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

Machine Learning 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 Machine Learning 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 Machine Learning 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 Machine Learning, especially Artificial Neural Network (ANN) and Adaptive Neuro - Fuzzy Inference System (ANFIS). ANN and ANFIS are two methods of Machine Learning which are the potential candidates for incorporating with MCDA.

Then, in section §4, we will design a system in which Machine Learning will play a major role in the MCDA process. With this system are two study cases which will show how effective Machine Learning can be in improving the decision making process.

Finally, we will conclude the thesis with some conclusions which summarise the points of this thesis. Because the study for this thesis still in the early stage, we will also propose some suggestions for future improvements.

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, 2017b). Each discipline has its way of studying and theorising Decision Theory, for example, a psychologist might want to investigate the behaviour 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 theorising 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 generalise 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 behavioural 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, Machine Learning techniques, 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 realising 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 realising 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 realise the goal of the decision situation. Finally, after realising the goal using the selected alternative, we get the outcomes from the realisation.

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

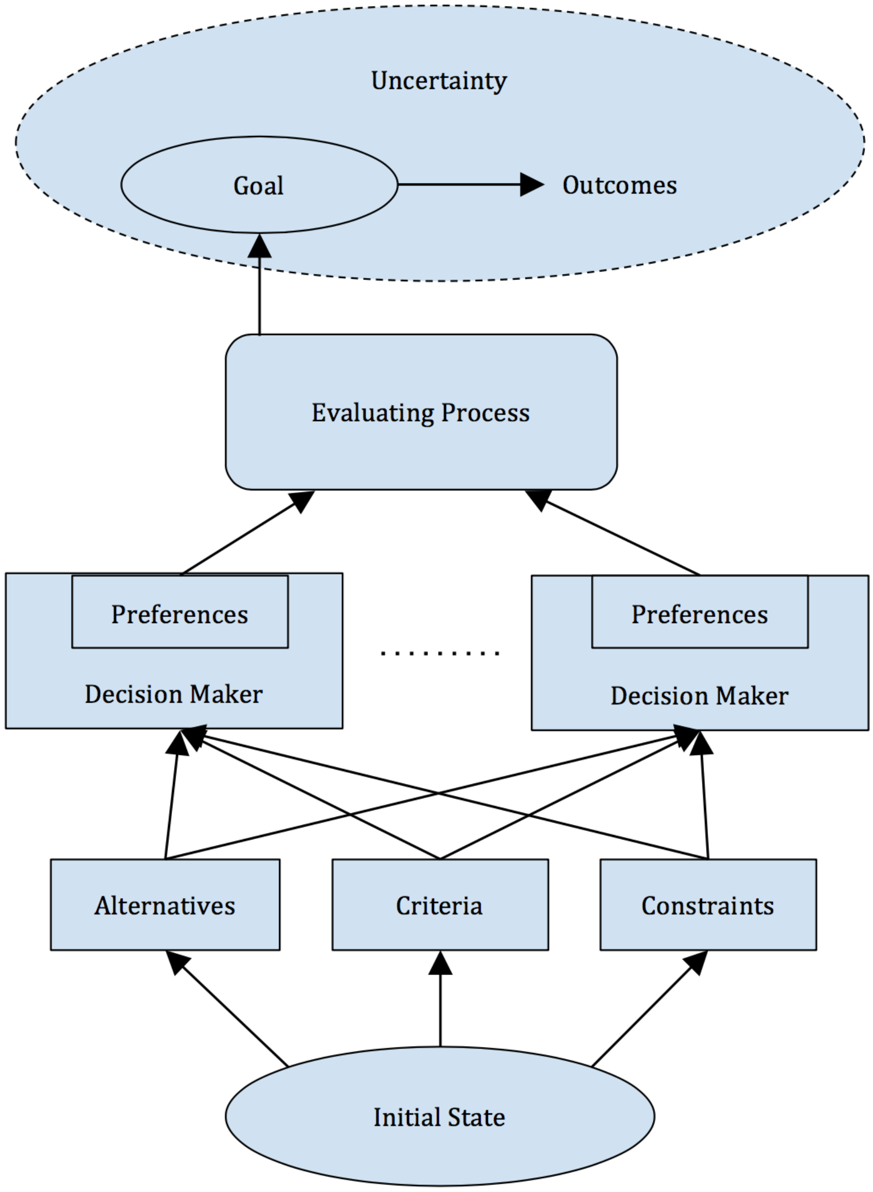

Figure 1.1. Elements of a decision-making process

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 have known that in a decision-making process, decision makers have the responsibility to choose the right decision in order to realise 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 formalised 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:

- Analysing the context of the decision-making process by various means such as identifying the actors, the possibilities of action, their consequences, or the stakes.
- Organising and structuring how the decision-making process should proceed to improve the consistency among the values underlying the goal and the final decision that realise 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 analysing 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 realise 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 generalise 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 realising 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 modelled in two ways (Roy, 2005):

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

Non-mutual exclusive: Various potential actions can conjointly be put together to realise 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 realising 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 analyse 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, 2017a):

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, 2017a), for example:

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

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

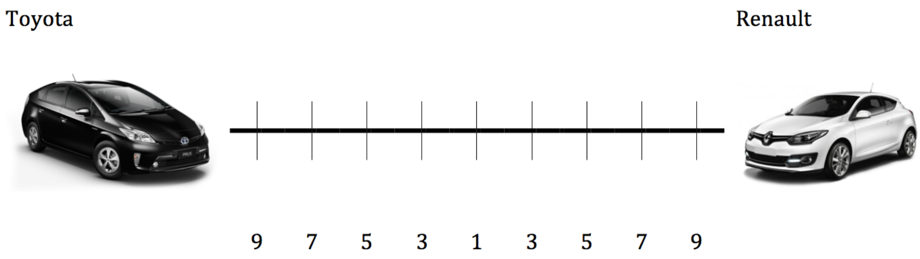

Figure 2.1. Relative scale between two car manufacturers

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)

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.

