Managing Intellectual Capital
and Innovation for Sustainable
and Inclusive Society
27–29 May 2015
Bari • Italy

make learn TIIM

Management,
Knowledge and Learning
Joint International Conference 2015
Technology, Innovation
and Industrial Management

CALCULATING THE DEPENDENCY OF COMPONENTS OF OBSERVABLE NONLINEAR SYSTEMS USING ARTIFICIAL NEURAL NETWORKS

Wolfram Rinke
University of Applied Science Burgenland, Austria
wolfram.rinke@fh-burgenland.at

Abstract:

A method for computing dependency factors between independent and dependent components of an observable system is presented. The dependency factors are derived from the analysis of an artificial neural network, which models the system observed. The generic approach applies already proven methodologies to calculate the input-output sensitivity of multilayer feedforward artificial neural networks with differentiable activation functions. The algorithm and the mathematics to compute the dependency factors are presented. Different examples from agriculture and marketing illustrate how this method can be applied to explain an observable system.

Keywords: innovation, dependency factors, sensitivity analysis, Jacobian matrix, artificial neural networks, artificial intelligence, information technology

1. INTRODUCTION

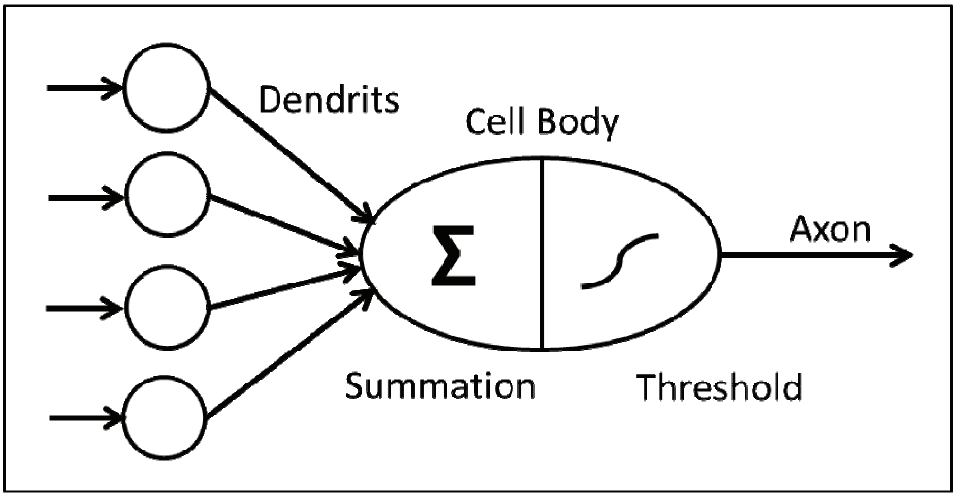
Over the last decades artificial neural networks (ANN) have emerged as an established powerfool tool in a broad range of engineering and scientific applications especially for process modelling and control. ANNs are also well known and widely used for data mining tasks or used for generic nonlinear function mapping applications. To find a function to map data set A to data set B can be done with different mathematical algorithms, ANNs are proved to be the best algorithms for nonlinear unknown relationships between data set A and data set B. Unfortunatly it is very difficult to get a simplified or quantified description of an artificial neural network to better understand the internal relationsships. This hurdle prevents non technical researchers from using ANN algorithms for their research work and forces them to rely on classical statistical methods to describe their observations. In many applications the input-output analyses or cause-effect relationships is the core topic. To know about the characteristic and the interrelationships within an observable system is one of the primary issues in many research areas. Traditional statistical methods are widely used in economic sciences or social sciences. Models are used to describe systems behaviour, but these are often too simplified or linearized to describe the system observed. Worth mentioning is the fact that neither the system dynamics is known, nor are the interrelationships quantified. Many empirical studies lack profound quantification methods, to describe the results or they use linear methods to describe nonlinear systems with poor results.

