La Minh Hoang
Promoter: dr hab. Zbigniew Świtalski, prof. UZ

# Artificial Neural Network and Application in Multi-Criteria Decision Analysis

University of Zielona Gora

Faculty of Mathematics, Computer Science and Econometrics

Poland

# Contents

## Introduction

Multi-criteria Decision Analysis (MCDA) which also can be call as "Multiple Criteria Decision-Making" or "Multi-criteria Decision Aiding" in English or "Analyse Multicritère" in French is an exciting field which has seen many achievements including increasing number of research papers, books, and theoretical over the years since the 60s (Pugnaire, 1992; Roy, 2005).

For many decision makers, MCDA is an excellent tool in getting comprehensive insight information when solving a decision problem especially when there are many different factors are taken into account (Pugnaire, 1992; Roy, 2005).

In this thesis, we will show to the readers how to incorporate Artificial Neural Network (ANN) in MCDA. Thus, to prove the feasibility of this incorporation; first, we will point out some difficulties which MCDA has when dealing with complicated decision problems, especially problems that repeat many times. Then we will see what advantages of ANN can offer to solve these difficulties.

To make more sense to the readers, the contents of this thesis will be structured as follow:

In the section §1, readers will be greeted with basic concepts of MCDA and what we should expect from MCDA when using it to solve a decision problem.

Next, in section §2, we will show the concept of Analytic Hierarchy Process (AHP) which is an MCDA method to derive priority or ratio scales from paired comparisons.

For the section §3, we will focus on the detail of ANN.

Then, in section §4, we will design a model in which Artificial Neural Network will play a major role in the MCDA process. We will also create a case study where we use real data to show how effective Artificial Neural Network can be in learning the decision making process.

Finally, we will conclude the thesis with some conclusions which summarizes the points of this thesis.

# 1. Multi-criteria Decision Analysis

To begin this chapter, we will examine the foundation theory of MCDA which is the Decision Theory in section 1.1.

Next, we will have a look at the structure of a typical decision-making process and how it operates in section 1.2 and section 1.3.

Then we will explain the reasons why people use MCDA for aiding the decision making process in section 1.4 and section 1.5.

After having a brief description of Decision Theory, decision-making process and why we need MCDA, we will delve deep into the philosophy and paradigms of MCDA in section 1.6.

We will also list some interest MCDA methods in subsection section 1.7.

## 1.1. The foundation of MCDA: Decision Theory

Decision Theory, in general, is simply a theory about making decisions. However, it is not an easy task to precisely define what Decision Theory is. It is because Decision Theory is truly an interdisciplinary subject which is pursued by researchers from many disciplines such as economists, statisticians, psychologist, political and social scientists or philosophers (Wikipedia, 2017d). Each discipline has its way of studying and theorising Decision Theory, for example, a psychologist might want to investigate the behavior of people when they are making decisions, an economist is likely to study the payoffs of each decision or a political scientist will try to find the optimal voting rule by studying Decision Theory. However, if we abstract Decision Theory from all disciplines, we can see that Decision Theory is all about theorizing human activities in situations where there are a goal and options to choose between in order to reach that goal, in other words, decision-making process.

We can generalize the Decision Theory into two definitions: one is broad, and the other one is narrow (Świtalski, 2016):

**Broad definition**: Decision Theory is the set of all possible disciplines connected with making decisions.

**Narrow definition**: Decision Theory is the set of theoretical consideration of what is a decision, what is decision situation, what are elements of a decision situation, what is right or optimal decision and how to make a right or optimal decision.

It is also important to know that Decision Theory has two main branches: Normative Decision Theory and Descriptive Decision Theory. The brief definitions of those two branches can be given as follow:

**Normative Decision Theory** concerns about constructing rules and methods of making decisions that in the context of an ideal decision maker (i.e. decision maker with perfect information) can identify the best decision.

**Descriptive Decision Theory** interests in the psychological or behavioral aspect of the decision maker (i.e. how do people in real situations make their decisions).

Also, readers should notice the direction of the thesis in term of which branch of Decision Theory we will go from here. Because the goal of this study is to introduce a new approach, in this case, Artificial Neural Network which will be used to enhance the process of making a decision; therefore, the majority of this study will be in the context of Normative Decision Theory.

Next, in section 1.2 we will examine the structure of a typical decision-making process.

### 1.2. Structure of a decision-making process

Typically, a decision-making process comprises of these elements (Świtalski, 2016):

**Decision makers**: The decision makers themselves, they are the ones who are responsible for the consequences of their decisions.

**Goal**: The desired state of our decision problem, usually it is what we want to achieve after solving the decision problem.

**Initial state**: The beginning state of our decision problem i.e. all the measurements, parameters, conditions or factors we have before starting the decision-making process.

**Decisions**: The options or alternative courses of action which are the possibilities of realizing our goal or desired state.

**Criterion**: A measurement or a standard which allows comparing between different options or alternatives. The criterion can be profit, cost, time, effort, property, measurable index, measurable attribute or measurable characteristic.

**Constraints**: Limitations stated by the decision problem. It can also be used to decide the polarity of a criterion, i.e. to decide if a criterion is a positive criterion or negative criterion. For example, if a decision problem has time constraint then evaluation from criteria related to time such as time to manufacture, time to delivery, response time or time to travel should be as small as possible. In other words, those criteria are negative criteria, i.e. the smaller the value, the better the evaluation.

**Consequences**: Outcomes or results we get after realizing the goal with the option or alternative we have chosen.

**Uncertainty**: The set of all consequences that are outside the set of expected consequences of the goal.

**Preferences**: The view of each decision maker which determines the importance of each criterion. For example, in a decision problem where the time and cost are criteria, a decision maker may think time is the most important criterion while the other decision maker may think the opposite, cost is the most important criterion. Therefore, an alternative with high evaluation from time criterion may be appealing to decision makers who value time criterion while at the same time being ignored by decision makers who value cost criterion. This element is the most important one in a decision situation because it directly affects the performance of an alternative in different decision maker view.

### 1.3. The process of making a decision

The relationship between elements in a decision-making process is presented in figure 1.1 (Świtalski, 2016):

In figure 1.1, all the elements of decision-making process are drawn as members of an analysis process. As we can see, the process starts first by formulating the Initial State of the decision problem, so then we can find the Constraints of the decision problem, determine the Criteria and construct the Alternatives. Then this information is given to Decision Makers.

After formulating the information from the Initial State, each Decision Maker with his or her Preferences takes part in an Evaluating Process along with the given Alternatives, Criteria and Constraints. During the Evaluating Process, one alternative or combination of alternatives which is considered the best comparing to other alternatives is chosen to realize the goal of the decision situation. Finally, after realizing the goal using the selected alternative, we get the outcomes from the realization.

These outcomes belong to the Uncertainty set because no matter how careful the Evaluating Process is executed by Decision Makers, there are always some

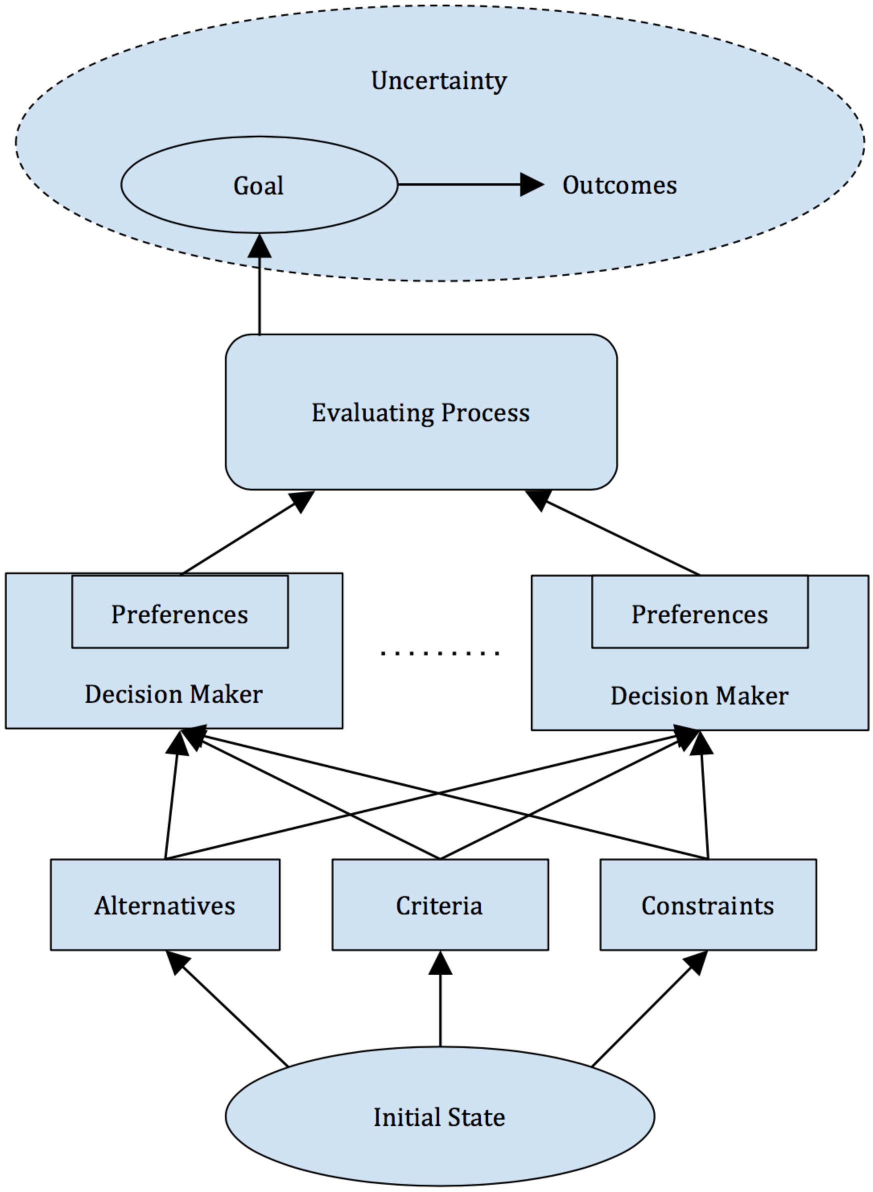


Figure 1.1. Elements of a decision-making process

uncontrollable and unforeseeable factors that could affect the outcomes. In other words, we can make a good decision by thorough deliberation in the process of Evaluating Process and take into account all possible information from Initial State, but because of the Uncertainty, we cannot know for sure if that decision is the right decision or not.

In section 1.4, we will present to the readers the concept of Decision Analysis and why it is important to have Decision Analysis in every decision-making process.

### 1.4. The necessary of Decision Analysis

In section 1.1, section 1.2, and section 1.3, we learnt that in a decision-making process, decision makers have the responsibility to choose the right decision in order to realize the goal in that particular situation. However, there are many factors with hidden information in a decision-making process that can hide the true value of each decision thus affect the judgement of the decision maker.

For example, when choosing suppliers for providing materials to build a particular product. If we do not spend the time to research the past performance of each supplier, we will never know the true value of each supplier in term of providing the needed materials. Thus our judgment for choosing the suppliers will not be certain. To overcome this difficulty of uncertain judgment, a decision maker will need to practice Decision Analysis in a decision-making process.

Decision Analysis or Decision Aiding (DA) can be defined as follows (Roy, 2005): It is the activity of using explicit and formalized models to help obtain elements of responses to the questions posed by a stakeholder in a decision process. In other words, it is the process of getting more information from the context of the decision-making process to clarify various components inside a decision problem.

With the above definition, DA's purpose is to establish the decision-making process with working hypotheses, formulations of propositions (satisfying solutions or possible compromises) which clears out the uncertainty in the decision maker's judgement (Roy, 2005) . In such case, DA can contribute to the follows:

— Analyzing the context of the decision-making process by various means such as identifying the actors, the possibilities of action, their consequences, or the stakes.
— Organizing and structuring how the decision-making process should proceed to improve the consistency among the values underlying the goal and the final decision that realize that goal.
— Getting the actors (such as the decision maker and the expert who is providing the information related to the decision) to cooperate by proposing a way to improve mutual understanding and a framework that comfortable to debate.
— Elaborating recommendations by using the models or procedures which are formulated within the context of a working hypothesis.
— Aiding in the final decision legitimization.

In section 1.5, we will present to readers the notions of Mono-criteria and Multi-criteria, thus the reason why MCDA is the prefer tool for many decision makers.

### 1.5. From Mono-criteria to Multi-criteria

In a decision-making process, a decision maker has two approaches to evaluate the options. They are Mono-criteria and Multi-criteria. To simply put, Mono-criteria is when the decision maker only uses one criterion for determining the value of each option of the decision problem. Multi-criteria, on the other hand, evaluate each option on many different criteria that associating with various aspects of the decision problem. Readers can find more information about criterion in section 1.6.

When a decision maker is dealing with a DA process, it is very rare for the decision maker to have in mind only one single clear criterion (Mono-criteria). In a multi-actor environment, it is even more unlikely for that single criterion to satisfy the point of view of every actor in the decision-making process because each actor plays a different role in that process which gives different preference to the actor own objectives and value system (Roy, 2005).

For any case, it is essential to consider every point of view dealing with many perspectives of a decision-making process, for example, the financial status, the current human resources, the affected ecosystem, security or quality of services. By regarding each related point of view individually, independently from the others, it is likely to arrive at a clear decision that meets all actors' preferences and requirements (Roy, 2005).

Therefore, it is typical to associating a particular criterion to each related point of view (Multi-criteria). For each associated criterion, it will be used to evaluate any potential action on an appropriate qualitative or quantitative scale (Roy, 2005).

As we can see, in many decision-making contexts, using Mono-criteria approach might overlook certain aspects of the decision problem, thus affecting the judgement of the decision maker. Using Multi-criteria approach in such cases will help the decision maker to avoid the before-mentioned danger of neglecting.

Next in section 1.6, we will explore the three basic concepts of MCDA which will give us a throughout view of the structure of a Decision Analysis process that connects with Multi-criteria to help the decision maker in solving his or her decision problem.

### 1.6. Basic Concepts of MCDA

The following concepts have a fundamental role in the process of analyzing and structuring the decision-making process with MCDA:

**Alternatives** which defines the options that the decision maker has to decide to choose one as the way to realize the goal in a decision-making process.

**Criterion or Criteria** which acts as a tool to evaluate and compare between alternatives.

The subsections below will present to the readers the definitions of each concept so the readers can generalize what an MCDA process is.

#### 1.6.1. Alternatives

In a decision-making process, to make a decision, the decision maker have to choose an action with the purpose of realizing the goal of the decision problem by using the chosen action.

The action in this sense can be defined as a potential action which constitutes to the object of the decision or one of the objectives of the decision analysis process. However, not all action can be qualified as a potential action. The action must be considered to be possible to implement it or justifies some interests within the decision analysis process (Roy, 2005).

With the definition of potential action, the concept of alternative is often modeled in two ways (Roy, 2005):

**Mutual exclusive**: Two distinct potential actions must not be conjointly put into realizing the goal of the decision problem.

**Non-mutual exclusive**: Various potential actions can conjointly be put together to realize the goal of the decision problem.

Many authors believed that potential actions should be mutual exclusive, but it is not always the case (Roy, 2005). In many real world decision analysis contexts, it is appropriate to combine several potential actions into realizing the goal of the decision problem rather than using only one potential action.

In any case, for any decision analysis process, there is always a set of more than two potential actions or alternatives, and this set is not fixed (Roy, 2005). During the decision analysis process, this set can evolve i.e. adding more potential actions as a result of investigating every aspect of the decision problem. For example, the investigation can help gain more knowledge about the goal, constraints, the preferences of the stakeholders, the possible outcomes of the actions, and other aspects of the decision problem. Thus, this new knowledge could open new boundary that helps to identify other actions as potential action.

For the set of all potential actions or alternatives in a decision-making process, we will use $A$ to denote it. To designate a potential action or alternative, we use $a$. When the number of actions is finite ($|A| = m$) we have:

$$A = \{a_1, a_2, \ldots, a_m\}$$

For each potential action, when we can model the action by referring to some variable $x1, x2, x3, ...$ we can write:

$$a = (x_1, x_2, ...)$$

We will use the above notions to describe the definition of Criterion in the next section 1.6.2.

