ARTIFICIAL INTELLIGENCE APPLIED TO REAL ESTATE VALUATION
An example for the appraisal of Madrid

Julio Gallego Mora-Esperanza

ARTIFICIAL INTELLIGENCE AND TRADITIONAL SYSTEMS

The computerisation of real estate valuation began in the early 1980s, coinciding with the development of information systems technology. Subsequently, different statistical techniques were incorporated to process market data, among which the method of Multiple Regression proved especially relevant.

The use of Artificial Intelligence Systems for real estate valuation is more recent, dating from about 15 years ago. Since then there have been numerous experiences, and the creation of new models is on the increase. Among the authors of the most interesting systems (some of whose papers have been used as the basis for this report), the following merit special mention:

Author	Year	Research Area
Borst	1991	New England (U.S.A.)
Tay y Ho	1992	Singapore
Do y Grudnitiski	1992	California (U.S.A.)
Evans	1993	United Kingdom
Worzala	1995	Colorado (U.S.A.)
Borst	1995	New England
Mc Cluskey	1996	
Rossini	1997	Southern Australia
Haynes y Tan	1998	Gold Coast (Australia)
Bonissone	1998	
Cechin	2000	Porto Alegre (Brazil)

It is unrealistic to consider this type of work as experimental and remote. Over the last decade, Artificial Intelligence has undergone strong development and is now operational in Spain in specific fields. As an example, the Tax Department has recently prepared a system of Artificial Intelligence to detect V.A.T. fraud.

In the field of real estate valuation, numerous studies have been carried out to compare Artificial Intelligence Systems with traditional appraisal methods, in particular the Multiple Regression model. The majority of these studies calculate the percentage of error of both the Artificial Intelligence System and the Multiple Regression model, and run them with a package of market samples in which the sales price is a known factor.

The results of these comparisons seem clear. Artificial Intelligence Systems show an average error rate of between 5 and 10%, while Multiple Regression presents a higher average, between 10 and 15%. In some tests the results have been similar from both systems, but overall, Artificial Intelligence Systems show a higher degree of precision.

These studies demonstrate that another advantage of Artificial Intelligence systems over Multiple Regression is the capability of the former to estimate the value of properties with characteristics that vary significantly from other surrounding properties (outliers). This is because Artificial Intelligence Systems submit samples to a mathematical process that is much more complex than the Multiple Regression model, which is limited to a simple polynomial equation. However, this advantage does not show through in all studies, in fact some tests indicate that Artificial Intelligence Systems also have difficulties to reach a precise value estimate of special properties (outliers).

THE BRAIN AND COMPUTERS

Artificial Intelligence has developed as a result of research into the working of the human brain.

Until recently, brain research methods allowed a view into the brain's interior and the detection of physical damage, but did not supply information on how the brain works. Today, modern scanning techniques allow the analysis of some brain processes. Knowledge of the brain is growing rapidly, opening the door to highly interesting comparisons between the working of the brain and computer processes.

Until recently, it was not unreasonable to believe that the computer and the brain were to an extent similar in structure and operation. Effectively, both are able to capture external information, both can store data in their memories for future use and, using external data and stored memory, can carry out internal processes to obtain results.

However, advances in brain science have discovered that the brain's structure and the way it works are completely different from computers.

This discovery is neither recent nor sudden: some relevant differences have been apparent from the beginning, when the first computers started to operate.

Firstly, it is evident that computers possess the capacity and precision for calculation, and a much higher processing speed than the brain. It is estimated that data transmission in the brain is approximately one million times slower than the inside of a computer. The calculations made by a small computer are unthinkable for a human being.

On this basis, it could be said that the brain is a "bad computer", but the crux of the matter is that the brain has very little to do with computers.

As their capacity increased, computers were assigned new tasks, in the belief that they would be capable of anything. However, it has been demonstrated that there are certain tasks that computers do not do well.

One of these tasks is Image Recognition, e.g. to classify types of fruit, or to recognise facial features for controlled access to a building. No matter how many lines of programming were added, the results continued to be unsatisfactory.

Why does a computer, with its processing speed and calculation capacity, have serious difficulties to recognise a face, when a person can do this instantly?

Computers have also been unsuccessful in imitating the way people walk. Take the attempts to equip robots with legs, rather than wheels, to access dangerous slopes (e.g. in volcanic areas). The results have been frustrating, not only because of the complexity of imitating the structure of human legs, but also because the computer that controls their movement has problems processing the enormous amount of data, which must change with each step in order to maintain the robot's balance. However, a person learns to walk fairly quickly in early infancy, as soon as the legs are strong enough to carry him/her.

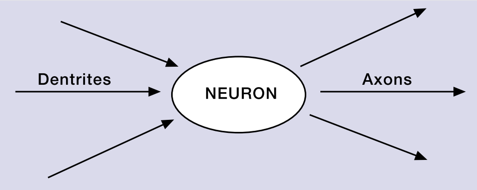
The answer to these questions is that the computer and the brain are very different in structure and the way they work.