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

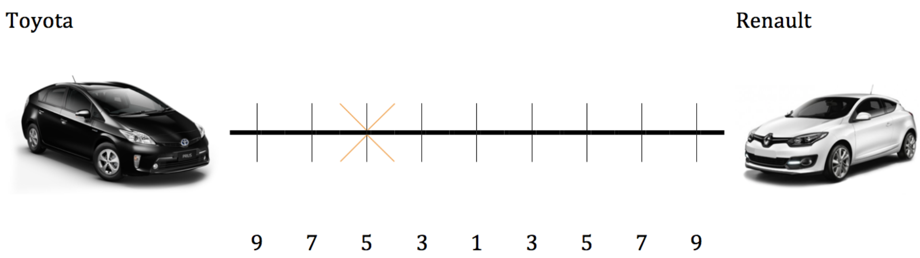

Figure 2.2. Toyota is being preferred on relative scale

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).

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	$\frac{n(n-1)}{2}$

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

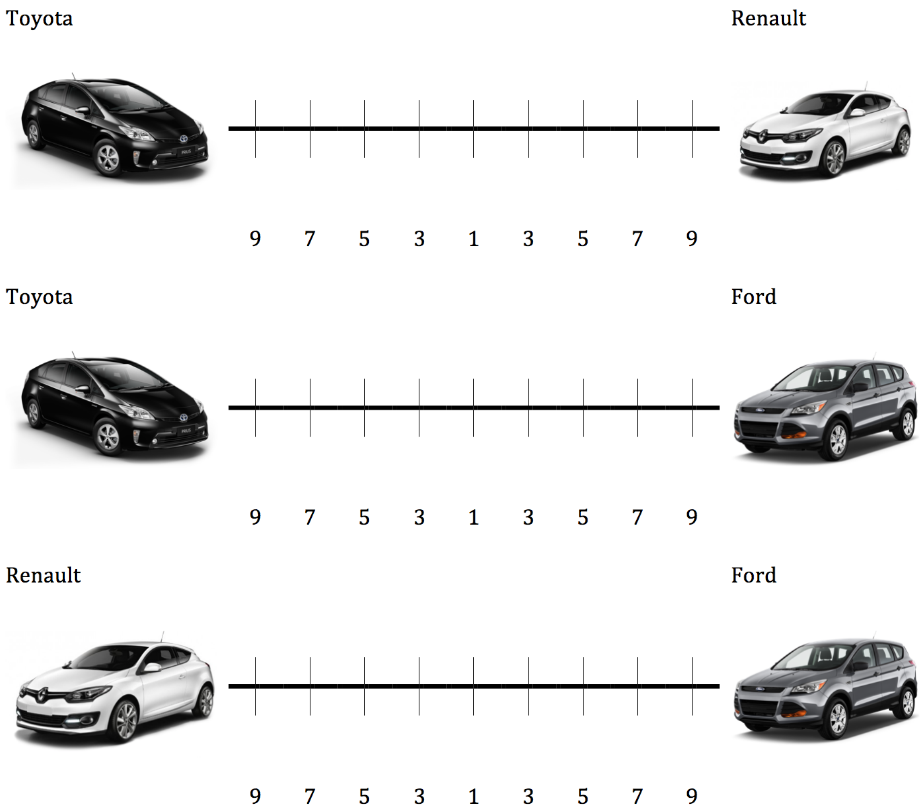

Figure 2.3. Comparisons between three car manufacturers

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**.

