A NEW APPROACH FOR NON-DESTRUCTIVE DETERMINATION OF CONCRETE COMPRESSIVE STRENGTH BY USING RADIAL BASIS FUNCTION NETWORK

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By
SADDAM NASER ATIYAH AL-GBURI

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Civil Engineering

NICOSIA, 2018

SADDAM ALGBURI

A NEW APPROACH FOR NON-DESTRUCTIVE DETERMINATION OF CONCRETE COPRESSIVE STREGNTH USING RADIAL BASIS FUNCTION NETWORK

NEU

2018

A NEW APPROACH FOR NON-DESTRUCTIVE DETERMINATION OF CONCRETE COMPRESSIVE STRENGTH BY USING RADIAL BASIS FUNCTION NETWORK

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By
SADDAM NASER ATIYAH AL-GBURI

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Civil Engineering

NICOSIA, 2018

Saddam Naser Atiyah AL-GBURI: A NEW APPROACH FOR DETERMINATION OF CONCRETE COMPRESSIVE STRENGTH BY USING RADIAL BASIS FUNCTION NEURAL NETWORK

Approval of Director of Graduate School of Applied Sciences

Prof. Dr.Nadire CAVUS

We certify this thesis is satisfactory for the award of the degree of Masters of Sciences in Civil Engineering

Examining Committee in Charge:

Assoc. Prof. Dr. Gözen Elkıran	Committee Chairman, Department of Civil Engieering, NUE
Asst. Prof. Dr. Boran Şekeroğlu	Department of Information System Engineering, NEU
Asst. Prof. Dr. Pınar Akpınar	Supervisor, Department of Civil, Engineering, NEU

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, last name: Saddam Algburi

Signature:

Date:

To my parents...

ACKNOWLEDGMENTS

All praises and thanks to Allah. It is by His grace that I have been able to access this point in my life.

I would like to express my sincere gratitude to my supervisor, Asst.Prof. Dr. Pinar Akpinar who has supported and directed me with her vast knowledge and also for her patience that ensured the completion of this thesis.

I dedicate my success to my parents, who always supported me in my studies.

Finally, I thank to my friends who supported me in every possible way.

ABSTRACT

Concrete's compressive strength is widely studied in order to understand many qualities and the grade of concrete mixture. Conventional civil engineering tests involve time and resources consuming laboratory operations which results in the deterioration of concrete samples. Proposing efficient non-destructive models for the prediction of concrete compressive strength will certainly yield advancements in concrete studies.

In this study, the efficiency of using radial basis function neural network (RBFNN) which is not common in this field, is studied for the concrete compressive strength prediction. Complementary studies with back propagation neural network (BPNN), which is widely used in this field, has also been carried out in order to verify the efficiency of RBFNN for compressive strength prediction. A total of 13 input parameters, including novel ones such as cement's and fly ash's compositional information, have been employed in the predictions models with RBFNN and BPNN, since all these parameters are known to influence concrete strength. Three different train:test ratios were tested with both models, while different hidden neurons, epochs and spread values were introduced in order to determine the optimum parameters for yielding the best prediction results. Prediction results obtained by RBFNN are observed to yield satisfactory high correlation coefficients and satisfactory low mean square error values when compared with the results in the previous studies, indicating the efficient of proposed model.

Keywords: Compressive strength of concrete; non-destructive strength prediction; radial basis function neural network; back propagation; factor affecting concrete strength

ÖZET

Beton basınç dayanımı, beton nitelikleri ve dayanım sınıfının anlaşılması için literatürde yürütülen pek çok kapsamlı çalışmaya konu olmuştur. Geleneksel inşaat mühendisliği testleri, maliyetli kaynak ve süre kaybına mal olan laboratuvar çalışmalarını içermelerinin yanı sıra, test edilen numunenin tahribata uğramasına ve dolayısıyla yeniden kullanılamamasına neden olur. Betonun basınç dayanımının belirlenmesi için tahribatsız ve etkili tahmin yöntemlerinin önerilmesiyle, bu alandaki beton çalışmalarında önemli bir ilerleme kaydedilmesi beklenmektedir.

Bu tez çalışmasında, beton basınç dayanımı konusunda literatürde kullanımı yaygın olmayan Radyal Tabanlı Fonksiyon Yapay Sinir Ağının kullanıldığı önerilen tahmin yönteminin verimliliği araştırılmıştır. Radyal Tabanlı Fonksiyon Yapay Sinir Ağı verimliliği ile yapılan basınç dayanımı tespitlerinin verimliliği, bu alanda sıklıkla kullanılan Geri Yayılım Algoritması ile hazırlanan ikinci bir model ile kıyaslanarak araştırılmıştır. İki modelde de, beton karışımında kullanılan çimento ve uçucu külün içeriğindeki kimyasal bileşenler gibi, daha önce benzeri çalışmalarda kullanılmamış yeni parametreleri de içeren toplamda 13 giriş parametresi kullanılmıştır. Çalışmalarda, üç farklı eğitim:test veri dağılımı, farklı sayılarda gizli nöronlar, farklı dağılım ve iterasyon değerleri ile birlikte kullanılarak doğruluğu yüksek basınç dayanımı tahmin sonuçları verecek optimum değerler tespit edilmiştir. İlgili literatürdeki çalışmalarının sonuçları ile kıyaslandığında, Radyal Tabanlı Fonksiyon Yapay Sinir Ağı ile yapılan beton basınç dayanımı tahminlerinde elde edilen korelasyon katsayısı ile karesel ortalama hata değerlerinin başarılı düzeyde oldukları tespit edilmiştir.

Anahtar Kelimeler: Beton basınç dayanımı; tahribatsız beton dayanım tahmini; radyal tabanlı fonksiyon yapay sinir ağı; geri yayılım algoritması; beton dayanımını etkileyen faktörler

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN:	Artificial Neural Network
MLP	Multilayer Perceptron
BPNN:	Back Propagation Neural network
RBFNN:	Radial Basis Function Neural Network
BPLA	Back Propagation Learning Algorithm
R:	Correlation Coefficient
MSE:	Mean Square Error
CaO:	Calcium Oxide
SiO_2 :	Silicon Dioxide
Al_2O_3 :	Aluminium Oxide

CHAPTER 1
INTRODUCTION

1.1 Overview on concrete and its Compressive strength

Concrete is the fundamental construction material around the world due to its important advantages such as its strength, easy workmanship for its manufacture, high availability and low cost of its constituent materials. Among these, its outstanding performance in compression, to withstand the compressive stress makes concrete primary building material. Determining the compressive strength of concrete for any structural element is also important in design process. Conventional concrete is made of several materials including cement water and different types of aggregates. The strength of concrete is affected by various factors such as water - cement ratio, types and amount of aggregates as well as cement type and content. Cement types are classified mainly according to their chemical compositions. Therefore, changing the chemical composition of cement yield to the formation of different concrete compounds and hence strength value varies. Therefore, chemical composition of all constituents plays important role in concrete strength.

Measuring the major feature of concrete is a critical stage in design process. Concrete compressive strength can be measured using two types of tests. Conventional test is the most widely used test for concrete compressive strength and it called as compression test. This test is held in laboratory using newly manufactured concrete specimen or extracted core from an existing structure. In order to measure the strength of concrete core or specimen, this test requires applying compressive stress on concrete sample (specimen or core) until the concrete sample fails under load. Thus its destructive and uneconomical. Also, the procedure of this test requires a considerable amount time, especially in preparing new concrete specimen (casting and hardening concrete specimen) therefore it is considered as a time consuming test. The other type of concrete compressive test consists of equipment such as ultrasonic pulse velocity device. This type of test measures the compressive strength of concrete through defining the amount of cracks, voids, porosity and corrosion in concrete structure. However, this test is classified as non destructive but its cost effective and time consuming test.

1.2 Definition of the problem

It is seen that an efficient tool to predict concrete compressive strength in non destructive, non-time and resources consuming way will be beneficial for both concrete industry and researchers working in this field. Artificial neural networks (ANNs) have been used in civil engineering by many researchers to various prediction tasks, especially in concrete compressive strength estimation task (Douma et al., 2016) (Khashman and Akpinar, 2017). ANN is a data processing model which tries to imitate the way of human biological brain works. However, the previous ANNs models for compressive strength have observed to not considered all critical input parameters. Also the radial basis function neural network (RBFNN) is not commonly used in concrete compressive strength prediction, even though its training is faster than the conventional neural network learning (Du and Swamy, 2013).

1.3 Objective and Scope of the Study

Design Based on previously mentioned problems, this study aims to employ Radial Basis Function Network (RBFNN) as an alternative in regression task for studying concrete compressive strength prediction in a different way. Detailed dataset based on 326 experimental cases with comparable conditions are formed from the related-literature. 13 input parameters including some parameters that will be used for the first time in the related literature are considered for prediction concrete compressive strength. Moreover, strength prediction will be studied with three selected train: test ratios with varying hidden neurons and spread values.

1.4 Significance of the Study

The importance of this study is due to the fact RBFNN has not yet been used in prediction of concrete compressive strength, thus it is employed for the first time in order to predict the concrete compressive strength considering influence of several material and manufacture criteria in a detailed and systematical way. This study also considers some new input parameters that affects the concrete strength which have not been used before in the related literature. These parameters include calcium oxide, silicon dioxide and aluminum oxide contents of fly ash and cement. Moreover, this study provides a comparison of compressive strength prediction performance of RBFNN with BPNN in a detailed way. In these aspects, a significant contribution will be made to the related literature.

1.5 Structure of the Thesis

Chapter one describes in detailed the problem that is detected and studies in this thesis, as well as the scope, objective and the significance of the study. Chapter two includes theoretical background about concrete compressive strength. Chapter three explains artificial neural networks and the most common types of these networks. Previous studies on concrete compressive strength prediction with artificial neural networks are discussed in chapter four, whereas, the proposed models and the procedure of implementation for this study are described in chapter five.

Results and discussion are presented in chapter six, whereas conclusion and recommendation are provided in Chapter seven.

CHAPTER 2
LITERATURE REVIEW ON CONCRETE

2.1 Overview Concrete as Construction Material

Concrete and steel are the most commonly used construction materials around the world. These materials are considered valuable materials due to their behavior. Concrete is well known to have high compressive strength and steel has high resistance to tension. Steel can show higher performance than concrete in both compression and tension. Use of steel only as construction material is uneconomical thus people are tended to use concrete in construction. Concrete has several advantages such as easy workmanship, raw materials of concrete are available everywhere and it is economical comparing to other materials such as steel. Also, concrete has ability to withstand high temperatures thus can be considered as fire resistant material. Conventional concrete is usually composed of cement, water and different type of aggregates. Cement is grey powder which is considered as the main ingredient of concrete. Aggregates are classified according to their size. If the size of aggregate particles is less than five mm then these aggregates are fine, whereas, the particles that have bigger size than those of fine are called coarse aggregates. In specific types of concrete, in order to have specific properties of concrete, admixtures are added to concrete mixture. The most common admixtures are set retarders, plasticizers and accelerators. In some cases, fly ash also can be used as admixture. The set retarders admixtures are used to provide postponement for the setting of concrete. The accelerators are motivating the hydration reaction and cause increasing in hydration temperature, and then eventually affect strength of concrete at early age. Plasticizers or water reducer are utilized to decrease water content. However, most concrete mixtures contain cementitious materials. These materials can be defined as materials that are as fine as the least cement particles. Furthermore, these materials can be formed from products from other processes or natural materials. Fly ash and silica fume are known as pozzlans. Fly ash does not have any cementitious properties except when it uses with Portland cement, it can react and form compounds. These compounds influence the strength of concrete (Neville, 2012), (Shetty, 2005), (Neville and Brooks, 2010), (Halstead, 2006) & (Ramezanianpour, 2014).

2.2 Cement and Cementitious Materials

In production of Portland Cement, clay and limestone are the chief raw materials that are fed into the rotary kiln and heated to form a new materials called the clinker. This clinker consists of "tricalcium silicate (C3S), dicalcium silicate (C2S), tricalcium aluminate (C3A) and tetracalcium aluminoferrite (C4AF)" which are the main compounds. The final product of the cement when mixed with water results in stiffening of fresh cement paste. The reaction with water yields hydration products responsible for development of cement microstructure. The hydration process yields Ettringite first and other hydration products Moreover, this product known to have different volume thus it causes expansion. Later on, lime (CH) and C-S-H gel start to form. Ettringite converts into monosulfate aluminate when the content of C3A exceeds the sulfate in the cement compound. However, C-S-H is calcium silicate hydrates gel constitute about 50-60 % of the hydrated cement paste by volume, and with large surface area which is responsible for the strength of the concrete. The compounds occupy the pore spaces between the particles of cement compounds. The portlandite is a long crystal that constitute about 20-25 % of the hydrated cement paste by volume and with less contribution to the strength due to it's low surface area (Neville and Brooks, 2010) & (Mehta and Monteiro) Materials that have high fineness as least cement cement particles are called cementitious materials. Fly ash, silica fume, ground granulated blast-furnace slag and filler are the most common cementitious materials. Fly ash can be defined as pulverized-fuel ash. It is the precipitated ash from exhaust gases of power plants. Fly ash is considered as one of the most common pozzlans cementitious materials. Fly ash known to have no cementitious properties, which reacts with water and precipitate portlandite to form denser cementitious compounds and these compounds can affect the strength of concrete. Fly ash has very fine particles size. The source of fly ash influences its composition. Moreover, fly ash of class C and F are the most used in construction. With respect to American Society for Testing and Materials (ASTM C 618) that class F can be made by burning anthracite or bituminous coal under normal condition. Class F has pozzolanic features.in other words, in itself has no cementitious properties but in the existence of moisture it can react with lime and form products that have cementitious features. The pozzolanic interaction of fly ash happens slowly at ordinary temperatures. Fly ash can be used as admixture or additives for blended cement **.** In general, fly ash is a cementitious material and it can be replaced with a part of the ordinary cement. This replacement can be made in two methods, either by adding a part of blended cement or in the concrete mixer. However, the quantity and quality of fly ash influence the concrete in both states. In other word, the selection of the appropriate proportion fly ash and type ensure good quality mixture. Furthermore, fly ash can effect fresh and hardened state of concrete. In fresh state, the fly ash can be influenced by the following:

- Workability

Fly ash particles are very fine and have spherical shape which allow greater workability for equivalent W/C ratio. In addition, in fly ash concrete mixture the volume of both cement and fly ash are generally greater than the cement content concrete mix with no fly ash. Furthermore, increasing the solid over water ratio leads to form a paste that improves the plasticity of mixture.

- Bleeding

For a given workability, bleeding can be reduced with increase in binder fineness and decrease in water to binder ratio.

- Time of Setting

The impacts of fly ash on setting time of concrete depend on the type and quantity of used fly ash. Generally, all class F fly ashes yield to reduce the temperature of hydration and this leads to delay setting time and early gain of concrete strength. Fly ash influences hardened concrete through the strength and rate of strength gain. The main reason for using fly ash in concrete is due to the fact that at early ages fly ash concrete promotes low heat comparing to normal concrete. In fly ash concrete, the pozzolanic interaction starts at a slow rate but it provides equal or greater ultimate strength than conventional concrete. Fly ash concrete can be defined as replacing a portion of the used cement with fly ash having proper pozzolanic features in order to maintain ultimately greater strength than typical concrete. The strength of fly ash concrete depends on the properties of the used fly ash. The general observance that fly ash concrete has lower early strength than typical concrete (Shetty, 2005),(Mehta and Monteiro , 2006), (Halstead, 2006) & (Ramezanianpour, 2014).

2.3 Concrete Strength

The strength scientifically can be defined as it’s the ability of an object to withstand an applied force without deformation. Concrete can significantly perform higher in compression than tension, so concrete strength is usually represented by the compressive strength which is widely considered as the most valuable feature of concrete. Most of buildings and other structures are subjected to various types of loads such compression and tension. Concrete is mainly used to provide resistance to the compressive stresses. Also, it’s used as a measure of other features such as elastic modulus and impermeability. Concrete quality usually depends on the concrete compressive strength. The strength of concrete is derived from the strength of bond that located between cement paste and aggregate particles. There is an inverse relationship between strength and porosity and direct relationship between strength and age. Usually when cracks are formed this referrer to the occurrence of failure. This failure might be due to various factors (Neville, 2012), (Shetty, 2005) & (Neville and Brooks, 2010).

2.3.1 Factors affecting concrete strength

Concrete’s strength is influenced by various factors including porosity, water/cement ratio, cement type and amount, type and shape of aggregates, humidity, and level of compaction. Also the bond that exists between cement paste and aggregates particles has effects on concrete compressive and tensile strength. The voids or pores that have not filled by hydration products have significant influences on concrete strength, especially those with size of larger than 50 nm are widely recognized to increase the permeability (Neville, 2012) (Neville and Brooks, 2010) & (Mehta and Monteiro, 2006).

Transition zone is considered as a critical factor, it is also called as interface zone or aggregates bond. The presence of pores and its sizes also affects the strength with the zone. The strength of this zone is improved within the time and thus concrete strength is improved. This improvement is occurred due to the formation of calcium silicate hydrated gel in existing voids. Moreover, Calcium Hydroxide (CH) has less surface area compared calcium silicate hydrated gel, therefore provide less binding property. Calcium Hydroxides are behaved as promote cracking zones because these Hydroxides are formed in oriented layers. Also, strength of the interface zone is influenced by size of existing voids, aggregate texture (crashed) and direction of the formed Calcium Hydroxides. (Neville, 2012), (Shetty, 2005), (Neville and Brooks, 2010) & (Mehta and Monteiro, 2006).

75 % of the concrete volume is occupied by aggregates, therefore it has major influences on concrete strength. The quality of Concrete is affected by aggregate's size, shape and texture. The surface texture has effects on the interface zone. the shape and surface texture of aggregates have considerable influences on concrete flexural strength However, the interface zone is improved when aggregates particles that have a crashed and tough surface are used. (Neville, 2012), (Shetty, 2005) & (Mehta and Monteiro, 2006).

During consolidation, poor crashed aggregates have potential to be segregated and eventually this segregation can affect transition bond negatively. Defining the proper water to binder ratio can result in satisfactory ratio of tensile to compressive strength. Moreover, use crushed aggregate in concrete mixture may lead to increase the tensile and compressive strength (Shetty, 2005).

2.4 Compressive Strength Test

Concrete can show a significant performance in compression. Therefor the compressive strength is defined as a major feature of concrete. Concrete's compressive strength is studied in order to understand many quantities and the grade of concrete mix. The concrete compressive strength is a measure of concrete ability to withstands the applied stress. The benefit of defining the concrete's compressive strength is to determine that the concrete mixture and to be used in designing structural members and other purposes. Concrete compressive strength test can be made for different purposes but the main two objectives of testing are quality control and compliance, now called conformity with specifications. However, type and size of test specimen, curing condition, hardness of the testing machine and rate of applying load influence the compressive strength test outcomes. Thus testing must be carried out according to a single code. The commonest test of concrete compressive strength involves using of a destructive machine that requires Applying load on the concrete specimen until the specimen break in order to identify the compressive strength, see figure 2.1. Furthermore, this test can be held in laboratory by using cylindrical specimen or cubic specimen depending on the testing code. in United states, Canada, Australia, and New Zealand the test specimen should be cylindrical specimen that mostly have diameter of 150 mm and a height of 300 mm. Whereas in most of the European countries the specimen should be in cubic form and must have dimension of 150 mm from each side. The most common standard test methods for compressive strength of concrete are ASTM C 39 (standard test method for compressive Strength of cylindrical concrete specimens) and BS EN 12390-3:2009 it is british and european standard test method for compressive strength of cubic concrete specimens. Nevertheless, compressive strength test also can be carried out using extracted cores form structure. The standard test method for obtaining and testing compressive strength of structural core and others strength is ASTM C42/C42M. The compressive strength test might be considered as a destructive test since it requires destroying or breaking the specimen. Furthermore, the tested specimen (broken specimen) can not be used for other test thus it can also be called uneconomical test (Neville, 2012), (Shetty, 2005), (Neville and Brooks, 2010) & (Mehta and Monteiro, 2006).

Figure 2.1: Classical test of concrete compressive strength

(a)

(b)

Figure 2.2: (a) Cylindrical and (b) cubical specimens with different types of failure

CHAPTER 3
NEURAL NETWORKS

3.1 Introduction

The nature of human brain structure is complex and precise and because of these properties of brain structure, makes brain to have capability to perform various difficult assignments. Scientifically, human brain uses biological neurons to perform these tasks. The exact number of neurons is unknown but these neurons can be approximated around billions of neurons (linked with each other). The principle of artificial neural networks (ANNs) is inspired by the mechanism of human biological brain; thus artificial neural networks can be defined as an imitation of the structure and the function of the biological brain. ANN can be used for various applications, such as pattern recognition and classification of the data by training operations (Haykin, 2009) & (Du and Swamy, 2013).

3.2 Historical Background

The first insight of artificial neural networks was emerged in 1943 by McCulloch and Pitts. the authors proposed that logic function can be made through modeling the neurons as threshold system. After some time, Hebbian learning rule was suggested in 1949 by Hebb, who showed that synaptic (weights) that are located between neurons can be influenced by learning. Moreover, Hebbian rule defines the amount of weight that must be raised or reduced in proportion to the product of their activation (Haykin, 2009) , (Du and Swamy, 2013) & (Kriesel, 2007).