### 1.6.2. Criteria

The readers should recall from section 1.5 that a criterion is used as a tool to determine the value of an alternative. By using the notion of potential action $a$ in section 1.6.1, we can denote a criterion as a function $g$ which evaluates a potential action or alternative; this function then outputs a $performance$ value associating with the input potential action. We will denote this evaluation as $g(a)$(Roy, 2005).

Often, $g(a)$ is a real number. However, it is necessary to define a set $X_g$ of all possible evaluations that the criterion g can produce. This set $X_g$ should also follow a scale system to be accepted by all stakeholders in an MCDA process(Roy, 2005).

Each element $x \in X_g$ is called *degree* or *score* of the scale. We can use any number, verbal statement or pictogram to designate a degree. During the comparison between two alternatives, we compare the two degrees which represent the respective performances of the two alternatives according to criterion $g$.

There are many types of scales which readers should notice; the following two scales are the most typical (Roy, 2005):

**Purely ordinal scale or qualitative scale**: The gap between two degrees does not have a clear meaning i.e. we only know the ranking of the degrees, we do not know how close or far from one degree to another.

**Quantitative scale**: The degrees in this scale are defined by a clear quantity in a way that it gives meanings i.e. we can count the differences between two different degrees.

## 1.7. MCDA Methods

Decision Makers can use different methods to analyze the alternatives and find the best alternative or the group of best alternatives, some of these methods do not require criteria to have weights, while others do require (Świtalski, 2016).

For example, the following methods do not use weights:

**Pareto Rule**: We will choose from the given alternatives the set of alternatives which are non-dominated by any other alternatives.

**Conjunctive Method**: We will define levels of satisfaction for all criteria then for each criterion we find the set of alternatives which satisfies it.

After getting all the sets of satisfying alternatives for each criterion, we will intersect all those sets and the resulting set from the intersecting will be our chosen set of alternatives.

**Disjunctive Method**: Similar to Conjunctive Method but we will choose alternatives which are good for at least one criterion. In other words, instead of intersecting all the sets of satisfying alternatives for each criterion, we find the union of all those sets.

On the other hand, these following examples are methods which do require Criteria to have weights:

**Lexicographic Method**: We will order the criteria from the most important to the less important. The ordering will be based on the weights of the criteria. Then, we will choose alternatives which are the best with respect to the first ranking criterion base on the evaluation of the Decision Table. If there is only one such alternative, then we will stop the process. If there are more such alternatives (more than two), we will continue to choose from these alternatives the ones which are the best with respect to the second ranking criterion. We will continue the process unless we reach the last criterion in the ordering or there is only one alternative left.

**Weighted Average Method**: We will calculate the weighted average for each alternative by multiplying the Scaled Decision Table with Criteria weights. By comparing the scores and select the biggest one, we will have our best alternative.

The above methods are just simple methods, however, there are other methods that are complicated and can be used in complex decision problem (Wikipedia, 2017b):

**Analytic hierarchy process (AHP)**: a method to derive priority or ratio scales from paired comparisons.

**ELECTRE**: this method is a "outranking method" of decision making which is applied to three main problems: choosing, ranking, and sorting.

**PROMETHEE**: is also another "outranking method" but it can be used both in normative approach and descriptive approach.

In the next section §2, we will present to the readers a comprehensive description of AHP method.

## 2. The Method of Analytic Hierarchy Process

In this section, we will get into the details of Analytic Hierarchy Process (AHP) by answering questions such as: What is AHP? Who created it? Which kind of decision problem can we apply AHP?

After getting the general ideas about AHP, we will delve deep into the methodology and core elements of AHP such as: What is the Pair-wise comparison? How to make a Comparison Matrix? How to calculate priority vector? How to use Consistency Index and Consistency Ration to verify the comparisons?

Explaining AHP using only its definitions can be tough at some points. Therefore we will use an example to explain each aspect of AHP. It is easier to understand the concept of AHP that way.

### 2.1. Brief History of Analytic Hierarchy Process

Analytic Hierarchy Process (AHP) is created by Professor Thomas L. Saaty as a Multi-Criteria decision-making method. This method is described in the book "Mathematical Principles of Decision Making" or "Principia Mathematica Decernendi" which Prof. Saaty wrote in the 70s (Goepel, 2013).

In general, Analytic Hierarchy Process is a method to derive priority or ratio scales (we will use the word "priority" and "ratio" interchangeably) from paired comparisons (Saaty, 2008). The input of AHP method is obtained from actual measurements such as price, weight, area, or from subjective opinion such as satisfaction feelings or preference (Teknomo, 2006). To derive the ratio scales, we need to solve the principal Eigenvectors problem and then to verify the scales we will use Consistency Index which is calculated from Eigenvalue.

To solve a decision problem by using AHP, we need to decompose the decision problem into the following steps (Saaty, 1980):

1. Define the problem and determine the kind of knowledge sought – in other words, we set the goal of the decision problem.
2. Structure the hierarchy of the decision problem. The goal of the decision problem will be on the top level of the hierarchy. Then you will put the objectives or criteria (and sub-criteria) in the intermediate levels. Finally, at the lowest level will be the set of alternatives.
3. Construct a set of pairwise comparison matrices. In which, each element in an upper level is compared to the elements in the level immediately below to it.
4. Calculate the priorities from the comparison matrices in each level of the hierarchy until the last priorities of the alternatives in the lowest level of our hierarchy are obtained.

After obtaining priorities for the alternatives in the decision problem, we should calculate the Consistency Index and Consistency Ratio to see if our comparisons are logical and consistent. If not, we have to revise our comparisons to get a more consistent set of priorities.

If the priorities are consistent, we can use them to evaluate our alternatives and select the optimal alternative for our decision problem, i.e. the alternative with the highest priority.

Because AHP allows small inconsistency in making comparison (Teknomo, 2006), we can use AHP to solve various decision problems, from making simple and straightforward decisions to complex and high-stakes problems in which human perceptions and judgements are required (Wikipedia, 2017b), for example:

1. Choice
2. Ranking
3. Quality management
4. Conflict resolution
5. Etc.

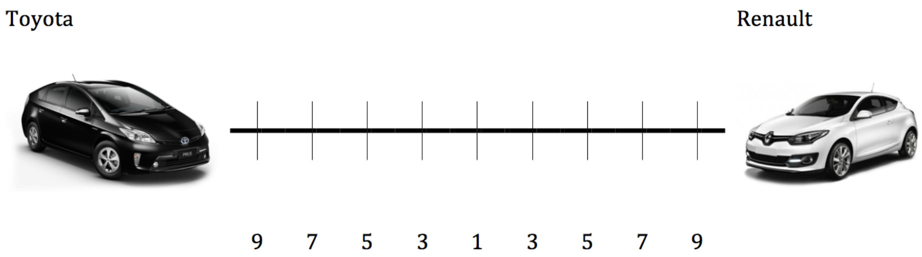


Figure 2.1. Relative scale between two car manufacturers

Table 1. Judgement table

| Preference Value | Description |
|---|---|
| 1 | Two cars are equal in comparison; no car is more preferred than the other. |
| 3 | One car is slightly preferred than the other. |
| 5 | One car is strongly preferred than the other. |
| 7 | One car is very strongly preferred than the other. |
| 9 | On car is absolutely preferred than the other. |
| 2,4,6,8 | Intermediate scales between two adjacent values. |

One fascinating example of AHP application is suspect identification by witnesses in criminal cases (Mu and Chung, 2013). Usually, candidates for identification are shown altogether or sequential, but in the AHP case, they are shown in pairs. Comparisons between pairs are then made by the witnesses. Studies have shown that by using AHP method, the reliability of identification has been increased from 55% to 83% and the false identification rate has been reduced from 20% to 17% (Mu and Chung, 2013). Also, the consistency index produced by AHP method is a good indicator for telling if the statements made by witnesses are consistent or not.

### 2.2. Pair-wise comparison

The most fundamental element of AHP method is the Pair-wise comparison. In short, it is like asking between two objects, which one do you prefer and how much do you prefer it to the other object.

Suppose we have two cars made by two different manufacturers, one is from Renault, and the other is from Toyota. To make a pair-wise comparison between these two cars, we will use a relative scale in figure 2.1 to measure how much we prefer a car compare to the other (Saaty, 2008).

In the above relative scale, if we prefer the Toyota car than the Renault car, we choose a number which is a preference value between one and nine on the left side of the scale. If we prefer the Renault car otherwise, we choose a number on the right side of the scale.

The preference values or judgement values are used to describe how strong our preference is over a car. The following table 1 explains the judgement values (Saaty, 2008)

For example, if we firmly prefer the Toyota car than the Renault car, we then mark the scale like in figure 2.2.

Now, if we add one more car from another manufacturer, our pair-wise comparison will become like in figure 2.3.

As we can see, the number of pair-wise comparisons has been increased from one to three comparisons; it is a combination of the number of objects to be compared. We can calculate the number of comparisons using table 2 (Teknomo, 2006).

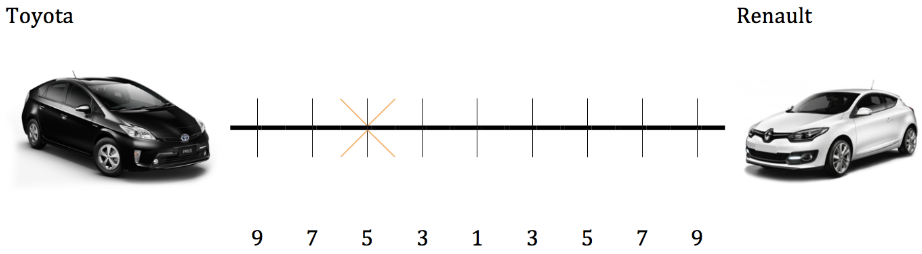


Figure 2.2. Toyota is being preferred on relative scale

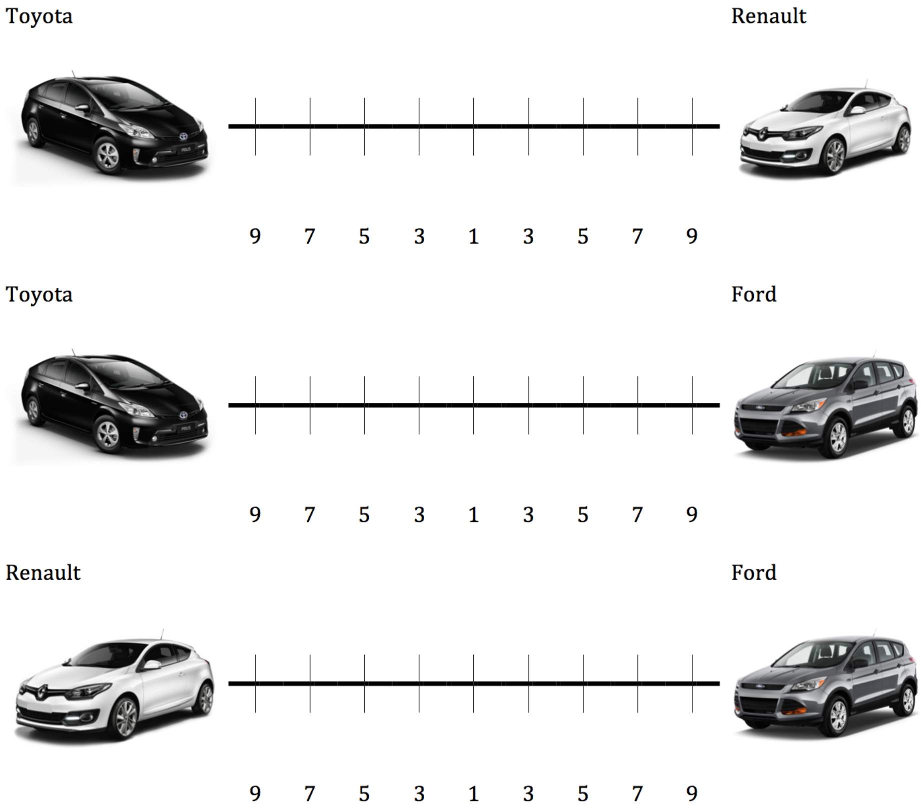


Figure 2.3. Comparisons between three car manufacturers

Table 2. Number of comparisons per number of objects

| Number of objects | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | n |
|---|---|---|---|---|---|---|---|---|---|
| Number of comparisons | 0 | 1 | 3 | 6 | 10 | 15 | 21 | 28 | $\frac{n(n-1)}{2}$ |

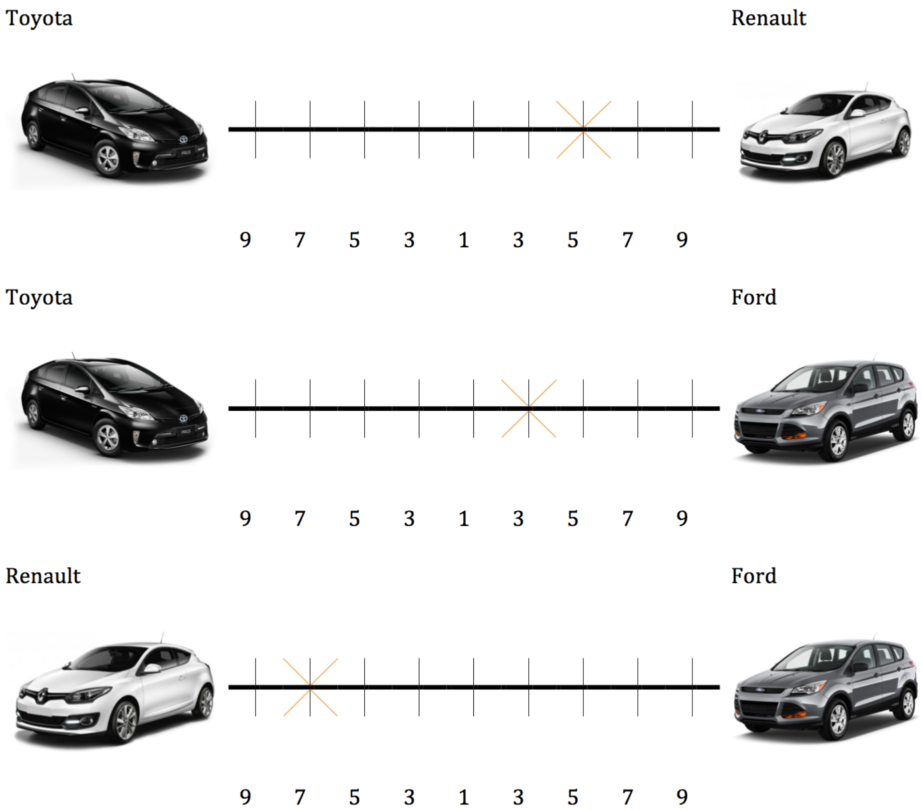

Figure 2.4. Darek's pair-wise comparisons

From table 2, it suggests that if we compare too many objects (30 cars for example), the number of comparisons will become immensely significant (435 comparisons for 30 cars). With a large number of comparison, it will confuse the people who make the comparison, and as a result, the likelihood of inconsistent comparisons will be high.

Next in section 2.3, as we have made our pair-wise comparisons between objects, we will aggregate them into a Comparison Matrix.

### 2.3. Comparison Matrix

We will use the comparison between three cars in the previous section as the example of how to make a comparison matrix from pair-wise comparisons.

In our example, we have a middle-aged man named Darek. Darek wants to buy a new car, but he does not know which car he should buy because there are just too many cars to choose. Thankfully, a friend who works in automobile industry recommends Darek three car models from three different manufacturers. These three cars are currently the best, but Darek can only choose one. Therefore he makes the comparisons in figure 2.4 based on his subjective judgement.

Because we have three pair-wise comparisons, in this case, our comparison matrix will be a three by three matrix. In this matrix, the values on the diagonal line will be one because it is the value of a comparison of two same objects. We only have to fill up the upper triangular matrix, for the lower triangular matrix, we can use the reciprocal values of the upper triangular matrix. To fill up the comparison value, we use the following rules (Teknomo, 2006):

1. If the preference value is on the **left side** of 1, we put the **actual preference value**
2. If the preference value is on the **right side** of 1, we put the **reciprocal value**.

