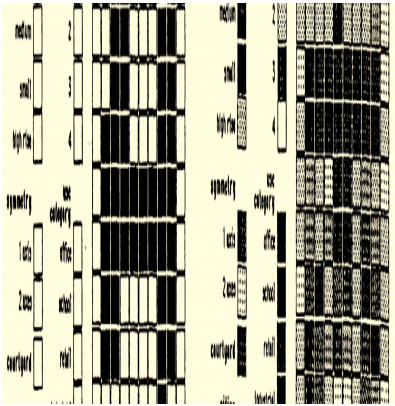




Automatic pattern completion

Description

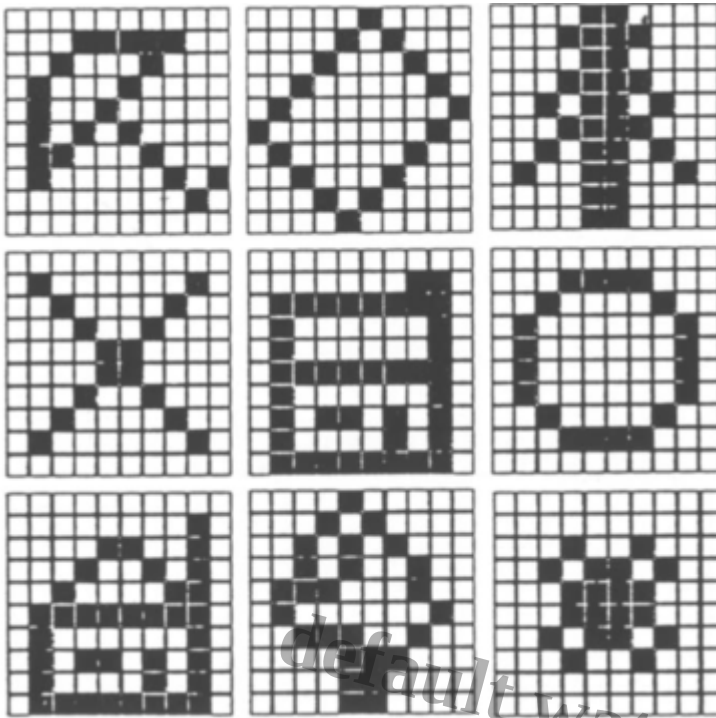
In 1990 [Arthur Postmus](#) and I published an article about “spatial applications” of artificial neural networks (ANNs). In a more recent article, Gabriele Mirra and Alberto Pugnale at the University of Melbourne developed up-to-date applications of AI in design. They generously cited our article (amongst articles by others). I concur with their assessment of the earlier work.

“Despite the potential of ANNs for the development of autonomous design assistants, these experiments did not develop into actual CAD systems. A major limitation was the low computing capabilities of the early ANNs, which progressively reduced interest in these techniques in and beyond the design fields” [114]

