#175

OPPORTUNITIES OF ARTIFICIAL NEURAL NETWORK GENERATED VGA

Training a Multilayer Perceptron to recognize the underlying structures of space

DÁNIEL SZEMEREY
University College London, London, United Kingdom
daniel.szemerey.15@ucl.ac.uk

SEAN HANNA
University College London, London, United Kingdom
s.hanna@ucl.ac.uk

AVA FATAH GEN. SCHIECK
University College London, London, United Kingdom
ava.fatah@ucl.ac.uk

ABSTRACT

This paper presents the research conducted with the aim of understanding if new advances in computer science, more specifically a type of supervised, feedforward Artificial Neural Network, a Multilayer Perceptron (MLP) is able to estimate the values of Visibility Graph Analysis (VGA) without the need for expensive calculation.

The overarching hypothesis is that an MLP can be setup in a way that it can be trained to learn the relationship between spatial configuration and the VGA (neighbourhood size and clustering coefficient) derived from it. Two hypotheses are stated: firstly, if such an MLP can be created than it will be able to generate spatial configurations for specific VGA values as inputs (mode A); secondly, the network would be able to generate VGA when presented with spatial configuration faster, compared to current method and with negligible error (mode B).

The hypotheses were tested by creating unique setups of an MLP for each mode, all of which had a different configuration. As each combination of possible setups were tested, the performance of the networks could be compared to each other and to the traditional method of VGA calculation.

Both mode A and mode B was able to achieve satisfying results that prove that an MLP is able to generate –with limitations- configurations based on VGA input and it is able to calculate the neighbourhood size and the clustering coefficient of a 2D layout substantially faster and with negligible error.

All MLPs were created at a generic space, therefore the MLP taught once can be adopted universally to most spaces. The implications of the two systems is that spatial analysis can be integrated into the design process, enabling interactive, instant analysis and the possible deployment of optimisation procedures, for instance a Genetic Algorithm.

KEYWORDS

Machine Learning; Visibility Graph Analysis; Multilayer Perceptron; Generating Spatial Configuration

1. INTRODUCTION

With the explosion of the computational power of personal computers we have for the first time adequate tools to understand the seemingly chaotic world around us. Through the combination of theoretical advances and new algorithms, designers reached a stage where they can predict the future through analysis and simulations far more accurately than it was previously possible. It is finally possible to understand Problems of Organized Complexity (Weaver, 1948).

Through the work of Bill Hillier, Julien Hanson and other colleagues (Hillier & Julienne, 1984) researchers can now begin to understand the complicated but close relationship between spatial configuration and human behaviour. Tools were developed to assist designers to carry out spatial analysis such as Axial Line Graphs or Visibility Graph Analysis (Al_Sayed, Turner, Hillier, Iida, & Penn, 2014).

However, as that these tools are crucial and they have been around for over thirty years, the question arises: What would be the effect if these tools were more accessible to the broader design community rather than just researchers? What would be the impact on early stage design, on the way we educate Architecture Students and on the way we generally look at spatial systems?

This paper aims to provide understanding if new advances in machine learning can be implemented to spatial analysis for it to be easier to understand and use.

The research problem

Methodologies developed for Space Syntax are implemented in software that requires a linear process of analysis: to create a design, feed it into a standalone software, wait for the analysis to finish, then evaluate the design and change it accordingly. This process requires, compared to the whole design period, a disproportionally large time and is not really viable in everyday practice.

Secondly, the processes involved in analysis is computationally heavy. They are also unidirectional, meaning that the current system cannot produce spatial arrangements according to desired analysis. It is possible to find a solution for both problems if we look at new advances in computational methods.

The human brain evolved to filter and process large amount of interlinked stimuli, such as understanding intricate scenery. On the other hand, computers are good in number-crunching 'human-hard' problems, such as precisely applying computational rules to a large set of data, but have difficulty in 'human-easy' problems, such as recognizing faces.

These barriers led to the birth of a new field of computer science, Artificial Neural Networks (ANN), that was founded on mimicking how the human brain might work (Mcculloch & Pitts, 1943). These new systems compute in parallel and are taught rather than given a pre-programmed set of rules (Flake, 2001), which means they are much faster and highly adaptable.

Multilayer Perceptrons are part of a unique set of ANNs. They are efficient systems that can associate two sets of correlated information, thus it is able to classify any new data it is given.

The main research question of the thesis is: *Can a Multilayer Perceptron understand the connection between a given spatial configuration and its Visibility Graph Analysis?*

This question has to be evaluated by testing the performance of the MLP created as part of the methodology.

Hypothesis 1: the MLP is able to 'understand' the underlying patterns in a data of spatial configuration and analysis and it is able to generate a 2D layout when given a desired analysis.

Hypothesis 2: the MLP is able to 'understand' the underlying patterns in a data of spatial configuration and analysis and is able to calculate a Visibility Graph Analysis faster than current methods.

Tool for early stage design

The creative process of Architectural Design has been often subject to change in the past century. In contrast to Gaudi's famous hanging models (REF), today's designer can have more iteration cycles before needing to physically building anything, as simulations speed up each cycle and still provide reliable results.

This shift towards early stage design is understandable, as with each new design phase the costs of change increases exponentially. Integrating analytical tools in the early stage design means minimizing the risk by complimenting the intuition of the designer with data.

There are three ways early stage design would benefit from a more efficient calculation of VGA.

Interactive Design

If the analysis would run hundred times faster, that would mean designers could get instant feedback on valuable VGA information during their initial steps of planning. Precedents of employing an interactive system for Architecture have been common, such as the use of interactive physics engine for form finding (Senatore & Piker, 2015).

As the designers are able to see instantly what a certain design decision result in, they can fix potential problems as early as possible.

Optimisation

With current methods VGA is of limited use in the fitness criteria of a Genetic Algorithm as it would need an uneconomically big computational power calculating the VGA of each individual of the population at each frame. Machine learning has been used to approximate models in similarly computationally expensive domains such as structural (Hanna, 2007) or aerodynamic (Wilkinson, Bradbury, & Hanna, 2015) simulation. A faster approximation of spatial analysis could similarly be used in the fitness criteria for the floor plan of a building.