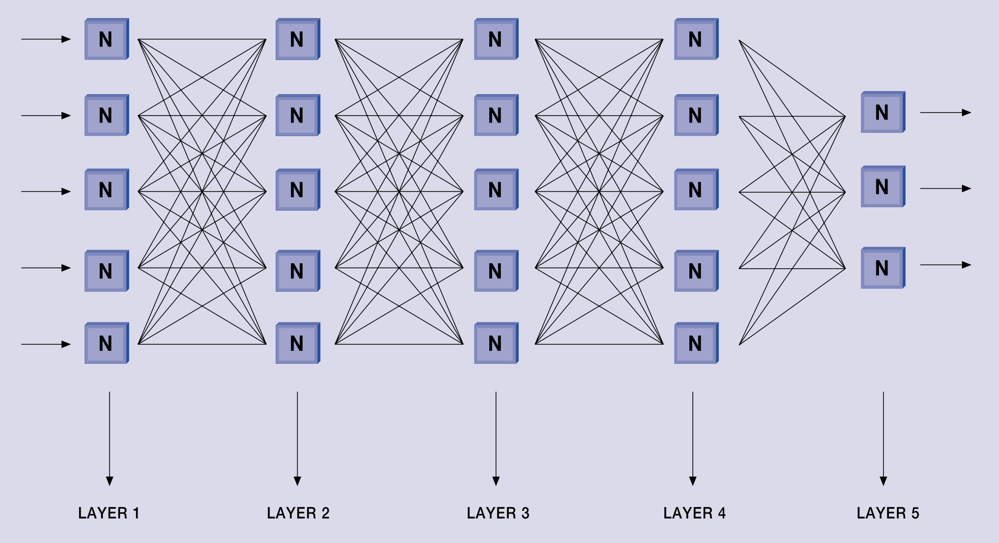
The computer contains a structure of microprocessors that are mainly connected "serially", allowing it to reach very high speeds of data transmission.

The brain's version of the microprocessor is the neuron, but organised in a different way.

Neurons are not serially interconnected. Each neuron is connected to several others, and receives information from some of them through connectors called "Dentrites". After performing its internal process, the neuron sends its information to other neurons through connectors called "Axons".

Therefore, the fundamental difference is that neurons are massively interconnected in parallel, forming layers.





This structure implies that each neuron can be connected to many others. Incoming information is distributed among a large number of neurons working in parallel. At any given time there are many neurons simultaneously processing information.

As mentioned earlier, this results in a slower process, but on the other hand the brain is able to process the enormous quantity of information needed for the above-mentioned tasks, and can also manage information that is partially incorrect, redundant or incomplete without this excessively affecting the result.

Further, compared with the brain, a computer has a very limited number of processors. It is difficult for a computer to have a million microprocessors, while the brain of an insect can contain this number of neurons. Although it is true that insects are incapable of mathematical calculation, they do perform complex functions necessary for their survival.

The human brain has between 10,000 million and 100,000 million neurons, connected in a complex network of layers.

This structure of the brain is oriented to learning, and this is what constitutes the fundamental difference with computers. For many tasks, the brain, rather than being programmed, "learns".

The programming of a computer, however complex, is fixed. On the contrary, the brain modifies its internal process as a result of the mistakes it makes, in order to reduce errors to a minimum.

The computer is "programmed", while the brain "learns", and this learning process takes place through trial and error.

Let us suppose that neurons send impulses to each other so that the body stands up and starts to walk. If at the first attempt the body falls to the right, this error is registered by the brain, which will adjust the quantity of neuron impulses to correct the error. At the next attempt the process between neurons will be slightly different as a result of this correction, and if the body now leans forward, the system will again adjust to correct the error, and so on, until the body is able to keep its balance.

This is how a person learns to walk, to keep his balance, to ride a bicycle, etc. The brain continuously adjusts its processes through the method of trial and error.

The huge processing capacity of the brain, and its learning system, make it highly adaptable to changing external circumstances, and it is able to address an enormous variety of tasks. Adaptability allows the brain to perform the tasks of an industrial engineer, a professional tennis player, or a musician.

In conclusion, traditional computers were designed to have a large capacity for calculation, but not to work in the same way as the human brain.

ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURONAL NETWORKS (ANN)

When these basic principles of brain structure and working became known, it was evident that information systems had to imitate them in order to successfully perform tasks that traditional computers were unable to do.

This is where Artificial Intelligence comes in.

Artificial Intelligence takes various forms. One of the most important is the Artificial Neuronal Network (ANN).

ANNs are computer systems whose microprocessors, rather than laid out in series as in traditional computers, are connected in parallel, forming layers and making multiple connections, imitating the way the neuronal network is organised in the brain.

Obviously it is not a matter of simulating the human brain, but simply building a system that works in a similar way and on the small scale. Some ANNs are built with as few as 20 "neurons" (microprocessors).

Initially, "artificial neurons" were microprocessors, but these have now been replaced in the majority of cases by computer programmes that imitate their function.

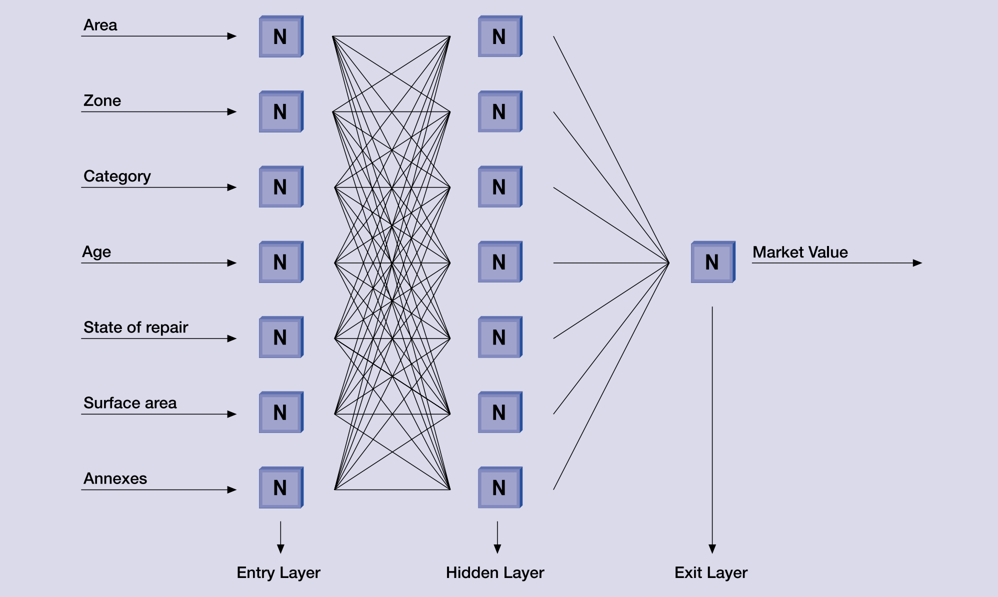
Currently ANNs are highly developed, and are being applied in numerous fields: to diagnose illnesses, for credit risk analysis, to predict the evolution of stock markets, etc.

