Unit 1: Simple Neural Networks

6. Multi-layer networks

9/17/2020

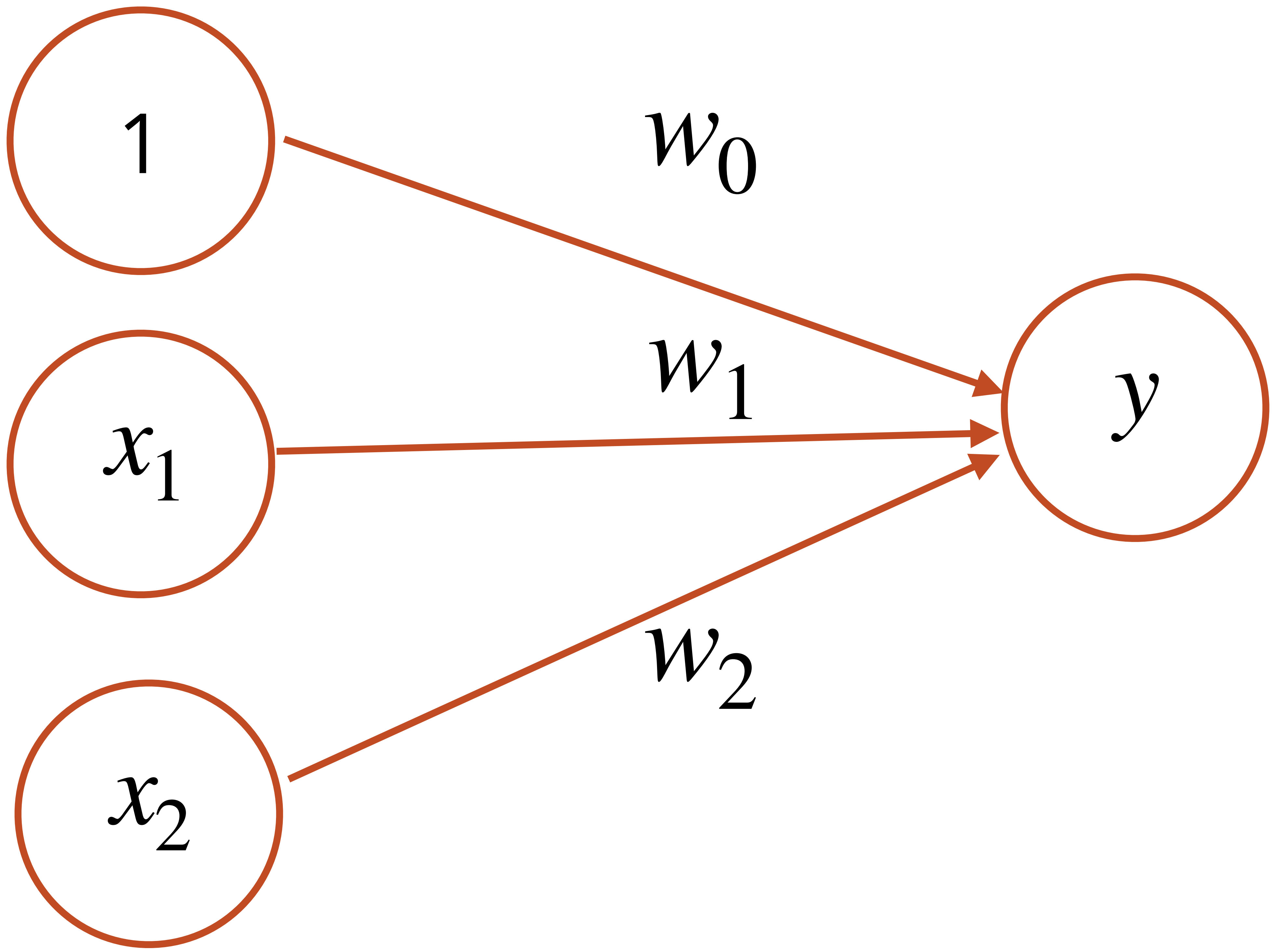
Multi-layer networks

1. Dynamics of neural networks can capture features of human information processing

2. Backpropagation is a general algorithm for learning in multi-layer networks

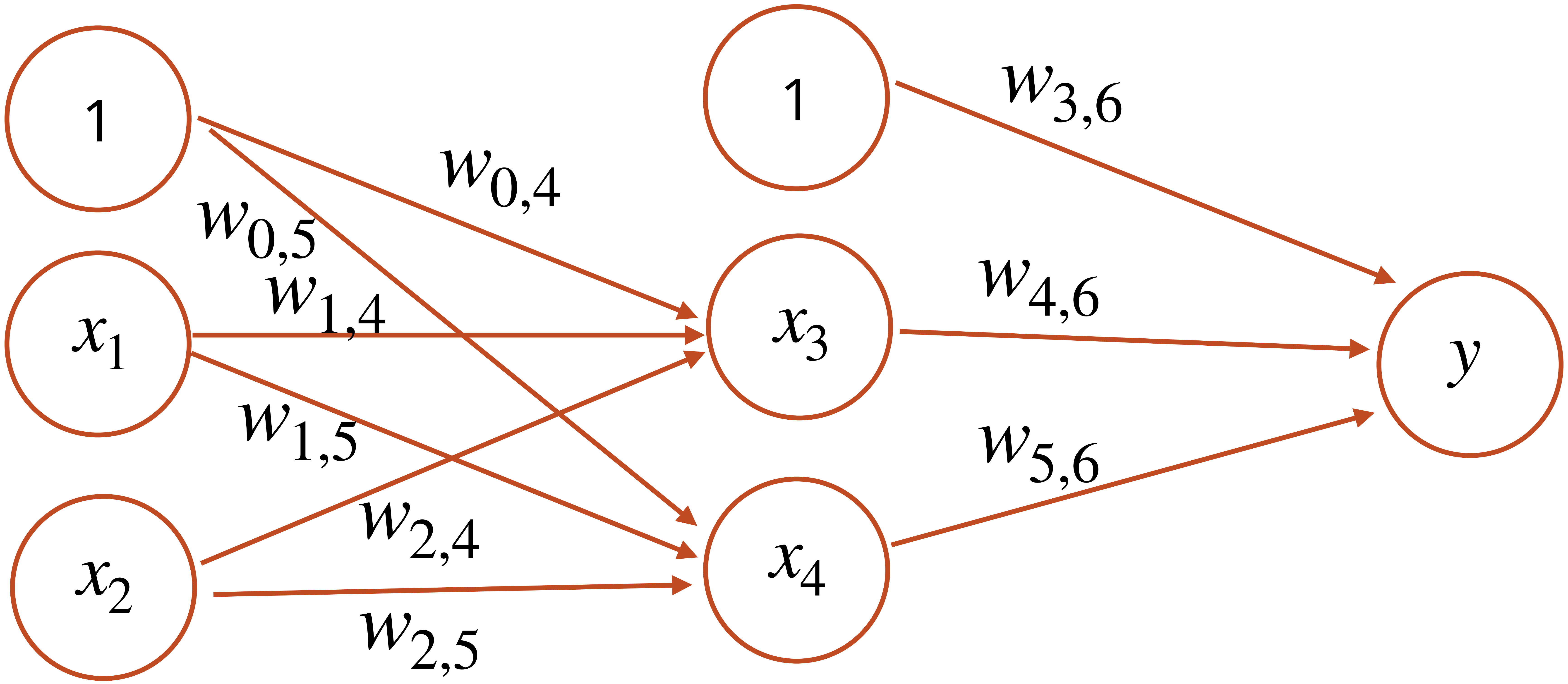
3. Neural networks can give rise to “emergent” learning phenomena

Single layer perceptrons are linear classifiers



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Multi-layer perceptrons are non-linear classifiers



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Intuition: Hidden layer nodes can encode arbitrary interactions between input layers

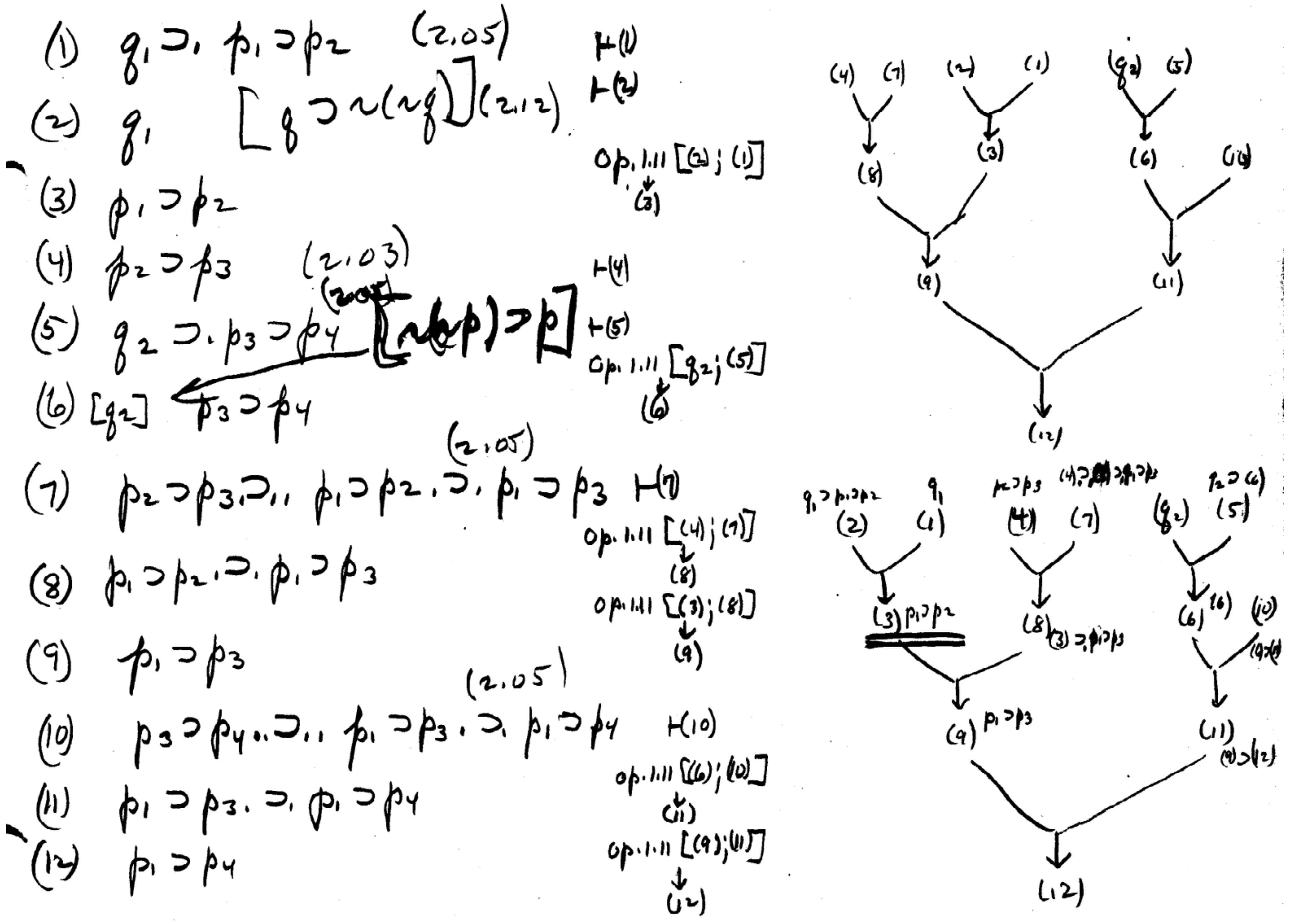
Who cares?

Why not just use regression?

Are there any inherent reasons to be interested in networks?

"A physical symbol system has the necessary and sufficient means for general intelligent action”

Newell and Simon (1976)



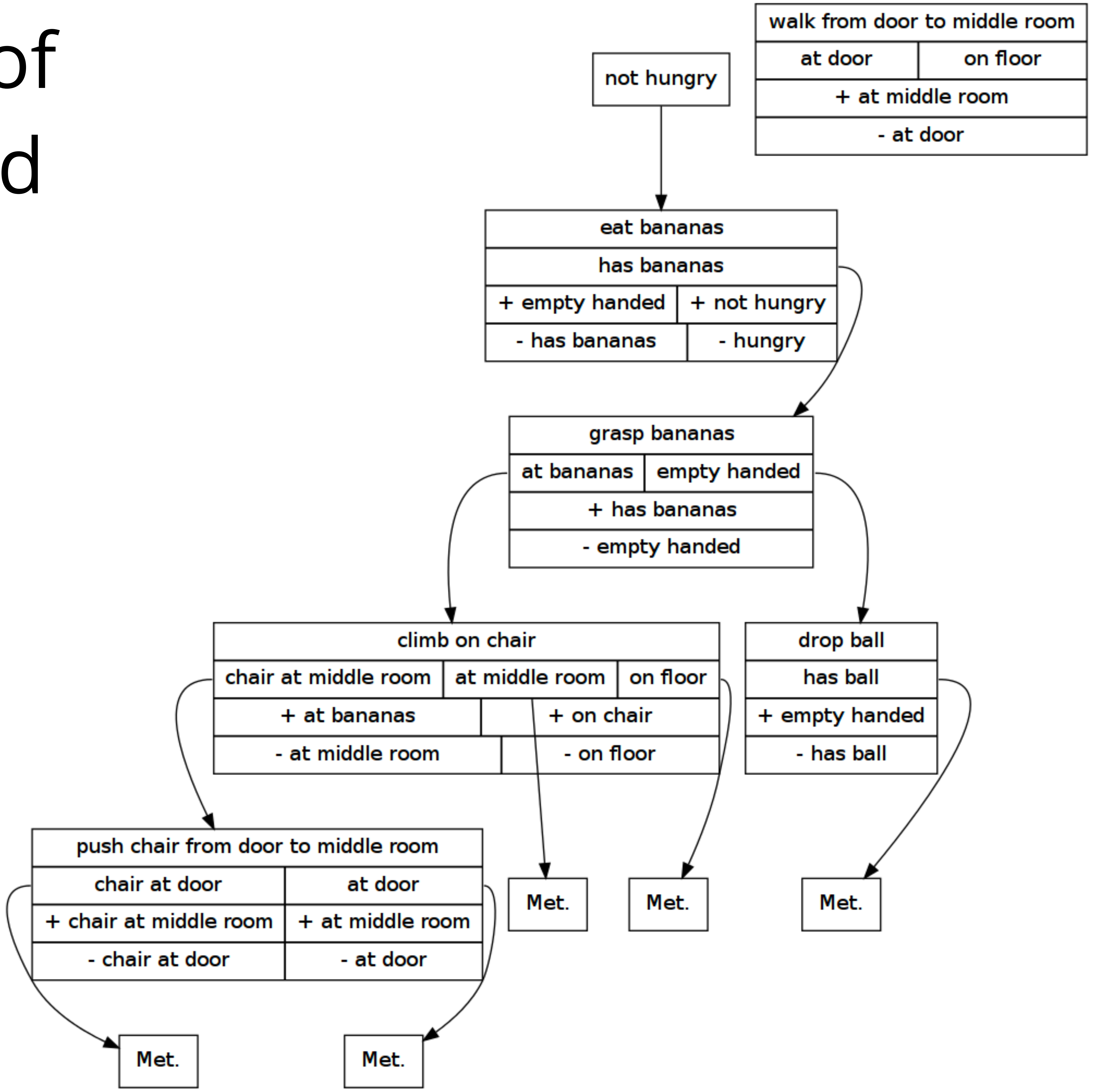
The physical symbol system hypothesis

1. The brain is a computer that manipulates symbols
2. You can distinguish between the hardware (neurons) and software (knowledge)
3. In principle, this intelligence software can be run on many different kinds of hardware, including potentially desktop computers (functionalism)
4. Our goal as cognitive scientists is to understand the software. Who cares about the hardware?

The General Problem Solver (1959)

If you can define a search space of transformations, a start state, and an ends state, the algorithm can determine how to find the goal

Problem: How do you get this search space?