The first neural network model (Perceptron) was invented by Rosenblatt in 1957. The perceptron consists of only one single neuron and threshold transfer function. Two years later, Hoff and Widrow developed a model called as adaline (adaptive linear element) which was based on McCulloch and Pitts theory. The proposed model (adaline) brought with training or learning method which known as Least Mean Square (LMS) technique. The activation function in perceptron was threshold, whereas in Adaline was linear while. At the same period, madaline paradigm was developed which was an extension for adaline, madaline consists of set of adaline parallel as its input layers. However, multilayer adaline or madaline was the first model employed for real problems. Both adaline and madaline have the capability of only to find solution for linear separable problem due to the type of transfer function (linear transfer function). Ph.D. student, (Paul J. Werbos), invented back-propagation algorithm for learning neural network in 1974. John Hopfield proposed new model of neural network known as recurrent or Hopfield network which consists of single layer use for information storage and for solving optimization tasks in 1982. Hopfield works with the hebbian learning rule (Du and Swamy, 2013) & (Kriesel, 2007).

Rumelhart, Hinton and Williams have suggested back-propagation algorithm to be used along with the multilayer perceptron (MLP) model for solving non linear separable problems in 1986. Two years later, radial basis function neural network(RBFNN) was developed as universal approximator by Broomhead and Lowe. Pearl introduced the Bayesian or Bayes neural network was proposed by Pearl in 1985 (Du and Swamy, 2013) & (Kriesel, 2007).

3.3 Human Brain “Biological Neural Network”

Human brain contains a lot of neurons which can be approximated by billions. and each single neuron in human brain is linked to thousands near by neurons. The essential anatomic and effective part in human brain is a nerve cell, the nerve cell is also known by (nervous system) or neuron. The neuron can be defined as an extension of the normal cell with an axon and dendrites. Moreover, the biological neuron composed of dendrites, soma, axon, and the weight or synapse . Figure 3.1 shows the components of the biological neuron. As shown in the figure that nucleus is located in middle of the soma. The soma generates input through gathering all the arriving signals. Also from the figure it can be seen that dendrites are directly related to the cell body (Soma). The function of dendrites is to receive signals from other neurons and transmit it to the soma. The output path to other neurons is represented by the axon which is branching into main and secondary branches to link the dendrites and next neuron’s soma. At the end of each branch of axon there are structures known as synapses. These synapses can be referred as the connection points between two different neurons. The synapses connections can be inhibitory or excitatory. These synapses uses to transit the signal between neurons in two directions. These signals are electrochemically transmitted in the junction points. The potential in the synapses changes based on the chemical materials being transmitted between the neurons. The potential affects soma and causes its activation if the received signals by dendrites are strong sufficient to flame the neuron. Moreover, if the received signals by dendrites are strong sufficient to flame the neuron, then the neuron neuron will transmit another signal by the axon to near by neurons in the same process. The signal is going also to be received by the connected dendrites, so can fire next neurons. (Xiao, 1996). In other words, the neurons collect signals from other neurons through fine structures known as dendrites and these neurons can be activated or deactivated based on the received electrochemical signals. For instance, When the sum passes a threshold or certain value, then the neuron will fire (activated) and the signal go along to the neighboring neurons through the axon which splits into thousands of branches known as synapses. But in the case that the sum is less than the activation value, then no neuron be fired and this results in deactivated neurons (Haykin, 2009), (Du and Swamy, 2013) (Kriesel, 2007), (Xiao, 1996)& (Fausett, 1994).

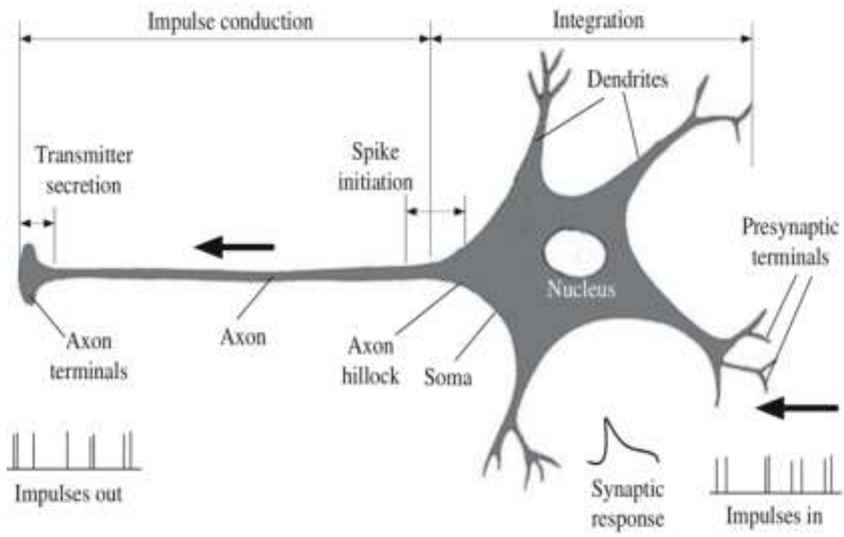


Figure 3.1: Architecture of human biological neuron. (Du and Swamy, 2013)

3.4 Neural Networks

Artificial Neural Networks (ANNs) can be defined as a data processing model which tries to imitate the way of human biological brain works. There are many nodes (neurons) that linked or connected with each other through lines (weight) in ANNs; these neuros work with each other to find solution for specific tasks. The processes of neural networks (NN) consist of two steps; the first step is training or learning of neural network through use of data (examples) which can be carried out by using learning algorithm. Whereas, the second step is recalling; this step means testing the trained network for new given data (examples). However, the structure, properties of neurons and training methods are factors that affects classification of neural networks or specify the type of neural network. The most common types of neural network are listed below. (Haykin, 2009), (Du and Swamy, 2013), (Kriesel, 2007), (Tino et al., 2015) & (Gurney, 1997).

3.5 Neural Networks Types

1. A Feed-Forward Neural Networks (FFNNs): are the most commonly used type of neural networks. FFNNs consist of three type of layers (inputs layer, hidden layer and output layer). the structure of FFNNs is sorted by the type of layers, such as the first layer is input layer and last layer is the output layer, whereas the middle layers (located between input and output layer) can be called as hidden layers, which can be one or more layers. Moreover, in FFNs, the neurons are connected to the following layer neurons by one-direction lines (weights). In other words, there is no feed-back connection in FFNN and the neurons of same layer are not connected with each other. The most common types of Feed-Forward neural networks are listed below (Haykin, 2009, (Du and Swamy, 2013), (Kriesel, 2007), (Tino et al., 2015) & (Gurney, 1997).

    a) Multilayer perceptron

    b) Radial basis function network

2. Recurrent neural network: is a less conventional type of neural network. The architecture of this network allows feed-back connection between neurons. Further, minimum amount of feed-back connection between neurons in this network must one feed-back connection. Also in this network, the neurons of same layer can be connected with each other. The commonly used types of Recurrent neural network are listed below (Haykin, 2009), (Du and Swamy, 2013), (Kriesel, 2007), (Tino et al., 2015) & (Gurney, 1997).

    a) Hopfield network

    b) Boltzmann machine.

3.6 Single Layer Perceptron

It is artificial neuron model that can be defined as a mathematical model of a biological neuron with several inputs (x_1, x_{j1}) and one single output (y). Furthermore, McCulloch and Pitts model also can be referred as a simple neuron paradigm that gathers input patterns and assign them as input parameters through the associated parameters of the weights. In other words, linear threshold system is a neuron that can operates all the number of inputs from another units and form an actual values, this process is performed in accordance to the activation function. The transfer function performs mapping from the input (real values) to the output (into interval); this mapping can be a linear or nonlinear. the sigmoidal function (hard-limiter) was used in McCulloch and Pitts model as transfer function, which referred by (*Ø*). The synapses in artificial neuron model is referred as weights (w) which is the connection lines between inputs and neuron. Moreover, in McCulloch and Pitts model the values of the weight (w) and threshold (θ) were fixed. Artificial neuron model can easily classify inputs set into two various classes (which means the output is binary). The output (y) in artificial neuron or McCulloch and Pitts model is specified by summation of the dot product between weight and input parameters (w_i. x_i) with respect to the activation function *Ø*. (Haykin, 2009), (Du and Swamy, 2013) & (Gurney, 1997).

$$\mathrm{N} = \sum_{i=1}^{J1} w_i\, x_i - \theta = w^T \mathrm{x} - \theta \tag{1}$$

$$\mathrm{y} = \varnothing\,(N) \tag{2}$$

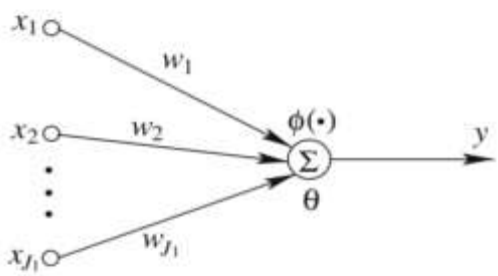


Figure 3.2: Architecture of artificial-neuron model (McCulloch and Pitts model) (Du and Swamy, 2013)

N = network of artificial neuron model, whereas, *x* is the input parameters.

w represents the weight or the connection lines between inputs and transfer function.

$\emptyset$ is the activation function (sigmoidal).

θ is the threshold which is an attribute uses to move the decision boundary away from the origin.

In 1957, the first perceptron (single-layer perceptron paradigm) was developed by Rosenblatt which was inspired by McCulloch & Pitts model and the idea of Hebb (Hebbian learning rule). Rosenblatt's Perceptron model has the capability to classify inputs set into more than two classes unlike artificial neuron (McCulloch & Pitts) model which can only classify inputs set into two classes. In single-layer perceptron model, different activation functions ($\emptyset$) have been used such as a bipolar. Also, the weights (w) and thresholds or biases (θ) is calculated analytically or by a learning algorithm. However, the output ('y) of single-layer perceptron can be written as fallowing. (Haykin, 2009), (Du and Swamy, 2013), (Fausett, 1994) & (Tino et al., 2015).

$$N = w^T x - \theta \tag{3}$$

$$'\mathrm{y} = \emptyset\,(N) \tag{4}$$

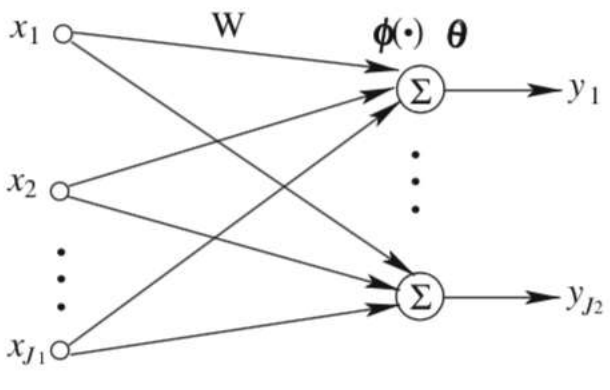


Figure 3.3: Architecture of of Rosenblatt's Perceptron (Du and Swamy, 2013)

Single-layer perceptron has capability only to find solution for linear separable problems. the weight between neurons can be adjusted by using learning algorithm (Rosenblatt's perceptron convergence theorem) and this can be driven through error equation ($E_{t,\ j}$). Moreover, the learning algorithm of perceptron can be written as following:

$$N_{t,j} \ = \sum_{i=1}^{J1} x_{t,i}\, w_{i,j}\ (t) - \theta_j = w_j^T x_t - \ \theta_j \tag{5}$$

$$'\mathrm{y}_{t,j} = \begin{cases} 1 & N_{t,j} > 0 \\ 0 & otherwise \end{cases} \quad , \tag{6}$$

$$E_{t,j} \ = \ \mathrm{y}_{t,j} - '\mathrm{y}_{t,j} \tag{7}$$

$$w_{ij}\ (t+1) \ = \ w_{ij}(t) - \ n \ \ x_{t,i}\ \ E_{t,j} \tag{8}$$

N = network of single-layer perceptron, whereas, $x_{t,I}$ is the *i*th input of the *t*th example.

w_{ij} is the *ith* weigh at the *t*th node (stand for connected lines between neurons).

θ is the bias or threshold for neuron. while, *Ø* is the transfer or activation function.

$E_{t,\ j}$ is denote to the error.

$y_{t,\ i}$ is referred to the real output (desired).

$'y_{t,\ i}$ is the actual output (predicted from network).(Du and Swamy, 2013).

3.7 Multilayer Perceptrons

Multilayer Perceptrons (MLPs) are feed-forward neural network which have the ability of approximating generic classes of functions and solving linear inseparable problems. MLPs structure consist of three different type of layers (inputs, hidden and output) and these layer are fully linked forward by connection lines called as weight (w). MLPs can have one or more hidden layers. Each layer in MLPs contains a number of neurons. the nodes in inputs layer represents the input parameters thus the number of units in inputs layer depend on the number of inputs parameters. The nodes in output layer denotes to the output parameter.

The number of hidden layer neurons can be identified experimentally (trial and error). The amount of layers and neurons of hidden influences the performance of a neural network. The neurons or nodes in hidden layer receive and send signals. In the output layer, the output of neuron can be generated through employing transfer or activation function to the weighted sum, the weighted sum can be calculated by multiplying the input by its related weight (w), Thereafter, the results are added to each other in order to form sum. Mathematically, the neuron output (y) of MLP can be written as following:

$$\mathrm{Y} = \emptyset \left(\sum_{i=1}^{J1} x_{i,j}\, w_{i,j} + \theta \right) \tag{9}$$

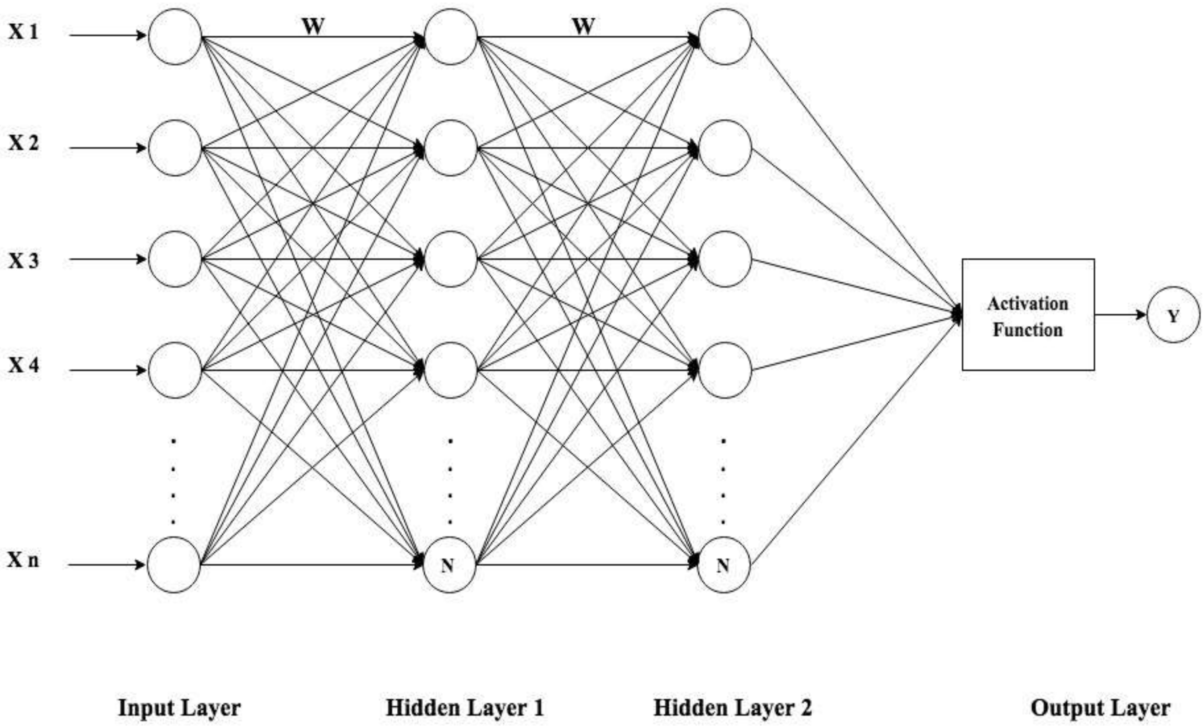


Figure 3.4: Multilayer perceptron (MLP).

3.7.1 Components of Multilayer Perceptrons

The main parts of Multilayer perceptron are layers, weights, and activation functions. Each part has important role in MLP. As shown in figure 3.4 there are three various type of layers (input layer, hidden layer & output layer). These layer are fully connected forward through lines (weights) which allows to the information to be passed between layers. Generally, the Input layer is usually placed at the beginning (considered as the first layer) of network. This layer consists of number of nodes and these nodes represents the number of inputs parameters. Moreover, input layer has no transfer function but, its allow to the information of inputs to be transferred to the hidden layer. Whereas, the hidden layers usually located between the input and output layers, and connected to them through lines (weights). Moreover, the weights start to be modified or updated constantly at hidden layers. The hidden layers are known as processing layers which consist of number of neurons and the number these neurons can be defined experimentally. The last layer known as output layer, which provide the final output of the all network, thus can be considered as processing layer. Figure 3.4 indicated to MLP with two hidden layer, the input layer is fed by inputs parameters. While, the first hidden layer is fed by output of input layer and second hidden layer is fed by the output of the first hidden layer. Moreover, the input of output layer is fed by the output of second hidden layer, whereas the output of output layer is uses to forms the output of the network.

The connection lines between layer are called as weights. These lines play an important role in determining the output in neural networks. At the beginning, the weight in the neural networks is sets at random, then this weight begins to be updated in order to get more accurate results. however, this update can be done through many iterations (epoch) (Haykin, 2009), (Du and Swamy, 2013), (Kriesel, 2007), (Tino et al., 2015) & (Shalev-Shwartz and Ben-David, 2014).

On the other hand, the purpose of using activation or transfer functions in most of neural networks is to provide boundary for the output of nodes. Furthermore, the format of inputs data can be influenced by the type of transfer function, in other word, defining the type of transfer function can indicate how inputs data must be formatted or arranged. neural network can have various types of transfer functions. Most common activation functions are listed below:

a) Linear or Identity: its the most basic type of activation function, the output is linearly, as shown in figure 3.5. The linear transfer function is defined by:

$$\emptyset(x) = x \tag{10}$$

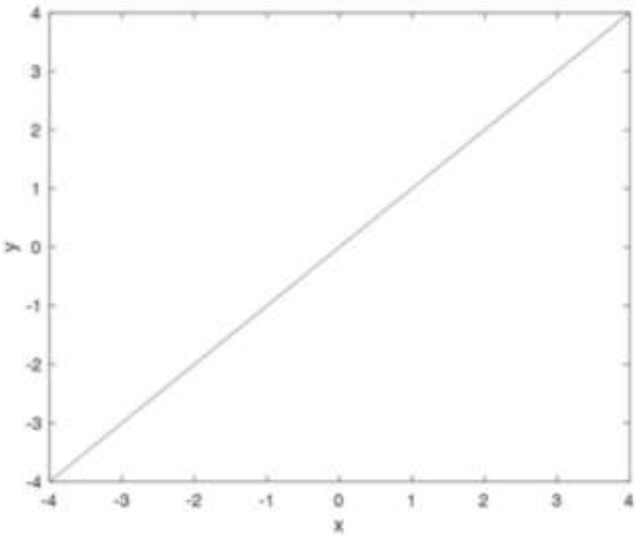


Figure 3.5: Linear activation function

b) Threshold or Step : in 1943, McCulloch and Pitts used this function as an activation. In this function, the outputs are 1 if the input (x) equal or grater than 0.5, while if the input (x) is less than zero then the outputs can be zero or minus one:

$$\emptyset(x) = \begin{cases} 1, & if \quad x > 0 \\ 0, & otherwise \end{cases} \quad , \tag{11}$$