Synthesis

Machine Learning has been widely used to synthesise solutions that would otherwise need a creative designer. David Cope's (1999) work on artificial music composition and Harold Cohen's (1984) work on algorithmic painting shows how creative programs can be. If an Artificial Neural Network was able to recreate the solutions that arithmetic VGA provides, then there is a possibility that the procedure could be reversed and the computer could provide design solutions for desired behaviour of people.

Teaching tool for Designers

Kinda al-Sayed research demonstrated, that a designer being guided by explicit knowledge (a factual knowledge that the person is aware of knowing) can "partially enhance their function-driven judgment producing permeable and well-structured spaces" (Al-sayed, Dalton, & Ho, 2010). As buildings are designed to enclose social interactions, it would be crucial for Architects to understand not just intuitively (without implicit knowledge) but explicitly the spaces they are designing. Making Space Syntax methods more interactive could be a way of improving Architecture Students ability to understand the quantitative connection between their design and social behaviour.

System Study

Artificial Neural Networks are parallel systems that process data at the local level but their output results in a global solution. There might have not been any connection between two points of the space during the calculation of the ANN generated VGA, however if the result is identical or similar to that of the traditional VGA, that would mean that the spatial system has properties of higher abstraction than that of simple numerical calculation of the graph.

This would explain why designers have an intuition on how people might move in a space without actually calculating quantitative properties of that space. As previously mentioned Kinda Al Sayed's work (Al-sayed et al., 2010) showed how learning factual, explicit knowledge improves the ability to design well-functioning spaces. An assumption to be made is that students learn these higher level structures of social behaviour in space during their studies.

Empowering designers

Current Spatial Analysis tools

'DepthMap' (new version: 'DepthMapX') was introduced by Turner and is the most widely used software for spatial analysis. It is able to do both Axial Maps and VGA analysis (Turner, 2001).

'SmartSpaceAnalyser' is a new add-on for 'Grasshopper' (a plugin for the popular software 'Rhinoceros') that performs instant analysis, but is not a competitor of 'DepthMapX' as it is unable to create neither Axial Maps nor VGA, just simple one location isovists (Happold, 2015). The advantage of it is that it is integrated in a CAD environment and can therefore give instantaneous feedback.

2. DATASETS AND METHODS

To handle the intricate relationship between any spatial configuration and its Visibility Graph Analysis (VGA), a Multilayer-Perceptron (MLP) was chosen as the core engine.

Such systems are taught by showing them a large number of labelled data samples, which can be considered as "cards" containing both the spatial configuration and its corresponding VGA. The network adjusts itself during the learning phase and thereby starts associating the expected VGA output with each plan. After training, the network would be expected to generalise to predict the output of plans it has not seen.

Firstly, a large dataset of cards was created, each card containing one pseudo-random arrangement of cubes on the same generic space and the VGA of that configuration. Secondly, the weights of the MLP are adjusted in the iterative process of showing those cards to the network. Thirdly, the system was tested and adjusted for better efficiency and validation (Fig. 1).

Two modes (A and B) were developed with the MLP. Mode A was taught so that if given a certain desired VGA outcome as input data then it will produce a solution to the configuration of the space. Mode B generates the VGA if a certain configuration of a space is fed in as input. The current paper only discusses Mode B in detail, Mode A is briefly shown.

2.1 CREATION OF THE DATASET

An MLP learns the regularities that exist in data shown to it. The larger and the more versatile the dataset is the more the network is able to understand global correlations rather than structure specific to a single data. In this section it is discussed how a large, inhomogeneous dataset was generated using Cellular Automata and multi-threading (Fig. 1).

