cost each factor have a certain weight. The weight is calculated by a neural network that can determine the weight of a factor that affect the medical cost. Backpropagation is a neural network and suitable one to be used as an algorithm to calculate weight of each factor. The data itself can't be calculated by the program as is. The data need to be normalized to numeric for it can be calculated. The data is normalized using min max normalization.

norm = (value – min) / (max – min)

value = The value to be normalized

norm = The value after normalized

min = Minimum value of a given attribute

max = Maximum value of a given attribute

The data will be calculated in the program using backpropagation algorithm after the data has been read and normalized. Backpropagation itself had a number of step. The backpropagation algorithm process is as written below :

1. Generate random value in certain range for all weight

   w_{ij} = Weight for input index i to hidden layers index j

   w_{0j} = Weight for input bias to hidden layers index j

   u_j = Weight for hidden layer index j to output

   u_0 = Weight for hidden layer bias to output

2. Calculate the neural activation for hidden layer (Forward Propagation)

   $h\_net\ j = w_{0j} + \sum x_i . w_{ij}$

   $h_j = 1 / (1 + e^{-h\_net\ j})$

   h_net j = neural activation value for hidden layer before transfered

   h_j = neural activation value for hidden layer

3. Calculate the neural activation for output layer (Forward Propagation)

   $y\_net = u_0 + \sum h_j . u_j$

   $y = 1 / (1 + e^{-y\_net})$

   y_net = neural activation value for output layer before transfered

   y = neural activation value for output layer

4. Calculate Δ weight for hidden layers to output

   $\delta_1 = (t - y).y.(1 - y)$

   $\Delta u_j = \alpha . \delta_k . h_j$

   t = target output

   Δu_j = delta weight hidden layer index j to output index k

5. Calculate Δ weight for input to hidden layers

   $\delta\_net = \delta_k . u_j$

   $\delta_j = \delta\_net . h_j . (1 - h_j)$

   $\Delta w_{ij} = \alpha . \delta_1 . xi$

   Δw_{ij} = delta weight input index i to hidden layer index j

6. Update weight

   $w_{ij} = w_{ij} + \Delta w_{ij}$

   $u_j = u_j + \Delta u_j$

7. Repeat step 2 – 6 until max iteration has been reached or loss < target error.

   loss $= (t - k)^2$

   Testing only use (Forward Propagation or Feed Forward) step 2 and 3 is used.

4.2 Desain

This program can be seen from the following diagram :

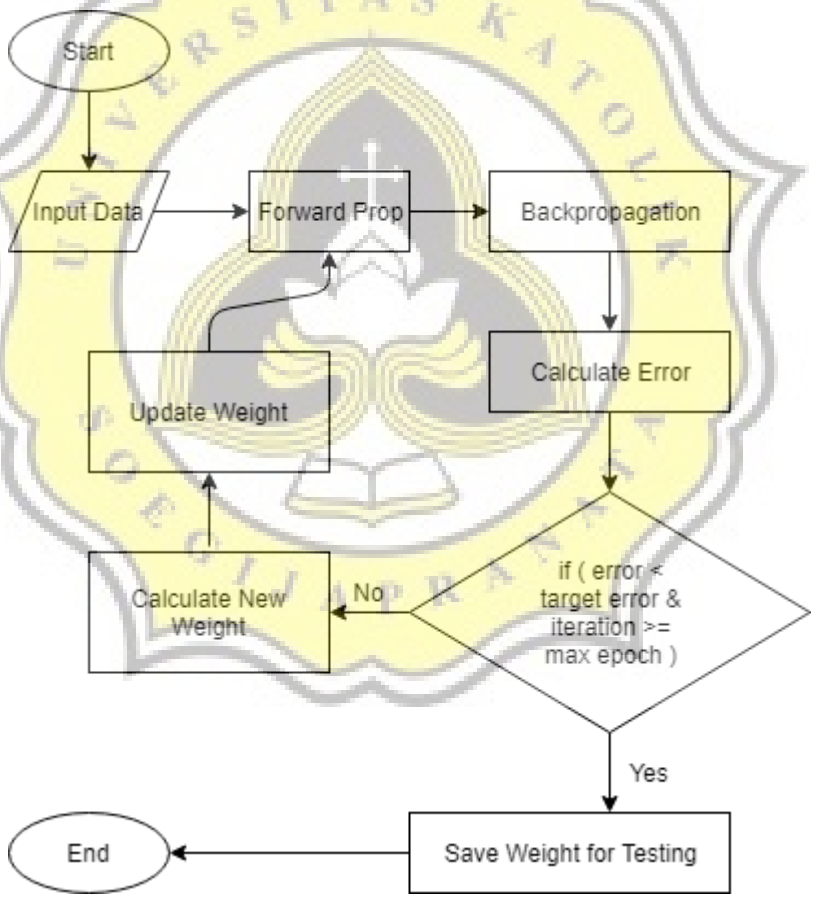

Illustration 4.1: Backpropagation Flowchart

The program start with the user inputing the data. The data is csv file, hidden layers, learning rate, iteration, and target error. The algorithm begin after the data has been inputted. This algorithm start with training the neural network with feed forwarding the input to a hidden layer using the weight and then to the

ouput. After the output has been calculated, it goes through the backpropagation. In the backpropagation it will calculate the loss using the output and the target, if the loss is higher than the target error it will calculate a new weight and update it. The method repeat in forward propagation and backpropagation until the loss is smaller than the target error or the max iteration has been reached. The testing will use the weight from the training and only use forward propagation to calculate the output. The whole process of the program run in the background. These process generated an output of csv for the prediction. The output of the prediction has been denormalized to make it easier to analyze.