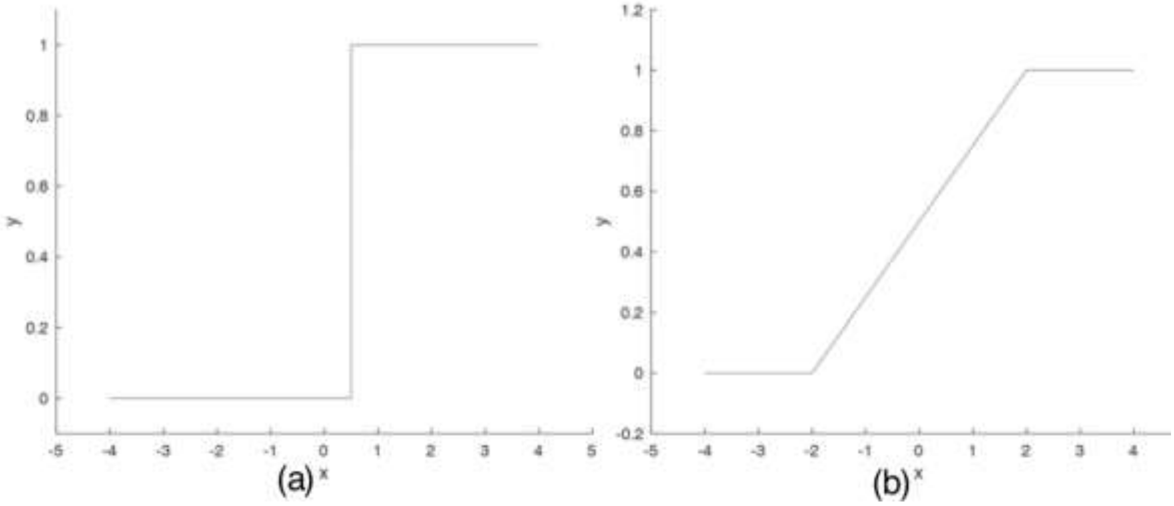


Figure 3.6: (a) Step activation function & (b) Linear threshold between bounds

c) In feed-forward neural networks when the outputs values are positive, sigmoid or logistic activation is the most commonly used activation or transfer function. The values in this function is limited between zero and one. Further, this function can be written as below:

$$\emptyset(x) = \frac{1}{1 + e^{-x}} \tag{12}$$

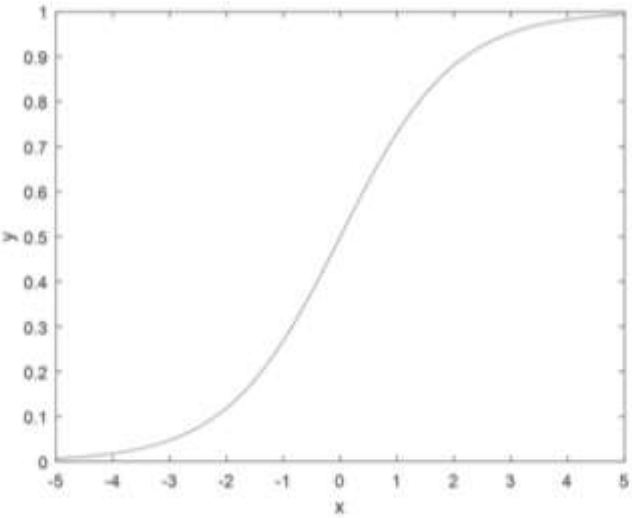


Figure 3.7: Sigmoid activation function

d) The hyperbolic tangent function. this function limited the values between -1 and 1. hyperbolic tangent function is as follows.

$$\emptyset(x) = \tanh(x) \tag{13}$$

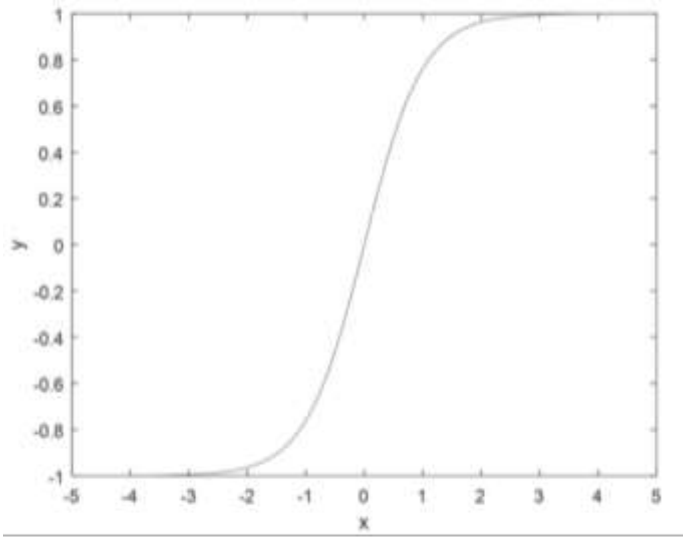


Figure 3.8: Hyperbolic tangent transfer function

Figure 3.9 shows, MLP with N number of hidden layers. And suppose that z need to be calculated then the output of z example can mathematically be written as below:

$$'y_z = y_z^{(N)} \quad , \quad y_z^{(1)} = x_z \tag{14}$$

$$Net_z^{(N)} = \left[w^{(n-1)} \right]^T y_z^{(n-1)} + \theta^{(n)} \tag{15}$$

$$y_z^{(n)} = \emptyset^{(n)} \left(Net_z^{(n)} \right) \tag{16}$$

$Net_z^{(n)} = [Net_{z,1}^{(n)}, \ldots, Net_{z,jm}^{(n)}]$ whereas, N is the number of hidden layers,

$y_z^{(n-1)} = [y_{z,1}^{(n-1)} \ldots, y_{z,Jn-1}^{(n-1)}]$

θ is the bias. where $\emptyset^{(n)}$ is the transfer or activation function $[\emptyset_1^{(n)} \ldots, \emptyset_{Jn}^{(n)}]$.

$y_z^{(N)}$ is referred to the output of the *i*th neuron in the *n*th layer.

$'y_z^{(N)}$, is the actual output (predicted from network) of *zth* example (Du and Swamy, 2013)

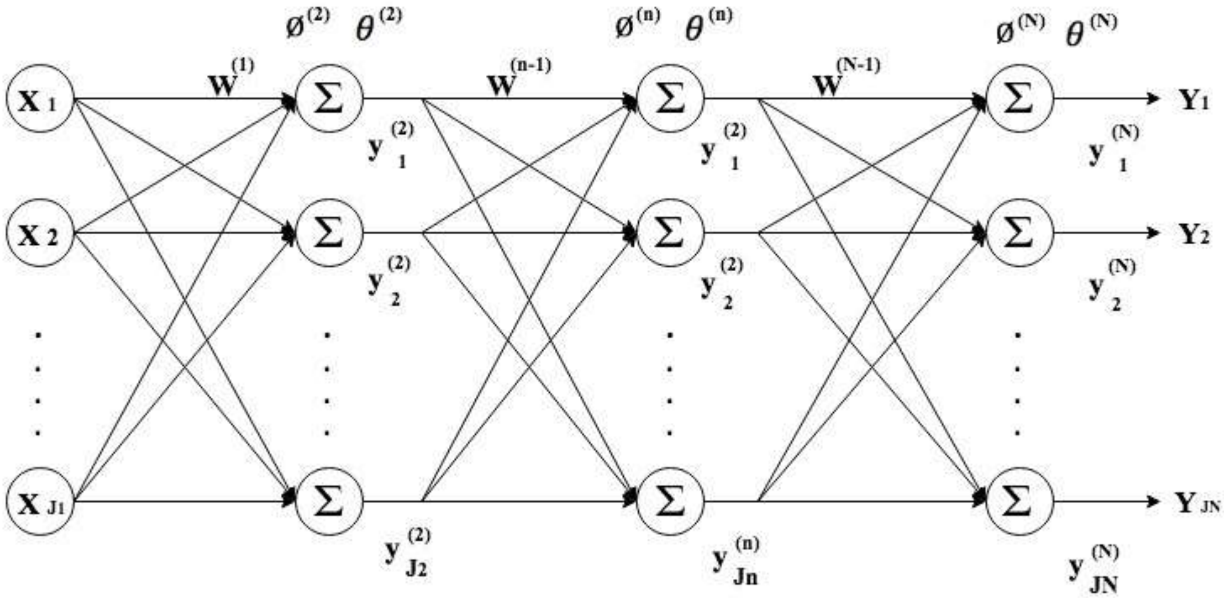


Figure 3.9: Multilayer perceptron (MLP) with N hidden layers

3.7.2 Learning of Multilayer Perceptrons

As previously mentioned, neural networks (NN) process involves training (learning) and generalization or recalling. The learning or training of neural network is represented by reducing the cost function, and can be carried out through locating the optimum weight (w) and sometimes, the parameters of other network. This process is also known as learning algorithm. Back-propagation algorithm is considering as the most commonly used algorithm for training Multilayer perceptron. The training in Neural networks can be carried out by epochs. An epoch can be defined as a full cycle when whole the examples in training are given to the network and are processed using the learning algorithm only once. When training of neural network is completed, the network starts to performs a complex relationship, and possesses the capability for recalling (Haykin, 2009), (Du and Swamy, 2013), (Kriesel, 2007), (Tino et al., 2015) & (Shalev-Shwartz and Ben-David, 2014). There are three different type of methods of learning:

1. Supervised learning

This type of learning is known as learning with teacher. In this learning, neural network is provided by target output values in order to modify the parameters of network through straightforward manner (finding the differences between the desired values and the predicted values). The predicted values are those that generate by the network. Training process is guided by the differences of error (E) which can be calculated using mean squared error (MSE) (Haykin, 2009), (Du and Swamy, 2013), (Kriesel, 2007), (Shalev-Shwartz and Ben-David, 2014)& (Maillard and Gueriot, 1997).

$$E = \frac{1}{M}\sum_{z=1}^{M} \|Yz \ -'Yz\|^2 \tag{17}$$

M represents the pattern value in the sample set, where *Yz* is the predict value of zth example and *'Yz* is the desired value. Generally, the procedure of gradient descent is employed to reduce the errors (E) between the predicted value and the desired value, because gradient descent technique always converges to the local minimum. moreover, back-propagation algorithm based on gradient descent.

2. Unsupervised Learning

In non-supervised learning, the neural networks are only provided with inputs data, where real outputs values are not given to the networks. The networks must be able to find relationship between information from the inputs data. In other words, the training algorithm must be able of finding appropriate subsets of samples of a training set. The most common type of unsupervised training is known as clustering which depend on similarity for instance Euclidean distance. (Haykin, 2009), (Du and Swamy, 2013), (Kriesel, 2007), (Shalev-Shwartz and Ben-David, 2014) & (Maillard and Gueriot, 1997).

3. Reinforcement learning

This kind of learning can be referred as a special status of supervised learning, where the accurate desirable value of output is unknown. in supervised learning, the instructor provides only reaction about success or failure of a result. (Haykin, 2009), (Du and Swamy, 2013), (Kriesel, 2007), (Shalev-Shwartz and Ben-David, 2014) & (Maillard and Gueriot, 1997).

3.7.2.1 Back propagation algorithm

It is a well-known and widely used training rule which is type of supervised learning. It is delta rule generalization which also referred as Least Mean Squares Algorithm (LMS). This algorithm aims to reduce the cost function analogous to the mean square error among the real and predicted output values through using gradient- descent method. In Back propagation algorithm, at the begin of first epoch, the input layer in network is fed by the input pattern and then the output is produced. The error (the difference between target and actual value) propagates to backward and thus a blocked-loop hold system is formed. The gradient-descent algorithm is used to modify the weights. The activation function plays important role in allowing to back-propagation rule to be applied. The error can be calculated by using mean square error MSE equation.

$$E = \frac{1}{M}\sum_{z=1}^{M} E_z = \frac{1}{2M}\sum_{z=1}^{M} \|Yz \ -'Yz\|^2 \tag{18}$$

$$E_z \ = \frac{1}{2}\|Yz - 'Yz\|^2 \quad = \ \frac{1}{2} \ e_z^T \ e_z \tag{19}$$

$$e_z = Yz - 'Yz \tag{20}$$

The Error (E) is reduced by employing gradient-descent which allows to the weights to be adjusted. This can be done using below equation.

$$\Delta_z \, \boldsymbol{W} = -n\frac{\partial E_z}{\partial W} \tag{21}$$

η is referrers to rate of learning and represents our step size which ranged between (0-1) and this can be chosen manually. W is representing the parameters of networks such as weights and bias. Furthermore, equation (22) referred back-propagation algorithm. Moreover, the algorithm can be better through involve using of (μ) momentum factor which analyze and the provide status for convergence (Haykin, 2009) (Du and Swamy, 2013), (Kriesel, 2007), (Xiao, 1996), (Tino et al., 2015) & (Shalev-Shwartz and Ben-David, 2014).

$$\Delta_z(t)\,\boldsymbol{W} = -n\frac{\partial E_z}{\partial W} + \mu\Delta W(t-1) \tag{22}$$

Figure 3.10: Effects of learning rate and momentum parameters on weight updating. (Du and Swamy, 2013)

3.8 Radial Basis Function Neural Networks

It is a special type of neural feed-forward neural networks (RBFNN). It can be defined as universal approximator. The structure of RBFNN is almost similar to Multilayer Perceptrons (MLP), consist of three layers (inputs layer, hidden layer and outputs layer) and these layers are linked forward through connection lines (weights). Except that RBFNN contains only one single hidden layer and each neuron in hidden layer has basis function as shown in figure (3.11). On the other hand, in this type of network, the transition of inputs is nonlinearly carry out by the hidden layer, while the output layer performs a simple weighted sum with a linear output. The learning of RBFNN is faster than MLP thus it is less consuming to the time compared to other conventional networks. RBFNN has more parameters comparing to MLP, such as center and in some case spread or sigma(σ). The process of hidden layer in RBFNN is quite different to other neural networks. The idea is that the patterns in the input space form clusters. If the centers of these clusters are known, then the distance from the cluster center can be measured. Moreover, this distance can be non- linearly measured, thus if a pattern is in an area that is close to a cluster center it gives a value close to 1. In fact, this area is radially symmetrical around the cluster center, so that the non-linear function becomes known as the radial-basis function. Further, there are different type of radial basis function such as Gaussian, thin-plate and logistic function. RBFNN can be used in classification and approximation patterns (Haykin, 2009), (Du and Swamy, 2013), (Tino et al., 2015), (Maillard and Gueriot, 1997), (Orr, 1996), (Schwenkar et al., 2000), (Buhmann, 2011), (Xie et al., 2011), (Khan, 2012), (Markopoulos et al., 2016) & (Wu et al., 2012).

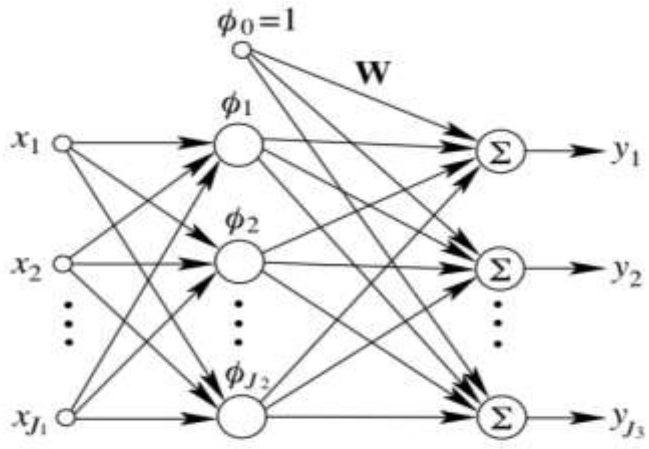


Figure 3.11: Structure of Radial basis function neural network**. (**Swamy, 2013)

3.8.1 Components of Radial Basis Function Neural Networks

RBFNN is feed forward neural network, thus its structure almost same as Multilayer Perceptrons architect. As shown in figure (3.11), three layers are connected together and arranged respectively as input, hidden and output layer. The values of inputs parameters uniformly pass to the hidden layer neuron with no multiplication these value with synaptic weights. As above mentioned that conventional feed forward neural networks may have more than one hidden layer, while in RBFNN there is only one hidden layer. Also in hidden layer of Multilayer Perceptrons network there is no requires of using activation function in each neuron, whereas each neuron in hidden layer of RBFNN has basis function Ø(r) which is nonlinear transfer function. Each basis function has its own center (c_i) . the hidden layer of RBFNN implements nonlinearly transformation of the inputs. The output of RBFNN can be written as below:

$$y_i(x) = \sum_{k=1}^{J_2} w_{Ki} \ Ø \ (\| x - C_K \|) \quad , \quad i = 1, \ldots, J3 \qquad (23)$$

y_i (x) referrers to the ith output of RBFNN whereas, the linking lines from the kth hidden neuron up to the ith output node is represents by wki. The centre of the kth hidden neuron is called as (ck).and || || represents the norm of the Euclidean distance. Ø is basis function. There is various types of basis functions such as Gaussian, thin-plate spline and logistic function. However, Gaussian function is considered as the typical basis function of RBFNN Haykin, 2009), (Du and Swamy, 2013), (Tino et al., 2015), (Maillard and Gueriot, 1997), (Orr, 1996), (Schwenkar et al., 2000), (Buhmann, 2011), (Xie et al., 2011), (Khan, 2012), (Markopoulos et al., 2016) & (Wu et al., 2012).

$$Ø(r) = e^{-\frac{r^2}{2\sigma^2}} \qquad \text{Gaussian function} \qquad (24)$$

$$Ø(r) = r^2 \ln(r) \qquad \text{thin} - \text{plate spline function} \quad (25)$$

$$Ø(r) = \frac{1}{1 + e^{\frac{r}{\sigma^2} - \theta}} \qquad \text{logistic func} \qquad (26)$$

The difference between data (x) to the center (c) is known as distance and symbolizes by r. And this distance always greater than zero. θ indicates the bias. σ represents the spread or sigma and its role is to cog the interpolating function smoothness.

3.8.2 Radial Basis Function Neural Network Learning

Learning process in radial basis function network is mostly similar to multilayer perceptron learning thus the RBFN learning is formulated as the decreasing of mean square error function (MSE)

$$E = \frac{1}{M}\sum_{i=1}^{M} \|y_Z - W^T \|^2 = \frac{1}{M} \| Y - W^T \Phi \|_O^2 \qquad (27)$$

$\| \|_o$ is the norm and it is defined as $\|A\|_o^2$ = tr A^T A; where the desired value for the zth sample in the learning dataset is represents by y_z and Y is equal to $[y_{1,\ldots,}y_N]$ and it represents the summation of y. Despite that radial basis neural network is type of feed forward network and it can be trained as MLP by finding the optimum value of major parameter which is the weight based on reducing MSE. The main parameters of network of RBFNN are center or prototype, and weights. Thus learning of RBFNN involves two process:

1) **Locating the center**

Determining the centre (c_i) considers as critical step in performing RBFNN. Also defining the width or standard deviations (σ_i) plays an important role since it influences the bell shape spacing of Gaussian function. in other word, width is a measure of the spread of the basis function (Gaussian); for instance, if the value of standard deviation is larger than data are more spread out, and not near to the average then prediction accuracy will be affected. However, the center can be located using different methods

- **Randomly**: in this method, the centers of RBFNN are randomly defined. by choosing randomly a subset or part of the input samples from the learning dataset. Whereas, the width σ can be determined by taking the average of all the Euclidean distances, or can be defined by taking the maximum differences between the i th RBF center and its nearest neighbor over the root of number of centers.

- **Clustering**: Clustering is the gathering of a specific group of data based on their features in RBFNN, Clustering can be utilized in order to locating the centres of RBFNN. The training dataset are arranged in form of suitable groups (prototypes), later on, these prototypes turn into centres of RBFNN. There are supervised and unsupervised clustering. The most common type of clustering is k-means.

2) **Finding the optimum weights W.**

When the parameters of radial basis function neural network (centers of RBFNN and widths of the centers) are defined. Then, the weights W can be updated by using the gradient-descent or Least Squares techniques Haykin, 2009), (Du and Swamy, 2013), (Tino et al., 2015), (Maillard and Gueriot, 1997), (Orr, 1996), (Schwenkar et al., 2000), (Buhmann, 2011), (Xie et al., 2011), (Khan, 2012), (Markopoulos et al., 2016) & (Wu et al., 2012).

3.8.2.1 Supervised Learning Algorithm for RBFNN

It is a well-known and widely used when centres, spread and weights of RBFNN are defined, the learning process in radial basis function network can be carried out using a supervised training method (gradient-descent technique) that aims to reduce cost function E (Du and Swamy, 2013) & (Stanford university, 2017)

$$Error\ function\ E = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{J_3}(\, e_{\,n,i}\,)^2 \tag{28}$$

en,i (error) is represents the differences between desired output and actual *i*th output neuron for the *nth* example.

$$e_{\,n,i} = \ y_{\,n,i} \ = \sum_{m=1}^{J_2} w_{mi}\ \emptyset\ (\|\ x_n - C_m\ \|) = \quad y_{\,n,i} - \ w_i^T\ \emptyset_n \tag{29}$$

by making derivative to E with respect to w*mi*, leads to

$$\frac{\partial E}{\partial w_{mi}} = -\ \frac{2}{N}\sum_{n=1}^{J_2} e_{n,i}\ \emptyset\ \ (\|\ x_n - C_m\ \|)\, , m = \ 1,..,J2\, , i \ = \ 1,..,J3 \tag{30}$$

by making derivative to E with respect to *cm*,, yields to

$$\frac{\partial E}{\partial c_m} = \frac{2}{N}\sum_{i=1}^{J_3} w_{m,i}\sum_{n=1}^{J_2} e_{n,i}\ \emptyset\ \ (\|\ x_n - C_m\ \|)\ \frac{x_n - C_m}{\|\ x_n - C_m\ \|}\ , m \ = \ 1,.\ ,J2 \tag{31}$$

The technique of gradient-descent is specified by the below equations:

$$update\ for\ \ weight \qquad \Delta\, w_{mi} \ = -\ n_1 \frac{\partial E}{\partial w_{mi}} \tag{32}$$

$$update\ for\ \ center \qquad \Delta\, c_m \ = -\ n_2 \frac{\partial E}{\partial c_m} \tag{33}$$

CHAPTER 4
PREVIOUS STUDIES ON CONCRETE COMPRESSIVE STRENGTH PREDICTION WITH ARTIFICIAL NEURAL NETWORKS

4.1 Recent Studies

Artificial neural networks have been previously used as prediction tools by many researchers in civil engineering topics. In 1997, Sergio & Mauro have predicted the compressive strength of concrete using artificial neural networks based back propagation algorithm. Cement class, type of sand, type of aggregate, w/c ratio and high range water reducer were the input parameters and the data set included around 250 cases. The authors used different networks, each of different number of hidden layers. The test has been performed and the overall prediction rate was higher than 95 % (Sergio & Mauro, 1997).

