Artificial Neural Networks

> *"If you try and take a cat apart to see how it works, the first thing you have on your hands is a non-working cat. Life is a level of complexity that almost lies outside our vision."*
>
> *Douglas Adams.*

Artificial Neural Networks (ANNs) are a mathematical model inspired by the functioning of the brain. As in nature, neurons which are weighted and synapses fire when stimulated as the network gradually *learns* over time.

The human brain can be seen as a massively parallel distributed system of these simple neurons, which only have local information and no global knowledge of their own. The fact that order should emerge from this agglomeration is certainly intriguing, and is the principle study of connectionism in neurobiology. Thus the whole is more than the sum of its parts and is not easy to untangle.

It is now known that some multi-dimensional inputs such as sensory experience have associated three-dimensional locations in the brain. The question is how do these structures in the brain manage to process such multidimensional signals? Put another way: how is a multidimensional input projected onto a three-dimensional (i.e. lower) neuronal structure?

Such mappings have been found to exist, for example in the perception process of the human eye where properties of an image are mapped directly to an area of the brain (figure 1). A form of cognition can thus be simulated by replicating the behaviour of the brain and its massively parallel architecture. Some Artificial Neural Networks attempt to artificially recreate this by mapping complex inputs into a lower dimensional space.

Neurons in an artificial neural network consist of layers – usually an input, an output, and some hidden layers to manipulate weights. A neural network will typically self-organise by nodes firing when stimulated and subsequently influencing their topological neighbours via synapses.

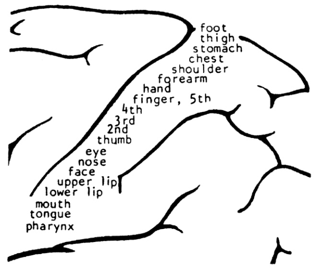

Figure 1: The somatotopic map shows rough areas of the brain being mapped with their topological structure on the body more or less retained.

	animal	dove	hen	duck	goose	owl	hawk	eagle	fox	dog	wolf	cat	tiger	lion	horse	zebra	cow
is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

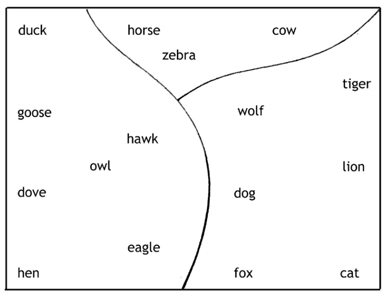

Figure 2: The Self-Organising Map reduces 13 characteristics onto the plane. The result is a map preserving relationships at higher dimensions. (for example the fox is similar to the cat, but not a duck).

The subdividing lines are drawn on later by visual inspection.

1. Learning Types

Neural networks typically fall into two categories: supervised and unsupervised. The supervised version requires some knowledge of the inputs and the output, so that the system may be *trained* to complete a particular task. An unsupervised neural network has unpredictable outcomes with no direct goal in mind, and hence somewhat 'autopoietic' in nature.

2. Relevance to Architecture

The traditional role of the architect can be seen as an organiser of objects in space in order to constitute some whole known as a building. The site where these 'buildings' are located is called the environment, but the users of the building are also part of this environment too. The people and other living creatures are dynamic, involving complex social relationships, and political interactions.

On some levels the architect can be seen as a dimensionality reducer- an organiser of space in three dimensions. In the organisation of a building, a brief usually has a list of constraints and requests, for example with building regulations and client requirements. The architect thus digests this information, attempts to reduce its complexity and make an informed interventions at many different stages of any project. In terms of reducing the complexity of the information, ANNs remain an un-tapped research topic in architetural design.

Artificial means of reducing complexity to a level of comprehension, reflection, and action by a machine are possible using dimensionality reduction. These methods can be seen as a new way of introducing the methods of connectionism and cybernetics into the design process, and seeing how computers can assist humans in the task of design.

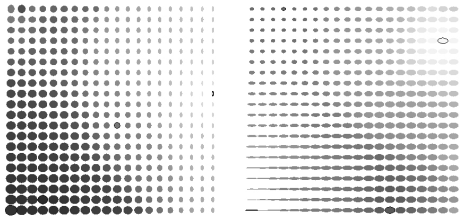

Figure 3: Two examples by the author of an SOM. Each node has an associated 3-vector containing ellipse width, height and grey shade information.

