Bachelor's Thesis

Bachelor's degree in Industrial Technology Engineering

Artificial Neural Network System Applied to Human Resource Management

Project Report

Barcelona School of Industrial Engineering

Author: Anna Farrés Rocabert

Director: Cecilio Angulo

Collaborator: Jenn Nguyen

Automatic Control Department

Date: September 2017

Summary

In this project we study how Artificial Neural Network can be applied to Human Resources by supporting users from XING, a career-oriented social network, to find the desired job, and also recruiters that use this resource to find the best candidate for a given job. Specially, an ANN is created, using NN Toolbox in Matlab, with the purpose of predicting those users that given an specific job offer are likely to interact by performing a bookmark or reply on the job post recommended. The sample used contains 714.800 user-item pairs with 19 inputs that give information about the users and the job posts and 1 output that determines wether the user has positively interact with the job offer or not. The ANN created is a two layer feed-forward network that uses Levenberg-Marquardt backpropagation algorithm and with the correct number of neurons on the hidden layer does a good prediction. Further research can be done to improve Human Resource Management using data mining.

En este proyecto estudiamos cómo la Red Neuronal Artificial puede tener aplicaciones en el área de Recursos Humanos apoyando a los usuarios de XING, una red social de ámbito profesional, a encontrar el trabajo deseado, y también a los reclutadores que utilizan este recurso a encontrar el mejor candidato para una determinada oferta de trabajo. Especialmente, se crea una RNA con la Toolbox de Redes Neuronales de Matlab, con el propósito de predecir qué usuarios guardaran o aplicaran para una determinada oferta de trabajo que se les recomienda. La muestra utilizada contiene 714.800 parejas usuario-oferta con 19 entradas que proporcionan información sobre los usuarios y los puestos de trabajo y una salida que determina si el usuario ha interactuado positivamente con la oferta de trabajo o no. La RNA creada es una red de feed-forward de dos capas que utiliza el algoritmo de propagación hacia atrás de Levenberg-Marquardt y con el número correcto de neuronas en la capa oculta hace una buena predicción. En un futuro, se pueden hacer más investigaciones para mejorar la Gestión de Recursos Humanos haciendo uso de la minería de datos.

Table of Contents

Summary ... 1
Summary tables ... 5
Summary figures ... 5
1. Glossary ... 6
2. Preface ... 7
2.1 Origin of the Project and Motivation ... 7
2.2 Previous requirements ... 8
2.3 Report structure ... 8
3. Introduction ... 9
4. Neural Networks History ... 10
4.1 Beginning of NN ... 10
4.2 Golden age ... 10
4.3 Quiet years ... 11
4.4 Innovation ... 12
4.5 Today ... 12
4.6 Future ... 12
5. Biological vs. artificial neural networks ... 13
6. Artificial Neural networks ... 15
6.1 Introduction ... 15
6.2 Neural network operation ... 15
6.3 Types of neural networks ... 16
6.4 Type of learning ... 17
6.5 Training of the neural network ... 17
6.6 Over learning or loss of generalization ... 18
6.7 Training, validation and test data ... 19
6.8 Techniques to avoid the loss of generalization ... 19
6.8.1 Early-stopping ... 19
6.8.2 Regularization ... 20
6.9 Pruned nets ... 20
6.10 Size of the neural network ... 21
6.11 Standardization of data ... 21
7. Applications ... 22
7.1 Business ... 22
7.2 Banking and Finance ... 22

7.3 Industrial and operations management ... 23

7.4 Operational analysis ... 23

7.5 Data mining ... 23

7.6 Medical ... 24

7.7 HR Management ... 24

7.8 Telecomunications ... 25

7.9 Speech ... 25

7.10 Insurance ... 25

7.11 Energy ... 25

7.12 Defense and security ... 25

7.13 Transportation, Aerospace and Automotive ... 25

8. Advantages of Neural Networks ... 26

9. Disadvantages of Neural Networks ... 27

10. State-of-the-Art in Human Resources ... 28

11. Programs to analyse, process and prepare the data ... 32

11.1 Excel and PowerPivot ... 32

11.2 Access ... 32

12. Programs to create an Artificial Neural Network ... 33

12.1 Matlab Toolbox ... 33

12.1.1 Data division ... 34

12.1.2 Network Architecture ... 34

12.1.3 Mean Squared Error and Regression values ... 34

12.1.4 Training Algorithms ... 34

12.1.5 Regression plots ... 35

12.1.6 ANN Performance ... 35

12.1.7 Error histogram ... 35

13. Database ... 36

13.1 Introduction ... 36

13.2 Description ... 37

13.2.1 Interactions ... 37

13.2.2 Users ... 39

13.2.3 Items ... 41

14. Data analysis of users and items ... 43

14.1 Career level ... 44

14.2 Country and region ... 44

14.3 Discipline ... 45

14.4 Industry ... 46
14.5 Users analysis of data ... 46
14.6 Items analysis of data ... 47
15. RecSys Challenge solutions ... 48
16. Choosing the right data ... 54
16.1 Users recommendations ... 54
16.2 Input and output data ... 54
17. Creating the ANN for users recommendation ... 56
17.1 Architecture ... 56
17.2 Algorithm ... 56
17.3 Input and output ... 57
17.4 Type of learning and data division ... 57
17.5 Number of hidden neurons ... 57
17.6 Script ... 58
18. Results ... 59
18.1 Regression values ... 59
18.2 Mean Squared Error ... 60
18.3 Number of iterations ... 60
18.4 Training Time ... 61
18.5 Final Discussion ... 61
Conclusions ... 65
Bibliography ... 66
Annex ... 68

Summary tables

Table 1. Biological vs. Artificial Neural Network 14
Table 2. Interactions data description 37
Table 3. Counting of interaction types 38
Table 4. Users data description 40
Table 5. Items data description 42
Table 6. Same information for Users and Items 43
Table 7. Comparison of users and items discipline 45
Table 8. Comparison of users and items industry 46
Table 9. User's field of studies 47
Table 10. Comparison between outputs and targets 62
Table 11. Summary of ANN results 68

Summary figures

Figure 1. Biological vs. Artificial Neurons 13
Figure 2. Tasks that Neural Networks can perform 15
Figure 3. Layer of neurons 15
Figure 4. Weight Matrix 16
Figure 5. Transfer functions types 16
Figure 6. Neural Networks Architectures 17
Figure 7. Example of over-learning 18
Figure 8. Comparison before and after pruning 20
Figure 9. ANN Operation Process 33
Figure 10. Bar Chart Career_level_items 44
Figure 11. Bar Chart Career_level_users 44
Figure 12. Bar Chart Country_users 44
Figure 13. Bar Chart Country_items 44
Figure 14. Pie Chart Region_users 45
Figure 15. Pie Chart Region_items 45
Figure 16. Edu_degree of users 46
Figure 17. CV_Entries on user's profile 46
Figure 18. User's years in current job 47
Figure 19. User's years of experience 47
Figure 20. Premium users 47
Figure 21. User's willingness to change job 47
Figure 22. Architecture of users recommendation ANN 56
Figure 23. 2D line plot for Regression R values 59
Figure 24. 2D line plot for MSE values 60
Figure 25. 2D plot for Number of Iterations 60
Figure 26. 2D plot for Training time 61
Figure 27. Error Histogram for ANN with 15 hidden neurons 61
Figure 28. Regression Plot for ANN with 15 hidden neurons 62
Figure 29. Validation performance plot for ANN with 15 hidden neurons 63
Figure 30. Range of output values for ANN with 15 hidden neurons 63
Figure 31. Confusion Matrix 64
Figure 32. Confussion Matrix with results 64

1. Glossary

AHP	Analytic hierarchy process
AI	Artificial Intelligence
ANN	Artificial Neural Network
ASP	Active Server Page
BP	Back Propagation
CBF	Content-Based Filtering
CF	Collaborative Filtering
DBMS	Database Management System
HR	Human Resources
HRM	Human Resource Management
HTML	Hypertext Markup Language
I/O psychology	Industrial and organizational psychology
KNN	K Nearest Neighbours
MIT	Massachusetts Institute of Technology
ML	Machine Learning
MLR	Machine-Learned-Ranking
MSE	Mean Squared Error
NN	Neural Network
RNN	Recurrent Neural Network
RS	Recommender Systems
R&D	Research and Development

2. Preface

2.1 Origin of the Project and Motivation

As a student of Industrial Engineering and Psychology, lots of people has asked me through these last years which was the point of studying these two degrees if they had nothing to do. Every time I thought about and answer to that question I found more sense of having knowledge from both fields.

I am a passionate about neuroscience and how the human brain works and the first time I read about Artificial Neural Networks I thought it was a great idea to imitate the characteristics of biologic nervous system.

After a little research I found out about the magnificent world of Artificial Intelligence and Machine Learning and the wide range of applications that it can be used to. Luckily the bit of computer science I’ve learned on my engineering degree let me do a better understanding of the algorithms and technical aspects used on AI.

¿What about all the data collected form our clicks while we navigate? Recommender systems use big data to do recommendations to users in a wide variety of products. It is used in Facebook to do friends and posts recommendations, also in YouTube to recommend videos, in Amazon to show the right products to the users…

As I am also interested in Industrial and Organizational Psychology I wanted to make a project that unified both my psychology and engineering interests. So, I came up with the idea of using ANN for Human Resources management. Then, instead of using the RS techniques such as Collaborative Filtering or User-Based Filtering I decided it was a good idea to see how ANN could be modelled for a recommender application.

My project director suggested that we could use big data form the career oriented social network XING in order to apply real data to a ML model such as ANN and see how it can perform good recommendations for those job seekers that are looking for a job and for those recruiters that are looking for candidates that accomplish the requirements of a specific job.

Xing data was used in RecSys Challenge 2016, where the teams that participated had to design and implement RS algorithms creating recommendations for both job seekers and recruiters.

2.2 Previous requirements

The basic requirements for being able to develop this project was, first of all, having a good knowledge of ANN, for that reason and under my collaborator advice I subscribed to the free course Neural Network for Machine Learning in Coursera, organized by the University of Toronto. Nevertheless, the course was too slow and I wasn’t learning the details I needed to develop my project. So, I decided to search articles and tutorials about ANN.

Also, it was interesting to know about the different programs I could use in order to deal with the big dataset and to create the ANN, the exploration of these tools would let me know those that could be more helpful for my project.

2.3 Report structure

The aim of this report is to offer a general vision about how ANN can be used for human resource personal selection process. A real case example is used with users and job posts information from XING professional network.

The rest of the report is structured as follows. First, we have a look into the history and development of ANN from its creation and we do a comparison between biologic and artificial neural networks with a deeper explanation on ANN.

Following we get to know some of the many applications that ANN has, as well as its advantages and disadvantages. Moreover, a review of current ANN applications to HR management is presented.

Later on, we discuss about the specific tools that are used to deal with big data, the steps of creating an ANN with Neural Networkk Matlab Toolbox. Also, we present each part of the data and we do analyse every component of XING database.

Finally, we do a review of all the papers that were accepted on RecSysy Challenge’16 and we present our recommendation solution by explaining how the data has been organized and the results and conclusions we have obtained.

3. Introduction

New technologies, in collaboration with psychology, are reshaping HR personal selection techniques. Companies have for a long time tried to find the best method to recruit the right employee. In addition to the traditional application procedure (cover letter, resume and interview), HR consultants are increasingly using social networking, online games and data exploration techniques to know more about the candidates. New ways of finding and screening candidates across the network have changed the way recruiters work. [1]

The main objective of a talent-spotter is to find the candidate who meets the requirements of the job and fit the company. Commonly, the job offer appears published in various media such as company website, job boards, newspaper employment section, social networks such as LinkedIn... Of the total number of applications received, those that are more appropriate for the job offer are selected to be on the next steps that are commonly telephone and personal interviews. [1]

Regarding the sincerity of the candidates, Jamie Guillory and Jeffrey Hancock in 2012 compared the reaction from some subjects to printed CV and public profiles on LinkedIn. They observed that in the professional network they lied on the same frequency as in traditional CV, although the way to do it was different: candidates with a digital profile were more sincere in relation to their professional experience and responsibility, but they were not as sincere when it came to inform about their hobbies and interests. The researchers attributed this behaviour to the public nature of LinkedIn. [1]

Computer science have done a lot of advancements using users data. All kind of companies use websites that are striving to provide quality recommendations to their users in order to increase and retain their customers. Some examples would be the friendship social network websites like Google+ or Facebook recommending friends, shopping websites like Flipkart, Alibaba or Amazon recommending products, and not to be outdone, professional networks such as LinkedIn or Xing, recommending jobs. [2]

On the use of professional network, there are two possible perspectives: when the candidate is looking for a job that satisfies its interests or when the hiring company is looking for the right candidate to cover the working position. [3]

4. Neural Networks History

Neural networks, as any other scientific field, has a history of development that starts on the early forties. For that reason, we would like to summarize how the creation and development of NN has occurred, which is the current situation of NN and the line of research that could NN have on the next years.

4.1 Beginning of NN

It is considered that NN had its beginnings in 1943, when the neurobiologist Warren McCulloch, and the statistician Walter Pitss, published the article "A Logical Calculus of the Ideas Immanent in Nervous Activity". This article was the basis and beginning of development in different fields such as Digital Computers, AI and had a lot of influence because it showed that simple neuronal networks could calculate any logic or arithmetic function. [4]

Norbert Wiener and von Neumann, wrote about neural networks and suggested that the search in the design of systems that simulate the human brain could be interesting. It was in 1949, when Hebb published his book “The Organization of Behavior” formulating the Hebbian rule and for the first time a specific learning law for the synapses of neurons was proposed. [5]

4.2 Golden age

First, Minsky in 1951 developed the neurocomputer Snark capable of adjusting weights automatically. Then, in 1956, the pioneers of AI, Minsky, McCarthy, Rochester and Shanon, organized the first conference of AI to discuss NN research. [4]

In the late sixties, Frank Rosenblatt published the largest research work on neural computation ever made. His work consisted in the development of an element called "Perceptron". Then, together with Charles Wightmann and other researchers of MIT, developed the first neurocomputer “Mark I perceptron” capable of recognizing simple numeric. [6]

Rosenblatt, in 1959 described different versions of perceptron and also formulates and verifies the Perceptron Convergence Theorem. Meanwhile, Bernard Widrow and

Marcian E.Hoff at Stanford University developed a new type of NN processing named "Adaline" (Adaptive Linear Neuron). “Adaline” and also a two-layered version called "Madaline", started to employ the least mean square learning rule, those systems were used in different applications such as voice and character recognition, and on commercial field for real time adaptive echo filtering. In 1965 Nils Nilsson published “Learning Machines” book where gave an overview of the NN research of that time. [7]

In 1969, Marvin Minsky and Seymour Papert belonging to the Electronics Research at MIT wrote “An Introduction to Computational Geometry” that was a deep critical work on the perceptron. The conclusion of this work was that perceptron and neuronal computation were not interesting subjects to study and develop and they published a precise mathematical analysis of the perceptron to show that NN were not capable of representing many of the important problems, such as learning one XOR type function. From this moment, the idea of neuronal networks as a methodology capable of solving all kinds of problems was demystified and the investments in the investigation of the neuronal computation descended drastically. [8]

4.3 Quiet years

Due to the lack of funding for research, there were neither congresses nor talks and, there were hardly any publications in NN. However, some leaders such as Flopf, Gose, Fukushima and Grossberg published articles about NN during 1970s. [7]

Also, in 1973 Christoph von der Malsburg used a nonlinear neuron biologically based. The following year, Paul Werbos developed the basic idea of the backpropagation algorithm for his thesis at Harvard University that was rediscovered in 1986 by Rumelhart and has a lot of importance nowadays. Later, Fukushima, Miyake and Ito introduced Neocognitron, a neuronal model capable of recognizing handwritten characters. [9]

Teuvo Kohonen of the University of Helsinki is one of the major influencers of neural computing in the 1970s. His research highlights two contributions: the first is the description and analysis of a large class of adaptive rules, rules in which the weighted connections are modified depending on the anterior and posterior values of the synapses. And the second contribution is the principle of competitive learning in which the elements compete to respond to an input stimulus, and the winner adapts himself to respond with greater effect to the stimulus. [7]

Between 1976 and 1980 Stephen Grossberg presents articles where he analyzes mathematically various neuronal models, he established the Adapted Resonance Theory (ART), which is a network architecture different from those previously invented that simulates other brain abilities as short and long term memory.[9]

4.4 Innovation

In 1982, Hopfield described the recurrent neural network and aroused the interest of many scientific. In 1986, independent groups of researchers, came up with similar ideas known nowadays as back propagation networks that distributes the pattern recognition errors across the entire network. Also, the publication of books about Parallel Distributed Process increased the interests for NN systems. [8]

Moreover on the late eighties, the *INNS journal Neural Networks*, the *Neural Computation* and the *IEEE Transactions on Neural Networks* were founded. [7]

4.5 Today

As AI is a topic of interest nowadays, NN are been studied in order to apply it to a wide range of fields.

In 2006, Hinton stated that the BP limitations could be solved by using learning multilayer NN models with top-down connections to train them and generate sensory data. The research group of Schmidhuber finished the development of RNN and deep feed-forward NN on 2012. Also, in 2014 scientists from IBM introduced the processor TrueNorth that has similar structure that exists in the brain. This processor is able to quickly simulate and execute millions of neurons and synapses work in real time. [7]

4.6 Future

Future of NN, however, lies in the development of hardware because fast and efficient neuronal networks depend on the hardware that is specified for its use, and due to limitations of processors, NN take long time to learn. [8]

5. Biological vs. artificial neural networks

The ability of the human brain to think, remember, learn and solve problems is a great challenge for scientists when it comes to model the operation of the human brain system, and it opens a wide range of possibilities on science and technology.

A brain neuron is the basic unit of the nervous system, the brain is composed of several billion highly interconnected neurons. The neuron receives signals at its entrance, combines them, and sends a response to other neurons, if the intensity of the set of signals received is high enough.

The fundamental parts of a brain neuron are the dendrites (inputs), the axon (output) and the synapse (junction between neurons). The axon of the neuron branches to be able to connect at the same time with different dendrites of other neurons, through the cerebral synapses. [10]

Similarly, in ANN, the fundamental unit is the processor element and it consists of a number of input variables and one or more outputs.

The signals used by brain neurons are electrical and chemical. The signal that is generated and that distributes to the axon is of electrical type. However, it uses chemical type signals to communicate the axon terminal of a neuron to the dendrites of other neurons.

The processor element combines the set of input signals, using a sum usually, to process the result in a transfer function (activation function) which will result in the value of the output variable. The interconnection of the neurons is done with a certain architecture, each connection between neurons receives a certain weigh.

The human brain contains approximately 100 billion neurons. Each neuron is connected to approximately 1000 other neurons, except in the cortex where the neuronal density is much greater. However, using computer science to perfectly emulate biologic system is practically impossible today, but little steps are being taken forward. [9]

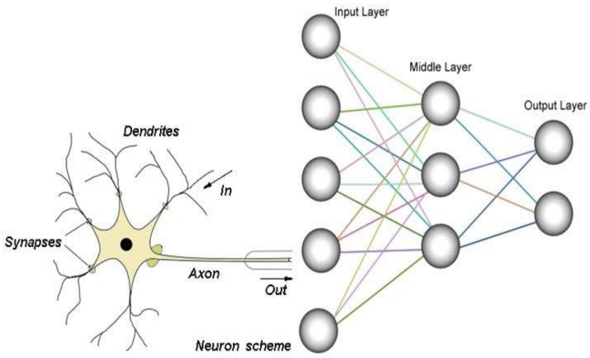


Figure 1. Biological vs. Artificial Neurons

On the table below we can observe the differences between a biological and an artificial neuronal system. [11]

BIOLOGICAL NEURAL NETWORK	ARTIFICIAL NEURAL NETWORK
Are capable of processing information in parallel	
Using past experiences to learn make improvements in their performance	
Both transmit information using electrical signals	
Both can take lot of data (inputs) and provide accurate outputs	
Biological parts of NN: soma, dendrites, synapse, axon	ANN components: input, weight, node, hidden layer, output
Do not need programming	Need programming to set their variables and operation
They learn by adjusting synaptic connections	They learn by adjusting weights
Synaptic strengths modified in response to synaptic activity	Weights altered mathematically in a computer network
Information storage is in the synapses	Information storage is in the weights matrix
Fault tolerance: able to robust performance	Under partial damage, performance degrade.
Information processing is slow (milliseconds)	Information processing is fast (nanoseconds)
Ability of connecting lots of neurons of the order of 10^{11} to 10^{14}	Able to connect a few neurons of the order of 10^{2} to 10^{4}
Data stored in the brain is apparently disordered	Data applied to ANN have to be strictly ordered and prepared
Can tolerate ambiguity and learn from poor and disorganized data	Need to use structured rules and data
Its result comes from an extremely complex process carried out in different parts of the brain	Its results are based on logical functions and algorithms
Energy consumption to execute an operation around 10^{-16} Joules	Energy consumption to execute an operation around 10^{-6} Joules

Table 1. Biological vs. Artificial Neural Network

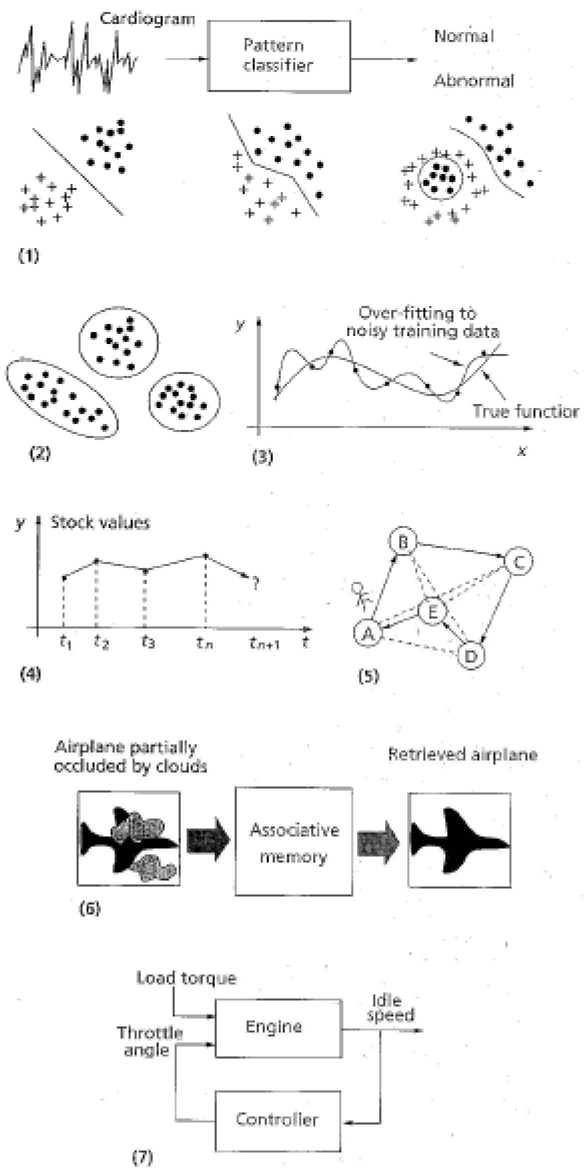


Figure 2. Tasks that Neural Networks can perform

6. Artificial Neural networks

6.1 Introduction

Artificial neural networks are a set of techniques belonging to the field of AI. Its structure consists of a network formed by nodes that represent neurons and connections, that's why they resemble the brains of human beings. ANN are massive parallel systems which are constituted by large number of simple processors with lots of interconnections and are able to solve computational problems. ANN have the abilities to self-organization, self-learning and self-adaptation, also it can conduct large parallel data processing. Today, NN are trained to solve problems that are difficult for conventional computers or humans, they are widely used in solving problems in pattern recognition(1), clustering(2), function approximation(3), forecasting/prediction(4), optimization (5), associative memory(6) and control(7). [10]

6.2 Neural network operation

A neural network is able to detect complex and non-linear relations between variables, using simple units like neurons in parallel. The data is divided into input and output variables related by some type of correlation or dependency. It is also possible that the output is the classification of the variables of entry into different groups. Neurons can be arranged in different layers and most of the NN simple layers consist of an input layer, a layer of neurons or hidden layer, and an output layer.

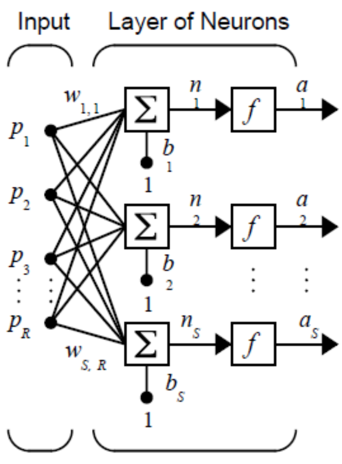


Figure 3. Layer of neurons

The functioning of a neuron consists in the transformation of the inputs through the connections, into an output. The output is obtained from a propagation function, an activation function and a transfer function.

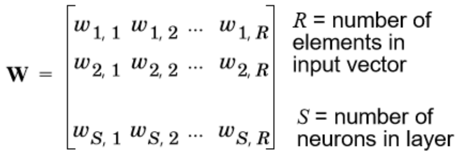


Figure 4. Weight Matrix

- The most common propagation function consists of the sum of all the inputs multiplied by the weights of the connections, plus a bias value.
- The activation function, if it exists, activates or deactivates the output of this neuron.
- The transfer function is applied to the result of the propagation function and usually consists of a bounded output function such as the sigmoid (logsig) [0,1], or the hyperbolic tangent (tansig) [-1,1]. Other functions of transfer can be a linear function (purelin) [-∞, + ∞], radial base (radbas) [0,1] or a discrimination function (hardlim) [0,1].

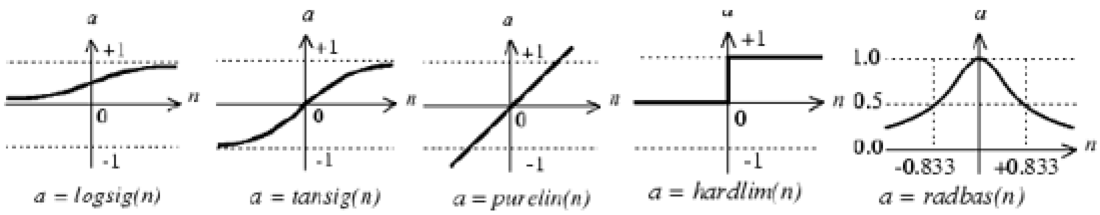


Figure 5. Transfer functions types

In general the sigmoidal logistic function (logsig) will be used when the output variables can only take positive values within a range from 0 to 1 or the hyperbolic tangent function (tansig) when the function is allowed to oscillate between positive and negative values in the range of -1 to 1. [11]

6.3 Types of neural networks

The most important criteria for classifying neural networks are:

Depending on the type of connections:

-Feed-forward networks, where connections go in one direction from the input layer to the output layer. They are static producing only a set of output values, and doesn't need memory because the response to an input does not depend on the previous network state.

- Recurrent networks, in which loops occur because of feedback connections. They are dynamic systems, and the inputs are modified because of the feedback paths, so the network state changes and with the new input pattern the output is computed. [10]

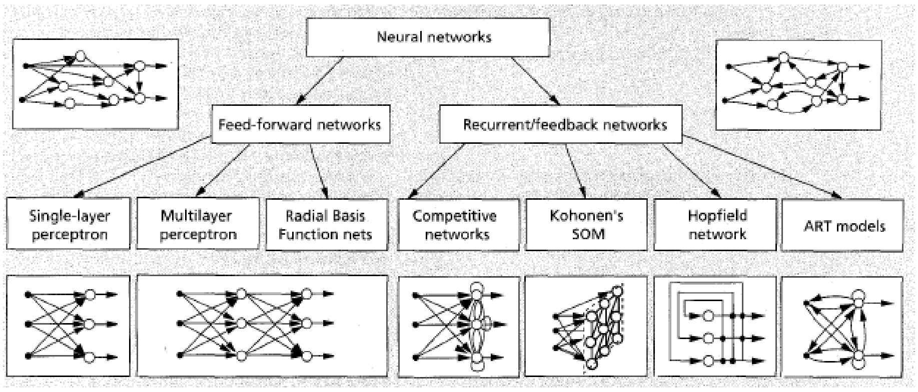


Figure 6. Neural Networks Architectures

6.4 Type of learning

-Supervised learning**:** The data (or inputs) have a response known (or target), with which the neural network adjusts or trains and gives solution (or output). The network uses the mean square error between target and the output.

-Unsupervised or self-organized learning**:** The data consist only on inputs. The network discovers patterns by itself and use it to classify objects and reproduce pattern recognition, it doesn't need to know the correct answer (target).

-Hybrid learning: it combines both supervised and unsupervised learning, some weights are determined by supervised and others by unsupervised learning. [10]

6.5 Training of the neural network

Given a structure and size of the neural network, it comes to train the net. The training or learning, whose objective is that the NN is able to reproduce the underlying behaviour in the data provided, consist basically in the minimization of a cost or error function, which is equivalent that the output of the network, approximates the target in the data. The cost function plus common is the squared error average (MSE). For the optimization of the NN, there are different methods of adjusting the parameters of the network (weights of connections and bias of neurons), from some values either random, or predefined (initialization of the network).

Some examples of the adjustment methods are those of gradient type or algorithms genetics:

-Gradient-type methods calculate error variation by varying each of the parameters (as a multidimensional derivative), and then modify all the parameters of the neural network obtaining a minimum error. It can be said that it is a series search of the solution or global minimum.

-The methods based on genetic algorithms, consist on the generation of a number of nets through mutations in parameters, evaluating the network error for each of them. The nets with less error have more probability of staying in the neural network complex, while nets with more error disappear. It is a parallel search of the solution.