Table 3. Comparison matrix for Darek's pair-wise comparisons

| | Toyota | Renault | Ford |
|---|---|---|---|
| Toyota | 1 | $\frac{1}{5}$ | $\frac{1}{3}$ |
| Renault | | 1 | 7 |
| Ford | | | 1 |

Table 4. Complete comparison matrix

| | Toyota | Renault | Ford |
|---|---|---|---|
| Toyota | 1 | $\frac{1}{5}$ | $\frac{1}{3}$ |
| Renault | 5 | 1 | 7 |
| Ford | 3 | $\frac{1}{7}$ | 1 |

In our case, because Darek strongly prefers Renault than Toyota, we put $\frac{1}{5}$ in row 1 and column 2 of the matrix. Next, Darek slightly prefers Ford than Toyota; we put $\frac{1}{3}$ in row 1 and column 3 of the matrix. Finally, between Renault and Ford, Darek very strongly prefers Renault then Ford, therefore we put 7 in row 2 and column 3 of the matrix. The comparison matrix is presented in table 3.

We only have set the value for the upper triangular matrix, for the lower triangular matrix we will use the reciprocal values of the upper triangular matrix based on the following rule (Teknomo, 2006): If $a_{ij}$ is the element of row $i$ column $j$ of the matrix, then the lower diagonal is filled using this equation:

$$a_{ij} = \frac{1}{a_{ij}} \tag{2.1}$$

.

Using the rule above we now have the complete comparison matrix in table 4.

Please be noticed that all values in the comparison matrix must be positive, or $a_{ij} > 0$.

In section 2.4, we will use this comparison matrix to calculate the eigenvalues and eigenvectors to calculate the final priorities or Priority Vector of our alternatives which are the three car models in Darek example.

### 2.4. Priority Vector

With the complete comparison matrix in the previous section, we can now find the priority vector for our alternatives. However, first, we need to calculate the eigenvalues and eigenvectors of the matrix because the priority vector is the normalized principal Eigenvector of the matrix.

Usually, eigenvalues and eigenvectors of a matrix are calculated by using computer programs such as MATLAB or Maxima due to the increasing complexity in calculation when the size of the matrix is getting bigger. However, for small matrices with size $n \leq 3$ like the one in our example, it is possible to use an approximation method to approximate the normalized principal Eigenvector with low error rate (Teknomo, 2006).

The approximation method is very easy to use. All we need to do is just to normalize each column in the comparison matrix. For example, we got table 4 from section 2.3, then we sum each column of table 4 to get the following table 5.

To get the approximation of the normalized principal Eigenvector we just need to average the matrix across the rows as follow:

$$v = \frac{1}{3}\begin{bmatrix} \frac{1}{9} + \frac{7}{47} + \frac{1}{25} \\ \frac{5}{9} + \frac{35}{47} + \frac{21}{25} \\ \frac{1}{3} + \frac{5}{47} + \frac{3}{25} \end{bmatrix} = \begin{bmatrix} 0.1000 \\ 0.7134 \\ 0.1866 \end{bmatrix} \tag{2.2}$$

Now we have got the priority vector or the approximation of the normalized principal Eigenvector of the comparison matrix. Since the sum of all elements

Table 5. Comparison matrix summed by column

| | Toyota | Renault | Ford |
|---|---|---|---|
| Toyota | 1 | $\frac{1}{5}$ | $\frac{1}{3}$ |
| Renault | 5 | 1 | 7 |
| Ford | 3 | $\frac{1}{7}$ | 1 |
| Sum | 9 | $\frac{47}{35}$ | $\frac{25}{3}$ |

in the priority vector is 1, it tells us the relative weights of all alternatives we have compared.

By having the priority vector, we can say the weight or the priority of each car model in our example:

— The Toyota car is 10%
— The Renault car is 71.34%
— The Ford car is 18.66%

According to these weights, the Renault car has the highest priority (71.34%) among the three car models. Therefore, Darek can easily choose the Renault car as his new car.

However, we need to verify if the comparisons made by Darek is consistent or not. To do this, we need to calculate the Principal Eigenvalue. It can be obtained in our approximation by the summation of products between each element of the priority vector and the sum of columns of the comparison matrix (Teknomo, 2006):

$$\lambda_{max} = 9 * (0.1) + \frac{47}{35} * (0.7134) + \frac{25}{3} * (0.1866) = 3.4129 \tag{2.3}$$

Using this Principal Eigenvalue, we can verify the consistency of our pair-wise comparisons by calculating the Consistency Index and Consistency Ratio which will be shown in section 2.5.

## 2.5. Consistency Index and Consistency Ratio

The consistency of pair-wise comparison is closely related to the transitive property (Teknomo, 2006). Which means for a person who prefers A than B or $A \succ B$ and he also prefers B than C or $B \succ C$ then according to transitive property, this person should prefer A than C or $A \succ C$. If for any reason, this particular person does not prefer A than C, but he prefers C than A or $C \succ A$, then we can say that his comparisons over A, B, and C are inconsistent with each other.

Let's check this consistency in our example insection 2.3. Based on the three comparisons in figure 2.4, we can see that:

— Darek prefers the Renault car more than the Ford car or $Renault \succ Ford$.
— Darek prefers the Ford car more than the Toyota car or $Ford \succ Toyota$.
— Darek prefers the Renault car more than the Toyota car or $Renault \succ Toyota$.

In this case, the comparisons made by Darek appear to be consistent:

$$Renault \succ Ford \wedge Ford \succ Toyota \Longrightarrow Renault \succ Toyota \tag{2.4}$$

(satisfies the transitive property)

If Darek had made the last comparison as follow: $Toyota \succ Renault$, then this comparison will be inconsistent with other comparisons thus making Darek judgement in choosing new car inconsistent and not reliable.

However, there is a problem with this check. This check does not take into account of how much Darek prefer each car. Therefore, we cannot know if the

Table 6. Pre-calculated Random Consistency Index

| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| RI | 0 | 0 | 0.58 | 0.9 | 1.12 | 1.24 | 1.32 | 1.41 | 1.45 | 1.49 |

values in the comparison matrix are consistent with each other or not. This fact can make the whole hierarchy of AHP (especially multi-level hierarchy) becomes inconsistent because we have to make a pair-wise comparison in each level of the hierarchy and the immediately lower level will use the priority vector produced by pair-wise comparison of the base level to calculate its global weights (Saaty, 2008).

According to Prof. Saaty, for a consistent comparison matrix to be consistent, the largest Eigenvalue should be equal to the size of the comparison matrix, or $\lambda_{max} = n$ (Saaty, 2008). Then, to know how many degrees of consistent a comparison matrix is, Prof. Saaty gave us a measurement of consistency, the Consistency Index or CI. To calculate this index, we use the following formula:

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{2.5}$$

Therefore, applying into our example which has $\lambda_{max} = 3.4129$ and the size of the comparison matrix is $n = 3$; we have the consistency index as follow:

$$CI = \frac{\lambda_{max} - n}{n - 1} = \frac{3.4129 - 3}{2} = 0.20645 \tag{2.6}$$

Now, how can we use this index to determine the consistency of our comparison? Once again, according to Prof. Saaty, by comparing it with the appropriate one we can know how many degrees of consistent our comparison matrix is. The proper consistency index is called Random Consistency Index or RI (Saaty, 1980).

Prof. Saaty had pre-calculated the Random Consistency Index for comparison matrix using a 1-9 scale based on a sample size of 500 random generated reciprocal matrices in table 6 (Saaty, 1980):

To compare the Consistency Index with the corresponding Random Consistency Index, Prof. Saaty proposed the Consistency Ratio which is calculated using the following formula (Saaty, 1980):

$$CR = \frac{CI}{RI} \tag{2.7}$$

If the value of Consistency Ratio is smaller or equal to 10%, then the inconsistency in the comparison matrix is acceptable. However, if the Consistency Ratio is greater than 10%, we need to revise our judgement in the pair-wise comparison process.

In our example, we have: $CI = 0.20645$ and $RI$ for $n = 3$ is 0.58, then we have the Consistency Ratio as follow:

$$CR = \frac{CI}{RI} = \frac{0.20645}{0.58} = 0.3559 \tag{2.8}$$

We have $CR = 35\% > 10\%$. Therefore we can see that Darek's subjective judgement in the pair-wise comparison process is not consistent. Thus, we recommend a revision for Darek's pair-wise comparison between three car models to improve the consistency.

# 3. Artificial Neural Network

When using Multi-Criteria Decision Analysis, especially AHP method, to deal with decision problems which have many criteria or alternatives, the cost of computation to find the best possible decision could increase exponentially. Also, the task in which a decision maker have to make the best possible decision as an output from a system where it takes existing information, and future predictions as the input is a challenging task (Golmohammadi, 2011).

However, there is an interesting approach to approximate the best possible decision from such system without paying too much computation cost; it is by using Artificial Neural Network (ANN).

ANN is an alternative approach to computing which emulates the remarkable ability of the human mind in reasoning and learning in an environment of uncertainty and imprecision. In fact, ANN simulates the human brain and its ability to learn, recall, and generalize from training data by modeling the essence of the human brain: the networks of biological neurons (Mashrei, 2012). These features make ANN a powerful data modeling tool that is capable of capturing and representing the complex relationships between input and output data (Golmohammadi, 2011).

With such properties, ANN has found its place in many applications such as function approximation, regression analysis, classification, data processing, robotics, and control engineering (Wikipedia, 2017c).

Therefore, in this section, we will get into the details of ANN, what inspires the model of ANN, what is the fundamental element of ANN, the architecture of ANN, how ANN can learn, and how to train ANN.

## 3.1. The Biological Neuron

In a sense, artificial neural network (ANN) simulate the function of the human brain. It is a type of network which nodes are artificial neurons. These artificial neurons are based on the mathematic model of biological neurons.

Biologically, a human brain is a network of about one hundred billion neurons, and each neuron connects to about ten thousand other neurons by using dendrites, and axons. Every single neuron receives electrochemical input signals from other neurons at the dendrites. Then, the cell body of the neuron will sum all the strength of all the electrochemical input signals which it receives from other neurons, and if the sum is greater than some threshold level, the neuron will be activated. When the neuron is activated, it will transmit or fire an electrochemical signal along the axon and pass this signal to the dendrites of other neurons that are attached to this neuron. It is important to know that a neuron does not fire any signal if the sum of electrochemical inputs does not surpass a particular level. In other words, a neuron can only do two actions: fire a signal or not, there is no other action in between (Clabaugh et al., 2000). The structure of a biological neuron is shown in figure 3.1.

Hence, the entire human brain is a huge network of interconnected electrochemical transmitting neurons. This property gives the human brain the power to perform exceptionally complex tasks. Because of this power, the human brain has become the model for the ANN to be based on in order to solve problems which are simple tasks for an ordinary human but challenging for a conventional computer. Some notable examples of these problems are image recognition and predictions based on past knowledge (Clabaugh et al., 2000).

## 3.2. The Artificial Neuron

To model ANN from the human brain, first, we have to model the most fundamental element, the biological neuron. Similar to the natural counterpart, an artificial neuron is a simple processing unit which consists of numerical input values (the receiving electrochemical input signals), which are multiplied by weights (the strength of the respective electrochemical input signals), and these

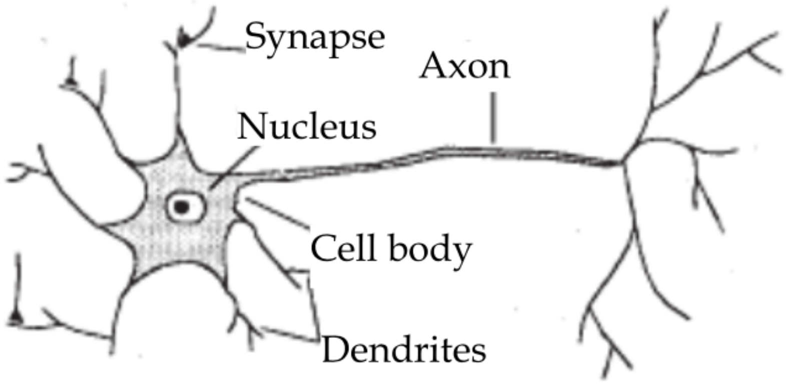


Figure 3.1. A Biological Neuron (Mashrei, 2012)

inputs will then be totaled by a sum function. Finally, an activation function will calculate the output of the artificial neuron by using the result of the sum function (Mashrei, 2012). A typical artificial neuron is shown in figure 3.2.

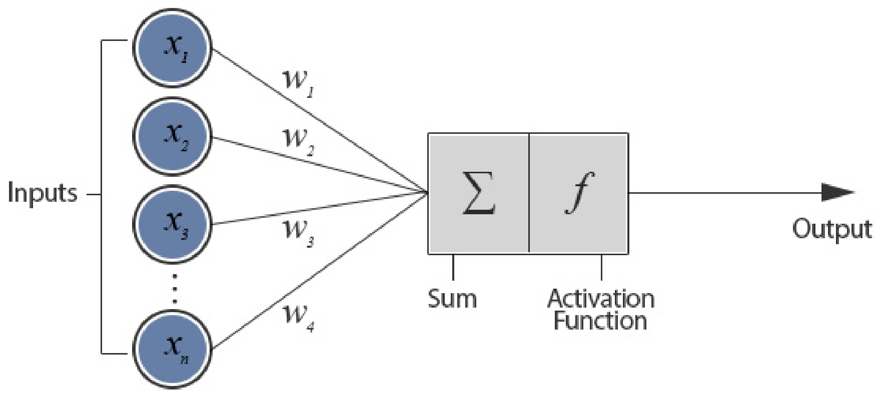


Figure 3.2. An artificial neuron (Jacobson, 2013)

In figure 3.2, we can see that an artificial neuron has many inputs $x_1, x_2, x_3, ..., x_n$ and each input is independently weighted $w_1, w_2, w_3, ..., w_n$ when we calculate the sum of all input signals. These weights have the responsibility to amplify or weaken the original input signals. For example, if the input $x_1$ has the initial value of 1 and the weight $w_1$ corresponding to $x_1$ has the value of 0.5, then the value we put into the sum function will be 0.5 because $x_1 w_1 = 1 * 0.5 = 0.5$. After all weighted input signals have been added together, the result will be passed into the activation function which will determine the output of the artificial neuron. There are several types of activation function such as step function, pure-linear function, log sigmoid function, or tangent sigmoid function (Wikipedia, 2017a). The table 7 compares the properties of those activation functions. The plots of those functions will be shown in figure 3.3, figure 3.4, figure 3.5 and figure 3.6.