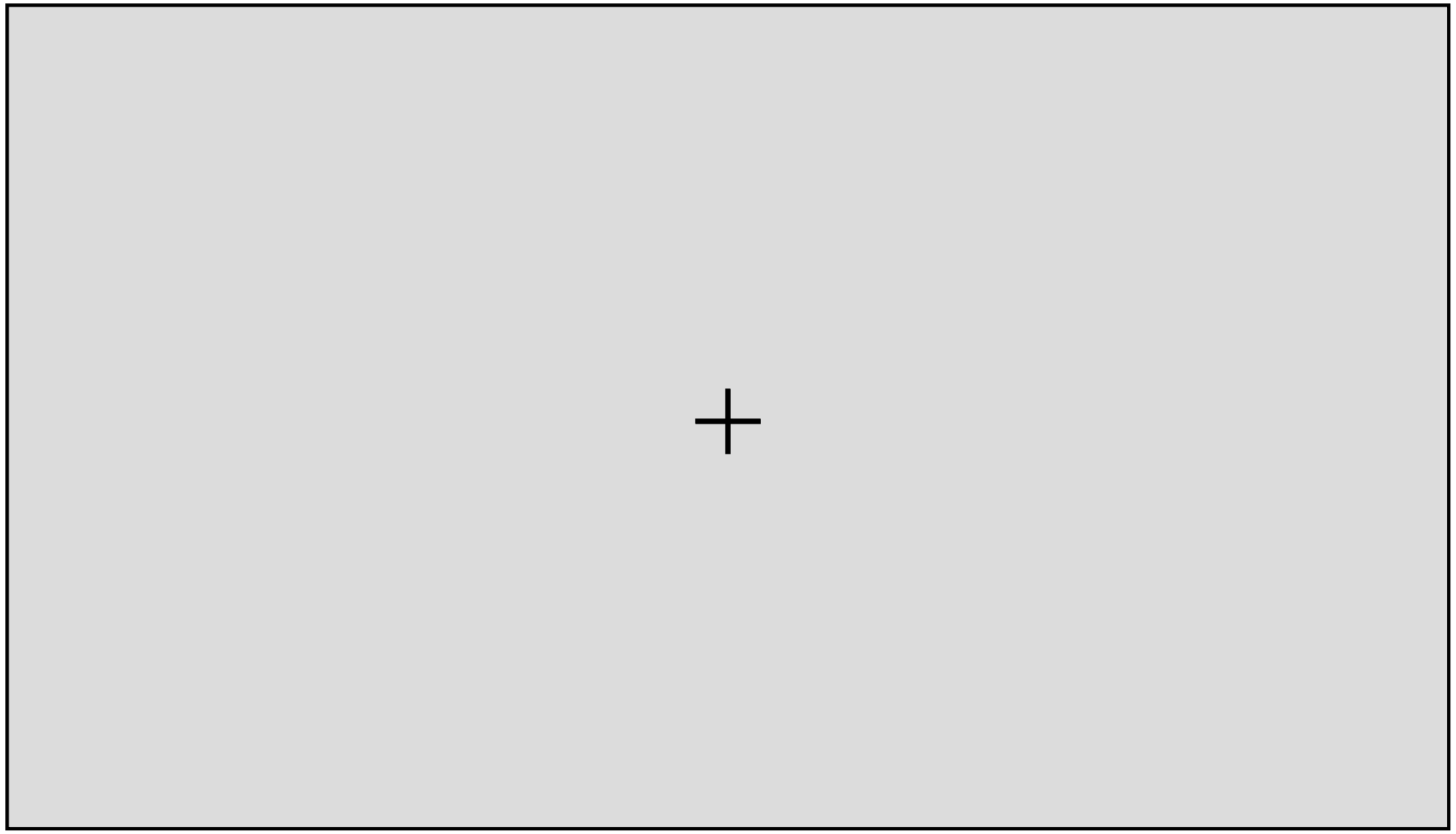
Strengths of connectionism

1. Each unit of the network is a simple computer, but the network as a whole can give rise to complex phenomena.

2. The framework is general—you don’t need a separate model for every domain (sort of).

3. Blurs the hardware/software distinction

The word superiority effect (Reicher, 1969)



The word superiority effect (Reicher, 1969)

COURSE

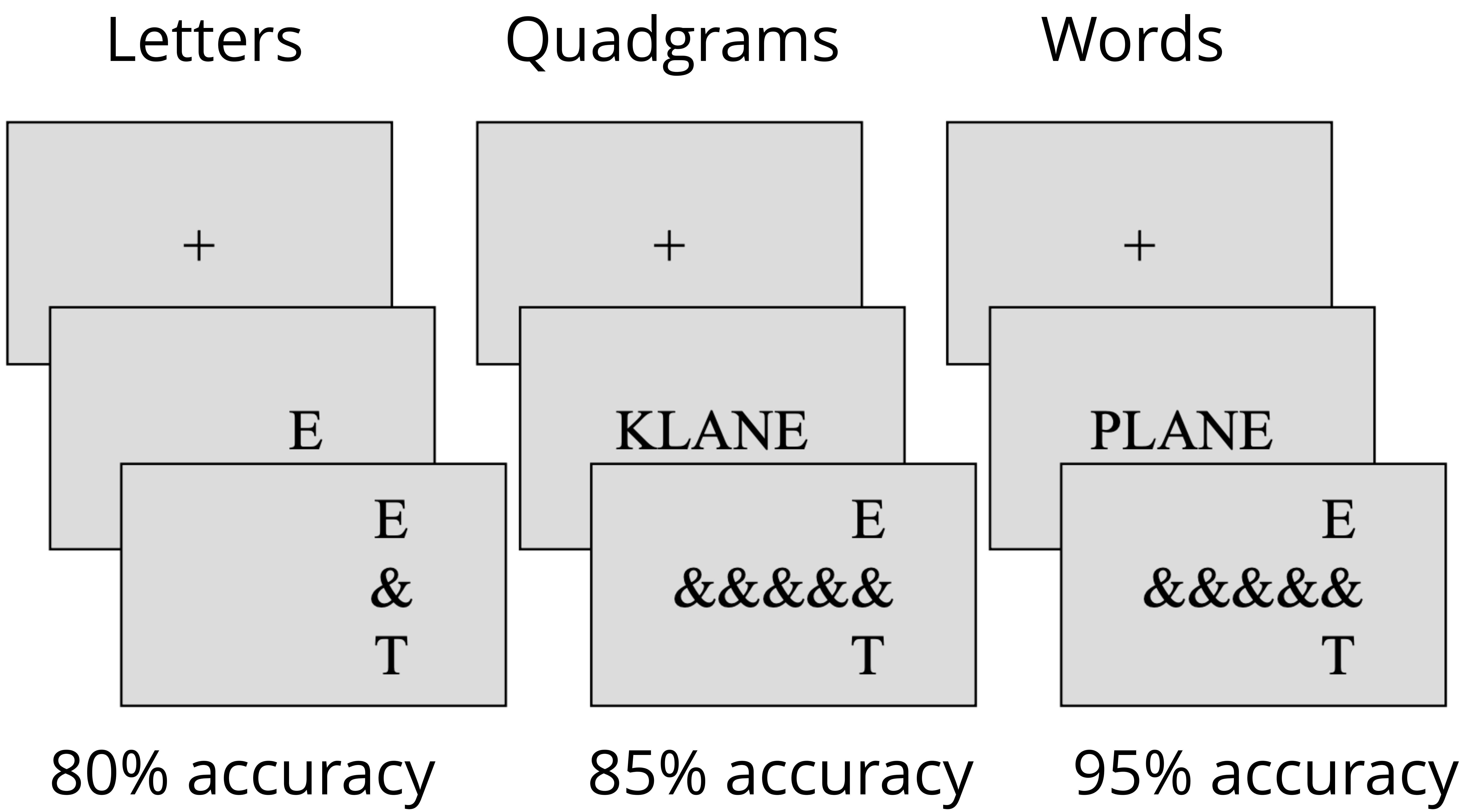
The word superiority effect (Reicher, 1969)

U

&&&&&

A

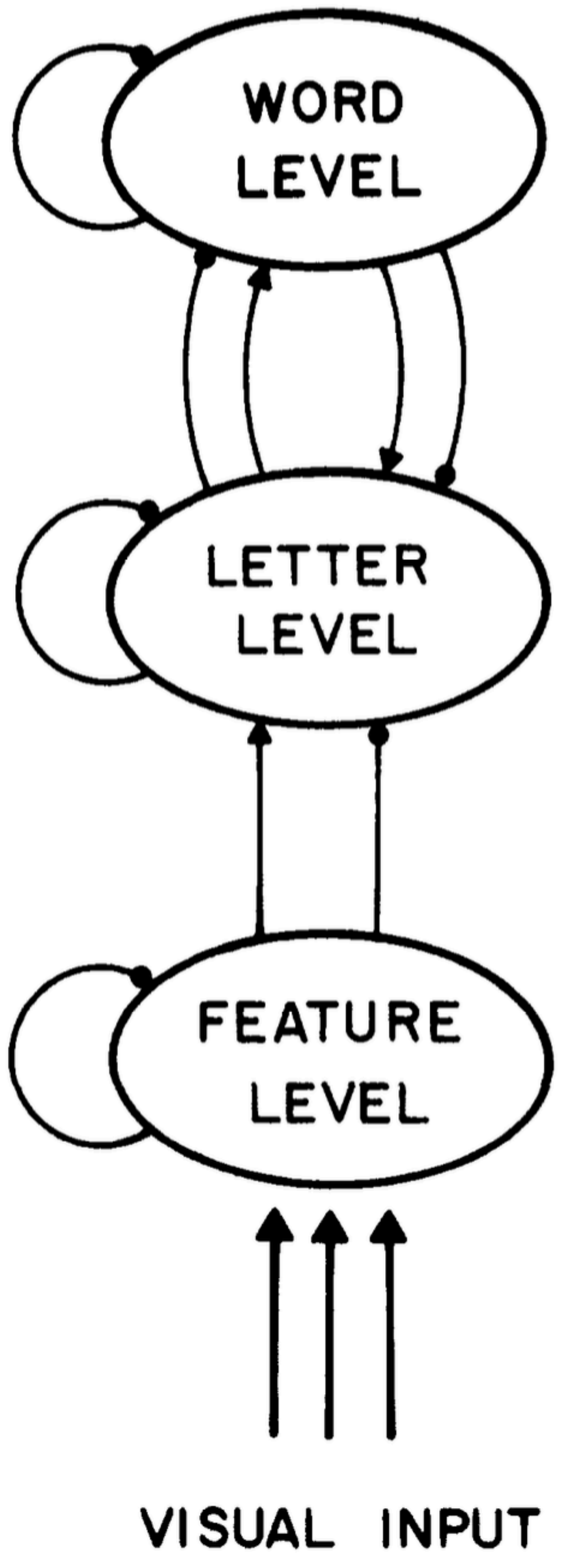
The word superiority effect (Reicher, 1969)



Interactive Activation Model (McClelland & Rumelhart, 1981)

Naive model of word processing:
You first perceive visual features,
These features are used to recognize letters,
You combine letters to recognize words

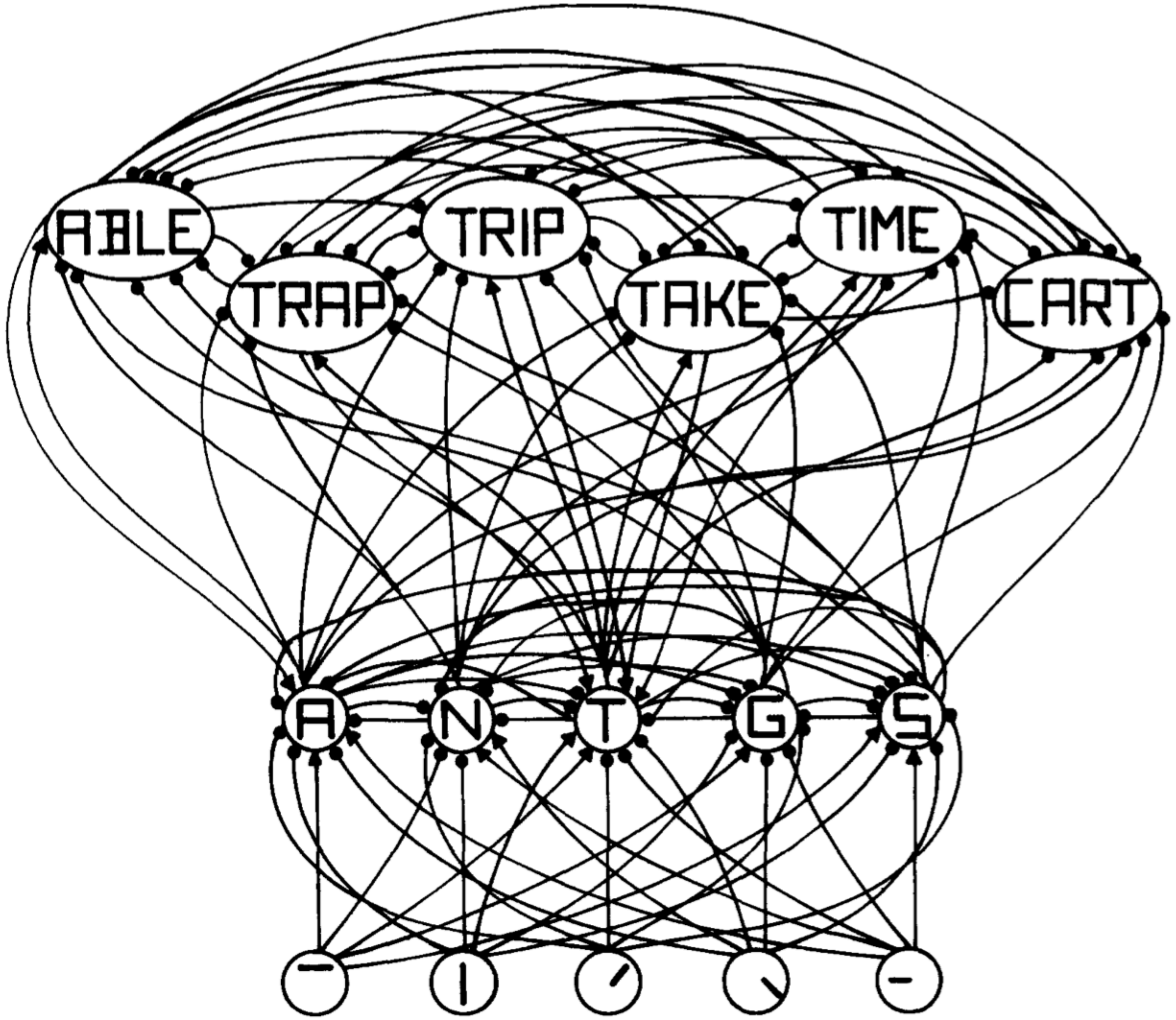
Key claim of the IAM:
All of these processing steps happen in parallel, and interact with each-other



A sketch of the Interactive Activation Model

Inhibitory connections within-levels
If the first letter is T, it isn't A

Inhibitory and excitatory connections between-levels
If the first is T, the word could TIME, but not WORK
If there is growing evidence that the word is TIME, the first letter is probably T