Both methods are iterative methods, which are repeated until meeting the stopping criteria. Some examples of stopping criteria are the number of iterations, obtaining a minimum error, or execution time. In any case, it is generally difficult to ensure that the solution obtained is not a local minimum. [11]

6.6 Over learning or loss of generalization

A possible problem comes when the training process is over learning or experimenting loss of generalization. Given a set of data, it is possible that the neural network reproduces very well the behaviour of such data, but not the new data. This problem is accentuated in case the data has noise or errors. Other types of function approximations such as interpolation with polynomials, can also correctly approximate the data with which it is performed adjustment, while mistakenly approaching new data not used in the adjustment. On figure 7 there is the response of a NN trained to approximate a noisy sine function. The dotted line is the sine function, the noisy measures are shown by '+' symbol and NN response is the solid line, The underlying sine function is shown by the dotted line, the noisy measurements are given by the '+' symbols, and the neural network response is given by the solid line. It has perceptibly over fit the data and will not do a good generalization. [11]

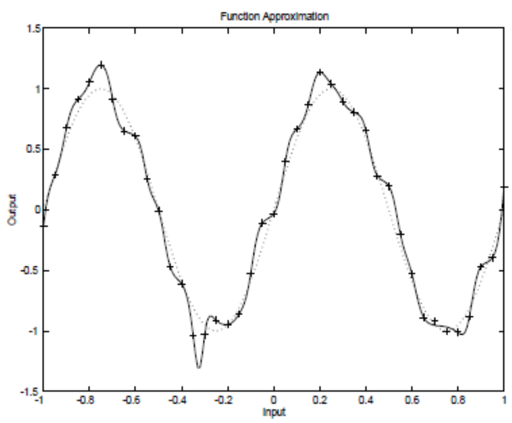


Figure 7. Example of over-learning

There are different ways to avoid over-learning. The first would be to get more data for training, although this is not always possible. Other possibility is to reduce the size of the network (fewer parameters), so that the neural network is less flexible and more robust against noise, although if it is too much reduced, it may not be able to learn or approach the function objective. Providing the NN with enough parameters to be able to learn and avoid over-learning, is the main aspect to take into account in the process of dimensioning a neural network.

6.7 Training, validation and test data

To control whether a neural network has over-learned or not, the data is divided into different groups:

- Training data: is used to adjust the parameters of the NN. They must be representative of the total data, so they are usually randomly selected.
- Validation data: is used after each iteration in the training, to check if over-learning occurs.
- Test data: it is used to check that the ANN is capable of doing a good performance on the prediction of new data.

The division of data can be for example 70% of training data, 15% of validation and a 15% for testing, although the choice of these percentages depends on the number of data available and their distribution.

As random procedures are used during training, it is advisable to perform multiple experiments from "zero" to see if in any of them a smaller error is achieved when evaluating the training data set. [11]

6.8 Techniques to avoid the loss of generalization

6.8.1 Early-stopping

From the use of data division in the groups mentioned, it is possible to apply a technique to avoid over-learning: early-stopping. During the process of iterative optimization of the network parameters, we compare the errors obtained with the training data and with the validation data. If during successive iterations, the error with the training data decreases, as long as the error with the validation data increases, the adjustment process is disrupted, as an additional stopping criterion.

6.8.2 Regularization

Other techniques to avoid over learning are the application of the principle of parsimony, by which, under equal conditions, the simplest explanation is correct. Regularization consists of adding the sum of weights of the network to the cost function, so that for similar error from two different neural networks, the one that has values of the smaller weights is better.

$$MSE_{reg} = \gamma \cdot MSE + (1-\gamma) \cdot MSW$$
$$MSE = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2$$
$$MSW = \frac{1}{n}\sum_{j=1}^{n}(w_j)^2$$

Equation 1. Regularization by MSE

Where γ is the weighting factor, N is the data number, e_i is the error for each data, calculated as the difference between target value t_i and the value calculated by the neural network a_i, n is the number of parameters of the network and w_j are the values of the weights and bias of the neural network. [11]

6.9 Pruned nets

Pruned neural networks are another technique, according to which, by means of the disconnection or elimination of neurons (reduction of the number of parameters), they achieve simpler neural networks, using a cost function that has into account the total number of network parameters. A possible type of cost function to be used in pruned neural networks, which takes into account the number of network parameters, is the PSE (Predicted Squared Error):

$$PSE = MSE \cdot \left(1 + \frac{2N}{N-n}\right)$$

Equation 2. PSE of pruned nets

With pruned neural networks, it is possible to disconnect neurons (simplification of the network) or disconnect inputs (redundant or little relevant variables).

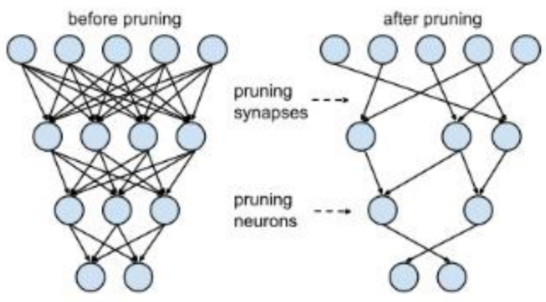


Figure 8. Comparison before and after pruning

6.10 Size of the neural network

The sizing of a feed-forward neural network with a hidden layer for the approximation of a function, consists in the choice of the number of neurons of the hidden layer. Although there is no technique to determine the number of neurons that each specific problem must contain, this can be found from the number of parameters or weights to be estimated, ensuring that such a figure does not exceed the amount of data available for training, because, mathematically, the system would be indeterminate. A direct effect of the use of neurons in excessive amounts is manifested in the inability of generalization of the network by the phenomenon of over fitting, in that case, it will be difficult for the network to provide a correct response to patterns that have not been used in the training and that have the same behavior.

6.11 Standardization of data

The training process of the neural network is performed faster if the inputs and outputs of the network are standardized, so that all of them are expressed in similar ranges. In theory this transformation is not necessary, but as the values of the weights of a network neurons are initialized in a random way with small values, the training works better with standardized inputs and outputs.

It can be standardized by forcing the data to be in a range determined, for example, by scaling all the data of a given variable to the range [-1, 1] (uniformly distributed variable) or in the ideal range [0,1], this is known as rank normalization.

Otherwise, the data can be transformed in such a way that it is centered at 0 with a deviation typical of 1 (as a normal distribution variable).

If any of the variables has another type of distribution, it is convenient to do the linearization of this variable. For example, if an X variable has an exponential distribution, the operation of the neural network will be much better if X is transformed using as variable Y = LN (X), that using X directly. In spite of that, neural networks are able to approximate any type of function, the distribution of input and output variables, if there is enough number of data and the network has sufficient parameters (number of neurons). [11]

7. Applications

Neural networks are applicable to a variety of fields due to their flexibility and ease of use. [11]

7.1 Business

Stock Market Prediction, Sales Forecasting, Target Marketing, Shopping cart analysis, Commercial applications, Service Usage Forecasting, Retail Margins Forecasting

Neural networks can help on business processes, by using historical data and analysing the weights of many factors, they can do predictions on the stock prices and forecast the future sales. Also, with costumers' information the net can do a segmentation regarding the consuming behaviours in order to implement a target marketing.

Moreover, gathering and treating products information with ANN helps understanding customers' preferences on purchasing and that helps on doing sales promotions or distributing the products on the store. In order to do an effective management of a service, ANN could forecast the level of usage in order to plan the number of staff to deal with the workload. Finally, it is useful to forecast the tendency of margins on the future to give a solution on the effects of price changes.

7.2 Banking and Finance

Credit Worthiness, Credit Rang, Price Forecasts, Economic Indicator Forecast, Credit Card Activity Checking, Fraud detection, Real State Appraisal, Bankruptcy Prediction

Neural networks can absorb customers' data and financial indicators to make decisions such as giving approval to applicants for loans if they are good credit risk, assigning credit rates depending on the financial situation of a client or forecasting economic indicators for the future or predict prices of commodities and products. Also, ANN are good at fraud detection and can detect unusual card credit activity and decline client transactions or fraudulent insurance claims. Moreover, with property parameters and other factors ANN can evaluate real state or automobiles. Finally, picking up the right information from a company it can classify it into a potential bankruptcy or not.

7.3 Industrial and operations management

Predict Output Gasses of Industrial Processes, Temperature and force predictions in factories, Quality Control, Process Control, Planning and Management, Machine maintenance analysis, Research and Development, Robots and Control system

Neural networks can be trained to predict output gasses of industrials processes and also temperature and force necessities on the factories. Also, to determine the best operation planning and control activity for a plant depending on demand forecasting, as well as doing the product design and analysis. ANN are useful to predict the quality of materials, make sure all the packages are filled properly or diagnose the machinery defects to do a maintenance analysis. NN have been used together with simulation modelling to learn better manufacturing system designs. Regarding all that, NN are good at operations management especially on scheduling and planning.

7.4 Operational analysis

Scheduling Optimization, Retail Inventories Optimization, Managerial decision making, Cash Flow Forecasting

Neural networks can predict the demand of schedule public transports during special events or hours of the day, also, it can select the best decision option by using classification capabilities. In addition it can maximize the use of resources by doing a better accurate cash flow forecasts. At last, it can predict the demand of a product based on past purchasing activity, as well as, forecasting the optimal level of stock for customer needs, that way it reduce the waste storage.

7.5 Data mining

Prediction, Change and Deviation Detection, Pattern recognition, Classification and Clustering, Time Series Analysis, Knowledge Discovery, Response Modelling

Neural network can use database to predict values of the variables of interests. Also, it can detect data that doesn't match or diverge from the pattern of the rest of the database. Moreover, it can map the data into different classes regarding its attributes and also forecast future values of a time series. Finally, you can build a NN based response or use the NN to find a hidden relationship trend in the dataset.

7.6 Medical

Medical Diagnosis, Treatment, EEG and ECG analysis, Prosthesis Design, Detection and Evaluation of Medical Phenomena, Patient's Length of Stay Forecasts, Optimization of Transplant Times

Neural networks can assist the doctors by analyzing the symptoms and creating a diagnosis and treatment for the patient, they also can analyze and detect the EEG and ECG unconformities, or help in designing a prosthesis by data from movements. Likewise they can detect medical phenomena such as detecting epileptic attacks or breathing abnormalities during anesthesia. Lastly, they can help on the hospital management by forecasting the length of stay for each patient and also by optimizing the times of surgery operations.

7.7 HR Management

Employee Selection and Hiring, Employee Retention, Personnel Profiling, Staff Scheduling

Neural networks can predict the candidates that would be suitable for a job, and those jobs that meets the expectations of a person who is actively seeking work, that's the HR application that we want to give to our artificial neural network. NN can use data from employees to identify those that are likely to stay in the organization for a certain amount of time. Also, depending on the time of the year or a particular time on the day, it is necessary to predict the number of staff necessary to cover the demands on restaurants, stores, etc. Finally, they can successfully identify employees that are suitable for specific tasks to effectively distribute employees.

7.8 Telecomunications

Analyze Customer Data, Image and Data Compression, Customer Payment Processing Systems, Real-Time Translation of Spoken Language, Automated Information Services, Tailoring marketing campains, Optimize routing and quality of service

7.9 Speech

Speech Recognition, Text to Speech Conversion, Speech Compression, Word Classification, Natural language processing

7.10 Insurance

Segmentation of policy holders, Product Optimization, Detection of Fraudulent Claims, Policy Application Evaluation, Manage the Offering to Customers

7.11 Energy

Predict Gas or Coal Index Prices, Electrical Load Forecasting, Energy Demand Forecasting, Exploration, Short and Long Term Load Estimation, Power Control Systems

7.12 Defense and security

Facial Recognition, Weapon Steering, New Kinds of sensors and radars, Recognition and follow-up at target shooting, Stock Trading Advisory Systems, Signal and Image Identification, Object Discrimination, Intelligent weapons

7.13 Transportation, Aerospace and Automotive

Vehicle Scheduling, Truck Brake Diagnosis System, Autopilot Enhancements, Flight Path Simulation, Aircraft component fault detection, Automobile automatic guidance System

8. Advantages of Neural Networks

Artificial neural networks offer many advantages because they are based on the structure of the brain. [9]

Adaptive learning: ability to learn to perform tasks based on training, this consists in providing an input data to the NN as well as the expected output data.

Self-organization: a neural network through a learning process can create its own organization or representation of the information that receives as input and target.

Flexibility: NN can handle non-important changes in input information, such as noise signals or other changes on the information.

Dynamic and self-adaptive systems: NN are able to change constantly in order to adapt to new conditions, also, they are adaptable because of the self-tuning ability of the neurons that make up the system.

Fault tolerance: NN have their information distributed in the connections between neurons, with some degree of redundancy in this type of storage, therefore, the networks can continue to perform their function (with some degradation) even if part of the network is destroyed. Otherwise, algorithmic computers and data retrieval systems store each piece of information in a single, localized and addressable space, that in front of defects it stops performing.

Real-time operation: neural computations are able to perform in parallel, for that reason, machines with special hardware must be designed and manufactured to obtain this capacity.

Easy insertion into existing technology: in order to facilitate the modular integration of NN in existing systems, specialized chips can be created to improve their capacity in certain tasks.

Speed of adjustment and simulation: NN can afford big amounts of data doing an effective adjustment of the output data, moreover, there is independence between the complexity of the problem and the dimensioning of the network.

9. Disadvantages of Neural Networks

The disadvantages of NN are based on the fact that actual machines cannot deal with the necessary specifications needed for a NN. [9]

Processing parallel information: Most neural networks suffer from our lack of hardware. The ability of neural networks lies in their ability to process information in parallel so it can process multiple pieces of data simultaneously. Unfortunately, common machines work in series and only execute one instruction at a time. For this reason, modeling parallel processes in serial machines can be a time-consuming process.

Lack of definitional rules: there are lots of factors that must be taken into account when we build a neural network for a specific problem: architecture, number of neurons per layer, number of layers, learning algorithm, representation of the data…Also, the random initialization of weights, and the random division of data into training, validation and testing, can give rise to different solutions (local minima).

Black box behavior: while in statistics models you can see the variables that algorithms select and use to predict or classify and also see the weights and the final equation, in NN it is not possible to obtain all this information.

Statistical models of probability vs. Neural Network: depending on the problem you might use one or other, both are complex models and requires solid knowledge to achieve satisfactory and valid results.

Over-learning or loss of generalization: sometimes NN obtain a perfect approximation of the output but it cannot get a good approach with new data. So it is important that the net doesn’t memorize the input data but can find the generalized weights to apply it to new data.

10. State-of-the-Art in Human Resources

ANN have been applied to HR management in the last years, we will do a review of the related research in order to know about some applications ANN have been modelled to be used in HRM.

In 2006, Acevedo, Caicedo and Loaiza studied the personnel selection problem by using computing intelligence to classify psychology patterns from the 16-PF test. The 16 factors that measures this personality test are: warmth, reasoning, emotional stability, dominance, liveliness, rule-consciousness, social boldness, sensitivity, vigilance, abstractedness, privateness, apprehension, openness to change, self-reliance, perfectionism and tension. Also, there is a last factor that detects those subjects who lie to offer better self-image or those who answer randomly.

To classify the psychological patterns, they used two different types of ANN, both of supervised training and multilayer structure: Multilayer Perceptron and Radial Basis Function. The database was composed by 552 students with its personality tests results, 200 profiles were used on training phase, 50 profiles in validation. So the inputs were the 17 factors from the tests and the output was to determine the performance of a student to pass the tests to enter the university.

The results showed that the network was accurate in 70% of predictions, as long as the NN is trained with more data, the accuracy can increase. [13]

In 2010, Lin Li and Hui Zhu used BP Neural Network to build the HR management outsourcing decision-making model with Matlab NN Toolbox.

The first step was to prepare the data, they chose as input values in terms of thousand dollars: Finding cost, Negotiation cost, Contract expenses, Managements costs and Uncertain cost. While the output indicators also in terms of thousand dollars were: HR cost-saving benefits and Outsourcing to improve the efficiency of the enterprise benefits.

Then they used 10 sample data to train the model and they got relative errors of 1,26% and 5,06% for the respective outputs. The model could perform better with large number of historical data. The results showed that this model was effective and it could help companies to do more scientific decisions on HR management. [14]

In 2010, Chang Ning studied the allocation of enterprise human resources with NN, he designed the Elman recurrent NN to predict the probability of HR transferences.

An effective personnel allocation should be by recognizing employer's abilities and the necessities of a specific work position, and then placing the employees in the most opportune position. Optimizing this matching would mean to maximize enterprise economic and social benefits.

He designed Elman NN as a feedback network model and BP learning algorithm to do the supervised training. The relevant indicators to use as input values were: the annual output value of the enterprise, the total number of the employees in the enterprise and the proportion of technical personnel in the enterprise. And with 10 sets of an enterprise relevant data from 1999 to 2008 as training data and 2 sets for testing data, the ANN had to give as output the personnel transference probability of a particular technical post in the enterprise.

According to the results, this NN is able to objectively forecast the probability of enterprise personnel transference accurately and effectively. [15]

In 2010, Wang and Jiang proposed an evaluation model for high tech enterprise human resources based on BP neural network model.

They chose 14 indicators to evaluate HR: ability to communicate/learn/promote project/development and innovation/share technical ideas/definition and conversion of requirement/solve problem, motivation, qualification and work experience, knowledge of industry, personality traits and conduct characteristics and values.

They trained and tested the NN for personnel evaluation and the results showed that the net was improved to be very effective to evaluate HRM. This model can also be used in personnel decisions on an enterprise such as recruitment and promotion. [16]

In 2012, Gao, Fan and Deng evaluated the core competence of an enterprise by using ANN, specifically BP neural network, as well as using analytic hierarchy process AHP to determine the output of the training sample. AHP determines the relative importance regarding different factors or weights in a system

According to Resource-Theory and Capability Theory, they describe the source of core competence with: Organization Ability, Human Resources, Intellectual Property, Enterprise Culture, Plan Ability, Market Capacity and Production Capacity.

First they used data from eight companies as the sample population and then selected two of them to use as evaluation samples, then standardize the data with min-max normalization. Each company had the evaluation index value for each of the seven core competence to train the NN. Then, the value of the output data consisted on the size of core competitiveness in 3 dimensions (value, sustainability and inimitability) using the thinking of AHP to rank the core competence of sample enterprises.

ANN had the function of automatically establish connection relations between inputs and outputs. Once they got the results, they could sort the size of core competence in each dimension and compare the sample companies. Comparing the evaluation samples they could use the results to do proposals such as that one company should strengthen the investment in the technology development or in HRM. The same model could be applied specifically to HR core competences. [17]

In 2013, two researchers from The Federal University of Technology in Nigeria worked on a neural network system that collects data from applicants through web-based interface and matches it with appropriate jobs.

The system is established on Internet Information server that uses ASP and Microsoft Access, as well as HTML language to authorize web pages.

Feed forward neural network of a web-based HRM system use data that have information from the applicant (personal data, academic and professional qualification, job history), information from the job (applicants' registration, job and organization requirements and job vacancy). So the inference engine uses cognitive filter and emotional filter to provide reasoning about these data. That way there is a match between applicants and jobs and recommendations are given to HR department. [18]

In 2013, three researchers from Rumania presented a method to study working conditions, in that case for workstations of manufacture industry, with a feed forward NN with a BP training algorithm created with Matlab NN Toolbox.

They used 6 input factors (temperature, noise, humidity, luminosity, load and frequency) and 3 output parameters ranking the workspace as good, medium or poor. In the training, 12 patterns were used, and 4 more patterns were used fort testing the network.

The results wanted to be used to identify critical parameters that characterize working conditions, and from these parameters train another NN to identify the best working conditions for a specific workstation. [19]

In 2016, four researchers from the University of Information Science in Cuba proposed the use of Multilayer Perceptron type of ANN to evaluate the HR competences using functions with the language PL/R.

The data was obtained from the BCEC database that includes 157 persons that were evaluated by experts into 8 different competences, and 75% was used to train and the 25% to validate and test the network. They designed different ANN changing the number of neurons and the activation functions, in order to compare them and find the net with better performance.

The results showed that the NN allowed a higher assessment of labour competences compared to traditional methods. Moreover, this model could also be used to recruitment processes to find those candidates that have the right competences to play a role in a project group. [20]

11. Programs to analyse, process and prepare the data

In order to analyse, make changes or organize the dataset it was necessary to use specific programs. Moreover, it was necessary to choose a program in order to create the ANN with the objectives of the project.

11.1 Excel and PowerPivot

Excel is a software developed and distributed by Microsoft that allows you to create and manipulate data tables, graphs, databases and it can automate much of your work. Nevertheless, Excel has worksheet and workbook limitations and the one that affected us was that it is only possible to process 1.048.576 rows in the same worksheet. As we had larger data it was not possible to treat it all in once and that made us think of using another program to analyse the data. However, we found that there is an Excel add-in named Power Pivot which you can use to perform big data analysis as you upload large data from different sources and analyse it rapidly by using Pivot Tables and Pivot Charts. So we tried to upload all the data and this tool had no problems to process it. In order to treat big data or in some fields such as Business Intelligence it is good to know that Power Pivot can support 2GB size files and let you work with 4GB of data in memory.

11.2 Access

Access is a database management system (DBMS) developed and distributed by Microsoft that is able to import or link data stored on other applications databases or documents. Microsoft Access is a Microsoft tool for defining and manipulating databases. Database systems are designed to handle large amounts of information, defining the structures for the storage of information and the mechanisms for managing it. Once we saw that Excel couldn't be able to manage our data, we explored Access tool and we saw that by using queries we were able to create tables with linked data between different parts of our database and that was really helpful in order to choose which data and with which format we should prepare in order to use as inputs and targets of our ANN.

12. Programs to create an Artificial Neural Network

ANN can be implemented with different program languages and tools: C++,Java, Matlab, Python, R, Perl, Excel. After having a look at them, we did some attempts with Python and R but we finally decide to use Neural Network Toolbox of Matlab because it did not require to spend lot of time programming.

12.1 Matlab Toolbox

Matlab is a programming language and a development environment, which is based on matrix computations. It integrates an editor, an interpreter and some visualization tools. It has also many toolboxes that are to be used in particular areas of science. For the development of NN applications, it has the Neural Network toolbox.

First of all, it is important to understand how NN Matlab Toolbox works in creating an Artificial Neural Network. [11] As we see on the *Figure 9*, given an input and a target data, the NN create different connections between neurons giving distinct weights to each connection, initially it makes random guesses. Then it performs an output regarding the connections created and this output is compared with the target data, then it makes iterations to adjust the node-connection weights till the minimum squared error does not decrease anymore. So the kind of learning it does it's a supervised learning. [11]

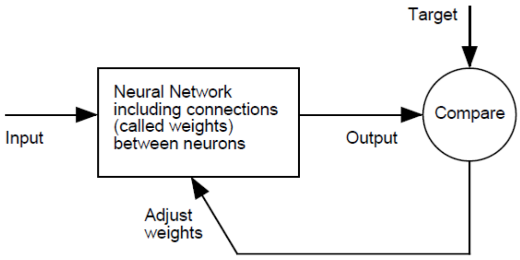


Figure 9. ANN Operation Process

NN Matlab Toolbox has algorithms, apps, functions to create, train, simulate and visualize neural networks. NN have a good performance on fitting functions, moreover it is proved that simple NN can fit any function. So, by using NN Fitting Tool there is the possibility to select your own data as matrix, so it is possible to upload the inputs that you want to present to the network and also the targets that would coincide with the desired network output. Then according to the organization of the data you can choose if the samples are matrix columns or matrix rows. Moreover, by using the fitting app of Matlab, the normalization of the data is done automatically.

12.1.1 Data division

After that, we have to determine the Training, Validation and Test data. For example, you can divide the dataset 75% of data to train, 15% to validate and the other 15% to test de data and the dataset would be randomly divide. The training data will be presented to the network during the training and the network will adjust according to its error. The validation data is used to measure network generalization and to halt training when generalization stops improving. Finally, the testing data has no effect on training because it provides an independent measure of network performance during and after the training.

12.1.2 Network Architecture

The standard network used on fitting is composed of two layer feed forward network and in the hidden layer has a sigmoid transfer function and in the output layer uses a linear transfer function. You can choose the number of hidden neurons that makes a better training performance, the number of neurons on the output layer is always going to be 1.

12.1.3 Mean Squared Error and Regression values

MSE shows the average squared difference between the targets given to the NN and the outputs that give the NN as a result after training, validating and testing. Low values of MSE means less error and that is better. Also, regression R Values of 1 shows that there is a high correlation between outputs and targets while values of 0 means random relation.

12.1.4 Training Algorithms

When it comes to train the data, it is necessary to choose the right algorithm. On Matlab Fitting Toolbox there are three different algorithms:

- Levenberg-Marquard: recommended algorithm for most problems that needs more memory because it has to storage matrix that can be quite large, but less time for networks of moderate size. When the generalization stops improving the training stops automatically.
- Bayesian Regularization: this algorithm may need more time, but it performs a good generalization when the dataset is small, noisy or difficult. In that case, training stops according to adaptive weight minimisation. Actually, it is a

modification of Levenberg-Marquard algorithm, it reduces the difficulty of determining the optimum network architecture.

- Scaled Conjugate Gradient: useful because it uses gradient calculations which are more efficient in terms of memory than the other 2 algorithms that use Jacobian calculations. When generalization cease improving it stops automatically. [11]

On Levenberg-Marquard and Scaled Conjugate Gradient, generalization stops improving when there is an increase in the MSE of the validation samples. Matlab uses the validation checks in order to detect when the generalization stops improving, for example, if the number of validation checks is 3 the training continues until the validation error fail to decrease for 3 consecutive iterations.

12.1.5 Regression plots

Regression plots are used to validate the performance of the ANN, comparing the targets with the outputs during training, validation and test phases. Ideally, the regression line should be a 45 degree line, which would mean that the network has done a perfect fit because the outputs are equal to the targets.

12.1.6 ANN Performance

There is the possibility to retrain the network if the results are not good enough, because the network uses different initial weights and biases each time and the performance could be improved. Also, it is possible to do other variations such as testing new data, increasing the number of neurons or upload a larger training data set. Sometimes the performance on test results are worse than on the training, this would be an indicator of over fitting and may be convenient to reduce the number of neurons.

12.1.7 Error histogram

The error histogram is also useful to validate the performance of the NN, it is a bar chart that shows the error (targets-outputs) of training, validation and test data. On that chart it is possible to detect outlier data that might be more difficult to see on the regression plot, and this outliers represent data that hasn’t fit as well as the rest of the data.

13. Database

13.1 Introduction

XING is a business-oriented social network site launched on November 2003, just half a year after LinkedIn, and its aim is to help people discover career-opportunities and recruiters to find the right candidate. Xing has about 18 million users worldwide, mostly from German-Speaking countries as 12 million users are from Germany, Switzerland and Austria.

The ACM RecSys Challenge each year give the opportunity to engineers and researchers all over the world to do team-working on real-world recommender systems problems. RecSys Challenge'16 consisted on designing and implementing recommender system algorithms for job recommendations in order to predict those users that may be interested in receiving a job posting as a push recommendations and at the same time meets the characteristics of an appropriate candidate for the offered job. So XING's aim was to create recommendations that satisfy the interests of both the job seekers that have some preference for their next professional experience and the recruiters that want to hire the best candidate for the given job. [21]

A large dataset was provided by XING especially for the RecSys Challenge that was held in Boston in September 2016. The data given has about 1.5 million distinct user profiles (job seekers), 1.3 million distinct items (job postings) and 8.2 million interactions between users and items.

XING data is not complete and has been enriched with noise in order to preserve the anonymity of the users and protect the privacy of both the users and the business. First of all, the data is composed only of some fractions of XING users and job postings and it also contains artificial users. IDs are used as a pseudonym for each user and some attributes have been removed from the data. Also, all properties were changed to numerical values, some interactions of the users were removed and some of them are artificial. [22]

There are some challenges that arise from the nature of XING data and the goals that need to be achieved. Data presents an extreme sparseness because on the 43% of the users and on the 24% of the items it is not included any interaction at all. Moreover, there is the purpose of balancing user interest and recruiter demand. Also, a smart targeting must be done to estimate the likelihood of a user showing interest in a job recommendation using its interactions with push recommendation notification. [22]

Moreover, it is also important to balance relevance and revenue because from one hand, premium users pay a subscription to receive benefits such as increase their profile visibility, and from the other hand, some companies pay for the content they post. So premium users should have advantages over the basic users as well as the content of payment should have preference compared to non-paid content. [21]

It is important to deeply understand the data, in order to predict postings that a user will interact with. So, we fist explore each dataset and we do analyse every detail in it to see how we can make a system that help us doing the appropriate job posting recommendation for a user. Once we decide which part of the data we want to use for experimenting and training our artificial neural network it is possible to study how to optimize the ANN results in its prediction.

13.2 Description

The dataset contains three key components: user's profile, job postings information and interactions between users and job offers. [22] Each one has been studied with Excel Power Pivot that can deal with large amounts of data.

13.2.1 Interactions

Details about the users that performed some kind of interaction with particular items (job postings) and the order of all the interactions performed during 12 weeks.

FIELD	DESCRIPTION	COMMENTS
user_id	Anonymised ID of the user who performed the interaction.	The same user can appear several times.
item_id	Anonymised ID of the item on which the interaction was performed.	The same item can appear several times.
interaction_type	Type of action that performs a user.	**0**=*Push recommendation*; **1**=*Click*; **2**=*Bookmark*; **3**=*Reply*; **4**=*Delete;* **5**=*Recruiter Interest*
created_at	Timestamp representing the time when the interaction got created.	Timestamps are shifted but they maintain the order of the interactions.

Table 2. Interactions data description

The following concepts have been defined as:

- *Click:* when a user click once or more on a push recommendation.
- *Bookmark:* when a user bookmarks the item.
- *Reply:* when a user clicks on the reply/application form button and applies for a job.
- *Delete:* when a user delete the recommendation from its list of recommendations, so it won't be longer shown (a new recommendation will be loaded and displayed to the user).
- *Recruiter Interest:* when a recruiter clicks on a user profile for the given job item.

From the total of 322.776.002 rows of interaction data we found out that Recruiter interest type of interaction had lots of rows repeated (it had to be the same interaction because it happened at the same time), so we decided to delete the 98.417 repeated rows from recruiter interest type. That's the reason why there is so less data on that interaction type compared to the other types of interaction. Moreover, we decided not to use created_at data because we wanted to focus on the probability that an interaction occurs no matter the time it happens.

The impressions are the push recommendations that Xing recommender system made to users and the rest of interactions are showed on the next table.

Interaction Type	Number of interactions
Impression	314.501.101
Click	6.867.579
Bookmark	281.672
Reply	117.843
Delete	906.836
Recruiter Interest	2.554
Total	322.677.585

Table 3. Counting of interaction types

13.2.2 Users

Information about all the users that appear in the dataset obtained from Xing users profiles, this attributes would be a summary of the CV they have created on their Xing profile.