The Artificial Intelligence system developed by the Tax Department to detect V.A.T. fraud is an ANN.

The majority of Artificial Intelligence systems developed for real estate valuation are also ANNs.

ARTIFICIAL NEURONAL NETWORKS AND REAL ESTATE VALUATION

To understand how an ANN works, we will view a typical example of their application to real estate appraisal.



This example shows 15 neurons organised in 3 layers. The first layer, consisting of 7 neurons, receives incoming data. The second layer is known as the hidden layer, and also has 7 neurons. The third layer is the exit layer, with a single neuron, where the final result – Market Value – is generated.

ANNs for real estate valuation usually work with between 10 and 50 variables, and therefore feature an entry layer with the same number of neurons, likewise the second hidden layer (although this can vary between half and double the number of variables), and an exit layer containing a single neuron.

There are numerous ways to design an ANN. More or fewer layers can be included, and more or fewer neurons per layer. Each neuron can send its data to neurons in the next layer (forward connection), in its own layer (lateral connection), or in the preceding layer (backward connection). A neuron may receive information from all the neurons in the preceding layer (total connection) or only from part (partial connection).

Likewise, numerous design recommendations exist to organise the structure of an ANN. These refer to the number of layers, number of neurons per layer, connections, etc. The reality is that there are no fixed rules, and the designs are adjusted through testing.

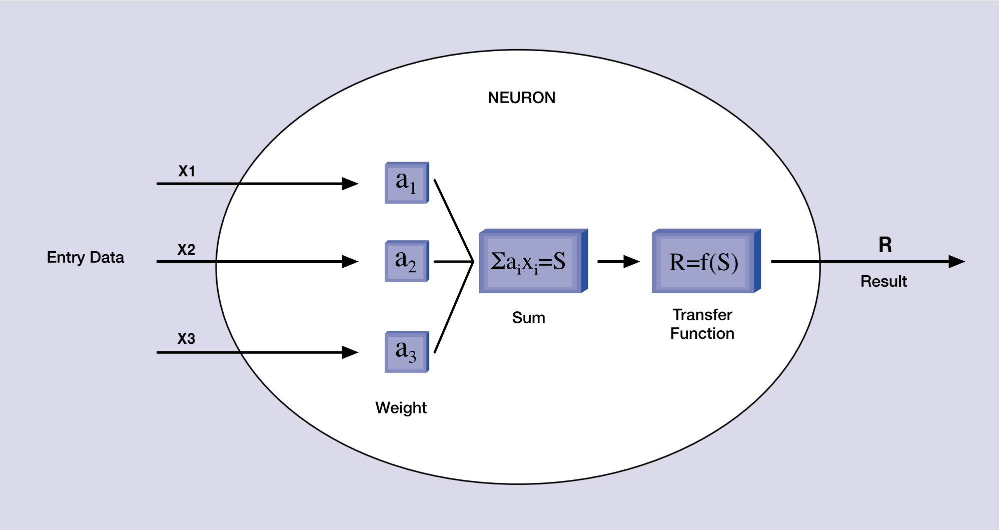
The majority of ANNs designed for real estate valuation are similar in structure to the example given above, i.e. they are systems with “Total Forward Connection”.

TRIAL AND ERROR

ANNs never work the first time round. They need to be “trained”, i.e., they need to “learn”. For this purpose they are submitted to Trial and Error cycles.

In order to understand this process, we will see how an artificial neuron works, in an attempt to imitate the workings of a brain neuron.

The neuron receives data from other neurons. In ANNs, these data are numbers. Therefore, the neuron receives a number from each neuron sending it information (x_1, x_2, x_3, etc.) and sends another number (the same number to all) that is the result of its process (R).



As shown on the graph, the numeric data sent by other neurons (x_1, x_2, x_3, etc.), are not processed forward with their original value, but are first weighted (a_1, a_2, a_3, etc.). Weighting plays a major role, since this is where the learning capacity of both the neuron and the system resides.

After weighting, the neuron performs two internal operations.

It first calculates the “S” value, as a result of the sum of each weighted variable.

Secondly, it calculates the “R” value, applying to “S” a transfer function, R=f(S) (the function used most frequently by ANN neurons is the sigmoid function, due to its ease of use in computer programming : $R=1/(1+e^{-S})$.

Lastly, the resulting “R” value is sent to the next neurons (unless the neuron is in the exit layer, in which case “R” is the final result, i.e. the market value).

Now that we have analysed a single neuron operation, let’s take a look at how the complete system works.

ANNs work in two modes: the “training and learning” mode; and the “production” mode.

In the training phase, a group of samples are selected for which all variables, as well as the market value, are known.

There are no established figures to determine the number of samples needed to correctly train an ANN, but recommendations exist indicating that this number should be proportional to the number of entry variables.

Let us assume for the purposes of our example, with 7 variables, that we have 60 training samples.

