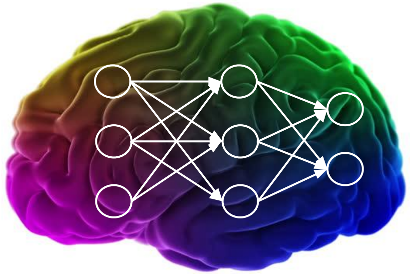
Using neural networks to simulate brain damage: The Farah & McClelland model

Miriam Schulz

UNIVERSITÄT DES SAARLANDES

Course: Kognitive Neuropsychologie
Lecturer: Prof. Axel Mecklinger

2 February 2021



Outline

1. Introduction
2. Principles of connectionist modeling
3. A model of semantic memory impairment: Farah & McClelland (1991)
4. A (simplified) reimplementation of the Farah & McClelland model
5. Conclusion

1. Introduction

Studying cognition

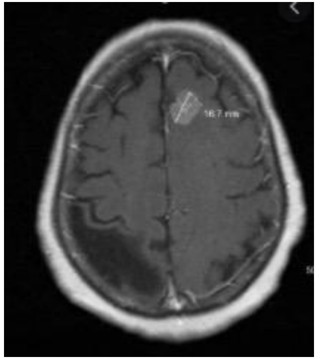
Cognitive neuropsychology

→ study cognition by examining the performance of brain-damaged subjects (changed performance w.r.t. controls)

Studying cognition

Cognitive neuropsychology

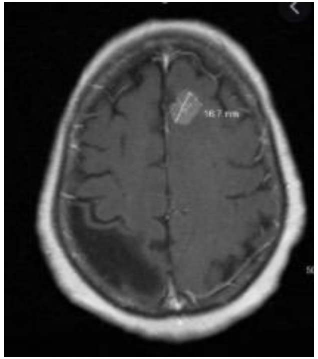
→ study cognition by examining the performance of brain-damaged subjects (changed performance w.r.t. controls)



Studying cognition

Cognitive neuropsychology

→ study cognition by examining the performance of brain-damaged subjects (changed performance w.r.t. controls)



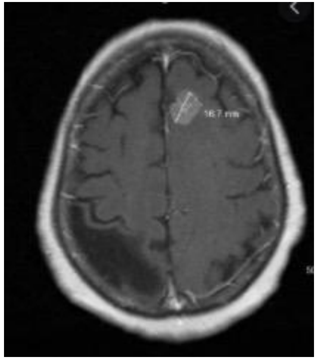
Connectionist modeling

→ study cognition by developing and testing "brain-like" computer models

Studying cognition

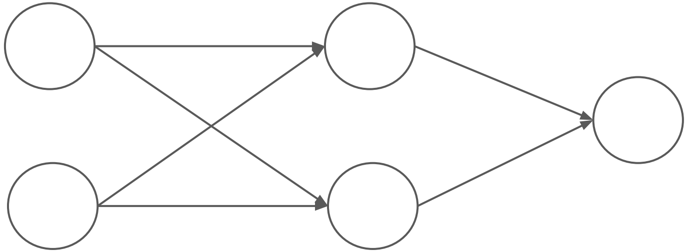
Cognitive neuropsychology

→ study cognition by examining the performance of brain-damaged subjects (changed performance w.r.t. controls)



Connectionist modeling

→ study cognition by developing and testing "brain-like" computer models



This presentation

➔ What do **connectionist models of cognitive processes** look like and how do they work?

This presentation

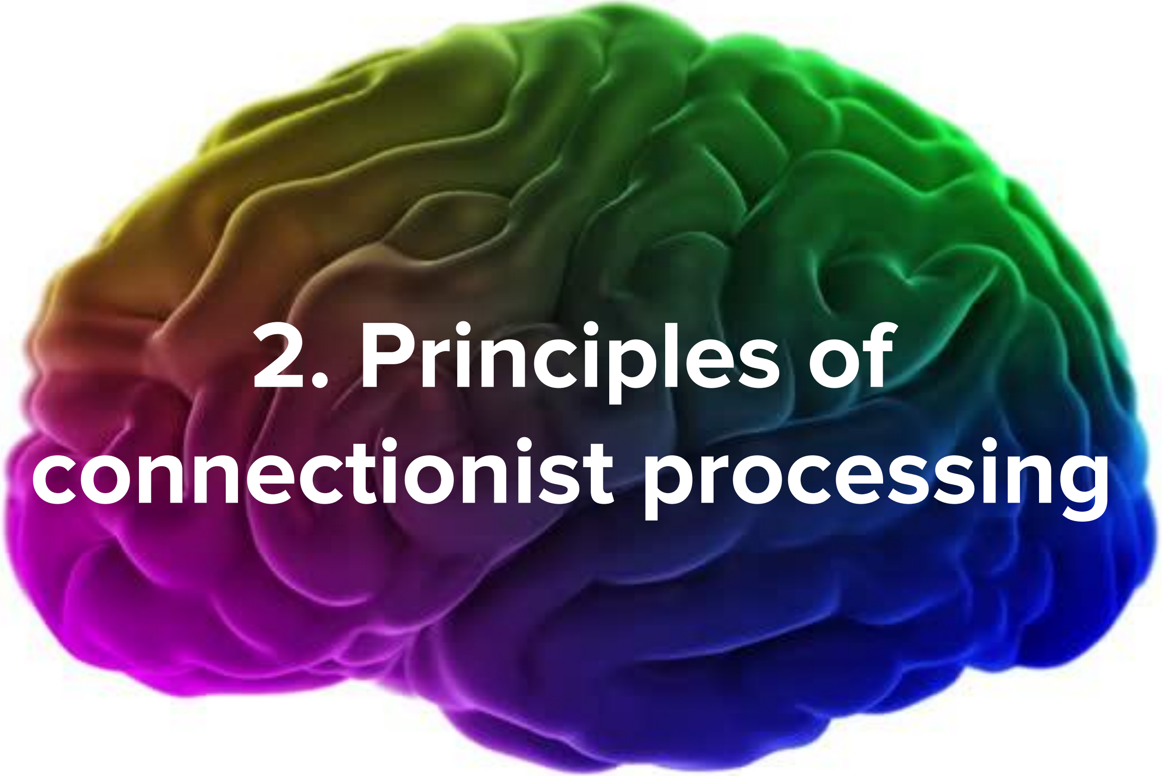
- ➔ What do **connectionist models of cognitive processes** look like and how do they work?
- ➔ Look at one model in particular: **Farah & McClelland (1991), experiment 2**

This presentation

➔ What do **connectionist models of cognitive processes** look like and how do they work?

➔ Look at one model in particular: **Farah & McClelland (1991), experiment 2**

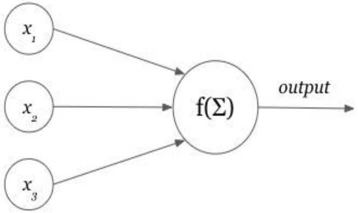
➔ Go through a (simplified) **reimplementation** of the model



2. Principles of connectionist processing

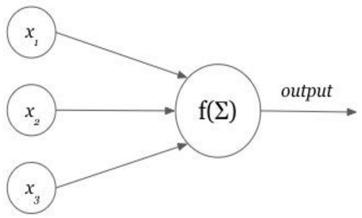
A minimal history of neural networks

- The **McCulloch & Pitts (1943) neuron**: simplest neural model



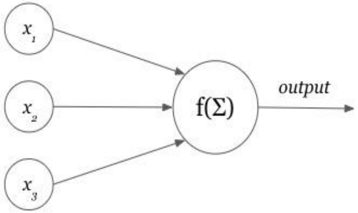
A minimal history of neural networks

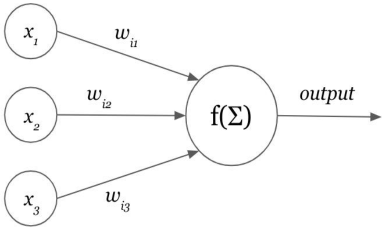
- The **McCulloch & Pitts (1943) neuron**: simplest neural model
- **Hebb's (1949) learning rule** ("neurons that fire together wire together")



A minimal history of neural networks

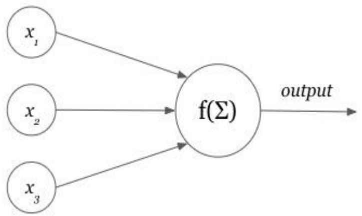
- The **McCulloch & Pitts (1943) neuron**: simplest neural model
- **Hebb's (1949) learning rule** ("neurons that fire together wire together")
- Rosenblatt's (1957, 1962) **perceptron**

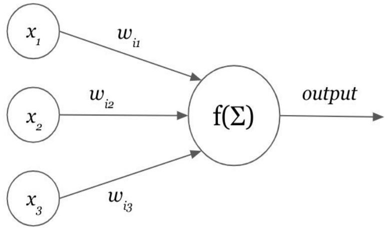




A minimal history of neural networks

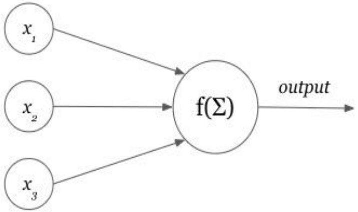
- The **McCulloch & Pitts (1943) neuron**: simplest neural model
- **Hebb's (1949) learning rule** ("neurons that fire together wire together")
- Rosenblatt's (1957, 1962) **perceptron**
- Minsky & Papert (1969): critique of perceptrons

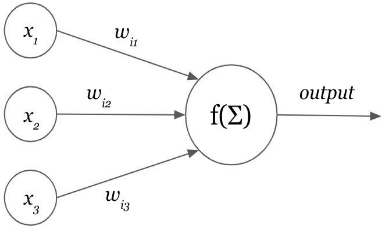




A minimal history of neural networks

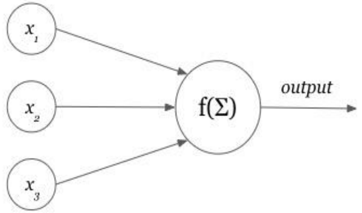
- The **McCulloch & Pitts (1943) neuron**: simplest neural model
- **Hebb's (1949) learning rule** ("neurons that fire together wire together")
- Rosenblatt's (1957, 1962) **perceptron**
- Minsky & Papert (1969): critique of perceptrons
- 1980's + 1990's: boom in connectionist models

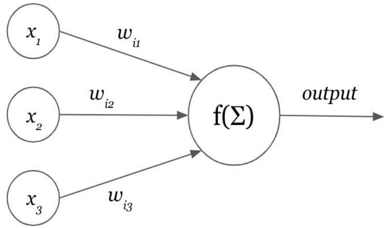




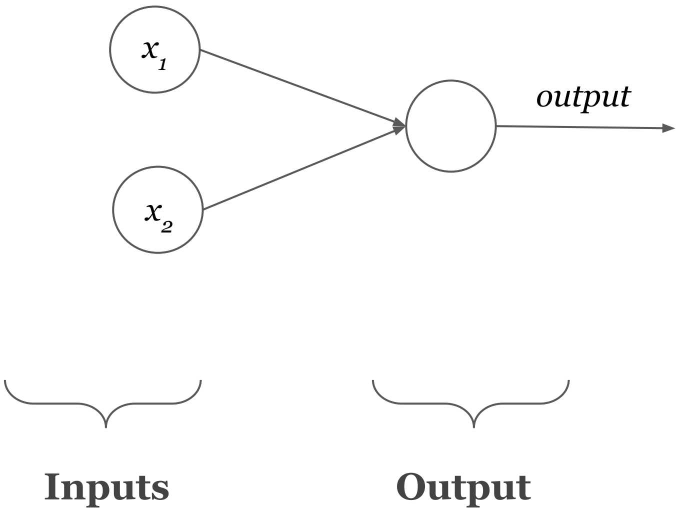
A minimal history of neural networks

- The **McCulloch & Pitts (1943) neuron**: simplest neural model
- **Hebb's (1949) learning rule** ("neurons that fire together wire together")
- Rosenblatt's (1957, 1962) **perceptron**
- Minsky & Papert (1969): critique of perceptrons
- 1980's + 1990's: boom in connectionist models
- In the past decade: huge boom of neural networks/**"deep learning"** in artificial intelligence/machine learning: Google Translate, Alexa, autocorrect, text prediction, recommendation systems like Spotify, ...