FIELD	DESCRIPTION	COMMENTS
user_id	Anonymised ID of the user.	
jobroles	Job role terms extracted from the user's current job title.	**0**=*Unknown*; Comma-separated list of job roles represented by 11.628 different words with an average of two words per user.
career_level	Career level of the user.	**0**=*Unknown*; **1**=*Student/Intern*; **2**=*Entry level/Beginner*; **3**=*Professional/Experienced*; **4**=*Manager/Supervisor*; **5**=*Executive (VP, SVP…)*; **6**=*Senior Executive (CEO, CFO, President)*
discipline_id	IDs that represent different disciplines.	**0**=*Unknown*; There are 23 different disciplines that can refer to *Consulting*, *HR*, *Marketing*, *Finance, etc.*
industry_id	IDs that represent industries.	**0**=*Unknown*; There are 23 different industries that can refer to *Internet, Automotive, Banking, Construction, etc.*
country	Country in which the user is currently working. Most of the users are from Germany.	**de**=*Germany*; **at**=*Austria*; **ch**=*Switzerland*; **non_dach**= *Other countries*

region	Specification of the region for those users that have *Germany* as country.	**0**=*Unknown*; **1**=*Baden-Württembert*; **2**=*Bavaria*; **3**=*Berlin*; **4**=*Brandenburg*; **5**=*Bremen*; **6**=*Hamburg*; **7=***Hesse*; **8**=*Mecklenburg-Vorpommern*; **9**=*Lower Saxony*; **10**=*North Rhine-Westphalia*; **11**=*Rhineland-Palatinate*; **12**=*Saarland*; **13**=*Saxony*; **14**=*Saxony-Anhalt*; **15**=*Schleswig-Holstein*; **16**=*Thuringia*
exp_n_entries_class	Number of CV entries that the user has listed as work experience.	**0**=*No entries*; **1**=*1-2 entries*; **2**=*3-4 entries*; **3**=*5 or more entries*;
experience_years	Estimated number of years of work experience that the user has.	**0**=*Unknown*; **1**= *less than 1 year*; **2**=*1-3 years*; **3**=*3-5 years*; **4**=*5-10 years*; **5**=*10-15 years*; **6**=*16-20 years*; **7**=*more than 20 years*
exp_years_in_ current	Estimated number of years that the user has already worked in the current job.	**0**=*Unknown*; **1**= *less than 1 year*; **2**=*1-3 years*; **3**=*3-5 years*; **4**=*5-10 years*; **5**=*10-15 years*; **6**=*16-20 years*; **7**=*more than 20 years*
edu_degree	Estimated university degree of the user.	**0**=*Unknown*; **1**=*bachelor*; **2**=*master*; **3**=*phd*
wtcj	User's willingness to change job.	**0**=*Low interest of changing her job soon*; **1**=*High interest in changing current position*
premium	Users that are subscribed to XING's paid premium membership	**0**=*No subscription*; **1**=*Active subscription*

Table 4. Users data description

13.2.3 Items

Information about all the job posts that appear in the dataset obtained from Xing website, this attributes describe the job position, the required career level as well as skills and the geographical location.

FIELD	DESCRIPTION	COMMENTS
item_id	Anonymised ID of the item.	
title	Concepts that have been extracted from the job title of the job posting	**0**=*Unknown*; Comma-separated list of concepts represented by 13.700 different words with an average of 6 words per job post.
career_level	Career level ID that requires the job position.	**0**=*Unknown*; **1**=*Student/Intern*; **2**=*Entry level/Beginner*; **3**=*Professional/Experienced*; **4**=*Manager/Supervisor*; **5**=*Executive (VP, SVP…)*; **6**=*Senior Executive (CEO, CFO, President)*
discipline_id	Anonymised IDs that represent disciplines.	**0**=*Unknown*; There are 23 different disciplines that can refer to *Consulting*, *HR*, *Marketing*, *Finance, etc.*
industry_id	Anonymised IDs that represent industries.	**0**=*Unknown*; There are 23 different industries that can refer to *Internet, Automotive, Banking, Construction, etc.*
country	Country in which the job is offered. Most of the items are job posts for Germany.	**de**=*Germany*; **at**=*Austria*; **ch**=*Switzerland*; **non_dach**= *Other countries*
region	Specification of the region for those job posts that have *Germany* as country.	**0**=*Unknown*; **1**=*Baden-Württembert*; **2**=*Bavaria*; **3**=*Berlin*; **4**=*Brandenburg*; **5**=*Bremen*; **6**=*Hamburg*; **7**=*Hesse*; **8**=*Mecklenburg-Vorpommern*;

		9=*Lower Saxony*; **10**=*North Rhine-Westphalia*; **11**=*Rhineland-Palatinate*; **12**=*Saarland*; **13**=*Saxony*; **14**=*Saxony-Anhalt*; **15**=*Schleswig-Holstein*; **16**=*Thuringia*
is_paid	Indicates if the posting is paid or not by the company.	**0**=*Non-paid*; **1**=*Paid*
latitude	Latitude information rounded to10km.	**0**= *Unknown*; Others: number of kilometres
longitude	Longitude information rounded to 10 km.	**0**= *Unknown*; Others: number of kilometres
employment	Type of employment.	**0**=*Unknown*; **1**=*full-time*; **2**=*part-time*; **3**=*freelancer*; **4**=*intern*; **5**=*voluntary*
tags	Concepts that have been extracted from tags, skills or company name.	**0**= *Unknown;* Comma-separated list of concepts represented by 63.933 different words with an average of 8 tag words per job post.
created_at	Timestamp representing the time when the interaction got created.	Timestamps are shifted but it maintains the order of the interactions.

Table 5. Items data description

14. Data analysis of users and items

As each job offer requires its particular knowledge and skills it is necessary to match each information that we have on both users an items datasets. The best fit between a job offer and the proper candidate may depend on many details, those that appear in user data that corresponds to item data are shown on the table below.

USERS	ITEMS
jobroles	title, tags
career_level	career_level
discipline_id	discipline_id
industry_id	industry_id
country	country
region	region

Table 6. Same information for Users and Items

Regarding to the relation between jobroles and titles or tags it would be useful to create one field to compare jobroles from a user with title from an item and another list to compare it with tags, using number 1 to indicate that there is one or more coincidences and number 0 if there is no coincidence.

After analysing all the dataset we saw that there is some information that we won't be using on our ANN to do recommendations. From the items dataset we would refuse to use the geographical coordinate's latitude and longitude. Also as we said with interactions data, created_at would not be used as it doesn't give any further information about a possible relation between a user and an item.

The dataset includes a total of 1.497.020 users and 1.306.054 items. We are going to do an analysis of all the fields from the original dataset because we believe that the design of any good system is principally based on how deeply the developers understand the system and the data.

As we will see on the graphics below, generally item data is more complete than user data, maybe if there were not as much missing data from users, we would have more complete samples for our ANN and that would lead to a better performance.

14.1 Career level

Mostly half of the users from which we have information belong to Professional/Experienced level, also most job posts requires this level of career.

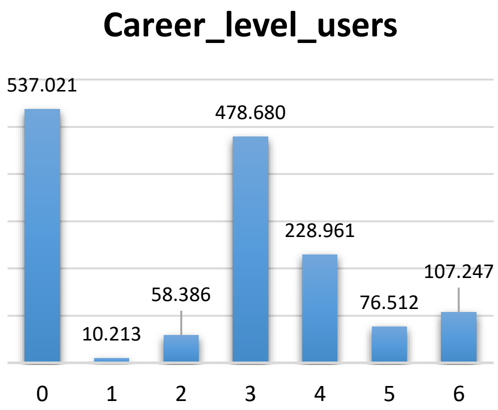


Figure 11. Bar Chart Career_level_users

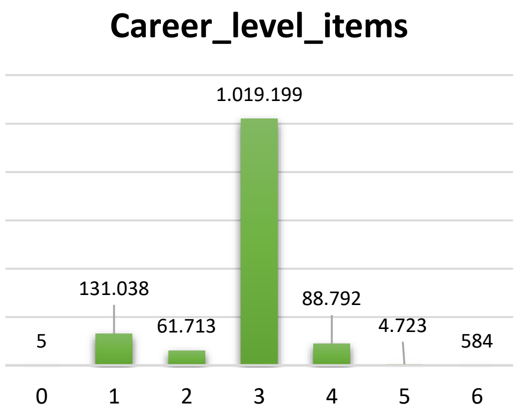


Figure 10. Bar Chart Career_level_items

Moreover, there is a 35,87% missing data from career level of users while in almost all the job offers the career level required is shown.

14.2 Country and region

As we already knew, a high percentage of users (82,55%) and job posts (84,05%) belong to Germany country.

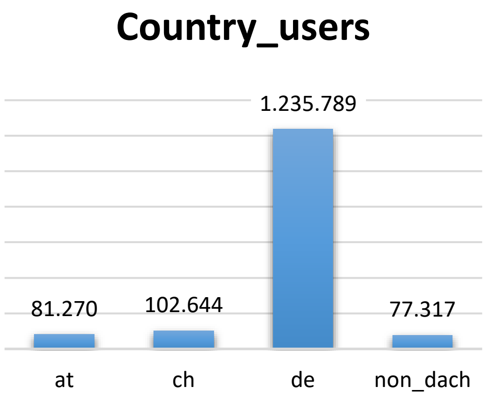


Figure 12. Bar Chart Country_users

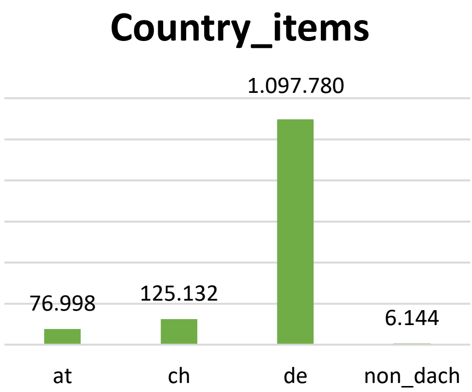


Figure 13. Bar Chart Country_items

As we can see on the pie charts, there are some regions with a higher percentage of users than others and also, more job offers in specific regions. Also, there is a lot of missing data on this field, almost half of the users don't specify their region and in 20,54% of job posts the region is neither pointed it out.

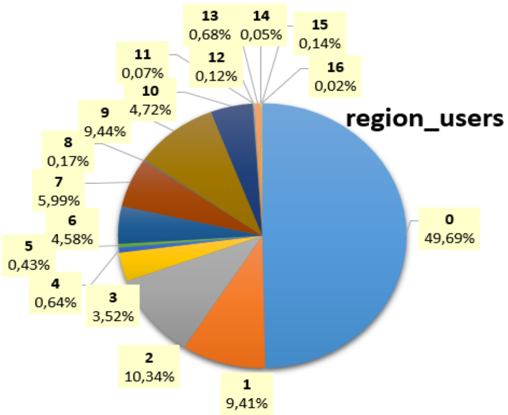


Figure 14. Pie Chart Region_users

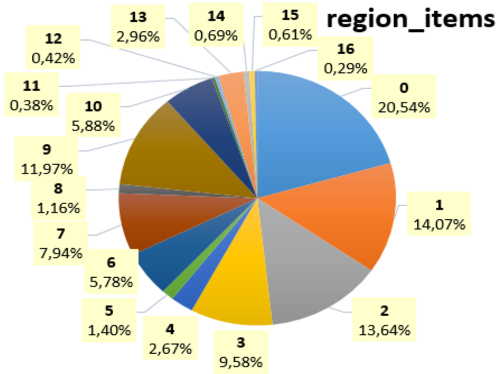


Figure 15. Pie Chart Region_items

14.3 Discipline

On the table below we can see an overview of the discipline field, another time we see the amount of data missing on users data, also, it seems that discipline number 9 does not appear on items data.

Discipline	Number of users	Percentage of users	Number of items	Percentage of items
0	1.115.980	74,55%	5	0,00%
1	11.850	0,79%	66.480	5,09%
2	4.819	0,32%	963	0,07%
3	27.950	1,87%	95.205	7,29%
4	50.388	3,37%	35.483	2,72%
5	61.599	4,11%	126.616	9,69%
6	22.102	1,48%	24.860	1,90%
7	5.333	0,36%	17.287	1,32%
8	7.555	0,50%	131.431	10,06%
9	4.000	0,27%	NA	NA
10	5.013	0,33%	90.855	6,96%
11	12.999	0,87%	84.077	6,44%
12	422	0,03%	544	0,04%
13	12.368	0,83%	101.160	7,75%
14	6.121	0,41%	4.262	0,33%
15	5.686	0,38%	196.558	15,05%
16	26.183	1,75%	60.468	4,63%
17	59.672	3,99%	128.008	9,80%
18	15.070	1,01%	41.820	3,20%
19	1.744	0,12%	16.028	1,23%
20	3.447	0,23%	9.615	0,74%
21	17.016	1,14%	27.896	2,14%
22	5.576	0,37%	18.315	1,40%
23	14.127	0,94%	28.118	2,15%
Total	**1.497.020**	100,00%	**1.306.054**	100,00%

Table 7. Comparison of users and items discipline

14.4 Industry

On the table below we can see a general vision of the industry field, there is no data missing on items but almost a third of user data is missing.

Industry	Numbrer of users	Percentage of users	Number of items	Percentage of items
0	486.175	32,48%	0	0,00%
1	17.276	1,15%	42.033	3,22%
2	28.451	1,90%	32.662	2,50%
3	83.591	5,58%	90.519	6,93%
4	30.225	2,02%	101.550	7,78%
5	29.927	2,00%	31.759	2,43%
6	61.371	4,10%	29.502	2,26%
7	117.766	7,87%	165.991	12,71%
8	49.003	3,27%	13.426	1,03%
9	14.956	1,00%	185.366	14,19%
10	14.930	1,00%	3.652	0,28%
11	12.392	0,83%	1.466	0,11%
12	19.585	1,31%	7.786	0,60%
13	17.865	1,19%	5.904	0,45%
14	28.825	1,93%	10.751	0,82%
15	68.461	4,57%	16.044	1,23%
16	156.798	10,47%	366.722	28,08%
17	55.906	3,73%	26.231	2,01%
18	22.284	1,49%	32.993	2,53%
19	19.353	1,29%	14.483	1,11%
20	112.508	7,52%	85.888	6,58%
21	22.057	1,47%	6.656	0,51%
22	13.553	0,91%	4.077	0,31%
23	13.762	0,92%	30.593	2,34%
Total	**1.497.020**	100,00%	**1.306.054**	100,00%

Table 8. Comparison of users and items industry

14.5 Users analysis of data

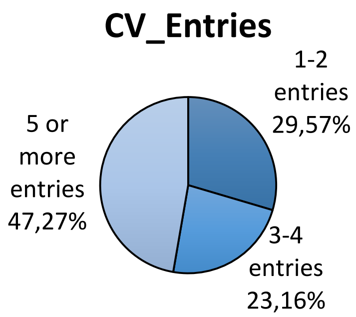


Figure 17. CV_Entries on user's profile

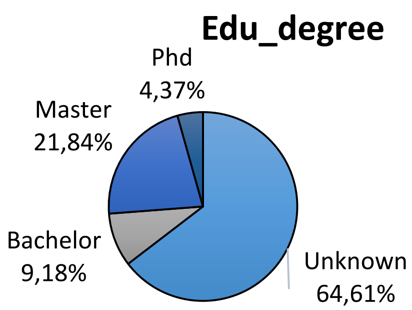


Figure 16. Edu_degree of users

Every user has CV entries on their profile but lot of them doesn't specify their university degree.

Most of the users from which we have data, have worked for 5 or more years, and have been in their current job 10 for less years.

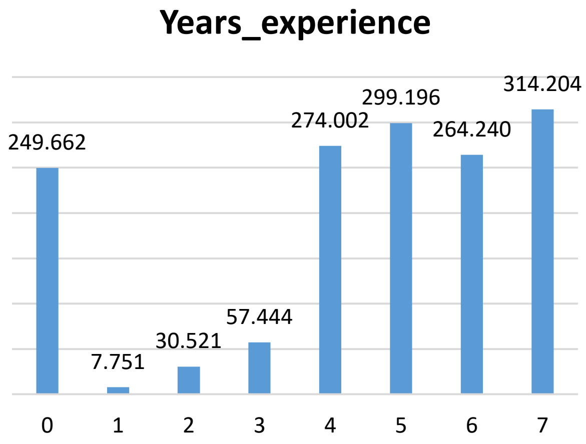


Figure 19. User's years of experience

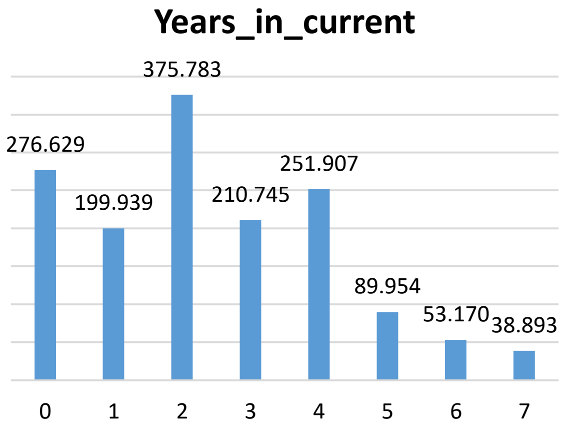


Figure 18. User's years in current job

From all the users, the 76,84% would like to change their current job but only 19,21% pay subscription to be premium. And again, the 60,02% of users does not specify their field of studies.

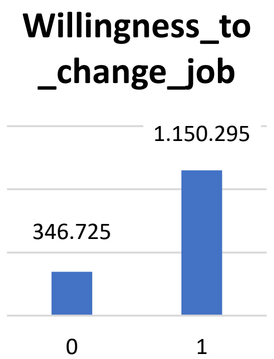


Figure 21. User's willingness to change job

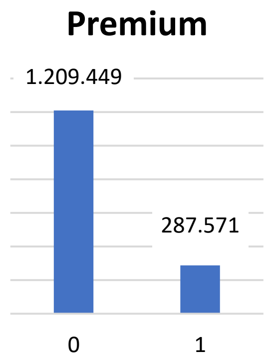


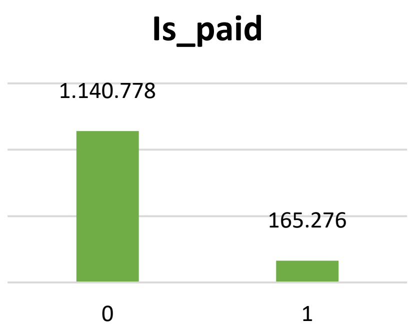
Figure 20. Premium users

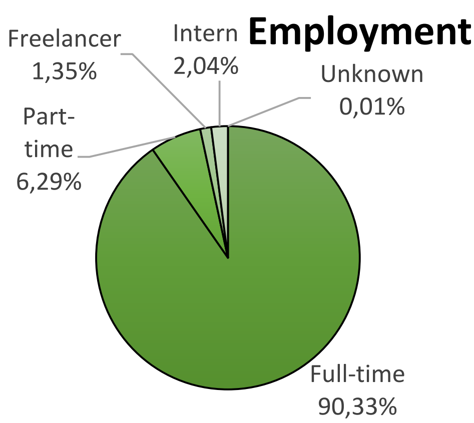
Field_of_studies	Number of users
0	901.096
1	3.225
2	33.572
3	168.925
4	18.202
5	116.948
6	8.808
7	31.129
8	213.036
9	2.079
Total	**1.497.020**

Table 9. User's field of studies

14.6 Items analysis of data

From all the job offers posted during 12 weeks only a 12,65% of them were paid by the company





15. RecSys Challenge solutions

Recommendation systems are software tools and techniques that play an important role in many domains such as e-government, e-business or the famous personal suggestions in e-commerce that is used to help users find those products (videos, songs, new articles) that match their interests. Most of the data used on recommender systems comes from user feedback signals such as user-product purchase. [23]

The main approaches of recommender systems are Collaborative Filtering, Content-Based Filtering, Knowledge Based and Hybrid Approaches. CBF tries to recommend items based on similarity in content, also, many researchers create hybrid techniques that are combinations of more than one filtering approach in order to improve the accuracy of its recommendations. [24]

For recruitment scenarios, where there are users that search jobs, CB recommender systems are often used to match a user profile with a job description and that can be useful for this candidates and recruiters that use career orientated platforms. Moreover, the fast growth of professional social networks like Linkedin and Xing that permits the exchange of information between users and recruiters need to predict the proper content to show at each user. With this objective, it is necessary to have data from user's profile, information about the content of the job offers as well as the historical activities of users, recruiters or job posts. [25]

There were 11 papers accepted in RecSys Challenge'16 were the goal was to design and implement recommender system algorithms for job recommendation, so for a specific user, the algorithm had to do a prediction about 30 job posts that the user would positively interact with (clicking, bookmarking or replying). So we would like to summarize the performance of all these teams in the RecSys Challenge in order to know about the algorithms and models they used and also its procedure.

Job and Talent team achieved the 11th position, Jobandtalent is an online employment platform that use Learning to Rank techniques, so this group wanted to test the recommendation generation approach that they use on Jobandtalent platform with Xing data in order to see their performance. Machine-learned ranking uses algorithms where first a set of queries with its associated relevant documents are used to build the training model, then similar data is used to evaluate the performance. In that case, the input for the MLR model would be users as queries and the output would be a ranking of items as documents. To create the relevance ranking they used the positive interactions between users and items such as clicks, bookmarks and replies giving more weight to reply interactions than a bookmark and more importance to a bookmark than a click. So, they used an evaluation metric that was able to consider the different levels of relevance that would be related to the different types of interaction. On their model they randomized the data and used an 80% of the queries to train and the resting 20% for the testing. Due to the large amount of data, they had to reduce the number of features in order to train the model because the algorithm they used couldn't deal with all the data, so they removed those features that didn't give much value. After trying different procedures they concluded that the features that contributed the most were those that were related with previous interactions. [26]

OneSearch team from Alibaba group took the 1st position of the challenge. They first build a hierarchical pairwise ranking model, the first layer used both Logistical Regression and Boosting Regression Tree to capture the semantic relevance and also the temporal characteristic of a user-item interaction. On the second layer, they did use the temporal information to capture the temporal interaction patterns between users and items, using a self-exciting point process named Hawkess Process. Integrating the relevance scores and the temporal intensity with ensemble models they did recommendations of the most relevant jobs at the right time. On their model they used the interactions click, bookmark, reply as positive feedback and delete as negative feedback. For the training period they used the interactions that took place during one particular week and then they did the testing with 150.000 users and 320.000 active post jobs. As they noticed some new users and new posts without historical interactions they used other features such as the similarity between demographic data, career level, education degree, work experience etc. and the matching between job roles with title or tags as well. They used data from 19 weeks as training data and validate the model with 1 week data and they noticed a strong correlation between validation and test scores. [25]

Information Sciences Institute team obtained the 5th place in the challenge's final leader-board combining Temporal Learning with Sequence Modelling. With that combination they wanted to capture the complex user-item activity patterns and improve jobs recommendations. They first propose a temporal-based ranking model to study the historical interactions between users and items, this evaluates how likely is that a user re-interacts with an item given its historical interactions. Then they use a Hybrid Matrix Factorization to recommend items than interacted with one user to another similar user. They also incorporate temporal information with the temporal re-weighted matrix factorization. Finally, they used sequence modelling with Recurrent Neural Networks to capture the set of interactions from the same user as a sequence ordered by time, that way, they obtain the sequential patterns in a user-item interaction. And finally, they develop an ensemble system to incorporate all the components above. Comparing their procedures, the proposed RNN-based model outperforms the matrix factorization models. [27]

PumpkinPie team from the Department of Electronics, Information and Bioengineering (DEIB) of the University of Milano reached the 4th position. Their approach consisted on using a Multi-stack Ensemble to combine different recommendation algorithms. The performance of ensemble is better when algorithms are diverse and can learn different relations on data. They used Collaborative Filtering to calculate user-based similarity and the similarity from two items. The Content Based algorithm was used to exploit both the similarity of the concepts of the items and the similarity of the concepts of the users with those items that the users interact with. According to DEIB team, it is different to recommend jobs than movies for example, because it is unlikely that someone watches or buy the same movie twice, while it is likely that a user views a job posting and later he returns to the same offer to compare it with others, so they processed the past interactions with filtering and ordering step. In the ensemble process, those items that were recommended by more than one different technique had higher probability to be a good recommendation. [28]

Falcon team from the Computer Science Department of the University of California got the 20th position using a Bottom Up approach. First, they analysed the data and how each dataset can help providing quality recommendations to users. Focusing on the interactions between users and items, deleted interactions were disregarded and reply interaction were on top, followed by bookmark and clicks on the bottom. So, a user that had interacted positively with an item, was more likely to interact with the same item again. Considering the Collaborative Filtering, similarity between users would determine that if user 1 interacts with item X and user 2 is similar to user 1, then user 2 is more likely to interact with item X. Moreover item similarity would say that if item X is similar to item Y, and user 1 interacts with item X it is more likely that user 1 also interacts with item Y. To define similarity, they first divide the users and items into clusters, so the ones from the same cluster would have similarity 1 and from different clusters would have similarity 0. Moreover, they used a score to assign weights to the components from users and items data. Gradient Boosting method was used to improve the results and deal with the missing data using Random Forests. [29]

Avito team was in 7th position in the challenge's final leader-board, using three different types of models in their solution. As almost half of the users had not made enough number of actions, they couldn't use Collaborative Filtering for them and had to use a Content Based approach. First of all, they performed the item-based CF using three different similarity metrics (Jaccard, Cosine and Pearson) that consisted on calculate the similarity of items that had been interacted for same user. Then, the Factorization Machine could predict the probability of any user-item pair interaction by learning latent factors for all variables. Finally, CB topic model was used to compute the coincidence between the job roles of a user and the title and tags of an item. They also added prior impressions information that consisted on determining whether a user had got an item in their list of recommendations before the week of the prediction. As Cosine similarity showed low quality they finally remove it from the model. So, doing a hybrid scheme with CF and CB approach let them had good results and they also tried to improve the results by using the prior impressions. [30]

UOP team belonging to the Department of Mathematics of the University of Padova divide the types of interactions into positives (click, bookmark and reply) and negatives (delete). Their framework contain evidence about users and items and similarity between users or between items. They describe some predictors that considered different aspects from the dataset and that would be cast into their general framework: interaction-based (if two users are similar they interact with the same items, if two items are similar they interact with the same user), evidence impression based (it has been a prior interaction between a user and an item), tag and title based (coincidence with job role of a user), popularity based (general ranking of items) and FOS-based (user representation identified by field of studies). Using a method that linearly combine the predictors and learns the weights by solving a quadratic optimization problem, given a user and an item it defines the scoring vector of both and it is tried to minimize the squared difference between the score predicted and the real score. As the data was very large, they decided to take 3.000 random positive and 3.000 random negative pairs and then divide the data into training data (70%) and testing data (30%). They saw on the results that combined predictors performed better than single ones. [31]

BUTE team belonging to the Department of Telecommunications and Media Informatics of the Budapest University of Technology and Economics defined various models analysing the dataset and then try different combinations of categories to optimize the result. They divided the data into two type of vectors: Events (time, user_id, item_id, interaction_type, value) and User/Item metadata (id, key1, key2,…,keyN). They also distinct three types of users: users who have metadata only, inactive users who have impressions only and active users who have interactions. Also they remark the point that users use to re-click in an already seen item, so they used recalling recommendations and already seen items method. Nearest Neighbors method performs better than Matrix Factorization due to the sparseness of user-item matrix. For users that are new and don't have a record of previous interactions, they used user metadata-based popularity making groups of users with job roles and edu_fieldofstudies. Then Cosine similarity is used to compare two items using tags, title, industry, discipline, country and region. In the optimization process, data from the first weeks were used to do the training and data from the last week to do the testing. A Forward Predictor Selection method is used to choose the candidate predictors and the sub-predictors, also using cross-validation to avoid over-fitting. Also an Omit Method is applied for those items that are recommended too often or targeted poorly, so the worst items are omitted from recommendations. [32]

MIM team from the Institute of Informatics of the University of Warsaw, obtained the 2nd position on the challenge and used a two phased algorithm that consisted of candidate selection followed by candidate ranking. The first part of the solution consisted on calculating for each user some candidate items using similarity of the data information between two users or two items. The second part was to learn the probability of interaction between a user-item pair, using Gradient Boosting Decision Trees. The training set was all the data except the last week that was used to evaluate the model, a later increase on the size of training set made an improvement of the results. They described 12 groups of features to use: Event based (percentages of items/users that have attributes equal to other items/users), Item global popularity (number of clicks by any user), Collaborative filtering most similar (measuring similarity between item Y and items clicked by user A and users similar to user A that interacted with item Y), Content based user-item similarity (comparison of attributes from users and from items such as career_level and jobroles with titles/tags)…Each feature group was ranked from more to less importance. [33]

CAS team from the Institute of Computing Technology got the 10th place in the challenge's final leader-board. They state two challenges: Matching (the best pair job-candidate depend on various attributes) and Ranking (determine the order to present the items to the user). They capture the relationship between the activity of a user and the popularity of an item so those new users would prefer popular items in most cases. The solution method consist first on ratings for implicit feedback data, using clicking, bookmarking, replying and deleting to rate the interest of the user. Then, CF approach using latent semantic modelling to find similar users in interactions and give a probability whether the user will or not interact with a particular item. They also used the CBF approach that computes the similarity of items in content and solve the new item cold start problem. So ensemble methods combine CF and CBF and perform better than a single model. For each user, their approach found popular job postings that are important to other similar users and new job postings whose content matches the user's specific interests. [34]

iMinds team from Ghent University proposed an hybrid algorithm that combines a Content Based and KNN approach. Content Based algorithm matches features of candidate's recommendations and job postings of historical interactions. They create an output that would be a recommendation score for each user-item pair. KNearest Neighbor approach searches for the job postings that are the most similar to the postings that the user interacted with in the past. Their goals for the hybrid recommendation were: Scalable in the number of users and jobs of the system (make it work when the number increases), Enabling incremental updates of the model (the possibility of adding data such as new users, job postings and interactions), Fast score calculation for new job postings (fast algorithm to generate recommendations for new items). The result of this combination that is weighted average (KNN's weight increases if there is more interaction data of the user available) is a fast, lightweight, scalable algorithm that allows incremental data updates, generating recommendations with a proper evaluation score. [23]

16. Choosing the right data

If we think of creating an ANN to match users with items in order to use it for recommendations, we should think about two type of recommendations:

- We can recommend a job post to a user that we will call *users recommendations*.
- We can recommend a user to the recruiter that posts a job that we will give the name of *recruiter recommendations*.

Having analysed all the data we see that for the second type of recommendations, we can only take the recruiter interest interactions. Nevertheless, we don't have much data because we found out that most of the recruiter interest interaction type were repeated on the dataset, so once we deleted the repeated ones there were only 2554 interactions left. Moreover, Recruiter interest interaction is described as clicks from a recruiter on a user profile, that doesn't exactly mean that the user would be a good candidate, recruiters may just do a screening of the candidates. So, we decided to focus on users recommendations.

16.1 Users recommendations

First of all, we have to think about the users data that shows us some kind of interest for a job post. Having a look on interaction types we got click, bookmark, reply and delete. First, we categorized as positive interactions click, bookmark and reply and as negative interactions the delete ones.

So first we tried to create an ANN with all the clicks interactions and the ANN had to determine those users that would perform a later bookmark or reply on the item, but the performance was not good enough. After that we realized that click interactions may show little or no interest because a user can click indiscriminately many job offers but commonly, with those items that has a higher interest it will bookmark or reply soon or later. Also, performing a delete interaction may not mean a disinterest or negative feeling, just a neutral feeling without intention of interacting.