Table 7. Activation functions (Wikipedia, 2017a)

| Name | Equation | Output range |
|---|---|---|
| Pure-linear (Identity) | $f(x) = x$ | $(-\infty, \infty)$ |
| Step | $f(x) = \begin{cases} 0 & x < T \\ 1 & x \geq T \end{cases}$ | $\{0, 1\}$ |
| Log sigmoid (Logistic) | $f(x) = \frac{1}{1+e^{-x}}$ | $(0, 1)$ |
| Tangent | $f(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$ | $(-1, 1)$ |

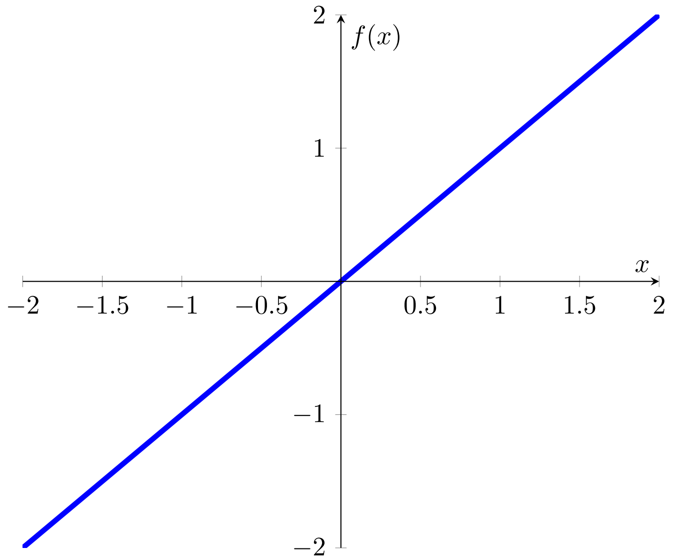


Figure 3.3. The plot of Pure-linear (Identity) function

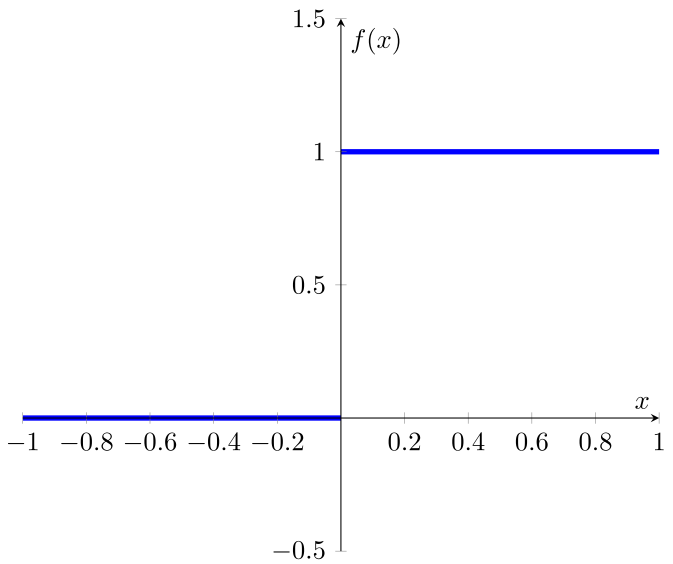


Figure 3.4. The plot of Step function with T = 0

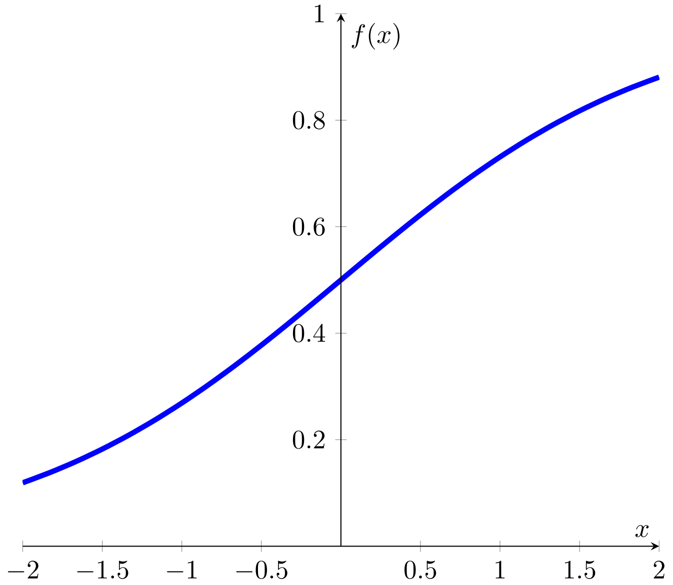


Figure 3.5. The plot of Log sigmoid function

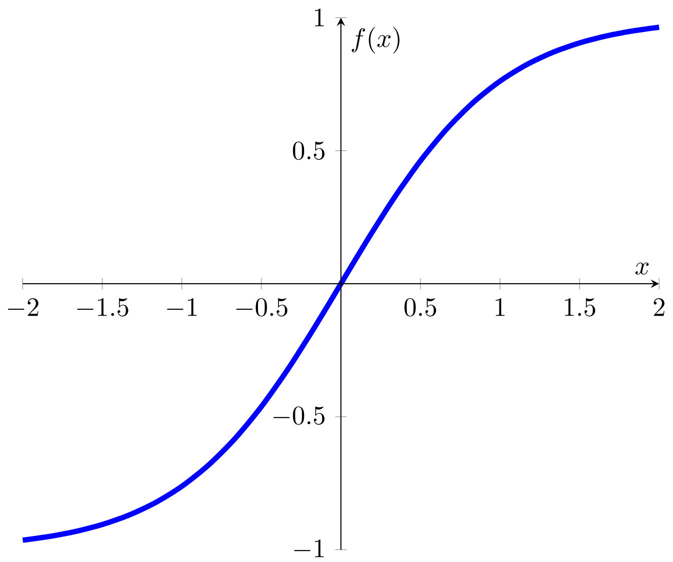


Figure 3.6. The plot of Tangent function

Lets us use the step function which is the simplest activation function in table 7 to demonstrate how an artificial neuron works. Typically, a step function will give the output of 1 if the input exceeds a specified threshold level, if not then the step function will produce the output of 0. For example, we have the following inputs, outputs and threshold level for an artificial neuron with the step function as its activation function:

— Input $x_1 = 0.3$
— Input $x_2 = 0.7$
— Weight $w_1 = 1.3$
— Weight $w_2 = 0.8$
— Threshold $T = 1.0$

According to figure 3.2, first we have to calculate the sum of all weighted input signals:

$$\sum = x_1w_1 + x_2w_2 = (0.3 * 1.3) + (0.7 * 0.8) = 0.95$$

Next, we input the calculated sum into the artificial neuron's activation function which is the step function. Because we have specified the threshold $T = 1.0$ for the step function, the calculated sum does not exceed this threshold $T$, therefore, the step function returns the output of 0, in other words, the artificial neuron does not fire.

$$0.95<1.0 \text{ therefore } f(0.95)=0$$

### 3.3. Architecture of Artificial Neural Network

In section 3.2, we have taken a look at how artificial neuron works, now we will look at the architecture of ANN to see the way an ANN connects artificial neurons together and processes information.

Because the human brain is very complex, a single model of ANN can not cover all the functions of the human brain. Therefore, there are many types of ANN, each type of ANN deals with different aspects of the human brain such as classification or segmentation (Wikipedia, 2017e). The followings are the most used types of ANN architecture (Wikipedia, 2017e):

**Feedforward neural network**: This type of network is the most simple type of ANN, the flow of the information moves in just one direction: begin from the input layer, information will be passed through the hidden layer and be transferred to the output layer. Also, there are no loops in the network; the calculation always goes forward, never goes back.

**Recurrent neural network**: Opposite to feedforward networks which information can only move from input to output, in recurrent neural networks, the information can go in a bi-directional flow. In other words, the information which has been processed in the later stages can be transferred back to earlier stages.

**Modular neural network**: Studies in biology have shown that the brain functions as a collection of small networks, therefore the concept modular neural network was conceived. In this concept, several artificial neural networks will try to cooperate or compete to solve problems.

In this section, we will discuss the detail of the feedforward neural network because it is the simplest neural network and also our primary ANN type for the case study of the thesis. First, let us consider the basic architecture of a feedforward neural network in figure 3.7.

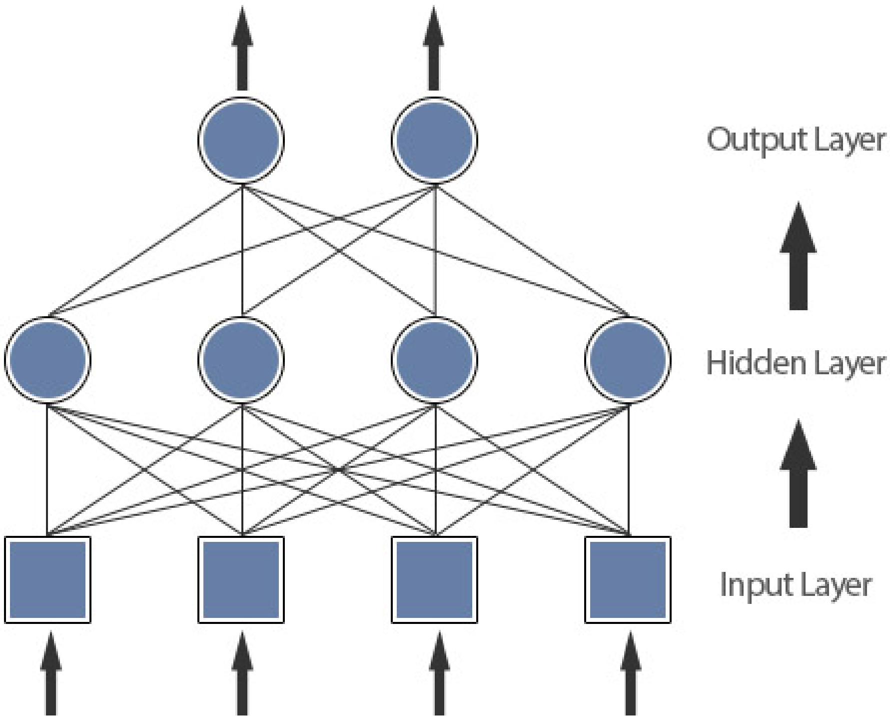


Figure 3.7. Feedforward neural network (Jacobson, 2013)

As we can see, every node in the network is an artificial neuron described in section 3.2 and these nodes are arranged in layers (Clabaugh et al., 2000):

**Input layer** which takes in inputs from the external world.

**Hidden layer** which can be one hidden layer or multiple hidden layers and does not have any connection with the external world.

**Output layer** which is responsible for producing the outputs.

In the input layer, because the artificial neurons only have the responsibility to bring the input from the external world into the network, therefore the artificial neurons in the input layer do not have any weight and will pass the same input it received from the external world to the next layer.

Each artificial neuron in a layer connects to every artificial neuron on the next layer. Thus, the information is always moving forward from one layer to the next layer. This characteristic is the reason why people call this type of artificial neural network "Feedforward" (Clabaugh et al., 2000).

The readers should also notice that there is no connection between artificial neurons in the same layer.

Often, it is required for a Feedforward neural network or ANN, in general, to have a hidden layer in order to solve problems. The reason lies in "Linear separability" of the problem on which the ANN models (Jacobson, 2013). We will demonstrate the necessity of the hidden layer by modeling OR function and XOR function using ANN.

First, let us take a look at the graphs of OR function and XOR function in figure 3.8.

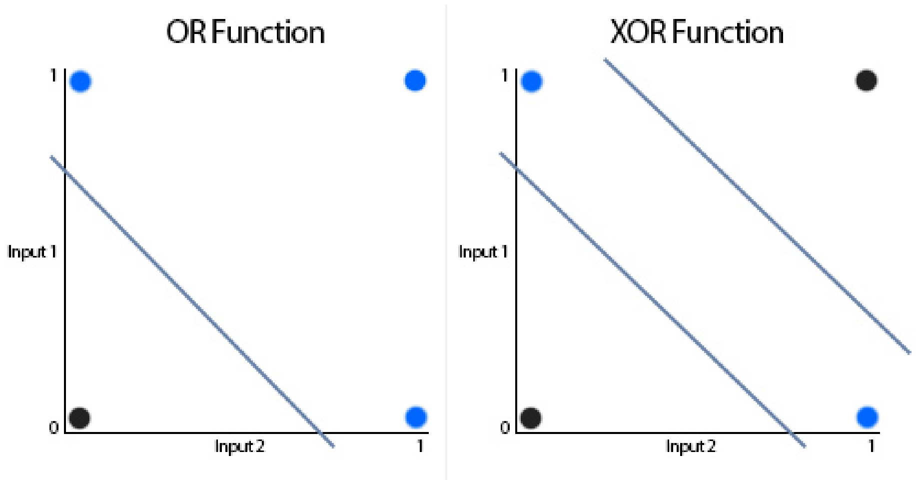


Figure 3.8. OR function and XOR function(Jacobson, 2013)

In the graph of OR function, we can see that the outputs of the OR function can be separated by a single straight line. This tells us OR function is "linearly separable", therefore it is possible to model the OR function using ANN without implementing any hidden layer (Jacobson, 2013). In fact, OR function can be modeled with a single artificial neuron like in figure 3.9.

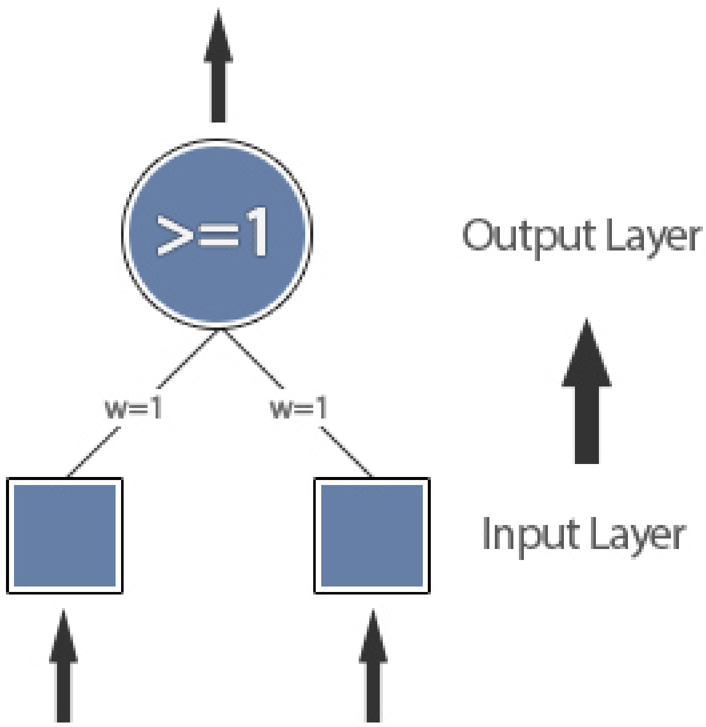


Figure 3.9. OR function modeled by a single artificial neuron (Jacobson, 2013)

For the XOR function, looking at the graph we can see that it is impossible to separate the outputs of XOR function by just using only one straight line. Therefore, we have to use an additional hidden layer to achieve the separation of the outputs of XOR function (Jacobson, 2013). The extra hidden layer is added to the ANN model like in figure 3.10.

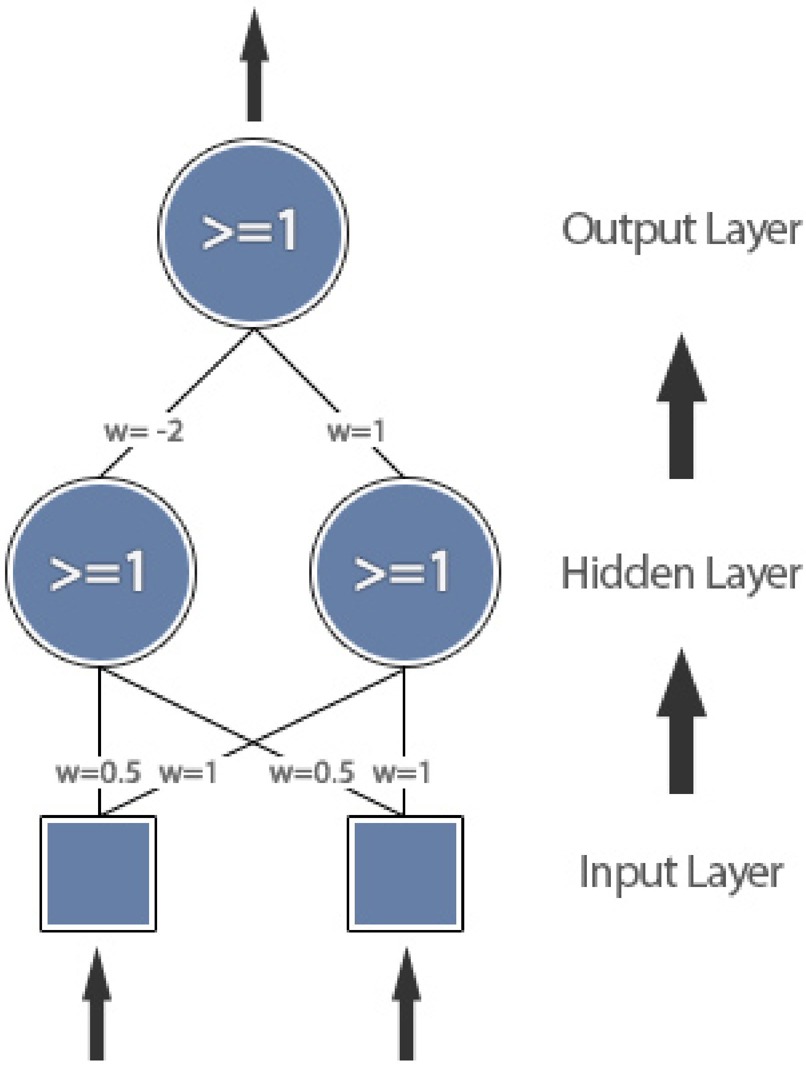


Figure 3.10. XOR function modeled with hidden layer (Jacobson, 2013)

### 3.4. The Learning Ability of Artificial Neural Network

In section 3.2 and section 3.3 we have been introduced to the basic concepts of ANN: what is an artificial neuron and how ANN connects artificial neurons to solve problems. Now we will explain the learning ability of ANN.

First, it is necessary to define what is the meaning of the word "learning" in the context of ANN. We can not say the way ANN learn is the same as how a human should learn because the possibility where machines can have awareness on what they are learning is still not yet decided. (Jacobson, 2014). However, it is possible for ANN to get to perform better with experience. Therefore, we will define the learning process of ANN as follow: Learning in the context of ANN is the ability to perform better at a given task, or a range of tasks with the experience (Jacobson, 2014).

