COGNITIVE ALGORITHMS IN STRUCTURAL OPTIMIZATION

Martin Kaftan
Molab, Faculty of architecture, Jugoslávských partyzánů 1580/3, Czech republic, aquaplas4@yahoo.com

Summary

The subject of investigation is to rationalize geometry from an unorganized point cloud by using artificial learning neural network. Furthermore, the focus is oriented upon aspects of efficient construction of generated topology. Neural network is connected with constrained properties, which adjust the members of the topology into predefined number of sizes while minimizing the error of deviation from the original form.

Keywords: Parametric design, Artificial neural network, unorganized point cloud, space frame.

1 Introduction

Unorganized point clouds can contain millions of points in space to represent geometry. Even though they demonstrate the shape in overall, each point has no relation to the others, containing only knowledge of its own spatial position. Therefore, just to reconstruct precise geometry from unorganized point clouds is a difficult problem, even more if we add other requirements, such as optimization of the geometry for manufacturing purposes. A common example of unorganized point clouds derives from 3D laser scanners or photogrammetric image measurements, which are often incomplete, sparse and noisy data. To construct from an unorganized point cloud requires highly efficient algorithm with adequate processing speed.

Over a century, architects and engineers have been pursuing self-generation through different techniques and forms. Part of the research process has been engagement with generative models inspired by fields, such as biology.An example of common source of inspiration is the research work of D'Arcy Thompson. In his book On Growth and Form, Thompson developed understanding of living forms in terms of physics, studying affection of dynamics on organisms and their transformation through growth and movement. Thompson demonstrates that form of organisms is obeyed by engineering principles of continuous variety in biological materials such as structure of bones where some parts of the bone's structure deal with tension while other with compression (Thompson, p.230-238). And recently, architects have become interested with new possibilities of generative exploration in design through emergent self organizing algorithms, such as are Boids, Cellular Automaton or Genetic Algorithms. These emergent systems produce complex global behaviour derived from local interaction between large amount of simple parts. One of the interesting computational emergent systems are the artificial neural networks.

The term Artificial Neural Network in generally describes apparatus that consists number of interconnected neurons. The Artificial Neural Network reflects some apparent similarities with the way the neural cells are interconnected in the human brain. The

neurons communicate with the neighbouring neurons, receiving and transmitting information with connections called synapses. Each neuron of the network has capacity to store and process some information. Often, the synaptic connections also store some information, usually in the form of scalar weights. Neural networks are in practice most often simulated by a software, even though a neural network can be implemented as hardware, such as attempts on neural processors. In the last twenty years neural networks were applied in numerous of applications in different science fields, such as pattern recognition, finance, bio-informatics and others. A typical application for a neural network, as for example the hand-writing recognition, is a simple problem which the human brain almost effortlessly solves, while the computer, despite its computational power, faces unexpected difficulties and requires sophisticated software for sometimes not very satisfactory results.

2 Artificial Neural Networks

2.1 Hypothesis

Acknowledging that the neural networks are capable to operate with large amount of data they seem to be suitable to solve the problems of construction from unorganized point clouds.

The focus is then on the following research questions:

1) Suitability of neural networks to generate geometry from unorganized point clouds
2) Adaptation of neural networks algorithm to adjust the structural members of thegeometry to predefined standard sizes for material manufacturing purposes

2.2 Models for experiments

The experiments are mostly first tested in 2D environment and if successful they are proceeded into 3D. To receive comparable results, the experiments were mostly performed on standard models. The 2D environment consists of input data randomly selected for each iteration from inside of a rectangular boundary. The amount of points is therefore immensely large, and the point cloud evenly covers that area. This provides good comparison of influence of good distribution of input data to other used models with lower data resolution.

For experimenting with more complex geometry the point clouds were obtained from the British sculptor Antony Gormley who experiments with creation of different types of geometrically based topologies from 3D body scan. (Fig.1, Fig.2); The data are text files with xyz coordinates received from the 3D scanner, the same data format, that Gormley uses for generating his sculptures. The scanned objects are human bodies. The point clouds comprise two resolutions of the identical body and position; one is 2, 000 points resolution and the other 20, 000 points. Extracted parts of the point clouds such as only the head part are also used.

