Standard feedforward neural nets cannot support cognitive superposition

Arno Vanegdom[1], Nikolay Nikolaev[1], Max Garagnani[12]

1. Department of Computing, Goldsmiths – University of London, London, UK
2. Brain Language Lab, Department of Philosophy and Humanities, Freie Universität Berlin, Berlin, Germany

Background

Superposition is defined here as the cognitive ability to simultaneously reactivate and hold in mind several conceptual representations that have been learned independently / separately.

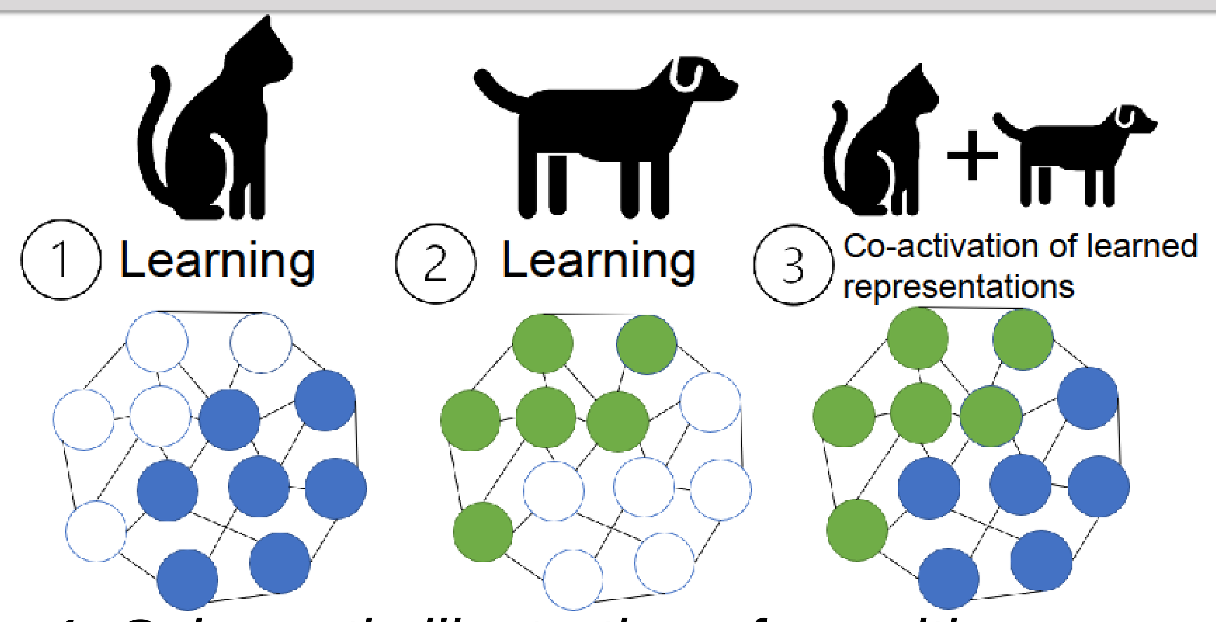


Figure 1: Schematic illustration of cognitive superposition

- Superposition is arguably a building block of higher-level cognitive functions, crucial to human intelligence
- Only a few studies have addressed the implementation of superposition in standard artificial neural networks [1,2]; these provided contrasting results, and attempted to achieve superposition by co-activating items already ***during*** training
- Taking a more ecologically accurate approach, here we assessed the ability of standard feedforward artificial neural networks (FFNNs) **[3]** to implement superposition of two internal representations which had been learned independently, i.e., that were ***never "experienced" together during training***.

Objectives

1. Assess the ability of FFNNs to implement cognitive superposition
2. Understand the underlying functional mechanisms and representational constraints that determine the above ability

Methods

- ***Experiments 1–3 (Objective 1)***

Using backpropagation [3], we trained three one hidden-layer FFNNs with 20 nodes in Input, hidden and output layer to map 2 sets of 5 distinct binary patterns in input to their set-corresponding single binary patterns in output for Exp 1-2, and 10 input patterns mapped to identical 10 output patterns for experiment 3, Across the 3 experiments, we varied the density of the I/O patterns, with sparse (1/20) Input and dense Output (5/20) in Exp. 1, sparse Input (2/20) and very dense Output (10/20) in Exp. 2, and sparse (1/20) input *and* output in Exp. 3.

	Set 1	Set 2
Input 1	1 0 0 0 0 … 0 0 0 0 0	0 0 0 0 0 … 0 0 0 0 1
Input 2	0 1 0 0 0 … 0 0 0 0 0	0 0 0 0 0 … 0 0 0 1 0
Input 3	0 0 1 0 0 … 0 0 0 0 0	0 0 0 0 0 … 0 0 1 0 0
Input 4	0 0 0 1 0 … 0 0 0 0 0	0 0 0 0 0 … 0 1 0 0 0
Input 5	0 0 0 0 1 … 0 0 0 0 0	0 0 0 0 0 … 1 0 0 0 0
Output	1 1 1 1 1 … 0 0 0 0 0	0 0 0 0 0 … 1 1 1 1 1

Figure 2. Training patterns for Experiment 1. Dots represent a serie of 0s

To assess a network's cognitive superposition ability, each network was given in input the superposition (inclusive OR) of two of the patterns it had been trained with. The resulting output was then compared against the correct output (the superposition of the two output patterns – see Fig. 2). (Note: real-value units' outputs in [0,1] were discretized into binary values using 0.5 as threshold).

- ***Experiment 4 (Objective 2)***

We trained 6 FFNNs, decreasing the network's size from 20 nodes per layer down to 2 nodes per layer and analyzed the weight configurations that emerged in the networks as a result of training.