One year later, Yeh applied artificial neural networks in order to estimate the compressive strength of high performance concrete. The test involved use of 1000 samples that were collected from various sources. Eight inputs parameters have considered (cement, fly ash, blast furnace slag, water, high range water reducer (HRWR), coarse aggregate and fine aggregate content and age of concrete). The test carried using back propagation network. The results of this study indicated a good prediction accuracy. Furthermore, artificial neural networks showed low error rate than regression analysis technique (Yeh, 1998).

Concrete compressive strength have been estimated by Hong-Guang and Ji-Zo in 2000. The authors employed conventional back propagation neural network. The inputs parameters involved concrete mixture and curing condition. Tow dataset are obtained; the first set involved 65 samples which have been conducted by the authors through experiments, whereas the another set contained around 100 samples that have been collected form public data base. Furthermore 135 samples have used for training and 30 samples were used for testing. The consequences of test indicated adequate prediction rate. also the results show that use of single data source can led to lower error rate (Hong-Guang and Ji-Zong, 2000).

In 2003, Chang Lee has predicted the concrete compressive strength at various age by using modular neural networks that consists of five ANNs which solved the problem existed in single one. The inputs parameters were mix proportion, measurement (slump test) and humidity & temperature up to 28 after concrete was casted. The test was performed and the prediction rate was around 90%. Furthermore, the author stated that using modular neural networks conduct to proper prediction accuracy of concrete strength than single network (ChangLee, 2003).

In the same year, Sebastia et al have employed multilayer neural network based Trajan simulator in prediction compressive strength of fly ash concrete. The data used were collected from pervious studies. The first neural network model was constructed using 20 various variables such as six different type of cements, 3 types of fly ash, water in two forms, six types of additives, silica fume and fine & coarse aggregates. In order to examine the effects of these variables on strength, the authors divided them into different groups based on sensitivity analysis and summing of few variables so that can be eliminated form input parameters. The sensitivity analysis was widely affected by the additives. other neural networks were using various group of input variables regard to sensitivity analysis and summing of few variables. One of these group contained five variables (cement, aggregate, water, additives and fly ash). The results of neural network that have used these five variables show lower regression coefficient comparing to the other models (Sebastia et al., 2003).

Asce et al. (2004) have predicted the compressive strength of concrete at age of four weeks using artificial neural networks based back propagation algorithm. The considered inputs parameters were w/c ratio, fine aggregate %, unit content of cement, unit content of water unit content of fine & coarse aggregate, admixture, and slump. The artificial neural networks test results were closely to the actual strength samples (Asce et al., 2004).

In 2006, Öztaş et al have used Feed-forward network (FFN) in order to predict the compressive strength of high strength concrete. FFN consists of one input layer, two hidden layers and one output layer. The inputs parameters were water/binder ratio, content of water, fine aggregate ratio, content of fly ash, content of air entraining agent, content of high range water reducer and silica fume content. 187 samples were gathered from various sources. 169 samples have been trained and eighteen samples were used to be tested. The test was conducted and the prediction rate was higher than 99 percent (Öztaş et al., 2006).

Kewalramani and Gupta. (2006) had estimated concrete compressive strength o. The authors used multiple regression analysis and back propagation neural network as prediction tools. Two various mixtures of concrete (M20 and M30) and eight hundred sixty-four samples were used for this study. The outcomes of the tests illustrated good prediction accuracy. Furthermore, the artificial neural networks provided higher prediction accuracy than regression analysis (Kewalramani and Gupta, 2006).

In 2007, Topcu and Sarıdemir had utilized artificial neural networks in prediction the strength of waste autoclaved aerated aggregate concrete. The network consisted of four layers, one input layer and two hidden layers and one output layer. Seven inputs parameters were considered (Cement, water, sand, two type of crushed rock, two types of waste autoclaved aerated concrete in fine & coarse form). The data were produced through experiment laboratory test. The ANN results indicated an adequate prediction rate (Topcu and Sarıdemir, 2007).

Topcu & Sarıdemir. (2008) employed artificial neural networks (multi-layer feed-forward neural networks based back propagation algorithm) and fuzzy logic to predict the compressive strength and splitting tensile strength of recycled aggregate concretes that contains silica fume at various ages. The network included four layers, one input layer , two hidden layers and one output layer. The content of cement, water, sand, aggregate, recycled aggregate, superplasticizer and silica fume were as the inputs parameters, whereas the outputs parameters were compressive and splitting strength. Around 210 samples were collected from public data base. Based on the results of ANN and FL models, the prediction rates were above 99 %.In this study, artificial neural networks exhibited slightly lower error rate than fuzzy logic (Topcu and Sarıdemir, 2008).

In 2008,Topcu and Sarıd emir used multilayer perceptron (MLP)and fuzzy logic (FL) in prediction fly ash concrete compressive strength. The used ANN was consisted one input layer, one hidden layer and one output layer. The input parameters included content of cement, water content, sand, content of aggregates, fly ash content, water reducer and calcium oxide content. The prediction accuracy of MLP and FL was satisfactory. The prediction rate was around 99 % (Topcu and Sarıdemir, 2008).

Prasad et al. (2009) estimated the compressive strength of high performance and self compacted concrete at age of four weeks using feed forward neural network. The network consists of one input layer and two hidden layers and one output layer. The input attributes are including constituents of concrete mixture and curing condition. The used samples were around 300, gathered from various sources. The authors experimentally examined several samples in laboratory to be compared with actual and prediction values. Artificial neural networks outcomes exhibited a low error rate (nearly 0.02 %). Also the predicted results were almost similar to experimental outputs (Prasad et al., 2009).

In 2009, intelligent technique (artificial neural network) has been conducted in concrete strength prediction by Trtnik, Kavčič & Turk. The authors used FFNN. The content of aggregate, max size of aggregate, kind of aggregate, shape of aggregate, and pulse velocity were employed as inputs parameters. Artificial neural network test illustrates a good prediction rate. furthermore, the prediction accuracy was around 99 % (Trtnik et al., 2009).

Light weight concrete (LWC) strength at various ages was predicted by Alshihri et al in 2009. The authors applied neural networks based back propagation algorithm. The data were prepared at laboratory experiments. sand, LW coarse aggregate, LW fine aggregate, W/C ratio, silica fume in solution % and silica fume in cement % considered as inputs parameters. The values of ANN test results were nearly similar to targets. Furthermore, the error rate was less than 0.02 % (Alshihri et al at., 2009)

Sarıdemir examined neural networks in predicting strength of concretes that contain pozzolanic materials (metakaolin powder and silica fume) at different ages. Two ANN models were prepared each ANN model with multi hidden layer. Concrete age, cement, metakaolin, silica fume, water, sand, aggregate and high range water reducer were used as attributed parameters. Data were collected form pervious experimental studies. ANN models outcomes show good prediction accuracy. The prediction rate of these models were around 99 % (Sarıdemir, 2009).

Slag - concrete had been predicted by Bilim et al in 2009. The authors used Multilayer feed-forward neural network. 225 samples were experimentally designed to be used for

ANN train and test. Three basic layers were used in the network. The six inputs parameters were involved such as cement, slag, water, aggregate, hyper-plasticizing and concrete age. The predicted strengths were slightly different than the actual (Bilim et al., 2009).

In 2009, Özcan et al had predicted silica fume concrete strength at five different ages; two non destructive methods were used fuzzy logic and artificial neural networks based back propagation algorithm). Cement content, silica fume, water, aggregate, water reducer and concrete age were used as inputs attributes for both techniques. 240 samples were empirically obtained. The results illustrated predicted value estimated by artificial neural network was higher than the one that produced by Fuzzy Logic (Özcan et al., 2009).

In 2010, very low or zero slump concrete strength had been predicted by Sobhani et al. Linear regression model, artificial neural network and adaptive network-based fuzzy were used as prediction tools. The authors prepared 96 samples experimentally. The inputs parameters were content of cement, Silica fume, water, Fine aggregate, coarse aggregate, filler and water / cementitious material ratio. The result showed that linear regression models produce higher rate of error than ANN models (Sobhani et al., 2010).

Słonski, (2010) employed ANN in predicting strength of high performance concrete. cement, the amount of fly ash, slag, water, super plasticizer, coarse aggregate, fine aggregate and age of concrete samples were the inputs attributes. 1030 samples were gathered from pervious studies. The predicted values were close to the actuals (Słonski, 2010).

Razavi, Jumaat & EI-Shafie, (2011) have been predicted light weight concrete compressive strength by Multilayer perceptron. The authors experimentally designed 288 samples. The inputs parameters were scoria, cement and w/c ratio. Neural networks results were almost similar to experimental outputs (Razavi et al., 2011).

In 2011, Atici predicted the compressive concrete containing fly ash and slag at various ages. artificial neural networks were used as prediction tool. 6 various models were constructed. Each model has different inputs parameters. The author experimentally prepared 27 samples. The ANN models results were slightly different to the designed values (Atici, 2011).

In 2011, Hakim et al employed ANN to predict strength of concrete containing cementitious materials. The considered inputs are cement content, water amount , coarse & fine aggregate content , Silica fume, fly Ash, slag and superplasticizer. The author collected around 368 form data base. Thirty NN models were developed. The prediction error was 12.64 percent (Hakim et al., 2011).

FFNN had been used in prediction the strength of Self compacting (SCC) and normal concrete by Siddique et al in 2011. Two data sets were prepared; first data set were collected for literature and used for normal concrete. Second set were empirically designed in laboratory and used for SCC. Two ANNs models were constructed. The inputs parameters of artificial neural network for SCC were cement, sand, coarse aggregate, fly ash, water/powder ratio, HRWR, Bottom ash and Water. The inputs attributes of artificial neural network for normal concrete were cement, sand, coarse aggregate, fly ash, water/powder ratio and high range water reducer (HRWR). The results of these models demonstrate that increasing the inputs parameters can result in higher prediction accuracy; the accuracy was over 91% (Siddique et al., 2011).

Uysal & Tanyildizi have studied the strength prediction for self-compacting concrete cores that contained powder of limestone and marble. The authors experimentally prepared 168 samples. cement, fly ash, powder of limestone, powder of marble, natural aggregate, two type of synthetic aggregate, HRWR, unit weigh and absorption of water were considered as inputs parameters. The used network consisted of four layers, one input layer and two hidden layers and one output layer. The ANN outcomes were nearly like the actual results (Uysal and Tanyildizi., 2011).

In 2012, the strength of concrete containing fly ash, slag & silica fume has been predicted by Boukhatem et al, through using six neural networks based back-propagation algorithms. 960 were gathered form previous studies. Principal component analysis technique (PCA) was employed to study of correlations between data. The prediction results were very similar to the target values. Also the results of using artificial neural networks with principal component analysis were more precise than using ANN alone (Boukhatem et al., 2012).

Singh & Kotiyal have employed learning machine namely artificial neural networks (based back propagation algorithm) to predict the compressive strength of concrete. The authors experimentally prepared 350 cubic samples. The outcomes of ANN models show a good prediction accuracy compared to the actual values (Singh & Kotiyal, 2013).

In 2013, Non destructive technique (artificial neural networks) have used for prediction compressive strength of concrete containing recycled aggregate at age of twenty-eight days by Duan et al. 168 samples were gathered form various data base. The inputs attributes are content of water, cement, sand, natural aggregate, recycled aggregate, impurity, coarse aggregate max size, w/c, saturated surface dry specific gravity of coarse aggregate, aggregate water absorption, coarse aggregate type and replacement of recycled aggregate. The accuracy of the ANN tests found to be higher than 98 % (Duan et al., 2013).

Dantas et al. (2013) have employed ANNs to predict the compressive strength of concretes containing waste of construction and demolition at different ages. The authors collected around 1178 samples from pervious studies. The principal component analysis technique was used to minimize inputs (24). Seventeen inputs were considered. The results of tests were closely to the targets (Dantas et al., 2013).

The compressive strength of lightweight concrete that subjected to high temperatures has been predicted by Bingöl et al in 2013. The authors used artificial neural networks as learning machine. The data were obtained from pervious experimental study. Heating period, pumice/aggregate ratio and target temperature were considered as inputs parameters. The outcomes of ANN models were closely to the actual strength samples (Bingöl et al., 2013).

Sadrmomtazi et al, have predicted the compressive strength of lightweight concrete that made with artificial lightweight aggregate (expanded polystyrene beads). The prediction tools were artificial neural networks (ANN), adaptive network-based fuzzy inference system (ANFIS) and regression modeling. The authors obtained the data experimentally. cement, silica fume, water, fine aggregate, coarse aggregate, Expanded polystyrene beads, waste carper propylene fiber. The results of ANN & ANFIS models were almost similar to the actual values. while regression results were non- reasonable (Sadrmomtazi et al., 2013)

In 2014, Yuan, Wang and Ji have predicted four weeks' compressive strength of 180 concrete samples using artificial neural networks (ANN) based on back propagation algorithm and adaptive network-based fuzzy inference system (ANFIS). In order to reduce the error between the real and desired values in ANN, the authors used genetic algorithms. Three models were formed (ANN, ANFIS, GA-ANN). The authors considered cement, slag, fly ash, water, fine aggregate, coarse aggregate and high range water reducer as inputs attributes. ANFIS model showed best prediction accuracy which was about 95 percent. While ANN and GA-ANN models prediction rates were 68 % and 81 % respectively (Yuan et al., 2014).

Douma et al., (2016) investigated the compressive strength of self compacting concrete by using artificial neural networks (ANN) based back propagation algorithm. 114 samples were obtained from public data base. The content of binder, fly ash %, w/b ratio, fine aggregates, coarse aggregates and high range water reducer were employed as input attributes. The outcomes of this study indicate that increasing the content of the binder and reducing the content of fly ash can lead to rise the strength of concrete at age of four weeks (Douma et al., 2016)

Vidivelli and Jayaranjini, (2016) had predicted the compressive strength of high performance concrete at various ages. The authors used two models of artificial neural networks. The first model was comprised of one input layer, one hidden layer and one output layer while the second model had one input layer, two hidden layer and one output layer. The considered inputs parameters were cement content, silica fume, metakaolin, fly ahs, bottom ash, sand, coarse aggregate and high range water reducer. The error rate between the actual and targets were less than 0.01 %. Also the result of first model were closer to the targets than first model outcomes values (Vidivelli and Jayaranjini, 2016).

Artificial neural networks and regression analysis have been employed in compressive strength of high performance concrete that containing nano silica and copper slag by Chithra et al in 2016. The authors experimentally obtained 45 concrete samples. Three ANNs models were constructed based on inputs parameters and numbers of samples. The inputs parameters were cement, nano silica, water, fine aggregate content, copper slag, age of specimen, high range water reducer and coarse aggregate. The values of models result were almost similar to the expected values. The accuracy of ANNs models was around 99 %, which was higher than regression analysis accuracy (Chithra et al., 2016).

Recently, Khashman & Akpinar had predicted the compressive strength of concrete using artificial neural networks based back propagation algorithm. 1030 samples were used from public data base. The authors considered eight inputs parameters that were water, cement, fly ash, slag, fine aggregate, coarse aggregate, high range water reducer and concrete age. ANNs models results were varying from the targets. The lowest error found to be around 0.26 % (Khashman and Akpinar, 2017).

Bharathi et al have studied the prediction accuracy of Artificial Neural Networks and regression analysis (RA) for self-compacting concrete strength. The authors considered various inputs attributes including as water/binder, cement content, fly ash content, amount of fine and coarse aggregate and superplasticizer. ANN model exhibited low Mean absolute percentage error (MAPE) compared to RA. (Bharathi et al.,2017)

Rebouh et al. (2017) estimated the compressive strength of concrete containing natural pozzolan using Artificial neural networks (ANN) with genetic algorithms (GA). The aim of using GA is to minimize the error between the real and desired values in ANN. Two models were constructed (ANN) and (ANN - GA). Seven-hundred data were collected from various public data bases. Binder, natural pozzlans %, water-binder ratio, gravel, sand, admixtures and age of sample were used as inputs parameters. The targets values were closer to the values that obtained by ANN – GA than ANN results (Rebouh et al., 2017).

.

CHAPTER 5
METHODOLOGY

5.1 General Concept

Concrete is made of various materials such as cement, water and aggregates. The compressive strength of concrete is affected by these materials. The right parameters should include all types of attributes that can affect the strength of concrete including materials compositions and age of concrete. In this work, neural network based models are employed in order to be trained to predict the strength of concrete. Multilayer Perceptrons neural network and a radial basis function network are both used in this research

5.2 Developed System

Concrete is made of various Extended literature review has been carried out on compressive strength tests and the factors that influence the compressive strength. Also a deep literature review was done on ANNs usage in predicting concrete strength in order to understand the types of used neural networks and the utilized parameters. The procedure for this study can be summarized by two stages; data collection, and defining the proper neural network architecture. In data collection stage, input and output attributes are defined. Input parameters are defined based on their influence on the compressive strength of concrete. Radial basis function neural network (RBFNN) which was observed to not be used in previous studies, is selected as a prediction tool in this study, in order to provide insights on its efficiency in potential concrete compressive strength prediction. Multilayer perceptrons (MLP) or back prorogation neural (BPNN) network is employed in order to examined the efficiency of RBFN. Figure 5.1 shows a flowchart of the presented work. It shows that the network is first trained on a part of the data, and once it converges, a new set of data that were not seen before is used for testing purposes.

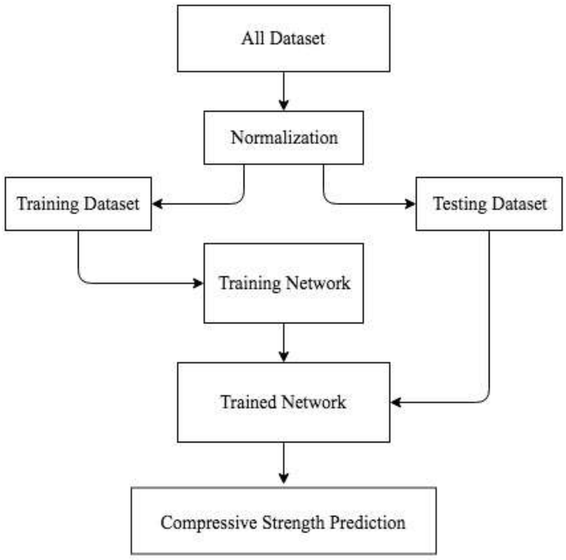


Figure 5.1: Flowchart of the developed network system

5.3 Data Acquisition

Machine learning systems are data hungry networks. Hence, more data results in better prediction and less errors (Haykin, 2009). Therefore, for this study a proper number of data was collected from different articles published in Scopus. Table 5.1 shows the data sources in addition to the number of samples taken from each source.

Table 5.1: List of Database Sources

No	Data Source	Number of obtained data
1	Ndihokubwayo, A. (2011). Compressive and flexural strengths for considerable volume fly-ash concrete. *Journal of Civil Engineering Research*, *1*(1), 21-23.	24
2	Liu, M. (2010). Self-compacting concrete with different levels of pulverized fuel ash. *Construction and Building Materials*, *24*(7), 1245-1252.	20
3	Saha, A. K. (2018). Effect of class F fly ash on the durability properties of concrete. *Sustainable Environment Research*, *28*(1), 25-31.	25

4	Gholampour, A., & Ozbakkaloglu, T. (2017). Performance of sustainable concretes containing very high volume Class-F fly ash and ground granulated blast furnace slag. *Journal of Cleaner Production*, *162*, 1407-1417.	16
5	Wongkeo, W., Thongsanitgarn, P., Ngamjarurojana, A., & Chaipanich, A. (2014). Compressive strength and chloride resistance of self-compacting concrete containing high level fly ash and silica fume. *Materials & Design*, *64*, 261-269.	48
6	Sounthararajan, V. M., & Sivakumar, A. (2013). Accelerated engineering properties of high and low volume fly ash concretes reinforced with glued steel fibers. *Frontiers of Structural and Civil Engineering*, *7*(4), 429-445.	18
7	Ramezanianpour, A. A., & Malhotra, V. M. (1995). Effect of curing on the compressive strength, resistance to chloride-ion penetration and porosity of concretes incorporating slag, fly ash or silica fume. Cement and concrete composites, 17(2), 125-133.	8
8	Atiş, C. D. (2005). Strength properties of high-volume fly ash roller compacted and workable concrete, and influence of curing condition. *Cement and Concrete Research*, *35*(6), 1112-1121.	69
9	Poon, C. S., Lam, L., & Wong, Y. L. (2000). A study on high strength concrete prepared with large volumes of low calcium fly ash. *Cement and Concrete Research*, *30*(3), 447-455.	8
10	Zhang, M. H., & Islam, J. (2012). Use of nano-silica to reduce setting time and increase early strength of concretes with high volumes of fly ash or slag. *Construction and Building Materials*, *29*, 573-580.	4
11	Choi, S. W., Jang, B. S., Kim, J. H., & Lee, K. M. (2014). Durability characteristics of fly ash concrete containing lightly-burnt MgO. *Construction and Building Materials*, *58*, 77-84.	10
12	Oner, A., Akyuz, S., & Yildiz, R. (2005). An experimental study on strength development of concrete containing fly ash and optimum usage of fly ash in concrete. *Cement and Concrete Research*, *35*(6), 1165-1171.	56
13	Atiş, C. D. (2003). Accelerated carbonation and testing of concrete made with fly ash. *Construction and Building Materials*, *17*(3), 147-152.	20
Total		326

As seen in the table, 326 samples are collected from different sources to be then used to train and test the networks. It should be noted that 266 of dataset are cubic and 60 of them are cylindrical, therefore the cylindrical samples are converted to cubic using the following relationship:

Strength of the cylindrical specimen of concrete = 0.8 * strength of the cubic specimen of concrete (Neville, 2012).