2.3 Designed neural network

An artificial neural network can be trained in a learning process. In this case the neural network processes the input signal by adjusting itself, and the state of the network is

considered the output. The adjustment of the network during the learning process can include changes in the information stored on the nodes and the connections, as well as changes in the architecture of the network. That is, the network adapts its state to the input signals by creating new nodes and connections, or/and by adjusting the existing ones. One model of neural network is the Growing Cell Structure algorithm developed by |Fritzke. It combines advantages of Hebbian's constantly self-organizing topology and Kohonen's adaptation to the input. It is an incremental neural networks that don't have predefined structure. Instead, the structure is generated by adding or deleting neurons in the network during the learning process. Each network consists of a number of neurons and number of synapses and each neuron has associated position with a stimulus. Upon receiving signal from inputs (point cloud), the nearest neurons (winners) are repositioned toward the signal and after each repositioning error factor is calculated for each winning neuron . The accumulated error determines after predefined number of iterations where to insert new neurons.

The neural network generates topology as the modification of the Delaunay triangulation which is the result of the embedded Competitive Hebbian Learning algorithm. A neural network implements ageing process to remove connections in the network that is not part of the Delaunay triangulation. After a connection is updated or a new connection is created by the Hebbian, the edge of the connection is set back to 0, signalling to the network that the connection is still active part of the network. If a connection reaches certain predefined value of its age, it indicates to the network that the connection is inactive and is removed from the network. If the removal of connection results in neuron without connection and the neuron is not reconnected after certain number of iterations, the neuron is proclaimed as inactive as if removed from the network.

The growing neural network shows two basic differences from the Kohonen neural network:

1 Only the best-matching unit and its direct topological neighbours are adapted.

2 The adaptation strength is constant over time. For adjusting strength of connection are used two constant adaptation parameters , one for the best matching neuron and the second for the neighbouring cells. The algorithm has also utility factor which if switched on has the ability to adapt to changing distribution by changing location of less useful neurons. As in the ageing process for the connections, the algorithm saves an utility factor for each neuron which determine its success in the global growth process. If the factor reaches predefined maximum value the neuron is not deleted, but its repositioned in place with the highest error factor and reconnected by the Hebbian back to the network.

The overall operation progress of the growth can be described in following steps (Oja,p.131-140):

1) First two connected neurons are positioned randomly in the environment. Their error value initialized as 0.
2) From the point cloud is randomly selected a point - the stimulus.
3) By measuring the Euclidean distance between stimulus and each of the neurons is determined the closest neuron - the winner and the second closest neurons - the neighbours.
4) The algorithm checks if the winner and the neighbour has already connection, if not, it creates it and sets age of the connection to 0.
5) The error value of the winner is updated by adding squared distance between the stimulus and the winner to the already existing error value.

6) The position of the winner and its topological neighbours is adapted by fraction of the distance to the stimulus.
7) The age of all connecting synapses between the winner and the neighbours is updated incrementally.
8) If there are synapses with an age higher then the maximum allowed value, those synapses are removed. If there are any neurons without synapses, these neurons are removed
9) If the number of neurons is less than the allowed maximum of neurons a new neuron is inserted by:
   a) A neuron with the largest error is found in the network
   b) Neighbours of this neuron are located and the neighbour with its largest error is selected
   c) New neuron is position between those two selected neurons and new edges are created between the new neuron and the selected neurons; the old edge is removed. The error of selected neurons is decreased by a fraction and error value is given to the new neuron.
      1) The error of all neurons is decreased by a fraction.

2.4 Testing the growth process

The constructed algorithm from the pseudocode is tested first in 2D dimension (Fig.1,2,3). The network samples stimuli from the rectangles. As already demonstrated by Fritzke, the network begins with positioning neuron to each of the rectangle and connecting all of them with the synapse. As the computation progresses, the algorithm adds more neurons inside the rectangles. Eventually, after number of iterations the neural network disconnects the synapses between the rectangle. The images also demonstrate non-induced Voronoi geometry.