For that reason, we finally decided to choose all the bookmark and reply interactions, that way, given a user and an item information we would ask the ANN to predict those user-item pairs in which the user is likely to bookmark and reply the item.

16.2 Input and output data

To prepare the input and output data we need to have a sample of user-item pairs that has either bookmark or reply at least one time and others that have not done any of those two interactions. First, we filtered the interactions dataset with only the bookmark and reply types. Then we randomized a list of users and items with excel functions and we created random pairs of users and items. Finally we got a total of 717.874 user-item pairs with half of them had performed a positive interaction and the other half were random.

From Access tool we imported all this user-item pairs and we started to create a table with all the information from the original database. The target data would be a column showing number 1 for the pairs that had positively interacted and a 0 with the random pairs that had not interacted.

The input data would be a total of 19 columns of information: career level (user, item), discipline (user, item), industry (user, item), country coincidence, region (user, item), willingness to change job, education degree, CV entries, years of experience, years in current job, employment, premium user, paid job post, jobrole/tags coincidence and jobrole/title coincidence.

For country coincidence we created a query on Access that classified with a 1 those pairs that the user and the item had the same country, and with a 0 those with no coincidence.

In order to create jobrole coincidence with tags and titles, from one hand we imported Access data with the user-item pairs and columns with jobroles and tags to one worksheet of Excel and then imported user-item pairs and columns with jobroles and titles to another worksheet of Excel. Then we used match function to see if there was any coincidence for each user-item pair and we created a yes/no column using number 1 on those pairs that had one or more coincidence and number 0 for those that didn't have any coincidence. After that we added that information to the Access table again. We first tried the data without the jobrole coincidence on the ANN in Matlab, and after that we tried it again adding the jobrole coincidence, the performance was better with all the inputs described before.

Moreover, we wanted to see if choosing part of the inputs, the performance would perform better or similar than using all of them. So, as I am working on a HR consulting, I created a document that explained the idea of the project in order to present it to the consultants of the company that have been recruiting people for a long time. That way they could give me better advice about which fields of the data were more likely to take into consideration for a candidate-job match. After the debate, it was decided that the more important data was: career level, discipline, region, jobrole-title coincidence, jobrole-tag coincidence and willingness to change job. So we tried to use this inputs in the ANN and the performance was worse than using all the inputs detailed before.

17. Creating the ANN for users recommendation

17.1 Architecture

The ANN will be a two-layer feed-forward network with sigmoid hidden neurons and 1 linear output neuron, as it is shown above in the Matlab schema.

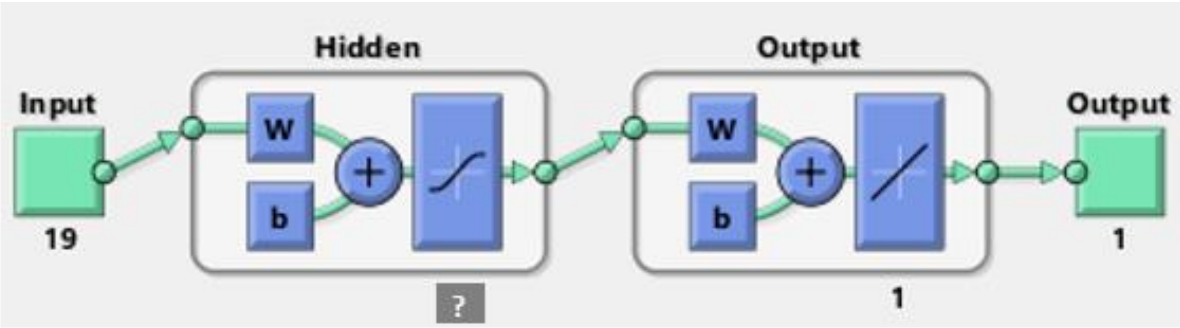


Figure 22. Architecture of users recommendation ANN

17.2 Algorithm

The network will be trained with Levenberg-Marquardt backpropagation algorithm, it is a very popular training method in NN because it has high accuracy and robustness. [35] Levenberg-Marquardt algorithm is an iterative technique that locates the minimum function that is expressed as the sum of square of nonlinear functions. [36]

In feed-forward networks the performance function have the form of a sum of squares, so the Hessian matrix is approximated like $\mathbf{H=J^TJ}$ and the gradient $\mathbf{g=J^Te}$; being **H** the Hessian matrix, **J** the Jacobian matrix with the first derivatives of errors from then net with respect to the weights and biases, and **e** the vector of errors from the net. That way, the Jacobian matrix is computed through backpropagation technique that is not as complex as computing Hessian matrix. [11]

Levenbert-Marquard backpropagation requires more memory because it has to storage matrix that can become large m x n (where **m** is the number of training sets and **n** the number or weights and biases of the network). Although Levenberg-Marquardt takes more memory, it performs better than Bayesian Regularization and Scaled Conjugate Gradient.

We determine the validation checks on 6 and the maximum number of iterations on 1000, so the training will continue till the generalization stops improving (until the validation error fail to decrease for 6 consecutive iterations) or till it performs 1000 iterations.

17.3 Input and output

As described before, the input data is a matrix that contains 717.874 samples of 19 elements and output (target) data is a matrix with also 717.874 samples with 1 element, representing static data. We will choose the matrix rows Matlab option because our data is organized by rows, each row makes reference to a user-item pair.

17.4 Type of learning and data division

Automatic learning is based on learning properties of a dataset and applying it to new data, for this reason in the experiments of automatic learning we make a partition of the data that we have. Our ANN will divide the data randomly, using 70% of the samples for training, 15% for validating and the other 15% for testing.

The ANN will use supervised learning so the algorithm generates a function that relates each input to the desired output. Each training body has an input vector x and a target vector t (target). So, a set of data is provided to the system with the expected values. The ANN will use regression because its aim is to get a real number. The regression model will have the form y = f(x,W), being x the inputs and W the weights, these are the parameters that ANN have to adjust to get the expected outputs.

Learning involves adjusting the parameters to reduce the discrepancy between the desired output (target), t, in each training and the output produced by the model. For the regression model the least squares method is used, $\frac{1}{2}(y-t)^2$.

17.5 Number of hidden neurons

On our research we have seen some formulas to calculate the number or neurons on the hidden layer but none of them where scientifically proved. Moreover, it depends on the data and the ANN architecture and algorithm. So, we have tested several ANNs by changing the number of neurons, that way, we can analyse the MSE and regression R values in order to decide which would be the best solution for the problem of the project.

17.6 Script

First of all, we import the input and the target data to Matlab Workspace and then we can use the Command Window to create the ANN.

```
% Import the data
x = input';
t = target';

% Choose the Training Function, we will use Levenberg-Marquardt backpropagation
trainFcn = 'trainlm';

% Create a Fitting Network, we will try with different number of neurons for the hidden layer.
hiddenLayerSize = 15;
net = fitnet(hiddenLayerSize,trainFcn);

% Division of Data into 70% for Training, 15% for Validation and 15% for Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

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

% Testing the Network (output, error, performance)
y = net(x);
e = gsubtract(t,y);
p = perform(net,t,y)

% View the Network
view(net)

% View the Error Histogram and the Regression Plot
ploterrhist(e);
plotregression(t,y);
```

18. Results

Once we have trained the ANN changing the number of hidden neurons, we can analyse which is the best size to use on the created ANN with the data selected.

Training several times with the same number of hidden neurons will come to different results because sampling is random and the network uses different initial conditions (weights and biases) each time. Nevertheless, we have perceived that if we train twice the same ANN with the same number of hidden neurons there can be a difference on the number of interactions and training time but the performance according to MSE and R regression values is not that distinct.

We are going to present the results for each hidden neurons size according to the regression R values, the MSE, the number of iterations and the training time. So we are going to choose the number of neurons that has:

- Regression R value close to 1: that means a close relationship between outputs and targets while values close to 0 mean random relationship.
- Low Mean Square Error: that means less average squared difference between outputs and targets.
- Low number of iterations and training time.

18.1 Regression values

If we look at the Regression R values results, we appreciate an escalation as number of neurons increase, also for 14 or more hidden neurons the values stay around 0,75.

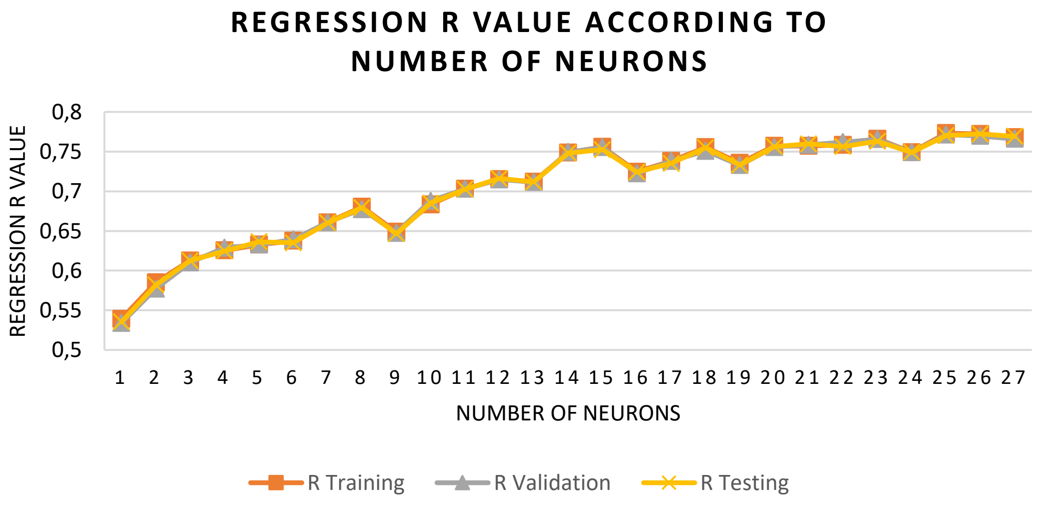


Figure 23. 2D line plot for Regression R values

18.2 Mean Squared Error

If we have a look at the MSE results, we appreciate a decrease while number of neurons are rising, also for 14 or more hidden neurons the values stay around 0,11.

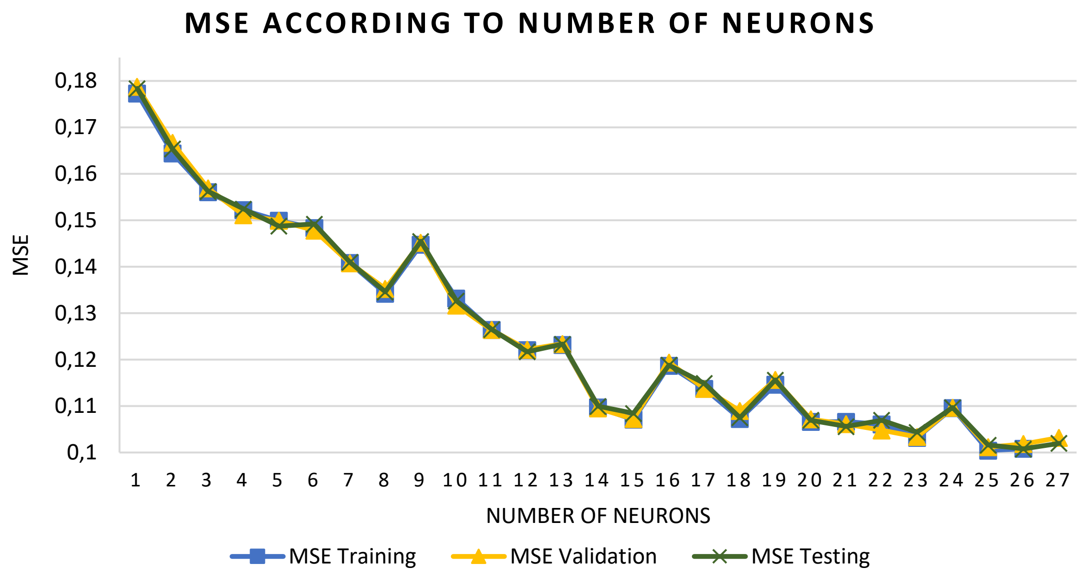


Figure 24. 2D line plot for MSE values

18.3 Number of iterations

The maximum possible iterations is 1000 because it is one of our stop criterion on the ANN training, the other stop criterion is when the generalization stops improving during 6 consecutive iterations. According to what we have seen on the other plots, the number of neurons with a high regression value, low MSE and low number of interactions are sizes: 14, 15, 24 and 26.

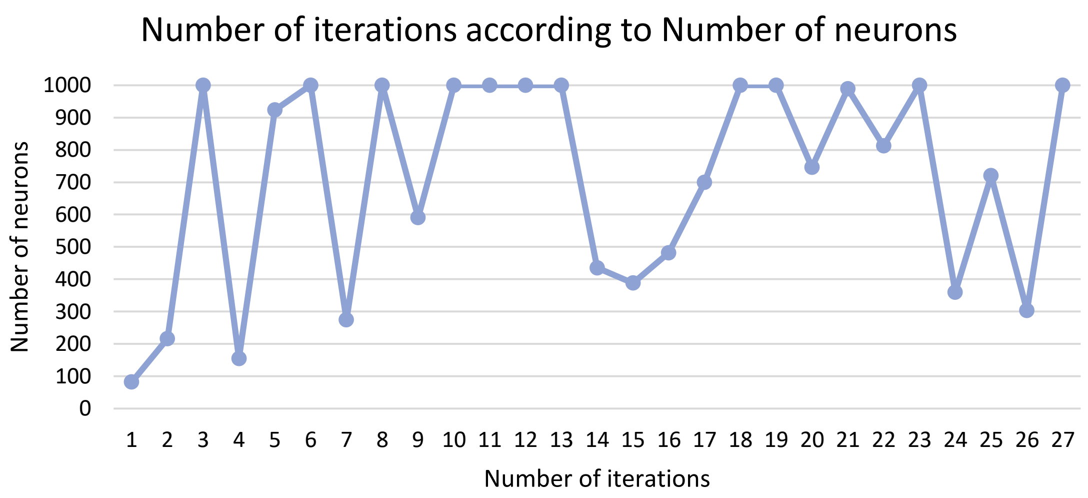


Figure 25. 2D plot for Number of Iterations

18.4 Training Time

From the hidden neurons sizes selected before, with that plot we can conclude that 15 hidden neurons would be the most effective size for our ANN.

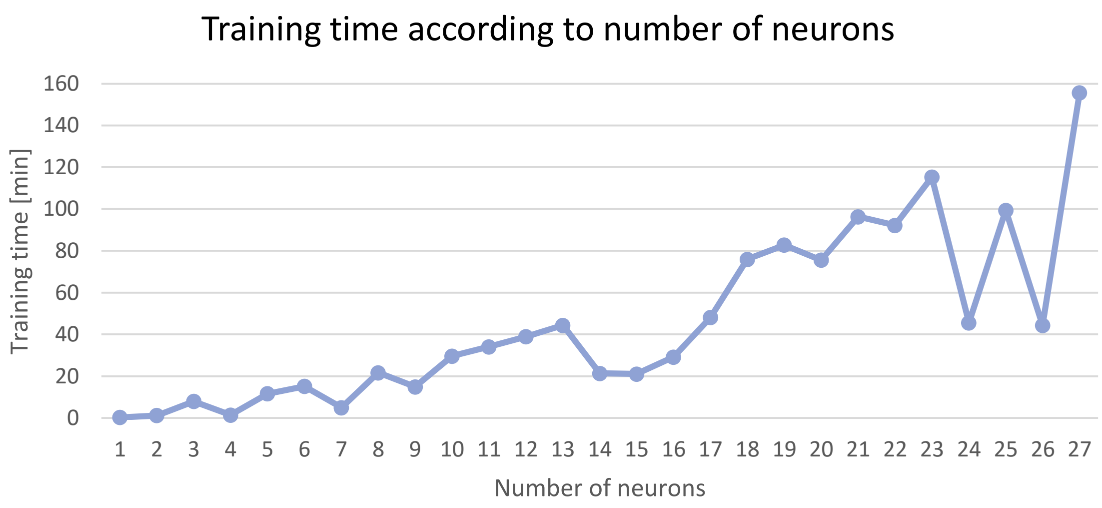


Figure 26. 2D plot for Training time

18.5 Final Discussion

Finally, our ANN will be a two-layer feed-forward network with 15 sigmoid hidden neurons and 1 linear output neuron, fitting a sample composed by 717.874 user-item pair with 19 inputs and 1 output.

On the results we don't perceive overfitting because the performance on test results and training are always quite similar.

The Error Histogram for the ANN with 15 hidden neurons, is useful to validate the performance of the ANN because it shows the error of training, validation and test data on the same bar chart. Moreover, we don't see any outlier, because generally all each pair has fit as well as the rest of data.

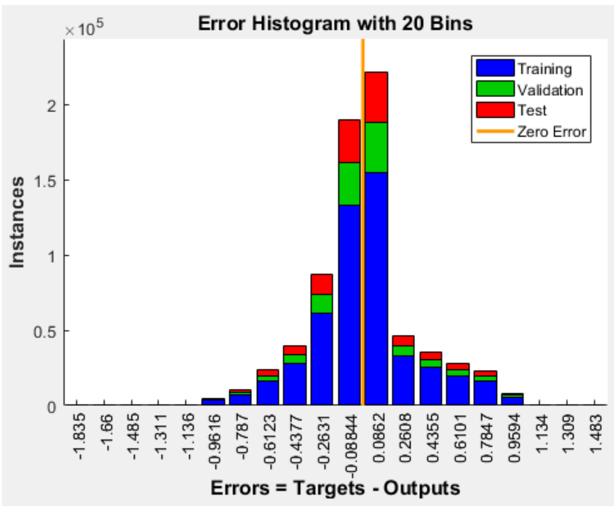


Figure 27. Error Histogram for ANN with 15 hidden neurons

The Regression Plot for the ANN with 15 hidden neurons is also useful to validate the performance of the ANN because it compares the targets with the outputs during the training, the validation and the test phases. Here, the regression line is y=0,7554·x, so it is a 37,07 degree line.

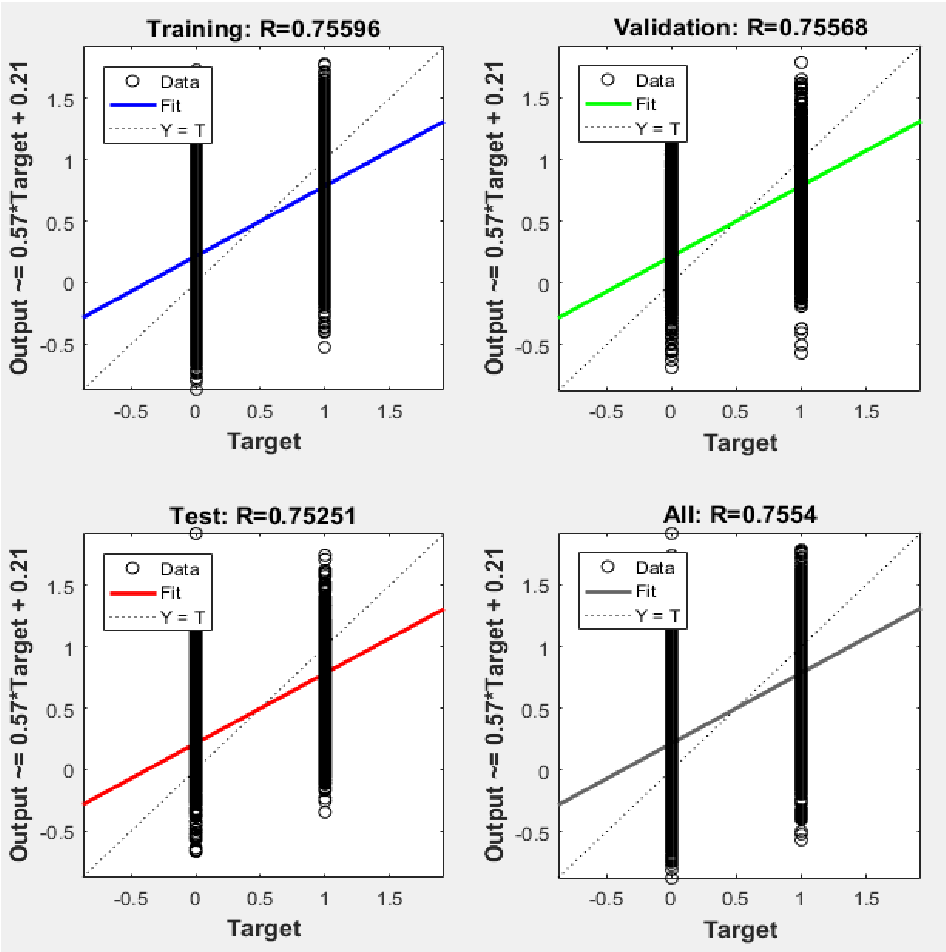


Figure 28. Regression Plot for ANN with 15 hidden neurons

On the results we also obtain the Output matrix, here there is an example of output values compared to target values:

TARGET	1	1	0	1	0	0	1
OUTPUT	0.9591	1.0050	0.1084	0.9291	0.1701	0.0350	0.8942

Table 10. Comparison between outputs and targets

In the Annex we can find a Summary table of all the results, also the Regression Plots and the Error Histogram according to the number of hidden neurons.

One common way to determine the accuracy of a prediction model is to compute the MSE because smaller values are better and a 0 means perfect prediction. On *Figure 29* we can observe the accuracy evolution on the ANN with 15 hidden neurons during its performance (388 iterations).

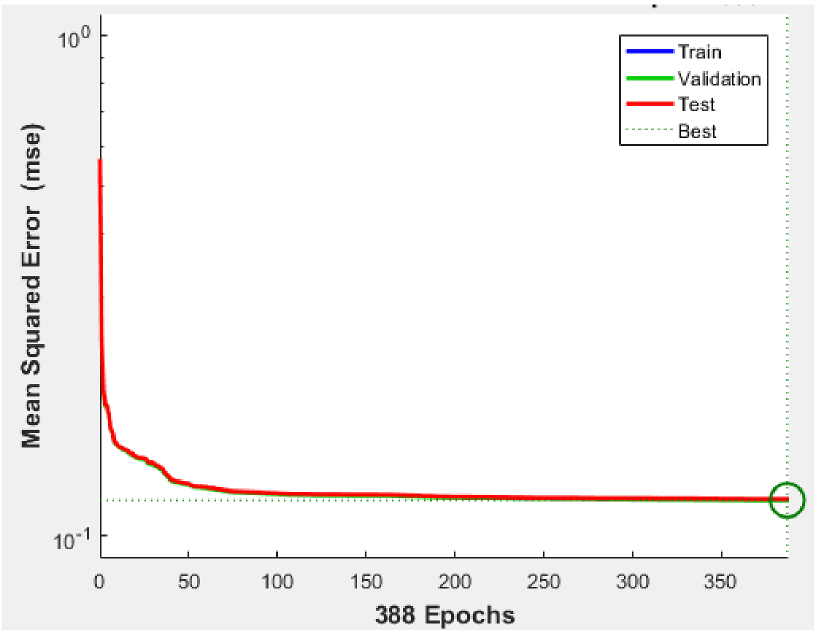


Figure 29. Validation performance plot for ANN with 15 hidden neurons

Our ANN model does not predict 0 or 1, it is predicting a range of values shown on the graphic below.

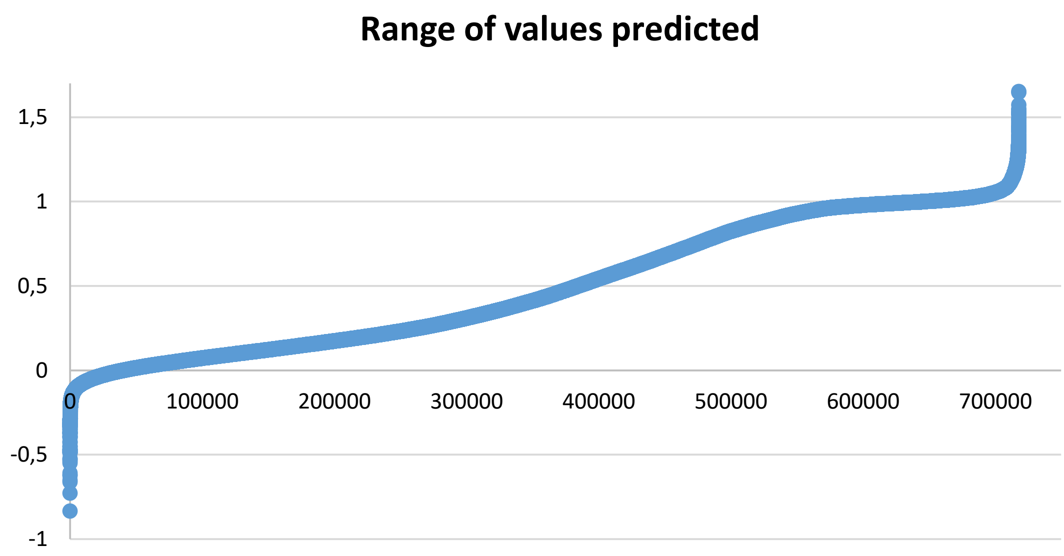


Figure 30. Range of output values for ANN with 15 hidden neurons

In order to compute the predictive accuracy as the percentage of correct predictions it can be that one model with better MSE have a worse predictive accuracy than a model with a worse MSE, because MSE and predictive accuracy do not always agree when it comes to identifying an optimal prediction model. Nevertheless, we will calculate the predictive accuracy for the outputs given by the ANN with 15 hidden neurons.

For predictive accuracy our hypothesis is that if the prediction value is lower than 0.5 we figure the prediction is 0 and if the prediction value is greater than 0.5 the prediction is classified as a 1. So, under that rule we can calculate the values of the confusion matrix comparing outputs with targets, P meaning 1 and N meaning 0.

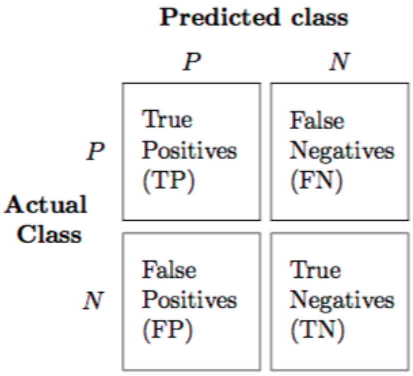


Figure 31. Confusion Matrix

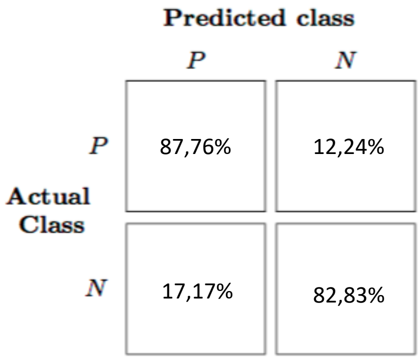


Figure 32. Confussion Matrix with results

True positive (TP): Correctly classified, user likely to perform a booking or reply for a specific item. That means that the 87,76% of the predicted outputs are correctly classified as a 1.

True negative (TN): Correctly classified, user unlikely to perform a booking or reply for a specific item. That means that the 82,83% of the predicted outputs are correctly classified as a 0.

False Positive (FP): Incorrectly classified, saying that a user is likely to perform a positive interaction when it is unlikely. That means that the 17,17% of the outputs are classified as a 1 when they had to be classified as a 0.

False Negative (FN): Incorrectly classified, saying that a user is unlikely to perform a positive interaction when it is likely. That means that the 12,24% of the outputs are classified as a 0 when they had to be classified as a 1.

If we calculate the total percentages of correct and incorrect predictions we get that with our hypothesis, the 85,12% of the predictions are successful while the 14,88% are wrong.

Conclusions

In this project report, a new method of creating recommendations matching candidates and job offers information has been explained.

Data mining is a set of techniques and technologies that allows the exploration of large databases, automatically through AI methods, statistics and database systems, with the aim of finding repetitive patterns, trends or rules that explain the behaviour of data in a given context. [37]

This project would be a clear example of what we call data mining: using Microsoft Access system we have explored XING database and we have fit the data with an ANN in the Human Resources context.

As we have seen, a study of ANNs requires knowledge from computer science, artificial intelligence, statistics and mathematics, neurophysiology, cognitive science and psychology, physics, control theory, parallel processing and hardware. [10]

The main problem we have had to develop this project has been how to handle big data, the first programmes we tried to use to manage with it were not successful, and till we didn't learned how to use Access we couldn't do much. After that it was necessary to take only a part of the data to use it on our ANN, and we finally succeed on choosing the meaningful data.

At this point we can conclude that we have achieved the necessary theoretical contents for the understanding of artificial neuronal networks and their learning, in particular through the back propagation algorithm, as well as its fields of applications and its advantages and disadvantages.

As future research on Human Resources Management, it is necessary that social networks oriented to employment collect data from its application or platform and try to make user's profile and job offer posts as complete as possible, in order to have the right parameters to develop recommendations and create effective matching between candidates and job positions.

Also, ANN applied to HRM can study how to make working conditions improve, how employees should be distributed in their company regarding their abilities as well as identify those employees that are keen to stay longer time in the organization.

Bibliography

[1] T. Chamorro-Premusic, C.Steinmetz (2015). *Perfil profesional en las redes sociales*. Mente y Cerebro

[2] S.K.Mishra, M.Reddy (2016). *A Bottom-Up Approach to Job Recommendation System.*

[3] J. Nguyen, G. Sánchez-Hernández, A. Armisen, N. Argell, X. Rovira, C. Angulo (2017). *A linguistic multi-criteria decisión-aiding system to support university career services.*

[4] R. Hecht-Nielsen (1990). *Neurocomputing*.