Firstly, the samples are divided into two groups: one to “train” the system and the other to “verify” the system. Thus, following the example, the 60 samples are divided into one group of 40 for training, and a second group of

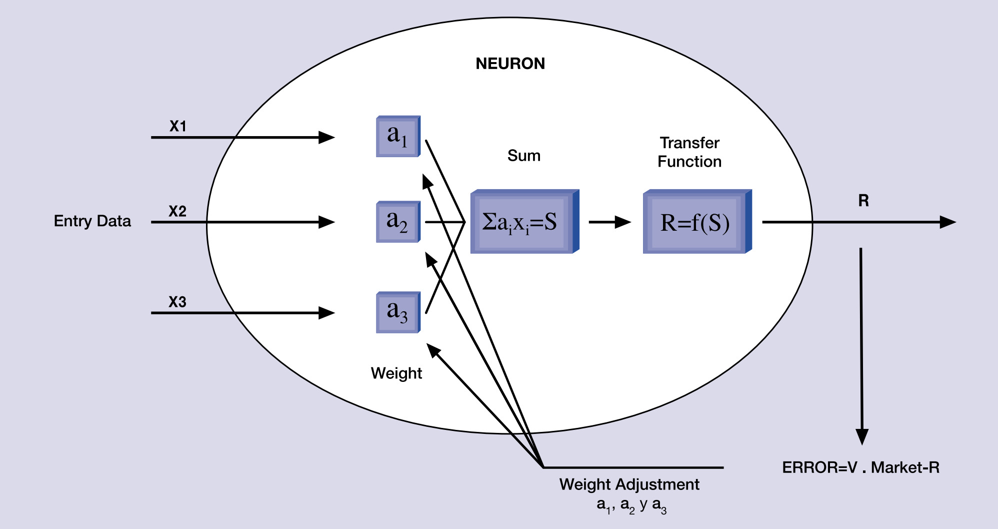
20 for verification. We will later see the utility of these verification samples.

We will now begin to train the system, feeding it with the data from the first sample. For the system to work, we must first assign the weightings (a_1, a_2, a_3, etc.) of each neuron. This is initially done assigning random values.

The system then calculates the market value of the first "R" sample.

After obtaining the first result, the system compares it with the sample's real market value, which is a known value, and obtains the "error" or difference between the calculated value and the real value. Logically, in this first round the error will be large.

After calculating this first error, a corrective algorithm modifies the weights of all the neurons in the network, with the objective of minimising the error.



The graph shows how a single neuron is corrected, but the system does this for all the neurons.

During network design, the designer can program whatever correction algorithm he/she prefers, although a standard "backward correction algorithm" (which is a generalisation of the Delta rule) is the one normally used.

Following adjustment of the neuron weightings (a_1, a_2, a_3, etc.), a test sample is again processed and a new result obtained, which is again compared with the known market value to obtain a new error.

The algorithm again adjusts the weightings, and this process is repeated with all the test samples until the error is reduced to the minimum (not zero).

The computer can repeat this adjustment process hundreds and even thousands of times in a few seconds, to reach close proximity to the real market value.

In this way the system corrects itself "by trial and error", the same way the brain does.

TRAINING AND VERIFICATION

How can we know when an ANN is fully trained?

How many times does the correction cycle have to be performed?

A hundred, a thousand, or ten thousand times?

In general these questions are answered by running tests. Network trainers handle a few basic concepts for correction, such as "learning ratio", "the moment", "entry noise", and "learning and testing tolerances".

It is not necessary to go into these concepts to understand how training should be performed. Firstly, it is evident that the higher the number of entry variables, the greater the quantity of test samples will be necessary, and with more samples, more correction cycles will be needed.

It might at first appear that the more times the correction cycle is repeated the better, since we would reduce the error more and more and get closer and closer to the exact market value of the test samples.

However, in practice this is not the case. After a certain number of error correction cycles, the Network becomes "over-trained", in other words, it over-adapts to the 40 training samples, and begins to produce poor results with other samples.

This is why in point 5 we mentioned that the first 60 samples featuring a known market value would be divided into two groups, one of 40 for training, and the other of 20 for verification.

The Network is trained using the group of 40 samples, but with the 20 verification samples we will check that we have run neither too few nor too many correction cycles.

In this way we can determine how many correction cycles are needed to train the Network.

Verification, using the 20 verification samples, that the Network has achieved an allowable error level (usually under 5%) marks the end of the "training" phase ends and the start of the "production" phase, in which we will use the ANN to estimate the market value of real estate in cases where this value is not known.

ANNs can return to the "training" phase whenever necessary. In fact it is advisable to train them periodically, to keep up with market changes.

ADVANTAGES AND DISADVANTAGES

We have already mentioned some of the advantages of ANNs compared with previous systems: higher precision, and increased capacity to estimate the value of special properties (outliers).

Another positive feature is that the system is user-friendly and easy to handle. The user only has to introduce known variables on one hand, and obtain the market value on the other. What goes on inside the system is not the user's concern. Network design and training must be performed by experts, but use of the network in the "production" mode is very simple.

Additionally, although at first glance ANN technology might seem complex, in practice it employs very few formulas, and further, these usually adapt to established standards. An ANN can run perfectly on a micro-computer, even on a spreadsheet application.

Regarding the disadvantages, there is one that is quite important.

ANNs are also known as "black boxes" because it is imposible to know what goes on inside them. There is no way to explain how an ANN calculates real estate market values, neither with equations, tables, or anything else. The complexity of the iterative weighting correction process and of sums and transfer functions within the multiple neuron connections makes this an impossible task.

This might not be a problem for private companies, and in practice it is not, since ANNs are quickly spreading and improving in this sector, for multiple applications.

However, when calculating real estate values for tax purposes, the Administration has to be able to explain, both to taxpayers and to courts of law, how this calculation has been done.

It is true that taxpayers and courts of law didn't understand the Multiple Regression process either, but formally these systems do have a valid, although complicated, explanation. The problem is that ANNs cannot be explained, either easily or otherwise.

For this reason some experts are developing methods that allow a simple description of how an ANN works.

In any case, this issue does not necessarily represent a major obstacle for the use of ANNs for real estate valuation by the Administration. The weight of the explanation of results does not need to rest exclusively on the Network. These systems should be considered as "tools" to help the valuator in his/her work, not closed systems that exclude the valuation expert. In effect, various countries have included ANNs in their real estate valuation computer systems, as help tools for their valuators.