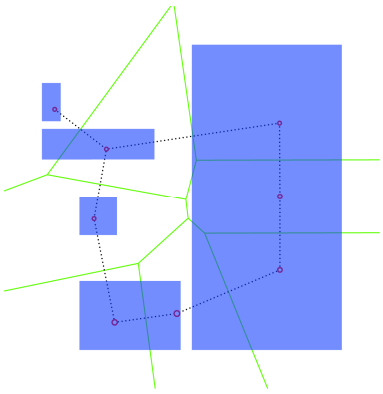

Fig. 1 30 iterations

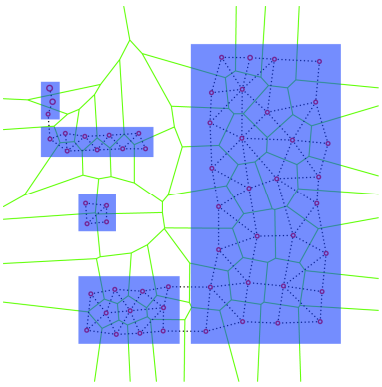

Fig. 2 160 iterations

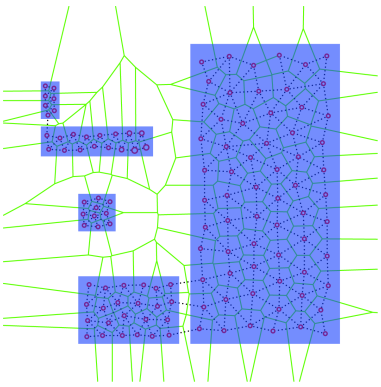

Fig. 3 380 iterations

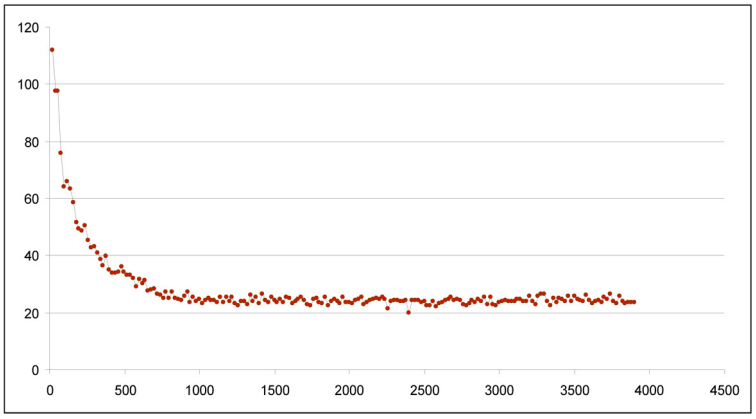

Fig. 4 Learning curve of the network. The curve is represented by decreasing average error value, which means that the network successfully adjusts its position close to the stimuli.

2.5 Geometry optimization

Because the nature of the incremental neural network is to constantly operate with an error, each member of the structure has a different length even during dense stimuli distribution. However, for the construction purposes we want to have a limited number of different member sizes. Therefore, we have to add constrains to adjust each member into one of the allowed sizes. The main problem of using any constrained parameters with growing neural network is that it directly affects the node placement and the mapping precision of the algorithm . As already described, the algorithm functions on placing neurons based on selecting the highest error factor, in this case the Euclidean distance between the neuron and the stimulus. If any constrained parameters are added during the incremental process, it limits the network's flexibility needed for correct learning and leads to fatal deviations and wrong node placement.

The adjustment of the lengths referring further to as the "Tuning Process" consists of two main stages:

1) Setting the travel distance of adjustment of a neuron towards the input affects how well the networks distributes the neurons. If the travel distance is too small the network does not spread well and the neurons are close to each other (Fig.5). However, if the travel distance is set high, the network distributes well, but in the case of a well distributed point cloud the range from its average length is greater; the neuron does not stabilize itself, but travel constantly between surrounding signals. For our purpose is desirable to switch from high to low travel distance during the process. The algorithm starts with high distance and when it reaches the desirable maximum amount of neurons (Fig.6), it switches to low distance. Because at that point the network is already well distributed on the point cloud, the high travel distance is no longer necessary and the low distance travel causes the network to spread more uniformly its existing neurons (Fig.7). That narrows the ratio from the average length.