Now, let us recall from the beginning of section 3.1 which has shown us the way we modeled ANN from its natural counterpart, the human brain. Although ANN does not capture all functionalities, ANN has been able to model though simplified the most interesting characteristic of the human brain that is the ability to learn. During the learning process of the human brain, it is believed that the biological neural network of the brain is adjusted in a way that it either increases or decreases the strength of synaptic connections between neurons depending on the subject of the learning process (Jacobson, 2014). This property is the reason why human remembers relevant information easier than information that has not been used for a long time; it is because relevant information has stronger synaptic connections while less used information will have its synaptic connection weakened, therefore, making it difficult to remember (Jacobson, 2014).

To model the learning process of the human brain, ANN adjusts the weighted connections between artificial neurons in the network. This adjustment simulates the strengthening and weakening of synaptic connections in the human brain therefore it gives ANN the ability to learn (Jacobson, 2014).

The learning ability gives ANN many advantages over problems that have challenged tradition computer programs. For example, facial recognition problem is extremely hard for a programmer to code the right solution accurately; however, it can be solved much easier by using an ANN with its learning ability. ANN also can pick up the underlying relationship in input data. Therefore, ANN can solve classification problem where a loan granting application uses past loan data to classify future loan applications (Jacobson, 2014).

### 3.5. Learning Paradigms

There are three major learning paradigms that can be used to train ANN, each paradigm has their advantages and disadvantages, but overall they share the same purpose: to find the best possible set of weights which ANN uses to accurately map any input to a correct output (Jacobson, 2014).

**Supervised Learning**: In this paradigm, ANN is provided with the desired output along with the training input as a pair when in training. By using pairs of training input - desired output as training data, we can calculate an error value based on the difference between the output we want and the output produced by ANN. Then, we can use this error value to make corrections to the network by adjusting the network's weights, therefore, improve its performance.

**Unsupervised Learning**: Opposite to supervised learning, ANN is only provided with a set of inputs, and it has to find the underlying pattern within the provided inputs without any outside intervention. Unsupervised learning is often used by data mining systems and other recommendation systems because of its ability to predict preferences of users based on the preferences of other users in the same group.

**Reinforcement Learning**: Instead of providing the desired output like in supervised learning, reinforcement learning introduces a reward system: a reward is awarded to the network based on its performance. The objective is to maximize the reward through trial-and-error. This kind of learning paradigm is similar to how animal learns in nature. For example, a dog is likely to remember the trick which the owner has given the dog the most candy (the reward in this case).

### 3.6. Training Artificial Neural Network with Back-propagation

Back-propagation is a training method for the feed-forward artificial neural network. It belongs to the supervised learning paradigm where pairs of input and output are fed into the network for many cycles until the network can learn the relationship between the input and output (Clabaugh et al., 2000).

The method starts by applying the inputs as an input vector to the input layer of the network. Then, this input vector is passed or propagated through the hidden layer, and an output vector will be produced at the end of the network which is the output layer. When the forward propagation is completed, the network evaluates the errors between the output vector generated by the network and the desired outputs. Next, it uses the evaluated errors to adjust the weights of each artificial neuron in the network according to a learning rule which aims to minimize the error. Finally, the network uses the adjusted weights to start a new cycle. This back-propagation cycle or as call as an epoch will repeat until the errors between the desired outputs and the actual outputs from the network are minimized (Mashrei, 2012). An diagram of a back-propagation cycle is depicted in figure 3.11.

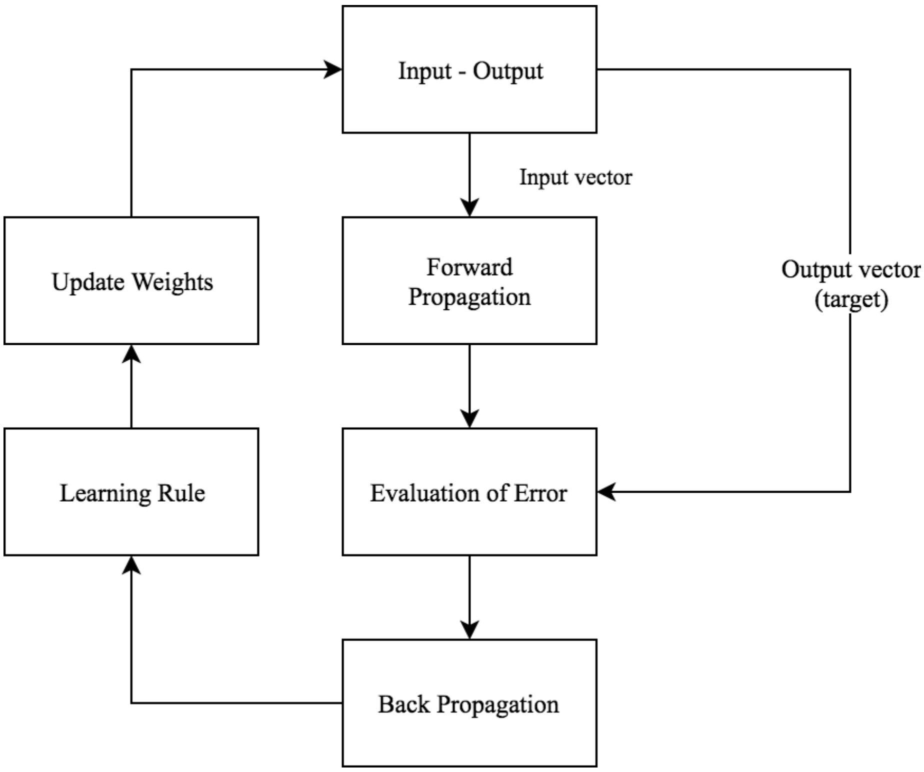


Figure 3.11. Back-propagation cycle (Mashrei, 2012)

To demonstrate the back-propagation method, let us make a simple example in the classification problem. First, we provide the network with training data which consists of an input vector $i$ and its corresponding desired output $d$, the training data could be something like in table 8.

Table 8. Example of training data

| Input $i$ | Desired output $d$ |
| --- | --- |
| $(0, 0)$ | 0 |
| $(0, 1)$ | 1 |
| $(1, 0)$ | 1 |
| $(1, 1)$ | 0 |

According to figure 3.11, every time we propagate the input vector $i$ through the network, we get the output vector $o$, we will then compare this output vector $o$ with the desired output vector $d$ to get the error value using the following calculation:

$$Error = (d - o)^2 \tag{3.1}$$

The squared difference between the output vector $o$and the desired output vector $d$gives us the sense of how far the desired value for a particular input (Clabaugh et al., 2000). The back-propagation method aims to minimize the sum of all $Error$ value for all samples in the training data thus improving the performance of the network i.e. the ability to learn the relationship between the training input and the desired output (Clabaugh et al., 2000).

$$Minimize(\sum(d - o)^2) \tag{3.2}$$

By using a continuous function such as Pure-linear or Log sigmoid as the activation function (refer to section 3.2 and table 7), we can express the change of $Error$ value or gradient with respect to the change of weight vectors $w$ as follow (Clabaugh et al., 2000):

$$Gradient = \frac{\delta Error}{\delta w} \tag{3.3}$$

To update the weight vectors every time a training sample is fed into the network we will need to use the learning rule in formula (3.4) which is based on the fact that if we decrease the value of weight vector $w$ in the direction of the gradient, the $Error$ value will decrease as well (Clabaugh et al., 2000).

$$w_{new} = w_{old} - n\frac{\delta Error}{\delta w} \tag{3.4}$$

Where:

| | |
|---|---|
| $w_{new}$ | is the newly updated weight vectors. |
| $w_{old}$ | is the old weight vectors. |
| $n$ | is the learning rate which should be a small number (about 0.1) |
| $\frac{\delta Error}{\delta w}$ | is the gradient or the change of $Error$ value with respect to the change of weight vector $w$. |

When the back-propagation cycle repeats for many epochs, by using the formula (3.4), the weight vectors are constantly adjusted so that the $Error$ value decreases to a minimum value (Clabaugh et al., 2000). When the $Error$ value is at minimum, it often means that the network is trained and is ready to produce output similar to the desired output vector $d$ when it is presented with corresponding input vector $i$.

## 4. Applying ANN in AHP method

In section §1, section §2, and section §3, we have learned the basic concepts of multi-criteria decision analysis (MCDA), the method of analytic hierarchy process (AHP), and artificial neural network (ANN). These concepts give us the rough idea of how a decision maker makes a decision in the process of MCDA. Then, by exploring the intricacy of the AHP method, we know that a decision maker can use his or her preferences to make judgments on alternatives and rank them. Moreover, we know that ANN is a powerful data modeling tool that can learn the complex relationships between input and output.

However, these concepts seem rather separated, and not much connection between them have been mentioned either. Therefore, in this section, we will introduce to the readers the proposed model from Dr. Golmohammadi which connects all three concepts mentioned above. This proposed model is used for reducing the decision maker's effort in making future ranking of alternatives (Golmohammadi, 2011). Then, we will make a case study using a real-life decision problem to introduce our implementation of this proposed model to show its feasibility to the readers.

### 4.1. The proposed model of Dr. Golmohammadi

According to the study of Dr. Golmohammadi (Golmohammadi, 2011), the task in which a decision maker have to make the best possible decision based on the existing information, and future predictions is a challenging task. Dr. Golmohammadi wants to create a model that can act as a tool and support the task above, and he chose ANN and AHP as two crucial components for his proposed model.

The proposed model of Dr. Golmohammadi can use historical data for making the future ranking of alternatives without the judgment effort of the decision maker(Golmohammadi, 2011). In figure 4.1 we can see the difference between this proposed model and other traditional models.

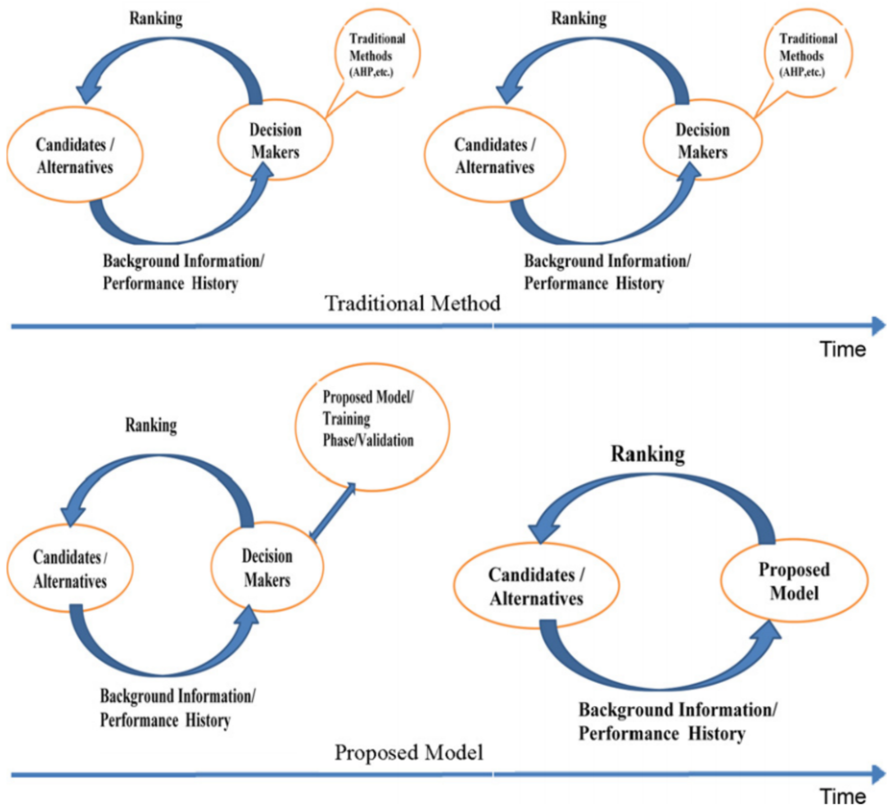


Figure 4.1. The difference between the proposed model and traditional models (Golmohammadi, 2011)

We can see in figure 4.1 that the main difference is in the second phase of the proposed model and other traditional models. In other traditional models, the decision maker has to repeat the whole process of ranking again while in the proposed model, the decision maker does not make the ranking, the proposed model makes the ranking by itself. From this difference, because the proposed model can make the ranking by itself, the decision maker does not have to spend more effort on the second phase and the following phases after.

In his proposed model, Dr. Golmohammadi chooses ANN as the mechanism for which the proposed model can use to make the ranking automatically (thanks to the learning ability of ANN, please refer to section 3.4) (Golmohammadi, 2011). Dr. Golmohammadi also chooses AHP as the pairwise comparison technique which is responsible for weight calculations which are then applied to the training data for ANN (Golmohammadi, 2011).

### 4.2. Implementation of The Proposed Model

In figure 4.2, Dr. Golmohammadi has described a model design for which we can use to implement the proposed model (Golmohammadi, 2011). However, the scope of the thesis does not allow us to follow all the steps from the model design of Dr. Golmohammadi; instead, we will make a simplified implementation design as in figure 4.3. In our simplified design, we will assume in the context that we will solve a decision problem which is a ranking problem using multi-criteria decision analysis with analytic hierarchy process as the main method. For the artificial neuron network which we will use to support the MCDA process, we will use a feedforward neural network and train it with the back-propagation method.

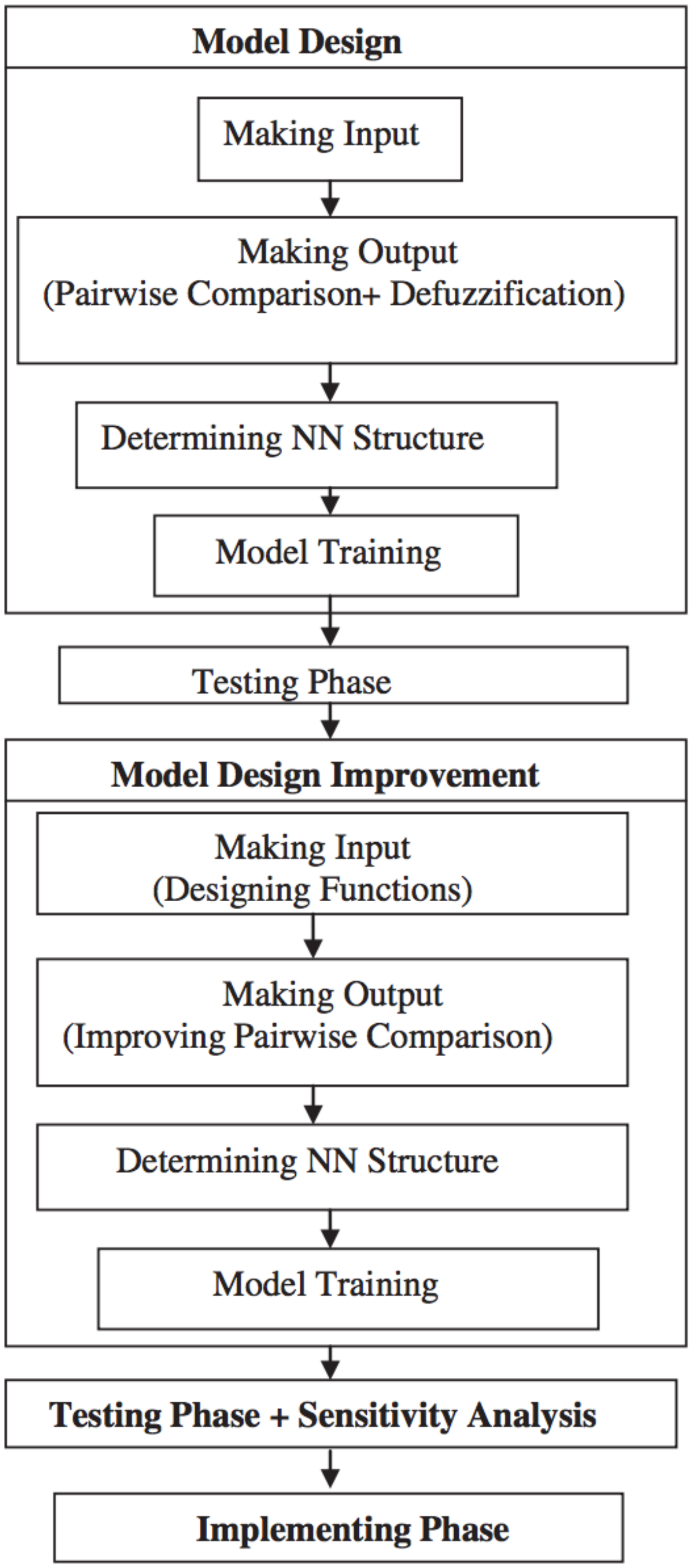


Figure 4.2. Model design steps (Golmohammadi, 2011)