[5] J. Zurada (2001). *Introduction to artificial neural systems.*

[6] S. Haykin (2002). *Neural networks: a comprehensive foundation.*

[7] N. Yadav, A. Yadav, M. Kumar (2015). *An Introduction to Neural Network Methods for Differential Equations.*

[8] B. Macukow (2016). *Neural Networks – State of Art, Brief History, Basic Models and Architecture.*

[9] D. Kriesel (2007). *A brief Intoduciton to Neural Networks.*

[10] A.K. Jain, J.Mao (1996). Artificial Neural Networks: A Tutorial

[11] H. Demuth, M.Beale (2002). *Matlab: Neural Network Toolbox User's Guide*

[12] O.S.Eluyode, D.T.Akomolafe (2013). *Comparative study of biological and artificial neural networks*

[13] G.L. Acebedo Orduña, E.F. Caicedo Bravo, H. Loaiza Correa (2006). *Selección de personal mediante redes neuronales artificiales.*

[14] L. Li, H. Zhu (2010). *Enterprise Human Resource Management Outsourcing Decision-Making Model Based on BP Neural Network.*

[15] C. Ning (2010). *The Application of Neural Network to the Allocation of Enterprise Human Resources.*

[16] X. Wang, Y. Jiang (2010). *The Study of Enterprise Human Resource Evaluation Research Based on Neural Network.*

[17] X. Gao, L. Fan, Y. Den (2012). *System Method of Evaluating Core Competence of Enterprise Based on Artificial Neural Network.*

[18] R.O.Akinyede, O.A.Daramola (2013). *Neural Network Web-Based Human Resource Management System Model (NNWBHRMSM).*

[19] D.Anghel, A. Ene, N. Belu (2013). *A Matlab Neural Network Application for the Study of Working Conditions.*

[20] S.Torres López, M.L. Aldana Cuza, P.Y. Piñero Perez, L.A. Piedra Diéguez (2016). *Multilayer neural network for assessment of labor competences.*

[21] F. Abel, A. Benczur, D. Kohlsdorf, M. Larson, and R. Palovics (2016). *Recsys challenge 2016: Job recommendations. Proceedings of ACM RecSys '16.*

[22] R. P. F. Abel, D. Kohlsdorf (2006). *ACM recsys challenge 2016: Training data.*

[23] T.Pesemier, K.Vanhecke, L.Martens (2016). *A Scalable, High-performance Algorithm for Hybrid Job Recommendations.*

[24] C.Zhang, X.Cheng (2016). *An ensemble method for job recommender systems.*

[25] W.Xiao, X.Xu, K.Liang, J.Mao,J.Wang (2016). *Job Recommendation with Hawkes Process.*

[26] J.I.Honrado, O.Huarte, C.Jimenez, S.Ortega, J.R.Perez-Aquera, J.Perez-Iglesias, A.Polo, G.Rodriguez (2016). *Jobandtalent at RecSys Challenge 2016.*

[27] K.Liu, X.Shi, A.Kumar, L.Zhu, P.Natarajan (2016). *Temporal Learning and Sequence Modeling for a Job Recommender System.*

[28] T.Carpi, M.Edemanti, E.Kamberoski, E.Sacchi, P.Cremonesi, R.Pagano, M.Quadrana (2016). *Multi-Stack Ensemble for Job Recommendation*

[29] S.K.Mishra, M.Reddy (2016). A Bottom-Up Approach to Job Recommendation System

[30] V.Leksin, A.Ostapets (2016). *Job Recommendation Based on Factorization Machine and Topic Modelling.*

[31] M.Polato, F.Aiolli (2016). *A preliminary study on a recommender system for the Job Recommendation Challenge.*

[32] D.Zibriczky (2016). *A Combination of Simple Models by Forward Predictor Selection for Job Recommendation*.

[33] A.Pacuk, P. Sankowski, K.Wegrzycki, A.Witkowski, P.Wygocki (2016). *RecSys Challenge 2016: Job Recommendations Based on Preselection of Offers and Gradient Boosting.*

[34] C.Zhang, X.Cheng. (2016). *An ensemble method for job recommender systems.*

[35] S. Konstantinidis, P. Karamiperis, M. Sicilia (2015). *Enhancing the Levenberg-Marquardt Method in Neural Network training using the direct computation of the Error Cost Function Hessian*

[36] M.I.A.Lourakis (2005). *A Brief Description of the Levenbert- Marquardt Algorithm Implemented.*

[37] S.Strohmeier, F.Piazza (2013). *Domain driven data mining in human resource management: A review of current research.*

Annex

Number of hidden neurons	Number of iterations	Training time [min]	MSE Training	R Values Training	MSE Validation	R Values Validation	MSE Testing	R Values Testing
1	82	0,33	0,177292	0,539288	0,178732	0,533944	0,178364	0,535312
2	216	1,28	0,164418	0,585086	0,166686	0,577312	0,165361	0,581854
3	1000	8,02	0,156095	0,612877	0,156898	0,610249	0,156232	0,612429
4	155	1,47	0,152194	0,625488	0,151117	0,628937	0,152449	0,624675
5	923	11,67	0,149925	0,632691	0,149926	0,632699	0,148757	0,636381
6	1000	15,22	0,148353	0,637641	0,147817	0,639322	0,149214	0,634947
7	274	4,98	0,140832	0,660809	0,140782	0,660962	0,140991	0,660332
8	1000	21,7	0,134239	0,680472	0,135286	0,677398	0,134576	0,679482
9	591	14,93	0,144737	0,648882	0,145198	0,647461	0,145486	0,646573
10	1000	29,7	0,133209	0,683492	0,13169	0,687928	0,132647	0,68513
11	1000	34,15	0,126402	0,703127	0,126403	0,703132	0,126529	0,702768
12	1000	38,97	0,122095	0,715273	0,122172	0,715062	0,121722	0,716319
13	1000	44,43	0,123195	0,712192	0,123417	0,711575	0,123337	0,711794
14	435	21,42	0,109734	0,749041	0,109531	0,749594	0,109984	0,748373
15	388	21,1	0,10713	0,755961	0,107236	0,755681	0,108442	0,752508
16	482	29,25	0,118713	0,724668	0,119411	0,722741	0,11882	0,724374
17	700	48,15	0,13755	0,738226	0,113752	0,738228	0,114894	0,735158
18	1000	75,98	0,107285	0,755553	0,108988	0,751032	0,107517	0,754938
19	1000	82,81	0,114616	0,73589	0,15655	0,733064	0,115621	0,733173
20	746	75,62	0,106618	0,757314	0,107168	0,755864	0,106964	0,756402
21	989	96,32	0,106674	0,757166	0,106154	0,758554	0,10563	0,759921
22	812	92,25	0,106082	0,758729	0,104815	0,762063	0,106976	0,756374
23	1000	115,27	0,103217	0,766244	0,103386	0,765804	0,104398	0,763184
24	360	45,61	0,109601	0,749396	0,109694	0,749144	0,109698	0,749138
25	720	99,4	0,100469	0,773383	0,101202	0,771486	0,101627	0,770393
26	303	44,42	0,100845	0,77241	0,101875	0,769751	0,100817	0,772484
27	1000	155,7	0,102486	0,768151	0,103231	0,766221	0,102014	0,769382