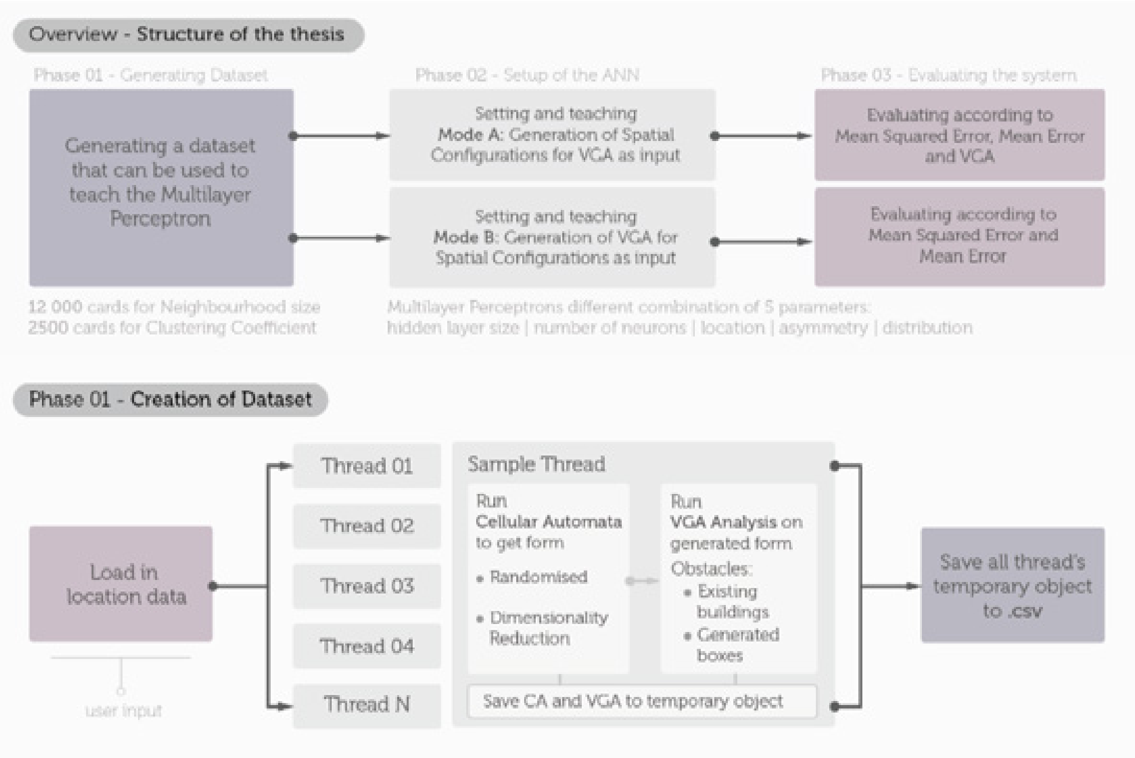

Overview - Structure of the thesis
Phase 01 - Generating Dataset
Phase 02 - Setup of the ANN
Phase 03 - Evaluating the system
Generating a dataset that can be used to teach the Multilayer Perceptron
Setting and teaching Mode A: Generation of Spatial Configurations for VGA as input
Setting and teaching Mode B: Generation of VGA for Spatial Configurations as input
Evaluating according to Mean Squared Error, Mean Error and VGA
Evaluating according to Mean Squared Error and Mean Error
12 000 cards for Neighbourhood size
2500 cards for Clustering Coefficient
Multilayer Perceptrons different combination of 5 parameters:
hidden layer size | number of neurons | location | asymmetry | distribution
Phase 01 - Creation of Dataset
Load in location data
user input
Thread 01
Thread 02
Thread 03
Thread 04
Thread N
Sample Thread
Run Cellular Automata to get form
• Randomised
• Dimensionality Reduction
Run VGA Analysis on generated form
Obstacles:
• Existing buildings
• Generated boxes
Save CA and VGA to temporary object
Save all thread's temporary object to .csv

Figure 1 - Overview of the structure of the paper. Before teaching the MLP a variety of solutions have to be generated: different spatial configuration of the cubes and their VGA. This diagram demonstrates the process of creating the dataset.

Setup

The amount of cards the ANN has to be presented with depends on the resolution, but more data results in equal or better performance. For the sake of testing the systems presented here, 12.000 solutions were calculated, which were split into 10000 teaching cards and 2000 testing cards. The input for each is the spatial configuration, expressed as a grid in which grid squares are either on or off. The output is the resulting neighbourhood values or clustering coefficient values calculated by VGA.

As the process of calculating 12.000 VGA requires a substantial amount of time, the system runs solutions in parallel threads. Although it is assumed that the system created here will be universally adaptable (the designers don't have to teach their own MLP), the speed to uniquely tailor the network can be important. Each thread is run on one of the physical or virtual cores found on multi-core processors.

Each thread contains three processes: the first generates spatial configuration by running a Cellular Automata, the second calculates the VGA of the existing environment and the generated spatial arrangement. The third process stores the values in a temporary object that will be later written as a .csv file.

The area that is analysed has to be broken down into smaller elements for both the creation of spatial configuration and analysis. The values of the grid are shown in Fig. 2.

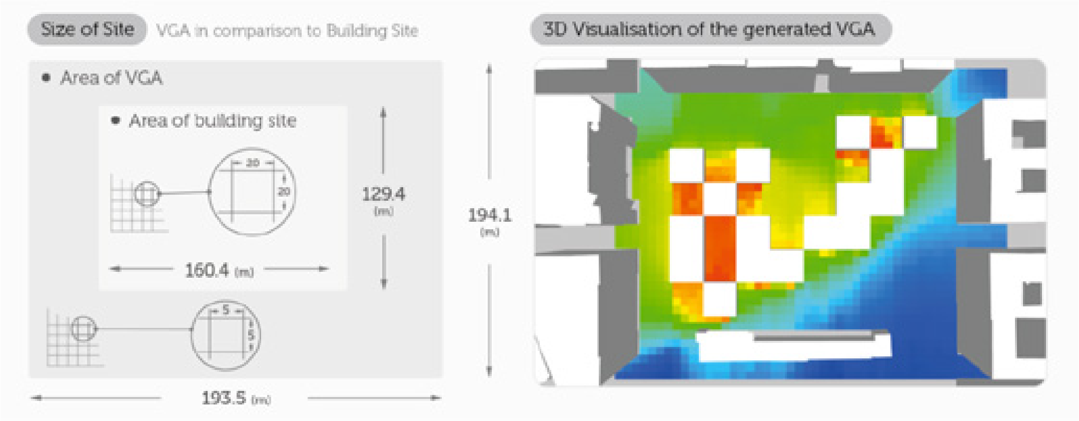


Figure 2 - Size of the area and of the grid for both the Visibility Graph Analysis (neighbourhood size) and the building site, where spatial arrangements are generated by placing boxes.

Generation of spatial configurations

The dataset has to consist of highly varied solutions, so that there is no bias towards one spatial configuration.

A grid of cubes was set up at the area of the building site. Each cube had a state of either up (1.0) or down (0.0). The initial grid consisted of 5 by 5 sized cubes, however a dimensionality reduction algorithm enlarges the cube size to 20 by 20. The area analysed in this thesis contained a grid of 6 * 8 boxes. For this reason, the number of possible states of the system is:

$$N = {n_{states}}^{n_{elements}}$$

$$N_{reducedSystem} = 2^{6*8} = 2^{48} = 2.81474977 * 10^{14}$$

The randomization process requires some form of generative process, otherwise if each cube's state is selected according to a probability it would result in a random noise of that magnitude. To tackle this issue a Cellular Automata was introduced that procedurally generate the spatial configuration.

Visibility Graph Analysis

To achieve the Visibility Graph, the given space has to be discretized into a fixed element size. For each element of the discretized grid the program checks if it 'sees' every other element. For this a double array of rectangle objects ('MyRect') is created, which store an Arraylist of 'MyRect' objects. Each time the program finds a rectangle that is visible to it adds that object to this Arraylist. The visibility at each cell of the grid is the size of the Arraylist. The clustering coefficient is calculated by implementing Turner's algorithm (Turner, 2001).

Each solution has a maximum and a minimum value of visibility. This is important as the values have to be mapped when fed into the MLP. The program records the local minimum and the local maximum of the solution and it also keeps track of those values between all the cards, which is the global range. During associative teaching the network has to understand these values in the global scale.