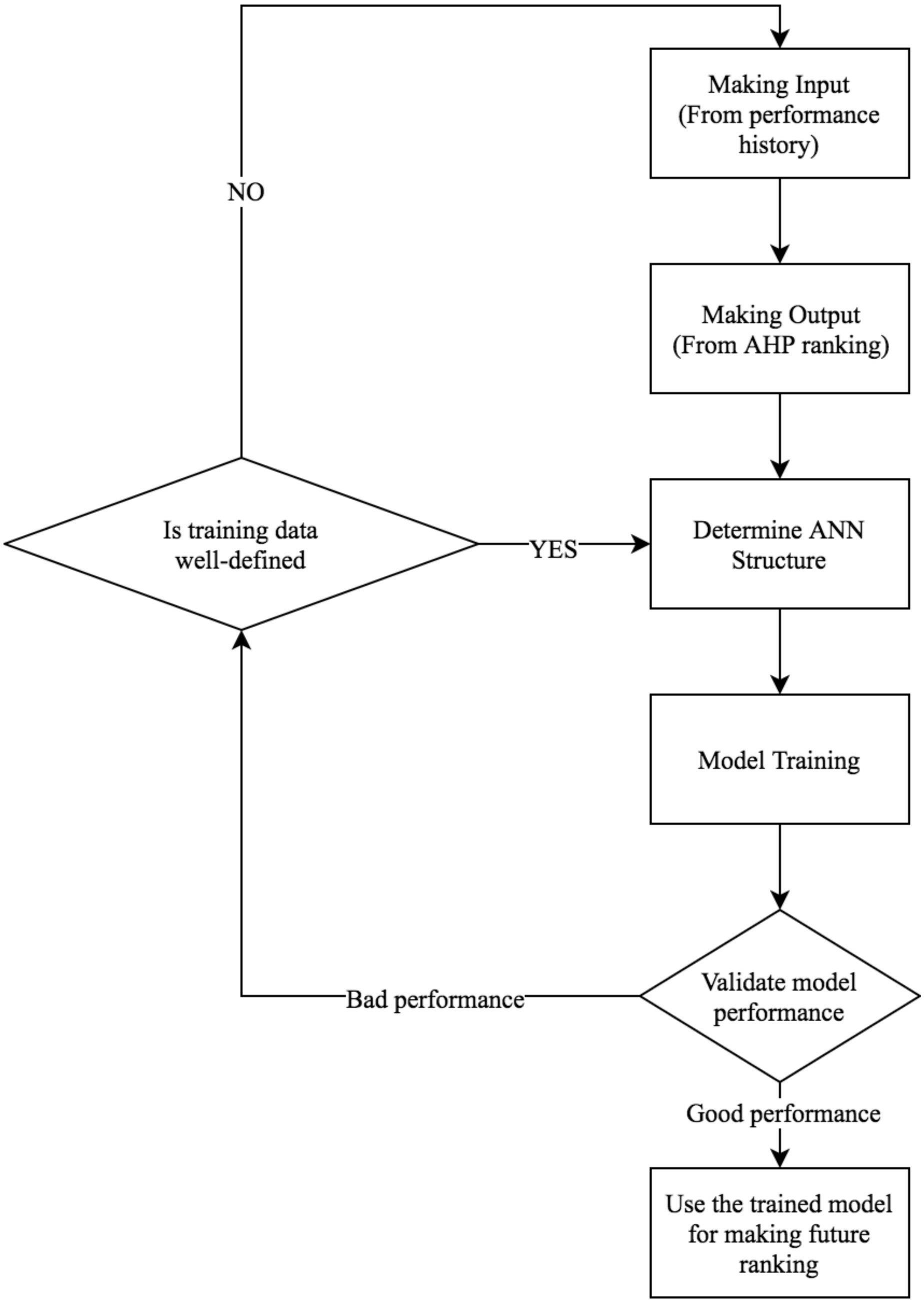


Figure 4.3. Simplified model design steps

### 4.2.1. Making Input

Recall from section 3.6, to train an ANN; we need to provide the network a set of training input. This set of training input will be the measurements of the criteria in our decision problem. These measurements are the history performance which the decision maker has used to make prior decisions.

It is also important to normalized the set of training input to have the same order of magnitude because if we do not normalize, some training input values may appear to have more significance than they actually do and this difference in order of magnitude between training input values could affect the performance of the ANN (Mashrei, 2012).

### 4.2.2. Making Output

Also from section 3.6, we will need a set of training output which is the scores for ranking of alternatives corresponding to the training input. This set of training output will be the desired output vector (refer to section 3.6 and table 8) and we will use them to train our ANN in order for the ANN to recognize the relationship between the measurements of the criteria in our decision problem and the ranking produced by the AHP method (refer to section §2 and section 2.4).

For the same reason as in section 4.2.1, we also need to normalized the set of training output.

### 4.2.3. Determine ANN Structure

Because we have assumed to use a feedforward neural network and train it with the back-propagation method, we only have to care about the number of artificial neurons in each layer of the ANN. For the input layer, the number of neurons will equal to the number of criteria we use in our decision problem. For the output layer, because our decision problem is a ranking problem therefore we just need one neuron to represent the score for ranking our alternatives.

The hardest part is to decide the number of neurons and the number of hidden layers for the hidden layer of our ANN. There is no rule of thumb to determine the exact number of neurons and number of hidden layers. Usually, the optimal numbers are found by trail-and-error in which we train the network with various configurations then to pick the configuration with the fewest number of layers and neurons that still keep the network at the minimum Mean Squared Errors (refer to section 4.2.5) (Mashrei, 2012).

### 4.2.4. Model Training

From section 3.6 and the assumption in the beginning of section 4.2, we knew that our ANN will be trained using the back-propagation method. We will keep training our ANN until the $Error$ value is at minimum, then we will test the performance of our ANN using Mean Squared Error and R-value (refer to section 4.2.5).

### 4.2.5. Validation

We evaluate the performance of ANN by using Mean Squared Errors (MSE) and R-value:

**Mean Squared Error** is defined as follow:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(o-d)^2 \tag{4.1}$$

Where:

| | |
|---|---|
| $n$ | is the number of observations. |
| $o$ | is the output vector produced by the ANN. |
| $d$ | is the desired output vector we want the ANN to learn. |

**R-value** is defined as follow:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(o-d)^2}{\sum_{i=1}^{n}(o-\overline{d})^2} \tag{4.2}$$

Where:

$\overline{d}$ is the average value of the desired output vector $d$.

When the training process is finished, we will test our trained ANN on a set of independent input and output which is exclusive from the training data. Based on the result, if the performance is good (MSE closes to 0 and R-value closes to 1) then we can use the trained ANN for future ranking. However, if the performance is bad then we have to investigate to see where is the problem in our model design.

Often bad performance is caused by bad configuration (Mashrei, 2012). Training an ANN requires a lot of parameters to be properly set such as the number of artificial neurons in the hidden layer, the number of hidden layers, the type of activation function, or did our training data is normalized yet? There are many factors that could affect the performance of the ANN and depend on the location (is problem relates to how we define the set of training input or is the number of artificial neurons is too much) we will have to repeat the model design at the suitable steps.

## 4.3. Case Study

In this section, we will demonstrate the proposed model of applying Artificial Neural Network (ANN) in multi-criteria decision making problems by applying it on actual problem based on real data sources.

### 4.3.1. Problem definition

The problem is to rank the eco-friendly levels of top twelve best-selling car manufacturers in the United Kingdom each year by using AHP method; we will call this problem Eco-Friendly Car Manufacturers Ranking Problem.

To have the list of top twelve best-selling car manufacturers for each year, we will use sale data provided by The Society of Motor Manufacturers and Traders of United Kingdom (SMMT, 2017). Each year, SMMT will release a sale report which tells us how many cars are sold in the United Kingdom in that year. The report also says the number of cars each car manufacturer has sold. Therefore we will know which car manufacturer has the best sale. Base on this information, we will select the top twelve best-selling car manufacturers for our problem.

We also need some environment related information such as fuel consumptions and emissions data for each car manufacturer in each year to determine their eco-friendly level. We will use the "Car fuel consumptions and emissions 2000-2013" dataset published by the Vehicle Certification Agency of United Kingdom Department for Transport (VCA, 2013). This dataset contains the environment related information of all car models from the year 2000 to the year 2013 in the United Kingdom. It also has the manufacturer information for each car model. Therefore, we can determine how much impact on the environment each manufacturer could make each year by averaging all the impacts on the environment of all car models each car manufacturer has for each year.

One thing about our data is that it will not contain the actual information of environment impact which each car makes each year, it just specifies how much a car model will impact on the environment. Also, the reason why we choose to rank the top twelve best-selling car manufacturers is to reduce the amount of pair-wise comparison which could get exponentially large if we have too many alternatives (refer to table 2).

Therefore, the ranking produced from the Eco-Friendly Car Manufacturers Ranking Problem will tell us the potential environment impact level of car manufacturers would make for each year. With this insight, if we want to purchase a new car, we can reference this ranking and make our decision so that the car we buy will be the most eco-friendly car.

Although the datasets from SMMT and VCA have data for more than a decade, we will pick only data from the year 2009, 2010, 2011, 2012 and 2013 to use in the demonstration of our problem. This amount of data is adequate for the training of ANN, and it also helps to reduce the scope of our problem.

#### 4.3.2. Data Sources

**Sale Report for Car in United Kingdom:** We will use the sale report provided by (SMMT, 2017) each year to select our top twelve best-selling car manufacturers. This report contains the total number of cars sold in one year span in United Kingdom. It also groups the car sales by manufacturers, therefore we simply sort the report in descending order and pick the top twelve car manufacturers to have our best-selling car manufacturers.

Table 9. UK Car Sales 2009

| Manufacturer | Sales (cars) |
|---|---|
| Ford | 129.287 |
| Vauxhall | 101.023 |
| Volkswagen | 62.892 |
| Toyota | 40.265 |
| Peugeot | 39.164 |
| Audi | 39.180 |
| BMW | 34.098 |
| Honda | 31.423 |
| Mercedes-Benz | 27.137 |
| Citroen | 25.788 |
| Nissan | 24.093 |
| Renault | 21.139 |

Table 10. UK Car Sales 2010

| Manufacturer | Sales (cars) |
|---|---|
| Ford | 280.364 |
| Vauxhall | 247.265 |
| Volkswagen | 174.655 |
| BMW | 109.418 |
| Peugeot | 109.324 |
| Audi | 99.828 |
| Renault | 95.608 |
| Nissan | 89.681 |
| Toyota | 87.396 |
| Mercedes-Benz | 74.977 |
| Citroen | 73.317 |
| Honda | 63.652 |

Table 11. UK Car Sales 2011

| Manufacturer | Sales (cars) |
|---|---|
| Ford | 265.894 |
| Vauxhall | 234.710 |
| Volkswagen | 179.290 |
| BMW | 116.642 |
| Audi | 113.797 |
| Nissan | 96.269 |
| Peugeot | 94.989 |
| Mercedes-Benz | 81.873 |
| Toyota | 73.589 |
| Citroen | 68.464 |
| Renault | 68.449 |
| Hyundai | 62.900 |

Table 12. UK Car Sales 2012

| Manufacturer | Sales (cars) |
|---|---|
| Ford | 281.917 |
| Vauxhall | 232.255 |
| Volkswagen | 183.098 |
| BMW | 127.530 |
| Audi | 123.622 |
| Nissan | 105.835 |
| Peugeot | 99.486 |
| Mercedes-Benz | 91.855 |
| Toyota | 84.563 |
| Hyundai | 74.285 |
| Citroen | 73.656 |
| Kia | 66.629 |

Table 13. UK Car Sales 2013

| Manufacturer | Sales (cars) |
|---|---|
| Ford | 310.865 |
| Vauxhall | 259.444 |
| Volkswagen | 194.085 |
| Audi | 142.040 |
| BMW | 135.583 |
| Nissan | 117.967 |
| Mercedes-Benz | 109.456 |
| Peugeot | 105.435 |
| Toyota | 88.648 |
| Citroen | 78.358 |
| Hyundai | 76.918 |
| Kia | 72.090 |

**Car Fuel Consumptions And Emissions 2000-2013:** The "Car fuel consumptions and emissions 2000-2013" dataset from VCA is a collection of fuel consumptions and emissions data from car models from all car manufacturers in United Kingdom from year 2000 to year 2013. This dataset has in total of 27 fields of information, we will not use all of those fields, instead we will use only the following 4 fields which give us enough information of how much the environment impact each car model has, they are:

**CO2**: CO2 emissions in grammes per kilometer (g/km).

**Noise level**: External noise emitted by a car shown in decibels as measured on the A scale of a noise meter (dB (A)).

**Combined fuel consumption**: average of the urban and extra-urban tests, weighted by the distances covered in each part, in liters per 100 Kilometers (l/100 Km).

**Engine capacity**: Engine capacity in cubic centimeters (cc).

These 4 fields will be used as criteria to determine how much a car model can impact the environment. However, these fields not only show us the environment impact information but also show us how efficient a car model can be in terms of saving the environment e.g. a car model might have high engine capacity but its CO2 emissions and Noise level might not as high as other car models with same engine capacity, this tells us that car model is using superior technology to minimize the amount of CO2 emissions and Noise level while maintain high engine capacity.

Table 14. Car Manufacturers With Averaged Environmental Information For Year 2009

| Manufacturer | Fuel consumption | Noise level | CO2 | Engine capacity |
|---|---|---|---|---|
| Ford | 6.53 | 70.87 | 164.06 | 1,847.74 |
| Vauxhall | 7.06 | 72.56 | 175.58 | 1,815.62 |
| Volkswagen | 7.20 | 72.30 | 182.38 | 2,004.47 |
| Toyota | 6.30 | 71.03 | 156.89 | 1,838.27 |
| Peugeot | 6.48 | 73.43 | 162.48 | 1,710.82 |
| Audi | 7.95 | 72.59 | 195.12 | 2,499.68 |
| BMW | 7.36 | 71.91 | 182.65 | 2,735.24 |
| Honda | 7.23 | 71.17 | 174.84 | 1,956.88 |
| Mercedes-Benz | 8.06 | 72.21 | 199.32 | 2,619.47 |
| Citroen | 6.57 | 72.84 | 165.01 | 1,734.65 |
| Nissan | 7.79 | 71.18 | 194.47 | 1,999.72 |
| Renault | 6.80 | 71.31 | 169.56 | 1,776.82 |

Table 15. Car Manufacturers With Averaged Environmental Information For Year 2010

| Manufacturer | Fuel consumption | Noise level | CO2 | Engine capacity |
|---|---|---|---|---|
| Ford | 6.49 | 70.66 | 162.67 | 1,860.81 |
| Vauxhall | 6.67 | 72.11 | 164.64 | 1,677.48 |
| Volkswagen | 6.96 | 72.10 | 176.54 | 1,942.49 |
| BMW | 7.32 | 71.77 | 180.84 | 2,752.63 |
| Peugeot | 6.21 | 73.28 | 154.67 | 1,660.64 |
| Audi | 7.97 | 72.60 | 195.13 | 2,550.52 |
| Renault | 6.47 | 71.91 | 161.85 | 1,760.33 |
| Nissan | 7.31 | 70.53 | 180.61 | 1,997.75 |
| Toyota | 5.94 | 71.10 | 147.53 | 1,853.17 |
| Mercedes-Benz | 7.87 | 72.04 | 194.87 | 2,604.66 |
| Citroen | 6.20 | 72.89 | 156.80 | 1,719.90 |
| Honda | 6.35 | 70.06 | 154.69 | 1,858.30 |