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

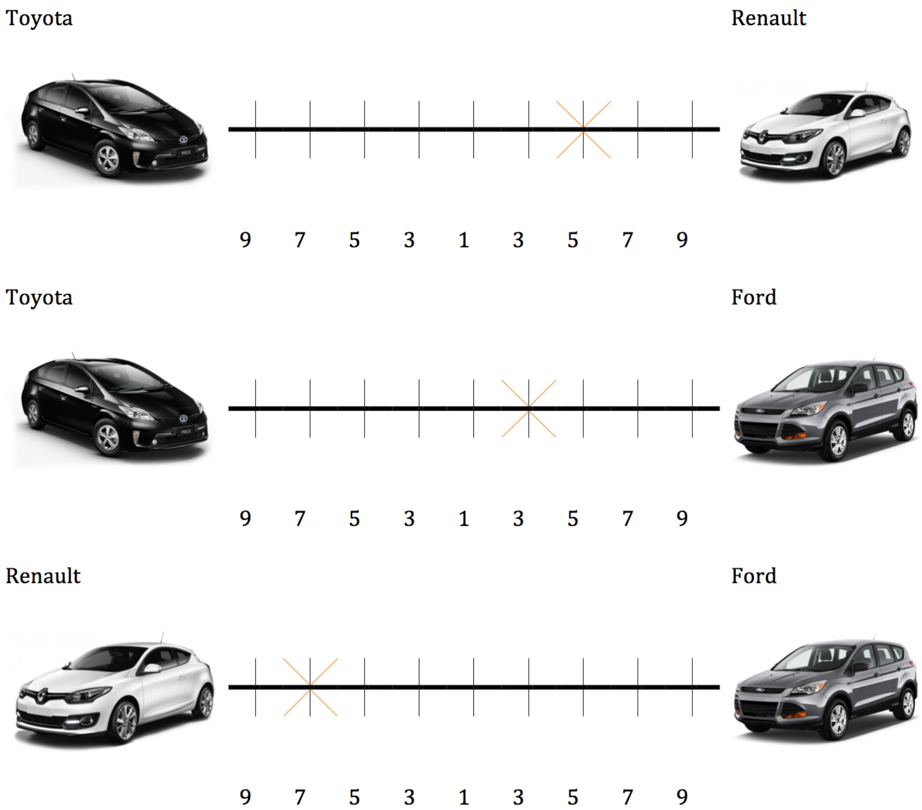

Figure 2.4. Darek's pair-wise comparisons

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.

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

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.

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

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 normalised 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 normalised principal Eigenvector with low error rate (Teknomo, 2006).

The approximation method is very easy to use. All we need to do is just to normalise 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.

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	$\frac{47}{35}$	$\frac{25}{3}$

To get the approximation of the normalised 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 normalised principal Eigenvector of the comparison matrix. Since the sum of all elements 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 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):

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

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

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. Soft Computing with Artificial Neural Network and Adaptive Neuro - Fuzzy Inference System

When using Multi-Criteria Decision Analysis, especially AHP method, to deal with decision problems of high complexity (i.e. decision problems with many variables, criteria, or alternatives) the cost of computation to find the exact solution could increase exponentially. Also, it is impossible to account all uncertainty elements of the problems too, thus reducing the effectiveness of the solution. However, there is an interesting alternative way to find the approximate solution for these particular decision problems without having high computation cost; it is by using Soft Computing.

Soft computing 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 (Jang et al., 1997). It consists of many complementary tools such as artificial neural network (ANN), fuzzy logic (FL), and adaptive neuro-fuzzy inference system (ANFIS). In this chapter, we will get into the details of ANN and ANFIS, the two tools of soft computing which we will incorporate into AHP method to solve multi-criteria decision problems.

3.1. Artificial Neural Network

In a sense, artificial neural network (ANN) simulate the architecture of the human brain. It is a type of network that sees nodes as artificial neurons. This network of artificial neurons is a computational model inspired by natural neurons. Natural neurons receive signals through synapses located on the dendrites or membrane of the neuron. When the input signal received by the neuron are strong enough (i.e. it surpasses a certain threshold), the neuron is activated and emits an output signal through the axon. This signal might be sent to another synapse, and might active other neurons in the network (Gershenson, 2003).

Because the human brain contains billions of neurons and connections, therefore the complexity of real neurons is highly abstracted when modelling artificial neurons. In fact, the basic model of artificial neuron just consists of inputs (represents synapses), which are multiplied by weights (represents the strength of the respective signals), and then computed by a mathematical function which determines the activation of the neuron. Another function computes the output of the artificial neuron. Finally, ANN combines these artificial neurons into the network to process information (Gershenson, 2003).

When we compared to traditional computing techniques, artificial neural networks have the advantage of parallel processing, distributed storing of information, low sensitivity to error, their very robust operation after training, generalisation and adaptability to new information.

In figure 3.1is an example of biological neuron.

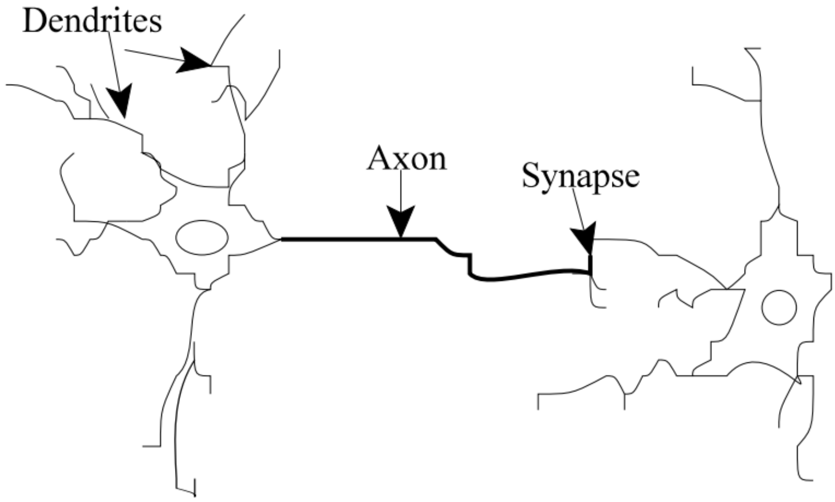

Figure 3.1. An example of biological neuron

3.2. Learning Process of ANN

To understand the learning process of ANN, we will have to examine the very basic element of the neural network, that is the artificial neuron. An artificial neuron has five main elements: inputs, weights, sum function, activation function and outputs. Inputs are information that enters the neuron from other neurons or the external world. Weights are values that express the effect of an input set or another processing element in the previous layer on this process element. Sum function is a function that calculates the effect of inputs and weights totally on this process element. This function calculates the net input that comes to a neuron. The activation function or as call as transfer function take the outcome from the sum function and then calculates the output of the neuron (Topcu and Sarıdemir, 2008).