To tackle the problem of the edge effect a larger area is calculated and the edge are discarded.

Training the Multilayer Perceptron

After the dataset is generated an MLP can be created to test both of the hypotheses: to understand if it can generate spatial configuration based on VGA fed in as input Mode A has been created, to see if it is possible for the network to generate VGA based on spatial configuration Mode B has been explored. The method of how the MLP for each modes was set up is illustrated in Fig. 3.

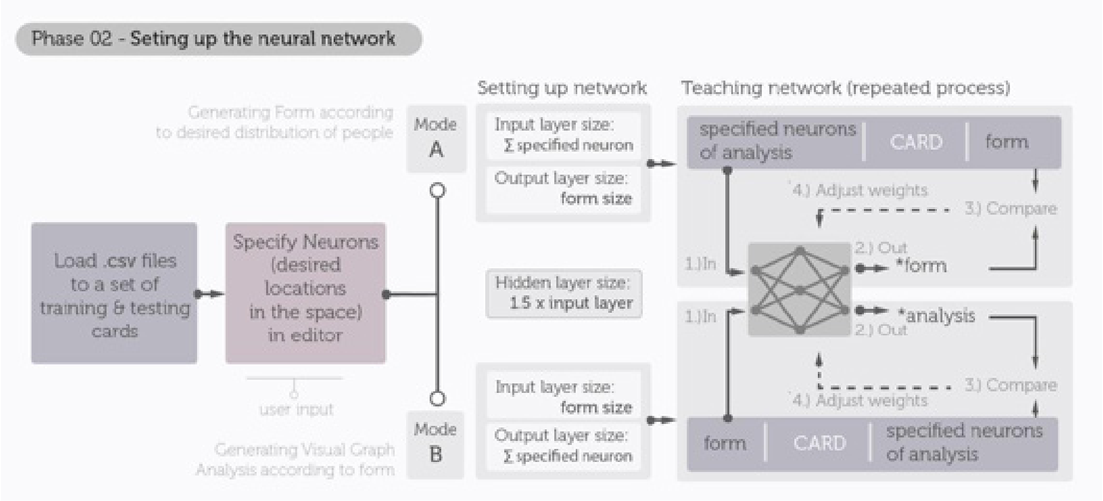

Phase 02 - Seting up the neural network
Setting up network
Teaching network (repeated process)
Generating Form according to desired distribution of people
Mode A
Input layer size: Σ specified neuron
Output layer size: form size
specified neurons of analysis
CARD
form
4.) Adjust weights
3.) Compare
Load .csv files to a set of training & testing cards
Specify Neurons (desired locations in the space) in editor
Hidden layer size: 1.5 x input layer
1.)In
2.) Out
*form
*analysis
user input
Input layer size: form size
Output layer size: Σ specified neuron
Mode B
Generating Visual Graph Analysis according to form
form
CARD
specified neurons of analysis

Figure 3 - How the Multilayer Perceptron was set up for Mode A and Mode B.

Once the required dataset was calculated and saved the data can be loaded into a set of teaching and testing cards, in the ratio of 5:1.

As the Neural Network uses a Sigmoid function when deciding if it should activate or not, all values fed into the network have to be mapped from -1 to +1.

Teaching the Multilayer Perceptron

An MLP is taught by showing it 'cards' that include two information: the spatial configuration form and its VGA. Every time a new card is presented as the input the system checks the difference between the output of the system and the solution on the card and adjusts itself by a pre-set learning rate.

Responding to inputs

The network is activated when all the output values of the neurons of the input layer are set. Then each neuron of the hidden layer calculates the weighted sum of the output of all neurons. The weighted sum is fed into a sigmoid function. The sigmoid function allows for a more continuous activation, thereby there is no drastic change between each state of the neuron during the learning phase. This model of the neuron is a modified version of the McCulloch & Pitts model, and can be written as the following formula. For each neuron:

$$y_i = sigmoid\ (\Sigma\, w_{ij}\, x_j)$$

Where y_i : the output of the neuron 'i' , w_{ij} : the weight of the connection between 'i' and 'j', x_j: the output of the input neuron 'j'.

After the output value of the hidden layer has been calculated the output of each neuron of the output layer is calculated. The inputs are now the weighted output values of the hidden layer.

Backpropagation

During the teaching process the weights are adjusted according to the difference of the network's output and the solution on the card. Each output neuron has an error that is the difference in value of its output and the value at the same location in the solution data. The error is multiplied by the steepness of the sigmoid curve at that location ('f(x)') to get the input value ('x') to the sigmoid function. It is also multiplied by a learning rate, which is needed to avoid local minima, as we are looking for global minimum.

The multiplied error gets then added to the weight. The same process is done for the hidden layer: the error values of the hidden layer's neurons are the sum of the errors of the output layer multiplied by the weight (more connected neurons are corrected more). The weights between the input and the hidden layer are than adjusted as described for the weights between the hidden and the output layer.

General methods of evaluation

This project used three ways to measure the performance of each MLP. The sum of all the errors of each neuron summed up, the Squared Mean Error and for Mode A a comparison VGA. The mean squared error is used in order to 'punish' large errors more than small deviations.

Each error described for 'Summed Error' is first squared and then divided by 2. This means if the difference between the output of the network and the solution on the card is very high, then the Squared Error will be even higher than simply taking the error and for small errors, it will be smaller.

The sum of all neuron's squared error is calculated and divided by the number of possible errors: the number of neurons times the maximum possible error, the range (2) squared and divided by 2. This multiplied by 100 gives the percentage of Mean Squared Error.

MLPs can overfit the data, which means even though the performance of the network seems high, it has only learned local correlations in the dataset and fails on new data presented. For this reason, the performance of the networks was measured on 100 cards from a separated testing set to see if no overfitting occurred.

3. RESULTS

The MLP used in this project consists of three layers of neurons. There are two modes: A for generation of spatial configurations and B for Creating faster VGA (Fig. 4) .