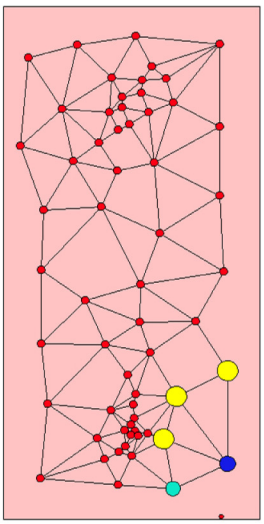

Fig. 5 poor distribution travel distance

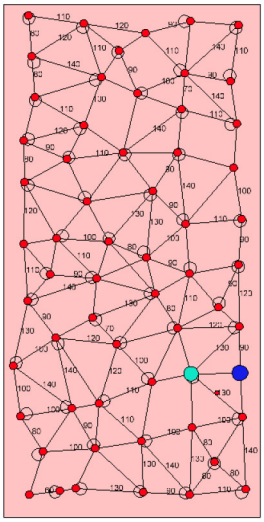

Fig. 6 poor distribution before tuning

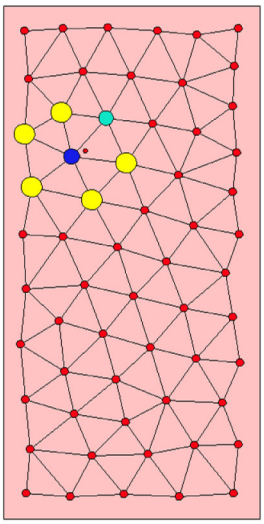

Fig. 7 poor distribution low after tuning

2) In the second phase the structure changes the adapting of its members/synopses into predefined members of standard sizes (Fig.15). The algorithm works in following steps:

a) The reaction of neural network to the stimuli is interrupted, therefore the geometry no longer reacts to the point cloud. The current position of all the neurons is stored in an array.

b) For the best results and the smallest deviation of the mesh, the average length of all of the member sizes is calculated which is considered to be 100 units. If the stopping mechanism of the growth was based on reaching the wanted size of the average length and not the maximum number of neurons, the lengths of all the members were already defined before the growth.

c) Each of the structural members are categorized into predefined dimensional ranges, based on the closest deviation from one of the desirable lengths. Each category are based on the range between the middle value of the closest lower length and the middle value of the closest higher length.

d) The standard maximum allowed deviation is calculated, that is how much is a neuron allowed to distant itself from the original position. The value of maximum error deviation is determined by the percentage from the average length. In 2D the deviation represents the radius for the circular area of allowable movement of neuron from its original position, in 3D the neurons allowable area is given by spherical volume with its center on the origin.

e) A sweeping algorithm selects a random neuron (the winner) and finds its closest connecting neurons in the valence (neighbours). From the neighbours are chosen randomly two neurons and their distance from the winner is checked. If both of the lengths have already wanted size, another neuron in the valence is selected and checked again. IF one the lengths deviates from the determined length, its closest locking length is selected from the catalogue.

f) Algorithm calculates the displacement of the center of the valence in such a way that only one member adjusts its length into desired one while the length of the second member remains unchanged. The calculation consists of finding intersections of two circles in the 2D, two spheres in the 3D scenario. One of the radius is unchanged member length, the second is the locking length.

g) algorithm checks the error deviation between the winner's stored origin and the closest found intersection - if the needed change of position distances neuron from its origin less than maximum allowed deviation, the neuron is positioned - if the needed position is more than the allowed deviation, the algorithm returns back to f) andselects the lower catalogued length, and repeats finding the intersection with the new length. If this point of intersection is in the allowed deviation range, the neuron is positioned by adjusting the length randomly through out the structure, the off-lengths are better distributed.