The information is propagated through the neural network layer by layer and always in the same direction. Besides the input and output layers there can be other layers of neurons in between, which are usually called the hidden layer. The following figure 3.2 shows the structure of a typical neural network.

For a single neuron, the input to the j^{th} neuron or node are represented as an input factor a with component $a_i (i = 1$ to n), and the output b_j. The values $w_{1j}, w_{2j}, ...,$ and w_{nj} are weight factors associated with each input to the node. The weights determine the intensity of the input signal. Every input $(a_1, a_2, ..., a_n)$ is multiplied by its corresponding weight factor $(w_{1j}, w_{2j}, ..., w_{nj})$, and the node uses this weighted input $(w_{1j}a_1, w_{2j}a_2, ..., w_{nj}a_n)$ to perform further calculations. If the weight factor is positive, the weighted input $(w_{ij}a_i)$ tends to excite the node. If the weight factor is negative, the weighted input $(w_{ij}a_i)$ inhibits the node. In the initial setup of an artificial neural network, weight factors may be chosen randomly according to a specified statistical distribution. Then, these weight factors are adjusted in the development of the network or $learning process$.

The other input to the node is the node's internal threshold T_j; This is a randomly chosen value that controls the *activation* or total input of the node through the following equation (Baughman and Liu, 2014).

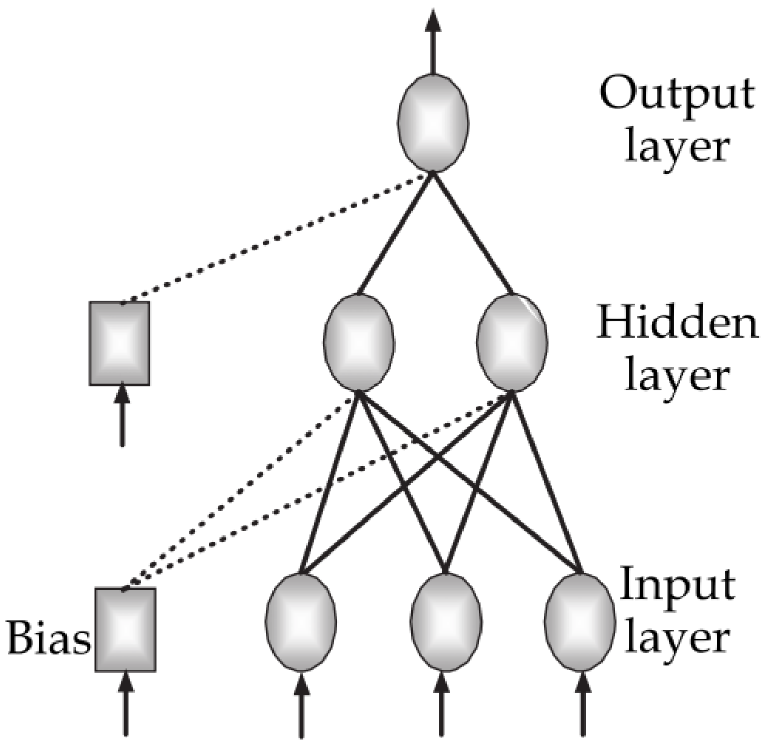

Figure 3.2. A typical neural network structure

Total Activation:

$$x_i = \sum_{i=1}^{n} (w_{ij}) * a_i - T_j \tag{3.1}$$

The total activation 3.1 depends on the magnitude of the internal threshold T_j. If T_j is large or positive, the node has a high internal threshold, thus inhibiting node-firing. If T_j is zero or negative, the node has a low internal threshold, which excites node-firing. If no internal threshold is specified, a zero value is assumed. This activity is then modified by transfer function and becomes the final output (b_j) of the neuron (Baughman and Liu, 2014).

$$b_j = f(x_i) = f(\sum_{i=1}^{n} (w_{ij}) * a_i - T_j) \tag{3.2}$$

This signal is then propagated to the neurons (process elements) of the next layer. The figure 3.3 depicts the process