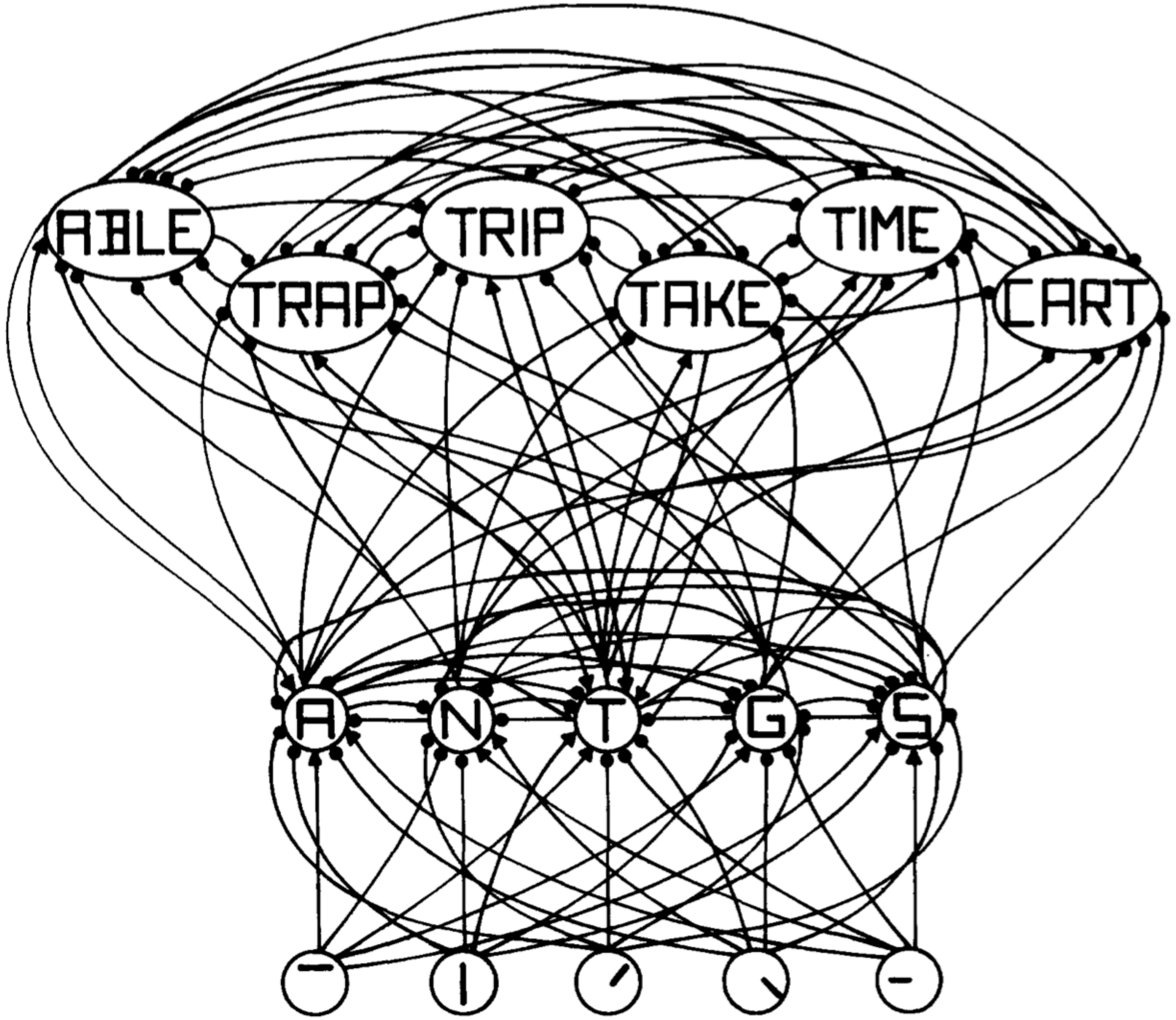


A sketch of the Interactive Activation Model

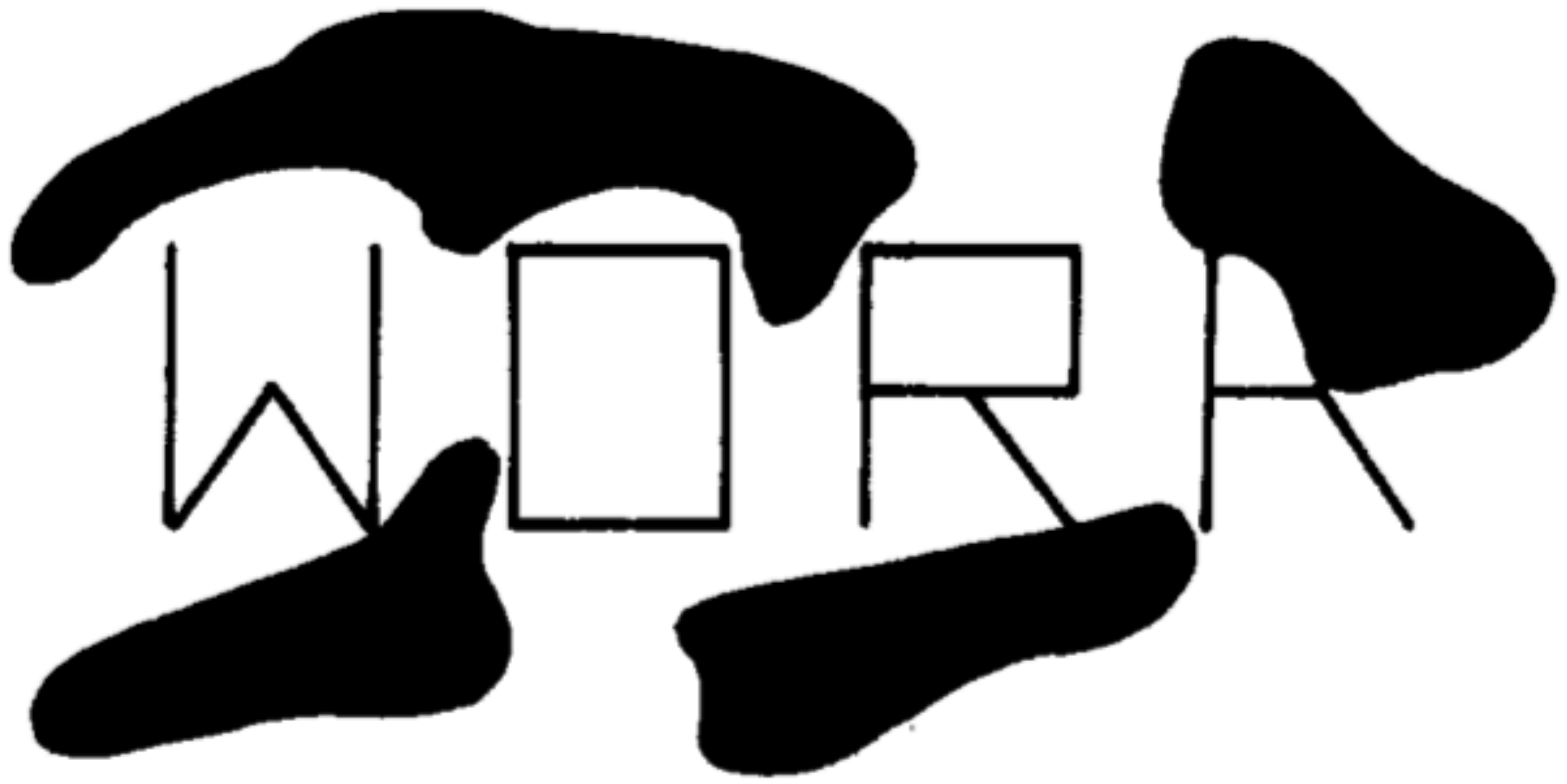
Word frequency affects expectations

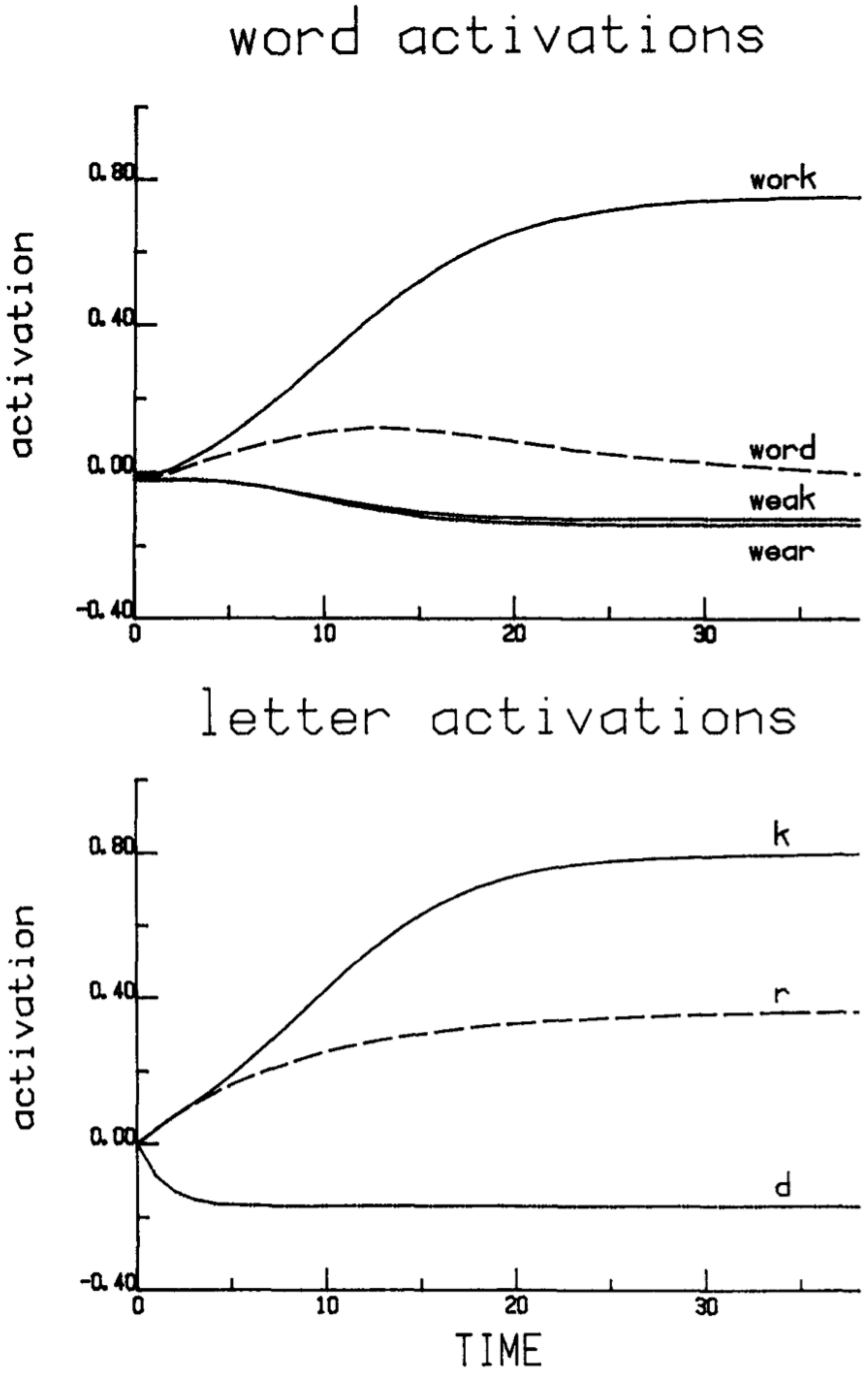
If you see T- - -, you are more likely to be reading TIME than TARP

Even if you see T- - -, you're very unlikely to be reading TPAR

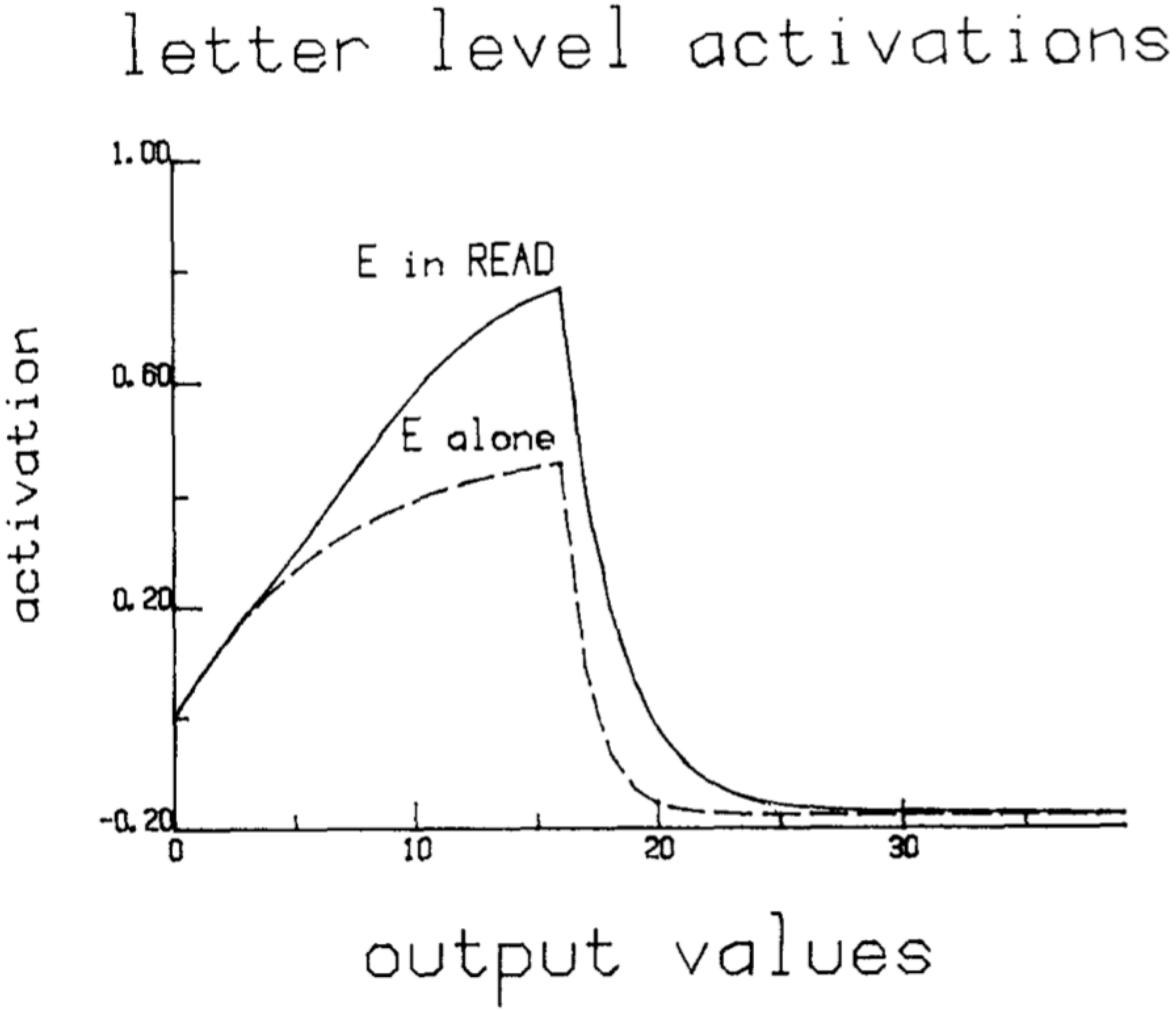


A sketch of the Interactive Activation Model





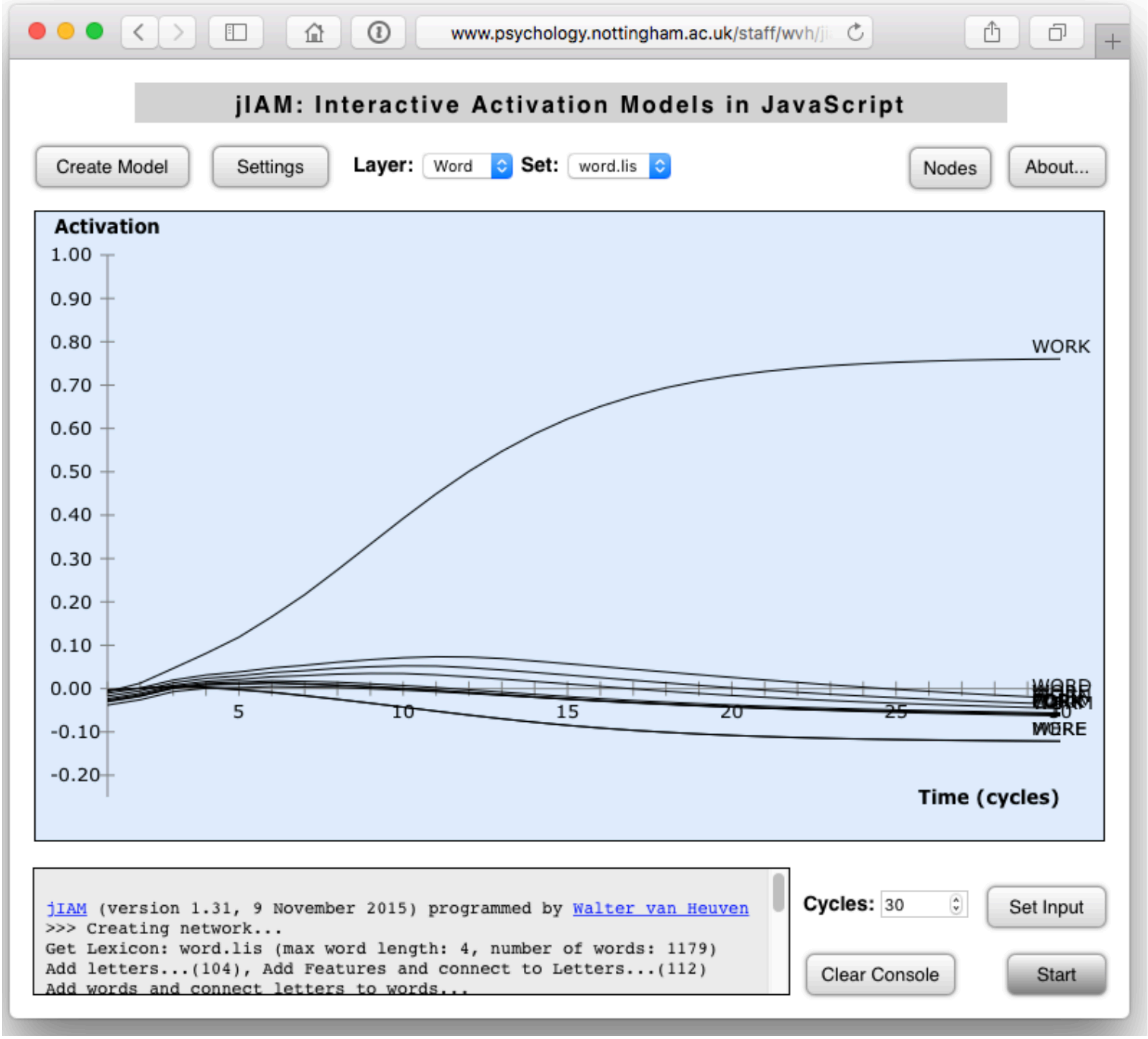
The IAM predicts the Word Superiority Effect



Do you get the Word Superiority Effect for non-words?

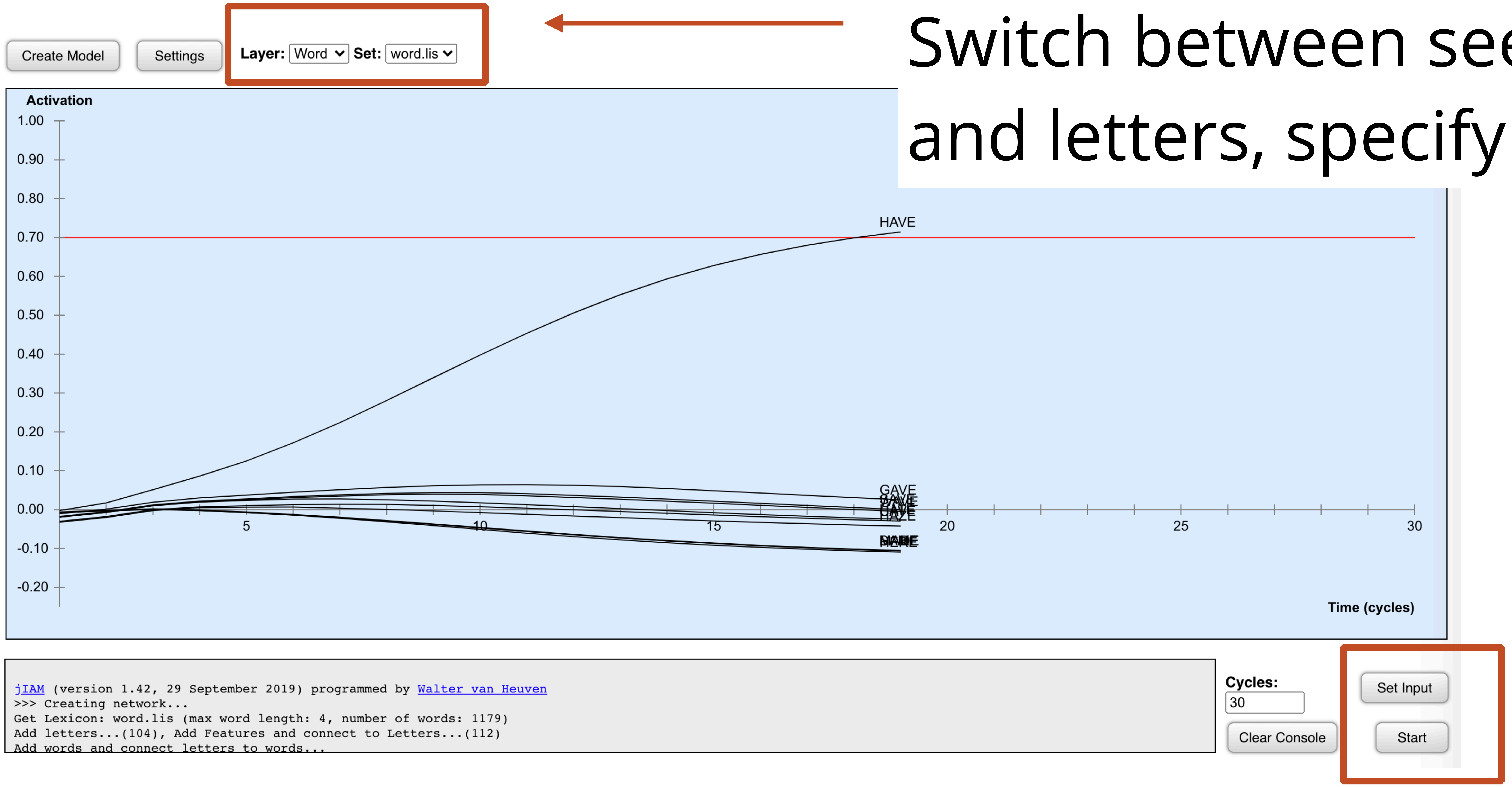
Try out some words and non-words in the app.