The input and the output criteria are different for Mode A and for Mode B. These modes are described below in detail:

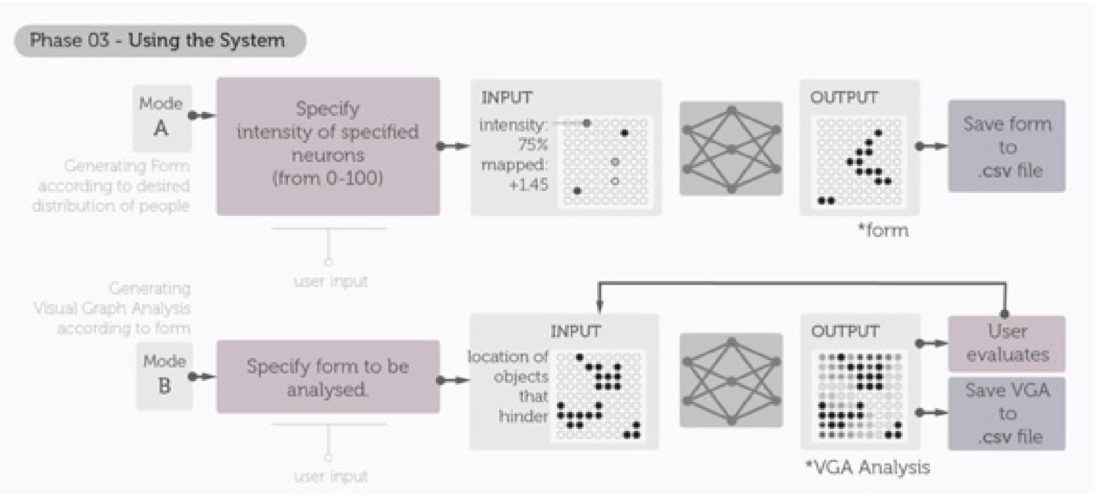

Phase 03 - Using the System
Mode A
Generating Form according to desired distribution of people
Specify intensity of specified neurons (from 0-100)
user input
INPUT
intensity: 75%
mapped: +1.45
OUTPUT
*form
Save form to .csv file
Generating Visual Graph Analysis according to form
Mode B
Specify form to be analysed.
user input
INPUT
location of objects that hinder
OUTPUT
*VGA Analysis
User evaluates
Save VGA to .csv file

Figure 4 - Two different modes and how the network is set up and responds to in each case.

3.1 MODE B – CREATING FASTER VGA

Depending on how small the space is discretised and how big the analysed park is, the VGA could take from seconds to more than an hour. This makes the design-analysis paradigm very slow and hinders possibilities for analysis incorporated interactive planning or for optimization processes.

The aim of Mode B is that the network performance can come close or equal that of the original analysis, but offers a much faster process. Fig. 5 shows the visualisation of an example output results compared to the solution on the card.

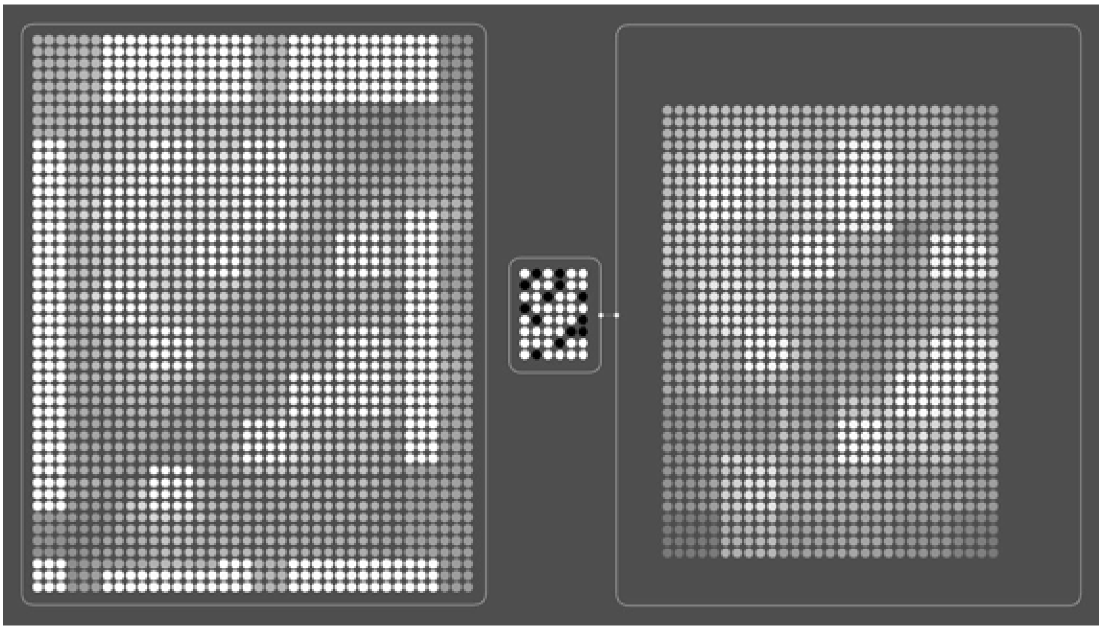


Figure 5 - MLP Generated VGA of Neighbourhood size. The space was divided into 5x5 networks.
Left: traditionally calculated VGA; Middle: Input spatial configuration; Right: MLP generated VGA

Possible design scenarios: Setup, Evaluation and Experiments

The MLPs input layer in this case are the values of the spatial configuration of the park and the solution to those inputs will be the VGA. Even though it is possible to specify location for Mode B, whereby only certain locations will be selected as outputs, it is presumed that users are more interested in using all locations to see the resulting analysis for the whole area.

To evaluate the performance of the system is straightforward for Mode B. Simply measuring the difference of the output of the network to the solution that is on the card. The performance is also clearly visible, as the input spatial arrangement should be clearly distinguishable on the output VGA, as those locations should have value 0 (shown as white). In the following various design scenarios are outlines and their performance are evaluated and discussed.