3. The Self-Organising Map

The 'SOM' was first presented by the Finnish computer scientist Teuvo Kohonen as the culmination of his research on associative memory techniques. It is one such process aimed at reducing complexity of high-dimensional inputs to lower level for easier comprehension.

The map attempts to replicate the functionality of the brain by setting out, in most cases 2d array of nodes (neurons) in a rectangular connective topology (synapses). The aim is to organise a given number of inputs with high dimensional data by comparing differences in an associative manner. The SOM is unsupervised in that no specific end result is presented for the network to pursue.

The maps are usually two or three dimensional for visual purposes, although this is not always the case. Each neuron in the map array has an associated 'synaptic vector' which is the same dimensionality as the input samples. Each value in this vector represents a feature.

As well as visual applications, some situations in mechanics require *actions* that may be lower dimensional themselves, but dependant on making sense of high dimensional inputs. One such example is in the control of robot arms in order to complete certain tasks as described by Kohonen [1] and Pouzet. [2]

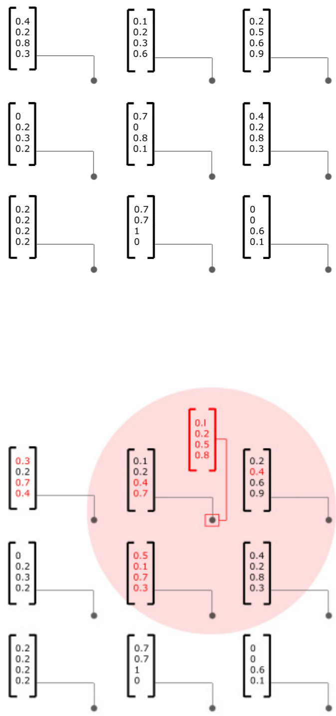

Figure 4: Each neuron has an associated 'synaptic' vector the same dimensions as the inputs. A winner is located by comparing the Euclidean distance in the input space.

In the standard SOM, a minimum radius is then set in order to adjust the weights of the neighbours.

3.1. Coding of Inputs

The map is presented with a sample set of inputs, each with a set amount of features. In the example shown in figure 2.2, this feature list contains 13 entries. The features are usually normalised in order to give a fair distribution between one another, or represented as a binary string as in Kohonen's own example of the encoding of various animals into an input 'codebook.'

3.2. Winning Node

When the inputs are presented, the node with the closest synaptic vector is declared the winner. Determining this distance can be done using various methods, including Hamming distance (binary comparison) and by finding the dot product of vectors (for small input dimensionalities). The most common method however is to take the smallest Euclidean distance in order to determine the winner, i.e. the distance in input space:

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}.$$

(Euclidean distance from an input vector **p** to an SOM node's synaptic vector **q**).

This winning node then adapts its feature weights slightly towards the input. In the standard SOM, the neighbours are influenced too based on a Gaussian radial basis function – the so called 'Mexican hat.' This can be extended to include inhibitory feedback for neurons that are distant from the winner. (see figure 2.5)

Figure 2.2 shows the resulting map from the 13 dimensional animals. The representation is 2d, i.e. on the plane – but preserves the characteristic differences present at the higher input space.

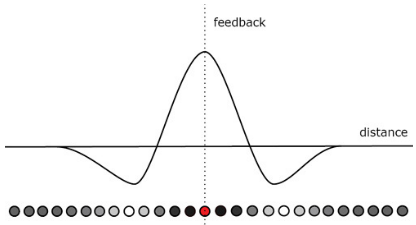

Figure 5: Diagram showing how the Mexican hat function controls the stimulation of adjacent nodes. Feedback is either positive or inhibitory dependant on the distance from the winner.

3.3. Geometrical Maps

Some architectural applications of SOMs involve geometric coordinates to be both inputs and synaptic vectors [3]. As input data is usually three-dimensional (though this is not always the case), the output can also be represented in 3d. This gives rise to visually dynamic geometries as the learning process takes place.

Such mappings in an architectural context were first proposed by Derix in 2000 [4]. Vertices of existing buildings are used as training data for an extended SOM that produces new mappings of space. These structures reveal patterns and relationships as cognized synthetically by the machine, revealing themselves visually to the human observer.

3.4. Topology

The topology of the network relates to the location of the synapses/ connections between the nodes at a given time.