Compare, e.g.
- - V -
HAVE
MAVE
AMVE
EMVA



https://waltervanheuven.net/jiam/index.html

Using the Interactive Activation Model app

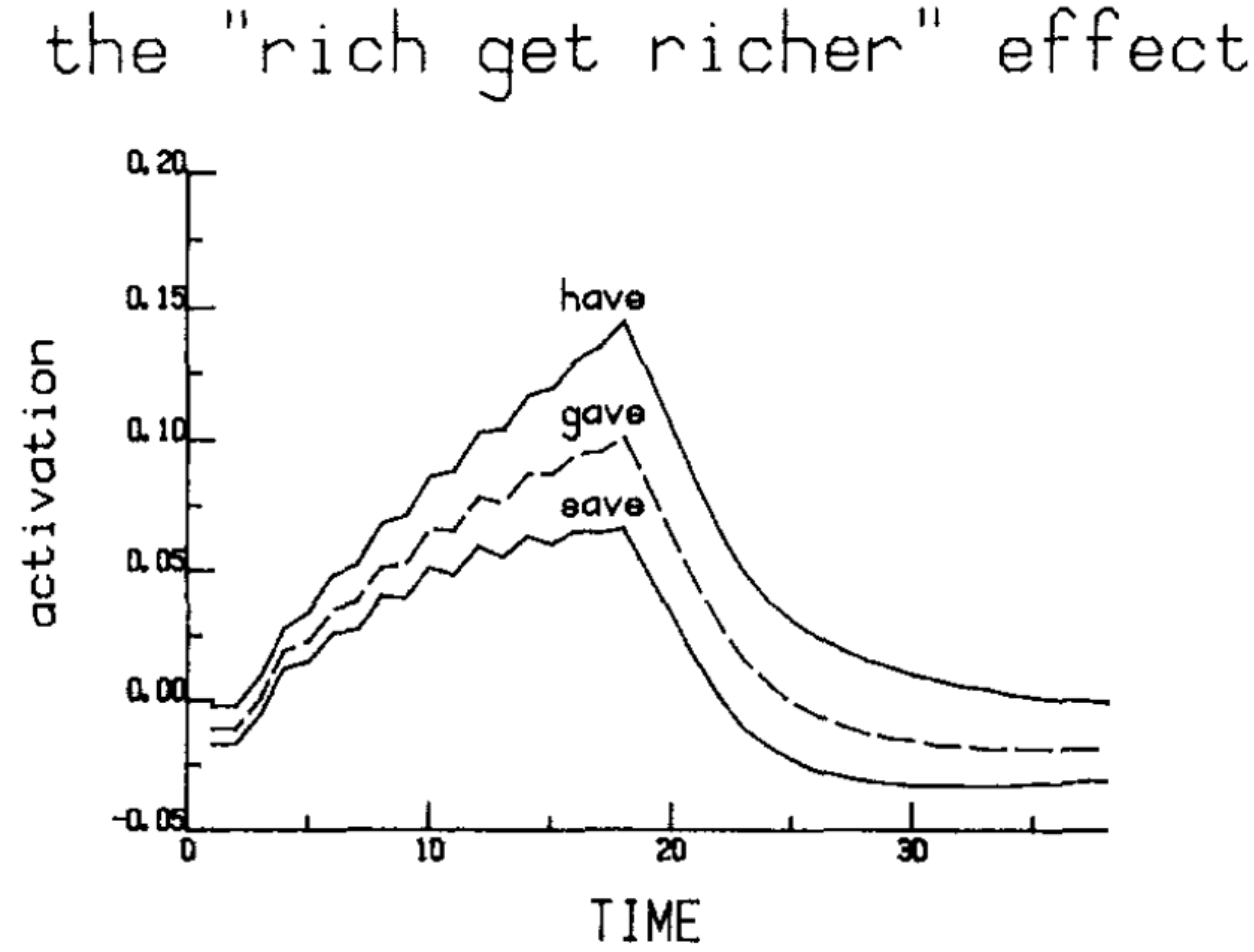


Switch between seeing words and letters, specify *which letter*

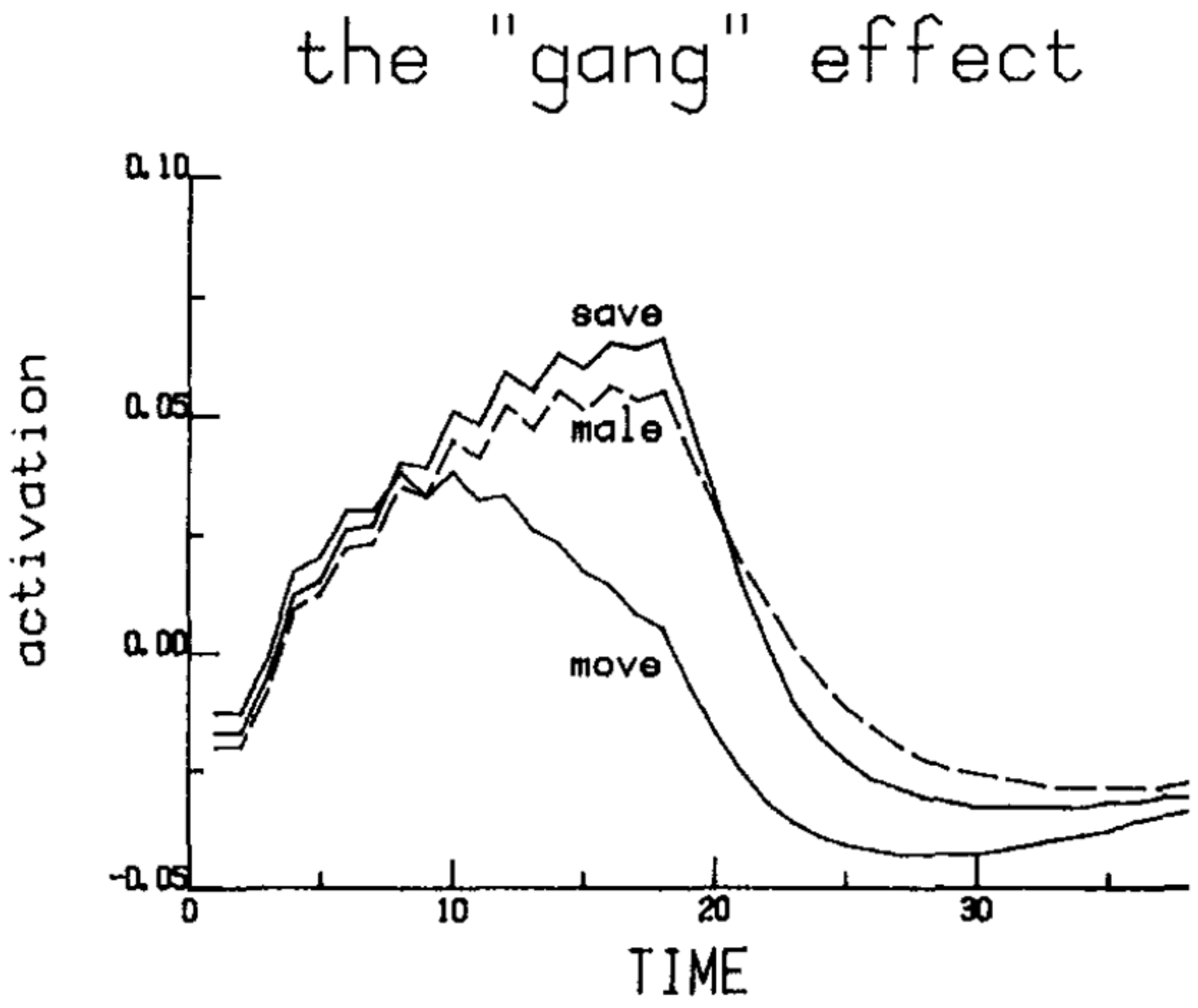
Change input and run

https://waltervanheuven.net/jiam/index.html

Two other interesting effects



Frequency differences get magnified over time



Similar words support each-other

Strengths and Weaknesses of the Interactive Activation Model

Strengths

1. A complex and surprising effect arises of individual connections with no goal
2. The dynamics of this network give rise to phenomena about timing of information processing that are testable

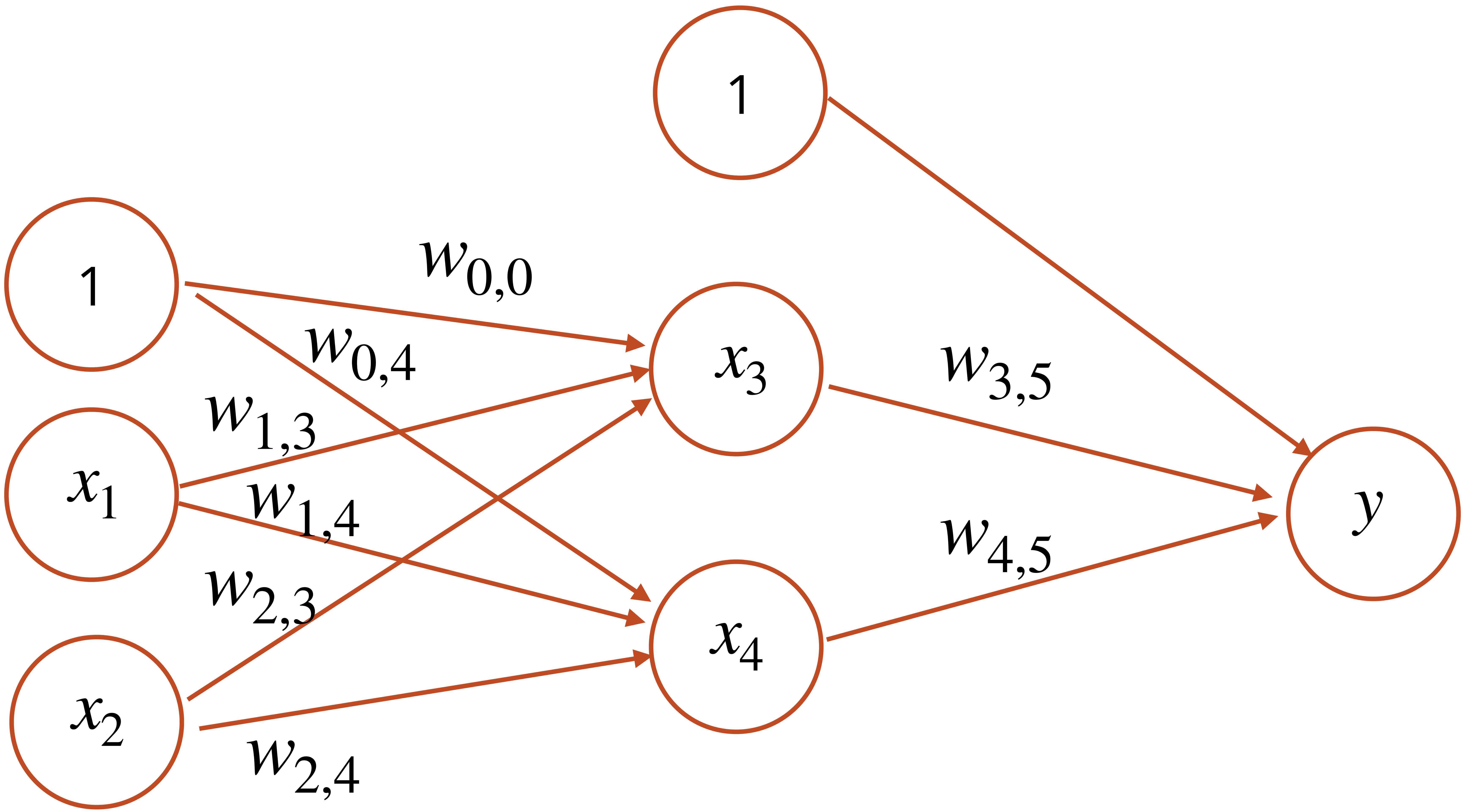
Weaknesses

1. Where do these weights come from?
2. How does the network know words and frequencies?

Table 1
Parameter Values Used in the Simulations

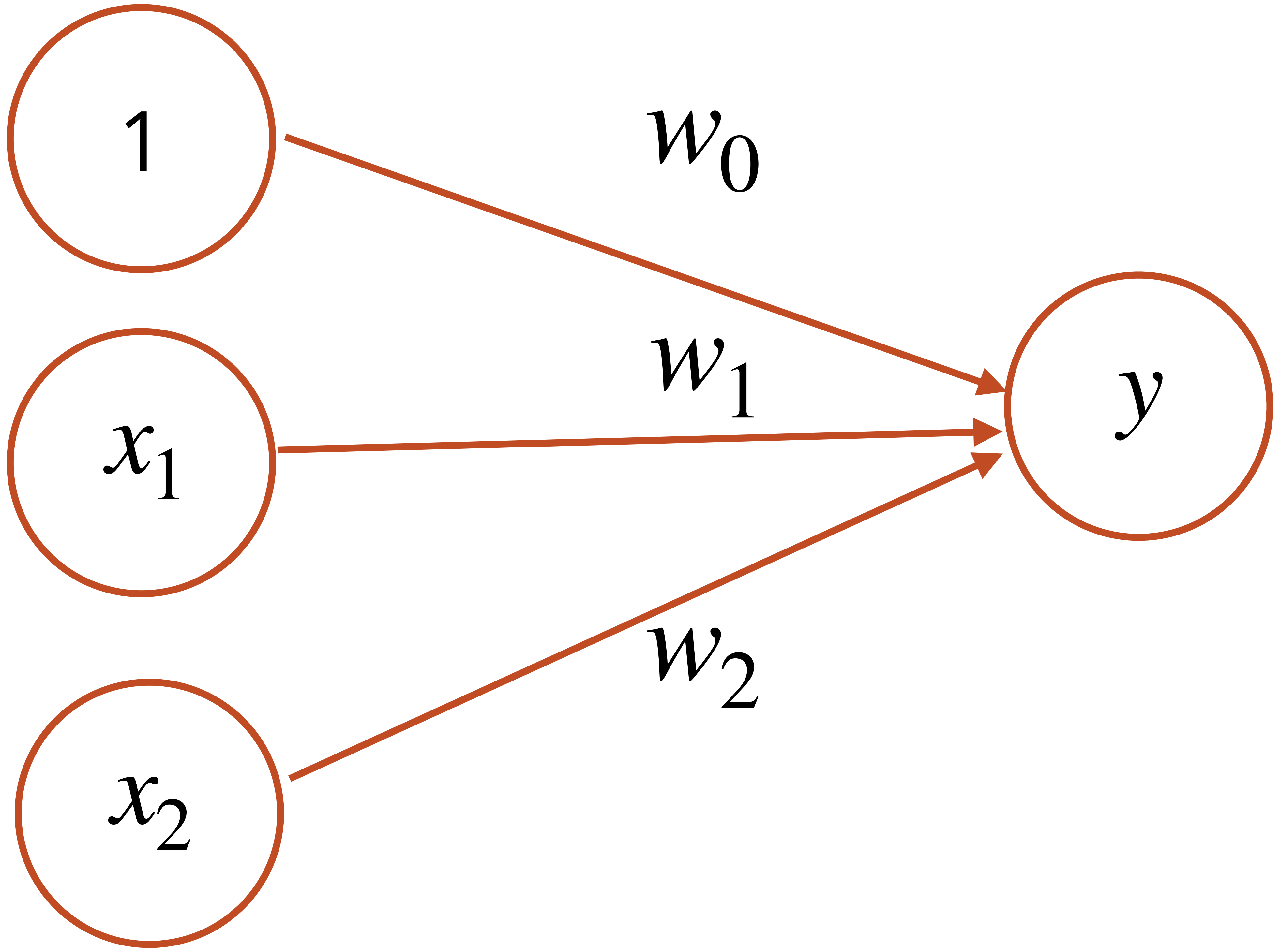
Parameter	Value
Feature–letter excitation	.005
Feature–letter inhibition	.15
Letter–word excitation	.07
Letter–word inhibition	.04
Word–word inhibition	.21
Letter–letter inhibition	0
Word–letter excitation	.30

But how do we learn connections weights in a multi-layer network?