Experiments with a simple setup

Early experiments showed that setting a system up, that consisted of a lot of selected neurons, was not able to learn local differences: it clearly produced valid results for the perimeter area, as it was mostly high visibility, but shows constant white at the area of the boxes (Fig. 6).

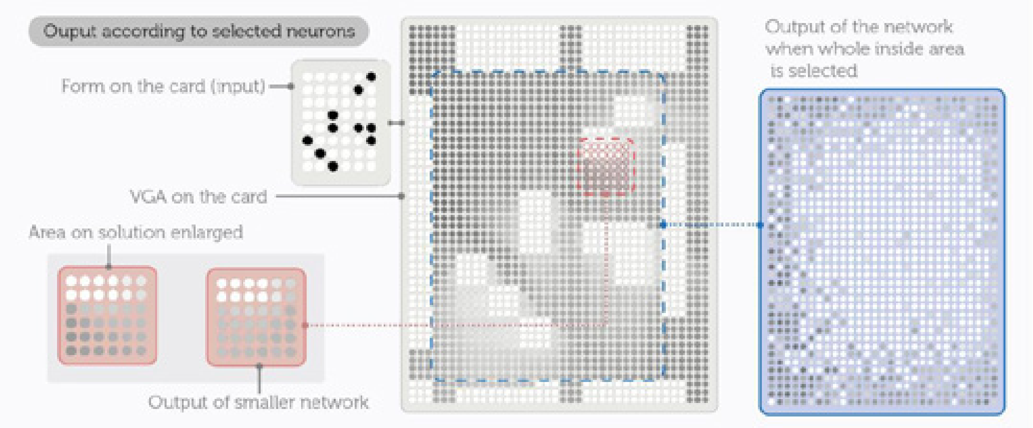


Figure 6 - Outputs of a network with only a small area selected and an output with the whole inside area.

It quickly became apparent that this is a problem originating from the amount of selected neurons. When only a small area was selected the network was able to produce valid VGA result below 1%(±0.14) mean squared error margin.

How it performs if network is separated in parts

A solution found to this problem was to divide the Visibility Graph Analysis into smaller areas, then a ANN is created for each of them. The input layer is always the same (the form) the output layer is one cell of the grid overlaid on the VGA.

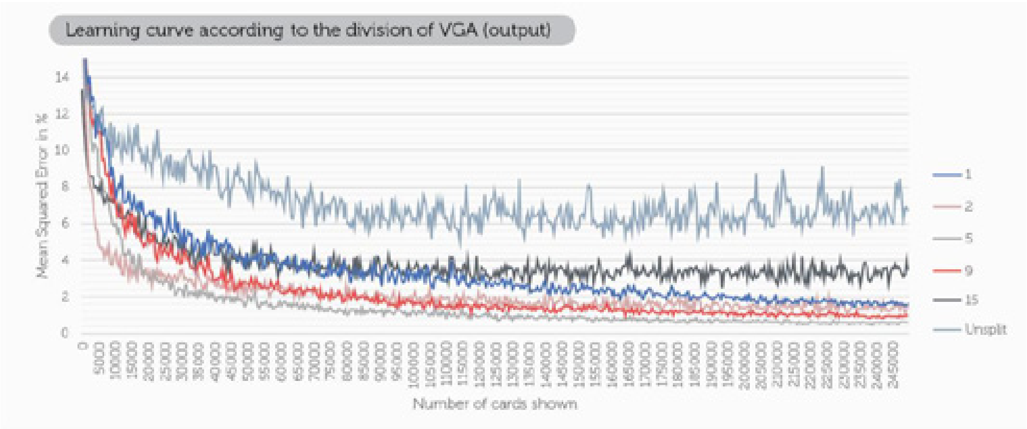
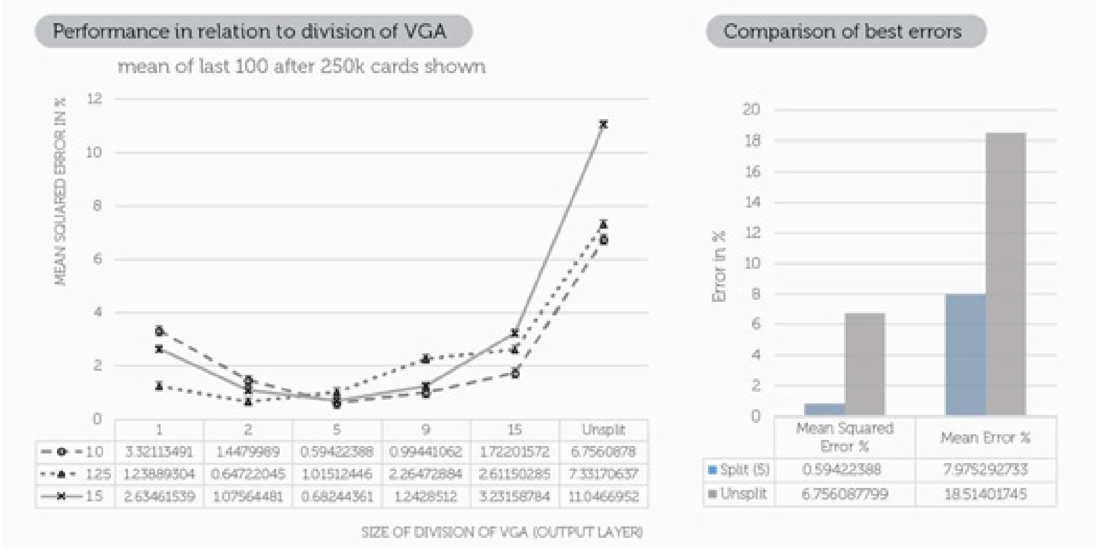


Figure 7 - Learning curve of six differently split systems.

Even though this procedure uses the input layer multiple times, the sum of all connections of the network is lower, as the size of the hidden layer is proportional to the output layer. These small networks can perform with less than 1% Mean Squared Error.

One question was related to the size of the overlaid grid. As Fig. 8 shows that the correlation between cell size and performance is not linear. The minimum of the curve is at cell size of 5 (meaning the grid consist of 5 x 5 cells). The reason this size performs the best is possibly due to the ratio of the cells sizes of the VGA grid to those of the boxes (also 5).