It is noteworthy to mention that three different learning schemes are used for training each employed network. Those learning schemes include different training and testing ratios in addition to different learning parameters settings. Table 5.2 shows learning schemes for this study.

Table 5.2: Learning schemes

Ratio	**Data = 326**	
	Training	**Testing**
60 : 40	195	131
50 : 50	163	163
40 : 60	131	195

5.3.1 Input coding

For this study, 13 input parameters that influence the strength of concrete are selected. In this study, six input parameters are selected in this work that have not been considered in previous studies. These new input parameters are including CaO, SiO_2, Al_2O_3 of cement, CaO, SiO_2 and Al_2O_3 of fly ash. However, these parameters are considered because they are known to affect the amount of cement hydration products in concrete, and hence they have various effect on concrete strength. The cubic specimens have different size of specimen; thus the volume of specimen was introduced as an input parameter in order to minimize the negative of varying size on the accuracy of strength prediction. Those parameters are then used as inputs for the networks that are trained using many examples or samples of data that include major oxides of cement and fly ash, in addition to other parameters. Table 5.3 shows a brief description of the input parameters used in this research.

Table 5.3: Database input parameters description

No	Attributes	Attribute Description
1	Cement Content (kg/m^3)	The amount of cement in concrete mixture
2	CaO of Cement (%)	The amount of calcium oxide in cement.
3	SiO_2 of Cement (%)	The amount of silicon dioxide (silica) in cement.
4	Al_2O_3 of Cement (%)	The content of aluminium oxide (alumina) in cement.
5	Fly Ash Content ($kg/m^{3)}$	The amount of fly ash in concrete mixture.
6	CaO of Fly ash (%)	The amount of calcium oxide in fly ash.
7	SiO_2 of Fly ash (%)	The amount of silicon dioxide (silica) in fly ash.
8	Al_2O_3 of Fly ash (%)	The content of aluminum oxide (alumina) in fly ash.
9	w/c ratio	Water – cement ratio in concrete mixture
10	Fine aggregates Content (kg/m^3)	Particles that are smaller than 5 mm (sand) in concrete mixture
11	Coarse aggregates Content (kg/m^3)	Particles that have size greater than 5mm (gravel) in concrete mixture
12	Age of Concrete (days)	Age of concrete specimen; 28 days and 90 days
13	Volume of Cube (m^3)	The volume of concrete specimen

Figure 5.2 shows a description of the presented model for the strength prediction of concrete.

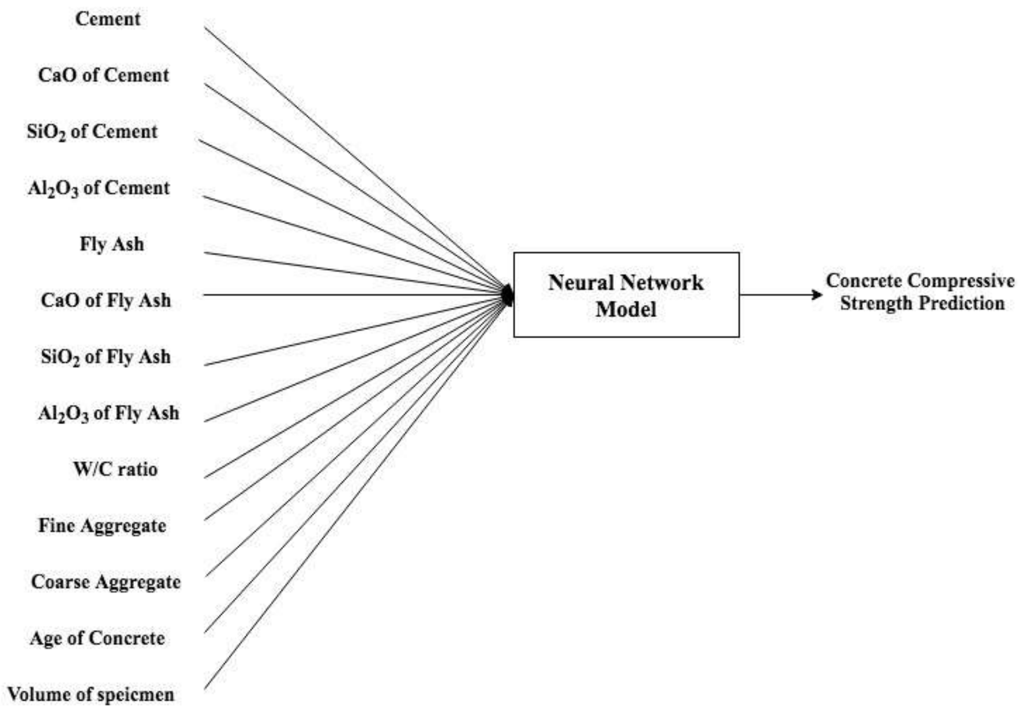


Figure 5.2: Model's inputs and output

As seen in Figure 5.2, 13 attributes are used as inputs for the network that is trained to predict the strength of concrete. Therefore, each network should have 13 neurons in its input layer. Note that the input data are first normalized before being fed into network. Normalization is a pre-processing stage and it means to transform the input and output data values into a range of 0 to 1. The purpose of normalization is make sure that data is roughly uniformly distributed between the inputs and the outputs of the network. (Markopoulos et al., 2016). The equation 1 shows how the data are normalized.

$$N = \frac{Parameter\ value - \min value}{Range\ of\ arrtibute\ value} \tag{34}$$

N is the data after normalization.

5.3.2 Output coding

The networks are developed to predict the strength of concrete by training them using many examples with their corresponding targets, which represent the strength of concrete for each sample or example. Thus, each network should have only one neuron in its output layer, to show the strength of concrete. Note that output values are also normalized to values of range 0 to 1 in this layer as they are in different ranges, which makes it difficult for the network to perform function approximation. Since the output is a decimal value between 0 and 1, which represents the strength of concrete, a linear activation function (Purelin) is used in RBFNN and sigmoid in MLP. This activation function is most commonly used in the regression tasks as it allows the network to show the output in any range, without scaling it.

5.4 RBFNN Model

As discussed in section 3.8 that Radial basis function neural network is an exceptional type of feed forward neural network. RBFNN consists of on single hidden layer. RBFNN has two critical parameters (center and spread) that affect the prediction accuracy. Therefore, these parameters must be determined before learning. These parameters can be defined using different techniques. Then, RBFNN can be learned using various methods (supervised and non supervised). The learning procedure of RBFNN for this study can be summarized into two stages;

1. Define center and spread of RBFNN; selecting the center (c) of the basis function randomly, whereas the spread (σ) is manually determined by trial and error.
2. Supervised learning to optimize the weights and centers; After defining the centers, spread and weights, the rest learning process in RBFNN can be carried out using a supervised training method (gradient-descent technique) that aims to reduce cost function. Learning algorithm for this study is mentioned in section 3.8.2.

Figure 5.3, shows input, hidden and output layers. In input layer, there are 13 neurons which are represents the amount of input parameters are used as inputs for RBFNN. Also from figure it can be seen that each neuron in hidden layer has basis function.

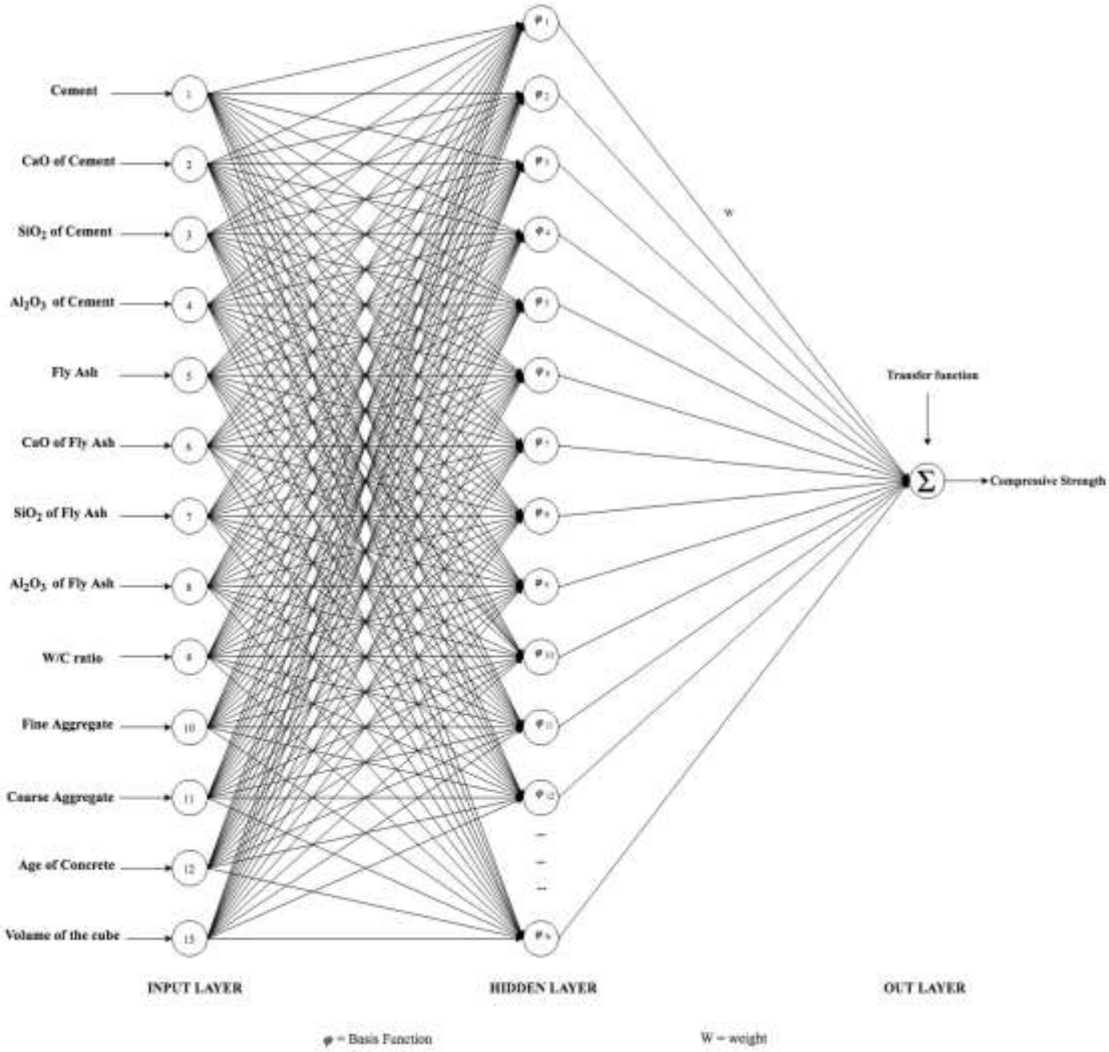


Figure 5.3: RBFNN Structure

5.5 MLP Model

As discussed in section 3.8 multilayer perceptron is a conventional type of neural network (feed forward neural network) that includes three layers; input, hidden and output respectively, where all these layers are connected forward through lines. These lines are called weights see figure 5.4. This network is known as BPNN. The learning process for this network is easily implemented using Back Propagation Algorithm which supervised learning see chapter 3 section 7.2.1. However, the proposed model of MLP is widely used in prediction concrete compressive strength (Chithra et al., 2016) (Khashman and Akpinar, 2017).

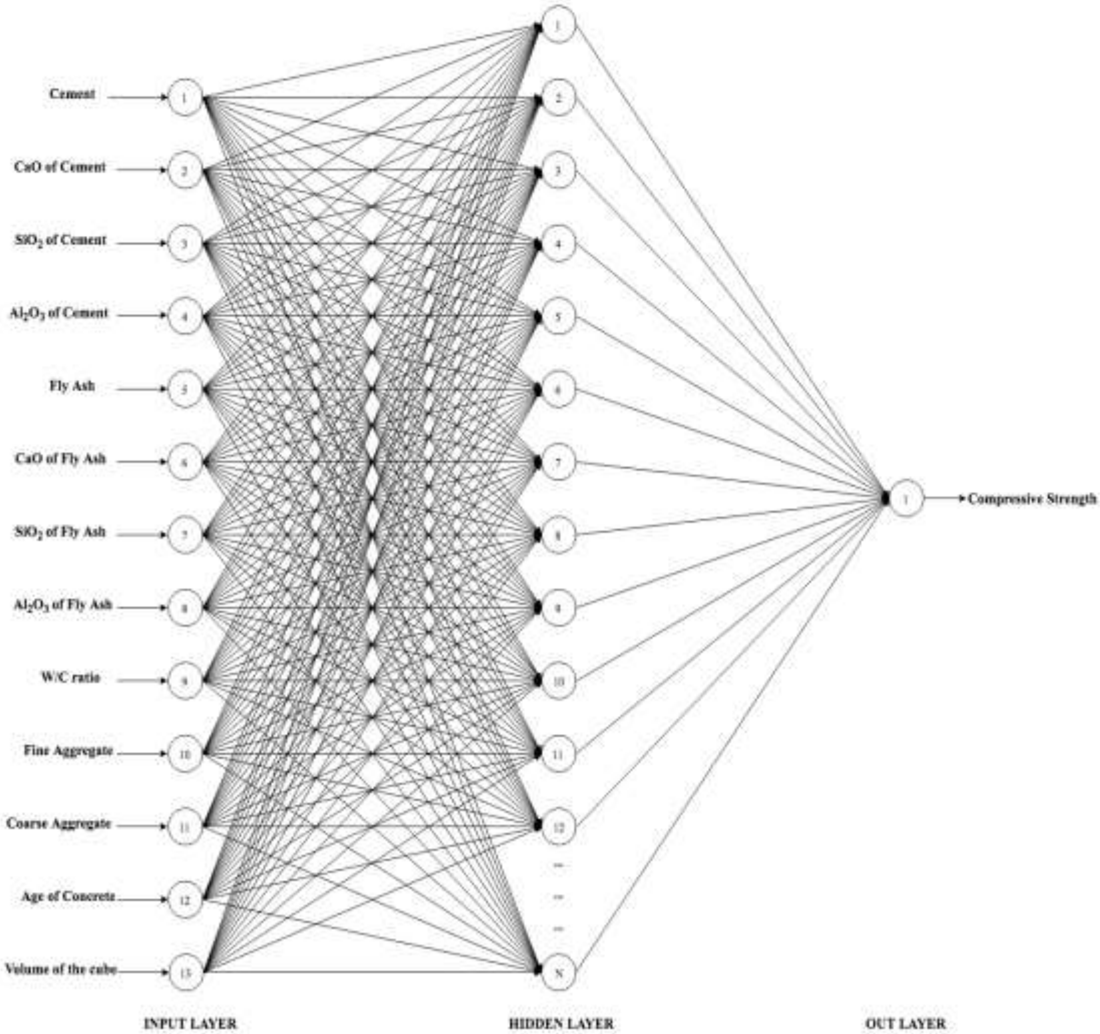


Figure 5.4: MLP Structure

CHAPTER 6
RESULTS AND DISCUSSIONS

6.1 Networks Training

As previously mentioned, the two employed models are trained and tested on a set of data obtained from pervious experimental works obtained from the related literature. The networks performance is investigated through many experiments that are conducted so that required parameters were adjusted to yield the best prediction results. Hence, in this work, train and test ratio, the learning parameters including the number of hidden neurons, spread, the learning rate, and momentum were all varied in order to study effects on the neural networks performance and generalization capability during testing on unseen data. All experiments were conducted using Matlab environment (Matlab version 2015A). studied with three selected train: test ratios with varying hidden neurons and spread values.

6.2 Results of Multilayer Perceptron Neural Networks

MLP is learned using supervised training method (BPLA). Therefore, its called back propagation neural network (BPNN). In this study, BPNN is used to predict the compressive strength of concrete. The network performance is experimented through different parameters (number of hidden neurons & epochs) and data ratios in order to obtain satisfactory prediction accuracy.

Network is first trained using three different train and test ratios:

- Ratio 1: 60:40
- Ratio 2: 50:50
- Ratio 3: 40:60

Moreover, network performance is also studied by selecting different number of hidden neurons and number of iterations. Its worthy to mention that all networks (MLPs) are trained using one training algorithm and one activation function, i.e. Levenberg-Marquardt and Log-Sigmoid, respectively.

6.2.1 MLP Results for Ratio 1; 60:40

Table 6.1 shows the training and testing results of the networks through a fixed ratio (ratio 1), and three different hidden neuron values and epochs. From the Table 6.1, it can be seen that the networks trained with ratio 1 achieved good correlation coefficients during training in which all networks achieved not less than 91.17 % with not more than 0.00659 as mean square error. It is important to note that the network that achieved the highest training accuracy (98.88%) could not achieve the same during testing (90.45%). This network uses 10 neurons in its hidden layer and it was trained using 200 iterations to reach an MSE of 0.0008428, which is the smallest achieved error between all networks shown in this table.

Table 6.1: Results of MLP for Ratio 1; 60 : 40

Ratio	Neurons	R % Training	R % Testing	R % Overall	Epochs	MSE	Learning Rate
1	10	**91.17**	89.21	90.17	10	**0.00659**	0.09
1	10	94.84	93.28	94.23	25	0.00398	0.09
1	10	95.88	95.45	95.57	50	0.00308	0.09
1	10	96.56	95.07	95.7	100	0.00266	0.09
1	**10**	**98.88**	**90.45**	**95.51**	**200**	**0.0008428**	**0.09**
1	30	92.85	91.86	92.23	10	0.00527	0.09
1	30	94.99	93.6	94.31	25	0.00391	0.09
1	30	98.78	91.82	95.94	50	0.000937	0.09
1	30	96.77	91.12	94.08	100	0.00252	0.09
1	30	97.09	92.08	95.09	200	0.0024	0.09
1	50	93.7	92.47	93.18	10	0.00492	0.09
1	50	96.42	92.92	94.67	25	0.00285	0.09
1	**50**	98.80	92.62	**96.25**	**50**	**0.0009366**	0.09
1	50	96.11	93.8	94.75	100	0.002967	0.09
1	50	96.81	85.86	92.77	200	0.0049	0.09
1	70	95.05	94.7	94.91	10	0.00392	0.09
1	70	97.46	90.95	94.83	25	0.00196	0.09

1	70	96.17	92.87	94.61	50	0.00275	0.09
1	70	96.57	91.24	93.69	100	0.00242	0.09
1	70	96.67	82.63	89.78	200	0.0024212	0.09

Moreover, it is seen that the network trained with 50 hidden neurons and 50 epochs achieved the highest overall correlation coefficient (R) (96.25%) with an error of 0.0009366. However, this error is not the least between all networks since another network achieved a lower error (0.0008428). respectively.

6.2.2 MLP Results for Ratio 2; 50:50

Table 6.2 shows the training and testing results of networks, but with different train/test ratio (ratio 2) and five different hidden neuron number and epochs that were defined with an earlier stage.

Table 6.2: Results of MLP for Ratio 2; 50 : 50

Ratio	Neurons	R % Training	R % Testing	R % Overall	Epochs	MSE	Learning Rate
2	10	91.38	86.35	88.96	10	0.00639	0.09
2	10	92.58	91.14	91.84	25	0.00571	0.09
2	10	94.04	92.84	93.42	50	0.00447	0.09
2	10	96.02	95.35	95.61	100	0.00314	0.09
2	10	96.81	90.64	93.83	200	0.00262	0.09
2	30	93.74	89.01	90.98	10	0.00441	0.09
2	30	93.42	91.62	92.47	25	0.00535	0.09
2	30	99.15	86.62	92.31	50	0.00059	0.09
2	30	96.23	90.15	91.33	100	0.00302	0.09
2	30	99.74	89.77	93.92	200	0.000164	0.09
2	50	94.1	90.92	92.55	10	0.00487	0.09
2	50	94.7	92.25	93.31	25	0.00433	0.09
2	50	99.28	90.05	93.49	50	0.000506	0.09
2	50	99.06	90.5	94.62	100	0.000698	0.09

2	**50**	**99.79**	**74.7**	86.32	**200**	**0.00015**	**0.09**
2	70	96.72	91.76	93.49	10	0.00251	0.09
2	70	97.94	91.05	93.52	25	0.00142	0.09
2	**70**	**99.4**	**93.62**	**96.07**	**50**	**0.000384**	**0.09**
2	70	99	84.46	91.78	100	0.000788	0.09
2	70	97.9	87.78	92.66	200	0.00158	0.09

As seen in table 6.2, the network that achieved the highest training (99.79%) accuracy performed unsatisfactory during testing; as it achieved a low testing accuracy (74.7%) compared to others. This network was trained on 50 hidden neurons and 200 iterations. Note that this network achieved the lowest MSE (0.000150) between all networks. Consequently, this shows that high training accuracy does not grant the network a good generalization capability during testing. Also, it is found that the network that was trained with 70 hidden neurons and 50 iterations has achieved the overall correlation coefficient (96.07%), but not the least error (0.000384).