Table 11. Summary of ANN results

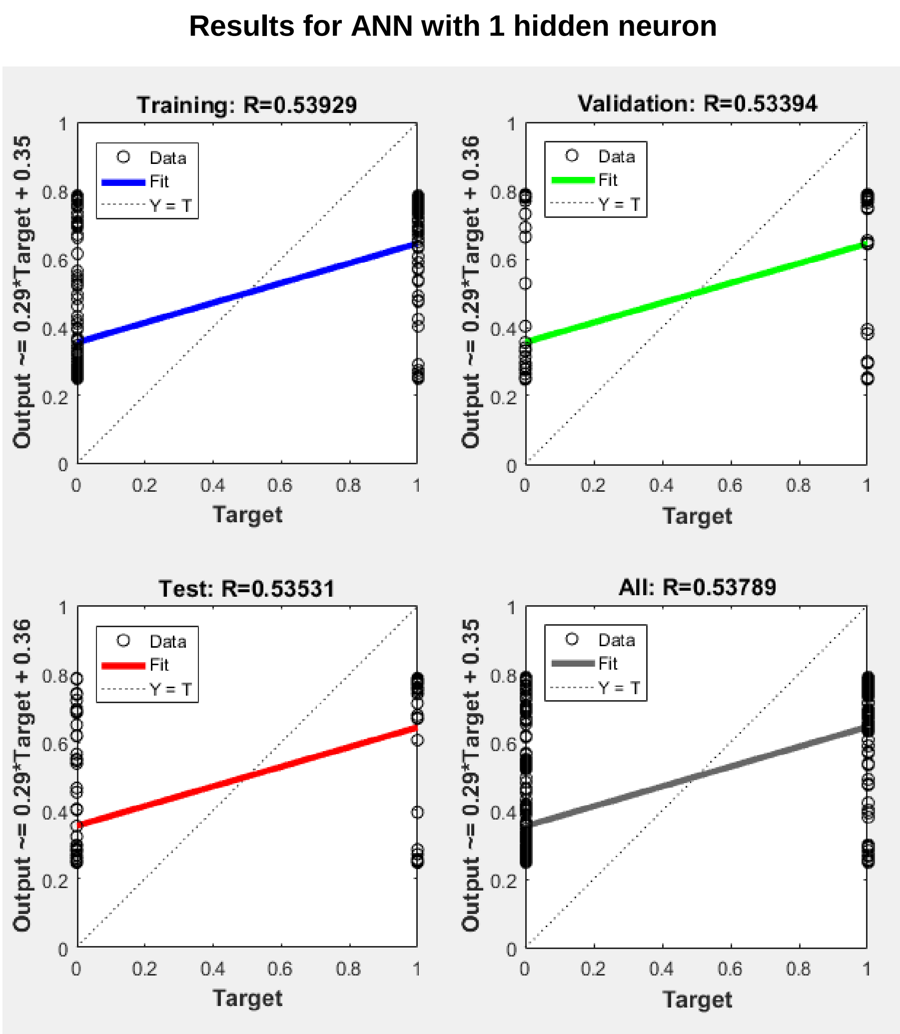


Figure 33. Regression plot for ANN with 1 hidden neuron

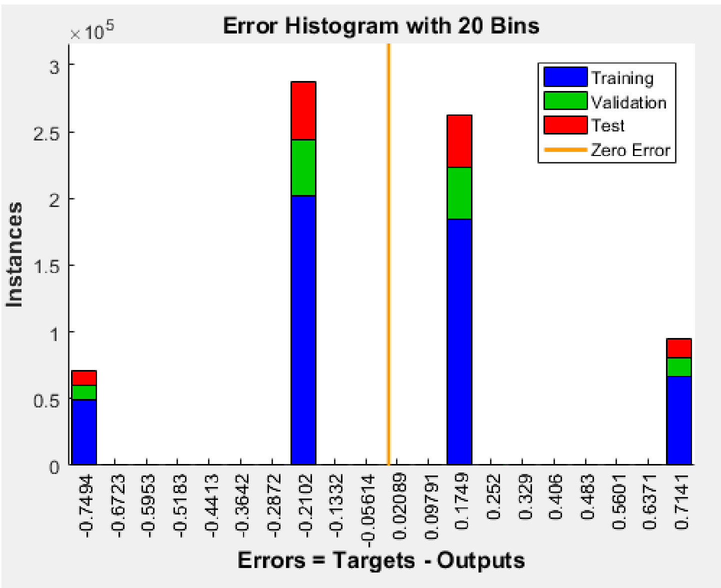


Figure 34. Error histogram for ANN with 1 hidden neuron

Results for ANN with 2 hidden neurons

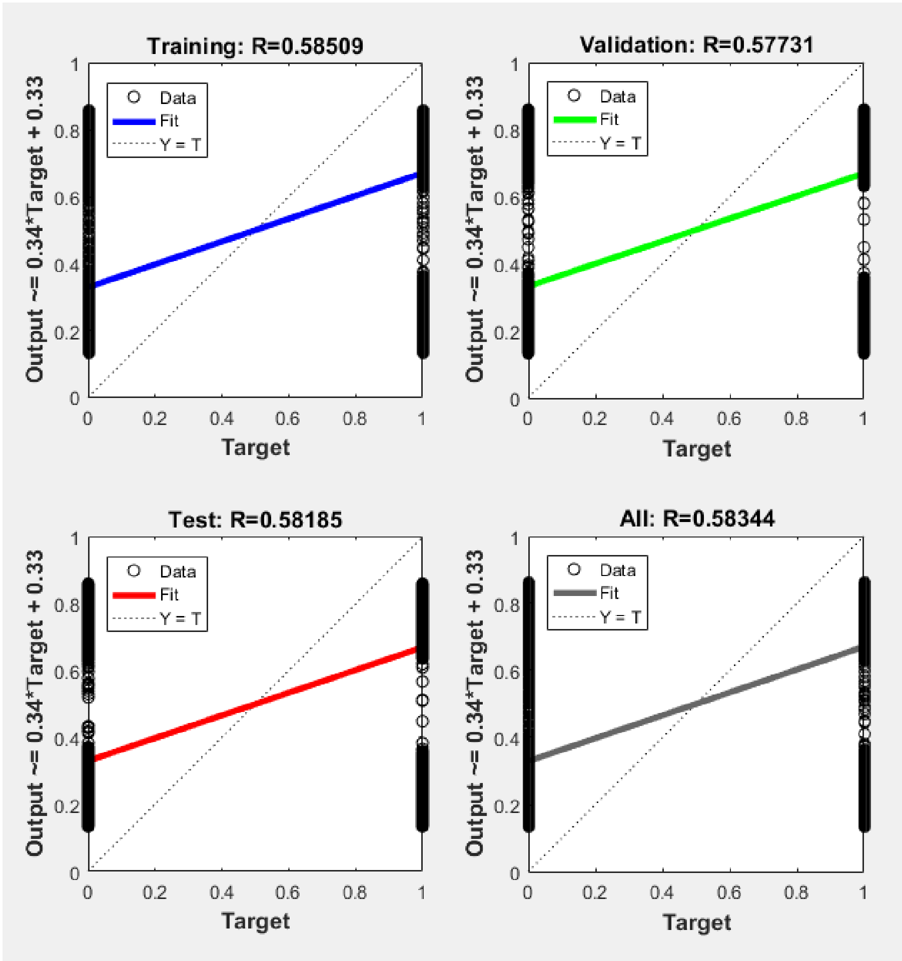


Figure 35. Regression plot for ANN with 2 hidden neurons

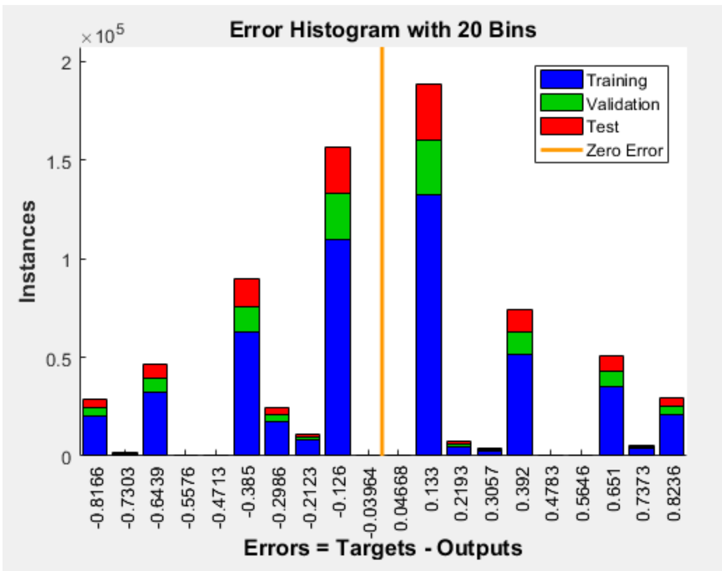


Figure 36. Error histogram for ANN with 2 hidden neurons

Results for ANN with 3 hidden neurons

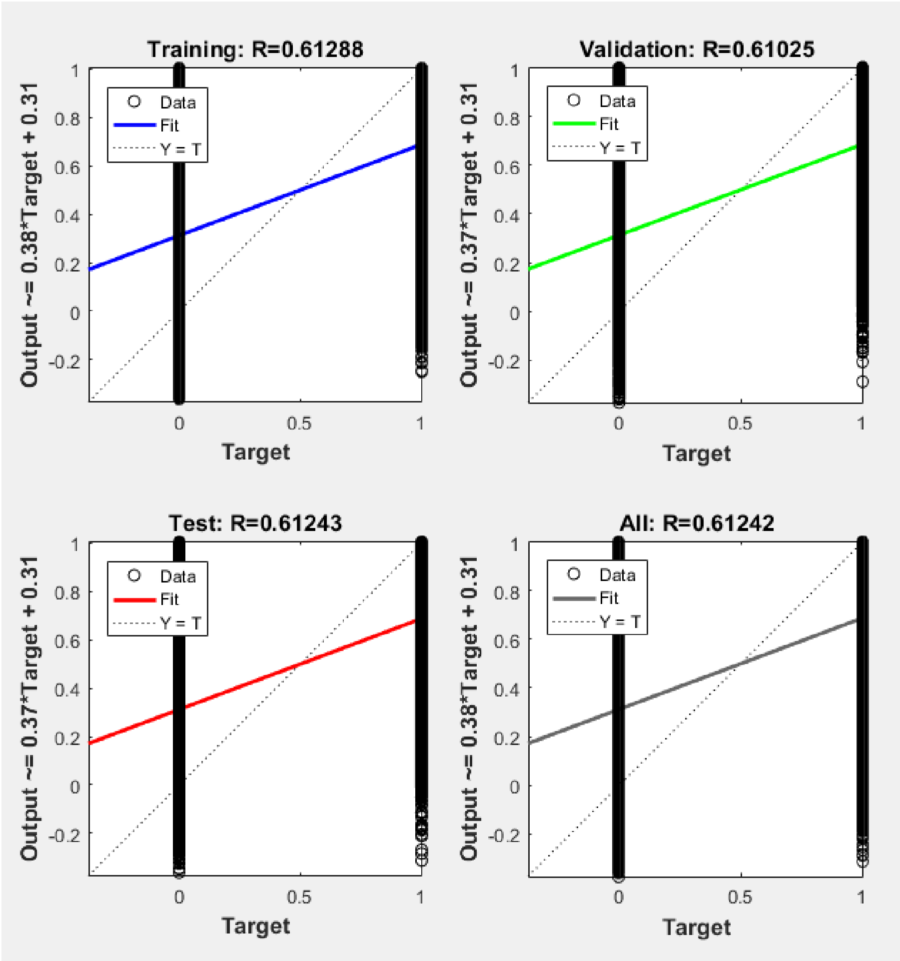


Figure 37. Regression plot for ANN with 3 hidden neurons

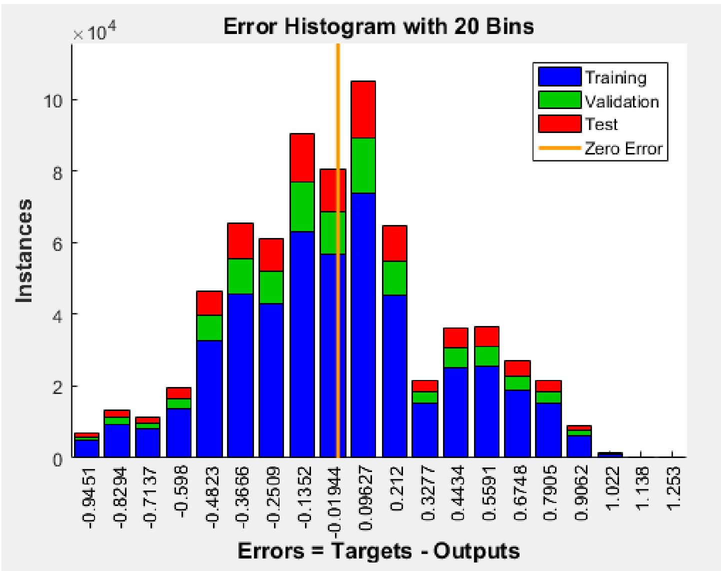


Figure 38. Error histogram for ANN with 3 hidden neurons

Results for ANN with 4 hidden neurons

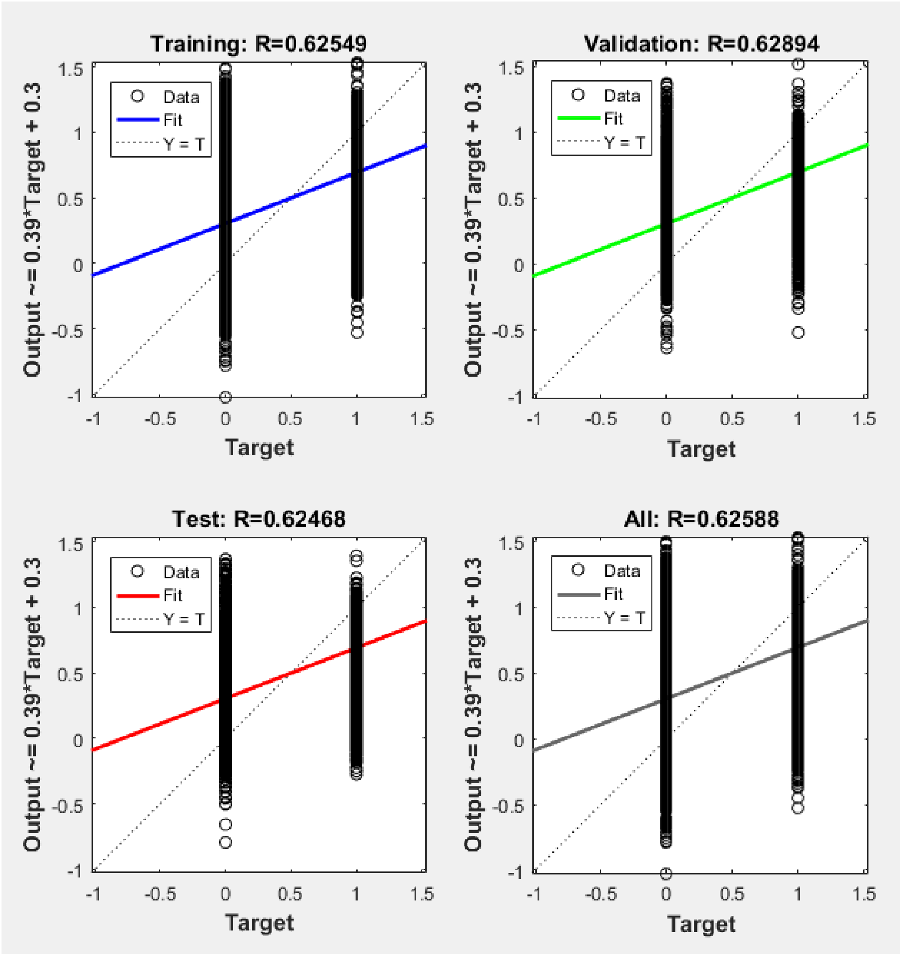


Figure 39. Regression plot for ANN with 4 hidden neurons

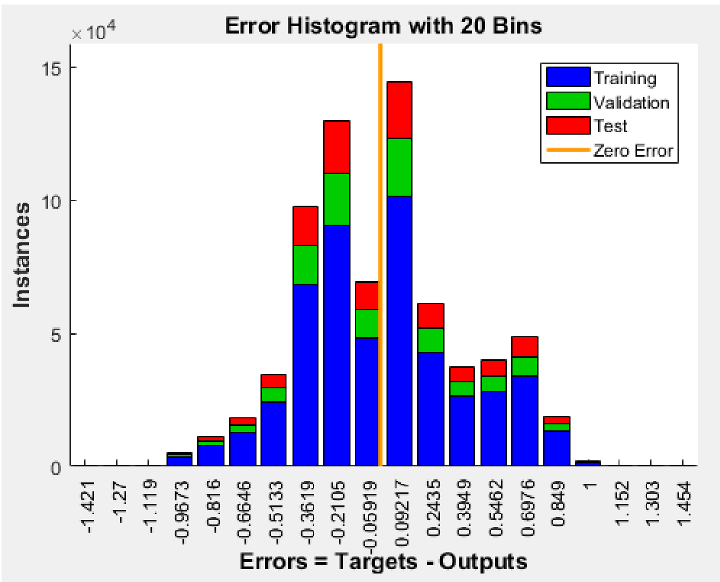


Figure 40. Error histogram for ANN with 4 hidden neurons

Results for ANN with 5 hidden neurons

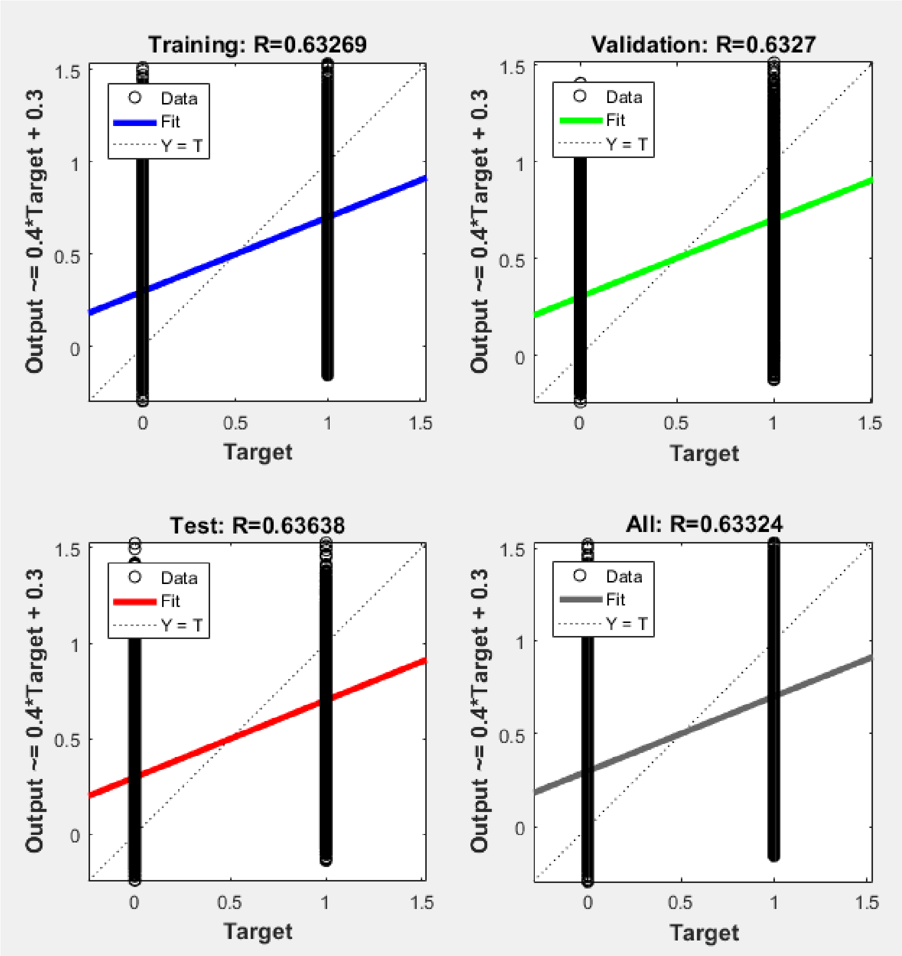


Figure 41. Regression plot for ANN with 5 hidden neurons

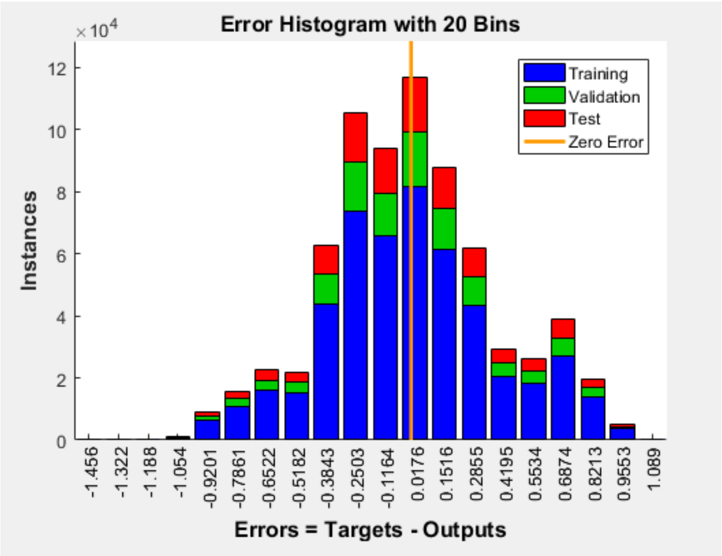


Figure 42. Error histogram for ANN with 5 hidden neurons

Results for ANN with 6 hidden neurons

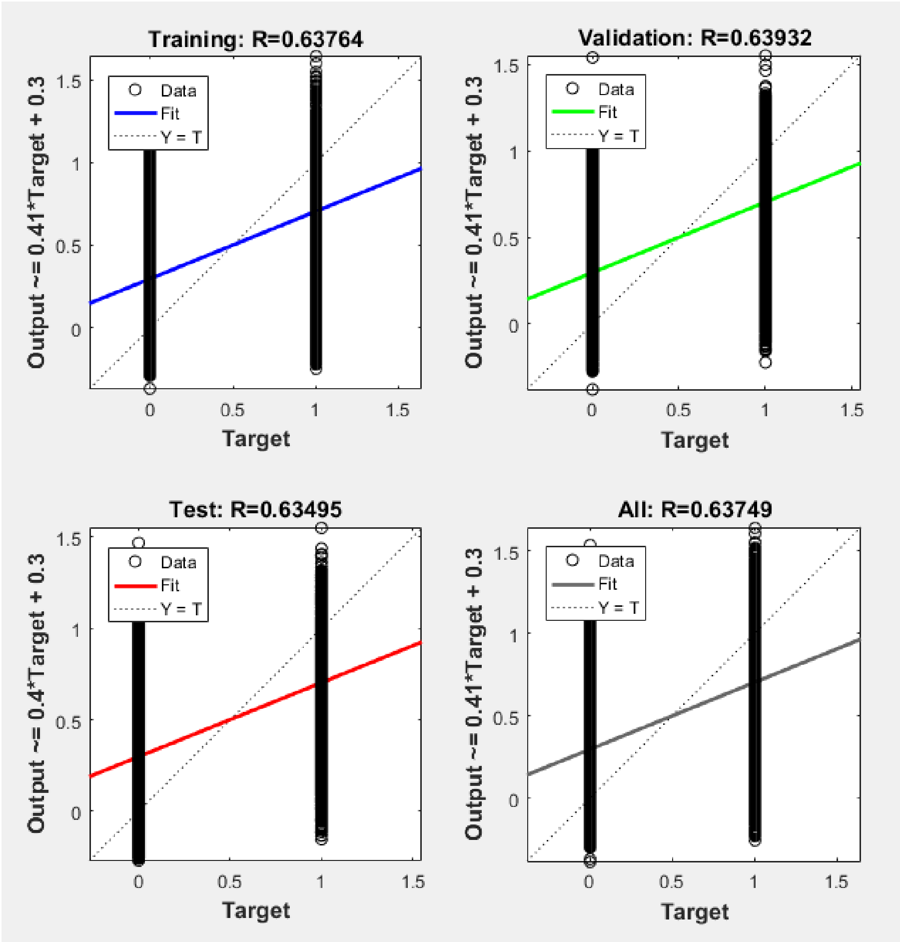


Figure 43. Regression plot for ANN with 6 hidden neurons

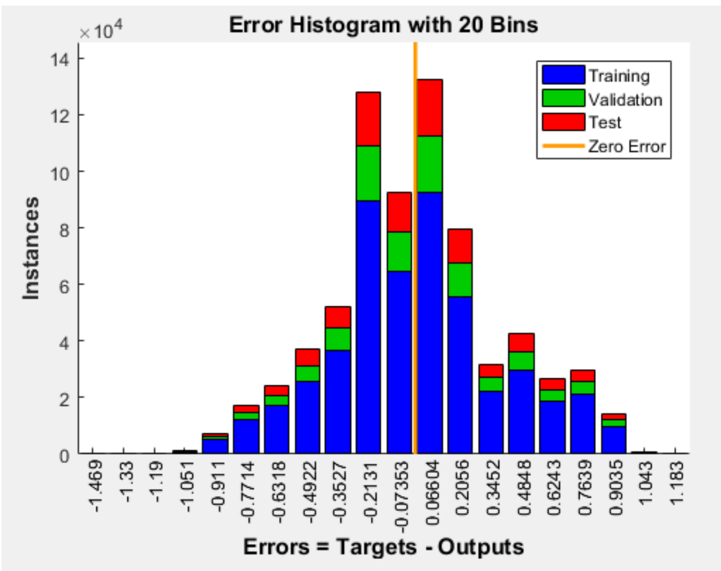


Figure 44. Error histogram for ANN with 6 hidden neurons

Results for ANN with 7 hidden neurons

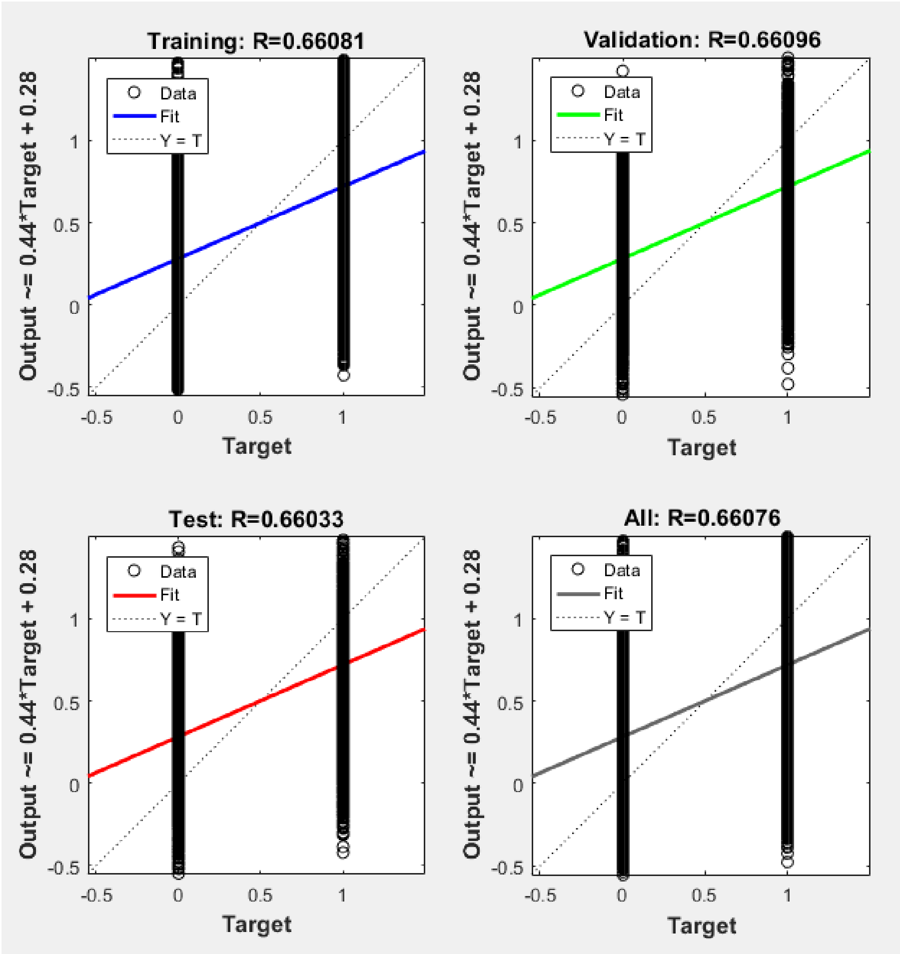


Figure 45. Regression plot for ANN with 7 hidden neurons

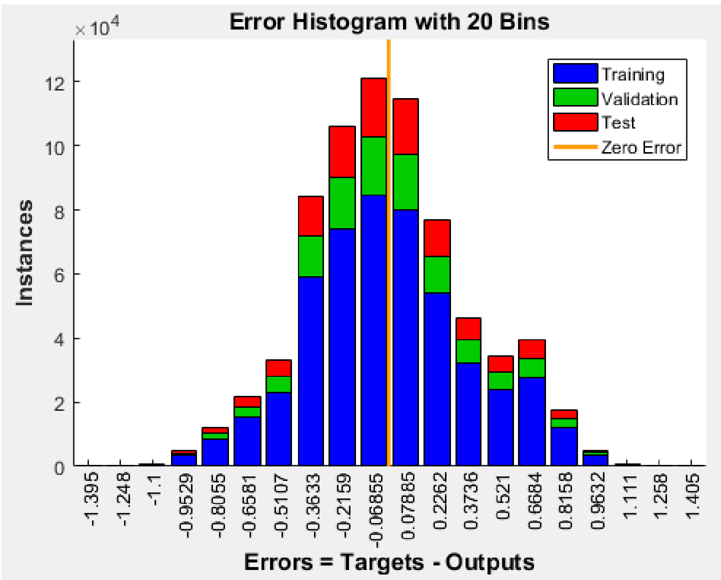


Figure 46. Error histogram for ANN with 7 hidden neurons

Results for ANN with 8 hidden neurons

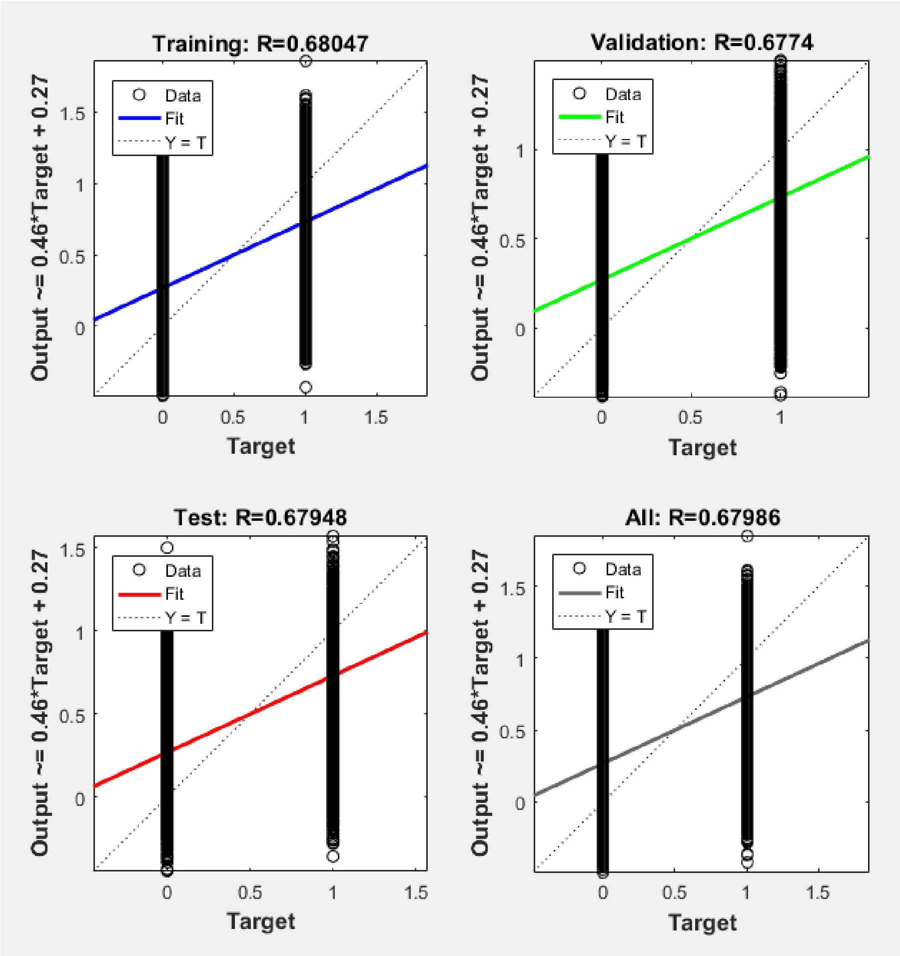


Figure 47. Regression plot for ANN with 8 hidden neurons

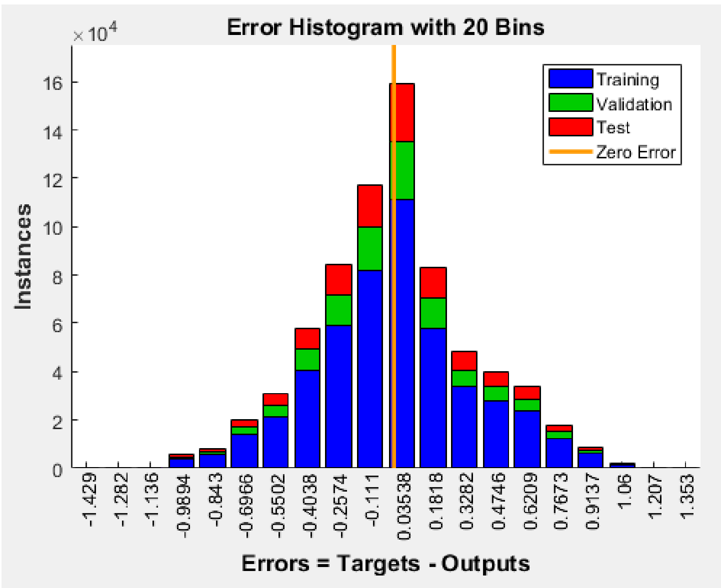


Figure 48. Error histogram for ANN with 8 hidden neurons

Results for ANN with 9 hidden neurons

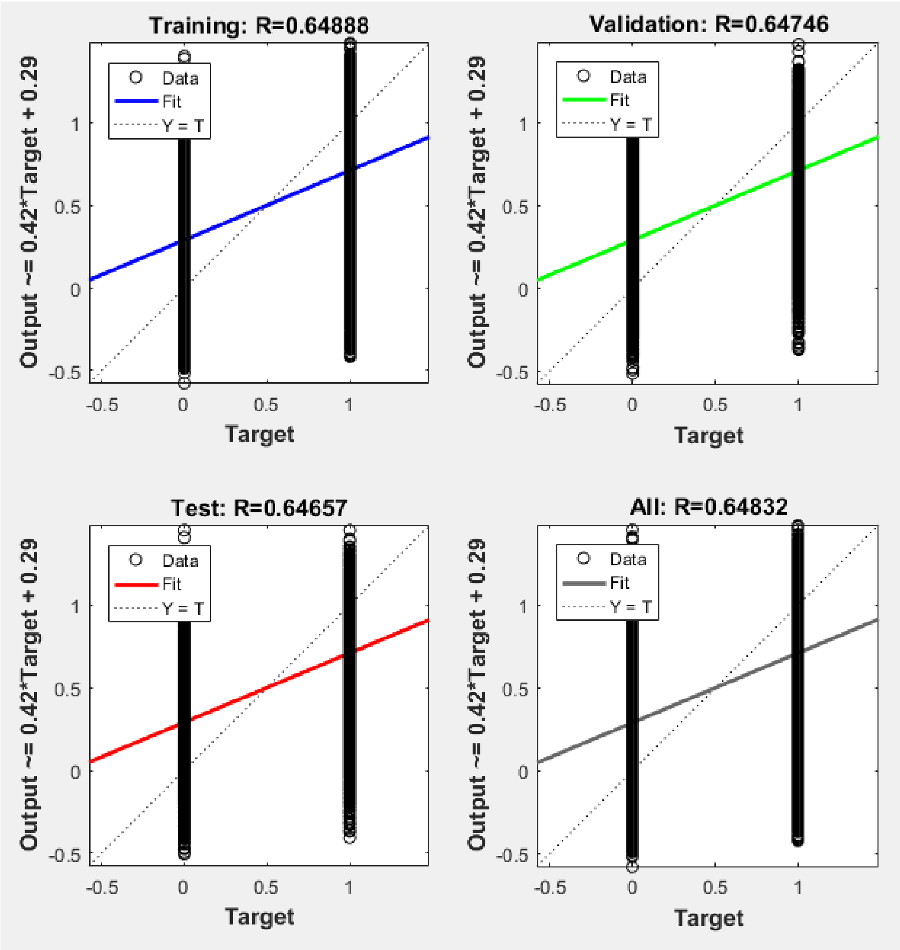


Figure 49. Regression plot for ANN with 9 hidden neurons

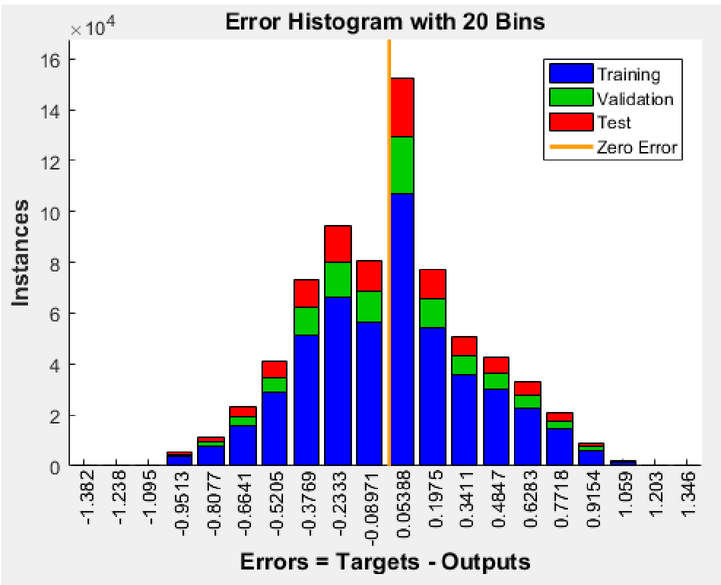