Table 16. Car Manufacturers With Averaged Environmental Information For Year 2011

| Manufacturer | Fuel consumption | Noise level | CO2 | Engine capacity |
|---|---|---|---|---|
| Ford | 6.47 | 70.46 | 160.00 | 1,856.30 |
| Vauxhall | 6.10 | 71.89 | 151.17 | 1,637.36 |
| Volkswagen | 6.62 | 71.64 | 166.15 | 1,881.00 |
| BMW | 6.96 | 71.78 | 171.05 | 2,633.44 |
| Audi | 7.19 | 72.54 | 174.72 | 2,402.03 |
| Nissan | 7.00 | 70.59 | 173.71 | 1,930.12 |
| Peugeot | 5.71 | 72.59 | 141.90 | 1,624.74 |
| Mercedes-Benz | 7.17 | 71.87 | 177.23 | 2,477.43 |
| Toyota | 5.81 | 70.98 | 142.77 | 1,777.88 |
| Citroen | 5.69 | 72.60 | 141.82 | 1,613.25 |
| Renault | 6.23 | 71.95 | 155.23 | 1,722.69 |
| Hyundai | 5.87 | 71.82 | 144.66 | 1,662.88 |

Table 17. Car Manufacturers With Averaged Environmental Information For Year 2012

| Manufacturer | Fuel consumption | Noise level | CO2 | Engine capacity |
|---|---|---|---|---|
| Ford | 6.14 | 69.99 | 149.60 | 1,813.85 |
| Vauxhall | 5.80 | 71.87 | 143.52 | 1,636.40 |
| Volkswagen | 6.24 | 71.24 | 156.70 | 1,795.22 |
| BMW | 6.60 | 72.01 | 162.87 | 2,574.89 |
| Audi | 6.65 | 72.04 | 161.67 | 2,309.72 |
| Nissan | 6.80 | 70.49 | 163.73 | 1,899.06 |
| Peugeot | 5.50 | 72.36 | 135.18 | 1,604.80 |
| Mercedes-Benz | 6.70 | 72.07 | 166.47 | 2,486.44 |
| Toyota | 5.52 | 71.31 | 135.42 | 1,800.60 |
| Hyundai | 5.55 | 71.92 | 137.62 | 1,590.19 |
| Citroen | 5.49 | 72.47 | 136.33 | 1,649.97 |
| Kia | 5.49 | 72.09 | 136.93 | 1,571.45 |

Table 18. Car Manufacturers With Averaged Environmental Information For Year 2013

| Manufacturer | Fuel consumption | Noise level | CO2 | Engine capacity |
|---|---|---|---|---|
| Ford | 5.70 | 69.80 | 141.16 | 1,735.91 |
| Vauxhall | 5.58 | 71.66 | 138.11 | 1,614.11 |
| Volkswagen | 6.06 | 71.35 | 151.75 | 1,779.67 |
| Audi | 6.70 | 72.01 | 162.52 | 2,381.38 |
| BMW | 6.08 | 72.18 | 149.66 | 2,339.21 |
| Nissan | 6.81 | 70.85 | 165.59 | 1,950.59 |
| Mercedes-Benz | 6.30 | 72.45 | 155.03 | 2,371.14 |
| Peugeot | 5.34 | 72.15 | 132.08 | 1,620.88 |
| Toyota | 5.46 | 71.78 | 133.03 | 1,802.49 |
| Citroen | 5.25 | 72.52 | 130.54 | 1,619.99 |
| Hyundai | 5.49 | 72.18 | 136.48 | 1,618.02 |
| Kia | 5.38 | 72.15 | 133.39 | 1,555.73 |

### 4.3.3. Solving Problem Using AHP Method

**Define Decision Hierarchy And Global Priorities** First, we must structure our Eco-Friendly Car Manufacturers Ranking Problem in a decision hierarchy with appropriate criteria which are: Combined fuel consumption, Noise Level, CO2, and Engine capacity. These criteria must also be assigned with global priorities which should be computed by pairwise comparisons to reflect the following order of importance:

$$CO2 > NoiseLevel > Combinedfuelconsumption = Enginecapacity \quad (4.3)$$

To compute the global priorities for our criteria in order to follow the above order of importance, we make the pairwise comparisons as in figure 4.4.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 0.20 | 0.14 | 1.00 |
| 2 | 5.00 | 1 | 0.33 | 5.00 |
| 3 | 7.00 | 3.00 | 1 | 7.00 |
| 4 | 1.00 | 0.20 | 0.14 | 1 |

Figure 4.4. Judgement Table For Criteria

Where:
1. Combined fuel consumption
2. Noise level
3. CO2
4. Engine capacity

After we had the pairwise comparisons, we can compute the priorities and ranking as in figure 4.5. Then when we got the priorities for our criteria, we can define our decision hierarchy as in figure 4.6.

| | Category | Priority | Rank |
|---|---|---|---|
| 1 | Fuel Consumption | 6.7% | 4 |
| 2 | Noise Level | 28.2% | 2 |
| 3 | CO2 | 58.3% | 1 |
| 4 | Engine Capacity | 6.7% | 3 |

Figure 4.5. Criteria Priorities And Ranking

**Decision Hierarchy**

| Level 0 | Level 1 | Global Priorities |
|---|---|---|
| Environment Friendly Car Manufactur | Fuel Consumption 0.0674 | 6.7 % |
| | Noise Level 0.2825 | 28.2 % |
| | CO2 0.5827 | 58.3 % |
| | Engine Capacity 0.0674 | 6.7 % |
| | | 1.0 |

Figure 4.6. Decision Hierarchy With Global Priorities

**Calculate Scores of Alternatives** When we have got the global priorities for our criteria, we can proceed to calculate the scores of our Alternatives. We will use the data in table 14 to demonstrate the calculation of our scores.

First, similar to the process of finding the global priorities for our criteria, we must find the priority vector of the 12 car manufacturers for each criteria. To find the priority vector for each criteria, we make the pairwise comparisons between 12 car manufacturers for each criteria. For example, we will make pairwise comparisons for the criterion fuel consumption. Because we have 12 alternatives therefore we have to make 66 comparisons (refer to table 2) as in figure 4.7, then we get the priority vector in figure 4.8 by solving the principal Eigenvectors problem (refer to section §2).

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 3.00 | 4.00 | 0.33 | 8.00 | 0.14 | 6.00 | 5.00 | 9.00 | 2.00 | 7.00 | 3.00 |
| 2 | 0.33 | 1 | 2.00 | 0.17 | 7.00 | 0.20 | 4.00 | 3.00 | 8.00 | 0.33 | 7.00 | 0.50 |
| 3 | 0.25 | 0.50 | 1 | 0.14 | 6.00 | 0.20 | 3.00 | 2.00 | 6.00 | 0.20 | 4.00 | 0.33 |
| 4 | 3.00 | 6.00 | 7.00 | 1 | 9.00 | 2.00 | 7.00 | 6.00 | 9.00 | 3.00 | 7.00 | 5.00 |
| 5 | 0.13 | 0.14 | 0.17 | 0.11 | 1 | 0.11 | 0.33 | 0.25 | 3.00 | 0.14 | 0.50 | 0.17 |
| 6 | 7.00 | 5.00 | 5.00 | 0.50 | 9.00 | 1 | 7.00 | 7.00 | 9.00 | 3.00 | 8.00 | 4.00 |
| 7 | 0.17 | 0.25 | 0.33 | 0.14 | 3.00 | 0.14 | 1 | 0.50 | 5.00 | 0.17 | 2.00 | 0.20 |
| 8 | 0.20 | 0.33 | 0.50 | 0.17 | 4.00 | 0.14 | 2.00 | 1 | 5.00 | 0.20 | 3.00 | 0.25 |
| 9 | 0.11 | 0.13 | 0.17 | 0.11 | 0.33 | 0.11 | 0.20 | 0.20 | 1 | 0.11 | 0.33 | 0.13 |
| 10 | 0.50 | 3.00 | 5.00 | 0.33 | 7.00 | 0.33 | 6.00 | 5.00 | 9.00 | 1 | 7.00 | 2.00 |
| 11 | 0.14 | 0.14 | 0.25 | 0.14 | 2.00 | 0.13 | 0.50 | 0.33 | 3.00 | 0.14 | 1 | 0.17 |
| 12 | 0.33 | 2.00 | 3.00 | 0.20 | 6.00 | 0.25 | 5.00 | 4.00 | 8.00 | 0.50 | 6.00 | 1 |

Figure 4.7. Judgement Table For Fuel Consumption in year 2009

Where:
1. Ford
2. Vauxhall
3. Volkswagen
4. Toyota
5. Peugeot
6. Audi
7. BMW
8. Honda
9. Mercedes-Benz
10. Citroen
11. Nissan
12. Renault

Repeat the process for the remaining criteria to find all priority vectors, then we multiply the priority vector of each criteria with the global priorities to get the consolidated priorities or the scores for ranking for our 12 car manufacturers in year 2009 in figure 4.9. Finally we get the ranking for our 12 car manufactures in year 2009 in figure 4.10. The scores and rankings for year 2010, 2011, 2012 and 2013 are shown in figure 4.11, figure 4.12, figure 4.13, and figure 4.14.

| Category | | Priority | Rank |
|---|---|---|---|
| 1 | Ford | 12.7% | 3 |
| 2 | Vauxhall | 6.3% | 6 |
| 3 | Volks | 4.4% | 7 |
| 4 | Toyota | 24.0% | 1 |
| 5 | Audi | 1.4% | 11 |
| 6 | Peugeot | 23.8% | 2 |
| 7 | BMW | 2.4% | 9 |
| 8 | Honda | 3.2% | 8 |
| 9 | Mercedes-Benz | 1.0% | 12 |
| 10 | Citroen | 11.3% | 4 |
| 11 | Nissaan | 1.7% | 10 |
| 12 | Renault | 7.9% | 5 |

Figure 4.8. Priority vector for Fuel Consumption in year 2009

**Decision Hierarchy**

| Level 0 | Level 1 | Global Priorities | Ford | Vauxhall | Volks | Toyota | Audi | Peugeot | BMW | Honda | Mercedes -Benz | Citroen | Nissaan | Renault |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Environment Friendly Car Manufactur | Fuel Consumption 0.0674 | 6.7 % | 0.0085 | 0.0043 | 0.0029 | 0.0162 | 0.0009 | 0.016 | 0.0016 | 0.0021 | 0.0007 | 0.0076 | 0.0012 | 0.0053 |
| | Noise Level 0.2825 | 28.2 % | 0.0677 | 0.007 | 0.0093 | 0.0579 | 0.0052 | 0.0033 | 0.0172 | 0.0444 | 0.0123 | 0.0041 | 0.0312 | 0.0229 |
| | CO2 0.5827 | 58.3 % | 0.0852 | 0.0249 | 0.0181 | 0.1552 | 0.0075 | 0.1158 | 0.0136 | 0.0338 | 0.0062 | 0.0659 | 0.0097 | 0.0468 |
| | Engine Capacity 0.0674 | 6.7 % | 0.0028 | 0.0015 | 0.008 | 0.0021 | 0.0099 | 0.0007 | 0.0177 | 0.0039 | 0.0134 | 0.0009 | 0.0053 | 0.0011 |
| | | 1.0 | **16.4 %** | **3.8 %** | **3.8 %** | **23.1 %** | **2.4 %** | **13.6 %** | **5.0 %** | **8.4 %** | **3.3 %** | **7.8 %** | **4.7 %** | **7.6 %** |