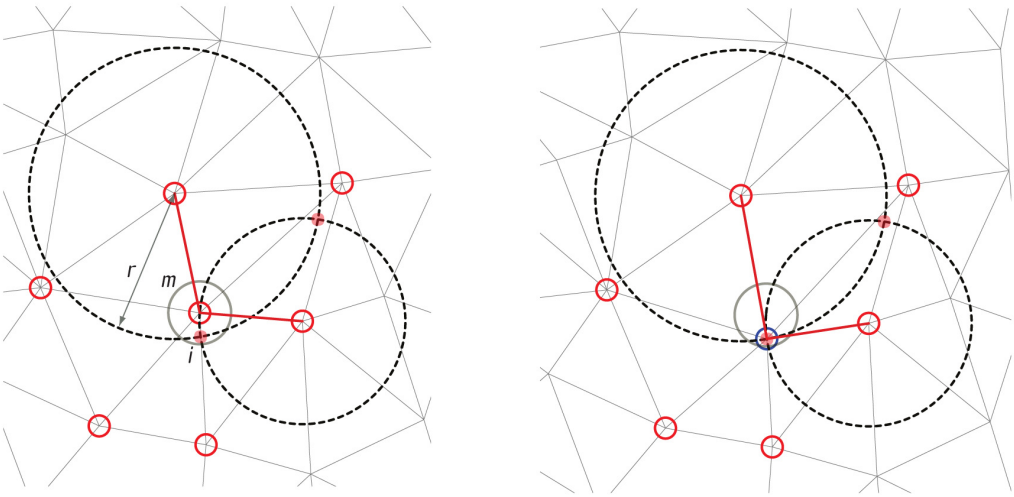

Fig. 8 Locking into predefined sizes.
r = radius of the closest desired member size
m = area of maximum allowable movement from origin
i = successful closest intersection < m radius

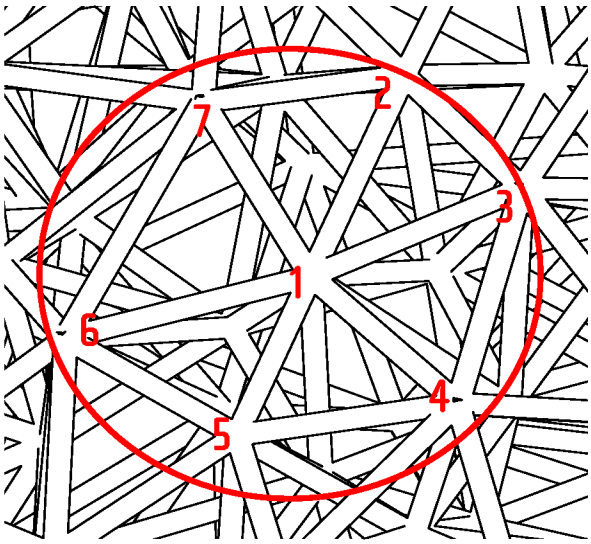

Fig. 9 Well distributed geometry

2.6 Efficiency

To see the efficiency of the tuning, the process is repeated under same conditions with several different length sizes :

The maximum number of neurons = 60
Number of samples = 50

Lengths (#)	Maximum Allowed Distance	Adjusted members(#)	Remained members(#)	Success (%)
25	15	101	38	72.7
13	25	84	55	60.4
7	30	71	68	51
5	35	67	72	48.2
3	45	61	78	43.9

As can be seen from the table, the efficiency increases with more given lengths. Also the deviation from the original shape decreases. With 25 different length option, the mesh adjusts at around 70% of success. The maximum allowable distance was adequately changed according to number of lengths because less number of sizes need to travel greater distance to lock into positions than higher number of lengths.

The experiment is repeated in 3D environment to compare the tuning success..

The maximum number of neurons = 60
Number of samples = 50

Lengths (#)	Maximum Allowed Distance	Adjusted members(#)	Remained members(#)	Success (%)
25	15	109	49	69
13	25	76	79	49
7	30	69	89	43.6
5	35	65	93	41
3	45	61	97	38.4

The results show lower success than in the 2D model. This is caused by less even neuron distribution over the point cloud, and its more difficult for the algorithm to lock into the positions. The 25 member sizes show still around 70 % while the other, less member sizes drop rapidly; the 13 member sizes adjusts only around 50 % of the total lengths.