One of the popular algorithms to build artificial neural network models is the algorithm for building and training an artificial neural network based on feedforward multilayer perceptrons. It is an excellent technology and has been proven as universal approximators. The error backpropagation learning algorithm presented by Rumelhart (Rumelhart, 1986) is typically used for network learning. Cybenko (Cybenko, 1989) and Hornik (Hornik, 1989) proved that any continuous mapping over a compact domain could be approximated as accurately as necessary by a feedforward artificial neural network with one hidden layer and differentiable activation function. These findings make the ANN technology so powerfull and generic, because it makes it possible to find a proper mapping function even for difficult or impossible to describe systems, because not all involved parameters or disturbances of the observed system are known or could be measured.

2. THEORETICAL BACKGROUND

The theory of artificial neural networks has developed over the last 3 decades and is now well established. They are a way to describe and simulate a simple biological model of a real nervous cell and its network. The electrical charge over the connected dendrite to the nerve cell body represents one input value to the perceptron. Excitatory dendrits have a positive energy load or positive input value and inhibitorical dendrits have a negative electrical load or a negative input value. The nerve cell itself calulates a new energy level based biochemical conditions. The mathematical model uses the summation function and a scale or connection weight for each input node. The electrical output of the nervous cell is then transported over the Axon to other nervous cells.

Figure 1: A typical perceptron

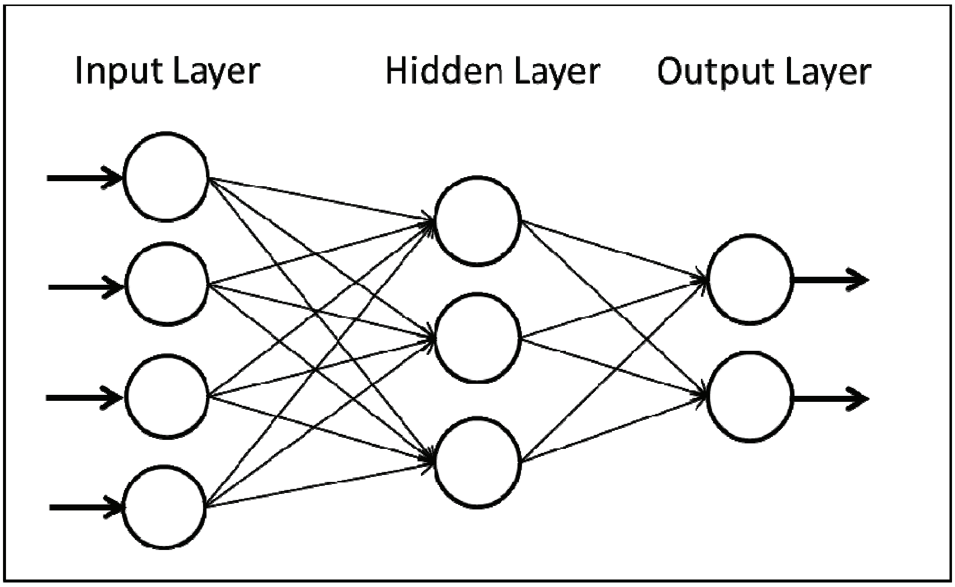


The mathematical model describes the output value of each perceptron as a result of the threshold function applied to the weighted sum of input values to the perceptron according to equation (1) and (2) below.

(1)

(2)

Figure 2: A typical feedforward 3 layer ANN built from many perceptrons



A feedforward artificial neural network is organized in typically 3 layers. One input layer, one output layer and in between at least one or more hidden layers. Each layer consists of several perceprons or nodes. The number of nodes of the input layer depend on the number of independant variables of the model and the number of nodes of the output layer depends on the number of dependant variables. The hidden layers in between are responsible for the processing and mapping. Each node of the input layer is connected to each node of the following hidden layer and each node of the hidden layer is connected to each node of the output layer. This kind of network is also called fully interconnected network. To build the proper functional mapping from the independant to the dependant values a training process has to be executed, which find the proper connection weights between two connected nodes. Typically you take the trained ANN as a blackbox.