Figure 4.9. Consolidated priorities for car manufactures in year 2009

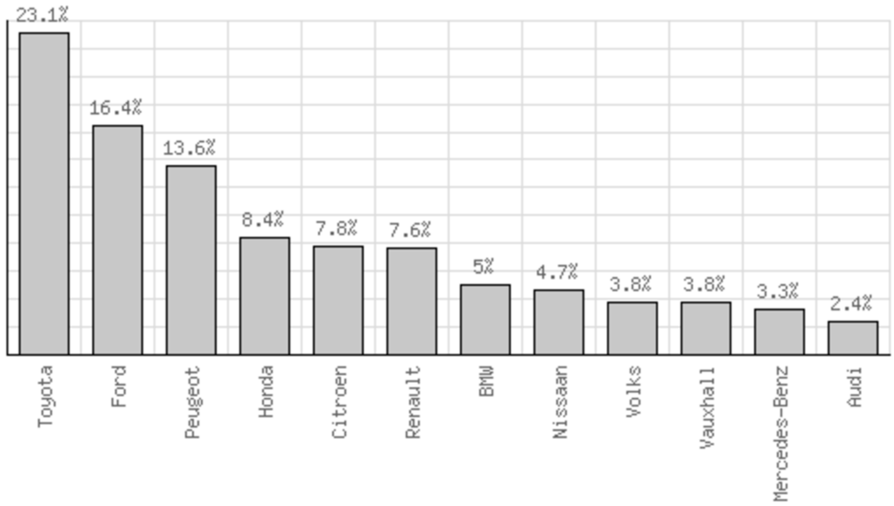


Figure 4.10. Car manufacturers ranking in 2009

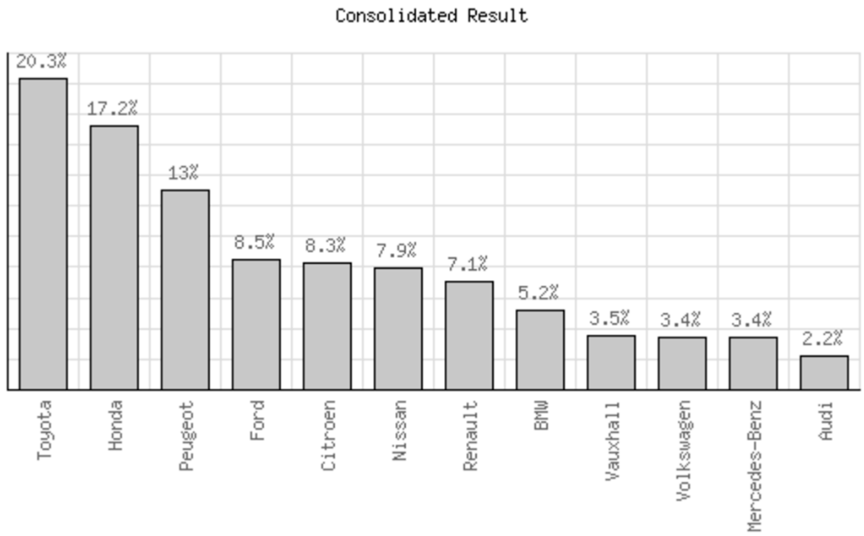


Figure 4.11. Car manufacturers ranking in 2010

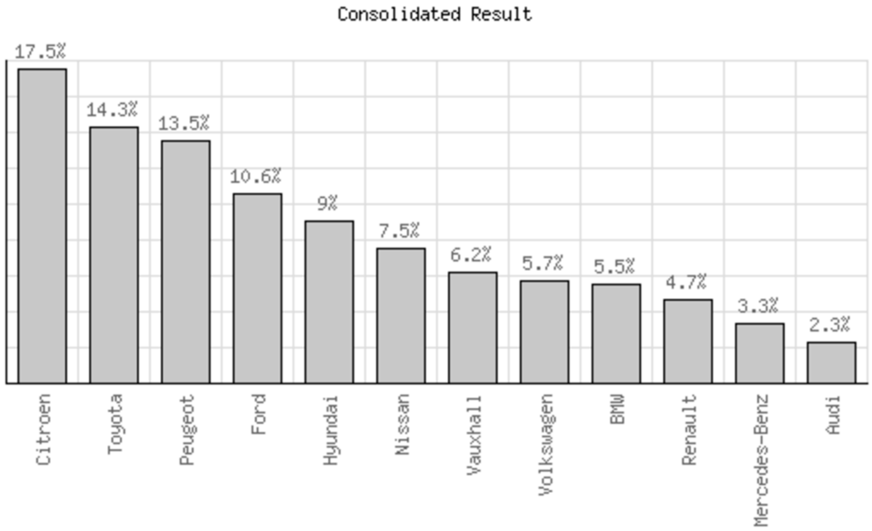


Figure 4.12. Car manufacturers ranking in 2011

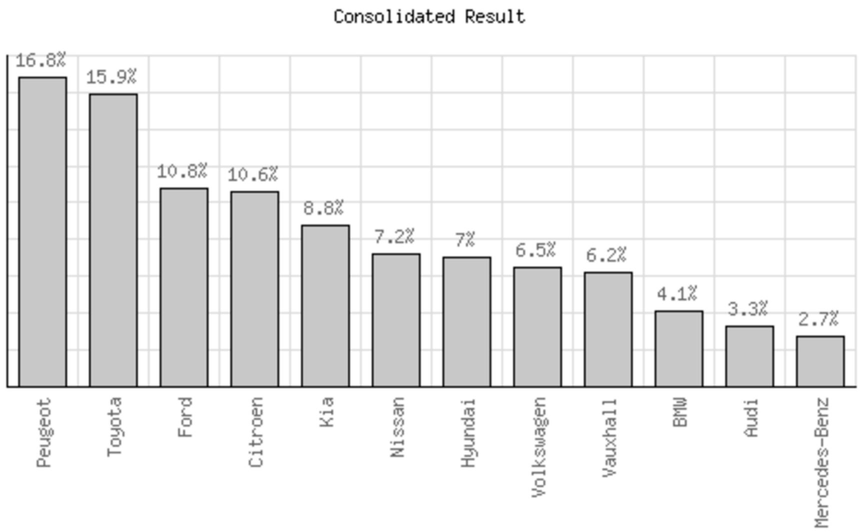


Figure 4.13. Car manufacturers ranking in 2012

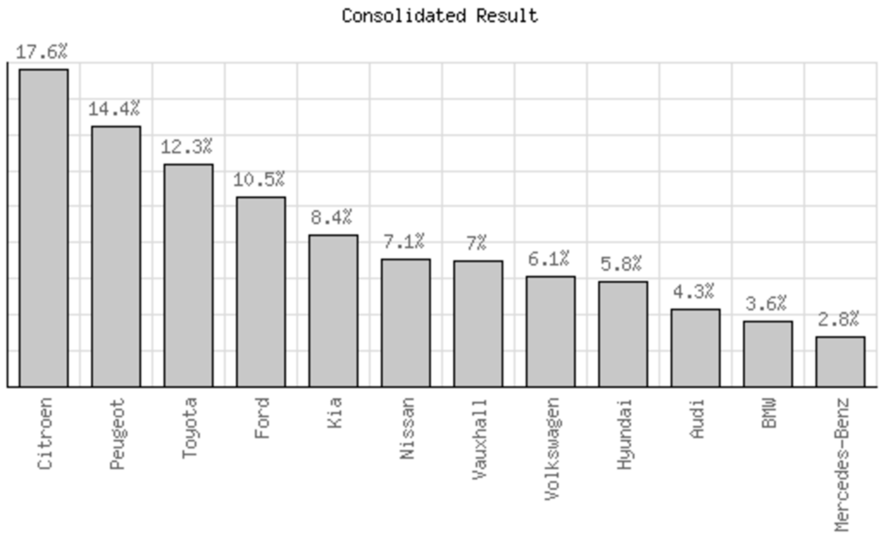


Figure 4.14. Car manufacturers ranking in 2013

### 4.3.4. Artificial Neural Network Training

**Making Input and Output:** Recall from section 4.2.1 and section 4.2.2, we will use the measurements from table 14, table 15, table 16, and table 17 as our training input. But we will not use table 18 as out training input, instead, we will use it for validation.

For the training output, we will use the scores from figure 4.10, figure 4.11, figure 4.12, figure 4.13. And for the same reason as input, we will only use figure 4.14 for validation.

The readers should notice that all training data are normalized to have the same order of magnitude.

**Determine ANN Structure:** Because we have 4 criteria, therefore we will have 4 artificial neurons in our input layers, as for output layer we will only need 1 neuron because the score for ranking for each car manufacturer is just a single numerical value.

For the number of neurons in hidden layer and the number of hidden layers, we will choose [94], this means we will have 2 hidden layers and the first layer will have 9 neurons and the second layer will have 4 neurons. The reason we choose this configuration is because after several tries, this configuration gives the best performance for our ANN. Also, the activation function for our neurons in the hidden layers will be the Tangent function (refer to table 7) and the activation function for the neuron in the output layer will be the Pure-linear function.

Therefore the shape of our artificial neural network will look like figure 4.15.

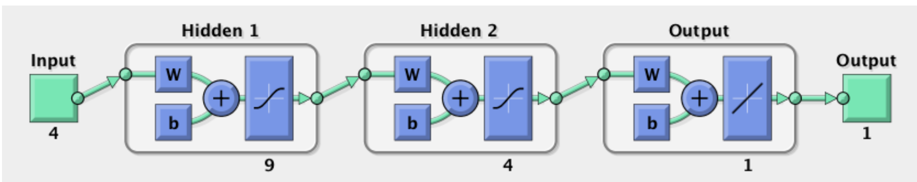


Figure 4.15. Artificial Neural Network Shape

**Model Training:** MATLAB version R2016b with Neural Network Toolbox will be our main tool for creating, training calculating the performance for our ANN.

To create an instance of ANN we will use feedforwardnet function from Neural Network Toolbox in MATLAB. The feedforwardnet function will generate a feedforward neural network consist of series of layers. The first layer for connecting the network input, subsequent hidden layers each has connection from previous layer, and the final layer for producing the network output.

We have to prepare these parameters before using feedforwardnet:

**hiddenSizes** - is a vector of one or more hidden layer sizes, we have chosen [9 4] so we will use "[9 4]".

**trainFcn** - is where we will specify our training function, we will use "traingdm" which is a network training function that updates weight and bias values according to gradient descent with momentum. We use this so we can control learning rate and momentum value.

The code to create an instance of ANN in MATLAB:

```
% Create a Feedforward Neural Network
hiddenSizes = [9 4];
trainFcn = 'traingdm';
net = feedforwardnet(hiddenSizes,trainFcn);
```

To train an instance of ANN we will use the train function from Neural Network Toolbox in MATLAB. The train function will train a neural network using the training parameters we provide to the neural network. Depending on the chosen training function, we will put in different type of training parameters, in this case we have chosen "traingdm" as our training function.

We have to provide these parameters to the neural network before using train:

**divideFcn** - we specify the function to divide training data.
**divideMode** - we specify the mode for the divide function.
**divideParam.trainRatio** - we specify ratio of data used for training.
**divideParam.valRatio** - we specify ratio of data used for validation. .
**divideParam.testRatio** - we specify ratio of data used for testing.
**trainParam.lr** - we will specify the learning rate of the neural network.
**trainParam.mc** - we will specify the momentum value of the neural network.
**trainingInputs** - is the inputs from training data.
**targets** - is the targets from training data.

The code to create and train an instance of ANN in MATLAB:

```
% Set seed so we can get the same result
% everytime we train the ANN with same parameters
RandStream.setGlobalStream (RandStream ('mrg32k3a','Seed', 1234));

% Create a Feedforward Neural Network
hiddenSizes = [9 4];
trainFcn = 'traingdm';
net = feedforwardnet(hiddenSizes,trainFcn);

% Set the parameters
net.divideFcn = 'divideblock'; % Divide data to sequential blocks
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 60/100;
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 20/100;
net.trainParam.lr = 0.35;
net.trainParam.mc = 0.3;
net.trainParam.max_fail = 1000;
net.trainparam.epochs = 10000;

% Train the Network
[net,tr] = train(net,x,t);
```

**Validation** After our ANN is trained, we will use the input data from table 18 and the output data from figure 4.14 to test the performance of our ANN. The result of R-value is shown in figure 4.16.

For the MSE, we will compare the the outputs produced from the ANN with the actual scores we have calculated for data in year 2013 and then calculate the MSE value for the ANN in table 19.

Table 19. Mean Squared Error

| | AHP-Score | ANN-Score |
|---|---|---|
| | 10.51% | 12.35% |
| | 7.03% | 5.76% |
| | 6.14% | 5.44% |
| | 4.29% | 3.11% |
| | 3.60% | 3.81% |
| | 7.10% | 3.68% |
| | 2.81% | 2.59% |
| | 14.41% | 10.66% |
| | 12.28% | 10.21% |
| | 17.59% | 14.13% |
| | 5.84% | 6.51% |
| | 8.42% | 8.41% |
| **MSE** | | 0.0004110 |

We can see that both MSE and R-value gives very high result, therefore we can be confident that our ANN has learned to accurately predict the score of a car manufacturer with the data that is not included in the training data.

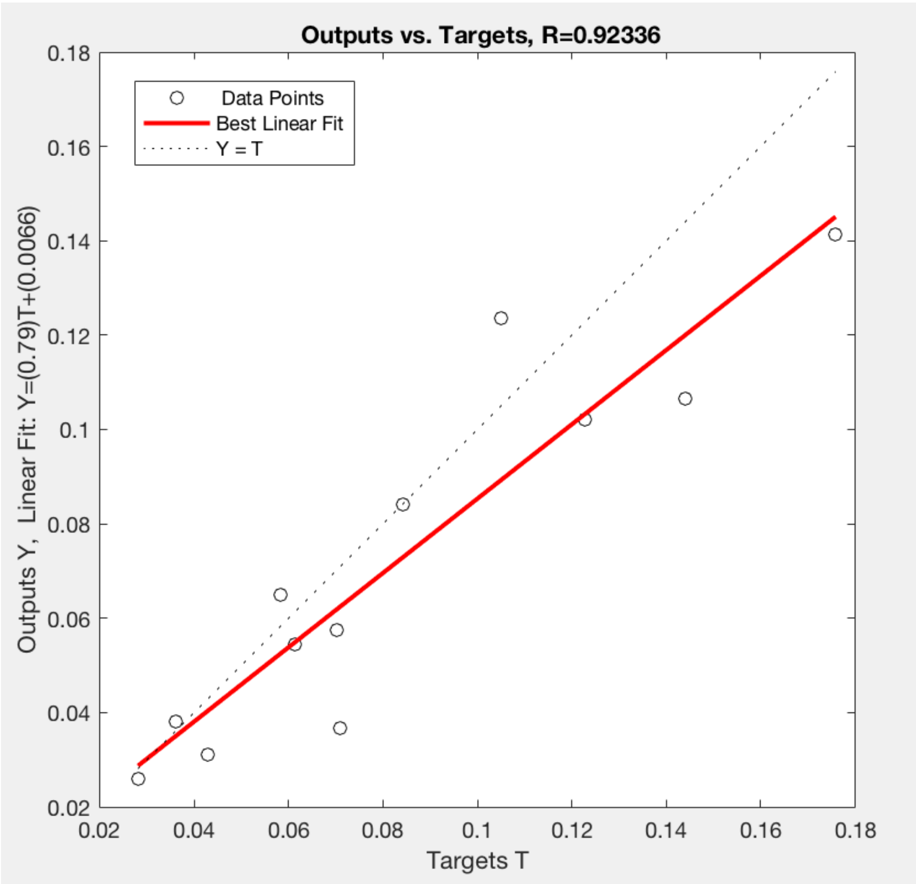


Figure 4.16. R-value graph

## Conclusions

Now we have come to the end of the thesis; we will conclude our study with some summarizes and discussions:

**Closing the gap between artificial neural network and multi-criteria decision analysis:** We have been shown with many great features of the artificial neural network: the inspiration from the biological neuron to the artificial neuron, how artificial neurons connect with each other in layers to create a particular kind of system that can simulate the human brain and its most important function - the ability to learn. With the capacity to learn the underlying complex relationship between input and output, an artificial neural network can even solve challenging classification problems where traditional computer program finds the most difficult.

With the learning capability of the artificial neural network, we have followed the proposed model from Dr. Golmohammadi (Golmohammadi, 2011) and made an attempt to implement this model. Although the implementation is simplified to some extent, we have been shown with promising results.

As Dr. Golmohammadi has suggested (refer to section 4.1), his proposed model can use historical data for making the future ranking of alternatives without the judgment effort of the decision maker. Although we can not use this model entirely and replace the traditional decision-making methods (according to figure 4.1), it is still an interesting approach to reduce the amount of work for the decision maker by using an artificial neural network to simulate the way the decision maker judge and make decisions.

However, when the validation results of our attempt in implementing Dr. Golmohammadi's proposed model has given such an excellent performance, we can be assured that this approach is not just only interesting but also bridges the gap between multi-criteria decision analysis and artificial neural network and opens the gate to many new kinds of application.

**Issues with the artificial neural network:** There are many problems which have been found during the process of implementing the proposed model of Dr. Golmohammadi.

First, the network will not work if we introduce new criteria to our problem in our case study (refer to section 4.3). This issue happens because the network has only been trained with the training data that only contains information related to the provided criteria. If we introduce new criteria, the network will stop working because it does not have the relevant information about the new criteria. In the situation of new criteria being introduced, we have to restart everything from scratch again, in other words, we have to build an entirely new training data which contains information about the new criteria so the artificial neural network can learn.

Second, if the training data is not well-defined, the performance of the artificial neural network will be deeply affected and can not be useful anymore. Therefore, we have to be extra careful when modeling the training data in order not to waste too much time in training the neural network, in particular with the case where we have a large set of training data, and the architecture of the artificial neural network is complex.

Third, this study only scrapes the tip of the iceberg. Until now, the artificial neural network seems very simple to us because we only pick the most basic and easiest to understand concepts of artificial neural network and machine learning in general. Looking at the code where we created an instance of artificial neural network in MATLAB (refer to section 4.3.4), we can see that there are many parameters to be configured. Moreover, yet those settings are for the feedforward neural network, the simplest kind of artificial neural network. Therefore, it is very hard and requires a lot of skill, experience, and patient to configure an optimal artificial neural network.

## List of Figures



## List of Tables

## References


Caroline Clabaugh, Dave Myszewski, and Jimmy Pang. 2000. Neural Networks. (2000). https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/index.html

Klaus D Goepel. 2013. AHP Introduction. (10 2013). http://bpmsg.com/ahp-introduction/

Davood Golmohammadi. 2011. Neural network application for fuzzy multi-criteria decision making problems. *International Journal of Production Economics* 131, 2 (2011), 490–504.

Lee Jacobson. 2013. Introduction to Artificial Neural Networks - Part 1. (12 2013). http://www.theprojectspot.com/tutorial-post/introduction-to-artificial-neural-networks-part-1/7

Lee Jacobson. 2014. Introduction to Artificial Neural Networks Part 2 - Learning. (3 2014). http://www.theprojectspot.com/tutorial-post/introduction-to-artificial-neural-networks-part-2-learning/8

Mohammed A Mashrei. 2012. *Neural network and adaptive neuro-fuzzy inference system applied to civil engineering problems.* INTECH Open Access Publisher.

Enrique Mu and Rachel Chung. 2013. A new approach to eyewitness police identification. In *Proceedings of the international symposium of the analytic hierarchy process (ISAHP).*

Philippe Vincke Pugnaire. 1992. *Multicriteria decision-aid.*

Bernard Roy. 2005. Paradigms and challenges. In *Multiple criteria decision analysis: State of the art surveys.* Springer, 3–24.

T.L. Saaty. 1980. *The Analytic Hierarchy Process.* McGraw-Hill, New York.

Thomas L Saaty. 2008. Decision making with the analytic hierarchy process. *International journal of services sciences* 1, 1 (2008), 83–98.

SMMT. 2017. The Society of Motor Manufacturers and Traders. (2017). https://www.smmt.co.uk/

Zbigniew Świtalski. 2016. Lectures in Multicriteria Decision Analysis. (2016). University of Zielona Gora.

Kardi Teknomo. 2006. Analytic Hierarchy Process (AHP) Tutorial. (2006). http://people.revoledu.com/kardi/tutorial/AHP/

VCA. 2013. Car fuel consumptions and emissions 2000-2013. (2013). http://data.okfn.org/data/amercader/car-fuel-and-emissions#data

Wikipedia. 2017a. Activation function — Wikipedia, The Free Encyclopedia. (2017). https://en.wikipedia.org/w/index.php?title=Activation_function&oldid=765427280 [Online; accessed 6-May-2017].

Wikipedia. 2017b. Analytic hierarchy process — Wikipedia, The Free Encyclopedia. (2017). https://en.wikipedia.org/w/index.php?title=Analytic_hierarchy_process&oldid=771550979 [Online; accessed 25-April-2017].

Wikipedia. 2017c. Artificial neural network — Wikipedia, The Free Encyclopedia. (2017). https://en.wikipedia.org/w/index.php?title=Artificial_neural_network&oldid=778640580 [Online; accessed 5-May-2017].

Wikipedia. 2017d. Decision theory — Wikipedia, The Free Encyclopedia. (2017). https://en.wikipedia.org/w/index.php?title=Decision_theory&oldid=774529628 [Online; accessed 25-April-2017].

Wikipedia. 2017e. Types of artificial neural networks — Wikipedia, The Free Encyclopedia. (2017). https://en.wikipedia.org/w/index.php?title=Types_of_artificial_neural_networks&oldid=778289315 [Online; accessed 7-May-2017].