EXAMPLE FOR THE APPRAISAL OF MADRID

To verify the qualities of this system, an ANN has been prepared for the valuation of collective housing in the Region of Madrid. This target has been selected because of the predominance of this type of housing in Madrid, with approximately 2 million units, representing nearly two-thirds of the total property in the Region.

Rather than working on collective housing in a specific town, we preferred to cover the entire regional territory, since this scope adds o the complexity of the project, thus allowing us to test the capacity of ANN. Further, if the ANN were capable of defining a regional model, this would allow us to perform continuous monitoring of the real estate market in a relatively efficient fashion.

The Network was prepared on a spreadsheet application, and contains the basic ANN elements. The system allows the increase or reduction of the number of layers and the number of neurons per layer, and introduction of the chosen transfer functions and correction algorithms. In any case, changes involve introducing the formulas into the spreadsheet, given that the intention was not to develop a standard ANN computer system, but simply a tool with which to conduct this research.

After running different tests and modifying the variables and the number of neurons, we chose a total forward connection ANN consisting of three layers, with a total of 20 neurons.

The entry level features 12 neurons, one for each selected variable.

The hidden, or middle layer, features 7 neurons.

The exit layer features a single neuron, whose result represents the Value calculated by the ANN.

The selected variables are not intended to constitute the definitive set for the valuation of collective housing in Madrid, but rather a group of variables to lead us to an initial approximation.

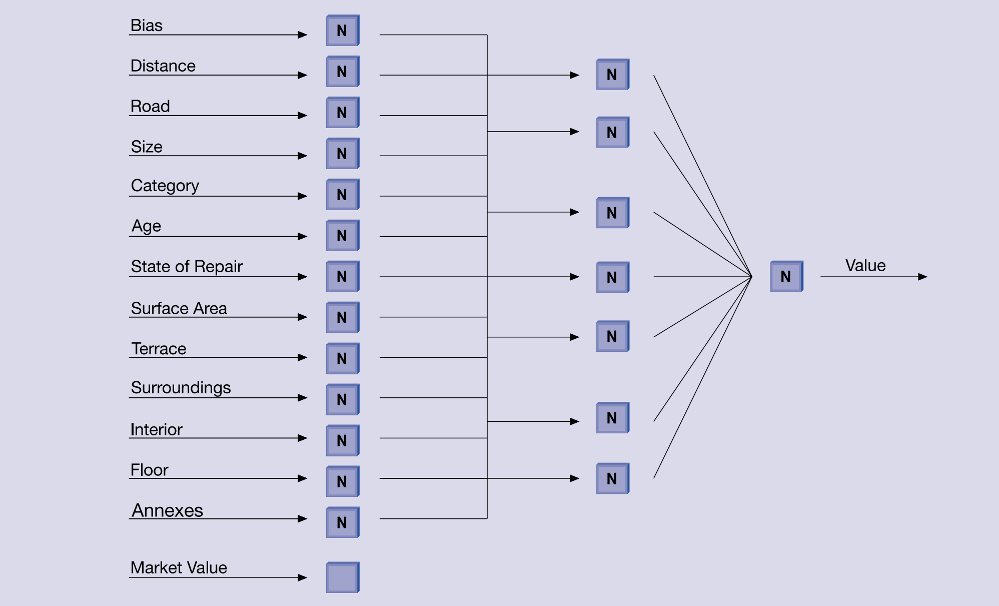
1. Distance: distance of the town, district or area from Madrid city centre.
2. Road: access road to the city of Madrid. Different access roads are coded.
3. Size: size of the town.
4. Category: construction category.
5. Age: age of the building.
6. State of repair: state of upkeep of the building.
7. Surface area: constructed area of housing.
8. Terrace: surface area of the terrace.
9. Surroundings: situation in the town, district or area

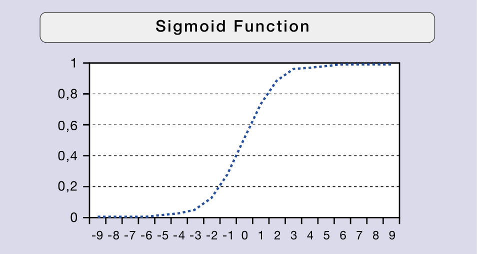
• Highly unfavourable situation	1
• Unfavourable situation	2
• Average situation	3
• Favourable situation	4
• Highly favourable situation	5

10. Interior: house interior
11. Floor: floor number
12. Annexes: coded per level of annexes (gardens, swimming pool, etc.).

Additionally, a constant value entry equal to "1" has been included to form the bias of each neuron. This is a frequent practice in ANN design, with weightings that mark the bias for each neuron, which are not constant, since they vary throughout the calculation process.

The transfer function chosen for the neurons is the sigmoid function, which is the one most frequently used in ANNs. The correction algorithm is that of backward retro-propagation.





The parameters used are ANN standards: Learning Ratio, Moment, and Entry Noise.

The Learning Ratio indicates the portion of error that is corrected in each adjustment action. A ratio of 0.1 signifies that 10% of the detected error is corrected. This is a basic parameter of ANNs, since during the learning phase the error must be reduced gradually. If it is reduced too suddenly, the behaviour of the ANN may become unstable. In this example we have used ratios between 0.6 and 0.

The Moment is a parameter that softens the correction process, so that when an error is detected, instead of applying the correction coefficient indicated by the Learning Ratio, the correction applied is halfway between the current error and the immediately preceding one. This ensures that a defective sample will not substantially change the direction of the learning. This example applies Moment 0.

Entry Noise is a parameter that randomly deforms the variables in a percentage. Its purpose is to ensure that an excessively perfect group of samples does not prevent the ANN from capturing a model that might be standardised. In this example, Entry Noise is considered equivalent to 0, since it is generally recognised that real estate market samples are “noisy” per se.