3. METHODOLOGY

Artificial Neural Networks (ANNs) have been used as a powerful modeling tool in industrial and other applications. ANNs are identified as universal approximators and offer a systematic approach for modelling all kind of observable systems. But somtimes you are interested in the internals of the network. If you do input-output analyses or dependancy analyses of the observed system, you need a better way to describe these relationships than simply writing down the complex formulas or using a Hinton diagram for visualising the connection weight of the network nodes. This information is essential for model validation and to better understand the observed system. What is usefull in many applications is a way to quantify the dependancy between the nodes of the input layer and the nodes of the output layer. Very often you are interested in the importance or impact of an input parameter in relation to a certain output parameter of the ANN. Some information can probably be gained from a principal component analyses (PCA) or factor analyses, but only under the assumption that the underlying system, which is linear, is valid. However, for nonlinear systems these results are meaningless. Therefore the sensitivity calculation of the ANN becomes important. It is the basis to calculate an overall input-output dependence factor for each input-output relation and representing them in one single matrix. The dependancy matrix as described in equation set (6) and (7) represents exactly this kind of information computed directly from the trained artificial neural network. To calculate the values of the dependancy matrix we need to calculate the first order sensitivity of the ANN. This can easily be done as proposed by Hashem (Hashem, 1992) to calculate the first order output sensitivity directly from the first order partial derivatives of the network directly from the backward chaining rule of the learning algorithm. For completeness reasons the underlying mathematics is reproduced according to Hashem's publication (Hashem, 1992) and presented below in equation (3) to (5). The first-order sensitivity is computed from applying the simple backward chaining partial differentiation rule.

- For the network nodes in the output layer N:

$$\frac{\partial O_k}{\partial h_i^N} = \frac{\partial O_k}{\partial net_k^{N+1}} \cdot \frac{\partial net_k^{N+1}}{\partial h_i^N} = O_k(1 - O_k) \cdot w_{ik}^N , \qquad \forall\, i, k. \qquad (3)$$

- For the network nodes in the remaining hidden layers (layer j, j = N – 1,…,1):

$$\frac{\partial O_k}{\partial h_i^j} = \sum_l \frac{\partial O_k}{\partial h_l^{j+1}} \cdot \frac{\partial h_l^{j+1}}{\partial net_l^{j+1}} \cdot \frac{\partial net_l^{j+1}}{\partial h_i^j} = \sum_l \frac{\partial O_k}{\partial h_l^{j+1}} \cdot h_l^{j+1}\left(1 - h_l^{j+1}\right) \cdot w_{il}^j \quad \forall\, i, k \qquad (4)$$

- Finally for the input layer (layer 0):

$$\frac{\partial O_k}{\partial I_i^j} = \sum_l \frac{\partial O_k}{\partial h_l^1} \cdot \frac{\partial h_l^1}{\partial net_l^1} \cdot \frac{\partial net_l^1}{\partial I_i} = \sum_l \frac{\partial O_k}{\partial h_l^1} \cdot h_l^1(1 - h_l^1) \cdot w_{il}^0 \qquad \forall\, i, k \qquad (5)$$

whereas O_k : output of the k^{th} node in the output layer (layer N + 1),
h_i^j : output of the i^{th} node in layer j, j = 1,…,N,
I_i : input of the i^{th} node to the ANN,
net_l^j : weighted sum to the l^{th} node in j^{th} layer, j = 1,…,N+1 ,
w_{ik}^j : connection weight between i^{th} node in j^{th} layer and the k^{th} node in layer j + 1, j = 0, …, N .

The partial derivatives for the output layer, the hidden layer and the input layer have to be calculated following (3), (4) and (5) above. According to Hashem (Hashem, 1992) the first order sensitivity of a feedforward ANN is equal to the first order function derivative of the observed system. To learn about the unknown input-output relationships you have to calculate the Jacobian matrix for a certain input-output-vector. The Jacobian matrix tells us something about the model at a certain point in value space, but it tells us nothing about an overall input-output relationship of the observed system.

This is done by a component-wise calculation of the average of the absolute values of the respective Jacobi matrix component: For each input node i and each output node k of the ANN we calculate the average dependency factor $\overline{D}_{ik}$ as the average over all N training samples s for the respective absolute value of derivatives of output vector O^s by input vector I^s.