Figure 50. Error histogram for ANN with 9 hidden neurons

Results for ANN with 10 hidden neurons

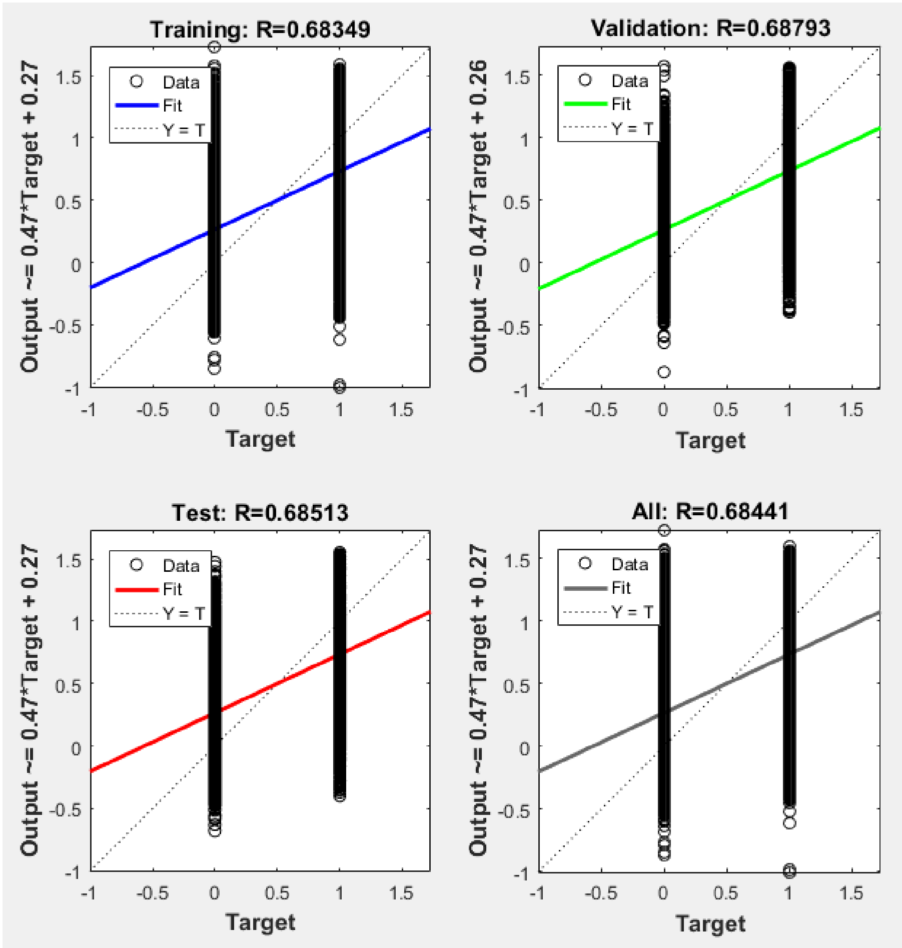


Figure 51. Regression plot for ANN with 10 hidden neurons

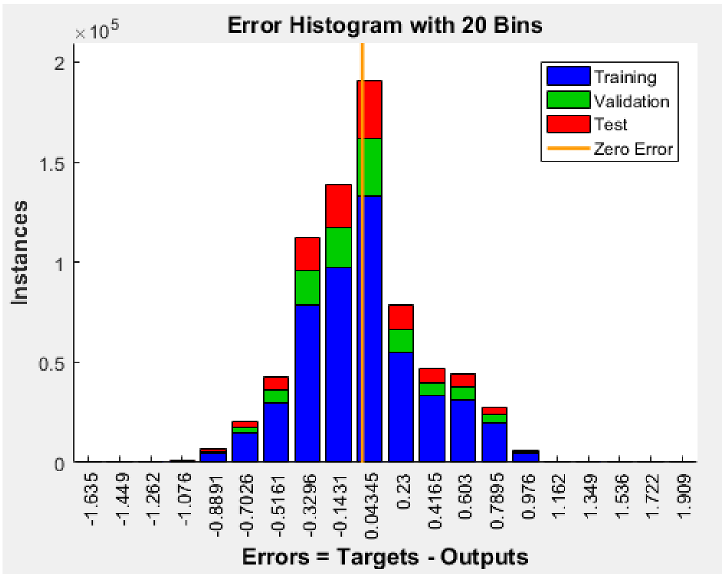


Figure 52. Error histogram for ANN with 10 hidden neurons

Results for ANN with 11 hidden neurons

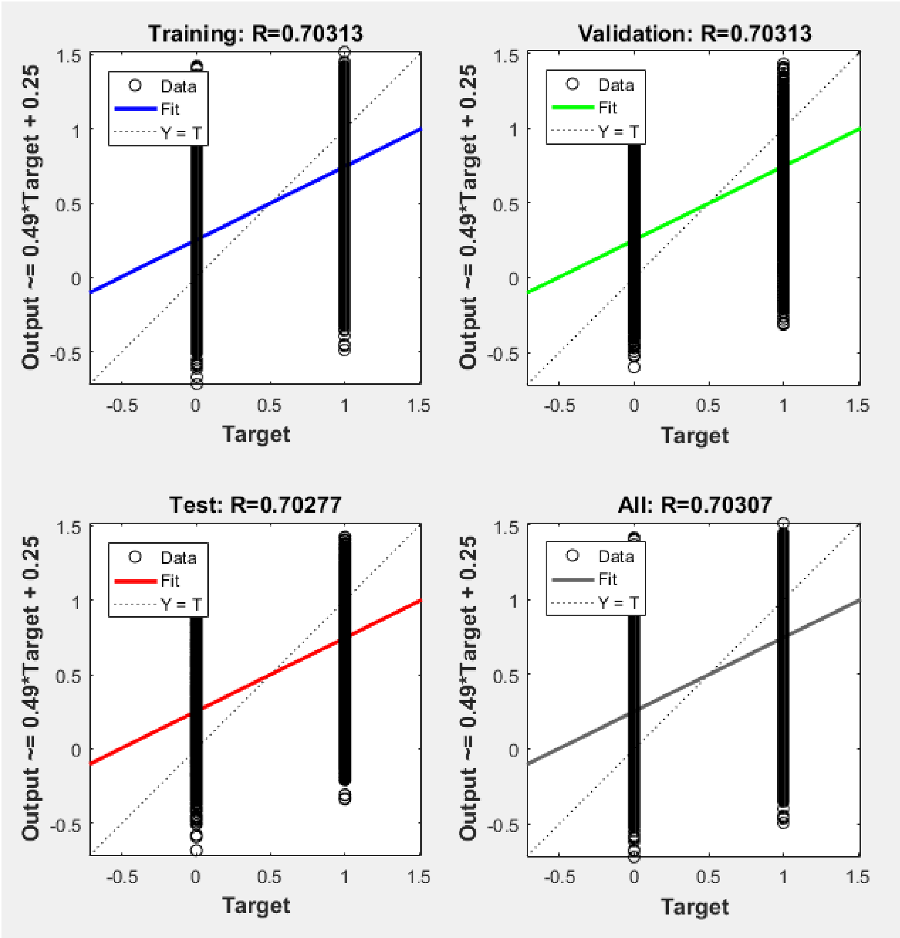


Figure 53. Regression plot for ANN with 11 hidden neurons

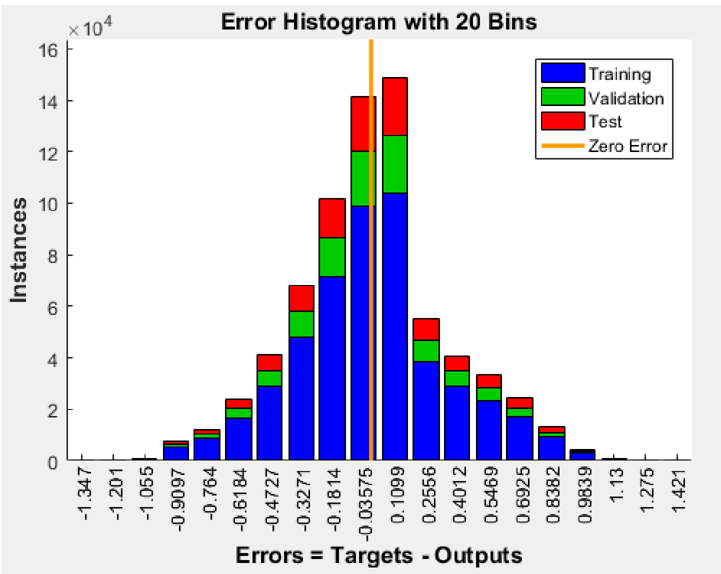


Figure 54. Error histogram for ANN with 11 hidden neurons

Results for ANN with 12 hidden neurons

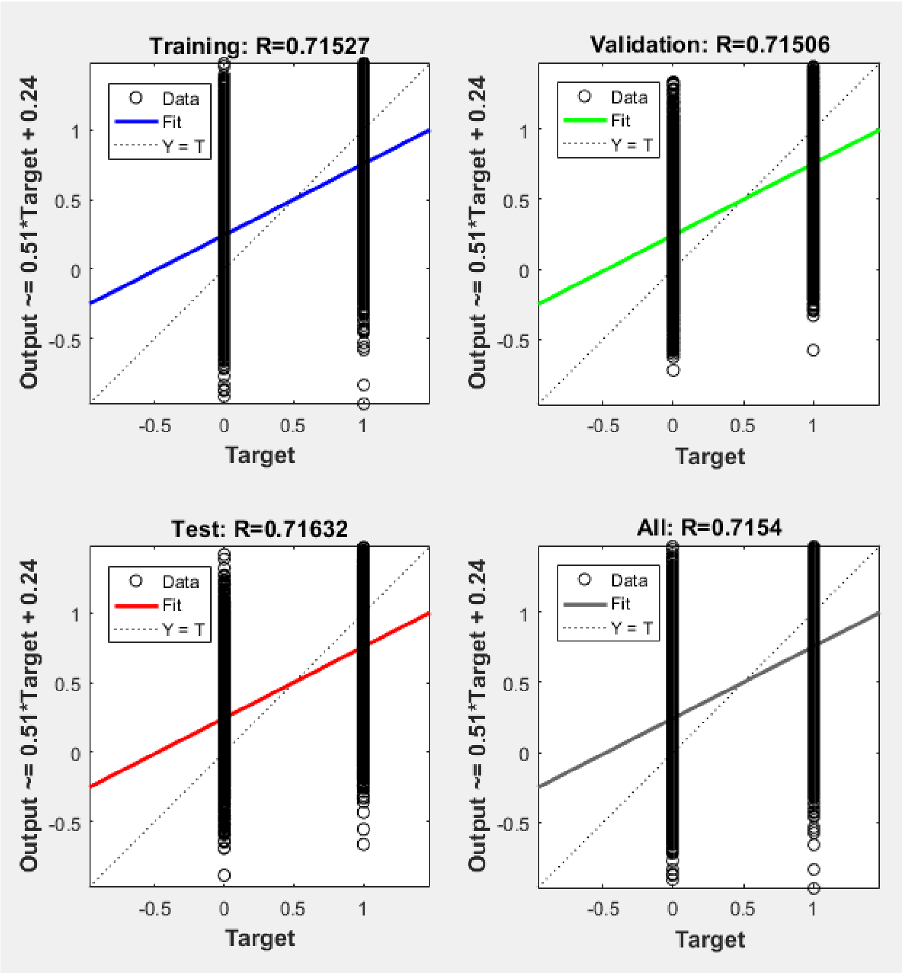


Figure 55. Regression plot for ANN with 12 hidden neurons

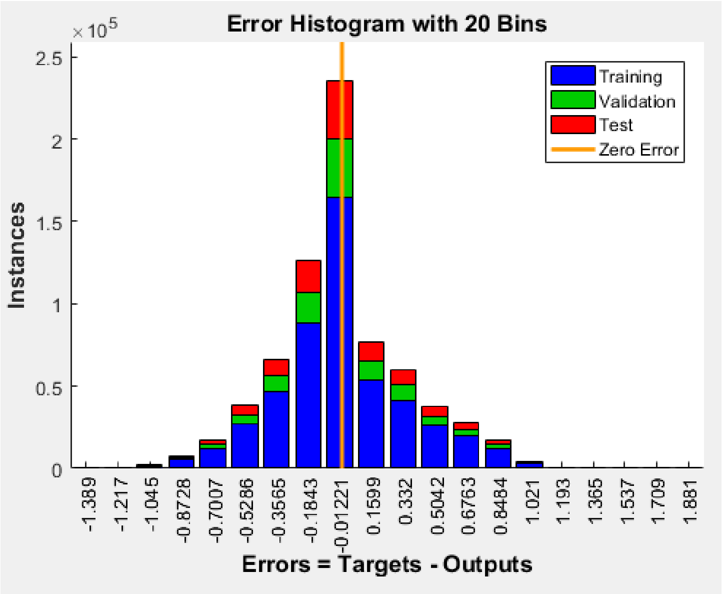


Figure 56. Error histogram for ANN with 12 hidden neurons

Results for ANN with 13 hidden neurons

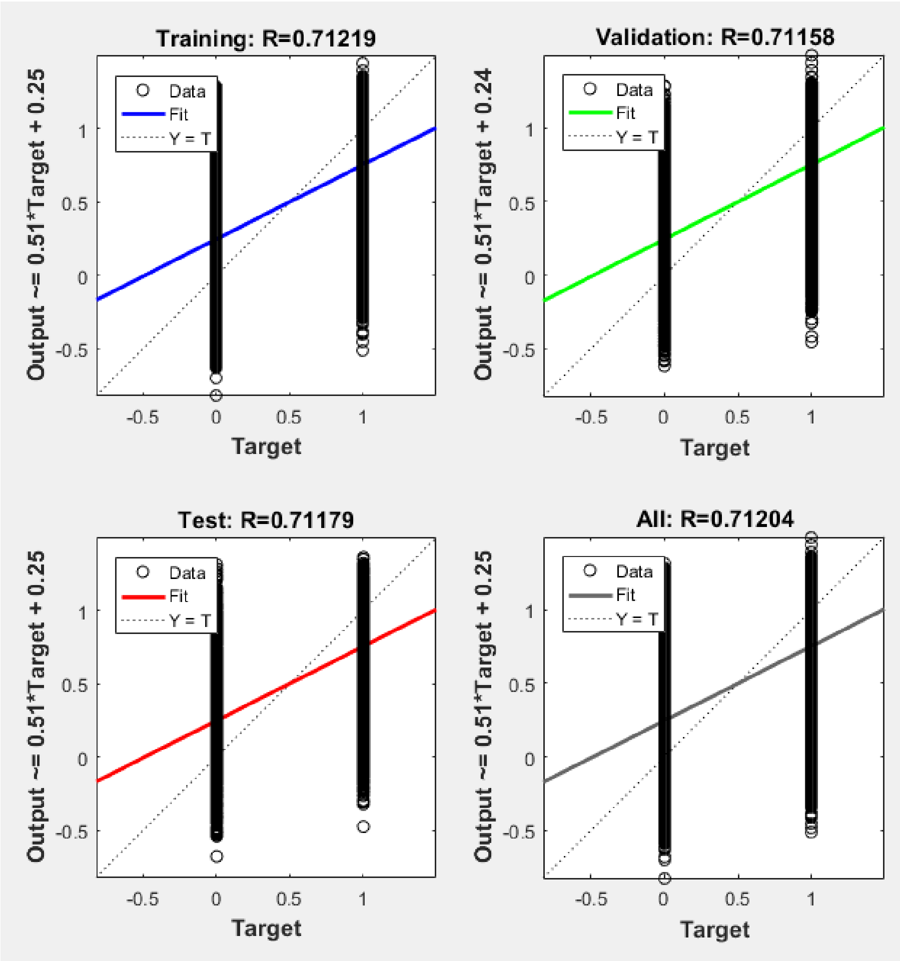


Figure 57. Regression plot for ANN with 13 hidden neurons

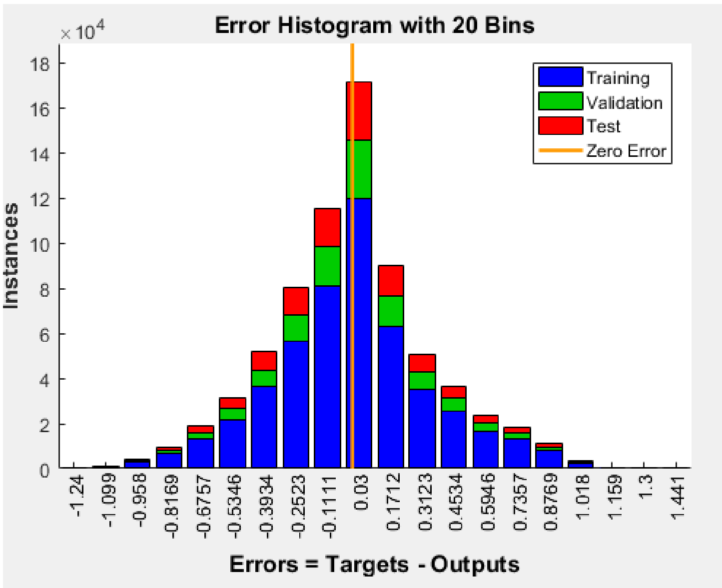


Figure 58. Error histogram for ANN with 13 hidden neurons

Results for ANN with 14 hidden neurons

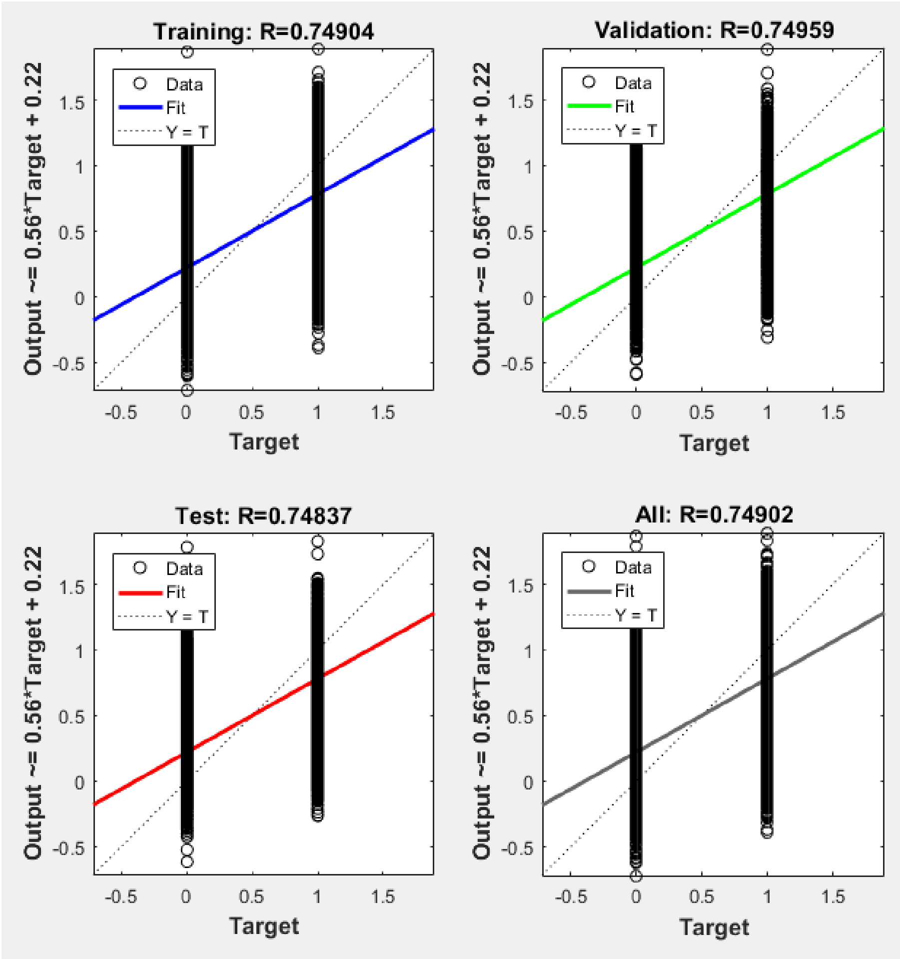


Figure 59. Regression plot for ANN with 14 hidden neurons

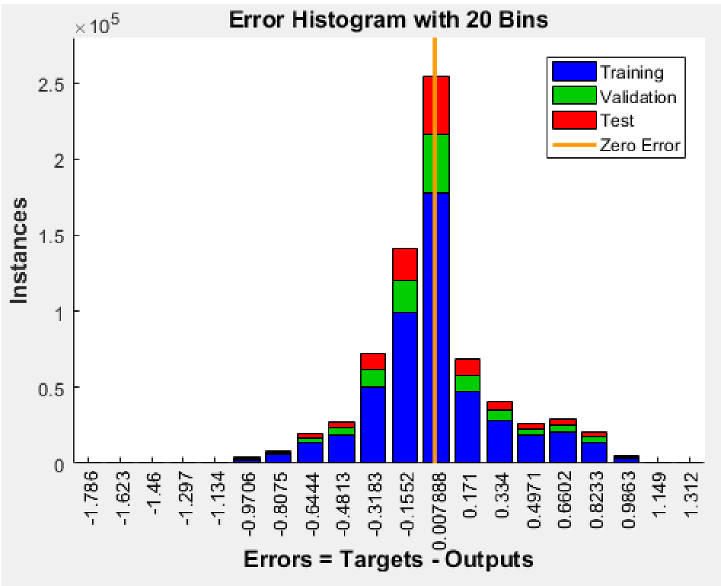


Figure 60. Error histogram for ANN with 14 hidden neurons

Results for ANN with 15 hidden neurons

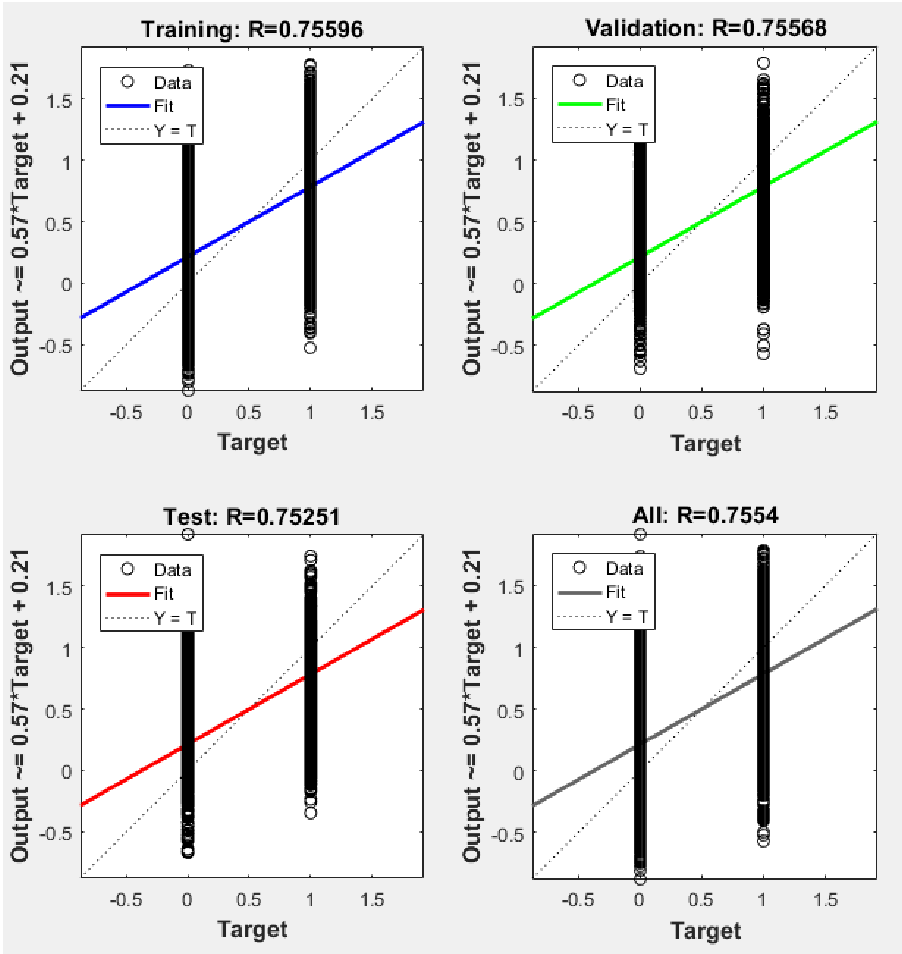


Figure 61. Regression plot for ANN with 15 hidden neurons

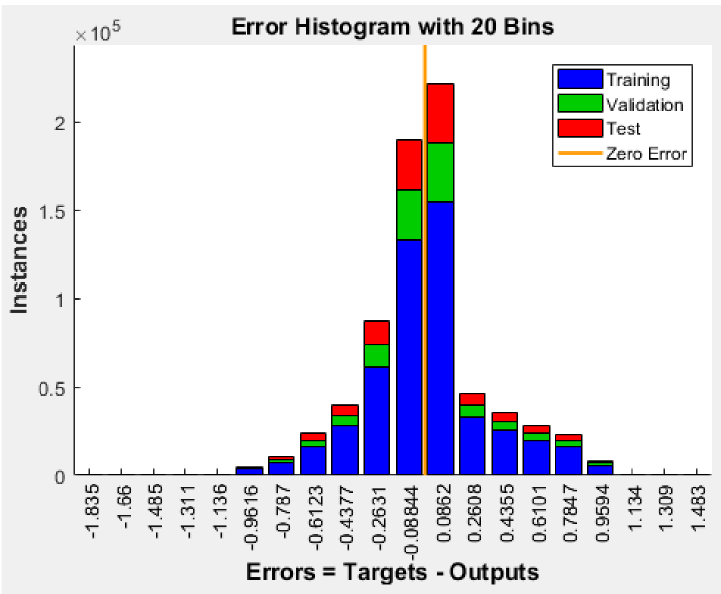


Figure 62. Error histogram for ANN with 15 hidden neurons

Results for ANN with 16 hidden neurons

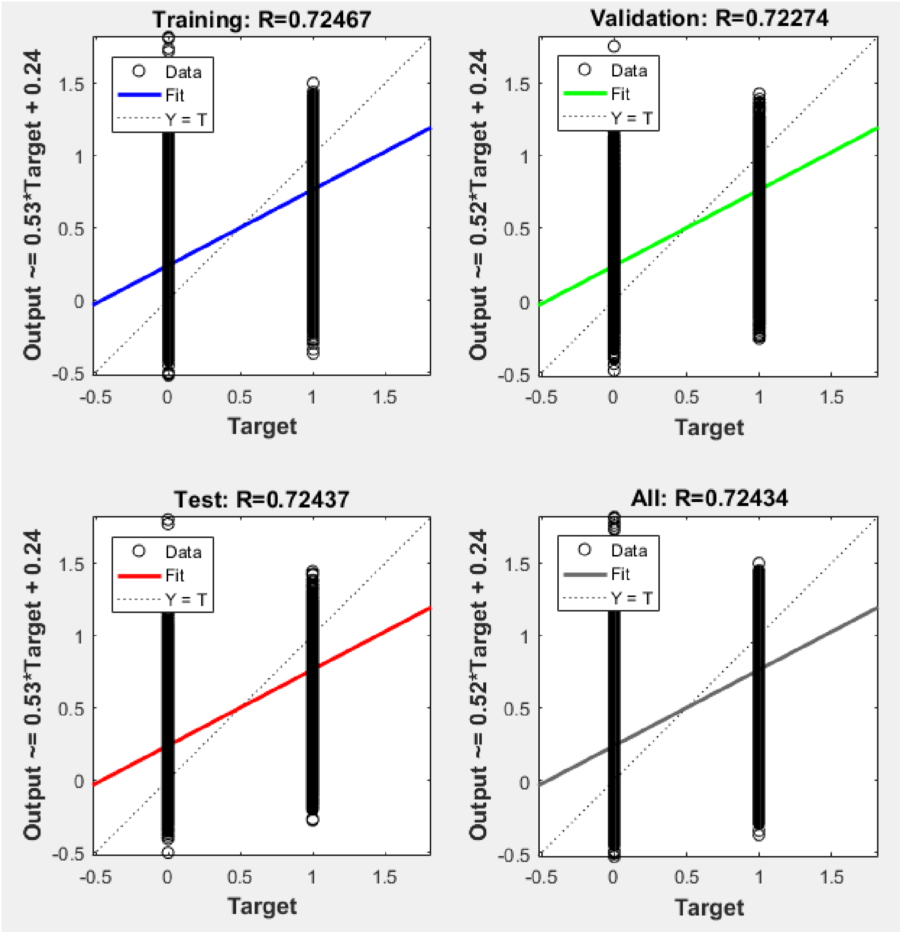


Figure 63. Regression plot for ANN with 16 hidden neurons

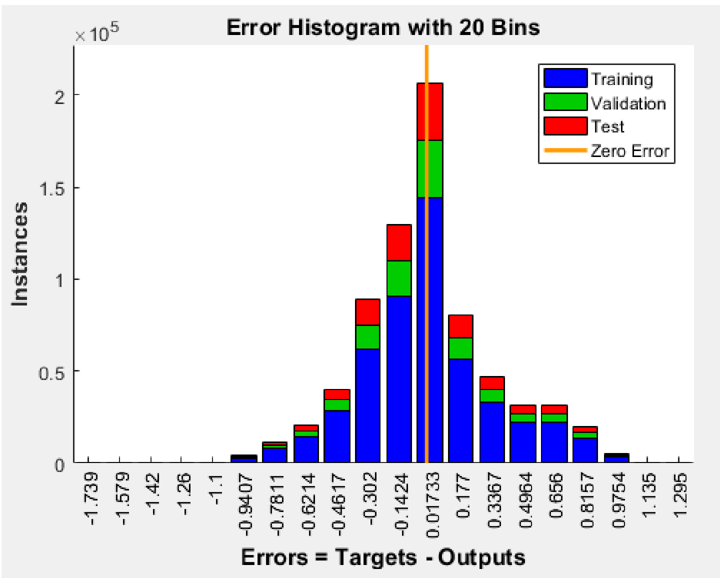


Figure 64. Error histogram for ANN with 16 hidden neurons

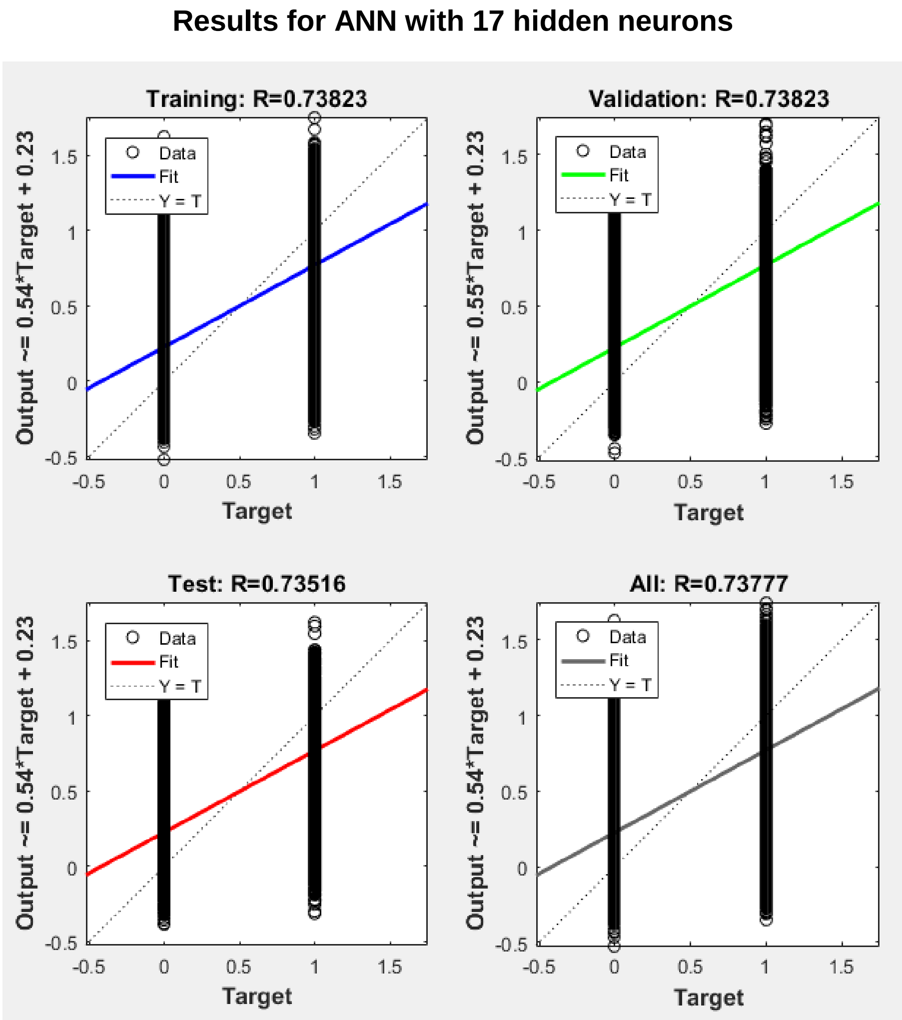


Figure 65. Regression plot for ANN with 17 hidden neurons

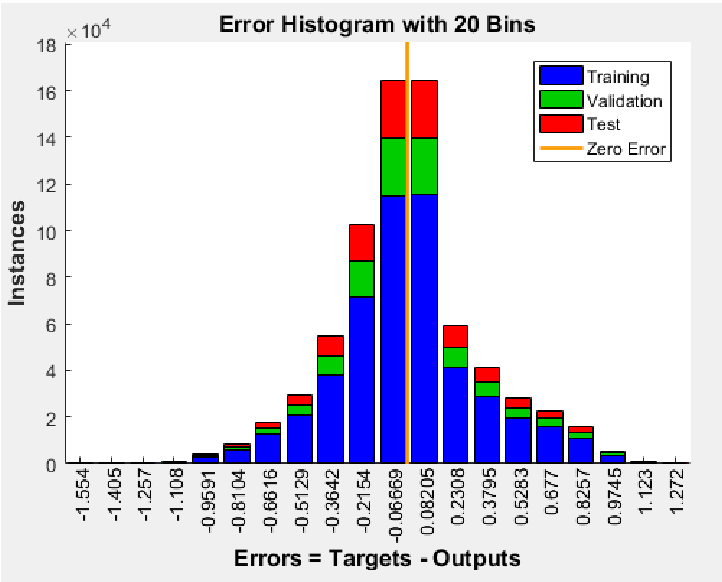


Figure 66. Error histogram for ANN with 17 hidden neurons

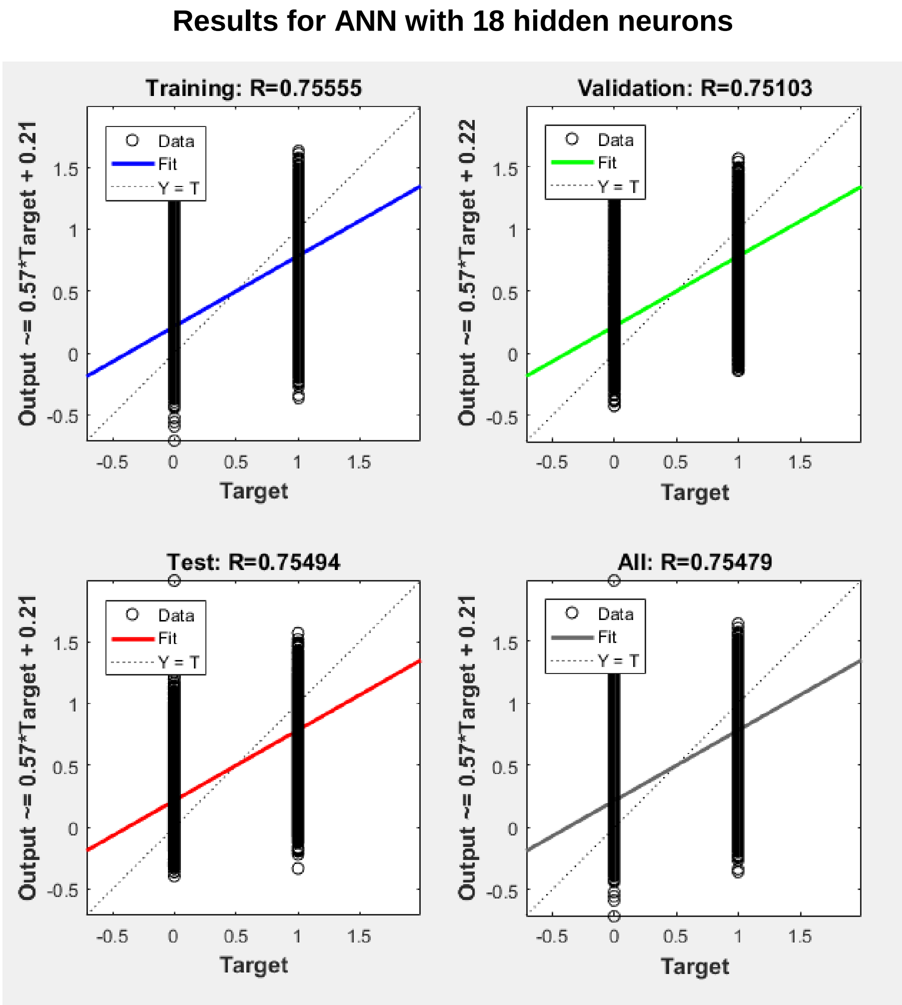


Figure 67. Regression plot for ANN with 18 hidden neurons