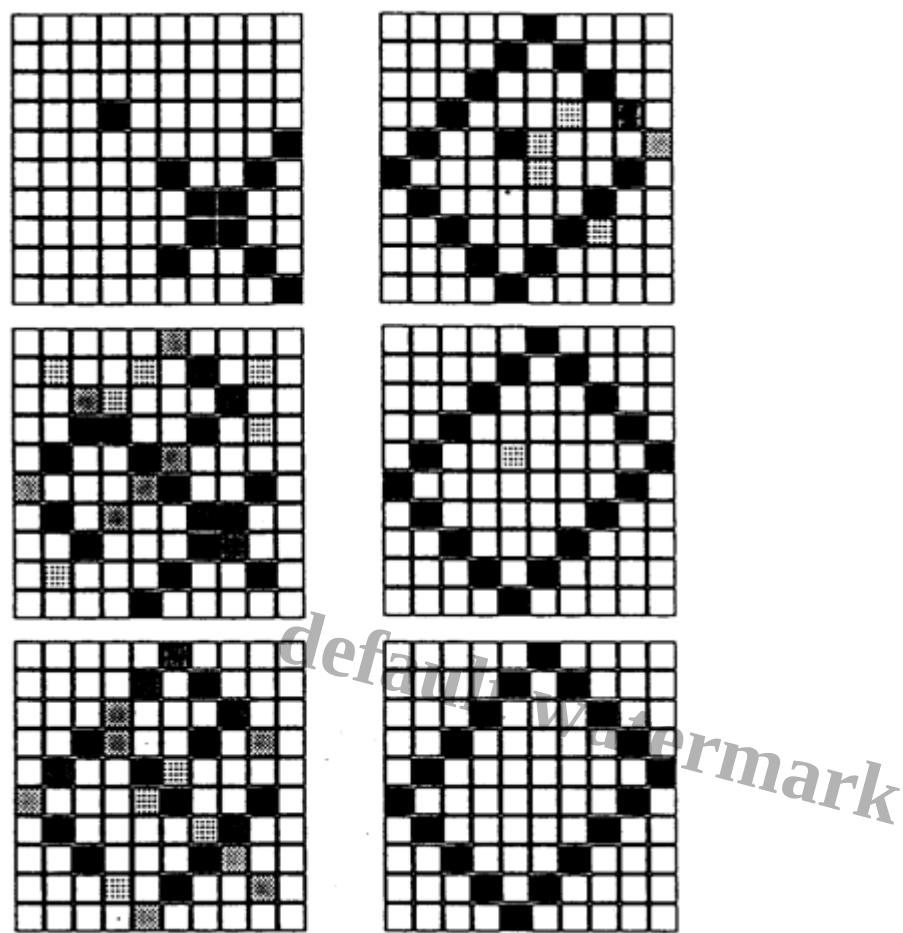
The technology has move on since 1990, and spectacular advances in imaging technology and natural language processing has revived my own interest, and that of other researchers, in such early experiments “an enthusiasm tempered I hope by well-informed skepticism about automation and machinic metaphors of human cognition.

Our model adopted the idea of an Auto-associative Neural Network (AANN), in which inputs are identical to outputs. That means that the NN is trained simply on patterns and the task of the network in run time is simply to complete those patterns. We avoided the technical challenge of implementing hidden layers in ANNs. How far could we go with a simple AANN?

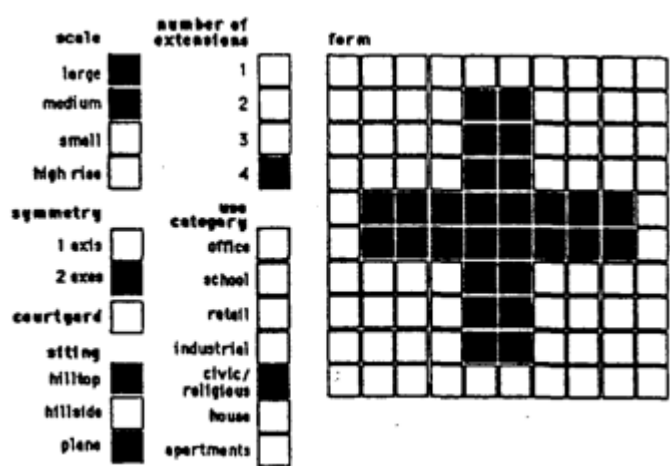
Initially, we set our training universe as a 10Ã210 grid of pixels containing simple shapes, a bit like emojis. Our AANN was trained on just 9 of these.



The idea was that after training, given a partial pattern, the AANN would be able to complete any partial pattern presented to it, albeit with some noise. Feeding that noisy reconstructed pattern back into the AANN algorithm would eventually home in on a precise match. The initial partial pattern could even be noisy and ambiguous as in this figure, top left. Successive iterations home in on the closest match, bottom right.