- The dependence factor $\overline{D}_{ik}$ is calculated as

$$\overline{D}_{ik} = \frac{\sum_{s=1}^{N} \left| \frac{\partial O_k^s}{\partial I_i^s} \right|}{N} \qquad \forall i, k \qquad (6)$$

- The final dependency matrix $\overline{\overline{DM}}$ is calculated as

$$\overline{\overline{DM}} = \begin{pmatrix} \frac{\overline{D}_{11}}{\max_j(\overline{D}_{j1})} & \cdots & \frac{\overline{D}_{1k}}{\max_j(\overline{D}_{jk})} \\ \cdots & \ddots & \cdots \\ \frac{\overline{D}_{i1}}{\max_j(\overline{D}_{j1})} & \cdots & \frac{\overline{D}_{ik}}{\max_j(\overline{D}_{jk})} \end{pmatrix} \qquad \forall j, k \text{ with } j \in \{1, .., N_I\} \; and \; k \in \{1, .., N_O\} \qquad (7)$$

whereas O_k^s : output value of training data set s of the k^{th} node in the output layer,
I_i^s : input value of training data set s of the i^{th} node to the ANN,

N : the total number of training data sets

s : the training data set,

$\max_j(\overline{D_{jk}})$: the maximum of all dependency factors of the k^{th} node in the output layer,

N_I : the number of nodes in the input layer,

N_O : the number of nodes in the output layer

The resulting matrix shows the normalized dependency factors as defined in equation (4) and (5) of the ANNs output parameters with respect to its input parameters.

4. SAMPLE APPLICATIONS

The dependency matrix as presented before, is easy to calculate because the underlying algorithms exist as part of a typical ANN programming library. It is a generic tool to calculate the input-output dependency directly from an ANN, which was built and trained to model an observable nonlinear system. It is an easy way to quantify input-output relationships. In the following examples the dependency matrix was used to gain an insight into the observed system respectively or to derive additional information to learn about unknown relationships. The following smaple applications of the dependancy matrix shall motivate the reader to apply this technology to describe parameter dependancies of nonlinear systems in an easy and accurate way, apart from traditional statistical methods.

4.1. Tourist marketing research application

Marketing research typically uses market models, to explain the markets. But it is particularly difficult to get a good insight into observed markets with traditional statistical methods. The following two studies of tourism market research in this and the next subchapter are presented, which use the dependency matrix as a tool for market analyses. The aim of this marketing research was to identify what influences the tourist expectations in relation to factors describing a touristic regional area or location. The first study (Stöckl et al; 2014) analyses the relationship between regional attachment factors and affective tourist reactions. One of the results is replicated below for descriptive purposes. Further details can be read in the given paper (Stöckl et al; 2014).

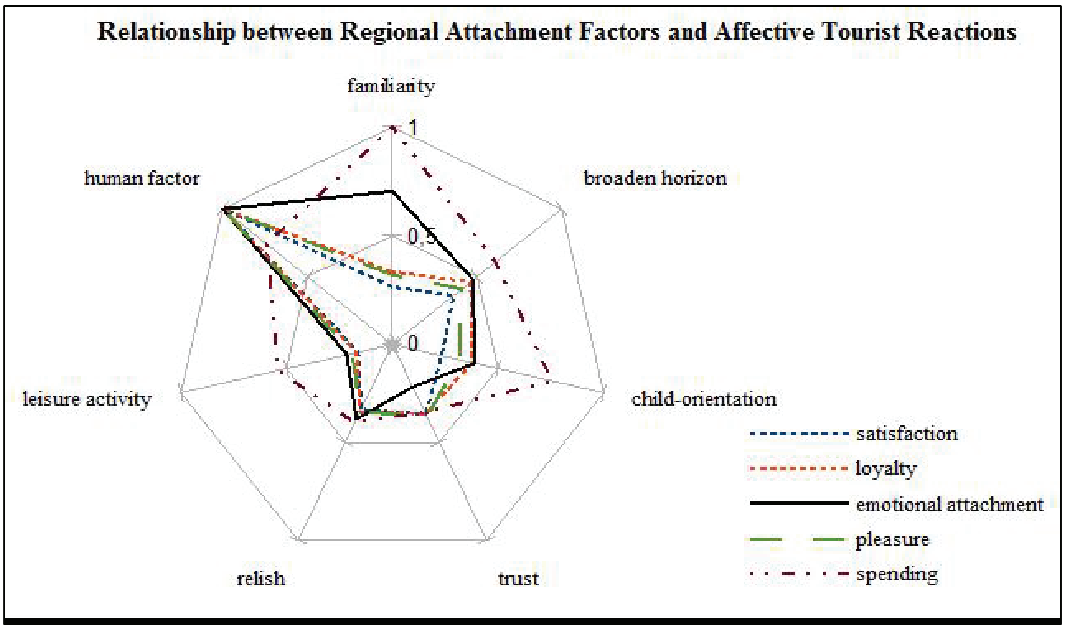
Table 1: Dependency matrix showing the relationship of regional attachment factors in relation to affective tourist reactions