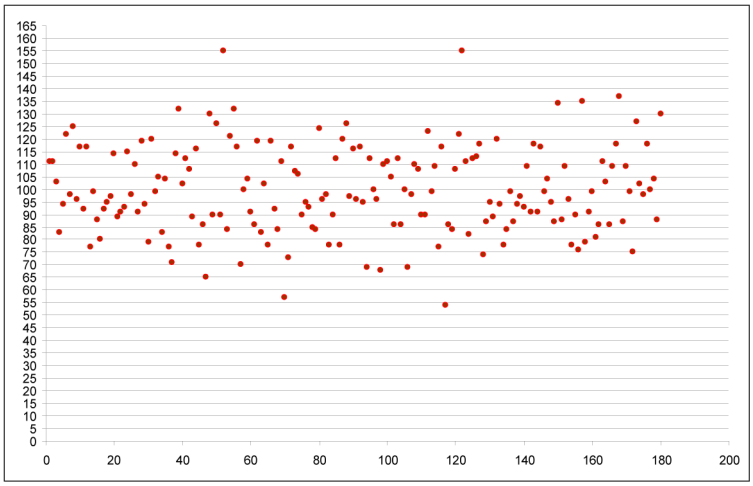

Fig. 10 Member lengths before tuning – 3D, 25 sizes

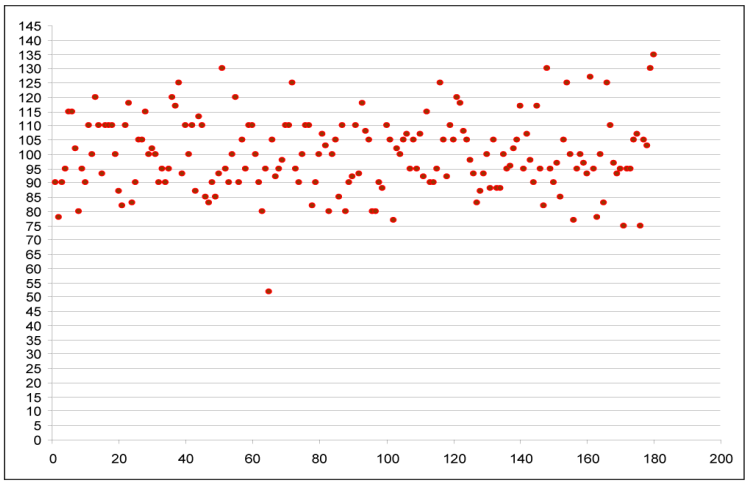

Fig. 11 Member lengths after tuning – 3D, 25 sizes, 66% successful adjustment

3 Conclusions

Regarding the first question of the hypothesis concerning the rationalizing of unorganized point clouds , the results show to be satisfactory. The adjusted incrementally growing neural network correctly creates the geometry. The speed of the algorithm proved to be very satisfactory, having practically no upper limit from the size of the input data. Even more, the higher is the resolution of the point cloud, the better and more even is the resulted geometry of the neural network.

The second task to improve construction possibilities from the point cloud proved also successful.The additional algorithm to the neural network demonstrated between 50-70 % success in the adjustment of members, depending on the chosen amount of standard lengths and the form of the geometry. The results were more successful in the cases of surface geometry then in the volumetric. The reason for this is the more constrained topology of layered network containing more connections. As a tool for the manufacturing purposes, the designer can chose the number of different amount of lengths

which affects the deviation error from the unadjusted geometry. More lengths results for less needed readjustment of original topology and better results of adjustment. For the practicality of the use for the construction the fabricator can export the sorted lengths as a text file and the geometry with correlated numbered members as a DXF file.

This outcome has been supported by doctoral grant of Czech Science Foundation 103/09/H095 „Sustainable Construction of Buildings and Sustainable Development of Urban Space“.

References

[1] Fritzke, Bernd. *A growing neural gas network learns topologies.* In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, Advances in Neural Information Processing Systems 7, pages 625-632. MIT Press, Cambridge MA, 1995.

[2] Fritzke, Bernd. *Growing cell structures - a self-organizing network for unsupervised and supervised learning*. Neural Networks (1994), Vol. 7, No. 9, page 1441-1460 .

[3] Oja,E. Samuel,K. *Kohonen Maps*. Elsevier, New York, 2003.

[4] Thompson, D.W., *On Growth and Form*, Cambridge University Press,Cambridge, UK, 1942.