It occurred to us that some of the cells in such grid arrangements could be hived off to serve as labels, features or attributes. So the training data could include floor plan outlines to provide a simple typology of building shapes. We provided 9 training patterns, each made up of 121 grid cells: 21 dedicated to labels, and 100 dedicated to the spatial layout with which each collection of labels was associated. Here is one of the patterns.



I appreciate that the labels are difficult to read in this reproduction from our original article. That’s just as well considering the coarse granularity of the information.

So the AANN would serve as a store of plans and their labels. We were really using a neural network to simulate what you could achieve with a regular database, but with more approximate matching from partial information.

After training, you could present the AANN with some labels (as requirements) and after several iterations it would produce the closest matching plan that met those requirements (along with some other attributes).

A more sophisticated AANN with a much larger training set might also generate new plans with as yet unseen lists of attributes, but that was beyond the scope of this exercise. Here is the progression from an initial set of labels (requirements) to the final state over several iterations.

default watermark

scale	number of extensions	form
large	1	
medium	2	
small	3	
high rise	4	
symmetry	use category	
1 axis	office	
2 axes	school	
courtyard	retail	
sitting	industrial	
hilltop	civic/religious	
hillside	house	
plane	apartments	

scale	number of extensions	form
large	1	
medium	2	
small	3	
high rise	4	
symmetry	use category	
1 axis	office	
2 axes	school	
courtyard	retail	
sitting	industrial	
hilltop	civic/religious	
hillside	house	
plane	apartments	

scale	number of extensions	form
large	1	
medium	2	
small	3	
high rise	4	
symmetry	use category	
1 axis	office	
2 axes	school	
courtyard	retail	
sitting	industrial	
hilltop	civic/religious	
hillside	house	
plane	apartments	

scale	number of extensions	form
large	1	
medium	2	
small	3	
high rise	4	
symmetry	use category	
1 axis	office	
2 axes	school	
courtyard	retail	
sitting	industrial	
hilltop	civic/religious	
hillside	house	
plane	apartments	

Towards the end of that article we describe attempts to apply the AANN method to sequences of actions that produce plans rather than the plans themselves.

The shapes in the diagrams shown here are fixed in the grid. The AANN contains no mechanism for recognising or generating a T-shaped plan that is inverted, at a different orientation, scale, or a slight variation on the gridded shape shown in the diagram above.

With a more sophisticated NN model that deploys hidden layers there is a chance that the system will detect and generalize on features rather than patterns of binary values with fixed locations on a grid.

I was reminded of the value of hidden layers as a means of identifying and generalising on features across many examples when I saw the use of the Modified National Institute of Standards and Technology (NMIST) training set for image recognition for handwriting recognition. NMIST provides an extensive set of handwritten digits 0-9 and is used to baseline test neural network models. An extremely helpful set of [videos](#) by Grant Sanderson shows the use of a gridded representation of the space of each handwritten numerical digit similar to the grids above, but with grey scale cell values for later discussion.

Bibliography

- Coyne, Richard, and Arthur Postmus. "Spatial applications of neural networks in computer-aided design." *Artificial Intelligence in Engineering* 5, no. 1 (1990): 9-22.
- Manfaat, D., A.H.B. Duffy, and B.S. Lee. "Review of pattern matching approaches." *The Knowledge Engineering Review* 11, no. 2 (1996): 161-189.
- Mirra, Gabriele, and Alberto Pugnale. "Expertise, playfulness and analogical reasoning: three strategies to train Artificial Intelligence for design applications." *Architecture, Structures and Construction* 2 (2022): 111-127.
- Sanderson, Grant. "Neural Networks: The basics of neural networks, and the math behind how they learn." *3Blue1Brown*, 2023. Accessed 12 February 2023.
<https://www.3blue1brown.com/topics/neural-networks>

Category

1. Artificial Intelligence

Tags

1. Auto-associative Neural Network
2. design
3. neural network
4. patterns

Date Created

March 11, 2023

Author

rcoyne99