Dependency Matrix	satisfaction	loyalty	emotional attachment	pleasure	spending
familiarity	0,26	0,33	0,70	0,32	1,00
human factor	1,00	1,00	1,00	1,00	0,74
leisure activity	0,16	0,17	0,21	0,19	0,54
relish	0,33	0,34	0,38	0,34	0,40
trust	0,36	0,35	0,22	0,37	0,35
child-oriented	0,23	0,38	0,39	0,32	0,76
broaden horizon	0,36	0,46	0,48	0,41	0,61

Source: (Stöckl et al, 2014)

The dependency matrix schon in table 1 is derived from the neural network model, that was built on the results of interviews done with tourists about their vacation experience in that touristic area. The final matrix quantifies how strong regional attachment factors (conditions, habits or offers) influence the tourist satisfaction.

Figure 3: Spider web diagram of the dependency matrix presented in table 1 above is showing the relationship between regional attachment factors and affective tourist reactions



Source: (Stöckl, 2014)

The dependancy matrix can easily be visualized using a spider web diagram as shown in figure 3 above.

4.2. Cross cultural research application

The second example application of the dependency matrix technology is used to study cross cultural differences in tourism. Along with other factors the emotional attachment to a certain region, which is dependent on regional attachment factors is analysed. The countries compared are Australia, Austria, France and Spain. The results of this study are described in detail in the given paper (Rinke, 2014). The following table and spider web diagram show the culture dependent differences of the defined attachment factors to a certain touristic region. The dependancy matrix is derived from an ANN that was build to model the mapping of regional attachment factors to different countries

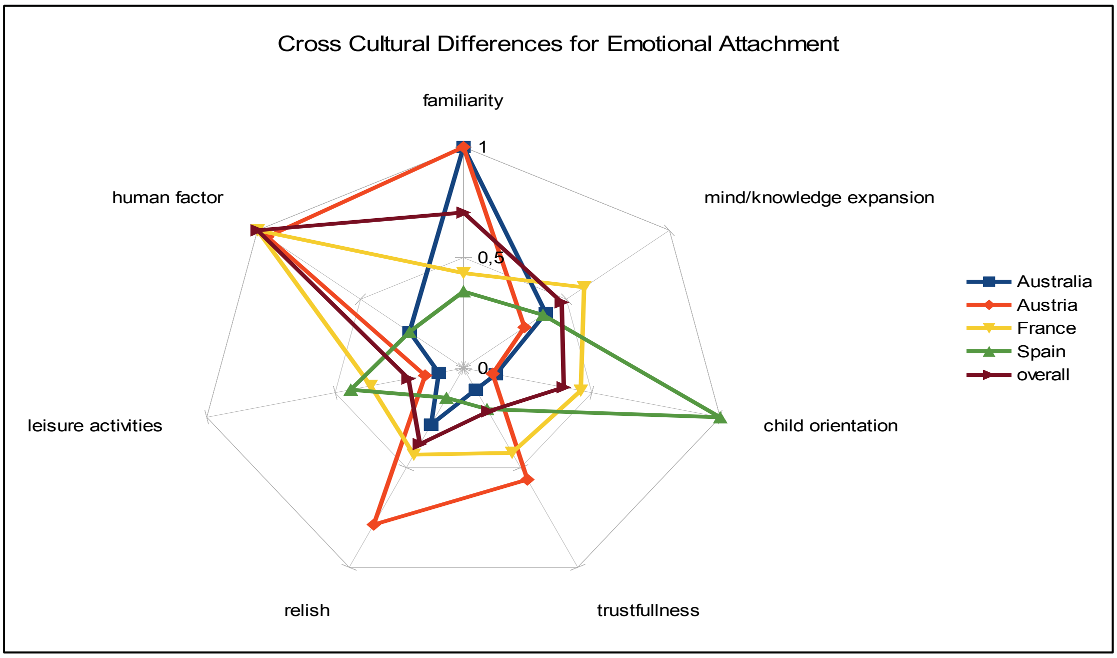
Table 2: Dependency matrix showing the differences between cultures and the relationship of regional attachment factors and emotional attachment to a region.