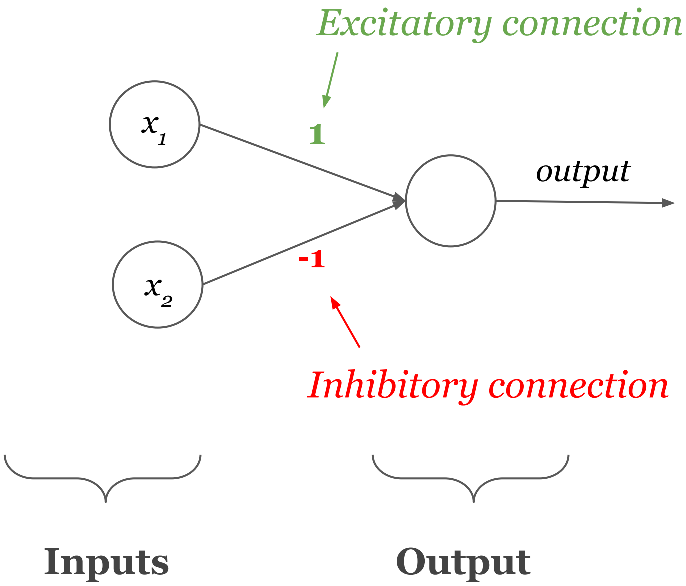




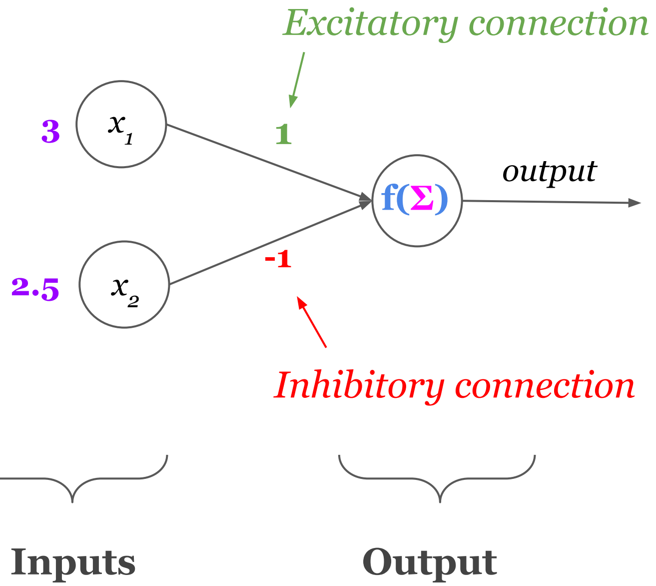
Example of a very small neural network



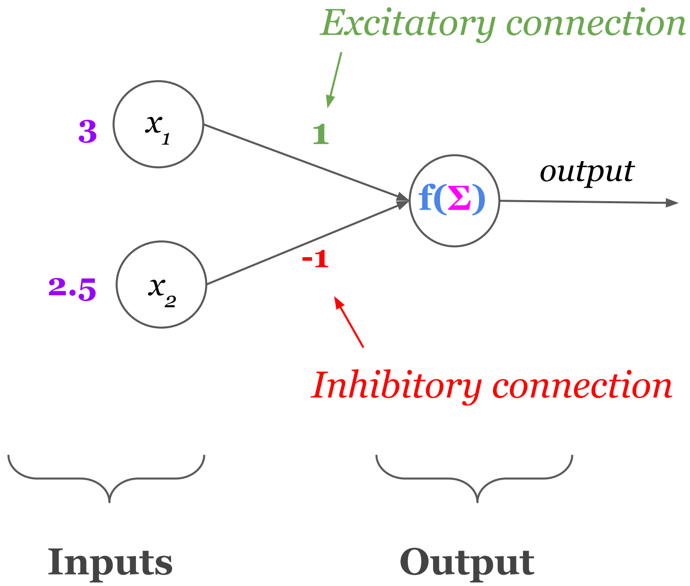
Example of a very small neural network



Example of a very small neural network



Example of a very small neural network

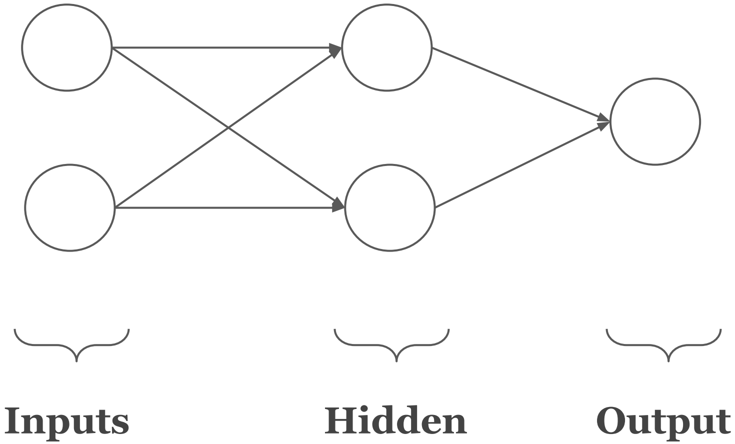


$$\Sigma = 3 * 1 + 2.5 * (-1)$$
$$= 3 - 2.5$$
$$= 0.5$$

$$f(x) = 1 \text{ if } x > 0$$
$$= 0 \text{ if } x <= 0$$

$$f(\Sigma) = f(0.5) = 1$$

A (slightly) more complex network



Why make computational models of cognition?

- Computational modelling requires a theory to be **fully specified** (remove potential ambiguity, make everything explicit and consistent)

Why make computational models of cognition?

- Computational modelling requires a theory to be **fully specified** (remove potential ambiguity, make everything explicit and consistent)
- **Break down the complexity** of a model

Why make computational models of cognition?

- Computational modelling requires a theory to be **fully specified** (remove potential ambiguity, make everything explicit and consistent)
- **Break down the complexity** of a model
- Make **precise, quantitative** predictions ("testable")

Why make computational models of cognition?

- Computational modelling requires a theory to be **fully specified** (remove potential ambiguity, make everything explicit and consistent)
- **Break down the complexity** of a model
- Make **precise, quantitative** predictions ("testable")
- Not a replacement, but a **complement** to verbal theories or box-arrow diagrams; an **additional test** for the consistency and completeness of a theory

Why make computational models of cognition?

Level 1 (computational)	Input-output behavior
Level 2 (algorithmic)	Mind-as-symbol-manipulator hypothesis vs. “brain-like” connectionist models
Level 3 (implementational)	Neurons vs. transistors in computers

Why make computational models of cognition?

Some useful features of neural networks for cognitive modelling:

Why make computational models of cognition?

Some useful features of neural networks for cognitive modelling:

➔ **Graceful degradation** (damaged input / damaged networks)

Why make computational models of cognition?

Some useful features of neural networks for cognitive modelling:

➔ **Graceful degradation** (damaged input / damaged networks)
➔ **Content addressability** (retrieve information from a partial specification)

Why make computational models of cognition?

Some useful features of neural networks for cognitive modelling:

➔ **Graceful degradation** (damaged input / damaged networks)
➔ **Content addressability** (retrieve information from a partial specification)
➔ **Generalization** to novel inputs

Why make computational models of cognition?

Some useful features of neural networks for cognitive modelling:

- ➔ **Graceful degradation** (damaged input / damaged networks)
- ➔ **Content addressability** (retrieve information from a partial specification)
- ➔ **Generalization** to novel inputs

These properties “come for free”; networks not specifically trained for them.

3. The Farah & McClelland model

The double dissociation between living and non-living things

Table 1

Performance of Two Patients With Impaired Knowledge of Living Things on Various Semantic Memory Tasks

Case	Living thing	Nonliving thing
	Picture identification	
JBR	6%	90%
SBY	0%	75%
	Spoken word definition	
JBR	8%	79%
SBY	0%	52%

The double dissociation between living and non-living things

Table 2
Performance of Two Patients With Impaired Knowledge of Nonliving Things on Various Semantic Memory Tasks

Case	Category: Animal	Flower	Object
Spoken word–picture matching			
VER	86%	96%	63%
YOT	86%	86%	67%
Picture–picture matching			
YOT	100%	—	69%

The sensory-functional hypothesis

Visual and functional knowledge play different roles in the representation of living and non-living things:

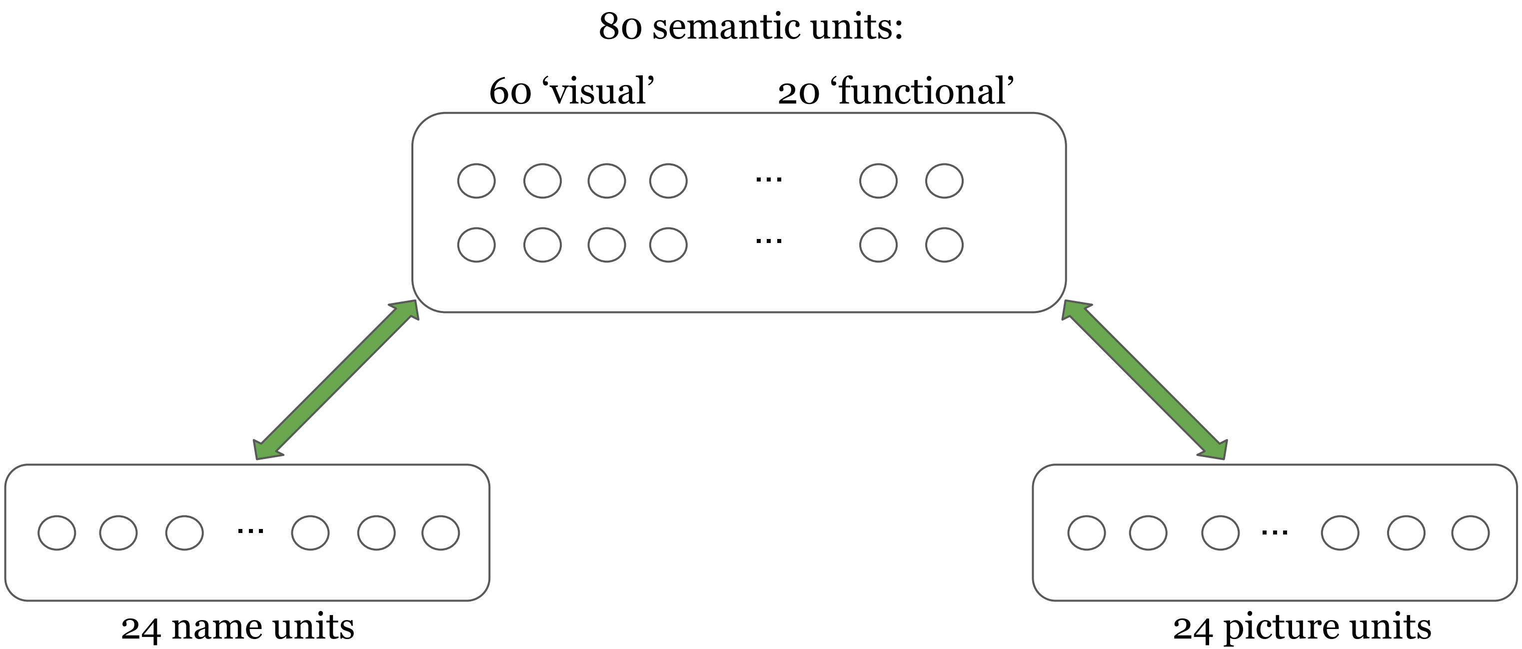
- Living things like ‘dog’ are represented as more **visual** in semantic memory
- Non-living things like ‘hammer’ are represented as more **functional** in semantic memory

Experiment 1

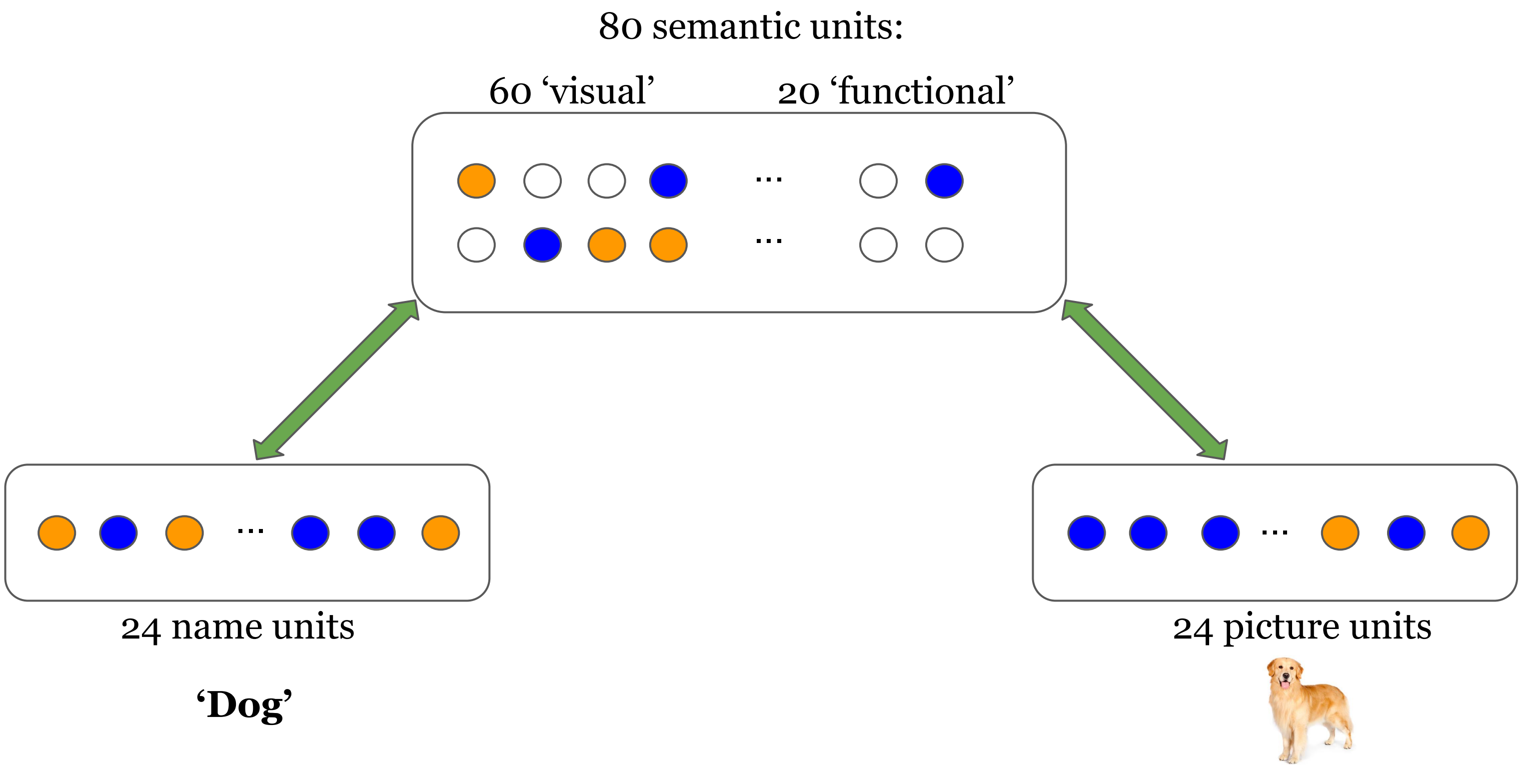
Visual and functional knowledge play different roles in the representation of living and non-living things:

- On average **2.68 visual** and only **0.35 functional** descriptors used to describe living things
- Vs. **1.57 visual** and **1.11 functional** descriptors used to describe non-living things