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

What does it mean to learn in a neural network?



We got (x_1, x_2)

We computed $\hat{y} = f(w_0 + w_1 x_1 + w_1 x_1)$

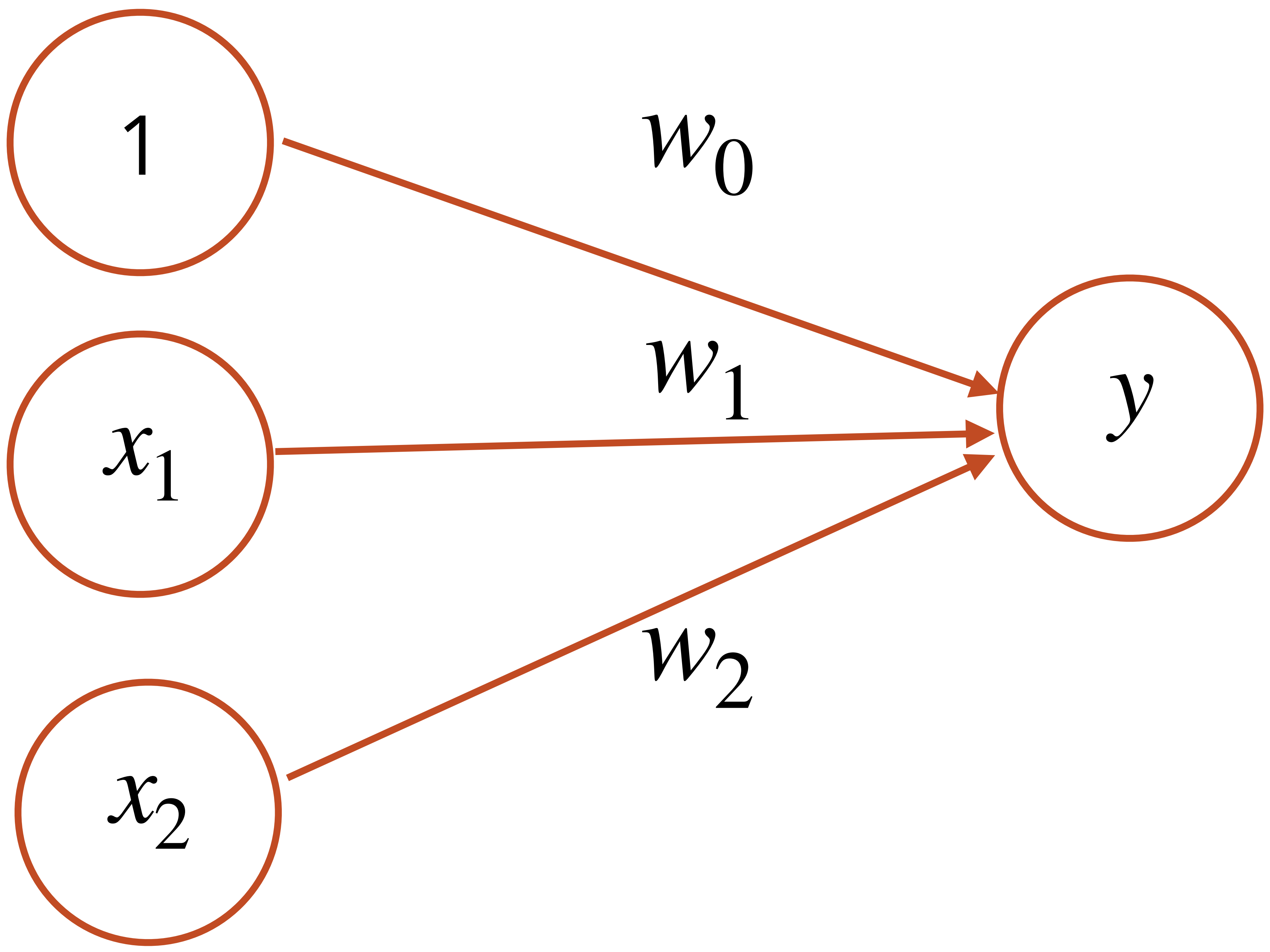
But we wanted to predict y!

Now we want to change w_0, w_1, x_2

So next time we see (x_1, x_2)

We predict something closer to y

Aside: Learning rates



So next time we see (x_1, x_2)

We predict something closer to y

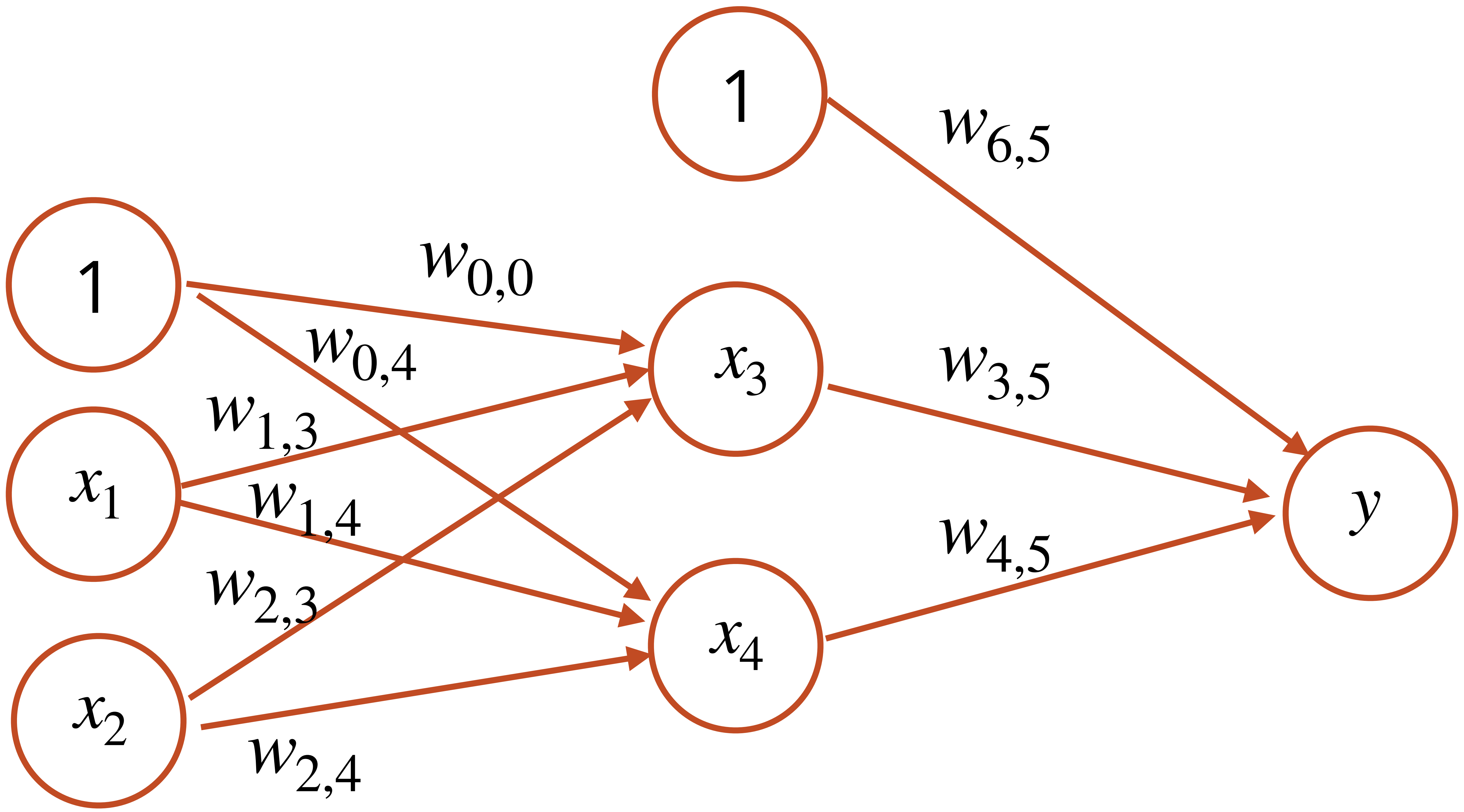
Why not predict exactly y ?

$$\Delta w_i = \alpha \cdot (y - \hat{y})\, x_i$$

Credit assignment in multi-layer networks

If you have an error, in $\hat{y}$, who do you blame?

Suppose we find that $x_3 \cdot w_{3,5}$ caused $\hat{y}$ to be too high



Gradient Descent

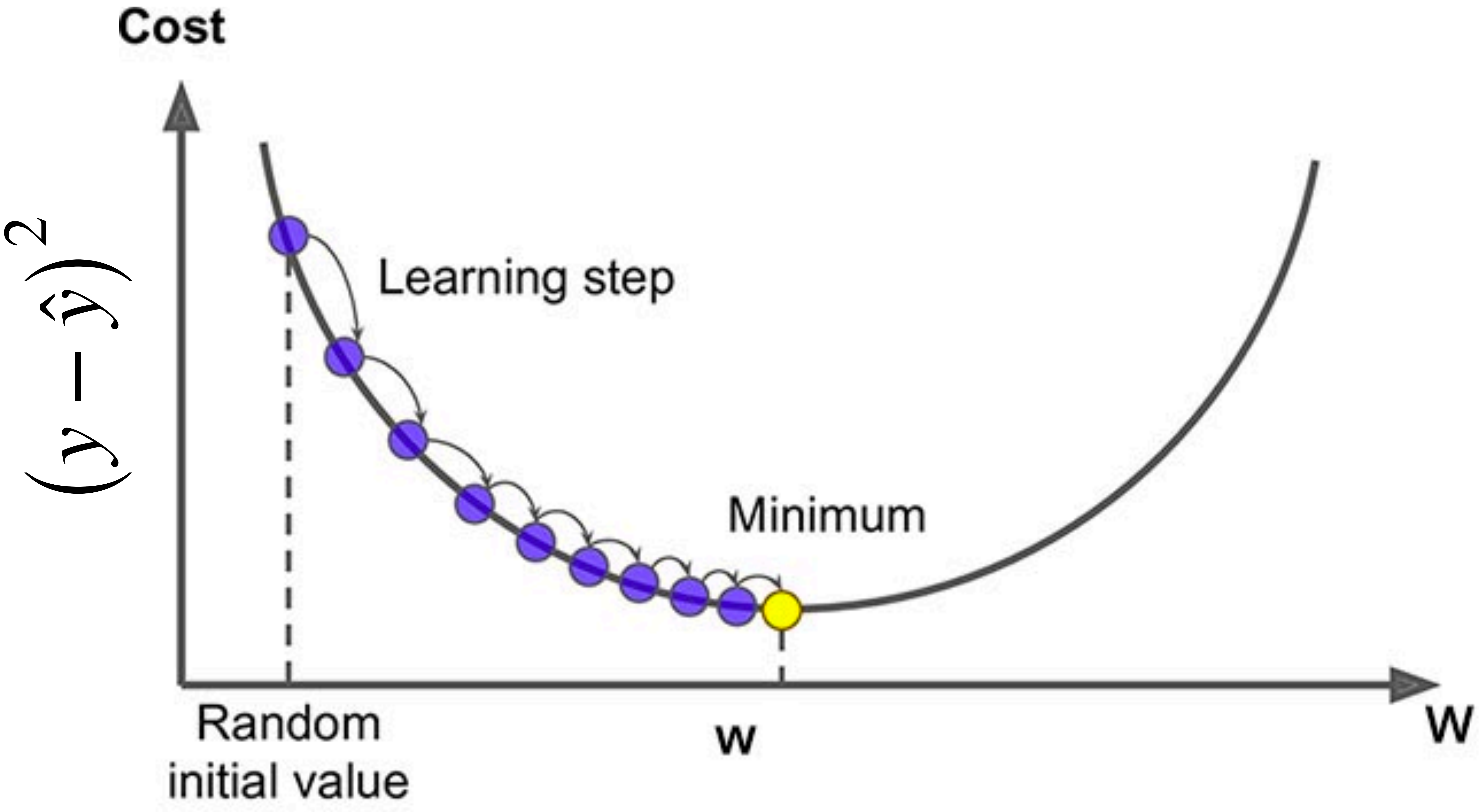
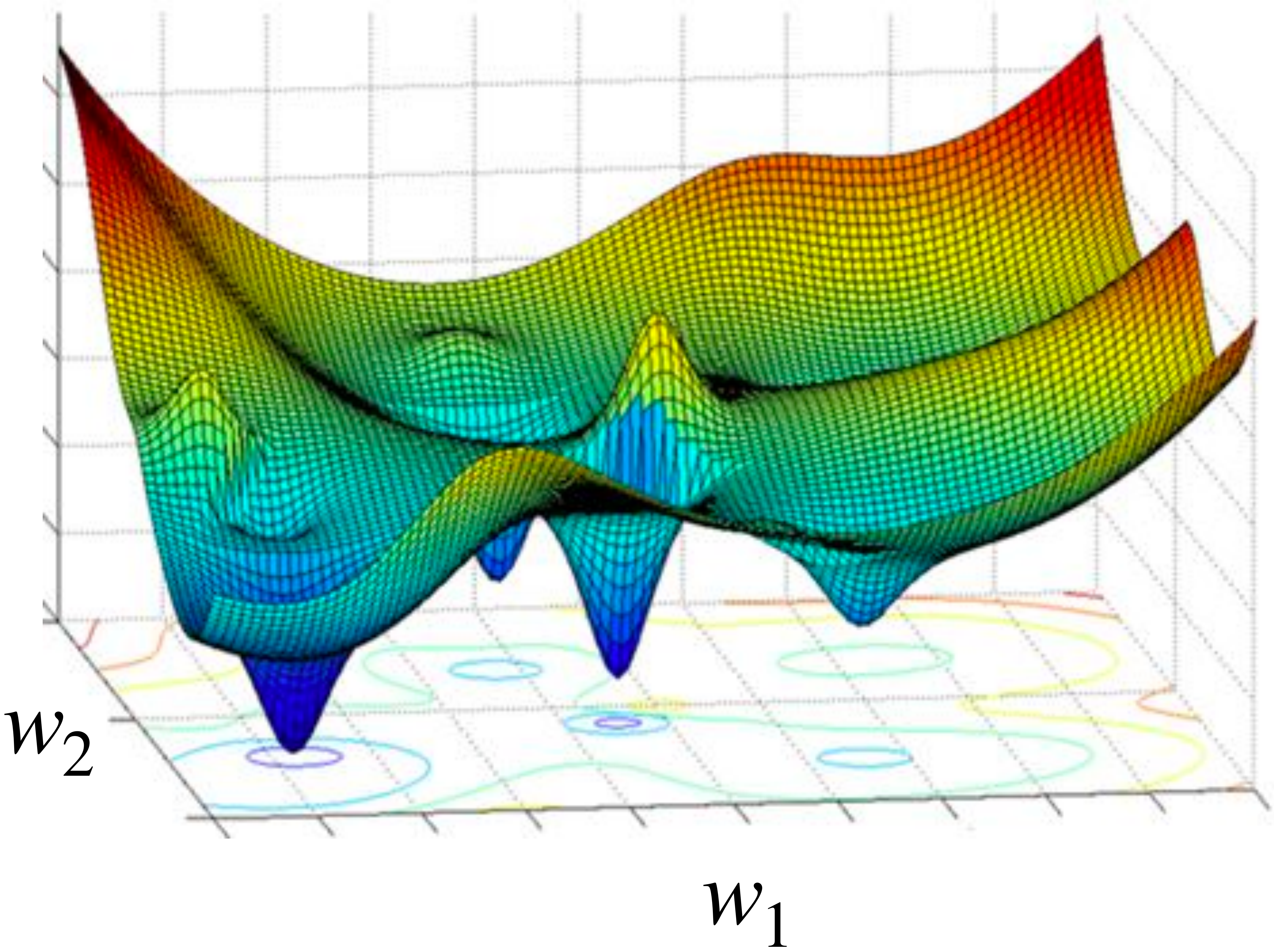