6.2.3 MLP Results for Ratio 3; 40:60

Finally, Table 6.3 shows the same networks rained with ratio 3 and experimented with different number of hidden neurons and iterations.

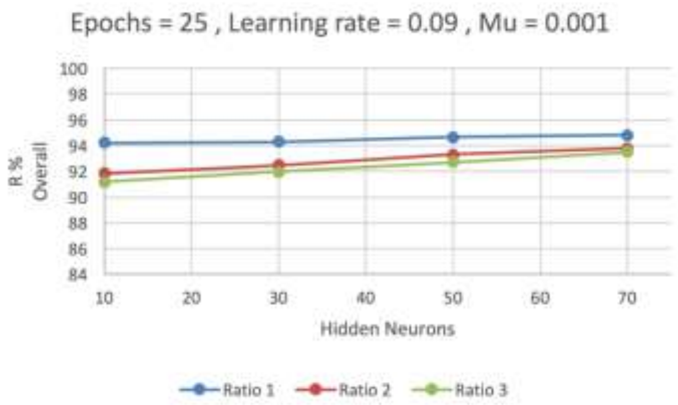
Table 6.3: Results of MLP for Ratio 3; 40 : 60

Ratio	Neurons	R % Training	R % Testing	R % Overall	Epochs	MSE	Learning Rate
3	10	91.29	84.66	87.19	10	0.00645	0.09
3	10	93.84	89.21	91.18	25	0.00481	0.09
3	10	98.42	90.18	93.55	50	0.0012	0.09
3	10	99.23	89.9	93.5	100	0.000588	0.09
3	10	99.48	89.07	93.35	200	0.000389	0.09
3	30	93.72	88.59	90.67	10	0.00517	0.09
3	30	94.05	91.16	91.96	25	0.00474	0.09
3	30	98.91	89.4	93.57	50	0.000926	0.09

3	**30**	**99.68**	**90.152**	93.92	**100**	**0.000247**	**0.09**
3	30	99.54	80.49	86.49	200	0.000362	0.09
3	50	94.62	90.16	91.54	10	0.00349	0.09
3	50	96.12	90.6	92.73	25	0.0307	0.09
3	**50**	**95.1**	**93.6**	**94.05**	50	0.00383	**0.09**
3	50	95.63	80.6	85.52	100	0.003378	0.09
3	50	94.59	93.15	92.8	200	0.00315	0.09
3	70	96.72	91.76	93.49	10	0.00251	0.09
3	70	97.94	91.05	93.52	25	0.00142	0.09
3	70	98.02	88.98	92.14	50	0.001453	0.09
3	70	96.33	88.34	91.35	100	0.0026398	0.09
3	70	95.61	81.47	86.31	200	0.00201	0.09

In this table, the same observation can be made; a network that achieved the highest training correlation (99.68%) and least error (MSE) (0.000247) could not perform well in testing (90.152%) compared to other network that achieved lower training accuracy (95.1%) but best performance during testing (93.6%).

(a)


Epochs = 25 , Learning rate = 0.09 , Mu = 0.001
R %
Overall
Hidden Neurons
Ratio 1
Ratio 2
Ratio 3

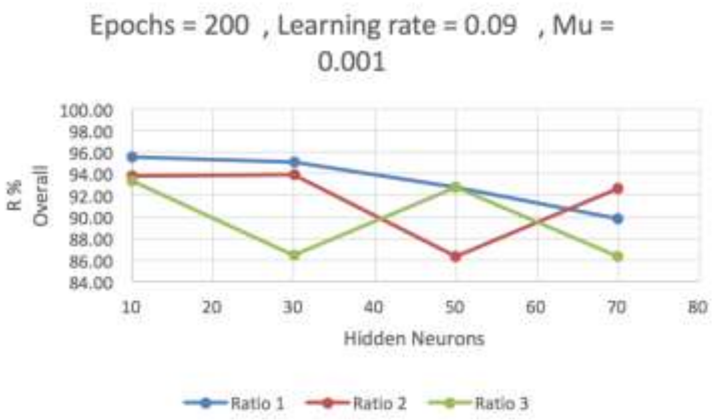
(b)


Epochs = 50 , Learning rate = 0.09 , Mu = 0.001
R %
Overall
Hidden Neurons
Ratio 1
Ratio 2
Ratio 3

(c)


Epochs = 100 , Learning rate = 0.09 , Mu = 0.001
R %
Overall
Hidden Neurons
Ratio 1
Ratio 2
Ratio 3

(d)



(e)

Figure 6.1: Variation of Overall correlation (R%) with respect to different number of hidden neuron

Figure 6.1 shows the variation of overall correlation (R%) with respect to the number of hidden neurons increase in which different number of epochs are used: 10, 25, 50, 100, 200.i.e. (a), (b), (c), (d), (e) respectively. It is noted that the ratio 1 is the best training/testing scheme as the networks achieved the highest overall correlation (96.25%) with 50 hidden neurons and 50 epochs. On the other hand, the lowest correlation (85.52%) was obtained with ratio 3 by network that have 50 neurons and 100 epochs (Fig. 6.1(c), (d)). It is seen that the increase of number of hidden neurons is resulting in an increase of the correlation when the epochs are between 10 and 50. In contrast, when epochs are more than 50, hidden neurons increasing starts negatively affecting the correlation.

Similarly, ratio 1 scheme is still the best between the other ratios, even after passing the optimum value of epochs as shown in Fig. 6.1(d). Also, ratio 2 exhibited slightly lower correlation compared to ratio 1. However, in here the number of hidden neurons needed for the network to achieve such high correlation is only 10 neurons. For ratio 3, Note that the correlation here decreased to 85.53 % which is lower than the one obtained using 30 epochs.

Lastly, Figure 6.1(e) states the same conclusion; ratio 1 is the best scheme regardless of the slight decrease in correlation compared to Fig. 6.1(d). This small decrease in correlation is more likely due to the increase in number of epochs. In other words, the overall correlation starts to decrease after exceeding the optimum value of epochs. (0.0008428). respectively.

6.3 Results of Radial Basis Function Network

A radial basis function neural network for the prediction of concrete strength is also experimented in this work. Similarly, many experiments were conducted by checking the performance of the network using different train/test ratios, different spread values, and number of hidden neurons. Spread was selected to be studied due to its effects on the learning process of this network.

This network is also trained and tested on three ratios:

- Ratio 1: 60:40
- Ratio 2: 50:50
- Ratio 3: 40:60

Note that Gaussian is used as basis function for all radial basis function neural network see equation (24) in section 3.8.1. Also same ratios were experimented with different spread values. In addition, different ratios with same spread were also experimented in order to find a relation between spread values and train/test ratios.

6.3.1 RBFNN Results for Ratio 1; 60 : 40

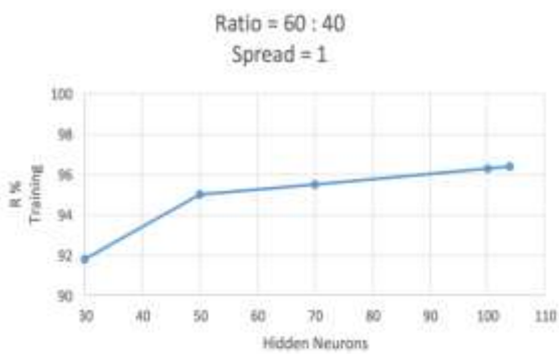
Table 6.4 shows the performance of the network trained with learning scheme (60:40) together with five different hidden neuron values and four spread values. By interpreting the 6.4 table, it is noted that the network testing correlation is increasing with the decrease of spread value. The network that uses spread of 1 achieved a testing correlation of 94% with 50 hidden neurons, and this correlation started to increase with respect to the decrease of spread value until it reaches a correlation of 96.07% with a spread value of 0.1349. Also, it is seen that the number of hidden neurons does not have a serious effect on the learning of the network. In contrast, a relation between spread value and hidden neurons number is noticed. Each spread value has its own optimum hidden neuron values in which the testing correlation starts to decrease after skipping this specific number of hidden neuron. For instance, when spread value is 1, the optimum number of hidden neurons is observed to be 70, which results in the highest testing correlation, 93.8%. When the number of hidden neurons rises, the testing correlation starts to decrease, with the same spread value. However, the best overall correlation (96.784) is archived when the network has 104 neurons and spread of 0.1349.

When the value of hidden neuron is considered for overall correlation, then it can be observed that for high spread values like 1 and 0.154, optimum hidden neuron value is 70, whereas for spread value like 0.135 and 0.1349, the optimum hidden neuron is observed to be 104, even though overall correlation for hidden neuron value of 70, 100 and 104 are observed to be really close.

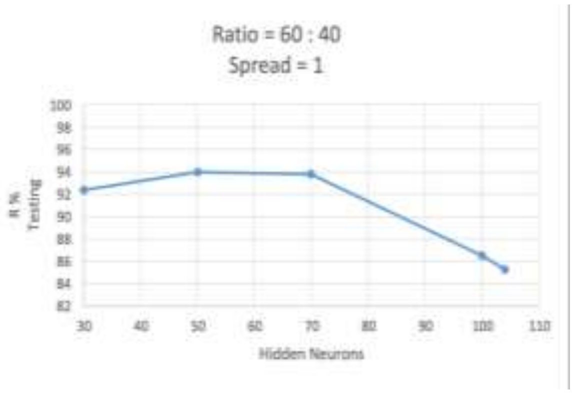
Table 6.4: Results of RBFNN for ratio 1; 60 : 40

Ratio	**Neuron**	**Spread**	**R Training %**	**R Testing %**	**R Overall %**	**MSE**
1	30	1	91.8	92.4	92.04	0.006946
1	50	1	95	94	94.6	0.004335
1	70	1	95.5	93.8	94.82	0.0039
1	100	1	96.3	86.5	92.38	0.003207
1	104	1	96.41	85.3	91.966	0.00315
1	30	0.154	94.24	93.87	94.092	0.00499
1	50	0.154	96.13	95.79	95.994	0.003385
1	70	0.154	96.7	95.19	96.096	0.002895
1	100	0.154	97.18	94.06	95.932	0.002485
1	104	0.154	97.22	83.9	91.892	0.002447
1	30	0.135	93.8	90.834	92.6136	0.00536
1	50	0.135	96	93.95	95.18	0.003464
1	70	0.135	96.76	95.9	96.416	0.00308
1	100	0.135	97.2	95.7	96.6	0.002439
1	104	0.135	97.26	95.79	96.672	0.002423
1	30	0.1349	93.8	90.83	92.612	0.00536
1	50	0.1349	96	93.9	95.16	0.003469
1	70	0.1349	96.7	96	96.42	0.002848
1	100	0.1349	97.23	95.8	96.658	0.002438
1	**104**	**0.1349**	**97.26**	**96.07**	**96.784**	0.002412

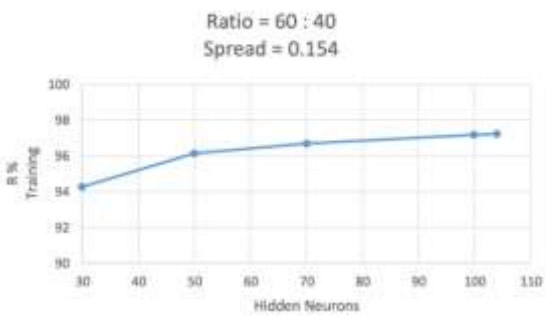
Figure 6.2 shows the variations of training and testing correlations with respect to the increase of number of hidden neurons, each using one spread value. It can be seen that the increase of hidden neurons number is contributing to improving the training correlation. However, this hidden neurons number has an optimum number with respect to the same spread, after which the networks testing correlation starts to decrease. Also, it can be noticed that best training and testing correlations for this ratio are archived within 0.1349 spread.



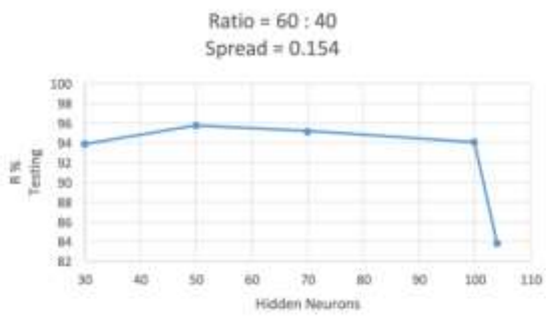
(a)



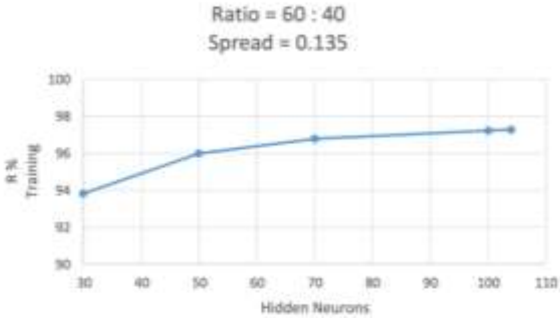
(b)



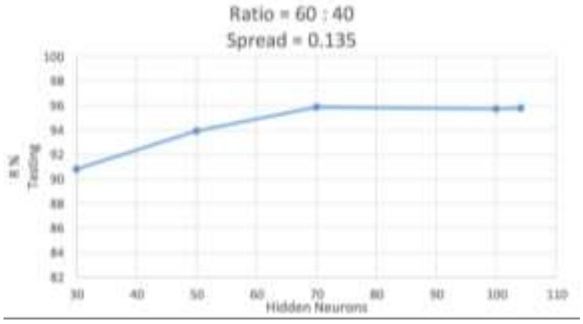
(c)



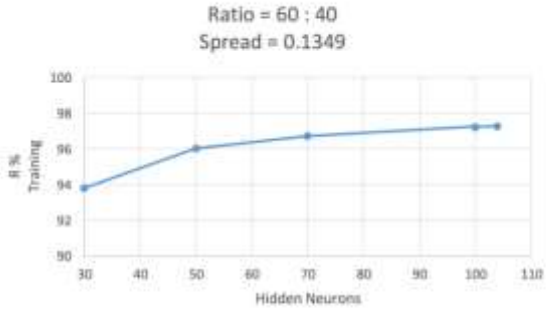
(d)



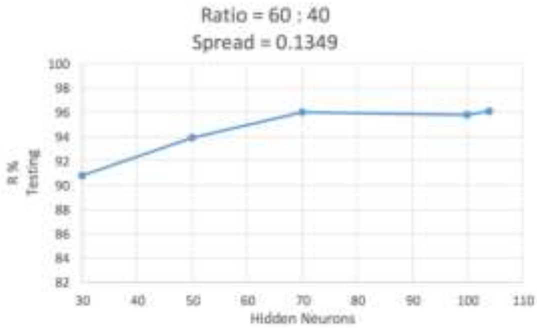
(e)



(f)



(g)



(h)

Figure 6.2: Variations of training and testing correlations with respect to hidden neurons number and fixed spread value using ratio 1; 60: 40

6.3.2 RBFNN Results for Ratio 2; 50 : 50

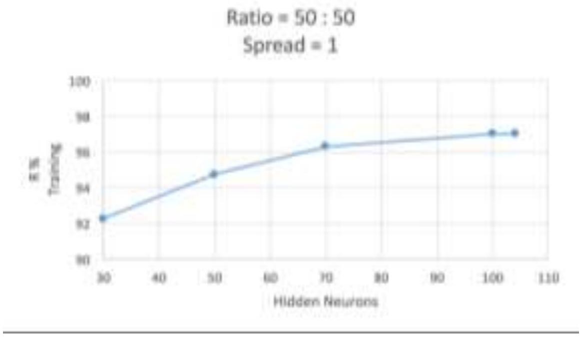
Similarly, network was trained using ratio 2 (50:50) with four various spread values (1, 0.161, 0.16 & 0.159) and hidden neuron values. Table 6.5 shows the networks performances with one ratio and different hidden neurons number and spread values.

Table 6.5: Results of RBFNN for ratio 2; 50 : 50

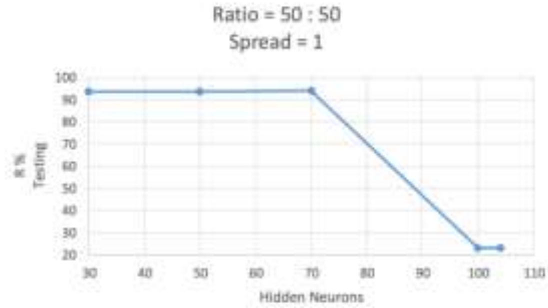
Ratio	Neuron	Spread	R Training %	R Testing %	R Overall %	MSE
2	30	1	92.27	93.6	92.935	0.008254
2	50	1	94.74	93.45	94.095	0.005684

2	70	1	96.32	94.02	95.17	0.004014
2	100	1	97.01	23.07	60.04	0.00327
2	104	1	97.02	23.2	60.11	0.003258
2	30	0.161	94.51	93.18	93.845	0.005932
2	**50**	**0.161**	**95.97**	**95.25**	**95.61**	0.0043823
2	70	0.161	96.77	89.24	93.005	0.003528
2	100	0.161	97.42	82.82	90.12	0.0028214
2	104	0.161	97.52	66.05	81.785	0.002723
2	30	0.16	94.16	92.94	93.55	0.0062976
2	40	0.16	95.45	93.04	94.245	0.0049387
2	50	0.16	96.37	87.34	91.855	0.003959
2	70	0.16	96.92	85.9	91.41	0.0033654
2	100	0.16	97.47	47.28	72.375	0.002776
2	104	0.16	97.51	45.2	71.355	0.0027319
2	30	0.159	94.16	92.92	93.54	0.006301
2	50	0.159	96.37	87.52	91.945	0.00396
2	70	0.159	96.91	86.07	91.49	0.0037755
2	100	0.159	97.35	42.23	69.79	0.0028983
2	104	0.159	97.54	39.03	68.285	0.0026928

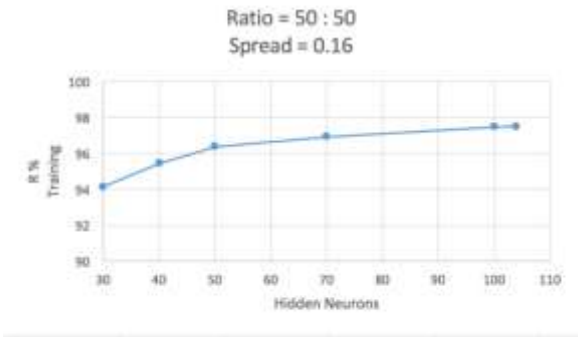
From table 6.5, it is noted that the networks trained with less number of data (i.e, 50 %) are ending up with lower testing correlations compared to those which uses ratio 1. This is concluded because the highest testing correlation achieved here is 95.25 %, However, in ratio 1, testing correlation of 96.07% was achieved. Moreover, it is seen that the spread value has a notable effect on the testing correlation. However, each spread value has a specific hidden neuron numbers. As the network skips that specific number, the correlation starts to decrease as seen in Figure 6.3.