Dependency Matrix	Australia	Austria	France	Spain
familiarity	1,00	1,00	0,43	0,35
human factor	0,26	0,95	1,00	0,26
leisure activity	0,10	0,15	0,36	0,44
Relish	0,28	0,79	0,43	0,15
Trust	0,11	0,56	0,43	0,21
child-oriented	0,13	0,11	0,46	1,00
broaden horizon	0,40	0,30	0,59	0,39

Source: (Rinke, 2014)

A spider web diagram is a perfect way to visualize the relationships between different cultures and important regional attachment factors.

Figure 4: Spider web diagram of the dependency matrix as presented in table 2 showing the cross cultural differences for the emotional attachment for a touristic region

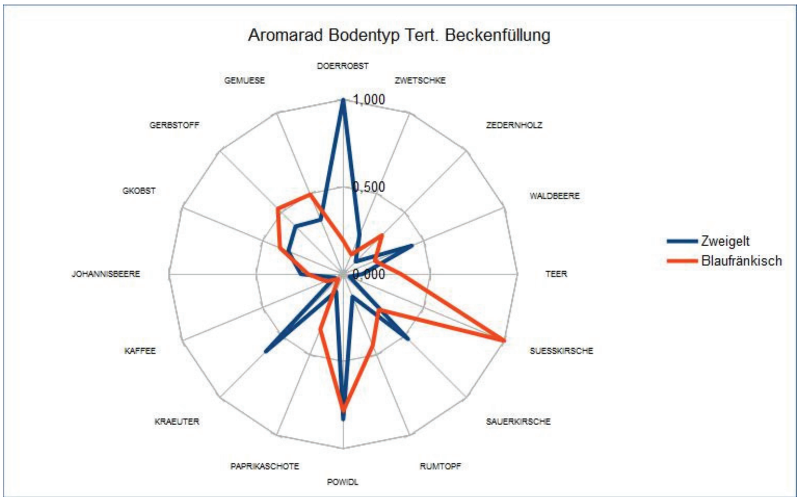


Source: (Rinke, 2014)

4.3. Agricultural research application

The third example refers to an agricultural application, that makes use of the described dependency matrix. This research analyses standardized micro vinified red wines ("Blaufränkisch", "Zweigelt", 2011) of geologically different areas of the wine growing region of the Burgenland, which were tasted and quantitatively assessed in tasting commissions. The dependency matrix was used for the presentation of tasting results, on a classification of wines according to increasing descriptive intensities. Based on this wines could also be classified according to the soils of their origin. Additionally those descriptors typical for the two red varieties on certain parent rocks were determined using ANN models for the first time to describe the areal influence (terroir) in relation to the characteristics of wine made from that grapes, grown in that specific region. The dependency matrix is used to calculate a terroir specific aroma profile for wines in relation to certain terroir based on the build and trained artificial neural networks. The results have been confirmed with cluster analyses (Flack, 2014).

Figure 5: This spider web diagram visualizes the aroma profile for two traditional wine types produced in the north of Burgenland, a district of Austria.