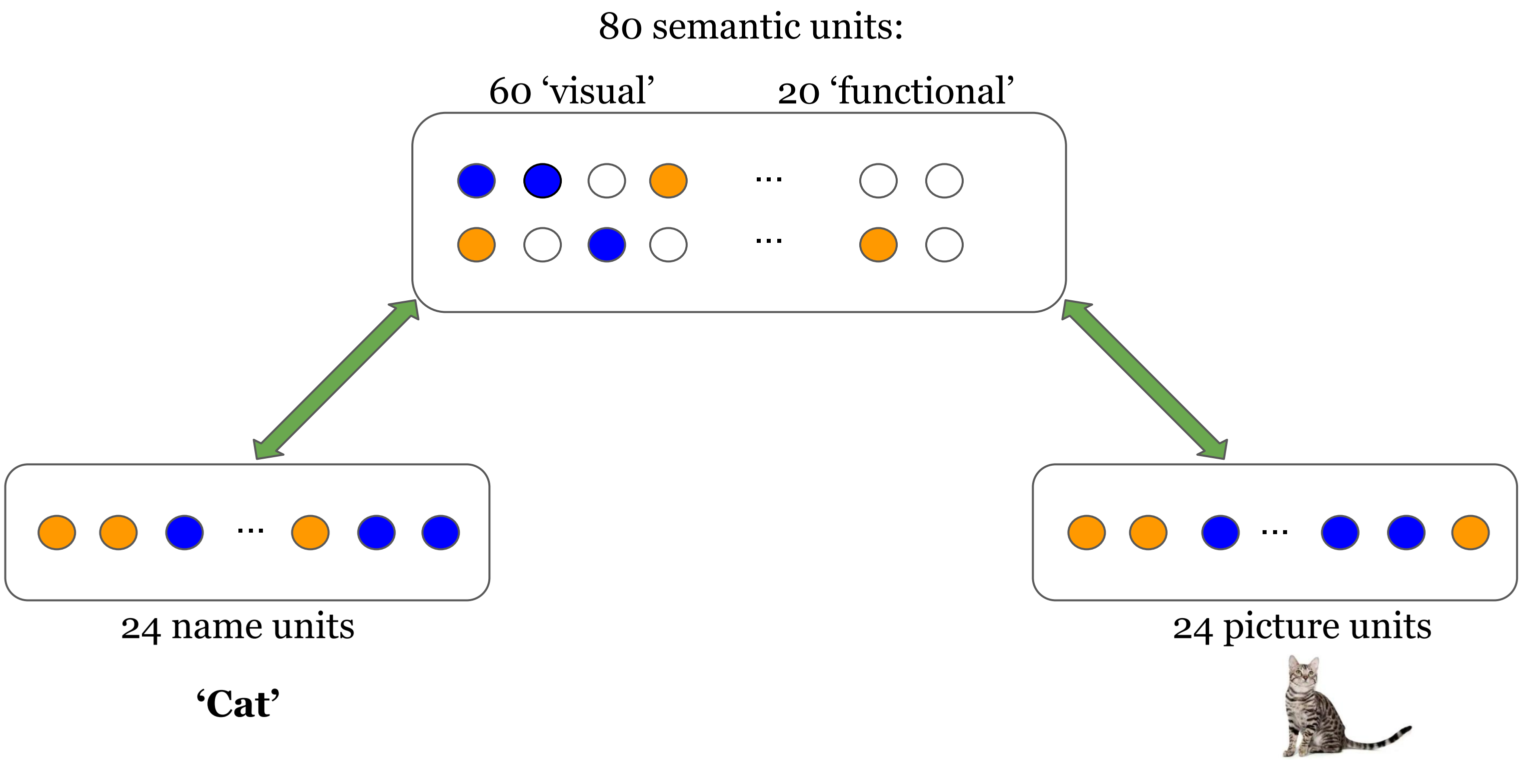
The Farah & McClelland model



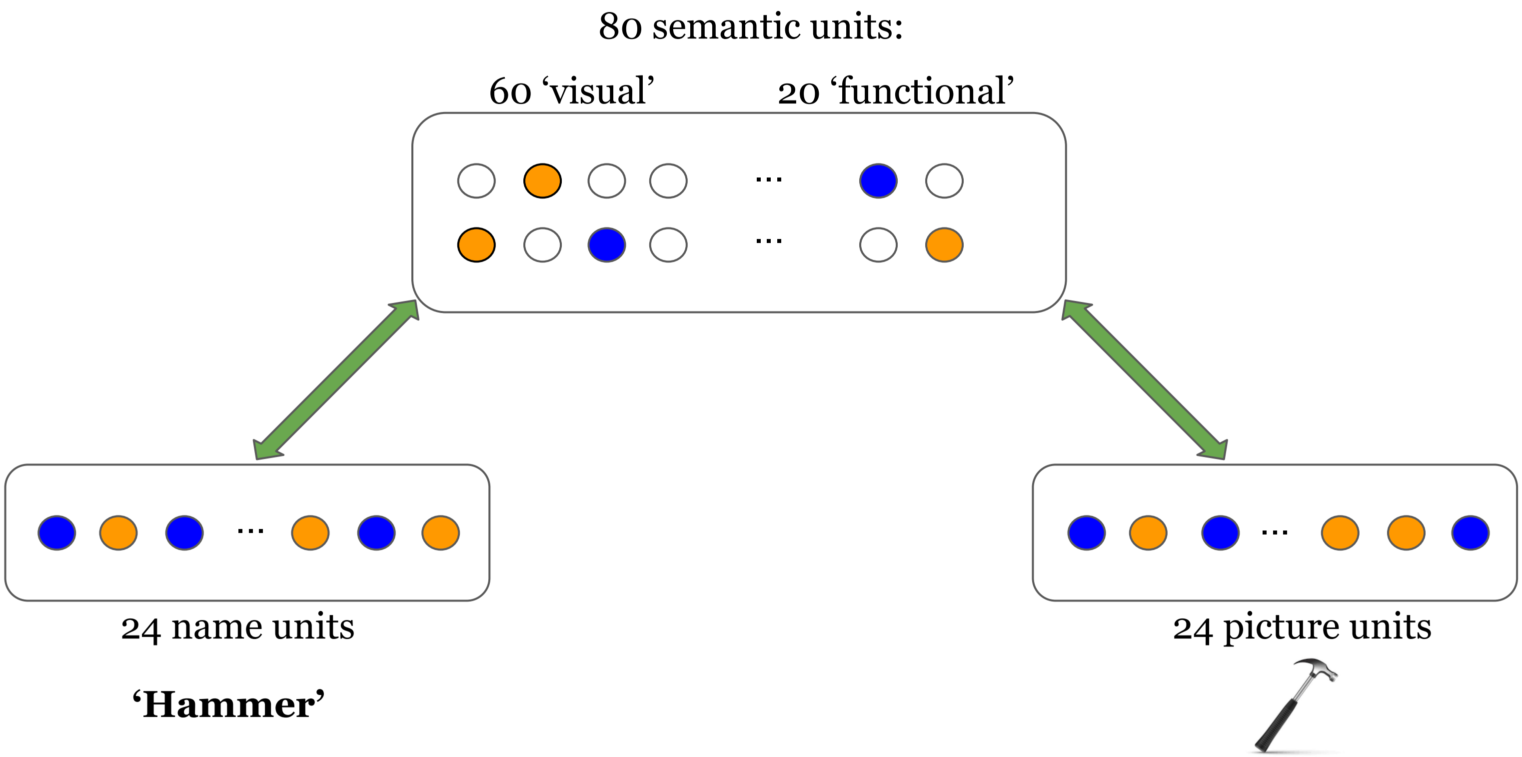
The Farah & McClelland model



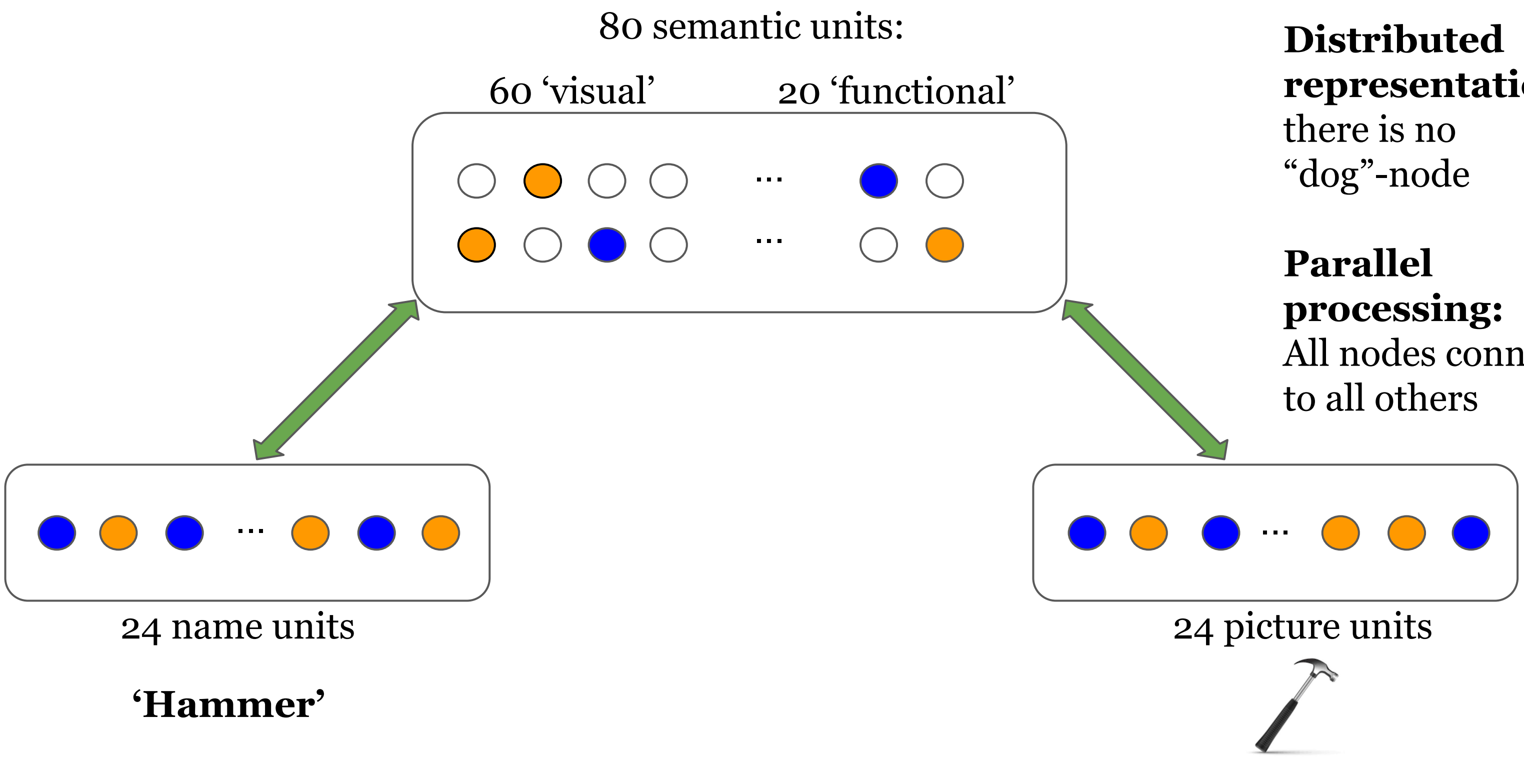
The Farah & McClelland model



The Farah & McClelland model



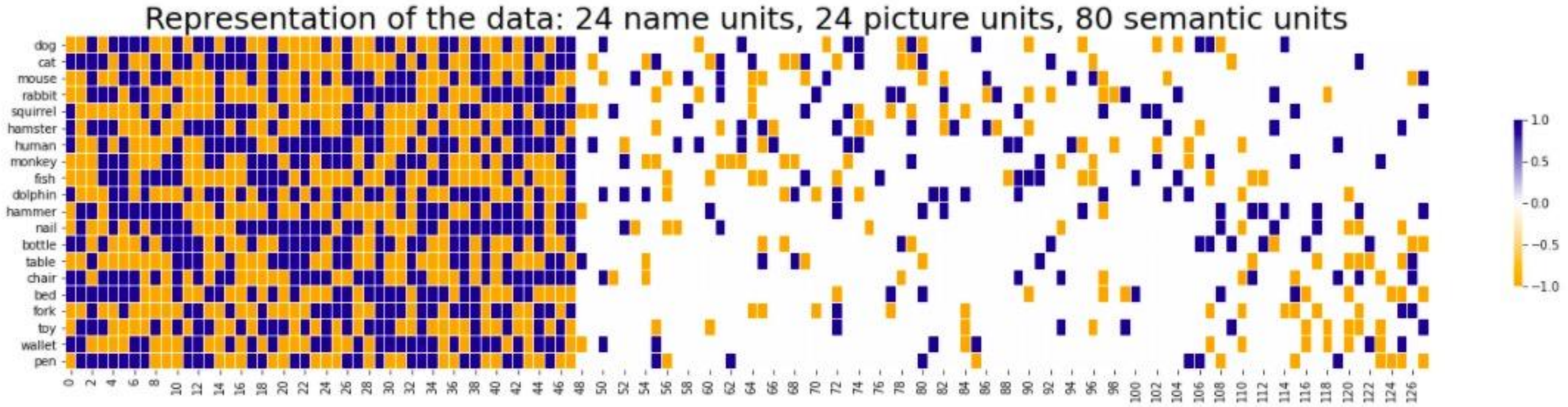
The Farah & McClelland model



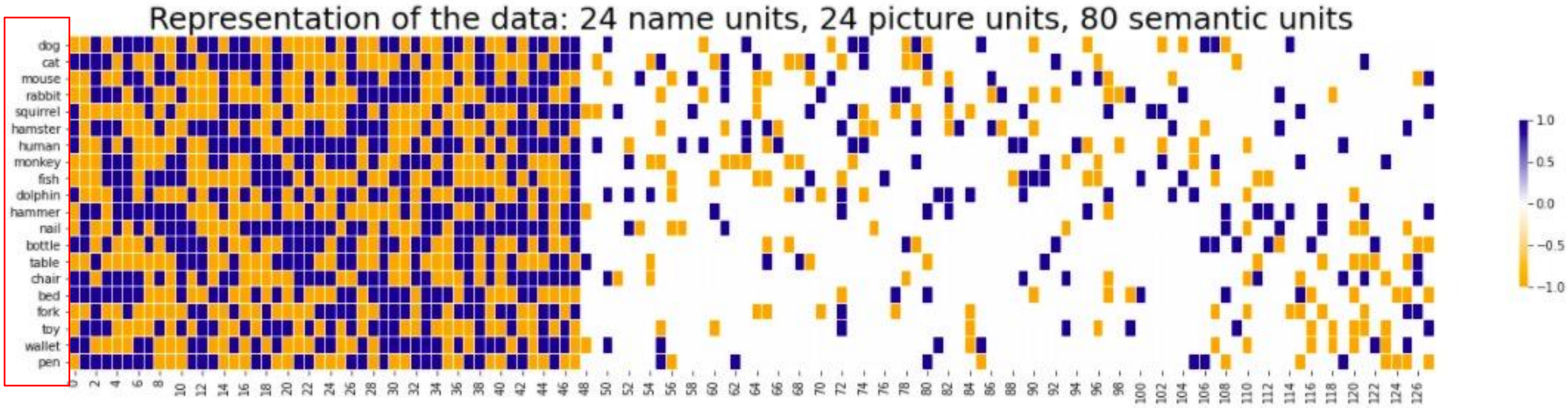
Distributed representations: there is no "dog"-node