	1	2	5	9	15	Unsplit
1.0	3.32113491	1.4479989	0.59422388	0.99441062	1.72201572	6.7560878
1.25	1.23889304	0.64722045	1.01512446	2.26472884	2.61150285	7.33170637
1.5	2.63461539	1.07564481	0.68244361	1.2428512	3.23158784	11.0466952

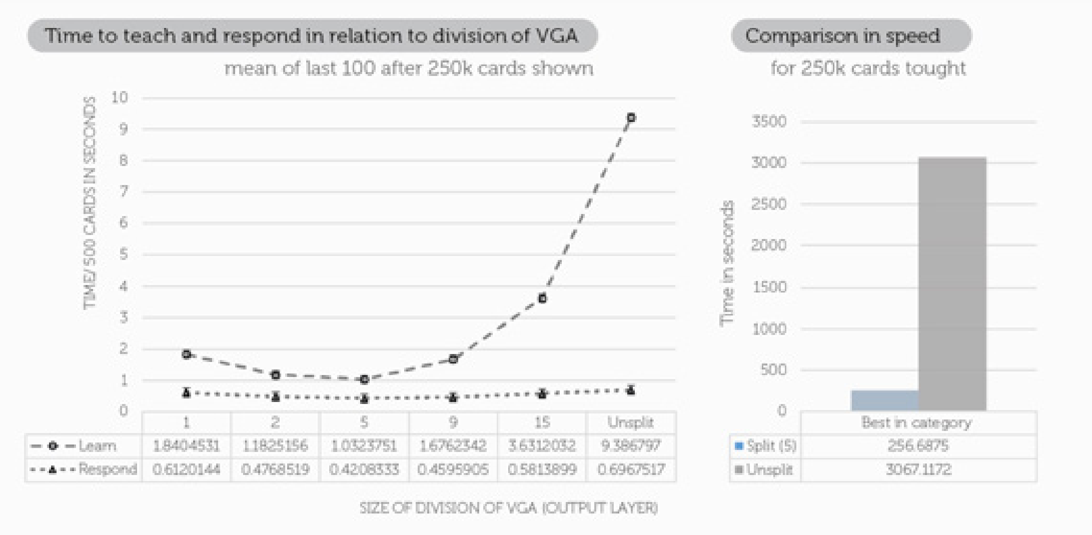
	Mean Squared Error %	Mean Error %
Split (5)	0.59422388	7.975292733
Unsplit	6.756087799	18.51401745

Figure 8 - Performance of the system according to how big the cells are of the VGA overlaid grid.

Another surprising outcome is that the smallest hidden layer size performed the best in most cases. The ideal system for this dataset was a cell size of 5 with a hidden layer size that is equal to the output layer. Thus, this system can be very efficient, consisting of fewer connections. This will contribute to a faster system – both in terms of learning and in term of reacting.

The efficiency of each network

Fig. 9 demonstrates how much faster split networks are. Unsplit networks require nearly 50 minutes to teach 250 thousand cards, while the best teaching time for split network (5x5, 1.00 hidden layer multiplier) is less than 4.5 minutes (257 seconds).



	1	2	5	9	15	Unsplit
Learn	1.8404531	1.1825156	1.0323751	1.6762342	3.6312032	9.386797
Respond	0.6120144	0.4768519	0.4208333	0.4595905	0.5813899	0.6967517

	Best in category
Split (5)	256.6875
Unsplit	3067.1172

Figure 9 - Correlation between cell size of overlaid grid on VGA to time for the system to learn and to respond.

As the hidden layer is dependent on the size of the input the questions arises: would this efficiency of split networks persist even if the area of building size would increase? The answer is that VGA has always an equal or larger grid (smaller cells) than the spatial arrangement it is analysing, therefore the overall connections of split systems of VGA will always be lower than those of unsplit ones.

How it performs with Clustering Coefficient

The performance of the MLP was also measured for Clustering Coefficient, which is one step more abstract than neighbourhood size, as it is a measure of the resulting VGA. As Fig. 10 shows, the network was able to learn rapidly for all networks that were split.

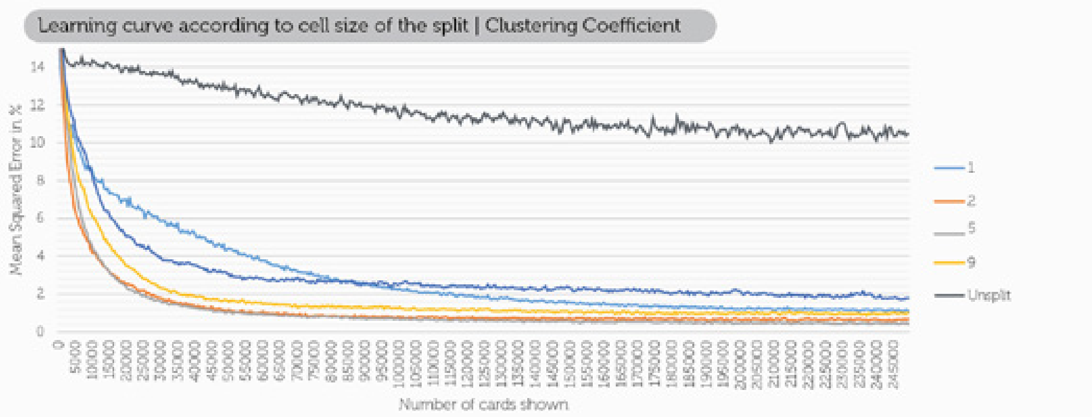


Figure 10 - The learning curve for Mode B when shown cards with Clustering Coefficient for different network splits.

How it performs in time compared to traditional method

The beauty of Artificial Neural Networks is that they are inherently a parallel computing system and can be easily multithreaded. The scope of this project didn't include the multithreading of the MLP, therefore a valid comparison is to measure both traditional system and the system created in this project on a single thread.

To calculate 10 VGA, the traditional calculation takes 40-50 seconds (about 15 seconds if run on multiple threads, while it takes less than 0.5 second to calculate it with the proposed system.