The standard SOM uses a rectangular or (as recommended by Kohonen) a hexagonal topology for the synapses. It is fixed, that is, it has no time dependency throughout the learning process. Some simple dynamic techniques, some suggested by Kohonen himself, can extend the original SOM. These include minimal spanning tree topologies between the nodes, or growing topologies which insert or delete rows of neurons.

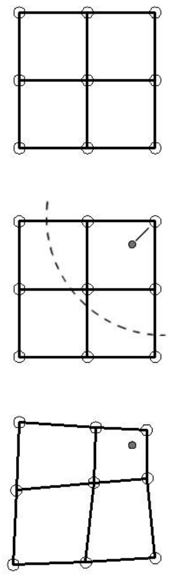

Figure 6: An input is generated at the same dimensional space as the map itself. The winner moves towards the input, whilst inhibitory feedback repels the bottom left hand node away.

3.5. Limitations of SOMs

Although the Self-Organising Map does indeed function well for static inputs, it struggles for inputs that are dynamic, for example let's say that one of Kohonen's animals grows extra legs whilst the simulation is running - the SOM cannot adjust appropriately.

The parameters for learning and the size of the map appropriate for the inputs must be determined beforehand. This requires some prior knowledge of what inputs are to come and hence the SOM can be inflexible. As the learning parameters gradually cool, the SOM finishes its learning and then stops. It is a one-time process.

Another issue is that clusters are identified manually as in the case with animal example shown in figure 2.2. Subdivisions are drawn on after the mapping is finished.

Figure 2.8: Some SOMs with RGB inputs by the author. The map only has memories of what it has seen; hence the full colour spectrum is not represented.

3.6. Meta Learning

A combination of using two SOMs has been investigated recently in the architectural design process by Ireland & Derix [5]. This approach uses an 'meta-level' SOM to compare various other SOMs representing living patterns specific to the individual. In order to arrive at a final form groupings are made for similar activity types in order to merge the boundaries between them.

4. Growing Neural Networks

The various issues concerning the self-organising map led to some further extensions to Kohonen's model. The first issue to be tackled was to free the map from its fixed topology. There are various methods around, for example to GSOM method [6] and the Kohonen's own variations using dynamically defined topologies [1]. However these still required an initial set topological structure (e.g. rectangular), usually with the added feature of rows and columns being able to be added or removed.

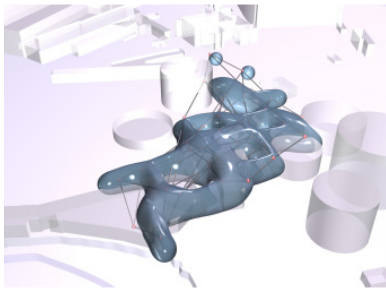

Figure 7:
Self Organising 'Phenotypical network' by Derix [4].

4.1. Neural Gas

The so called *Neural Gas* was first introduced by Martinez and Shulten in 1991 [7]. As opposed to the fixed grid on the SOM, the method aims to represent the topological clusters from the input space in the learning process, and hence is a *Topology Representing Network* (TRN).

On latter versions of the neural gas, 'Competitive Hebbian learning' is used. That is, the winner is adjusted towards the input using a given rate parameter, with the second place also being adjusted towards the input, albeit to a lesser degree. A topological connection is then made between the winner and the second place node.

When the neuron subsequently fires, only local neighbours receive a signal dependant on the current topology of the network. This is in contrast to the radial neighbourhood function used with the SOM (as shown in figure 2.4).

Although improvements are made on Kohonen's model, the number of nodes must be predetermined, and the learning parameters still converge- hence the process cannot be run indefinitely. There is no dimensionality reduction either, so the applications of the method are limited.

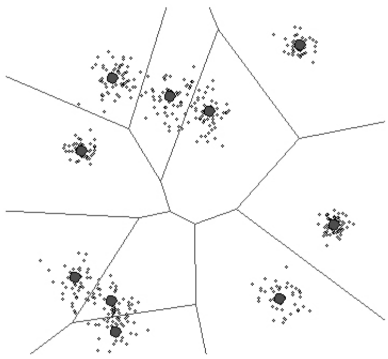

Figure 8: A Neural gas forms clusters around input nodes. Each cluster is then represented by a Voronoi region done by visual inspection afterwards.