Parallel processing: All nodes connected to all others

The training data

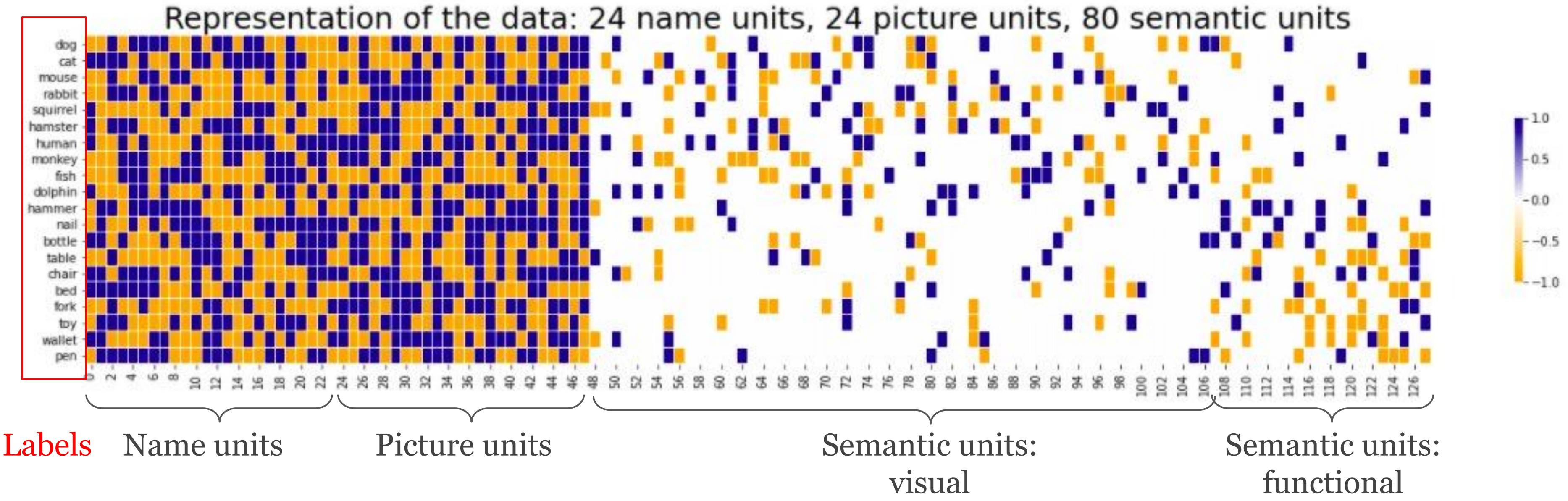


The training data

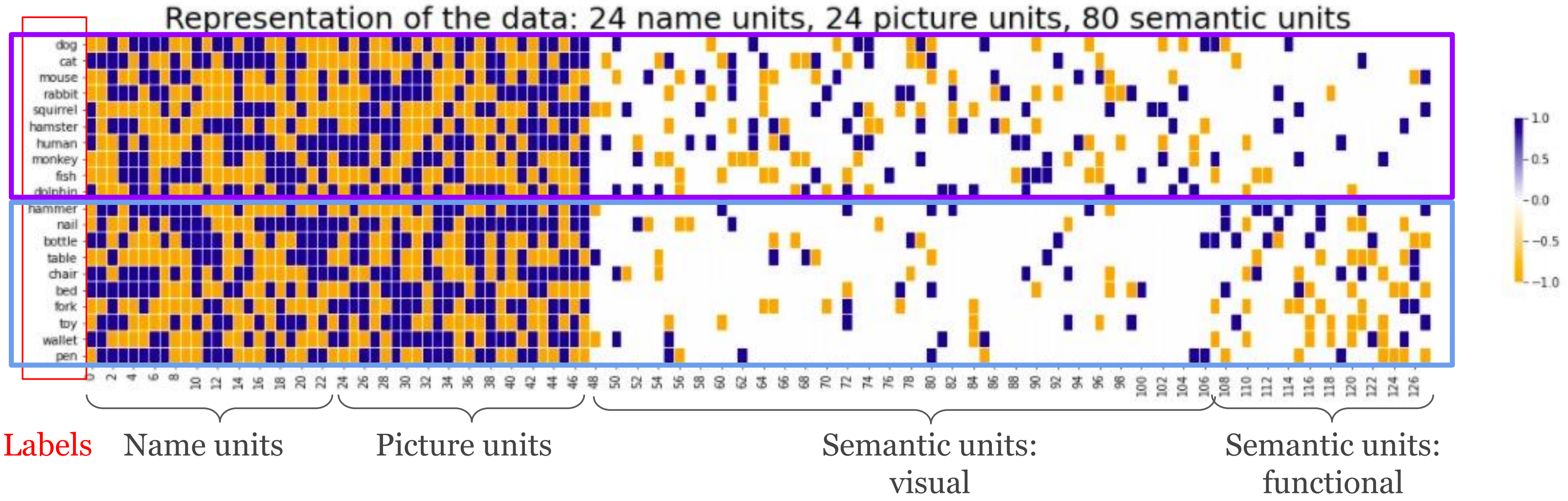


Labels

The training data

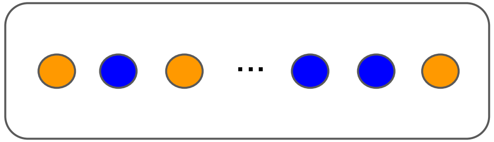


The training data



Training the model (I): name-2-picture task

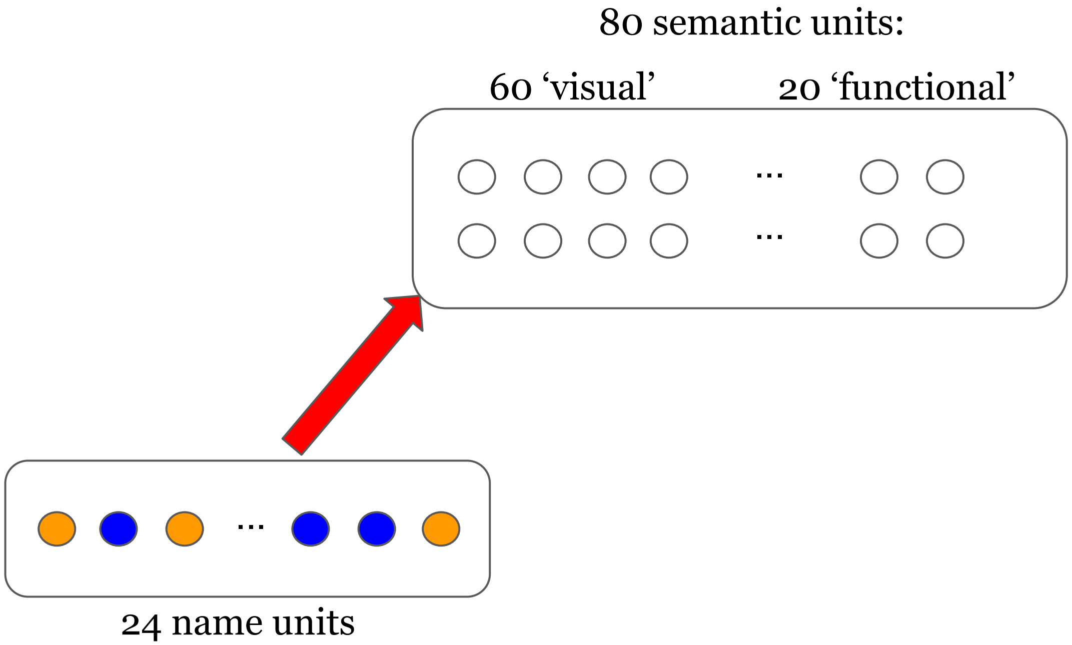
Training the model (I): name-2-picture task



24 name units

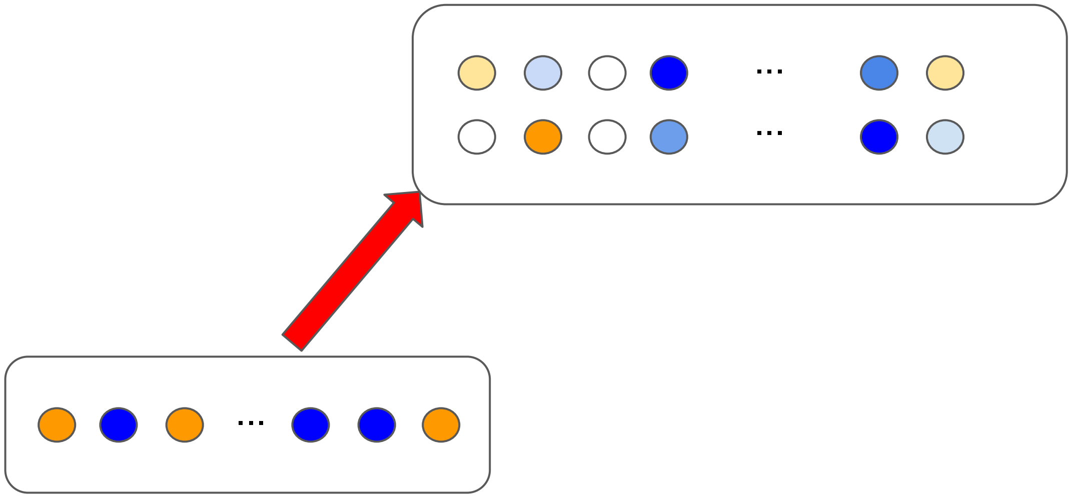
Input: ‘Dog’

Training the model (I): name-2-picture task

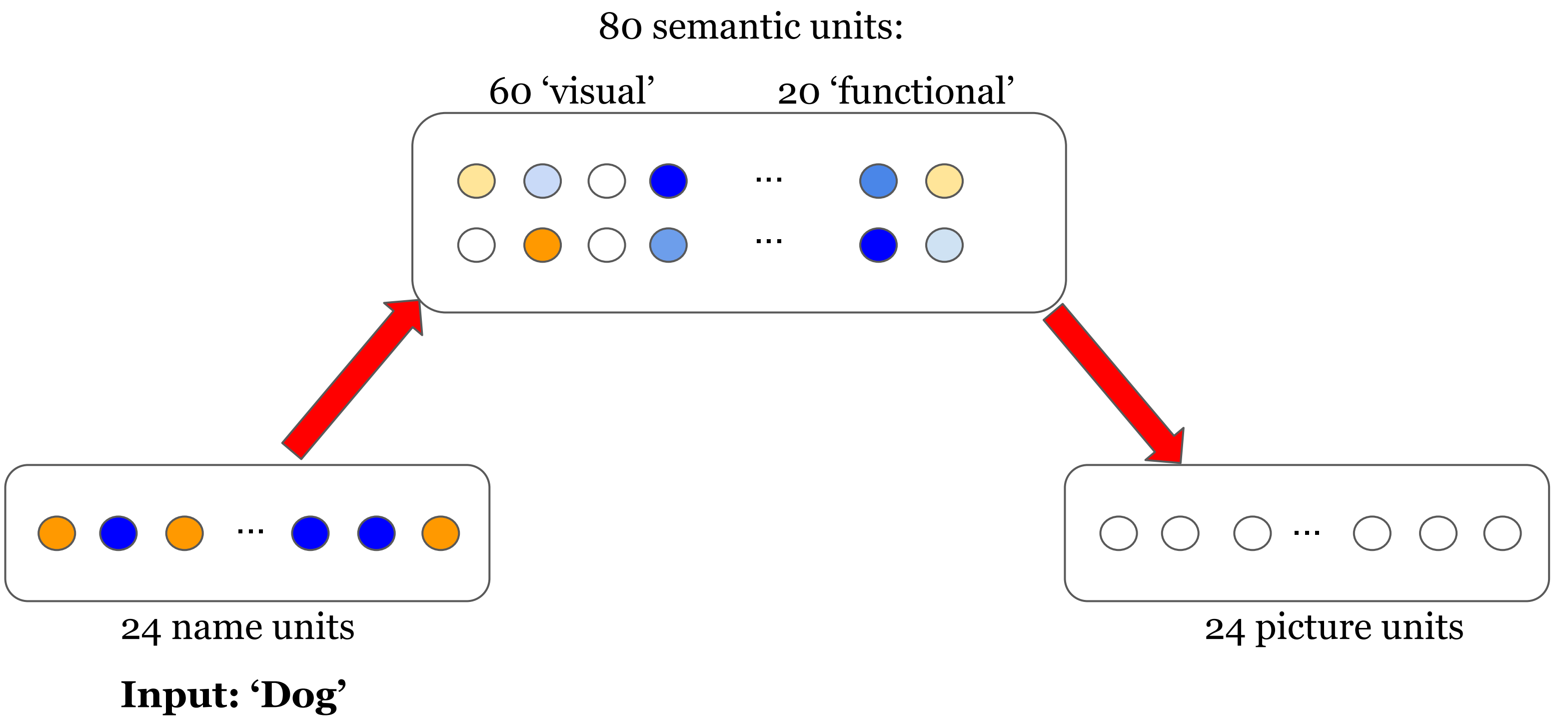


Input: 'Dog'