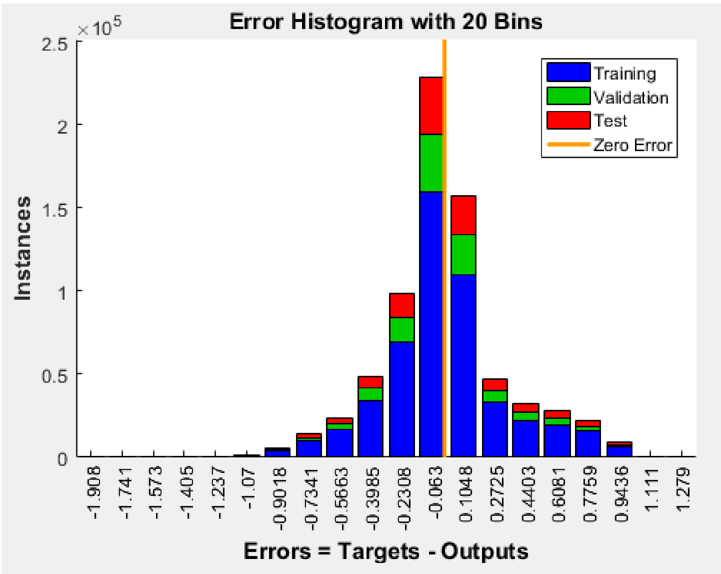


Figure 68. Error histogram for ANN with 18 hidden neurons

Results for ANN with 19 hidden neurons

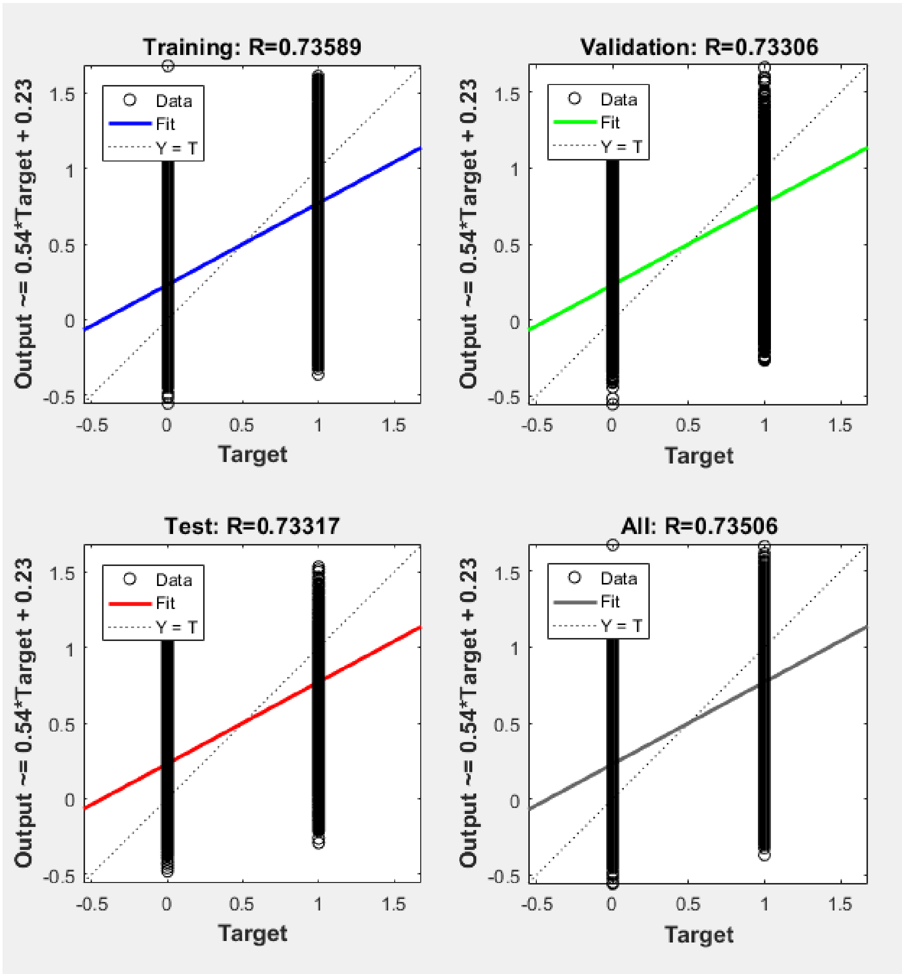


Figure 69. Regression plot for ANN with 19 hidden neurons

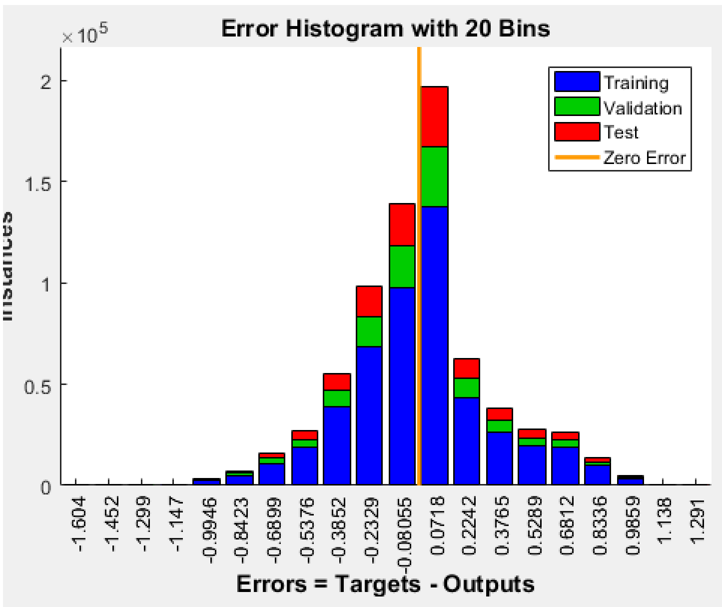


Figure 70. Error histogram for ANN with 19 hidden neurons

Results for ANN with 20 hidden neurons

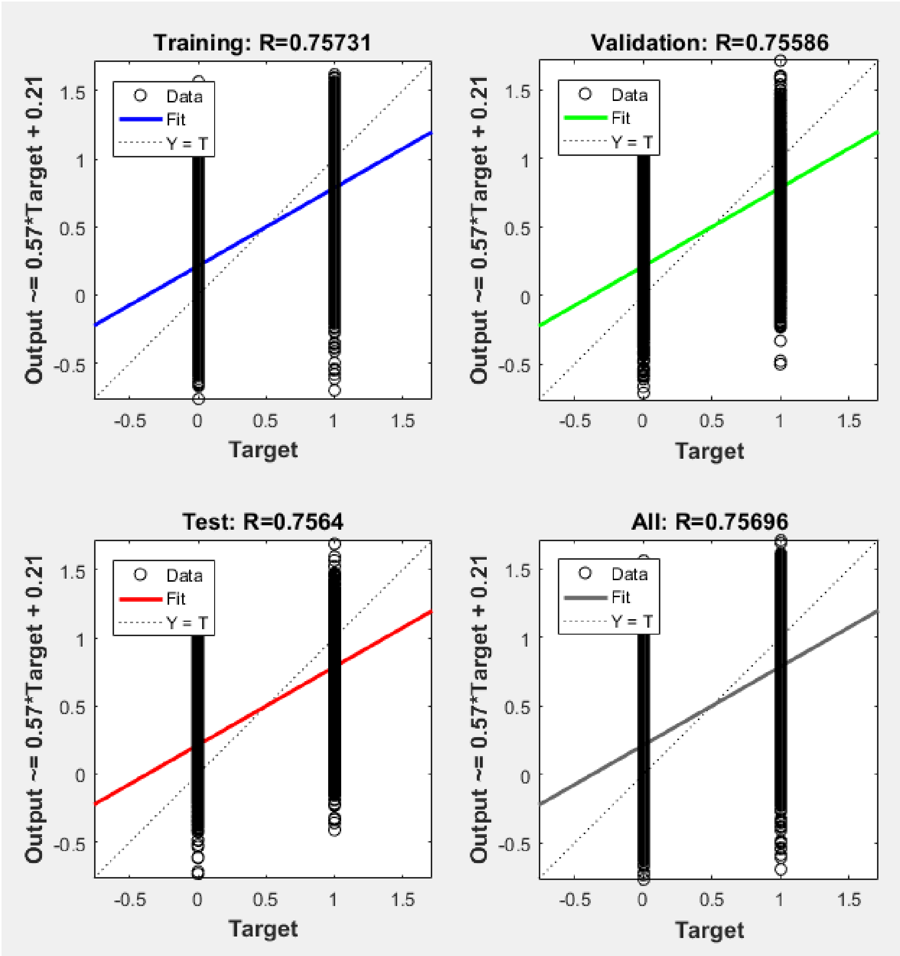


Figure 71. Regresion plot for ANN with 20 hidden neurons

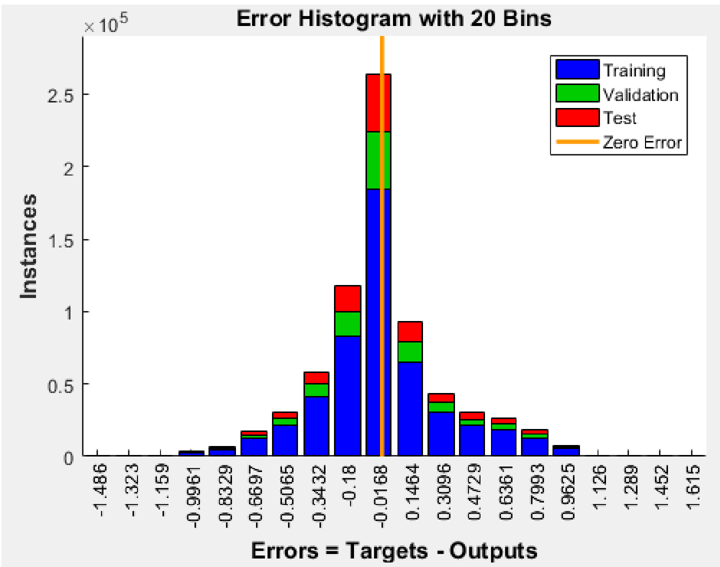


Figure 72. Error histogram for ANN with 20 hidden neurons

Results for ANN with 21 hidden neurons

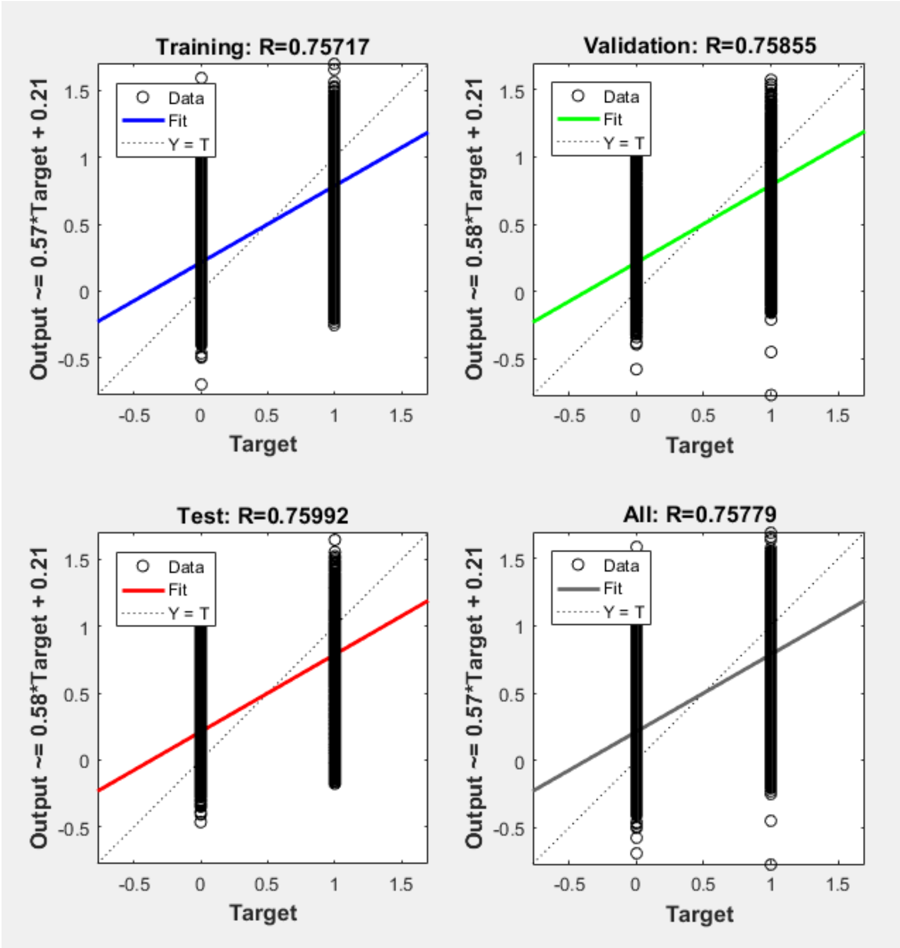


Figure 73. Regression plot for ANN with 21 hidden neurons

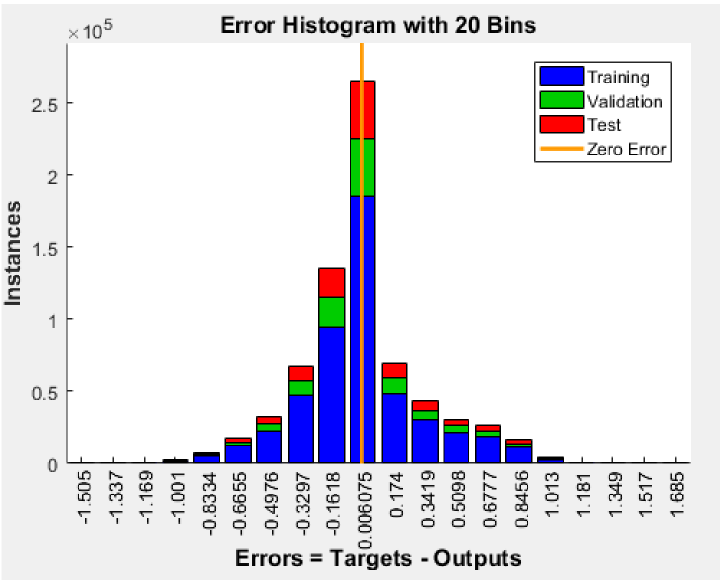


Figure 74. Error histogram for ANN with 21 hidden neurons

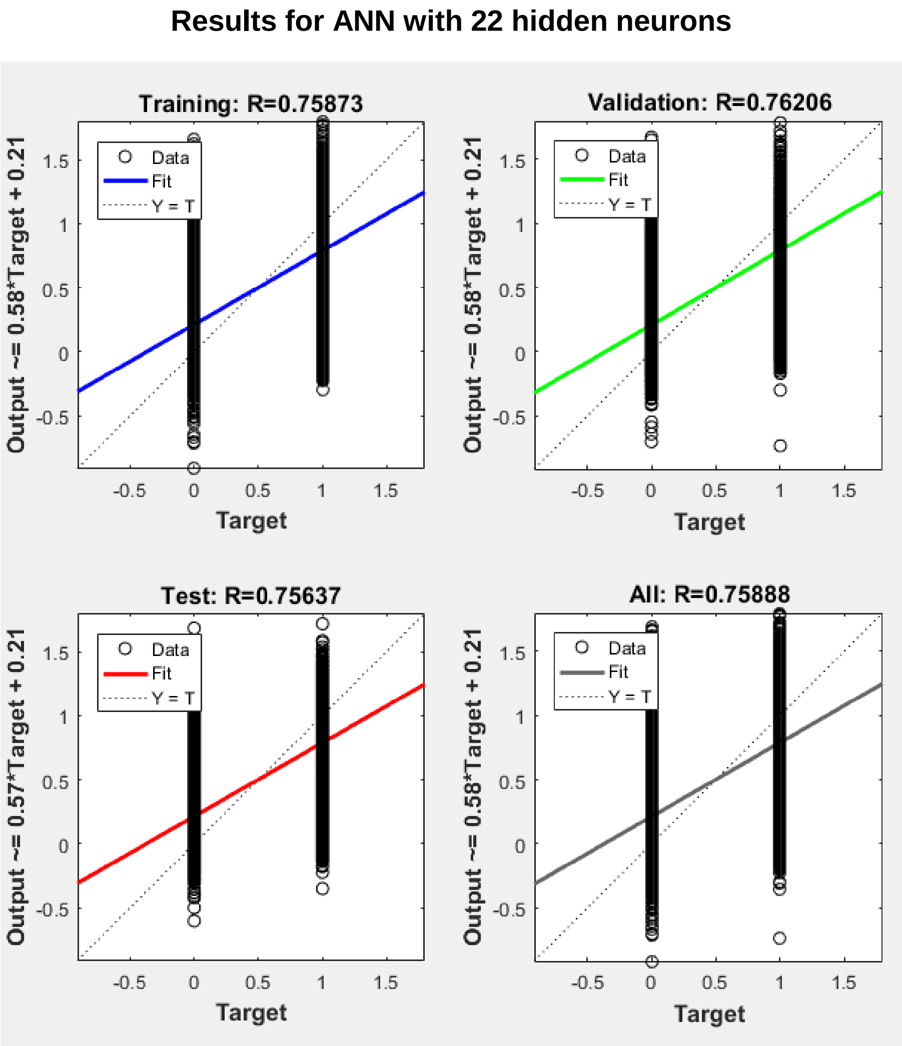


Figure 75. Regression plot for ANN with 22 hidden neurons

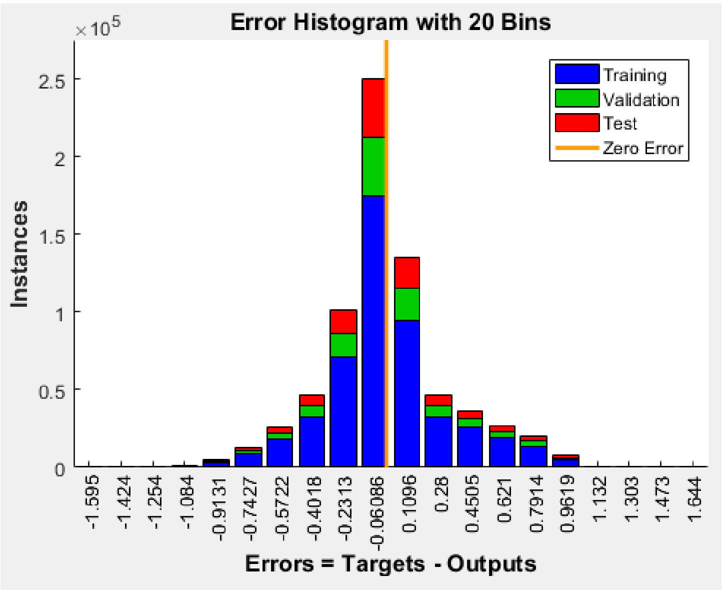


Figure 76. Error histogram for ANN with 22 hidden neurons

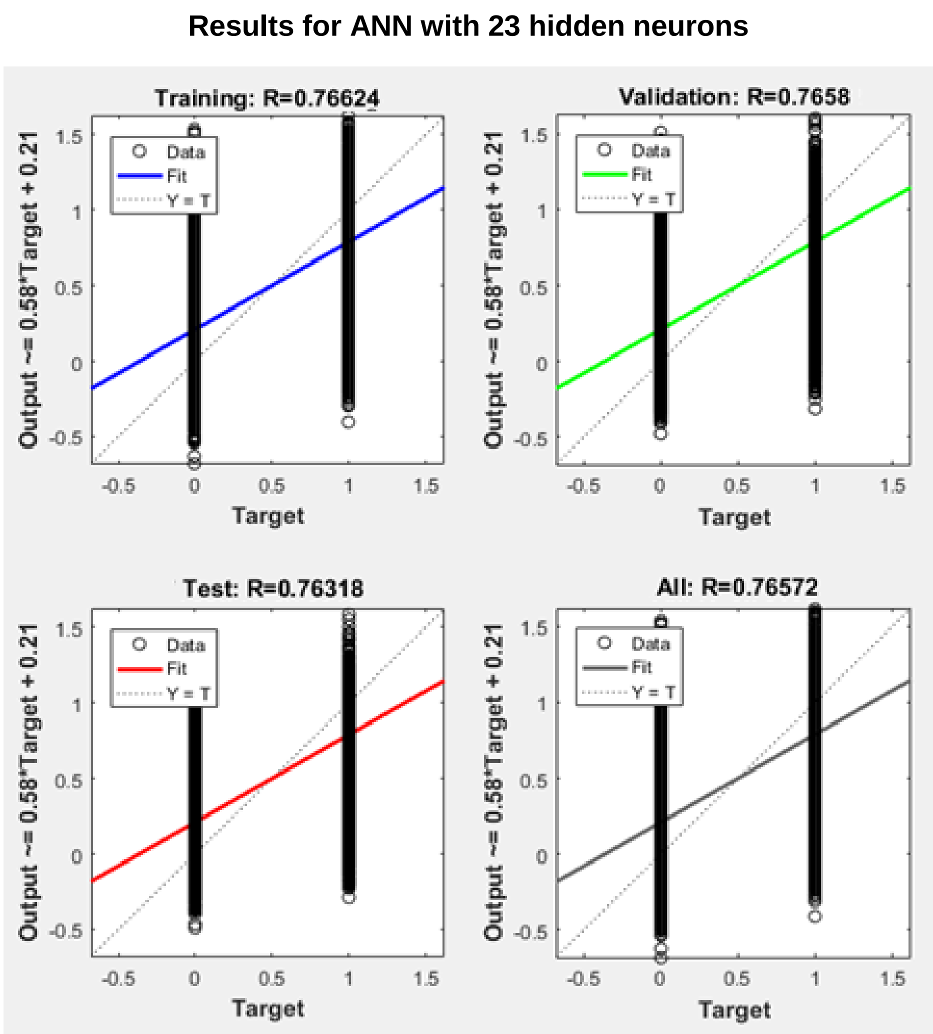


Figure 77. Regression plot for ANN with 23 hidden neurons

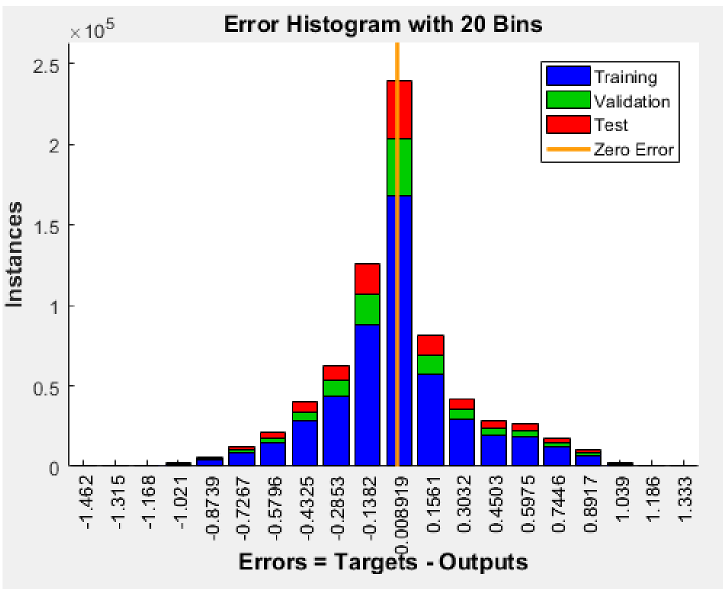


Figure 78. Error histogram for ANN with 23 hidden neurons

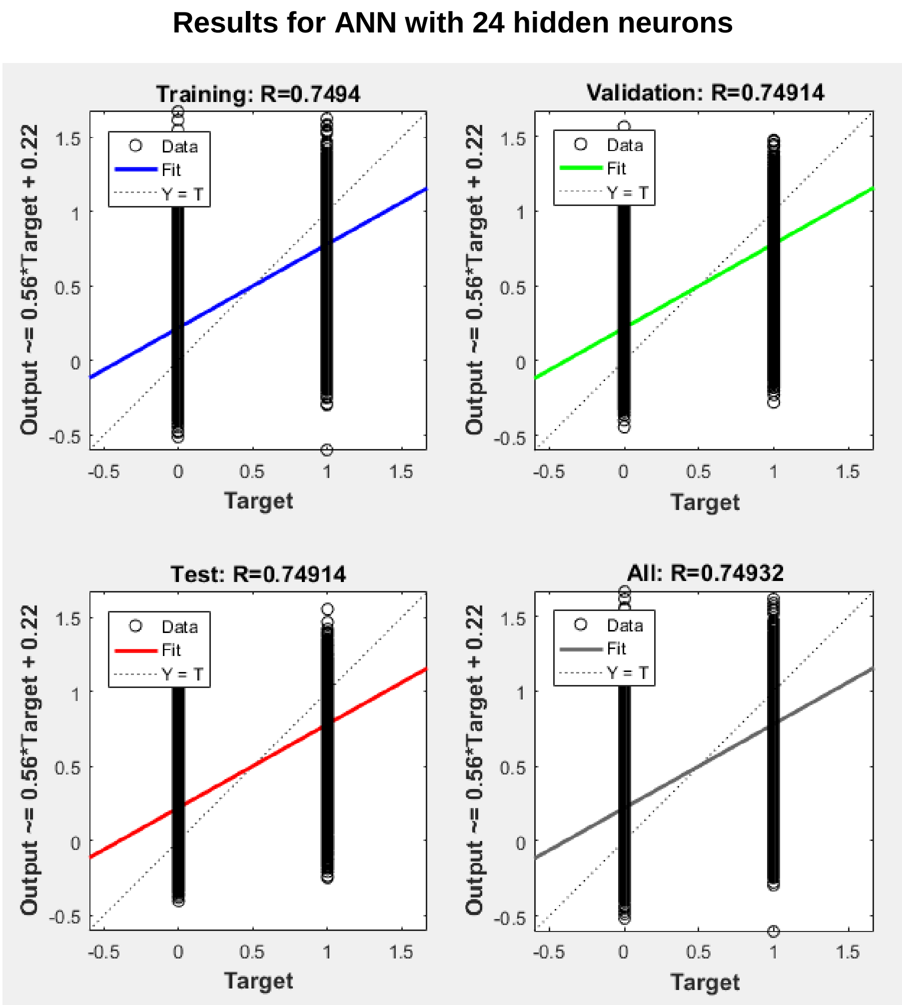


Figure 79. Regression plot for ANN with 24 hidden neurons

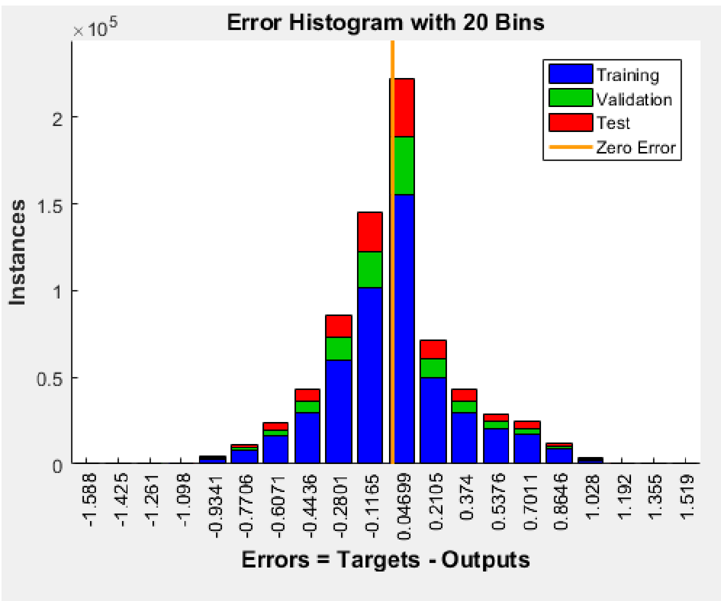


Figure 80. Error histogram for ANN with 24 hidden neurons

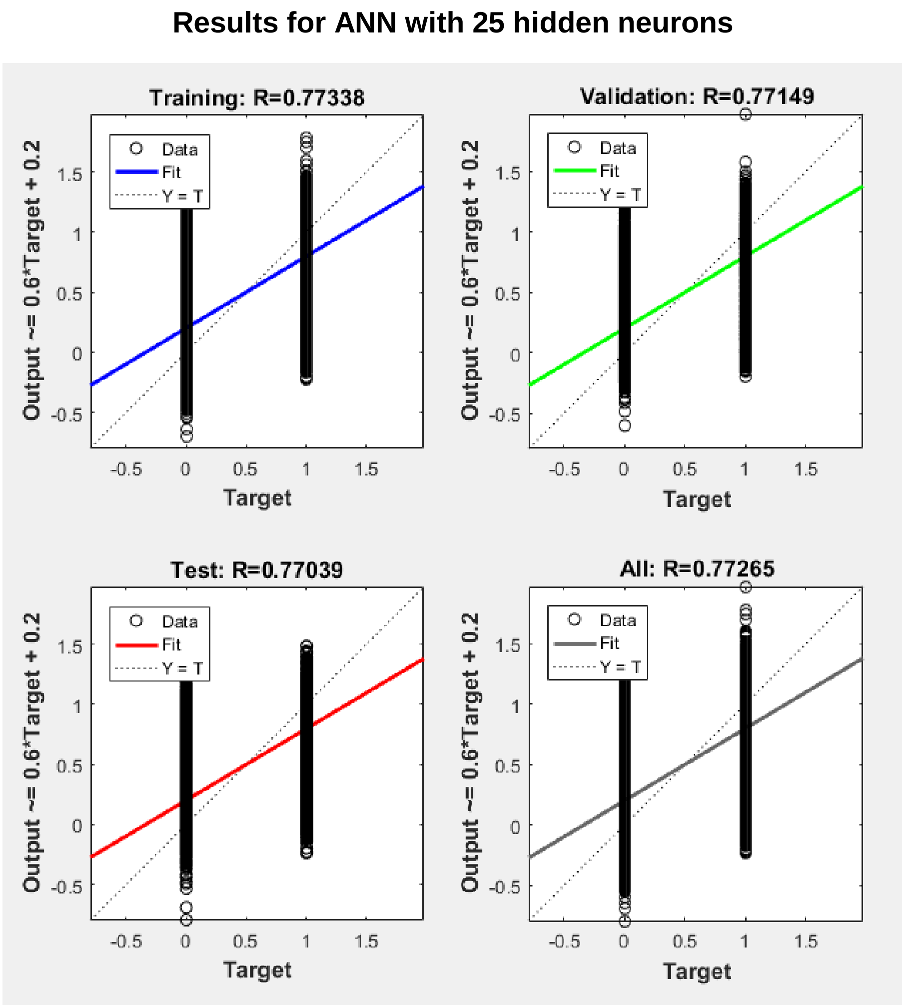


Figure 81. Regression plot for ANN with 25 hidden neurons

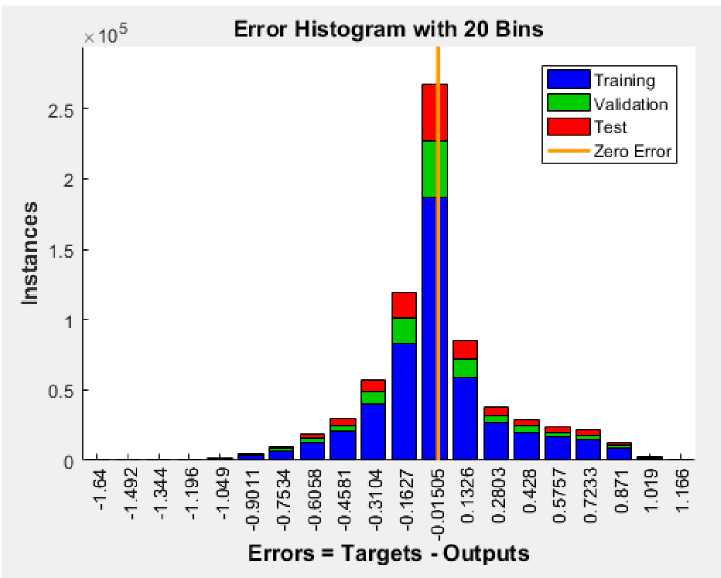


Figure 82. Error histogram for ANN with 25 hidden neurons

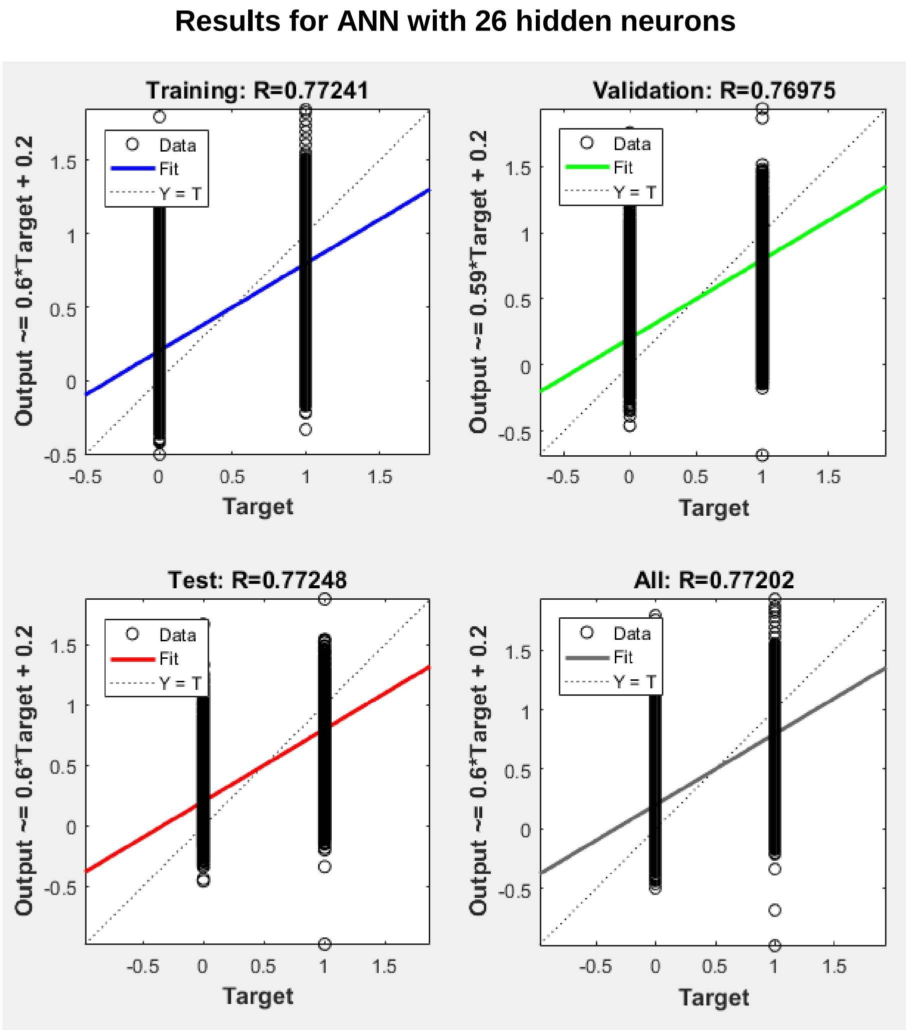


Figure 83. Regression plot for ANN with 26 hidden neurons

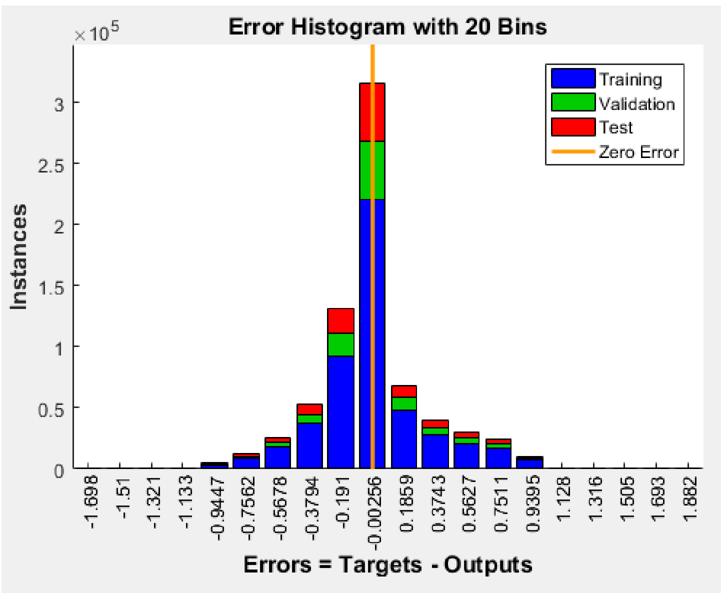


Figure 84. Error histogram for ANN with 26 hidden neurons

Results for ANN with 27 hidden neurons

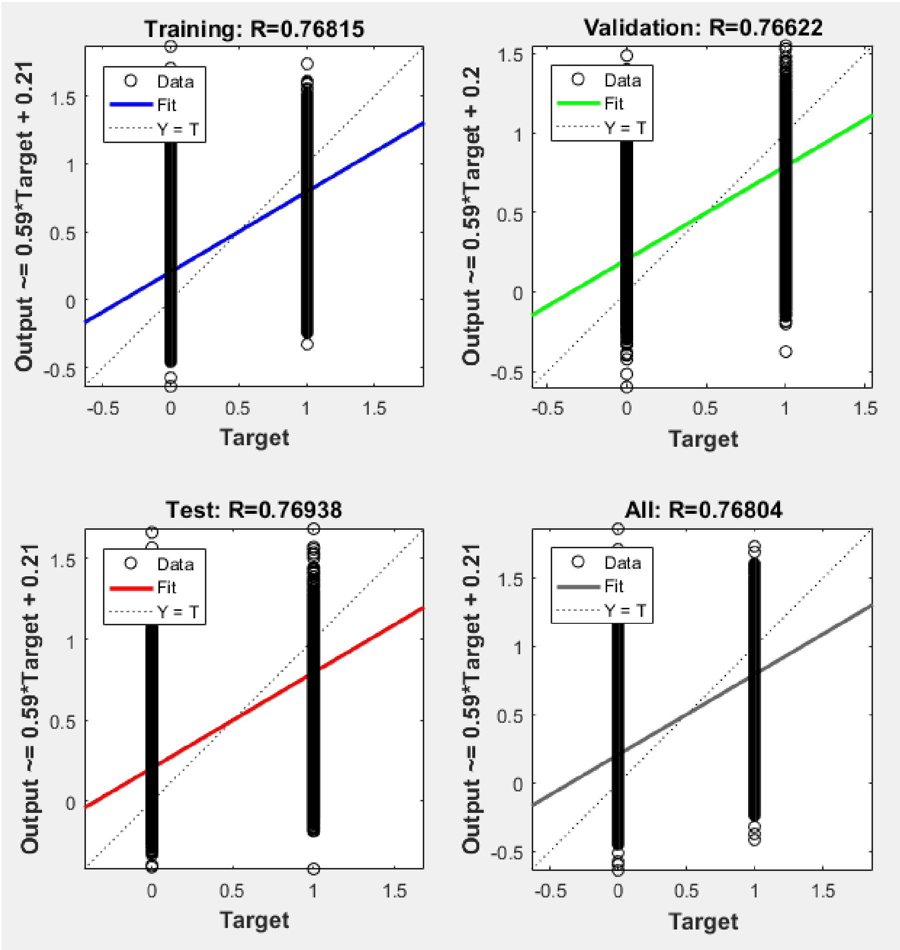


Figure 85. Regression plot for ANN with 27 hidden neurons

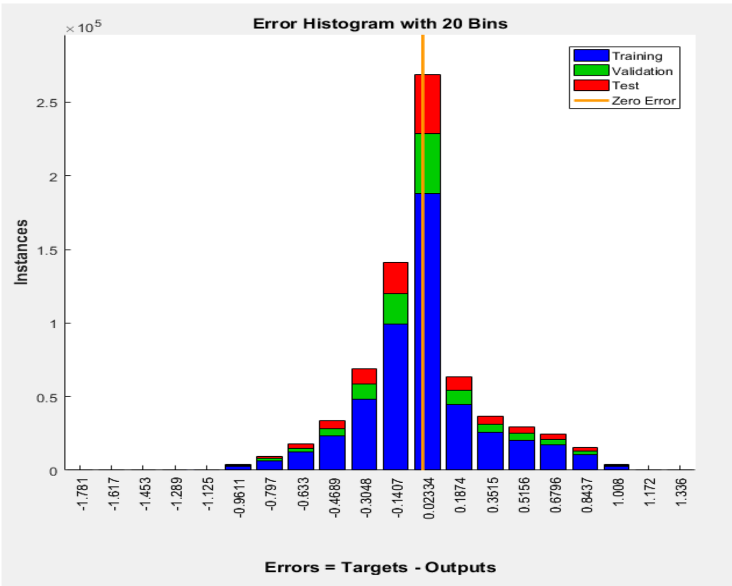


Figure 86. Error histogram for ANN with 27 hidden neurons