Ratio = 50 : 50
Spread = 1
R %
Training
Hidden Neurons

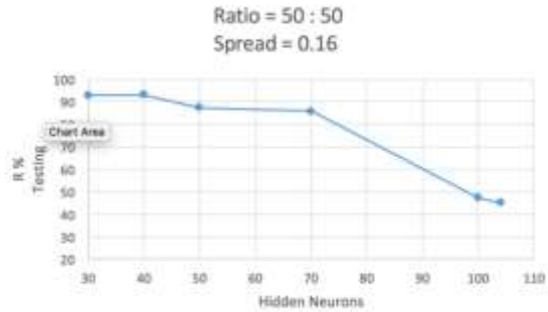
(a)


Ratio = 50 : 50
Spread = 1
R %
Testing
Hidden Neurons

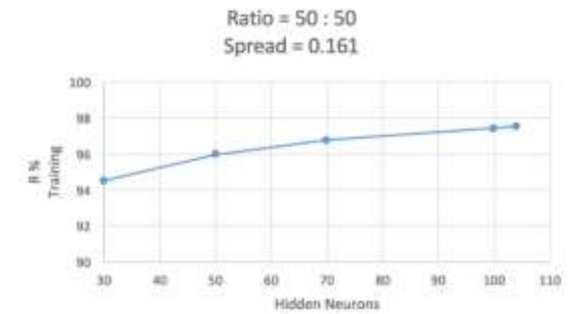
(b)


Ratio = 50 : 50
Spread = 0.16
R %
Training
Hidden Neurons

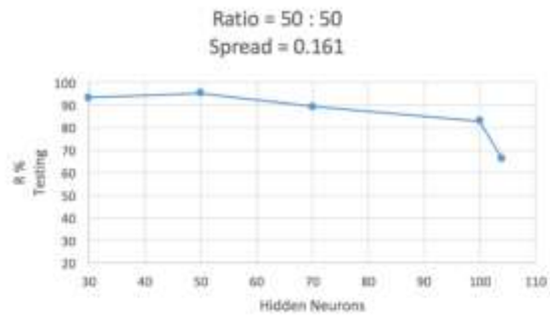
(c)


Ratio = 50 : 50
Spread = 0.16
Chart Area
R %
Testing
Hidden Neurons

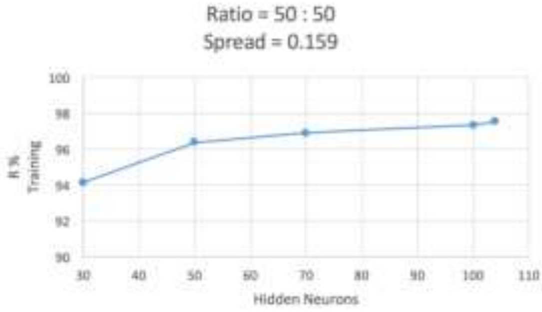
(d)


Ratio = 50 : 50
Spread = 0.161
R %
Training
Hidden Neurons

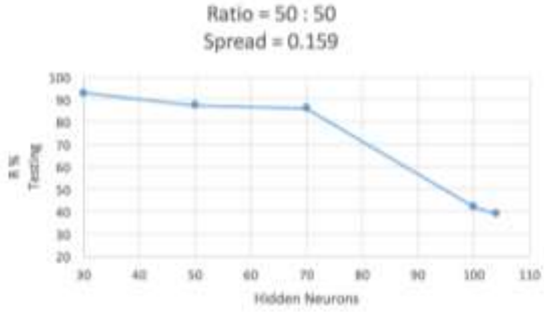
(e)


Ratio = 50 : 50
Spread = 0.161
R %
Testing
Hidden Neurons

(f)



(g)



(h)

Figure 6.3: Variation of training and testing correlations with respect to hidden neurons number and fixed spread value using ratio 2; 50:50

6.3.3 RBFNN Results for Ratio 3; 40 : 60

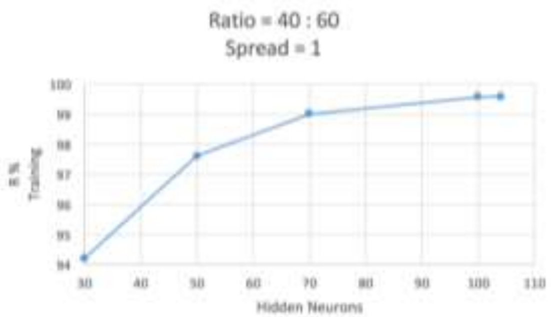
This ratio is also experimented with three different hidden neurons numbers and spread values. Table 6.6 show the network performances when trained on ratio 3 with different hidden neurons number and spread values of 1, 0.3, 0.157 and 0.142.

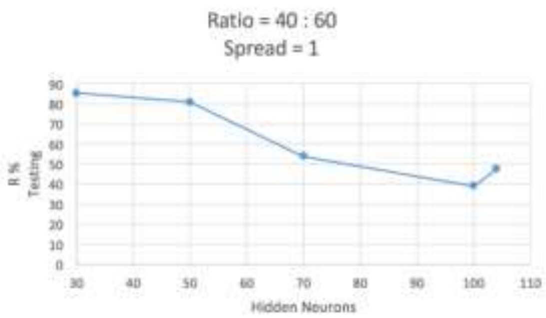
Table 6.6: Results of RBFNN for ratio 3; 40 : 60

Ratio	Neuron	Spread	R Training %	R Testing %	R Overall %	MSE
3	30	1	94.21	85.3	88.864	0.003878
3	50	1	97.61	80.83	87.542	0.0016257
3	70	1	99.02	53.84	71.912	0.000669
3	100	1	99.57	39.1	63.288	0.0002926
3	104	1	99.59	47.54	68.36	0.0002819
3	30	0.157	97.67	85.07	90.11	0.001589
3	50	0.157	99.23	86.55	91.622	0.0004669
3	60	0.157	99.49	88.1	92.656	0.0003483
3	70	0.157	99.58	84.63	90.61	0.0002888
3	100	0.157	99.91	29.33	57.562	<0.0001711
3	104	0.157	99.95	11.28	46.748	<0.0001711
3	30	0.42	95.34	87.87	90.858	0.003137

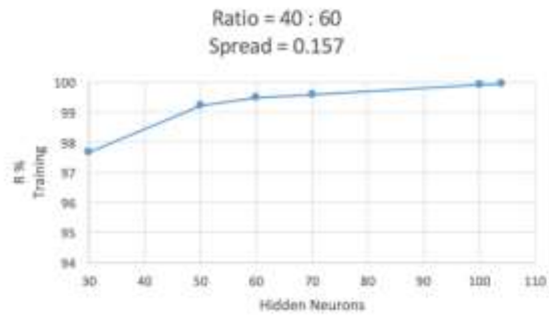
3	**36**	**0.42**	**97.22**	**90.16**	**92.98**	0.001888
3	50	0.42	98.48	81.86	88.508	0.001036
3	70	0.42	99.48	75.34	84.996	0.00035324
3	100	0.42	95.34	63.33	76.134	0.0002016
3	104	0.42	99.75	11.87	47.022	0.0001711
3	30	0.3	94.71	**90.27**	92.046	0.0035544
3	50	0.3	98.42	78.8	86.648	0.00108
3	70	0.3	99.44	72.33	83.174	0.000385
3	100	0.3	99.9	34.7	60.78	<0.0001711
3	104	0.3	99.9	16.07	49.602	<0.0001711

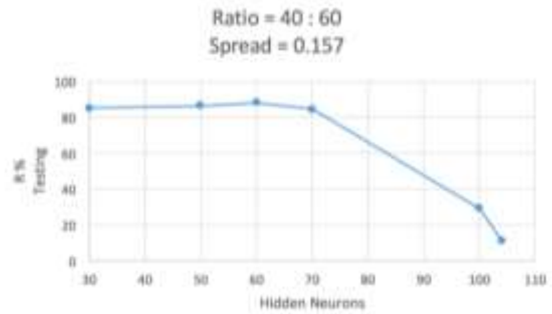
From this table, it is found that the best testing correlation (90.27 %) was archived when when the spread is 0.3 and number of neurons are 40. The highest overall correlation (92.98%) for this ratio was archived when the network used 36 neurons and 0.42 as a spread value. Also above table indicates that the testing correlations increase with the decrease of spread values till specific value of 0.3. Once the spread is reduced to 0.157 the testing correlations starts to decrease. This is due to the ratio used, which means that the effects of spread on the testing correlations are related to the train/test ratio employed. However, in here the spread has an optimum value, once it passes 0.3; the network performance starts to be lower.



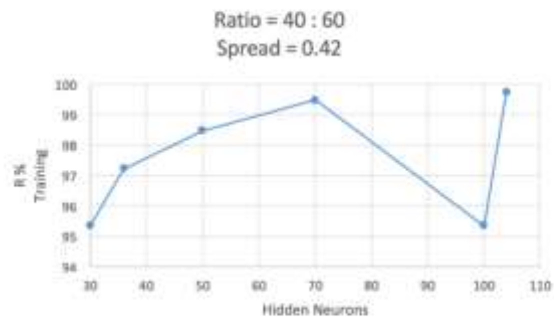


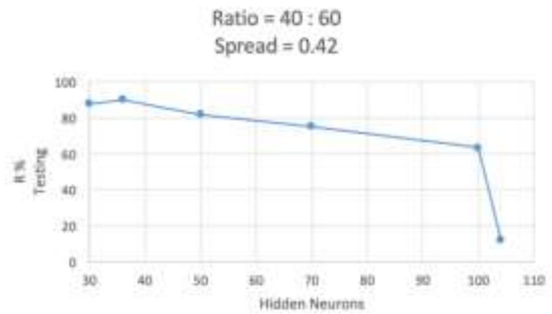
(a) (b)



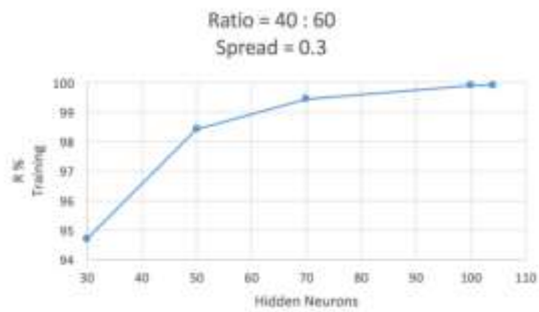


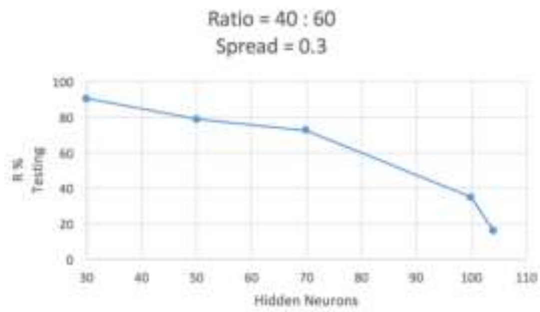
(c) (d)





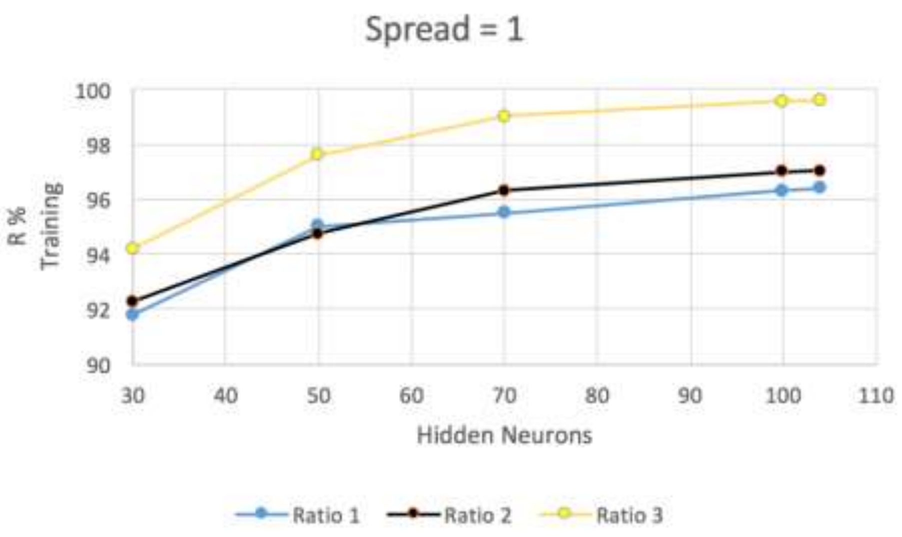
(e) (f)



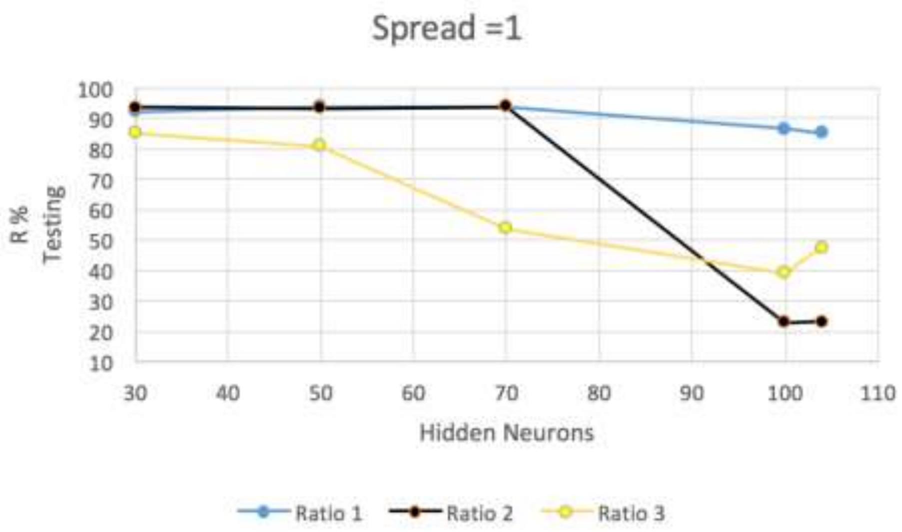


(g) (h)

Figure 6.4: Variation of training and testing correlations with respect to hidden neurons number and fixed spread value using ratio 3; 40:60



(a)



(b)

Figure 6.5: Variations of training and testing correlations with respect to hidden neurons number and fixed spread value (1) using all ratios

Figure 6.5 shows the training and testing correlations with respect to the three different employed ratios. From figure 6.5 (a) it is noted that network trained with ratio 3 (40:60) and spread 1 has achieved the highest training correlation. Also, it is seen that the training correlation is inversely proportional to the number of hidden neurons after a certain value of hidden neuron.

On the other hand, this relation cannot be applied to the Fig. 6.5(b) which shows the testing correlations. During testing, it is seen that the network trained with ratio 2, spread of 1 and 70 hidden neurons was capable of achieving the highest testing correlations (94.04%) in contrast to Fig. 6.5(a) in which the ratio 1 was the best. Furthermore, it can be seen that each ratio has a unique hidden neurons number, after which the network performance starts to decrease. For instance, the performance of the network trained with ratio 3 starts to decrease after passing the hidden neurons number of 50; however, the performance of the network trained with ratio 1 starts to decline once the hidden neurons number passes 70.

6.4 Discussions

A comparison of the developed networks employed in this work with some earlier works is shown in Table 4. Note that only works which provide explicitly achieved accuracies and number of data used for train and test are considered for comparison. Our results can show that applying backpropagation neural network and radial Basis function network to the problem of concrete strength prediction is promising, in a way that correlations achieved in some experiments were relatively good.

Table 6.7: Comparison between results of this study with other studies

Network	Parameters		
	Dataset	**Train/Test Ratio**	**Overall Accuracy**
BPNN	326	60:40	96.25%
RBFNN	326	60:40	96.78%
BPNN (Khashman & Akpinar., 2017)	1030	50:50	86.02%
BPNN (Chithra et al., 2016)	264	70:30	99.87%
EANNs (Nikoo et al., 2015)	173	80:10:10	93.5%

Firstly, it is seen that the employed backpropagation neural networks achieved higher correlation compared to some other works such as (Khashman and Akpinar., 2017) and (Chithra et al., 2016) networks. The used nertowrk by Nikoo et al., (2015) is a combination of evolutionary algorithm with a backpropagation algorithm that was trained to predict strength of concrete. On the other hands, Chithra et al., (2016) networks were outperformed our employed network, which can be probably due to the use of higher rate of training as compared to the train ratio used in this work.Moreover, the radial basis function network was also compared to some related works, in which the concrete strength was predicted using RBFN. Wang and Xu., (2013) applied an algorithm for determining the spread value in RBF network using a genetic algorithm. the authors concluded that if the spread constant is too big, much more neurons are demanded to adapt function's rapid change, and the result will be not accurate enough but can only show the approximate trend of function (underfitting); if the spread is too small then more neurons are demanded to adapt function's slow change, the result will be accurate enough but cannot represent change trend of function (overfitting). Note that this is the same conclusion that was derived through our study on the effects of spread value and hidden neurons number on the learning process of the RBFN.

CHAPTER 7
CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

In this thesis, neural networks are implemented for the task of non-destructive prediction of concrete's compressive strength. Neural networks have shown a good efficacy in prediction tasks, depending on the defined ask parameters and their relevance to the predicted outcome. Thus, in this study, parameters that are conventionally known to influence the strength of concrete were proposed to be used. Some of these parameters were selected through a review of many published articles that discussed the parameters that determine the strength of concrete. However, the remaining input parameters were proposed for very first time, as they are known to affect the final chemical composition of concrete mixes; therefore, they are considered to affect the concrete's strength. These new input parameters include CaO, SiO_2, Al_2O_3 of cement, CaO, SiO_2 and Al_2O_3 of fly ash.

Since neural networks are data-hungry systems, numbers data were collected from different articles published in respected literature sources. 326 samples were collected, classified systemically according to inputs and normalized to be then used for learning and testing purposed of the networks.

Two different neural networks were selected to be the classifiers behind this prediction task. Radial basis function neural network (RBFNN) and back propagation neural network (BPNN) are utilized in order to predict concrete compressive strength. Networks performance was evaluated through conducting different experiments.

Conclusions drawn from the results of the studies made are as the following:

- Ratio 1 (60:40) achieved best overall correlation in both BPNN and RBFNN. This implies increasing train: test ratio yield result in better prediction accuracy.
- All the networks that archived highest correlation in training dose not necessarily have high generalization capability.
- The best overall correlation of BPNN is achieved with number of epochs of 50.

- The training correlation of BPNN and RBFNN improves with the increasing of hidden neurons.
- Each value of hidden neuron in RBFNN has an optimum value of spread. When the spread exceeds this value, testing correlation stars to decrease.
- The best values of hidden neuron and spread in RBFNN are 104 and 0.1349 respectively.
- The best prediction accuracy of is 96.78% and it was achieved by RBFNN.
- The least mean square error was 0.00015 and it was achieved by BPNN.

The obtained results were all compared with works in literature and it was seen that the proposed networks outperformed the other related works in terms of overall correlation.

7.2 Recommendations for Further Studies

Results found in this study indicate that the network achieved the best when specific hidden neuron numbers and spread value are selected. Therefore, it is recommended that further hidden neurons and other learning parameter combination are tested in order to ensure the optimum values of parameters yielding the further increased accuracy in strength prediction. Also, using different learning method of RBFNN (hybrid) is recommend for predicting concrete compressive strength.

REFERENCES

Abdi, H., Valentin, D., Edelman, B., & O'Toole, A. J. (1996). A Widrow–Hoff learning rule for a generalization of the linear auto-associator. *Journal of Mathematical Psychology*, *40*(2), 175-182.

Alshihri, M. M., Azmy, A. M., & El-Bisy, M. S. (2009). Neural networks for predicting compressive strength of structural light weight concrete. *Construction and Building Materials*, *23*(6), 2214-2219.

Atici, U. (2011). Prediction of the strength of mineral admixture concrete using multivariable regression analysis and an artificial neural network. *Expert Systems with applications*, *38*(8), 9609-9618.

Atiş, C. D. (2003). Accelerated carbonation and testing of concrete made with fly ash. *Construction and Building Materials*, *17*(3), 147-152

Atiş, C. D. (2005). Strength properties of high-volume fly ash roller compacted and workable concrete, and influence of curing condition. *Cement and Concrete Research*, *35*(6), 1112-1121

Bharathi, S. D., Manju, R., & Premalatha, J. Prediction of Compressive Strength for Self-Compacting Concrete (SCC) using Artificial Intelligence and Regression Analysis.

Bilim, C., Atiş, C. D., Tanyildizi, H., & Karahan, O. (2009). Predicting the compressive strength of ground granulated blast furnace slag concrete using artificial neural network. *Advances in Engineering Software*, *40*(5), 334-340.

Bingöl, A. F., Tortum, A., & Gül, R. (2013). Neural networks analysis of compressive strength of lightweight concrete after high temperatures. *Materials & Design*, *52*, 258-264.

Boukhatem, B., Kenai, S., Hamou, A. T., Ziou, D., & Ghrici, M. (2012). Predicting concrete properties using neural networks(NN) with principal component analysis(PCA) technique. *Computers & Concrete*, *10*(6), 557-573.

Buhmann, M. D. (2011). Radial basis function networks. In *Encyclopedia of Machine Learning* (pp. 823-827). Springer US.

Chithra, S., Kumar, S. S., Chinnaraju, K., & Ashmita, F. A. (2016). A comparative study on the compressive strength prediction models for High Performance Concrete containing nano silica and copper slag using regression analysis and Artificial Neural Networks. *Construction and Building Materials*, *114*, 528-535.

Choi, S. W., Jang, B. S., Kim, J. H., & Lee, K. M. (2014). Durability characteristics of fly ash concrete containing lightly-burnt MgO. *Construction and Building Materials, 58,* 77-84

Dantas, A. T. A., Leite, M. B., & de Jesus Nagahama, K. (2013). Prediction of compressive strength of concrete containing construction and demolition waste using artificial neural networks. *Construction and Building Materials*, *38*, 717-722.

Douma, O. B., Boukhatem, B., Ghrici, M., & Tagnit-Hamou, A. (2016). Prediction of properties of self-compacting concrete containing fly ash using artificial neural network. *Neural Computing and Applications*, 1-12.

Du, K. L., & Swamy, M. N. (2013). *Neural networks and statistical learning*. Springer Science & Business Media.

Duan, Z. H., Kou, S. C., & Poon, C. S. (2013). Prediction of compressive strength of recycled aggregate concrete using artificial neural networks. *Construction and Building Materials*, *40*, 1200-1206.

Fausett, L. (1994). *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Inc.

Gholampour, A., & Ozbakkaloglu, T. (2017). Performance of sustainable concretes containing very high volume Class-F fly ash and ground granulated blast furnace slag. *Journal of Cleaner Production*, *162*, 1407-1417

Gurney, K. (1997). *An introduction to neural networks*. CRC press.

Hakim, S. J. S., Noorzaei, J., Jaafar, M. S., Jameel, M., & Mohammadhassani, M. (2011). Application of artificial neural networks to predict compressive strength of high strength concrete. *International Journal of Physical Sciences*, *6*(5), 975-981.

Halstead, W. J. (1986). Use of fly ash in concrete. *NCHRP Synthesis of Highway Practice*, (127).