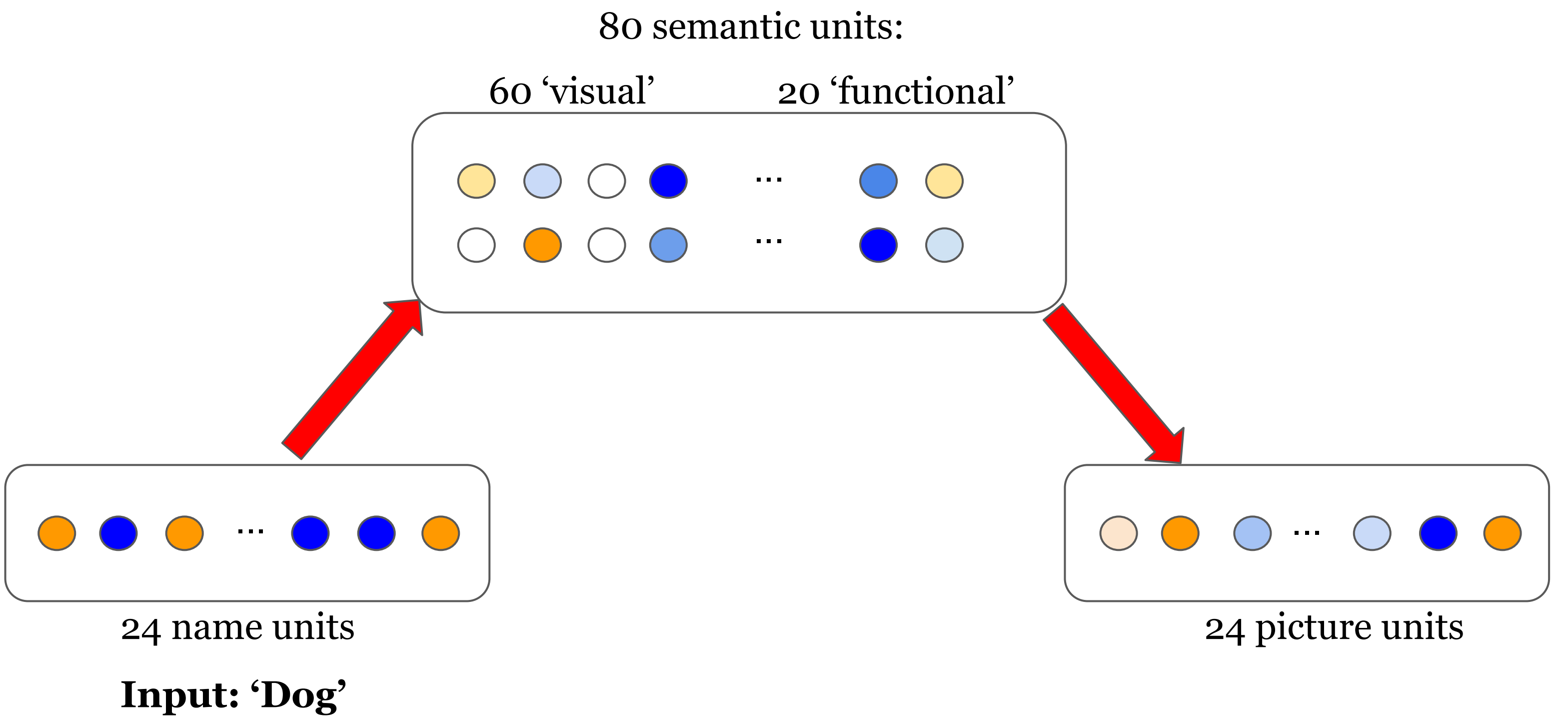
Training the model (I): name-2-picture task



Training the model (I): name-2-picture task



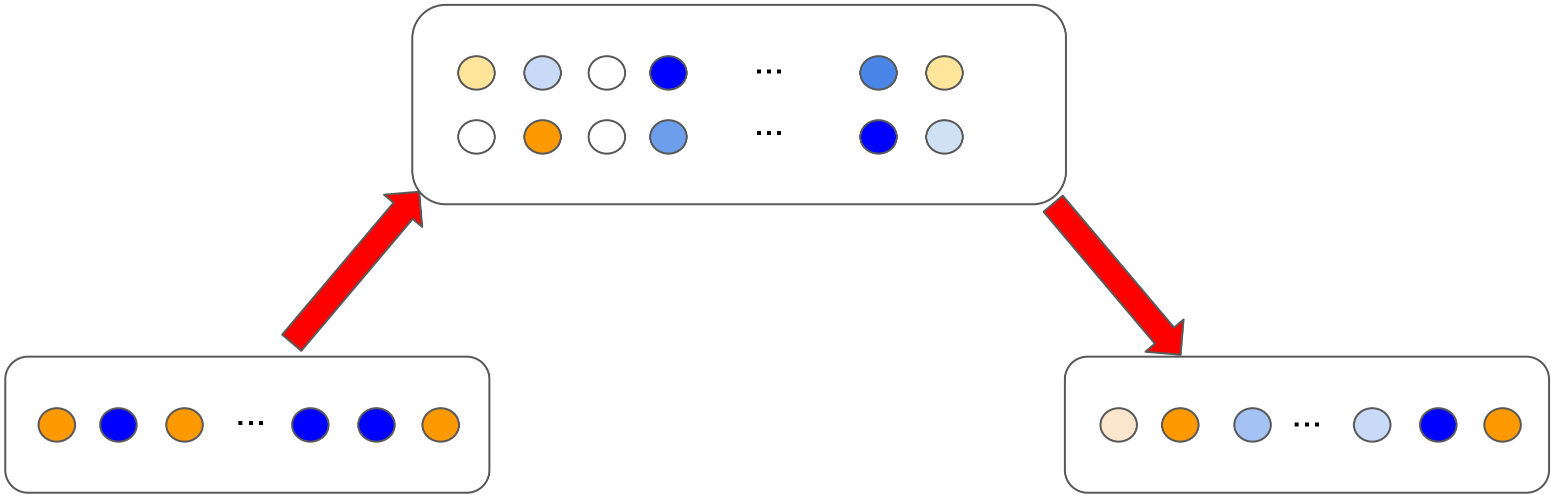
Training the model (I): name-2-picture task



Training the model (I): name-2-picture task

80 semantic units:

60 ‘visual’ 20 ‘functional’



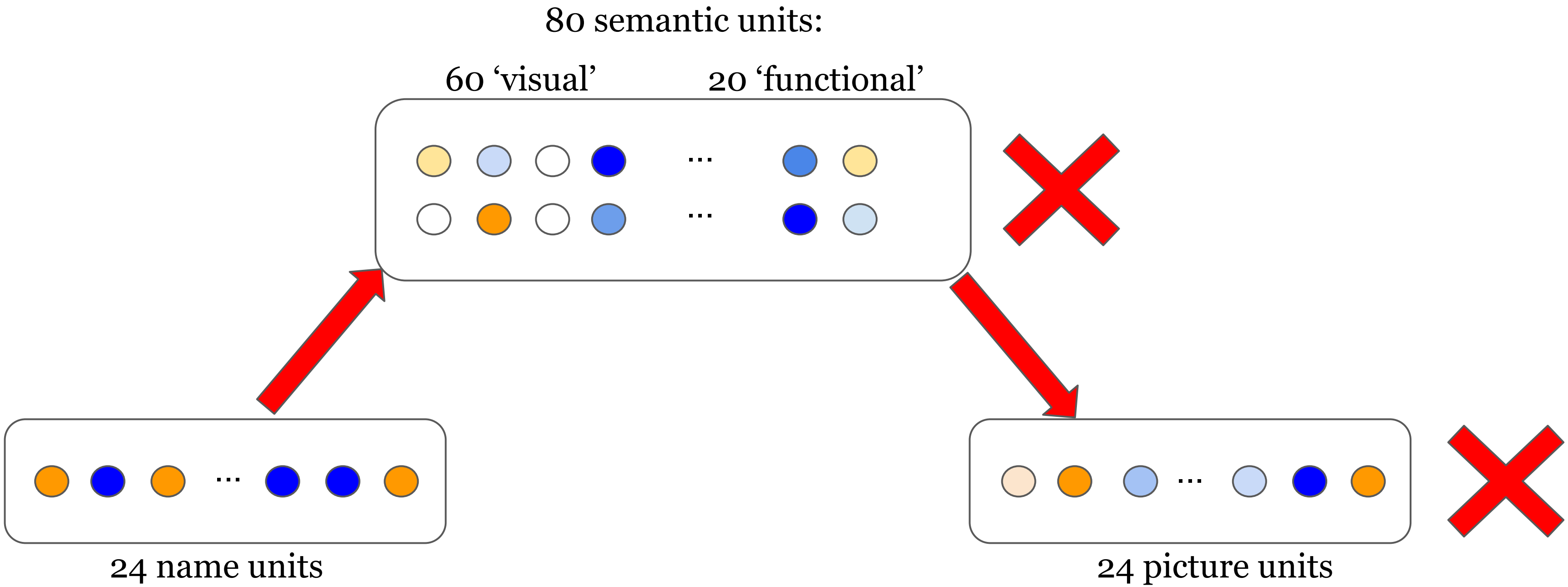
24 name units

24 picture units

Input: ‘Dog’

Output: it must be a cat!