To verify and interrupt ANN operations, a Market Reference coefficient has been used – MR – as a percentage between the ANN-calculated value and the known market value of the sample (an MR of 100% means that the calculated value and the real value are the same). The objective is for the median of MR of the samples to be close to 100% and dispersal of MR of the samples (dispersal coefficient) less than 10%.

A group of 100 market samples have been used featuring different towns in the Region of Madrid. 85 samples are used for training, and 15 for verification. It is worth mentioning that, for an ANN of 12 variables, it would be advisable to use approximately 250 samples, 200 for training and 50 for verification. However, given the investigative nature of this work, 100 samples should suffice to show some interesting results. All the samples pertain to year 2002.

Following correction of the ANN, the results obtained during the training phase were: Median 100%, Dispersal 8%.

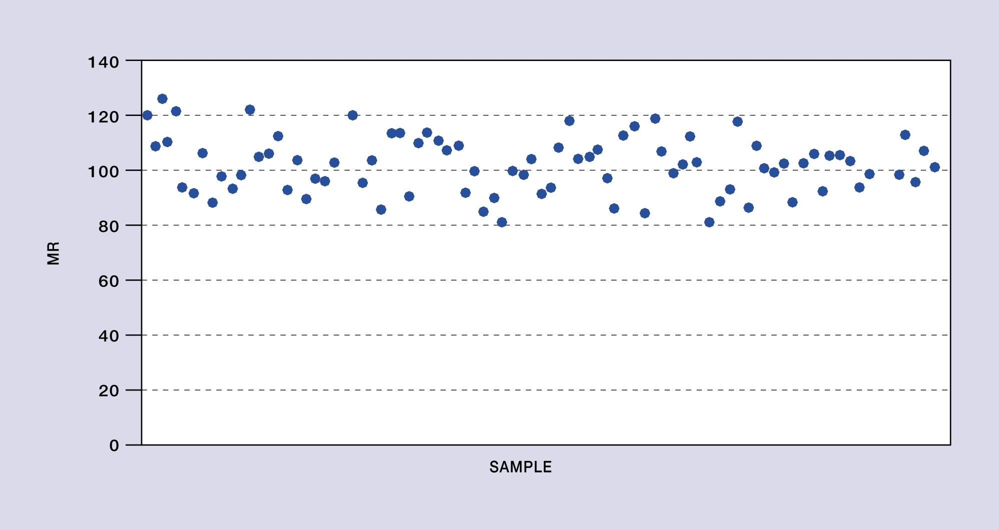
The graph shows the final dispersal of the samples after completion of ANN training.

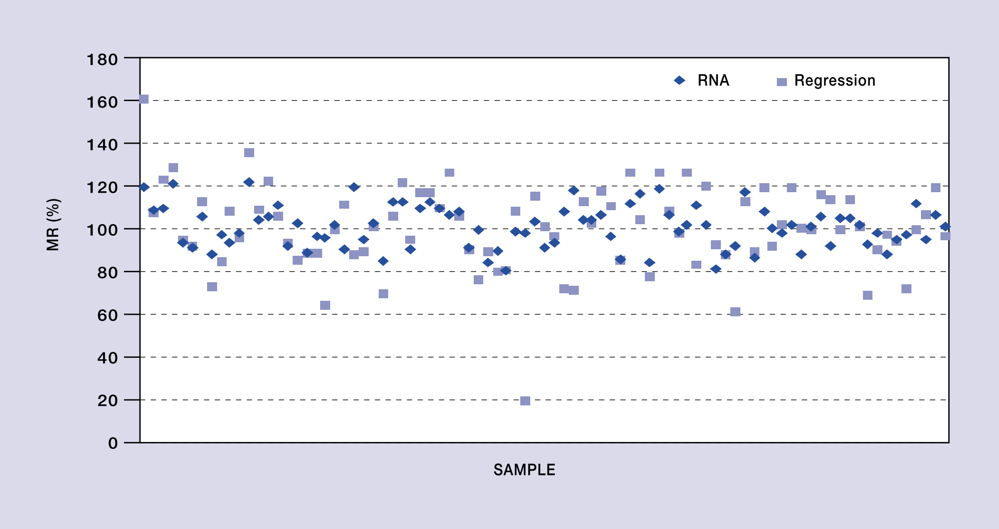
We can compare this result with the result that would be obtained by applying the Multiple Regression model to the same 85 samples.

In this case, using Multiple Regression the MR median is also close to 100%, but the dispersal rate rises to 15%, i.e., 7 per cent more than with ANN.

The following graph offers a comparison of the two dispersal rates, and shows how, for the majority of samples, the ANN result is closest to the 100% line.

It also shows that Multiple Regression indicates the existence of two Outliers, but that ANN places them closer to the 100% line.





Thus, in this example the two advantages that some investigators attribute to ANN over Multiple Regression are demonstrated – increased precision, and improved processing of outliers. However, this is a concrete example and its results can not be generalised.

Following completion of ANN training with the 85 samples, we proceeded to the verification phase using the 15 remaining samples. In this phase, ANN is applied directly to the verification samples, whose market value is also known, to verify if the system continues to perform well with samples that have not participated in the training. In this case, results were satisfactory, since ANN maintained the 100% median and 8% dispersal over the 15 samples. Using Multiple Regression, the results were 95% median and 14% dispersal. However, as mentioned previously, this ANN would need to be verified with a higher number of samples. Normally, indicators tend to perform less well in the verification phase.

After verifying that ANN maintains good results with the verification samples, it can now be applied to appraise collective housing whose value is unknown. It can be used to find individual values of specific houses or for mass valuation processes.