Haykin, S. S. (2009). *Neural networks and learning machines* (Vol. 3). Upper Saddle River, NJ, USA:: Pearson.

Kewalramani, M. A., & Gupta, R. (2006). Concrete compressive strength prediction using ultrasonic pulse velocity through artificial neural networks. *Automation in Construction*, *15*(3), 374-379.

Khan, M. I. (2012). Mix proportions for HPC incorporating multi-cementitious composites using artificial neural networks. *Construction and Building Materials*, *28*(1), 14-20.

Khashman, A., & Akpinar, P. (2017). Non-Destructive Prediction of Concrete Compressive Strength Using Neural Networks. *Procedia Computer Science*, *108,* 2358-2362.

Kim, J. I., Kim, D. K., Feng, M. Q., & Yazdani, F. (2004). Application of neural networks for estimation of concrete strength. *Journal of Materials in Civil Engineering*, *16*(3), 257-264.

Kiran Kumar Kaki," Parallelized Backpropagation Neural Network Algorithm using Distributed System", Thesis submitted in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science and Engineering, January 2009.

Kriesel, D. (2007). A brief introduction on neural networks.

Lai, S., & Serra, M. (1997). Concrete strength prediction by means of neural network. *Construction and Building Materials*, *11*(2), 93-98.

Lee, S. C. (2003). Prediction of concrete strength using artificial neural networks. *Engineering Structures*, *25*(7), 849-857.

Liu, M. (2010). Self-compacting concrete with different levels of pulverized fuel ash. *Construction and Building Materials, 24*(7), 1245-1252

Maillard, E. P., & Gueriot, D. (1997, June). RBF neural network, basis functions and genetic algorithm. In *Neural Networks, 1997., International Conference on* (Vol. 4, pp. 2187-2192). IEEE.

Markopoulos, A. P., Georgiopoulos, S., & Manolakos, D. E. (2016). On the use of back propagation and radial basis function neural networks in surface roughness prediction. *Journal of Industrial Engineering International*, *12*(3), 389-400.

Mehta, P. K. (2006). Concrete. microstructure, properties and materials.

Ndihokubwayo, A. (2011). Compressive and flexural strengths for considerable volume fly-ash concrete. *Journal of Civil Engineering Research*, *1*(1), 21-23.

Neville, A. M. (1995). *Properties of concrete* (Vol. 4). London: Longman.

Neville, A. M., & Brooks, J. J. (1987). *Concrete technology*. (Neville and Brooks, 2010)

Ni, H. G., & Wang, J. Z. (2000). Prediction of compressive strength of concrete by neural networks. *Cement and Concrete Research*, *30*(8), 1245-1250.

Nikoo, M., Torabian Moghadam, F., & Sadowski, Ł. (2015). Prediction of concrete compressive strength by evolutionary artificial neural networks. *Advances in Materials Science and Engineering*, *2015*

Oner, A., Akyuz, S., & Yildiz, R. (2005). An experimental study on strength development of concrete containing fly ash and optimum usage of fly ash in concrete. *Cement and Concrete Research*, *35*(6), 1165-1171

Orr, M. J. (1996). Introduction to radial basis function networks.

Özcan, F., Atiş, C. D., Karahan, O., Uncuoğlu, E., & Tanyildizi, H. (2009). Comparison of artificial neural network and fuzzy logic models for prediction of long-term compressive strength of silica fume concrete. *Advances in Engineering Software*, *40*(9), 856-863.

Öztaş, A., Pala, M., Özbay, E., Kanca, E., Caglar, N., & Bhatti, M. A. (2006). Predicting the compressive strength and slump of high strength concrete using neural network. *Construction and Building Materials*, *20*(9), 769-775.

Poon, C. S., Lam, L., & Wong, Y. L. (2000). A study on high strength concrete prepared with large volumes of low calcium fly ash. *Cement and Concrete Research*, *30*(3), 447-455

Prasad, B. R., Eskandari, H., & Reddy, B. V. (2009). Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN. *Construction and Building Materials*, *23*(1), 117-128.

Ramezanianpour, A. A. (2014). Fly ash. In *Cement replacement materials* (pp. 47-156). Springer, Berlin, Heidelberg.

Ramezanianpour, A. A., & Malhotra, V. M. (1995). Effect of curing on the compressive strength, resistance to chloride-ion penetration and porosity of concretes incorporating slag, fly ash or silica fume. Cement and concrete composites, 17(2), 125-133

Rebouh, R., Boukhatem, B., Ghrici, M., & Tagnit-Hamou, A. (2017). A practical hybrid NNGA system for predicting the compressive strength of concrete containing natural pozzolan using an evolutionary structure. *Construction and Building Materials*, *149*, 778-789.

Sadrmomtazi, A., Sobhani, J., & Mirgozar, M. A. (2013). Modeling compressive strength of EPS lightweight concrete using regression, neural network and ANFIS. *Construction and Building Materials*, *42*, 205-216.

Saha, A. K. (2018). Effect of class F fly ash on the durability properties of concrete. *Sustainable Environment Research*, *28*(1), 25-31.

Sarıdemir, M. (2009). Prediction of compressive strength of concretes containing metakaolin and silica fume by artificial neural networks. *Advances in Engineering Software*, *40*(5), 350-355.

Schwenkar, F., Kestler, H. A., & Palm, G. (2000). Radial-basis-function networks: learning and applications. In *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on* (Vol. 1, pp. 33-43). IEEE.

Sebastia, M., Olmo, I. F., & Irabien, A. (2003). Neural network prediction of unconfined compressive strength of coal fly ash–cement mixtures. *Cement and Concrete Research*, *33*(8), 1137-1146.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shetty, M. S. (2005). *Concrete Technology Theory & Practice*, Published by S. CHAND & Company, Ram Nagar, New Delh.

Siddique, R., Aggarwal, P., & Aggarwal, Y. (2011). Prediction of compressive strength of self-compacting concrete containing bottom ash using artificial neural networks. *Advances in Engineering Software*, *42*(10), 780-786.

Singh, V. P., & Kotiyal, Y. C. (2013). Prediction of compressive strength using artificial neural network. *International Journal of Civil, Environmental, Structural, Construction and Architectural Engineering*, *7*(12), 976-980.

Słoński, M. (2010). A comparison of model selection methods for compressive strength prediction of high-performance concrete using neural networks. *Computers & structures*, *88*(21), 1248-1253.

Sobhani, J., Najimi, M., Pourkhorshidi, A. R., & Parhizkar, T. (2010). Prediction of the compressive strength of no-slump concrete: A comparative study of regression, neural network and ANFIS models. *Construction and Building Materials*, *24*(5), 709-718

Sounthararajan, V. M., & Sivakumar, A. (2013). Accelerated engineering properties of high and low volume fly ash concretes reinforced with glued steel fibers. *Frontiers of Structural and Civil Engineering*, *7*(4), 429-445

Stanford university. (2017). Neural networks. Retrieved from https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html

Tino, P., Benuskova, L., & Sperduti, A. (2015). Artificial Neural Network Models. In *Springer Handbook of Computational Intelligence* (pp. 455-471). Springer, Berlin, Heidelberg.

Topcu, I. B., & Sarıdemir, M. (2007). Prediction of properties of waste AAC aggregate concrete using artificial neural network. *Computational Materials Science*, *41*(1), 117-125.

Topcu, I. B., & Sarıdemir, M. (2008). Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic. *Computational Materials Science*, *41*(3), 305-311.

Topçu, İ. B., & Sarıdemir, M. (2008). Prediction of mechanical properties of recycled aggregate concretes containing silica fume using artificial neural networks and fuzzy logic. *Computational Materials Science*, *42*(1), 74-82.

Trtnik, G., Kavčič, F., & Turk, G. (2009). Prediction of concrete strength using ultrasonic pulse velocity and artificial neural networks. *Ultrasonics*, *49*(1), 53-60.

Uysal, M., & Tanyildizi, H. (2011). Predicting the core compressive strength of self-compacting concrete (SCC) mixtures with mineral additives using artificial neural network. *Construction and Building Materials*, *25*(11), 4105-4111.

Vidivelli, B., & Jayaranjini, A. (2016). Prediction of Compressive Strength of High Performance Concrete Containing Industrial by products Using Artificial Neural Networks. *International Journal of Civil Engineering and Technology*, *7*(2).

Wang, H., & Xu, X. (2013). Determination of spread constant in RBF neural network by genetic algorithm. *Int. J. Adv. Comput. Technol.(IJACT)*, *5*(9), 719-726.

Wongkeo, W., Thongsanitgarn, P., Ngamjarurojana, A., & Chaipanich, A. (2014). Compressive strength and chloride resistance of self-compacting concrete containing high level fly ash and silica fume. *Materials & Design*, *64*, 261-269

Wu, Y., Wang, H., Zhang, B., & Du, K. L. (2012). Using radial basis function networks for function approximation and classification. *ISRN Applied Mathematics*, *2012*.

Xiao, N. Y. (1996). Using the modified back-propagation algorithm to perform automated downlink analysis (Doctoral dissertation, Massachusetts Institute of Technology).

Xie, T., Yu, H., & Wilamowski, B. (2011, June). Comparison between traditional neural networks and radial basis function networks. In *Industrial Electronics (ISIE), 2011 IEEE International Symposium on* (pp. 1194-1199). IEEE.

Yeh, I. C. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, *28*(12), 1797-1808.

Yuan, Z., Wang, L. N., & Ji, X. (2014). Prediction of concrete compressive strength: Research on hybrid models genetic based algorithms and ANFIS. *Advances in Engineering Software*, *67*, 156-163.

Zhang, M. H., & Islam, J. (2012). Use of nano-silica to reduce setting time and increase early strength of concretes with high volumes of fly ash or slag. *Construction and Building Materials*, *29*, 573-580

Zhang, Q. J., & Gupta, K. C. (2000). *Neural networks for RF and microwave design (Book+ Neuromodeler Disk)*. Artech House, Inc.

Zurada, J. M. (1992). *Introduction to artificial neural systems*(Vol. 8). St. Paul: West

APPENDICES

APPENDIX 1
MATLAB CODE FOR BPNN

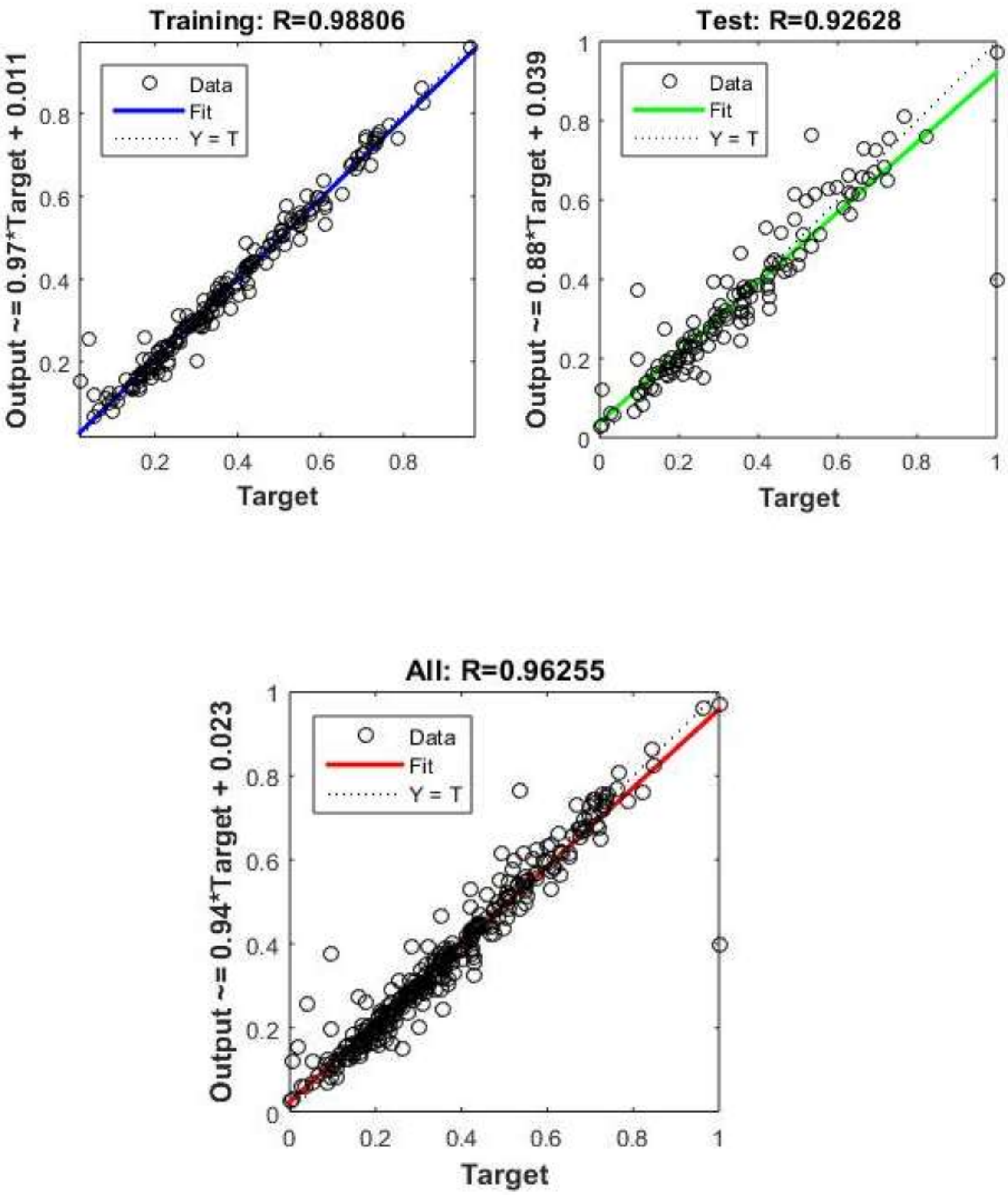
```
%generate inputs and output data

input=xlsread('Data326all.xlsx', 'allinputs');

target=xlsread('Data326all.xlsx', 'alloutputs');

%Normalization for inputs and output

input=mat2gray(input);

target=mat2gray(target);

% creating and initiating the network

net = newff(minmax(input),[70 1],{'logsig','logsig'},'trainlm');

% Defining network parameters

net.trainParam.goal = 0.0000000000000000001; % Sum-squared error goal.

net.trainParam.lr = 0.09;  % Learning Rate.

net.trainParam.epochs =25;% Maximum number of epochs to train.

net.trainParam.mc = 0.5 % Momentum Factor.

net.trainParam.max_fail = 500;

net.divideFcn= 'dividerand'; % divide the data randomly

net.divideParam.trainRatio= 40/100; % for training

net.divideParam.valRatio= 0/100; % for validation

net.divideParam.testRatio= 60/100; % for testing

[net,tr] = train(net,input,target);

ActualOutput=sim(net, input)

errors = target-ActualOutput;

% Mean squared error
```

```
net.performFcn = 'mse';

% Choose Plot Functions

net.plotFcns = {'plotperform','plottrainstate','ploterrhist', 'plotregression', 'plotfit'};
```

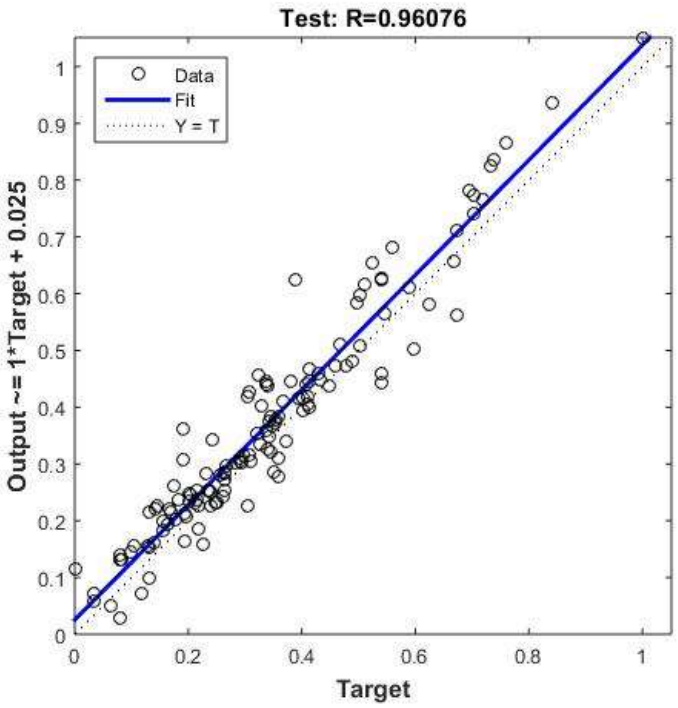
APPENDIX 2
MATLAB CODE FOR RBFNN

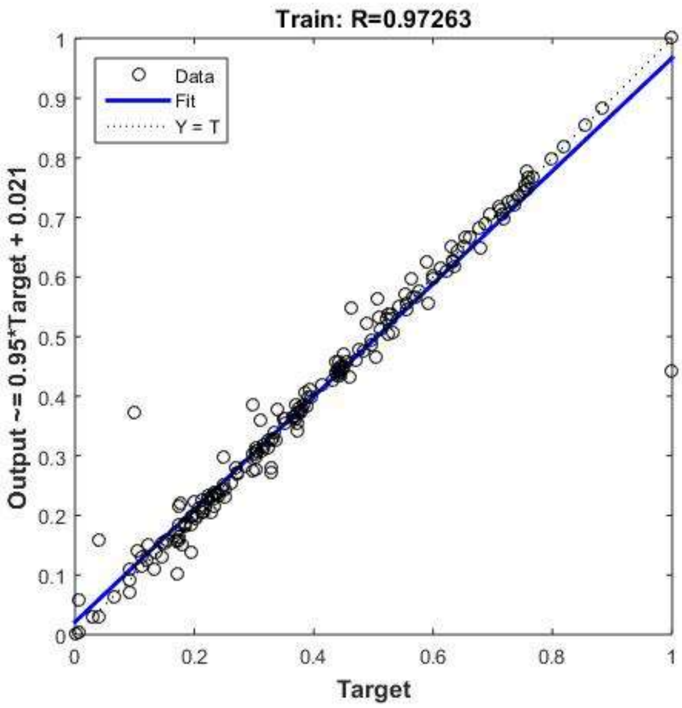
```
%generate train data
p =xlsread('Data326Ratio2.xlsx', 'traininputs');
t =xlsread('Data326Ratio2.xlsx', 'trainoutputs');
%generate test data
p1 = xlsread('Data326Ratio2.xlsx', 'testinputs');
t1 = xlsread('Data326Ratio2.xlsx', 'testoutputs');
%Normalization for p, t, p1 and t1
p=mat2gray(p);
t=mat2gray(t);
p1=mat2gray(p1);
t1=mat2gray(t1);
% choose a spread constant
spread = 0.161;
% choose max number of neurons
K = 50;
% performance goal
goal = 0;
% number of neurons to add between displays
Ki = 1;
% create a neural network
net = newrb(p,t,goal,spread,K,Ki);
% simulate trained network for train inputs (Training)
Yo = sim(net,p);
% simulate trained network for test inputs (Testing)
Y1 = net(p1);
figure,plotregression(t,Yo,'Train')
figure,plotregression(t1,Y1,'Test')
```

APPENDIX 3
REGRESSION PLOT FOR BEST PERFORMED NETWORKS IN THIS STUDY



Regression plot for Best BPNN





Regression plot for Best RBFNN

Secure | https://turnitin.com/t_inbox.asp?r=36.1832843727335&svr=338&lang=en_us&aid=68823229

Pınar Akpınar | User Info | Messages | Instructor ▼ | English ▼ | Community | Help | Logout

turnitin

Assignments | Students | Grade Book | Libraries | Calendar | Discussion | Preferences

NOW VIEWING:

HOME > MSC-THESIS-A NEW APPROACH FOR NON-DESTRUCTIVE DETERMINATION OF CONCRETE COMPRESSIVE STRENGTH BY USING RADIAL BASIS FUNCTION NETWORK > COMPLETETHESIS

About this page

This is your assignment inbox. To view a paper, select the paper's title. To view a Similarity Report, select the paper's Similarity Report icon in the similarity column. A ghosted icon indicates that the Similarity Report has not yet been generated.

CompleteThesis

INBOX | NOW VIEWING: NEW PAPERS ▼

Submit File

Online Grading Report | Edit assignment settings | Email non-submitters

AUTHOR	TITLE	SIMILARITY	GRADE	RESPONSE	FILE	PAPER ID	DATE
Saddam Naser Atiyah ...	Abstract	0%	--	--		962350907	11-May-2018
Saddam Naser Atiyah ...	Chapter-1	0%	--	--		962351033	11-May-2018
Saddam Naser Atiyah ...	Conclusions	0%	--	--		962363167	11-May-2018
Saddam Naser Atiyah ...	Results	0%	--	--		962363040	11-May-2018
Saddam Naser Atiyah ...	Chapter-5	1%	--	--		962351696	11-May-2018
Saddam Naser Atiyah ...	Chapter-2	9%	--	--		962351404	11-May-2018
Saddam Naser Atiyah ...	Chapter-3	9%	--	--		962351247	11-May-2018
Saddam Naser Atiyah ...	Complete Thesis	11%	--	--		962362782	11-May-2018
Saddam Naser Atiyah ...	Chapter-4	12%	--	--		962351544	11-May-2018