Image from Saugat Bhattari

Gradient Descent

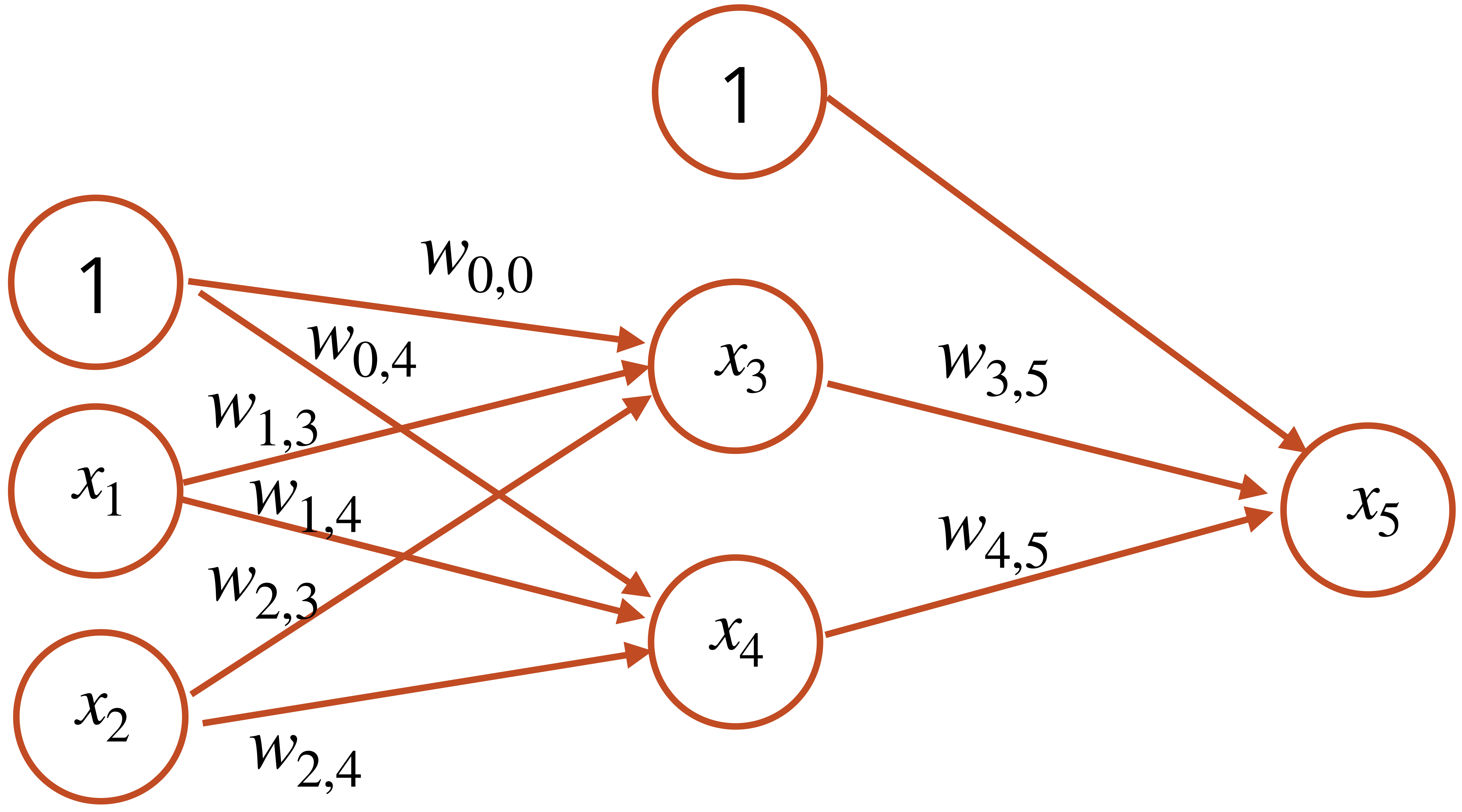


Credit assignment in multi-layer networks

Suppose we find that $x_3 \cdot w_{3,5}$ caused $\hat{y}$ to be too high

Want to separate error parts:

1. Error cause by $w_{3,5}$
2. Error caused by x_3



We're going to do this using partial derivatives

Updating one weight

Terms:

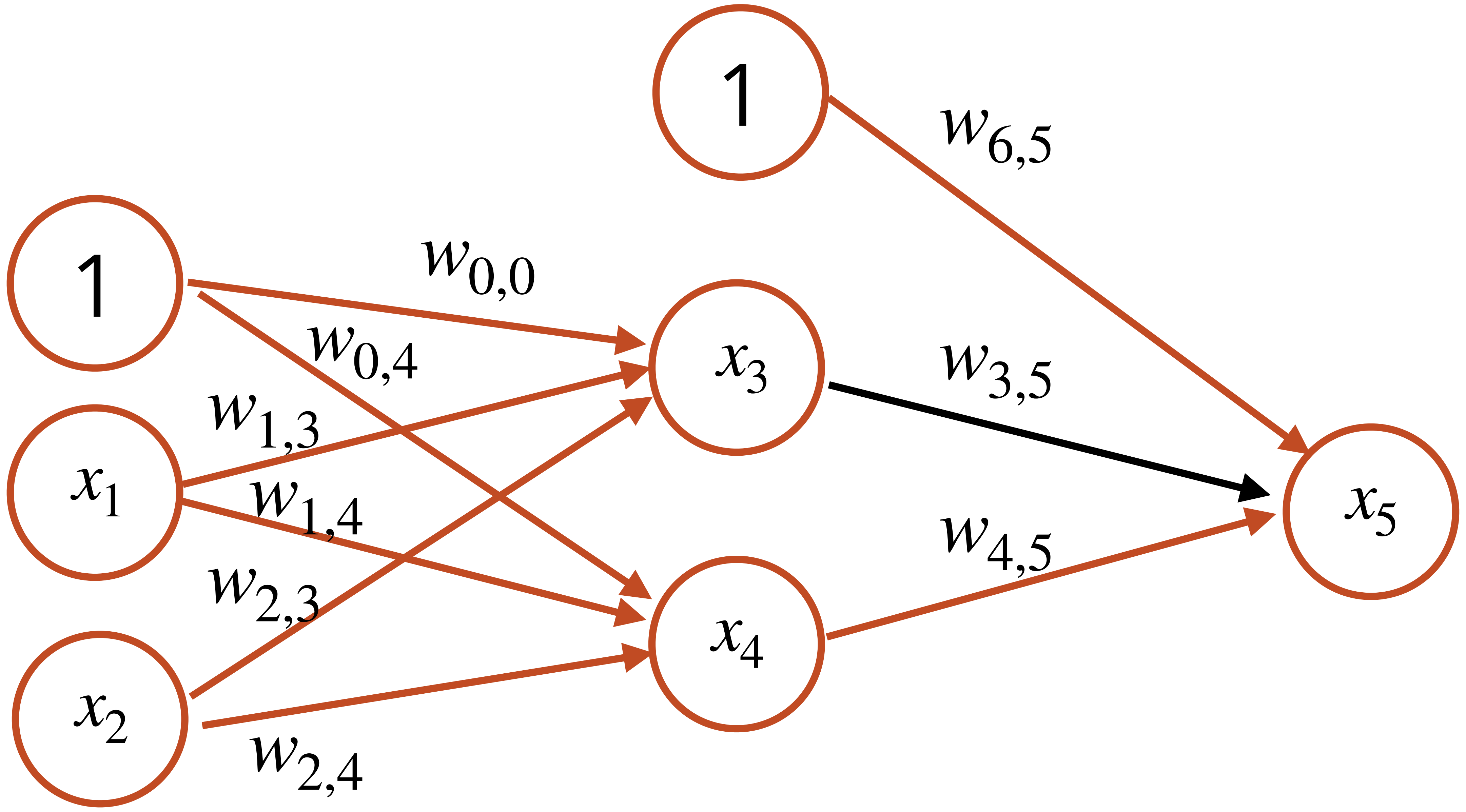
E Squared Prediction error

x_5 The summed input to $x_5 = w_{6,5} + w_{3,5} \cdot x_3 + w_{4,5} \cdot x_4$

a_{x_5} The activation of $x_5 = \frac{1}{1 - e^{x_5}} = \sigma\left(x_5\right)$

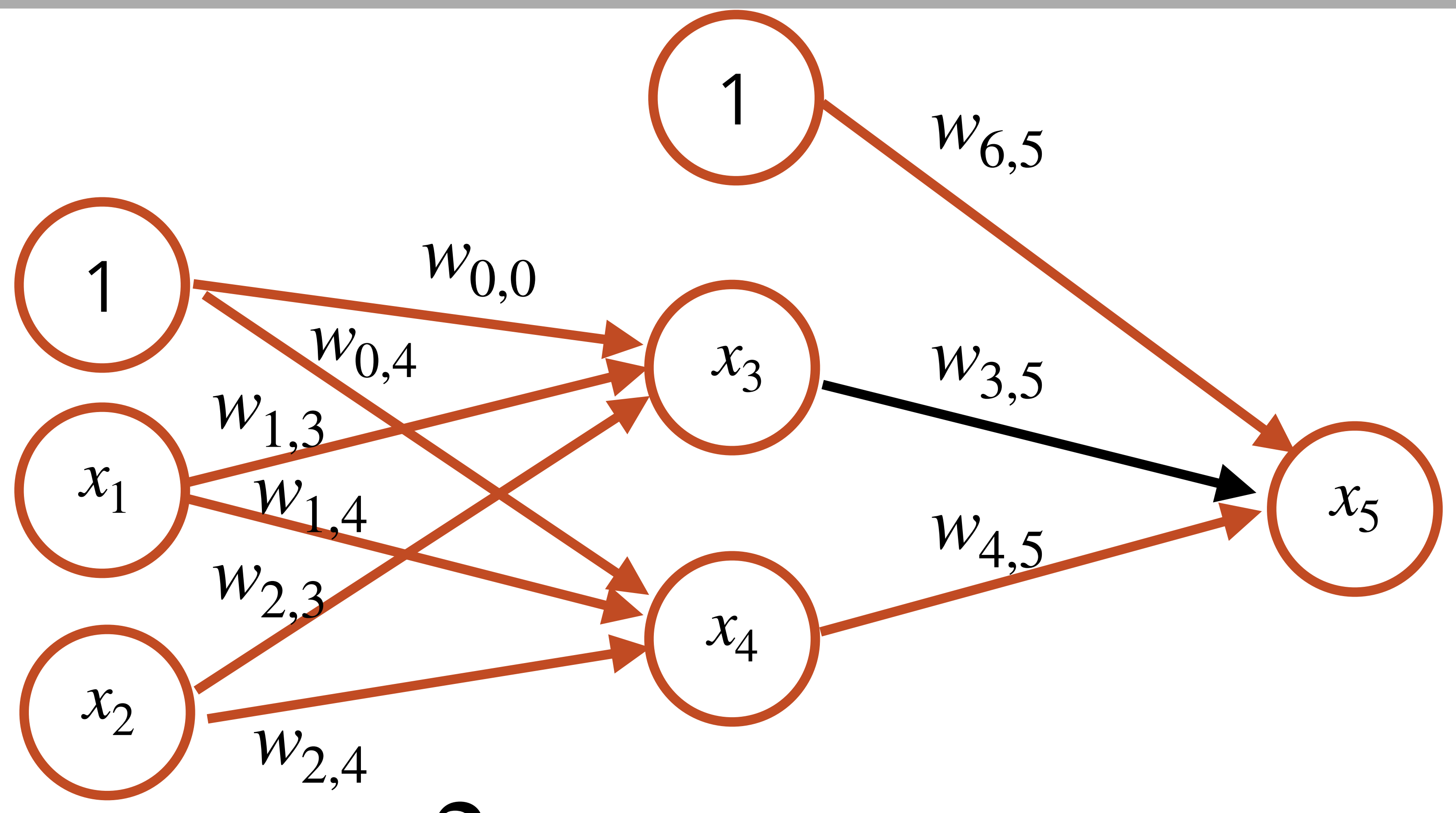
$$\frac{\partial E}{\partial w_{3,5}} = \frac{\partial x_5}{\partial w_{3,5}} \frac{\partial a_{x_5}}{\partial x_5} \frac{\partial E}{\partial a_{x_5}}$$

By the chain rule



Updating one weight

$$\frac{\partial E}{\partial w_{3,5}} = \frac{\partial x_5}{\partial w_{3,5}} \frac{\partial a_{x_5}}{\partial x_5} \frac{\partial E}{\partial a_{x_5}}$$



$$\frac{\partial E}{\partial a_{x_5}} = 2\left(y - a_{x_5}\right)$$

$$E = \left(y - a_{x_5}\right)^2$$

$$\frac{\partial a_{x_5}}{\partial x_5} = \sigma\left(x_5\right)\left(1 - \sigma\left(x_5\right)\right)$$

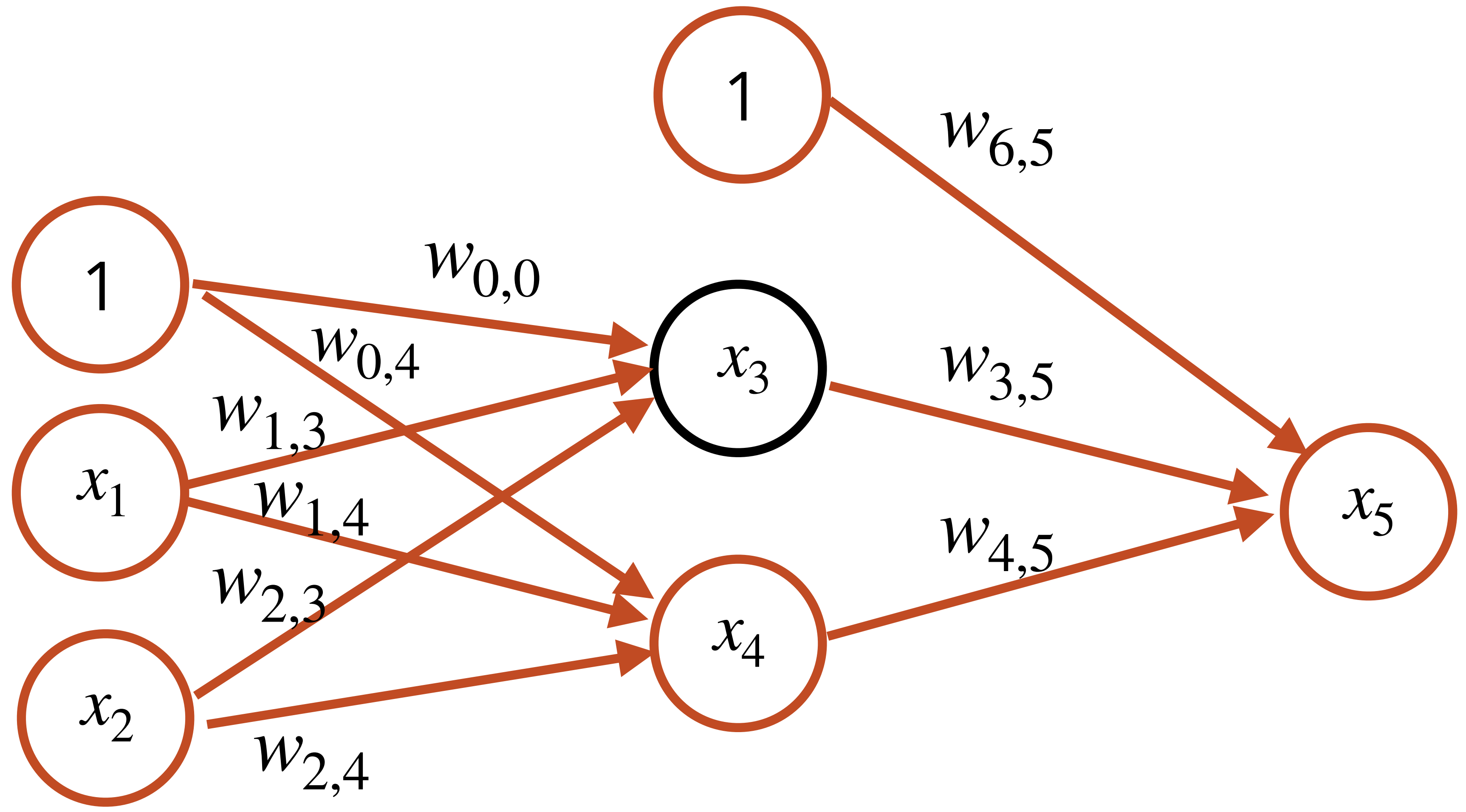
$$\sigma'(x) = \sigma(x)\left(1 - \sigma(x)\right)$$

$$\frac{\partial x_5}{\partial w_{3,5}} = a_{x_3}$$

$$x_5 = w_{6,5} + a_{w_{3,5}} \cdot a_{x_3} + w_{4,5} \cdot a_{x_4}$$

The gradient for one activation

$$\frac{\partial E}{\partial a_{x_3}} = \frac{\partial x_5}{\partial a_{x_3}} \frac{\partial a_{x_5}}{\partial x_5} \frac{\partial E}{\partial a_{x_5}}$$



$$\frac{\partial E}{\partial a_{x_5}} = 2\left(y - a_{x_5}\right)$$

$$E = \left(a_{x^5} - y\right)^2$$

$$\frac{\partial a_{x_5}}{\partial x_5} = \sigma\left(x_5\right)\left(1 - \sigma\left(x_5\right)\right)$$