Category 1

Input	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Output	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Category 2

Input	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Output	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Superposition

Input	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Expected	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

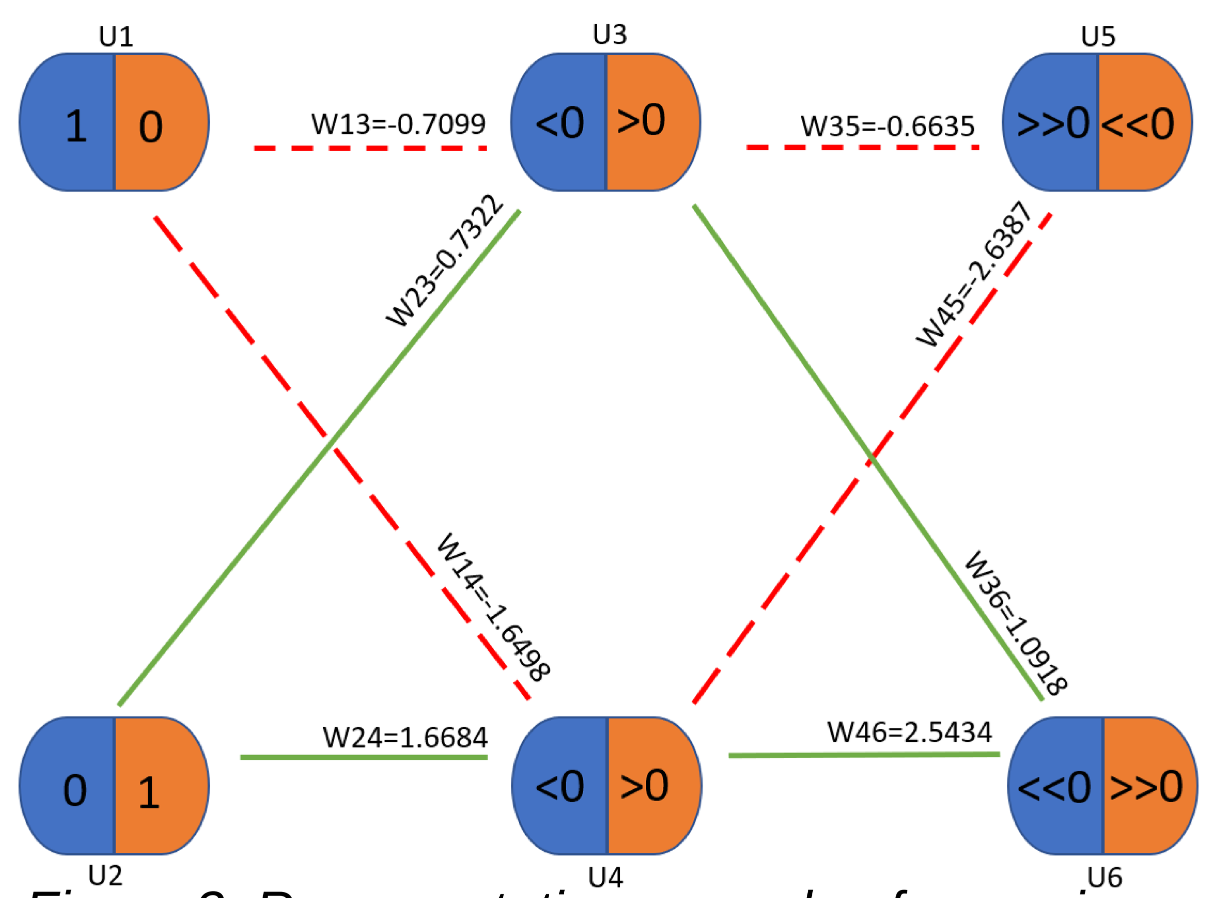
Figure 2. Schematic of the testing for experiment 1

Results: Experiments 1–3

Across the **75** trials, the NN was systematically unable to successfully produce the complete superposed pattern in the output layer. The average accuracy across all trials was **36%** with a standard deviation of **0.31**. This result suggests that standard FFNNs are generally unable to implement cognitive superposition as defined here.

Results: Experiment 4

The analysis of the smallest network (figure 3) made us identify that the underlying cause of the NN's inability to implement superposition was due to the inherently fully distributed representation determined by the backpropagation algorithm and its "greediness" : backpropagation gives a role to all the weights of the network in creating the representation.



The blue color represent the nodes values for the 1st training pattern, orange for the 2nd . Dashed red lines represent negative weights value and solid green lines positive weights values. WXY represents the weights linking the node X to the node Y.

Figure 3. Representative example of emerging weight configuration.

Summary

- Standard FFNNs trained with backpropagation appear to be generally very limited in their ability to support superposition of two previously learned internal conceptual representations.
- The mechanisms and representational constraints characterizing FFNNs that prevent these networks' internal representations to be superposed are the interaction of the all-to-all connectivity topology with the backpropagation algorithm leading to internal representations distributed across the entire set of hidden nodes, which render the co-activation of several representations impossible.

1. Bowers, J. S., Vankov, I. I., Damian, M. F., & Davis, C. J. (2014). Neural networks learn highly selective representations in order to overcome the superposition catastrophe. ***Psychological Reviews,*** 121(2), 248-261
2. Martin, N. D. (2021). Selectivity in neural networks (Doctoral dissertation, University of Bristol).
3. Haykin, S.O. (2009). Neural networks and learning machines. Pearson Prentice Hall, New Jersey. 3rd ed.