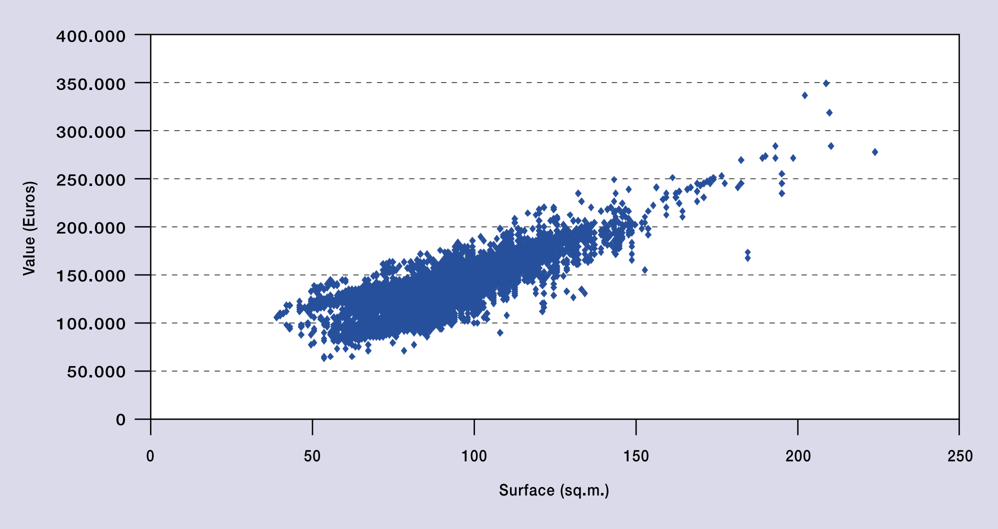
This ANN has been used to valuate all collective housing in the Region of Madrid. The total number of houses appraised is 1,893,000, distributed through all the townships of the region, including the capital.

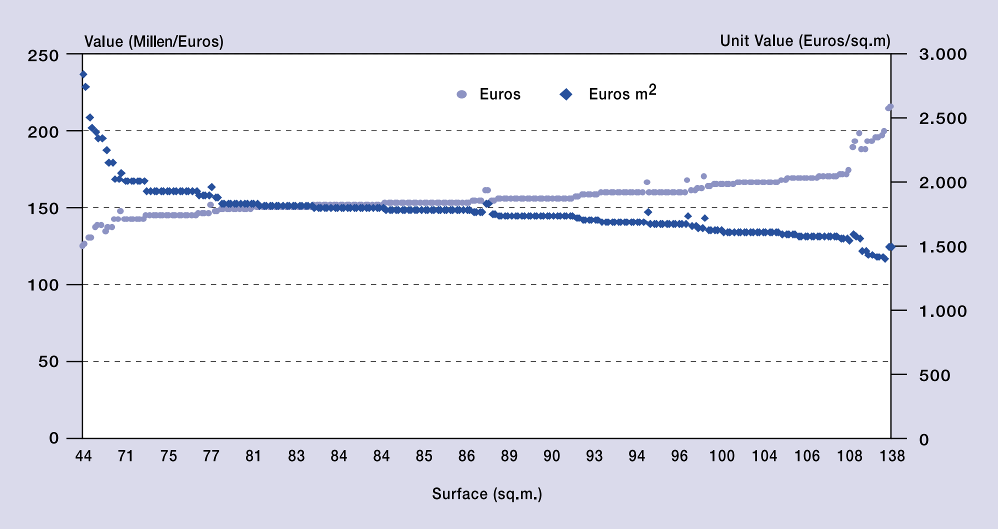
To conduct the mass appraisal process a Database Manager was used featuring the 12 ANN variables for all properties, and the described system formulas were programmed. The appraisal process took between 15 and 20 minutes on a P.C.

The results obtained are shown on the following graphs, and would appear to indicate a fairly reasonable approximation to market values, bearing in mind the investigative nature of this work, the small number of samples, and the fact that these are values for 2002.

The following graph shows the results for the town of Pinto. Values are shown per size of the approximately 100,000 collective housing residences in this district.

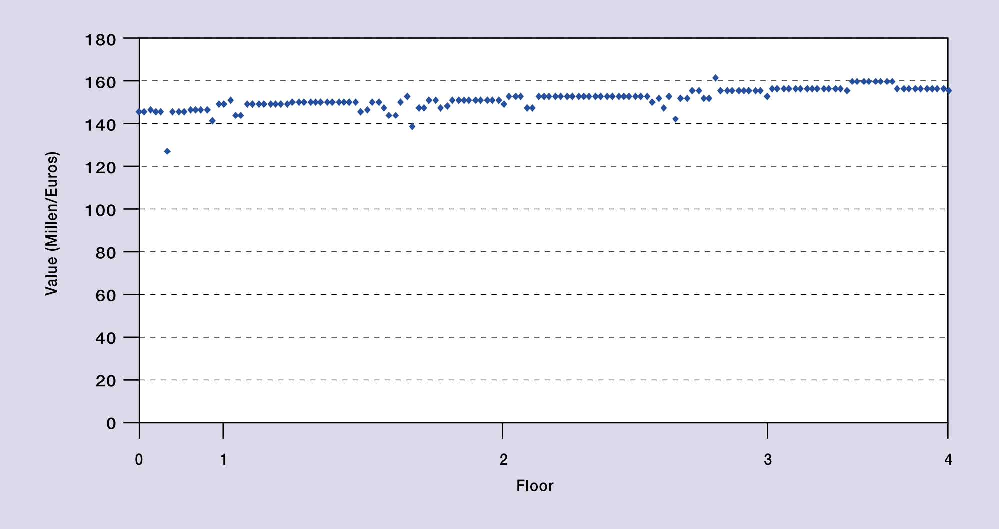
The distribution profile curve indicates that values are proportionally lower for larger surfaces (in other words, a 200 sq.m. flat is not worth twice the value of a 100 sq.m. flat). This is observed more clearly in the following graph.