$$\sigma'(x) = \sigma(x)\left(1 - \sigma(x)\right)$$

$$\frac{\partial x_5}{\partial a_{x_3}} = w_{3,5}$$

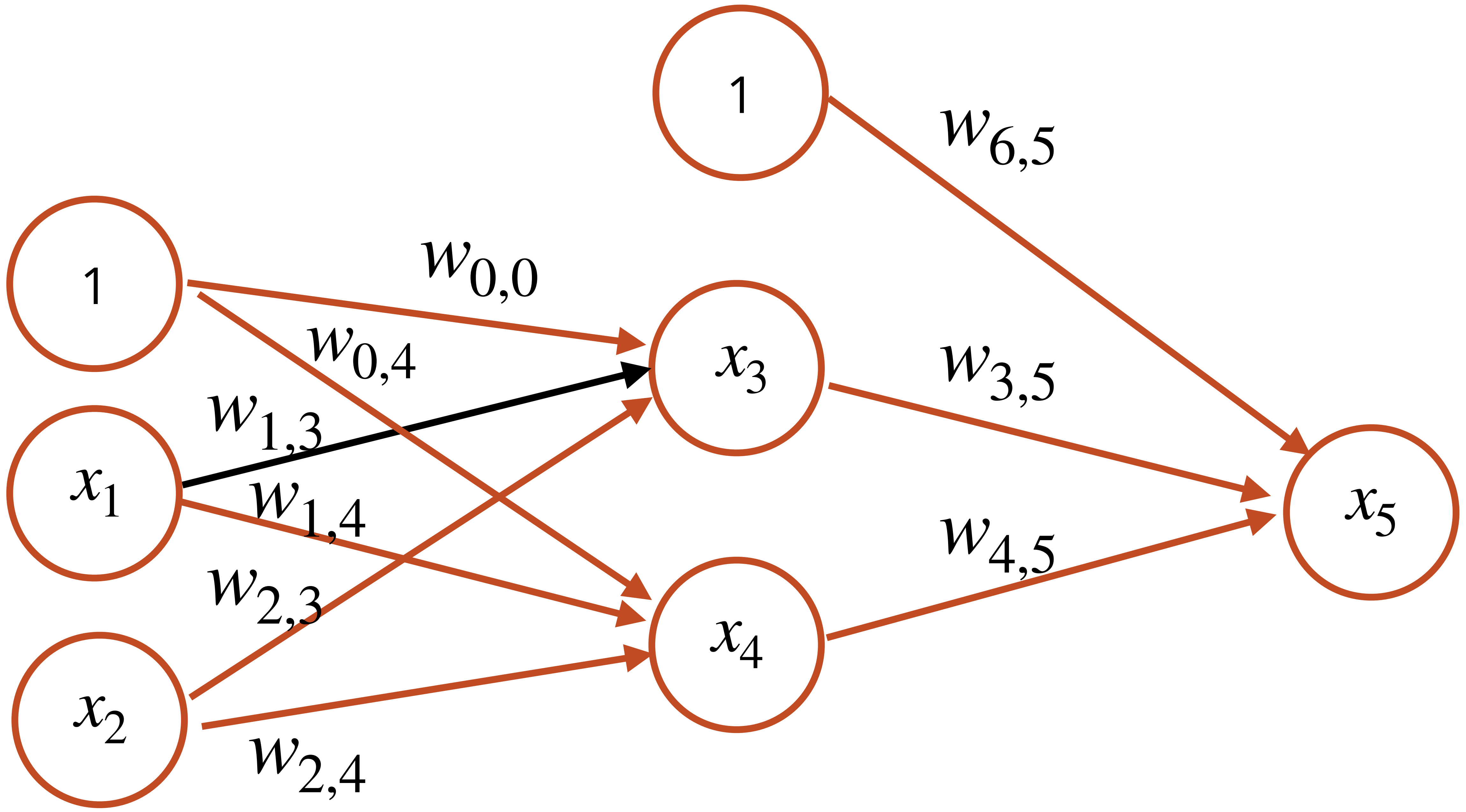
$$x_5 = w_{6,5} + a_{w_{3,5}} \cdot a_{x_3} + w_{3,5} \cdot a_{x_4}$$

Backpropagation

$$\frac{\partial E}{\partial w_{1,3}} = \frac{\partial x_3}{\partial w_{1,3}} \frac{\partial a_{x_3}}{\partial x_3} \frac{\partial E}{\partial a_{x_3}}$$

$$= a_{x_1} \cdot \sigma'\left(x_3\right) \frac{\partial E}{\partial a_{x_3}}$$

$$= a_{x_1} \cdot \sigma'\left(x_3\right) \frac{\partial x_5}{\partial a_{x_3}} \frac{\partial a_{x_5}}{\partial x_5} \frac{\partial E}{\partial a_{x_5}}$$



Can we use these ideas to model semantic memory?

Semantic Cognition: Our intuitive understanding of concepts and their properties (e.g. birds lay eggs, dogs have 4 legs)

Questions:

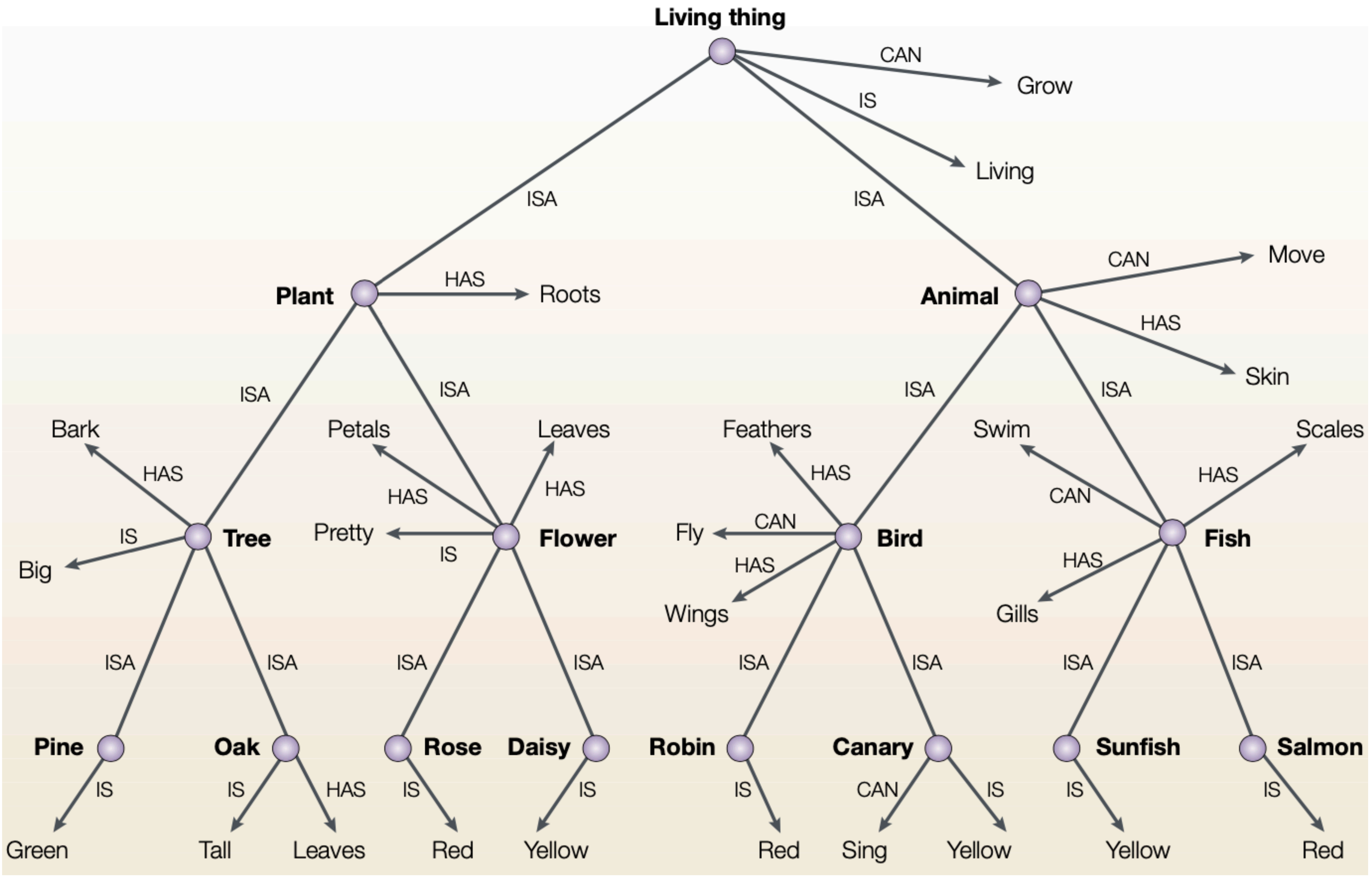
1. How do we know what properties a concept has and how they should be generalized?
2. How is this knowledge acquired?
3. How does it degrade?

A classical model of semantic concepts (Quillan, 1968)

Concepts organized hierarchically from general to specific

Propositions stored once at highest level to which they apply

Strengths: Efficient, new concepts inherit a lot of information



Weaknesses: How do you handle exceptions?
How do you know where to store a property?

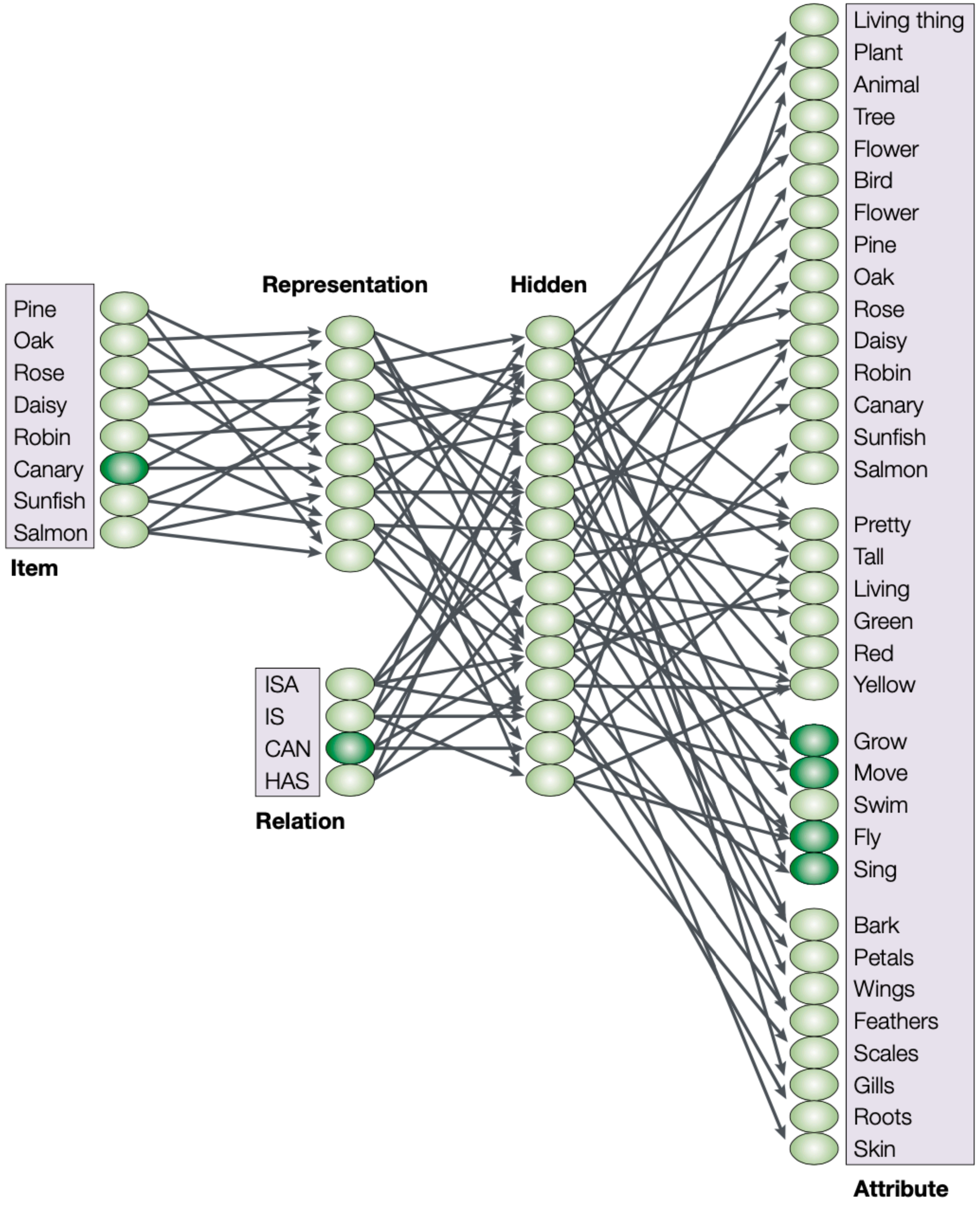
From McClelland & Rogers (2003)

The Rogers & McClelland Model

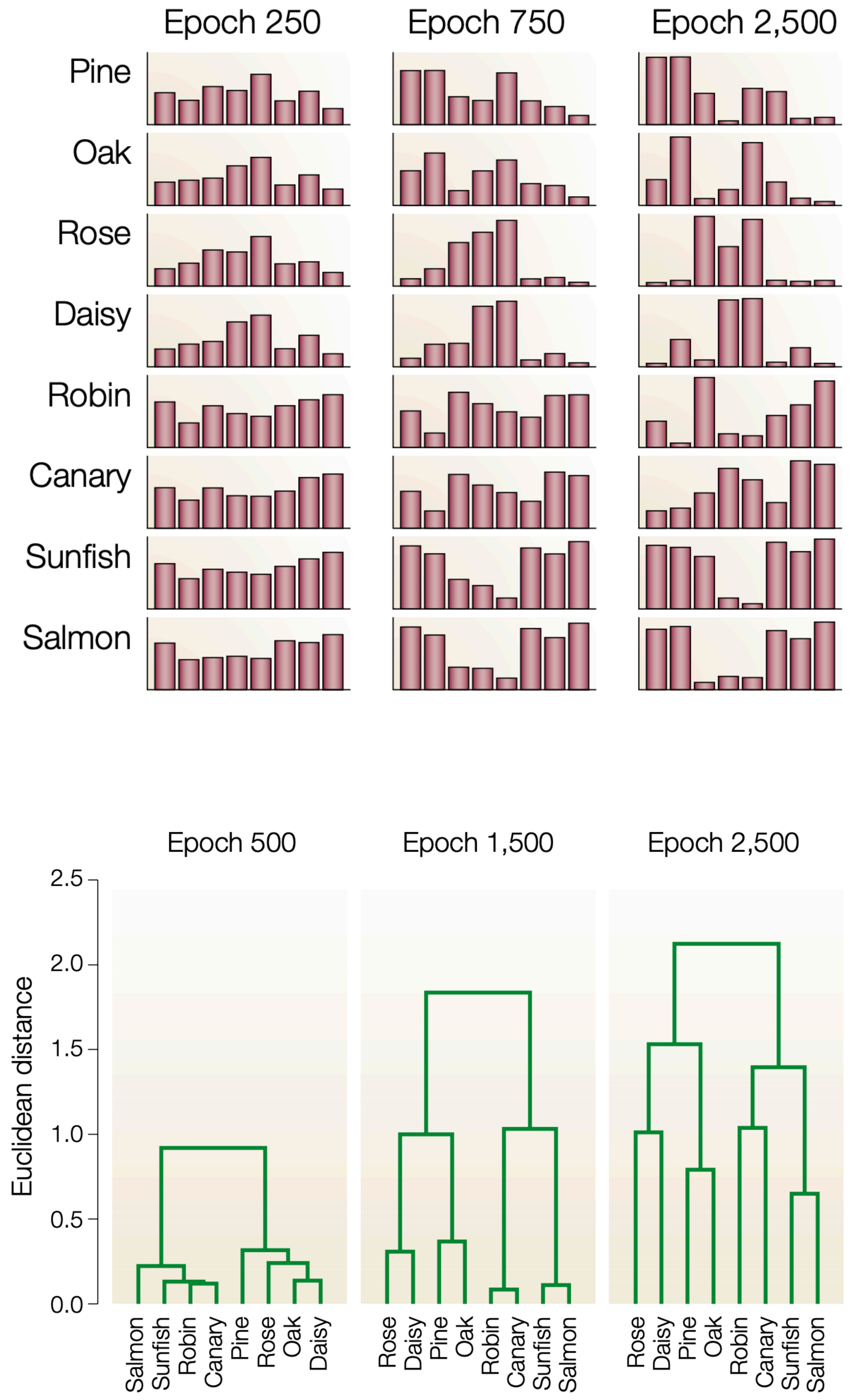
Network trained to answer triplet questions: Given **item** and **relation**, output **attributes**