4.2. Growing Neural Gas (GNG)

An extension to the method was proposed by Bernd Fritzke [8] in order to address some of these limitations. To further represent areas of high input probability, nodes are added in these areas. This is achieved by each node containing a cumulative error tag that is re-calculated each time it is winner. If errors become high, the input density is not being represented sufficiently, and a new node is required. Conversely, areas receiving little or no signals remove nodes.

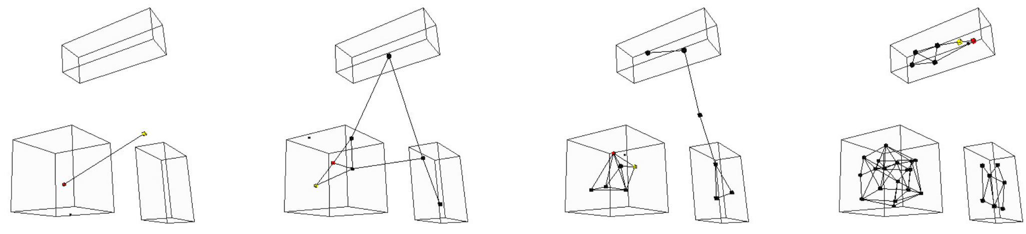

Figure 9: example by the author of a growing neural gas. Inputs were given randomly inside the cube areas. Notice how the gas has mapped the topology of the cubes, and has three distinct components.

Winner nodes and second place nodes are joined with an edge. Edges that are adjacent to these nodes (nearly winners) are aged. At the end of each loop, edges above a certain threshold die out as well as any associated nodes.

Again, as with the neural gas, the dimensionality of the network is the same as the input space, so visualising the network at higher than three dimensions is not possible. One improvement on the original gas model however, is that the algorithm will run indefinitely, as there is no need for a decaying learning parameter due to the error tags.

The mapping of dynamic inputs is also a struggle for the growing neural gas. This problem arose with the author's own simulations (see figure 2.12). Connections do not age since aging is only performed for edges adjacent to the winner [9].

Non stationary data distributions can be found in many technical, biological, economic and indeed architectural applications such as here– hence the need for a development to the GNG.

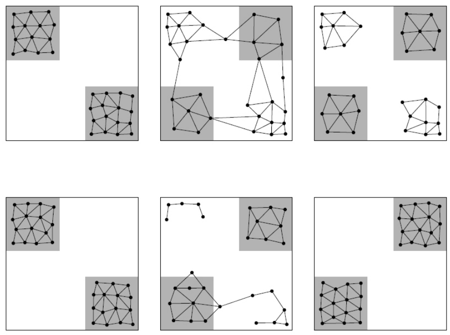

Figure 2.10: Top: A growing neural gas tries and fails to track a dynamic input (inputs come from the shaded square that moves halfway through the process.

Bottom: A GNG-U algorithm successfully completes the task.

4.3. GNG-U

This inability to map dynamic inputs led to an extension to the GNG by Fritzke (1997) to his model to include a 'local utility measure' for each edge again dependant on cumulative error for input space. The author has used his own simpler version however which achieves a similar goal to a sufficient level (see section 3.2).

Figure 2.11: shows how the GNG-U algorithm successfully adapts to the moving signal space.

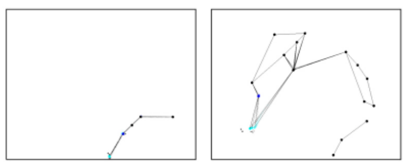

Figure 2.11: gng simulation. Inputs are initially generated in the bottom right hand corner. However, when inputs are moved to the left, a remnant remains.

4.4. In Architecture

A GNG is used by Langley [19] to generate mappings of dynamic activity in existing urban environments. Event mapping of this kind using a neural gas has also been investigated by Parvin [4]. These remain the only published applications of a GNG in architectural design to the best of the author's knowledge.

5. Growing Cell Structures

The previous GNG-U model can successfully map dynamic inputs, however in none of the methods above are the advantages gained through dimensionality reduction in the SOM present.

The precursor to the GNG however, known as the growing cell structures is able to do such as task. The learning part of the algorithm is the same as the GNG, but instead of nodes and edges, the network consists of a more rigid topology of nodes and hyper-tetrahedrons. This means a triangle If the embedding space is 2d, and a tetrahedron for 3d.

Each node has both a representation in the embedding space (i.e. for two-dimensions and x and y coordinate) and one for the input space (a synaptic vector of high dimensionality).