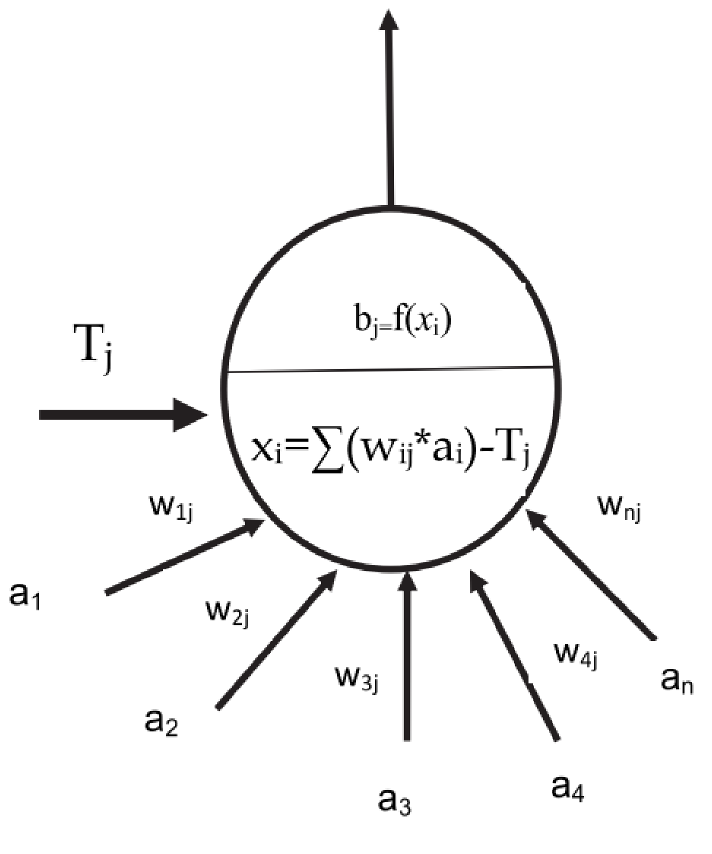

Figure 3.3. Propagation of information in an artificial neuron

A back-propagation neural network has been successfully applied in various applications such as pattern recognition (image recognition, face recognition), stock market prediction, or playing games (AlphaGo).

Learning with back-propagation technique starts with applying an input vector to the network, which is propagated in a forward propagation mode which at the ends, produces an output vector.

Next, the network evaluates the errors between the desired output vector and the actual output vector. It uses these errors to tune the connection weights and biases according to a learning rule that tends to minimise the error.

This process is referred to as "error back-propagation" or back-propagation. The adjusted weights and biases are then used to start a new cycle. A back-propagation cycle, also known as an epoch, in a neural network is illustrated in figure 3.4.

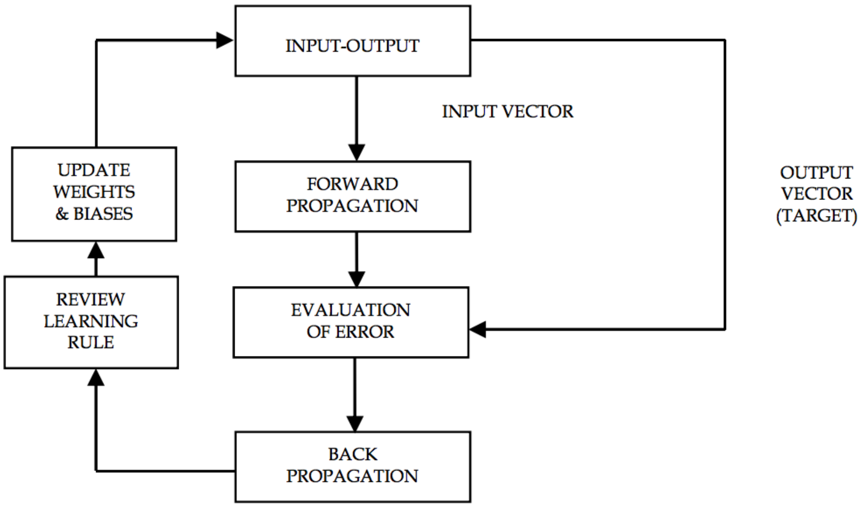

Figure 3.4. Back-propagation cycle

For some epochs, the weights and biases are tuned until the errors from the outputs are minimised.

Activation function or transfer functions are the processing units of a neuron. The node's output is determined by using a mathematical operation on the total activation of the node. These functions can be linear or non-linear. Three of the most common transfer function are depicted in figure 3.5.

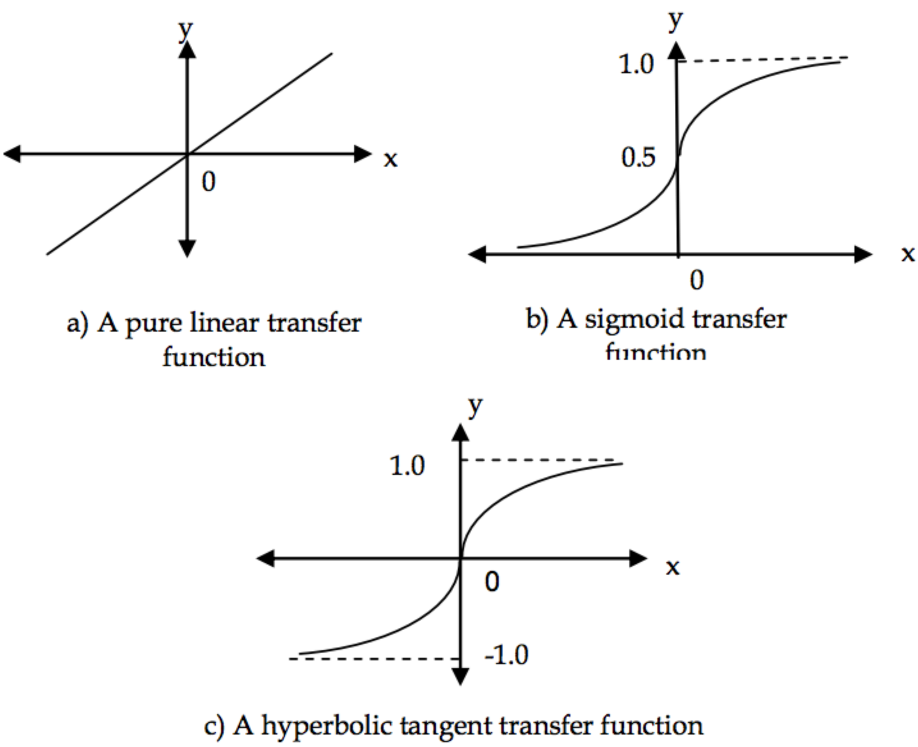

Figure 3.5. Common transfer functions

The formula of the activation functions is given as follows (MATLAB, 2017):
Pure-Linear:

$$f(x) = x \tag{3.3}$$

Log-sigmoid:

$$f(x) = \frac{1}{1 + e^{-x}} \qquad 0 \leq f(x) \leq 1 \tag{3.4}$$

Tangent sigmoid:

$$f(x) = tanh(x) = e^x - \frac{e^{-x}}{e^x + e^{-x}} \qquad -1 \leq f(x) \leq 1 \tag{3.5}$$

3.3. Generalisation

After the training is completed, the network error is usually minimised, and the network output shows reasonable similarities with the target output, and before a neural network can be used with any degree of confidence, there is a need to establish the validity of the results it generates.

A network could provide almost perfect answers to the set of problems with which it was trained, but fail to produce meaningful answers to other examples. Usually, validation involves evaluating network performance on a set of test particular that were not used for training. Generalisation (testing) is so named because it measures how well the network can generalise what it has learned and form rules with which to make decisions about data it has not previously seen.

The error between the actual and predicted outputs of testing and training converges upon the same point corresponding to the best set of weight factors for the network. If the network is learning an accurate generalised solution to the problem, the average error curve for the test patterns decreases at a rate approaching that of the training patterns. Generalisation capability can be used to evaluate the behaviour of the neural network.

3.4. Selecting the right number of hidden layers

The number of hidden layers and the number of nodes in one hidden layer are not straightforward to determine. No rules are available to find the exact number. The choice of the number of hidden layers and the nodes in the hidden layer(s) depends on the network application. Determining the number of hidden layers is a critical part of designing a neural network, and it is not straightforward as it is for input and output layers(Rafiq et al., 2001).

To determine the optimal number of hidden layers, and the optimal number of nodes in each layer, the network is to be trained using various configurations, and then to select the configuration with the fewest number of layers and nodes that still yields the minimum mean-squares error (MSE) quickly and efficiently. (Eberhart and Dobbins, 1990) recommended the number of hidden-layer nodes to be at least greater than the square root of the sum of the number of the components in the input and output vectors. (Hajela and Berke, 1991; Carpenter and Barthelemy, 1994) suggested that the number of nodes in the hidden layer is between the sum and the average of the number of nodes in the input and output layers.

The number of nodes in the hidden layer will be selected according to the following rules:

1. The maximum error of the output network parameters should be as small as possible for both training patterns and testing patterns.
2. The training epochs (number of iteration) should be as few as possible.

3.5. Pre-process and post-process of the training patterns

Neural networks require that their input and output data are normalised to have the same order of magnitude. Normalisation is very critical; if the input and the output variables are not the same order of magnitude, some variables may appear to have more significance than they do. The normalisation used in the training algorithm compensates for the order-of-differences in the magnitude of variables by adjusting the network weights.

To avoid such problems, normalisation all input and output variables is recommended. The training patterns should be normalised before they are applied to the neural network. Besides, the activation function used in the back-propagation neural network is a sigmoid function or hyperbolic tangent function. The lower and upper limits of the function are 0 and 1 respectively for sigmoid function and are -1 and +1 for hyperbolic tangent function.

The following equation (3.6) is used to pre-process the input data sets whose values are between -1 and 1 (Baughman and Liu, 2014).

$$x_{i,norm} = 2 * \frac{x_i - x_{i,min}}{x_{i,max} - x_{i,min}} - 1 \tag{3.6}$$

Where:

$x_{i,norm}$:	The normalized variable
$x_{i,min}$:	The minimum value of variable x_i (input)
$x_{i,max}$:	The maximum value of variable x_i (input)

However, for the sigmoid function the following equation (3.7) might be used:

$$O_{i,norm} = \frac{t_i - t_{i,min}}{t_{i,max} - t_{i,min}} \tag{3.7}$$

Where:

$t_{i,min}$:	The minimum value of variable t_i (input)
$t_{i,max}$:	The maximum value of variable t_i (input)

3.6. Adaptive Neuro-Fuzzy Inference System (ANFIS)

The fuzzy set theory developed by (Zadeh, 1965) provides a mathematical framework to deal with vagueness associated with the description of a variable. The commonly used fuzzy inference system (FIS) is the actual process of mapping from a given input to output using fuzzy logic.

Fuzzy logic is particularly useful in the development of expert systems. Expert systems are built by capturing the knowledge of humans: however, such knowledge is known to be qualitative and inexact. Experts may be only partially knowledgeable about the problem domain, or data may not be fully available, but decisions are still expected. In these situations, educated guesses need to be made to provide solutions to problems. This is where fuzzy logic can be employed as a tool to deal with imprecision and qualitative aspects that are associated with problem-solving (Jang, 1993).