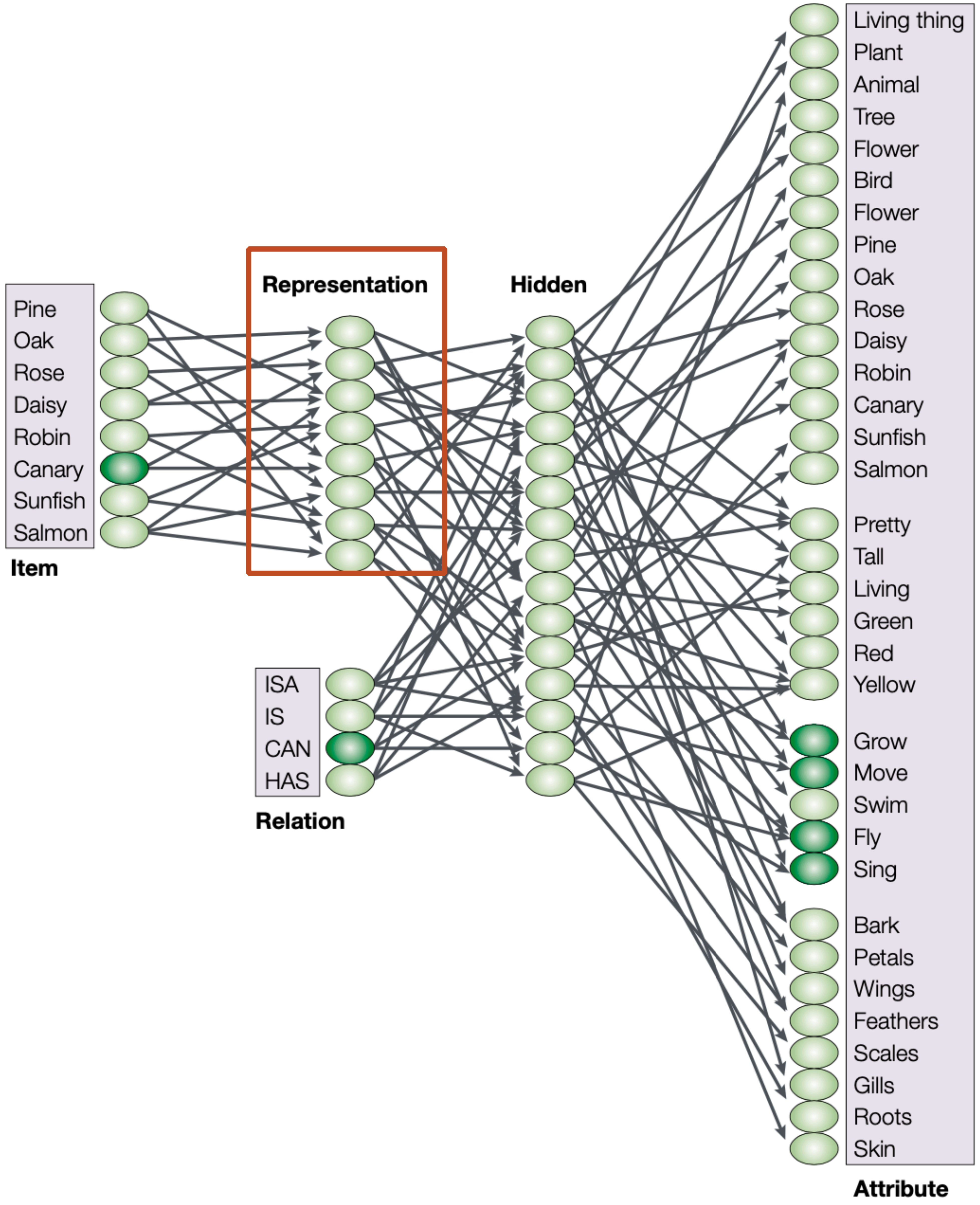
No explicit hierarchy

Started with random weights, trained on Quillan’s data

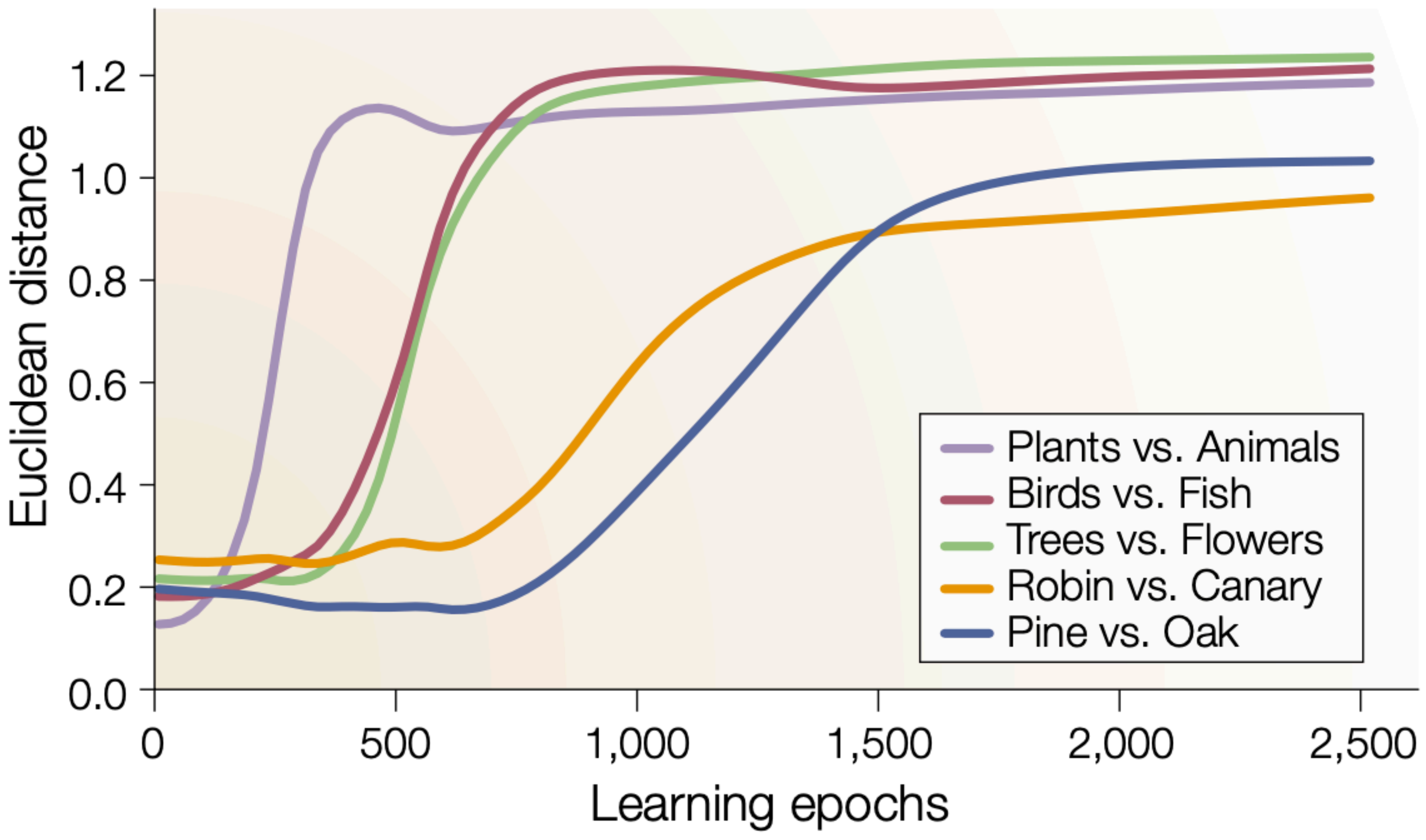


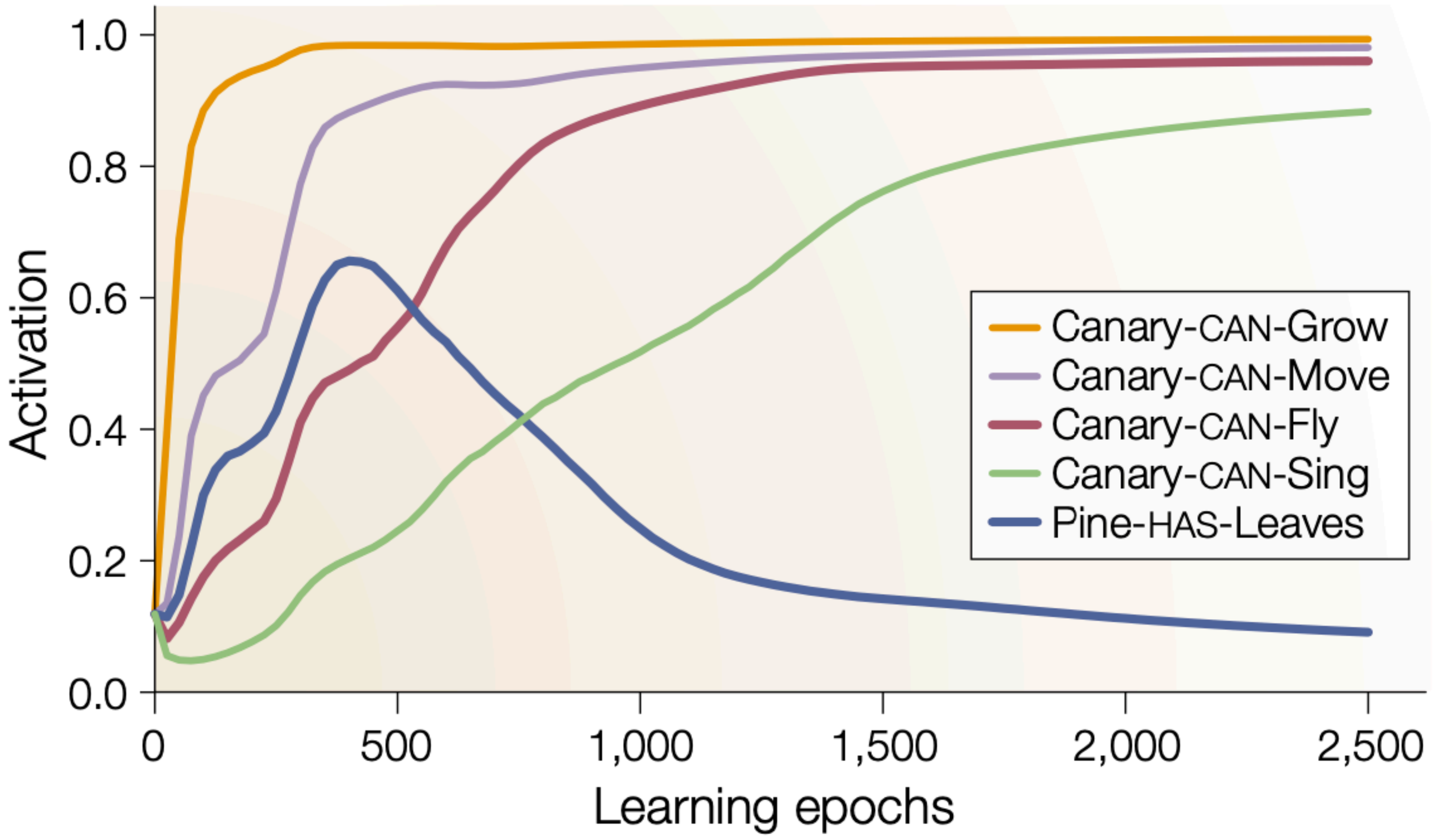
Learning semantic relations through backpropagation





Key result 1: Progressive differentiation





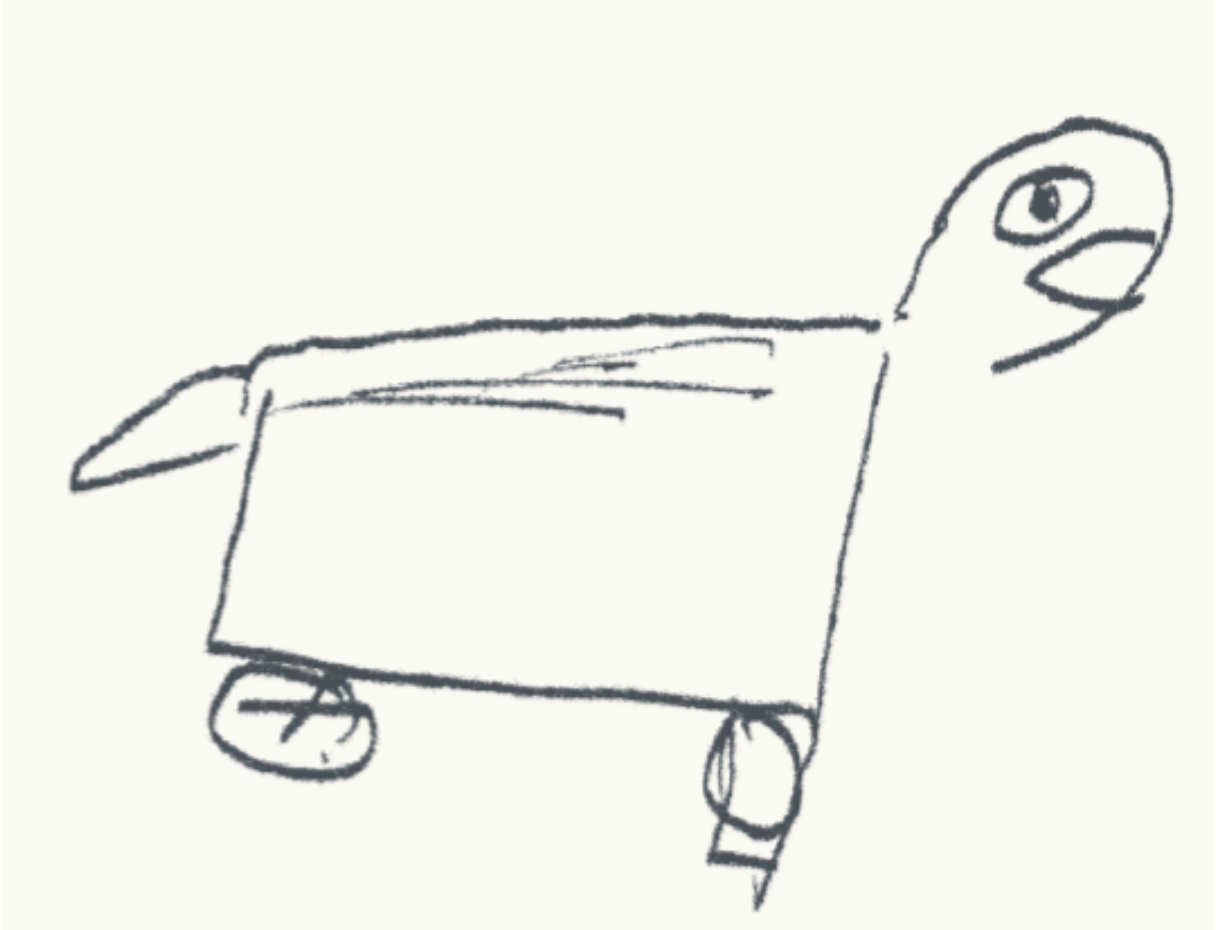
Broad distinctions made first

Broad properties learned first

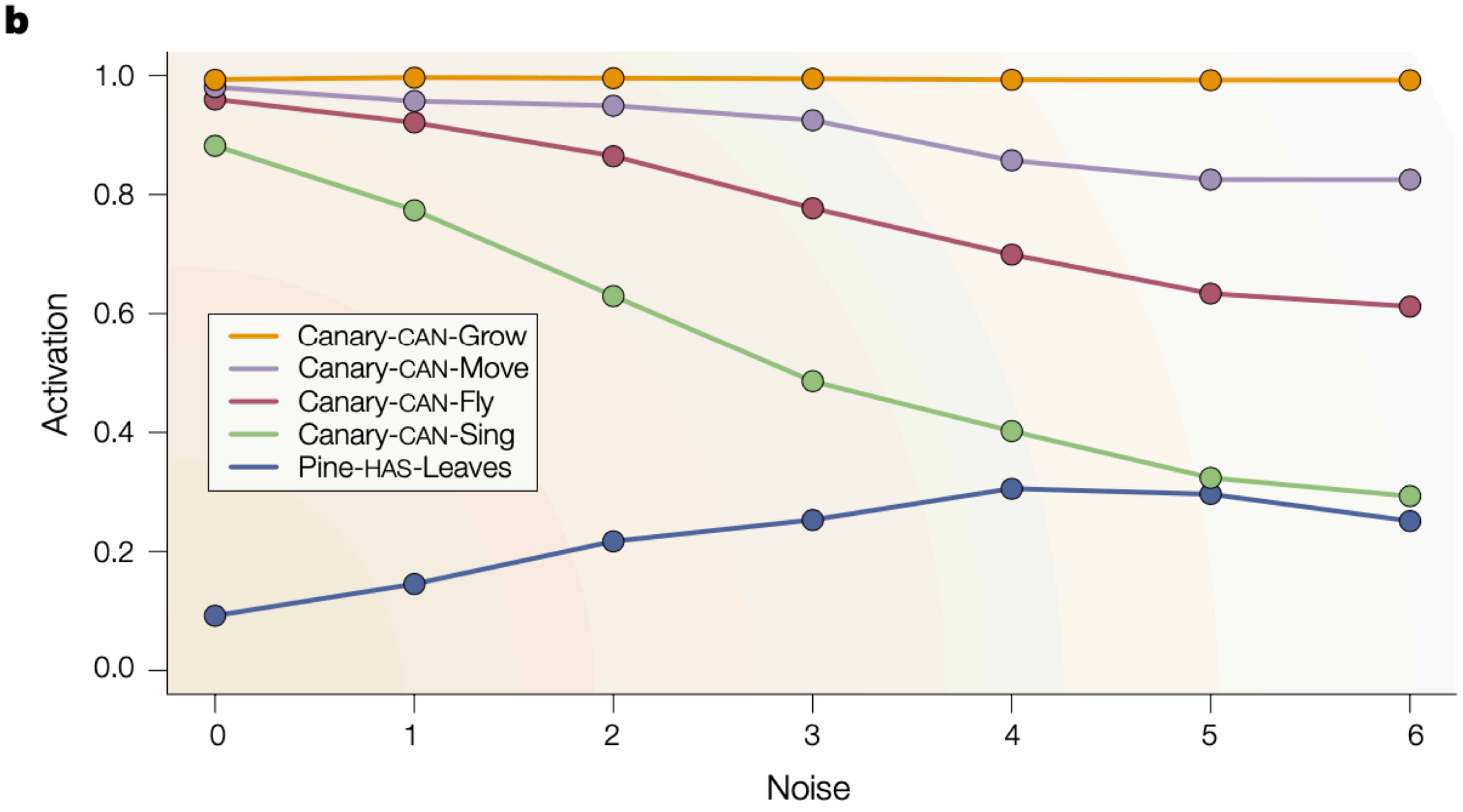
Key result 2: Graceful degradation

Picture naming responses for JL

Item	Sept. 91	March 92	March 93
Bird	+	+	Animal
Chicken	+	+	Animal
Duck	+	Bird	Dog
Swan	+	Bird	Animal
Eagle	Duck	Bird	Horse
Ostrich	Swan	Bird	Animal
Peacock	Duck	Bird	Vehicle
Penguin	Duck	Bird	Part of animal
Rooster	Chicken	Chicken	Dog



Delayed copy of a camel



Noise added to representations disrupts specific features

Multi-layer networks

1. Dynamics of neural networks can capture features of human information processing

2. Backpropagation is a general algorithm for learning in multi-layer networks

3. Neural networks can give rise to “emergent” learning phenomena