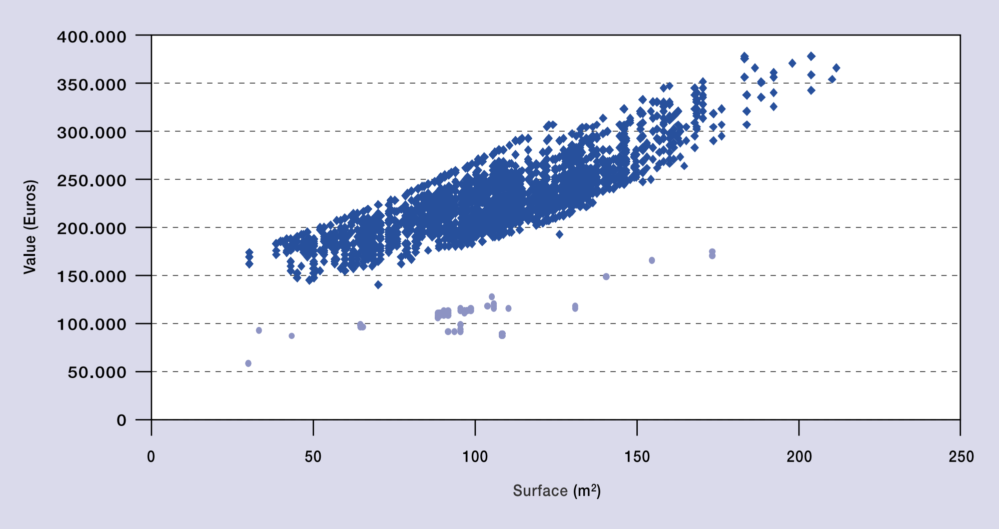
In order to ensure that the variables would not alter the result, this graph was built from newly constructed category 4 houses located on the 2nd floor of their respective buildings. We observe that the unit values diminish the larger the surface of the residence, in accordance with the profile that ANN has captured from the training samples. We should mention that this is not a simple profile, since it depends on the interaction of all the variables, and therefore for each town, surroundings, category, etc. the results would be different.

Another interesting variable whose behaviour can be analysed is that of floor level. The following graph shows values by floor of category 4 housing measuring 85 sq.m.



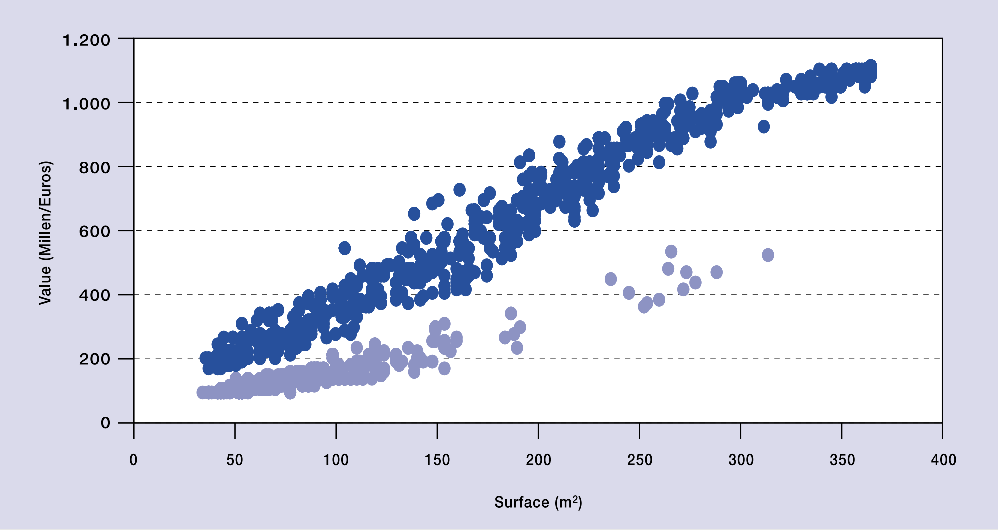
We observe that values increase the higher the floor. ANN has captured a value increase based on floor level that demonstrates a soft but noticeable profile.

The next graph shows the distribution of values by size of two other towns in the Region: Tres Cantos and Villaconejos. Tres Cantos is a large town, located 22 km. north of Madrid, and is very uniform in character, since almost all buildings have been constructed in the last 20 years and are similar in category. Villaconejos is a smaller town, further away from the metropolitan area of Madrid (approx. 50 km. south), where collective housing is less prevalent.



In the same manner the distribution of values can be analysed by district, area, housing estate or street. The following graph is an example of the distribution of values by size in two streets in Madrid: calle Velázauez, located in the city centre, and calle Marcelo Usera, located to the south.

In view of the results, and taking into account the experimental nature of this work, it appears that ANNs are capable of capturing the collective behaviour of real estate market variables, even in a widespread territorial area with a wider variety of products and more complex variable relationships.



Bibliography

MARGARITA M. LENK; ELAINE M. WORZALA and ANA SILVA (1996): “High-tech valuation: should artificial neural networks bypass the human valuer?”

FRANCISCO PIZARRO REDONDO: “The paradigm of artificial neuronal networks”.

PETER ROSSINI (1998): “Using Expert Systems and Artificial Intelligence for Real Estate Forecasting”.

PETER ROSSINI (2000): “Improving the Results of Artificial Neural Network Models for Residential Valuation”.

RAVI MADAPATI (2002): “Artificial Neural Networks: A Primer”.

MARCO AURELIO STUMPF GONZALEZ; LUCIO SOIBELMAN and CARLOS TORRES FORMOSO: “Explaining the Results in a Neural-Mass Appraisal Model”.

JOHN D. HAYNES AND CLARENCE N.W. TAN: “An Artificial Neural Network Real Estate Price Simulator”.

CLARENCE N.W. TAN: “An Artificial Neural Networks Primer with Financial Applications Examples in Financial Distress Predictions and Foreing Exchange Hybrid Trading System”. ■