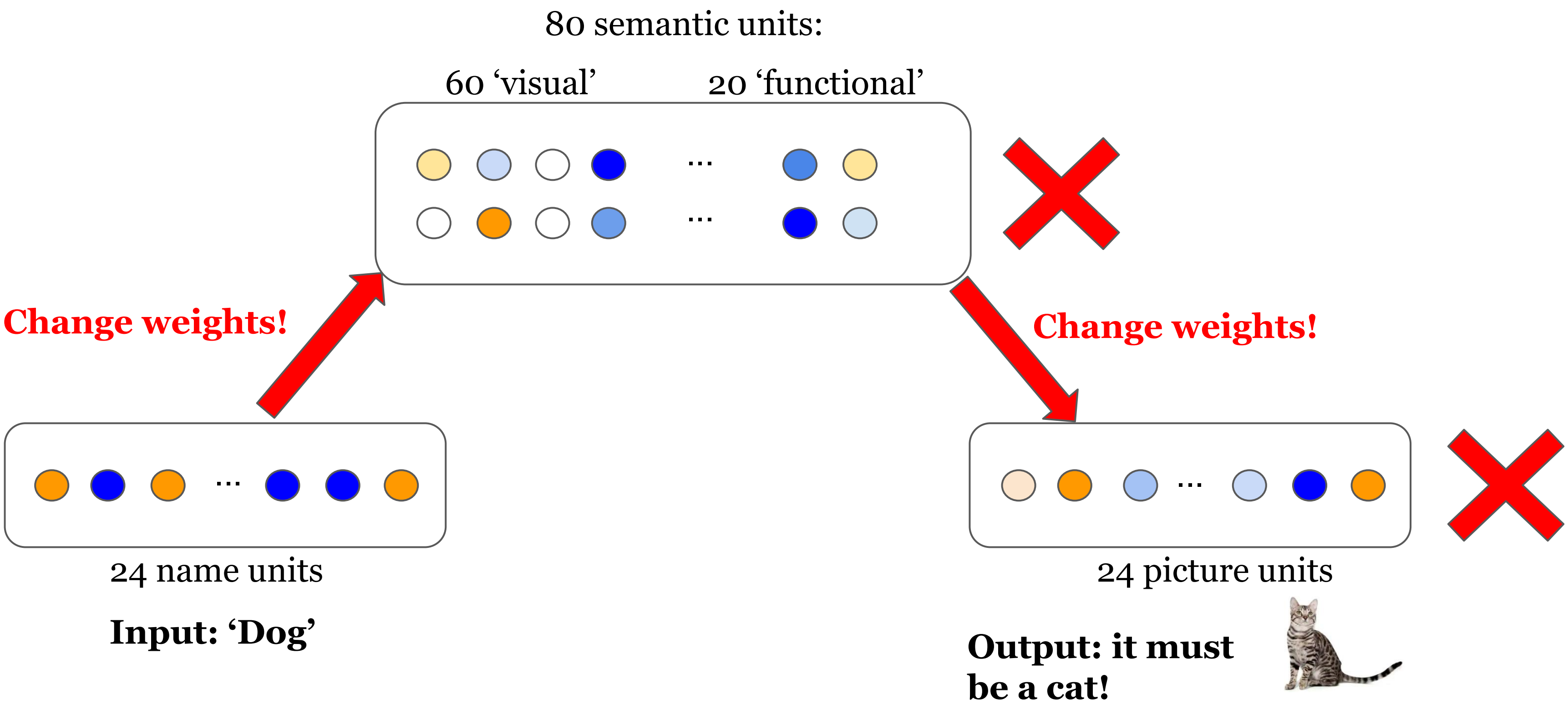
Training the model (I): name-2-picture task



Input: ‘Dog’

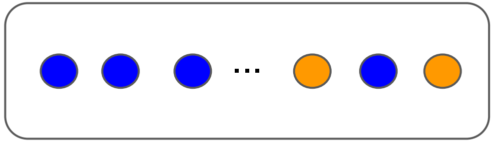
Output: it must be a cat!

Training the model (I): name-2-picture task



Training the model (II): picture-naming task

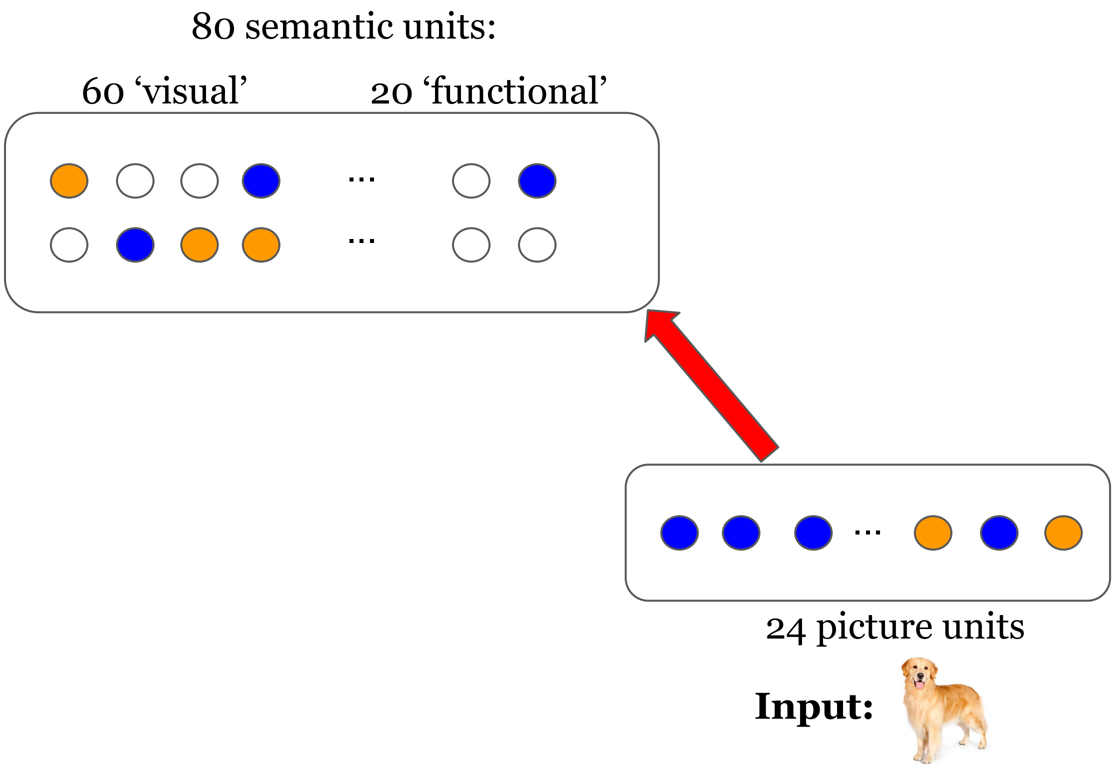
Training the model (II): picture-naming task



24 picture units

Input: 

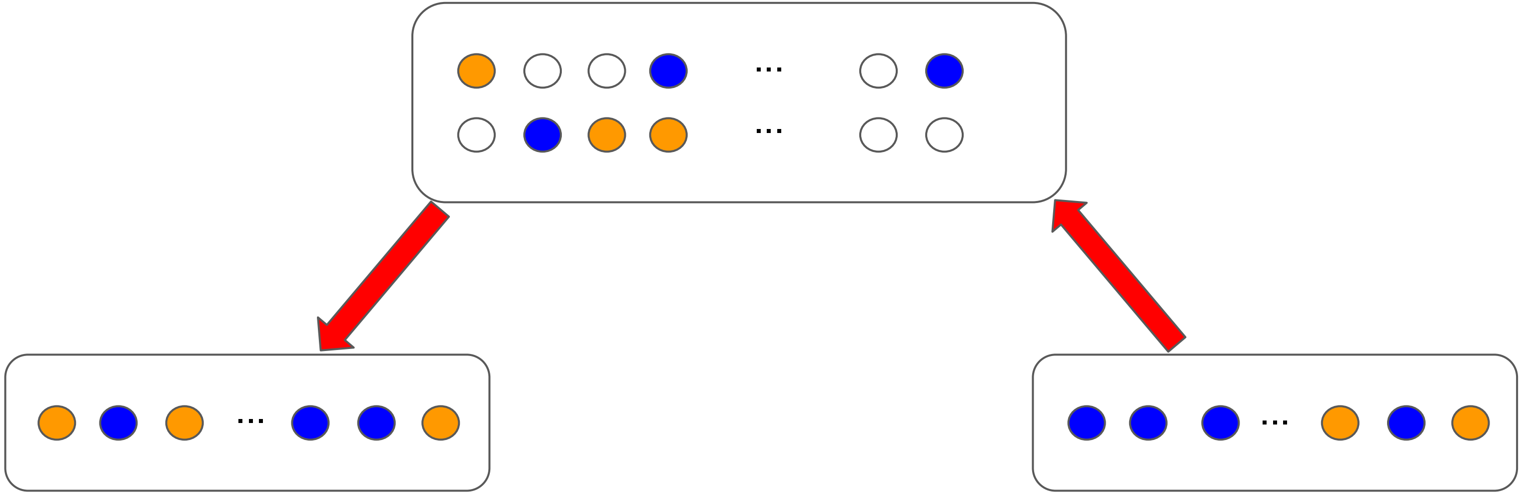
Training the model (II): picture-naming task



Training the model (II): picture-naming task

80 semantic units:

60 ‘visual’ 20 ‘functional’