A fuzzy set is a set without clear or sharp boundaries or without binary membership characteristics. Unlike a conventional set where object either belongs or do not belong to the set, partial membership in a fuzzy set is possible. In other words, there is a softness associated with the membership of elements in a fuzzy set (Jang, 1993). A fuzzy set may be represented by a membership function. This function gives the grade (degree) of membership within the set. The membership function maps the elements of the universe onto numerical values in the interval. The membership functions most commonly used in control theory are triangular, trapezoidal, Gaussian, generalised bell, sigmoidal and difference sigmoidal membership functions (Jang et al., 1997; Zhao and Bose, 2002; MATLAB, 2017).

As mentioned previously, the fuzzy inference system is the process of formulating the mapping from a given input to an output using fuzzy logic. The dynamic behaviour of a FIS is characterised by a set of linguistic description rules based on expert knowledge.

The fuzzy system and neural networks are complementary technologies. The primary reason for combining fuzzy systems with neural networks is to use the learning capability of the neural network. While the learning capability is an advantage from the viewpoint of a fuzzy system, from the viewpoint of a neural network there are additional advantages to a combined system. Because a neuro-fuzzy system is based on linguistic rules, we can easily integrate prior knowledge into the system, and this can substantially shorten the learning process. One of the popular integrated systems is an ANFIS, which is an integration of a fuzzy inference system with a back-propagation algorithm (Jang et al., 1997; Lin and Lee, 1996).

There is two type of fuzzy inference systems that can be implemented: Mamdani-type and Sugeno-type (Mamdani and Assilian, 1975; Sugeno, 1985). Because the Sugeno system is more compact and computationally more efficient than a Mamdani system, it lends itself to the user of adaptive techniques for constructing the fuzzy models. These adaptive techniques can be used to customise the membership functions so that the fuzzy system best models the data. The fuzzy inference system based on neuro-adaptive learning techniques is termed adaptive neuro-fuzzy inference system (Hamidian and Seyedpoor, 2010).

For an FIS to be mature and well established so that it can work appropriately in prediction mode, its initial structure and parameters (linear and non-linear) need to be tuned or adapted through a learning process using a sufficient input-output pattern of data. One of the most commonly used learning systems for adapting the linear and non-linear parameters of an FIS, particularly the first order Sugeno fuzzy model, is the ANFIS. ANFIS is a class of adaptive networks that are functionally equivalent to fuzzy inference systems (Jang, 1993).

3.7. Architecture of ANFIS

The following figure shows the architecture of a typical ANFIS with two inputs and , two rules and one output, for the first order Sugeno fuzzy model, where each input is assumed to have two associated membership functions (MFs). For a first-order Sugeno fuzzy model a typical rule set with two fuzzy if-then rules can be expressed as (Jang, 1993):

— If X_1 is A_1 and X_2 is B_1, then $f_1 = m_1X_1 + n_1X_2 + q_1$
— If X_1 is A_2 and X_2 is B_2, then $f_2 = m_2X_1 + n_2X_2 + q_2$

Where: m_1, n_1, q_1 and m_2, n_2, q_2 are the parameters of the output function.

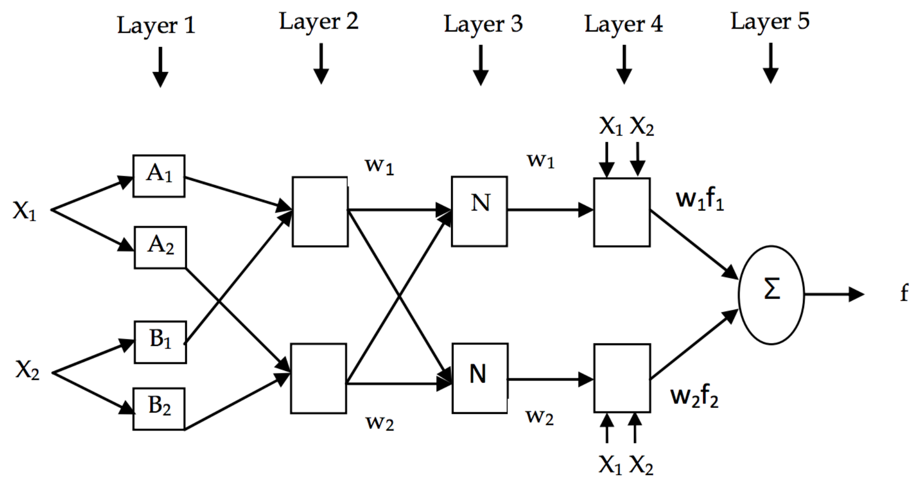

Figure 3.6. Architecture of a typical ANFIS

The architecture of the proposed ANFIS model containings five layers where the node functions in the same layer are of the same function family. Inputs, outputs and implemented mathematical models of the nodes of each layer are explained below:

Layer 1:

The node function of every node in this layer takes the form of equation (3.8).

$$O_i^1 = \mu A_i(X) \tag{3.8}$$

Where X is the input to node i, μA_i is the membership function (which can be triangular, trapezoidal, Gaussian functions or other shapes) of the linguistic label A_i associated with this node and O_i is the degree of match to which the input X satisfies the quantifier A. In the current study, the Gaussian shaped MFs defined in equation (3.9) are utilised.

$$\mu A_i(X) = exp(-\frac{1}{2} * \frac{(X - c_i)^2}{\sigma_i^2}) \tag{3.9}$$

Where c_i, σ_i are the parameters of the MFs governing the Gaussian functions. The parameters in this layer are usually referred to as premise parameters.

