Explanation of the "Virtual Input" Phenomenon

Editor:

We write this letter to comment on the "virtual input" phenomenon reported in Thaler (1995). The author attributed the phenomenon to the network's ability to perform pattern classification and completion, and reported that pruning probability directly affects the number of virtual inputs observed. Our independent study of Thaler's results, however, reveals a simpler explanation of the "virtual input" phenomenon.

1 "Virtual Input" in Pattern Associator

Thaler (1995) trained a fully connected 3-5-9 neural network to map 3-bit inputs to 3×3 output patterns (Fig. 1). After training, the network was run with a fixed input of $\{0,0,0\}$ while its connection weights were pruned. The author found that, despite the fixed input, the network generated various training output patterns as more and more connections were pruned. In particular, patterns 0, 2, 4, 5, 6, 7 in Fig. 1 were generated at various stages of the pruning process, and a novel pattern (Fig. 1, pattern 8) was also produced.

A close examination of the trained network's connection weights reported in Thaler (1995) shows the following:

- The output units' biases are approximately equal to 1.
- The weights of the connections to the output units are approximately either -2, 0, or $+2$.
- The connection weights to the output units that form the corners of the 3×3 grid are approximately the same. Similarly, the weights to the side units are roughly identical.

In essence, the units in the hidden and the output layers form 3 sub-networks as shown in Fig. 2.

Given the above network architecture, it is obvious that the activation of the corner units are synchronized; likewise for the side units. As a result, the trained network can only produce the 8 training output

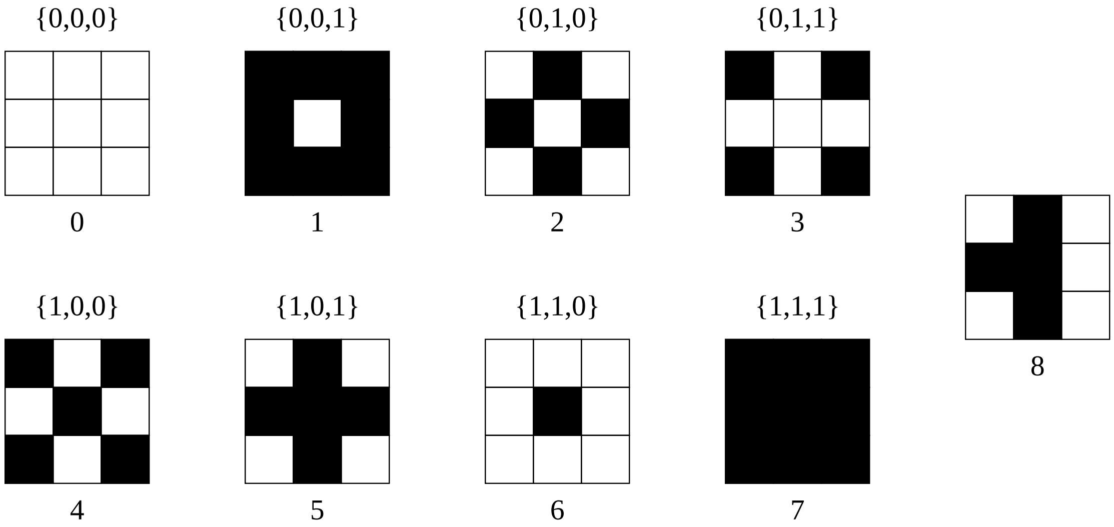

Figure 1: The eight training samples (0–7) consists of 3-bit inputs and 3×3 output patterns. Pattern 8 is a novel pattern generated by the trained network when its connections are pruned.

patterns depending on which hidden units are activated. Even if the hidden layer contains real-numbered instead of binary units, the network will still generate the trained output patterns. Furthermore, even if the connection weights are not exactly equal to the nominal values of -2, 0, and $+2$, the network will still generate the trained output patterns as long as the weights are close to the nominal values. The closer are the weights to the nominal values, the higher is the frequency of generating trained output patterns.

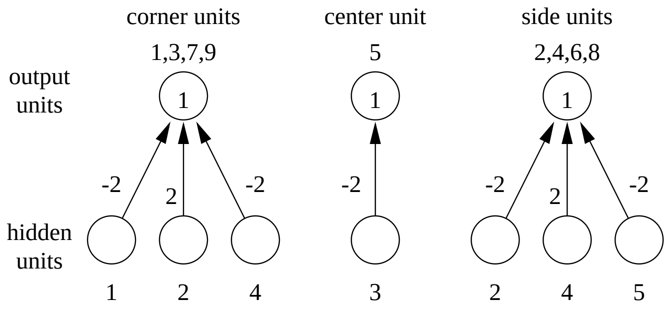

Figure 2: The hidden and the output layers form 3 sub-networks. The corner units have identical weight structures. Similarly, the side units also have identical weight structures. For clarity, hidden units 2 and 4 are drawn twice.

2 The Mystery Explained

Having understood how the network operates, generation of trained output patterns becomes a trivial matter: just activate any hidden units. The network will produce trained outputs regardless of whether the hidden units are activated randomly, stochastically, or as a result of network pruning. That is, network pruning is redundant in the generation of trained output patterns.

The generation of novel output pattern, on the other hand, is indeed due to the removal of some hidden-to-output connections. However, such output generation amounts to nothing more than the hidden units making errors in activating the output units. In other words, the generation of the so-called novel outputs is actually imperfect recalls of learned patterns rather than pattern classification and completion.

Wee Kheng Leow and Rudy Setiono
Department of Information Systems and Computer Science
National University of Singapore
Lower Kent Ridge Road, Singapore 119260
Singapore

References

Thaler, S. L. (1995). “virtual input” phenomena within the death of a simple pattern associator. *Neural Networks*, 8(1):55–65.