Source: (Flack, 2014)

5. CONCLUSIONS AND IMPLICATIONS FOR THEORY AND PRACTICE

The dependency matrix is a useful tool to quantify the relationship of input and output parameters of an observed system, as long as it is modeled with ANN technology. In all areas where an input-output analyses is required and the underlying system is nonlinear, it is very easy to calculate the dependancy matrix out of the trained network. It can be used as a standalone tool to gain an inside view of any observable system, if the model building has been done with artificial neural network technology. The results gained from the dependency matrix are easy to understand and practical to use in many different applications and areas of research. The dependancy matrix has been applied to many different kind of applications, where the modelbuilding is done with an artificial neural network. Appropriate applications are input-output analyses, feature detection, assoziation analyses, dimensionality reduction or impact analyses. A comparision of more traditional technology for this kind of application has only be done with cluster analyses so far. This leaves room for further research and analyses of the dependancy matrix in combination with artificial neural network modelling.

REFERENCE LIST

1. Bagozzi, Richard P. and H. Baumgartner (1994), "*The Evaluation of Structural Equation Models and Hypothesis Testing*" in Principles of Marketing Research, ed., Richard Bagozzi, Cambridge, Mass: Blackwell, 386-422
2. Cybenko, G. (1989), "Approximation by superpositions of a sigmoidal function." Mathematics of Control*, Signal and Systems, 2*:303-314
3. Eftekhar B, Mohammad K, Ardebili HE, Ghodsi M, Ketabchi E (2005): *Comparison of artificial neural network and logistic regression models for prediction of mortality in head trauma based on initial clinical data*. BMC Med Inform Decis Mak, 5:3. PubMed Abstract
4. Flak, W., Krizan, R., Passmann, G., Rinke, W., Stöckl, A., Tiefenbrunner, W., ... & Wuketich, A. (2014). Differentiation of the terroirs of true-to-type wines from the Burgenland based on descriptive sensory evaluations-Part 1:'Blaufränkisch'and'Zweigelt'. Mitteilungen Klosterneuburg, Rebe und Wein, *Obstbau und Früchteverwertung, 64*(1), 20-29.
5. Hashem Sherif, (1992): *Sensitivity Anlyses for Feedforward Artificial Neural Networks with Differentiable Activation Functions*, Proceedings 1992 IJCNN vol1 pp.419-424, IEEE (1992), ISBN 0-7803-0559-0
6. Hornig K., Stinchcombe M., and White H. (1989), "Multilayer feedforward networks are universal approximators." *Neural Networks, 2*:359-366
7. Omidvar O., Elliott D. L. (1997), *Neural Systems for Control, Academic Press*, ISBN 0-12-526430-5
8. Pao, Yoh-Han (1989): *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley Publishing Company, ISBN 0-201-12584-6
9. Patterson, Dan W. (1996): *Artificial Neural Networks: Theory and Applications*, Prentice Hall Simon & Schuster, ISBN 0-13-295353-6
10. Stöckl A., Rinke W., Eisingerich A. (2015): *Using ANNs to Determine Place Evoked Effective Consumer Reactions in Wine Tourism*, book chapter, Kyuho Lee (editor), Strategic Winery Tourism and Management, CRC Press, ISBN 978-1-9268-9568-0
11. Rinke W., Stöckl A., Eisingerich A. (2014): *Cross-Cultural comparison in Marketing Research-The Role of Equivalence Models and Appropriate Data Analyses Techniques*, Proceedings CCBC 2014 pp.159-168, ISBN-978-3-8440-2760-0
12. Runkler, Thomas A. (2010): *Data Mining, Vieweg+Teubner Fachverlag*, Wiesbaden 2010, ISBN 978-3-8348-0858-5
13. Swingler, Kevin (1996): *Applying Neural Networks*, Academic Press Harcourt Brace & Company Publishers, ISBN 0-12-679170-8
14. Terrin N, Schmid CH, Griffith JL, D'Agostino RB, Selker HP (2003): *External validity of predictive models: A comparison of logistic regression, classification trees, and neural networks*