Winners and connections are determined in high dimensional space, but connections are retained for the structure in the plane. In the images below, the left hand side shows a typical network with inputs in 3 dimensions, with its associated partner embedded in the plane on the right with the same topology. Images (a) and (b) show different stages of the simulation. Notice how the topology is preserved in a later stage of the network (b) in that there are two distinct structures.

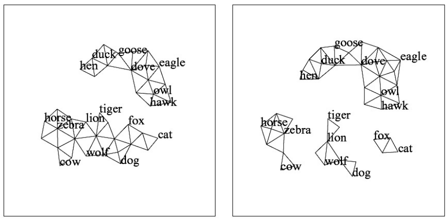

Figure 2.12: Growing Cell Structures with Kohonen's inputs. Clusters between animals are found automatically, as opposed to the Self-Organising Map.

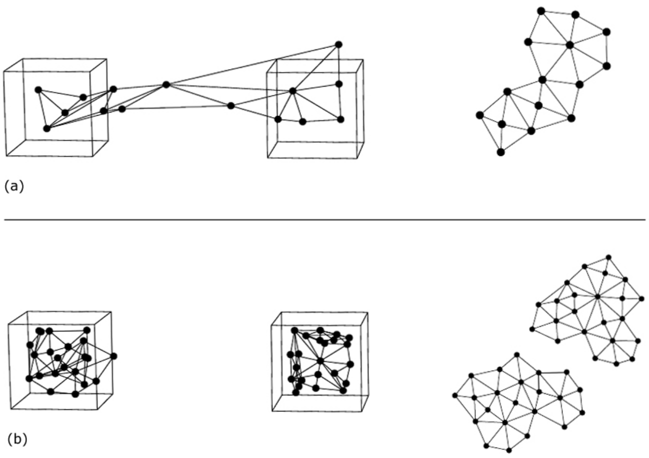

Figure 2.14: (a) initial stage of the process. Left hand side shows 3 dimensional input space, with the embedded 2d version on the right. (b) Later in the process. Notice that the two distinct parts are represented accurately.

In order to realise the representation in 2d, Fritzke converts each edge into a spring in the embedding space. These springs attract adjacent nodes, but all nodes repel each other too. The result is that the structures avoid getting tangled together and a visulisation in 2d is possible. A full description of the original method is found in Fritzke's paper [9]; however the method I will describe later is based on a simple linear spring model.

'Dynamic Cell Structures' is an extension to this model encompassing the ability to track dynamic inputs. This is however a complex algorithm to successfully employ involving the computation of hyper-Voronoi regions, and is beyond the scope of this study.

5.1. Summary

The neural network works as an autopoietic system in that it compares only what it sees inside and not from any outside criteria. This is because the examples discussed here are un-supervised, not directed towards any set goal and hence self-organising. The SOM example in Figure 2.8 shows this nicely with RGB inputs – it only makes a map of what *it* knows and so *our* knowledge of the full colour spectrum from human experience is irrelevant.

References

[1] Kohonen, T. (2000) Self-organizing Maps. Springer-Verlag Berlin, 3Rev Ed edition.

[2] Martinetz , T.M. & Schulten, K.J. (1991) A neural-gas network learns topologies. In Kohonen, T. et al, Artificial Neural Networks, pp 397-402. North-Holland, Amsterdam.

[3] Derix, C. & Coates, P. & Parvin, T. et al. (2005) Topological Approximations for Spatial Representations. 8th Generative Art Conference GA2005.

[4] Derix, C. & Thum, R. (2000) Self-Organizing Space. Generative Arts Conference, Milan.

[5] Derix, C. & Ireland, T. (2003) An Analysis of the Poly-dimensionality of Living. eCAADe Conference, Graz.

[6] Bauer, H.U. & Villmann, T.H. (1995) Growing a Hypercubical Output Space in a Self-Organizing Feature Map. International Computer Science Institute, Berkeley.

[7] Martinetz , T.M. & Schulten, K.J. (1991) A neural-gas network learns topologies. In Kohonen, T. et al, Artificial Neural Networks, pp 397-402. North-Holland, Amsterdam.

[8] Fritzke, B. (1994) A growing neural gas network learns topologies. NIPS, Denver.

[9] Fritzke, B. (1993) Kohonen Feature Maps and Growing Cell Structures – A Performance Comparison. In Advances in Neural Information Processing Systems, Giles, C.L. et al.