Layer 2:

Every node in this layer multiplies the incoming signals from layer 1 and sends the product out as in equation (3.10).

$$w_i = \mu A_i(X_1) \times \mu B_i(X_2) \qquad i = 1, 2 \tag{3.10}$$

Where the output of this layer (w_i) represents the firing strength of a rule.

Layer 3:

Every node i in this layer is a node labelled N, determine the ratio of the i-th rules's firing strength to the sum of all rules's firing strength as in equation (3.11).

$$w_i^- = \frac{w_i}{w_1 + w_2} \tag{3.11}$$

Where the output of this layer represents the normalised firing strengths.

Layer 4:

Every node i in this layer is an adaptive node with a node function in the form of equation (3.12).

$$O_i^4 = w_i^- f_i = w_i^- (m_i X_1 + n_i X_2 + q_i) \qquad i = 1, 2 \tag{3.12}$$

Where w_i^- is the output to layer 3, and m_i, n_i, q_i is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

Layer 5:

There is only a single node in this layer that computes the overall output as the weighted average of all incoming signals from layer 4 as in equation (3.13).

$$O_i^5 = \sum_i w_i^- f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \qquad i = 1, 2 \tag{3.13}$$

3.8. Learning Process of ANFIS

As mentioned earlier, both the premise (non-linear) and consequent (linear) parameters of the ANFIS should be tuned, utilising the so-called learning process, to optimally represent the factual mathematical relationship between the input space and output space. Normally, as a first step, an approximate fuzzy model is initiated by the system and then improved through an iterative adaptive learning process.

Basically, ANFIS takes the initial fuzzy model and tunes it by means, of a hybrid technique combining gradient descent back-propagation and mean least-squares optimisation algorithms. At each epoch, an error measure, usually defined as the sum of the squared difference between actual and desired output, is reduced. Training stops when either the predefined epoch number or error rate is obtained. There are two passes in the hybrid learning procedure for ANFIS. In the forward pass of the hybrid learning algorithm, functional signals go forward till layer 4, and the consequent parameters are identified by the least squares estimate.

In the backwards pass, the error rates propagate backwards, and the premise parameters are updated by the gradient descent. When the values of the premise parameters are learned, the overall output in equation (3.14) can be expressed as a linear combination of the consequent parameters (Jang, 1993).

$$f = w_1/(w_1 + w_2)f_1 + w_2/(w_1 + w_2)f_2 = w_1^- f_1 + w_2^- f_2$$

$$= (w_1^- X_1)m_1 + (w_1^- X_2)n_1 + (w_1^-)q_1 + (w_2^- X_2)m_2 + (w_2^- X_2)n_2 + (w_2^-)q_2 \quad (3.14)$$

Which is linear in the consequent parameters $m_1, n_1, q_1, m_2, n_2, q_2$.

4. Case Studies

Conclusions

List of Figures

List of Tables

References

D Richard Baughman and Yih An Liu. 2014. *Neural networks in bioprocessing and chemical engineering.* Academic press.

William C Carpenter and Jean-Francois Barthelemy. 1994. Common misconceptions about neural networks as approximators. *Journal of computing in civil engineering* 8, 3 (1994), 345–358.

RC Eberhart and RW Dobbins. 1990. Neural network PC tools: a practical guide. (1990).

Carlos Gershenson. 2003. Artificial Neural Networks for Beginners. *CoRR* cs.NE/0308031 (2003). http://arxiv.org/abs/cs.NE/0308031

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

P Hajela and L Berke. 1991. Neurobiological computational models in structural analysis and design. *Computers & Structures* 41, 4 (1991), 657–667.

D Hamidian and SM Seyedpoor. 2010. Shape optimal design of arch dams using an adaptive neuro-fuzzy inference system and improved particle swarm optimization. *Applied Mathematical Modelling* 34, 6 (2010), 1574–1585.

J-SR Jang. 1993. ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics* 23, 3 (1993), 665–685.

Jyh-Shing Roger Jang, Chuen-Tsai Sun, and Eiji Mizutani. 1997. Neuro-fuzzy and soft computing, a computational approach to learning and machine intelligence. (1997).

Chin-Teng Lin and CS Lee. 1996. *Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems.* Prentice-Hall, Inc.

Ebrahim H Mamdani and Sedrak Assilian. 1975. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies* 7, 1 (1975), 1–13.

MATLAB. 2017. Neural Network Toolbox User's Guide for Use with MATLAB. (2017). https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf

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.*

MY Rafiq, G Bugmann, and DJ Easterbrook. 2001. Neural network design for engineering applications. *Computers & Structures* 79, 17 (2001), 1541–1552.

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.

Michio Sugeno. 1985. *Industrial applications of fuzzy control.* Elsevier Science Inc.

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/

Ilker Bekir Topcu and Mustafa Sarıdemir. 2008. Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic. *Computational Materials Science* 41, 3 (2008), 305–311.

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

Lotfi A Zadeh. 1965. Fuzzy sets. *Information and control* 8, 3 (1965), 338–353.

Jin Zhao and Bimal K Bose. 2002. Evaluation of membership functions for fuzzy logic controlled induction motor drive. In *IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the]*, Vol. 1. IEEE, 229–234.