4. CONCLUSIONS

Three key characteristics describe the benefits of the proposed system.

Firstly, Artificial Neural Networks are able to calculate in parallel, which means the time it requires to calculate any solution is faster than methods developed for DepthMap. Although the accuracy of each system (shown in the result section) can not equal that of the traditional calculation, it is possible to set up systems for both Mode A and Mode B to achieve a negligible error margin (<1%).

Another key aspect is the system's universality. Even though a real public space was defined (Elisabeth Square, Budapest), both systems were also tested for generalised cases, which means they were set up and trained to just include a generic rectangular area on which spatial configurations were generated. This means the system only has to be taught once and it can be employed for any other space, therefore the computationally heavy and extensive process of generating data and teaching the network doesn't need to be repeated.

However, the system's limitation lies in its scalability. For cases when the resolution has to be finer for the same area or a larger space with the same resolution has to be analysed, than a separate network has to be used.

Lastly, the systems created are flexible. The reduction of the output layer doesn't influence the result of the calculation, so the system can be tailored for cases when only a smaller portion of the area needs to be analysed or generated.

4.1 MODE B

By splitting the discretised space into smaller areas the systems developed for Mode B, it was possible to generate VGA with high accuracy. As time is a crucial factor in the planning process, instantaneous analysis means that more design iterations are possible. Spatial analysis will also be used likelier, as accessibility is a key factor in adopting methodologies.

Speeding up the iteration also bring the values that a Genetic Algorithm (GA) or other multi-objective optimisation could integrate spatial analysis as an objective. Even though an optimisation algorithm could have implemented VGA as an objective with traditional calculation of VGA, it would run impractically slowly, as for e.g.: a GA has to evaluate immense amount of solutions to find the global minima of the fitness landscape.

During this project, an early prototype of an interactive system was created that shows how analysis could be embedded in the design phase, creating an interactive system.

Although Mode B allows for a faster calculation, the methodology of how to integrate it in a fitness function has to be worked out, more specifically what quantitative measure derived from the analysis should be taken into consideration. One suggested possibility could be to achieve a spatial configuration that has the highest visibility overall with the most built obstacles.

Further work

The MLP trained through the standalone programme could be tested with designers. The prerequisite is the integration of the trained network in an existing CAD environment (Grasshopper in Rhinoceros).

To see how far the methodology of this thesis reaches, other VGA measures, such as Mean Shortest Path or Point Depth Entropy have to be generated and tested. Axial Line graphs could be also shown to MLPs; this however would require a different methodology.

Another important question is whether the performances of these networks can be improved by introducing more hidden layers. These could be trained for combined measures to generate more types of solutions, for e.g.: along with clustering coefficient one could introduce parameters of building regulations, such as required distances between buildings.

5. CONCLUSION

This thesis set out to understand how a supervised, feedforward Artificial Neural Network, a Multilayer Perceptron (MLP) can be set up and taught to understand the underlying patterns of spatial configurations and their Visibility Graph Analysis (VGA). The aim was to provide a faster and more accessible tool for spatial analysis.

For this reason, two modes were explored: MLPs for mode A were created in order to generate spatial configurations for VGA as inputs, while the MLPs in mode B aimed to generate accurate VGA, both neighbourhood size and clustering coefficient.

The main contribution of the research is to show that space can be intelligible for an Artificial Neural Network and that current methods of calculating VGA can possibly be exchanged by MLPs.

More specifically the research found that for mode A a three-layer MLP was able to generate valid solutions, for specific variation of inputs. The main experiments were conducted for the purpose of understanding what properties of input arrangements influence the performance of generating solutions. A set of guidelines, such as symmetrical selection of inputs, were derived from these experiments.

Experiments for mode B came upon a method of creating MLPs that contribute to a substantial increase in performance, a decrease in teaching and in responding time. The main property of the method was to split the area into parts and create individual MLPs for each area.

To conclude, the thesis was able to take a step in a direction that if explored could yield additional improvement alternative methods of implementation for VGA and possibly for other Space Syntax methods, such as Axial Line Analysis.

REFERENCES

Al_Sayed, K., Turner, A., Hillier, B., Iida, S., & Penn, A. (2014). Space Syntax Methodology. *Book*, 1– 117.

Al-sayed, K., Dalton, R. C., & Ho, C. (2010). Discursive design thinking : The role of explicit knowledge in creative architectural design reasoning, 211–230. http://doi.org/10.1017/S0890060410000065

Cohen, H. (1984). What is an Image?

Cope, D. (1999). Facing the Music: Perspectives on Machine-Composed Music. *Leonardo Music Journal*, 9(1999), 79–87. http://doi.org/Article

Flake, G. W. (2001). G.W. Flake, The Computational Beauty of Nature. *Artificial Intelligence*.

http://doi.org/10.1016/S0004-3702(01)00076-5

Happold, S. S. team at B. (2015). SmartSpaceAnalyser. Retrieved August 18, 2016, from http://www.grasshopper3d.com/group/smartspaceanalyser

Hillier, B., & Julienne, H. (1984). *The social logic of space. The social logic of space*.

http://doi.org/10.1016/0169-2046(86)90038-1

Mcculloch, W. S., & Pitts, W. (1943). a Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5(1), 99–115. http://doi.org/10.1007/BF02478259

Senatore, G., & Piker, D. (2015). Interactive real-time physics : An intuitive approach to form- finding and structural analysis for design and education. *Computer-Aided Design*, 61(April), 32– 41.

http://doi.org/10.1016/j.cad.2014.02.007

Turner, A. (2001). Depthmap: a program to perform visibility graph analysis. *Proceedings 3rd International Symposium on Space Syntax, 30* (May), 7–11.

Weaver, W. (1948). Science and Complexity. *Foundation, Rockefeller City, New York*, 536, 1–11.

Wilkinson, S., Bradbury, G., & Hanna, S. (2015). Reduced-order urban wind interference, “Simulation: Transactions of the Society for Modelling and Simulation International,” (91(9)), 809–824.