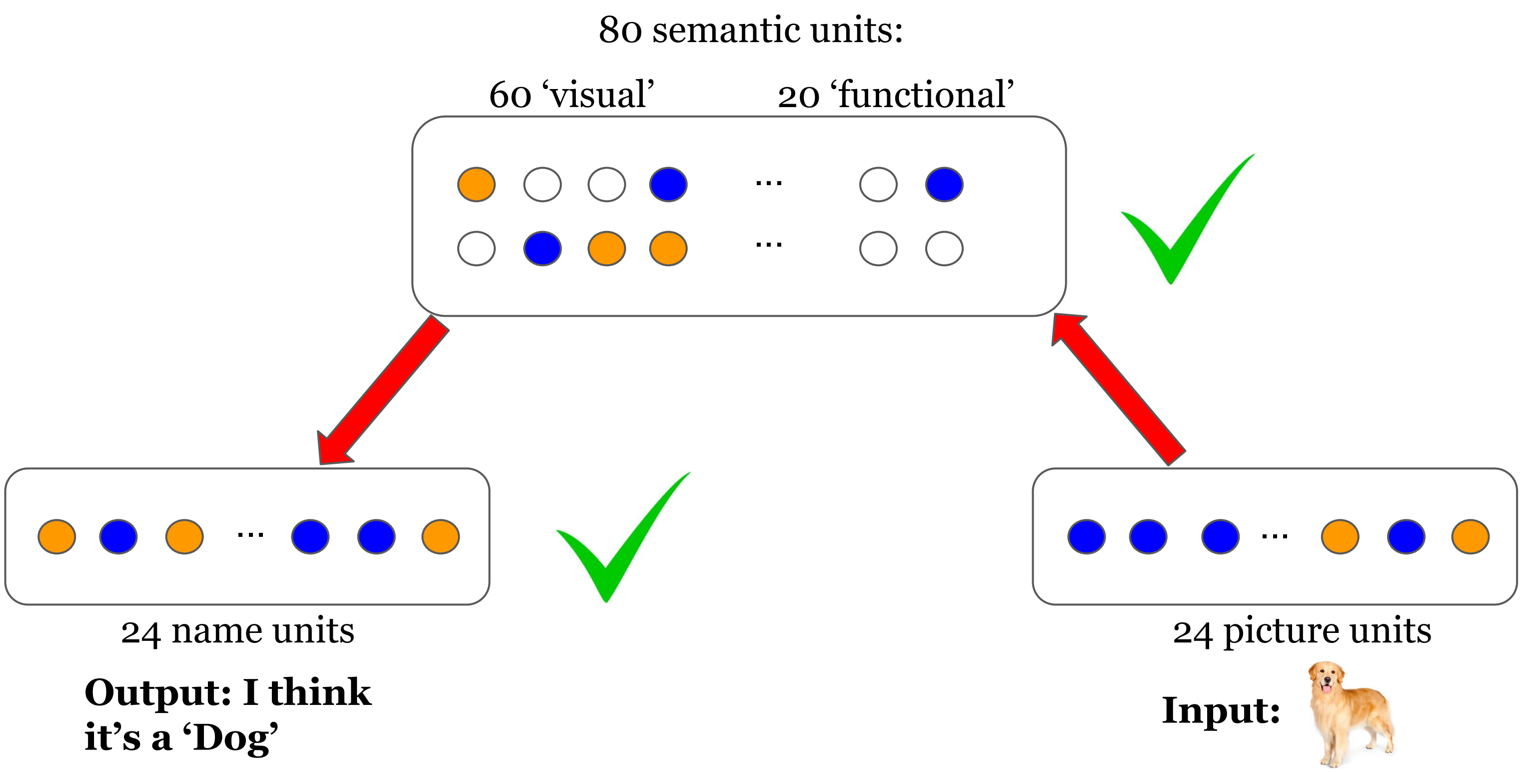
24 name units

24 picture units

Output: I think it’s a ‘Dog’

Input:

Training the model (II): picture-naming task



Training regime of Farah & McClelland

- 20 patterns: dog, cat, ...

Training regime of Farah & McClelland

- 20 patterns: dog, cat, ...
- Present each pattern twice: once for name-2-picture task, and once for picture-naming task → 2x20 = 40 **'trials'**

Training regime of Farah & McClelland

- 20 patterns: dog, cat, ...
- Present each pattern twice: once for name-2-picture task, and once for picture-naming task → 2x20 = 40 **‘trials’**
- Adjust the weights after each trial → ‘learning’!

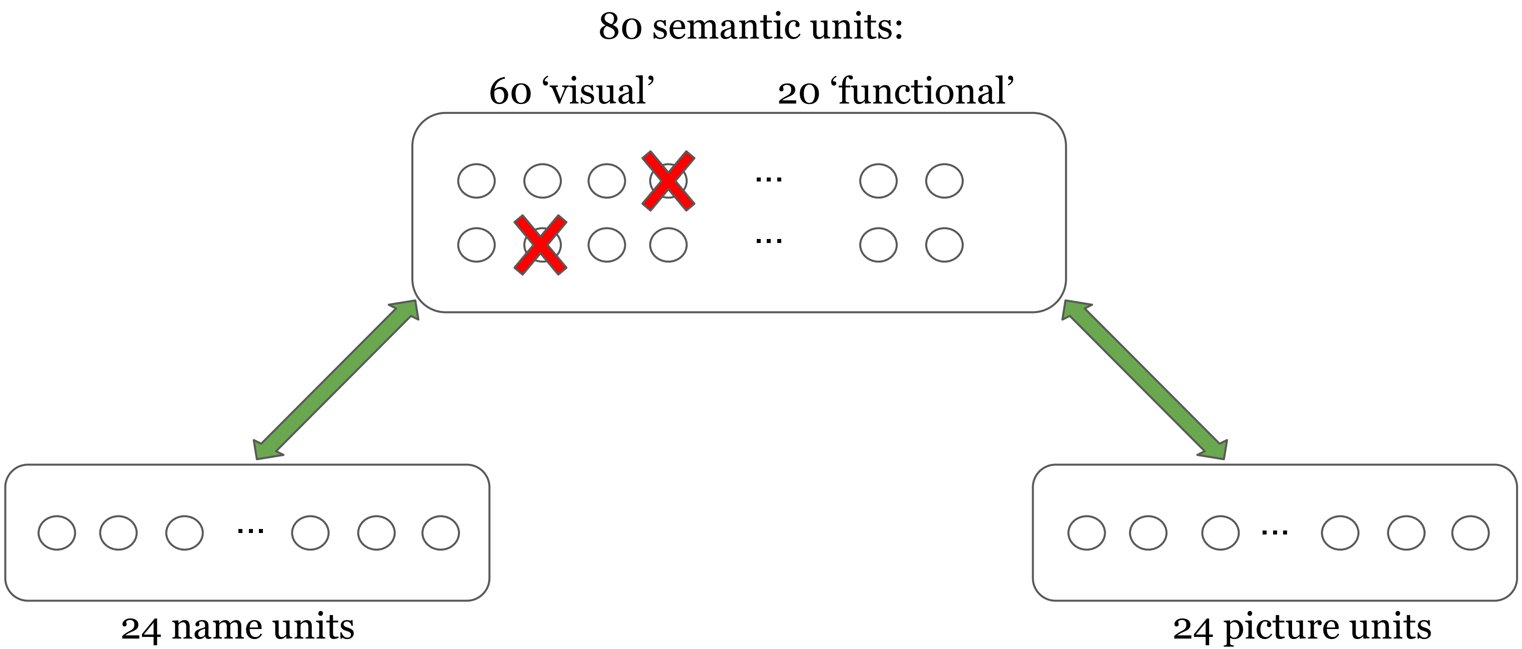
Training regime of Farah & McClelland

- 20 patterns: dog, cat, ...
- Present each pattern twice: once for name-2-picture task, and once for picture-naming task → 2x20 = 40 **‘trials’**
- Adjust the weights after each trial → ‘learning’!
- Repeat all this 100x (= 100 **‘epochs’** of training)

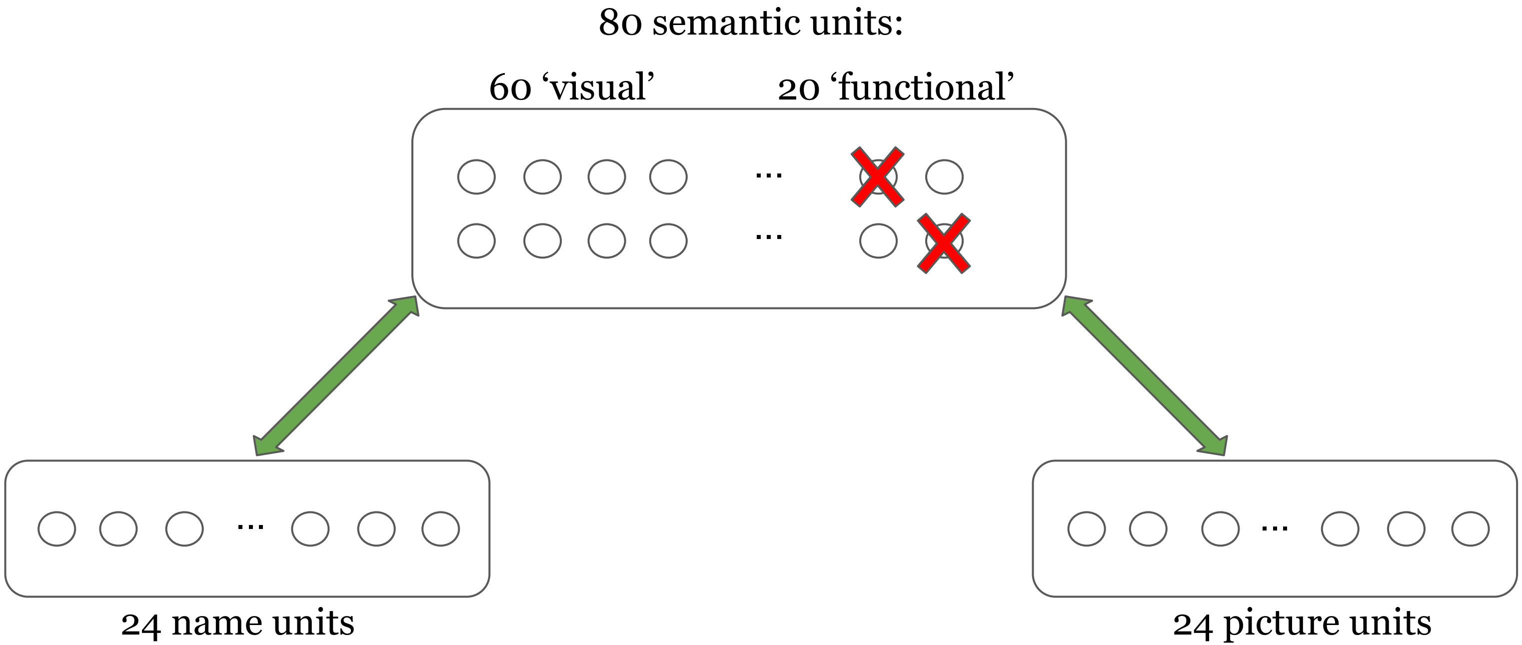
Training regime of Farah & McClelland

- 20 patterns: dog, cat, ...
- Present each pattern twice: once for name-2-picture task, and once for picture-naming task → 2x20 = 40 **'trials'**
- Adjust the weights after each trial → 'learning'!
- Repeat all this 100x (= 100 **'epochs'** of training)
- The final, trained model has perfect accuracy in both tasks!

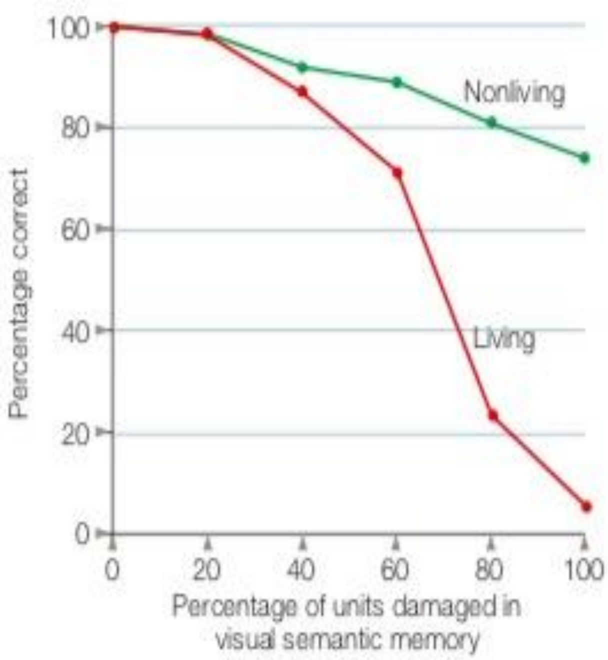
Lesioning the model (I): damage visual memory



Lesioning the model (II): damage funct. memory

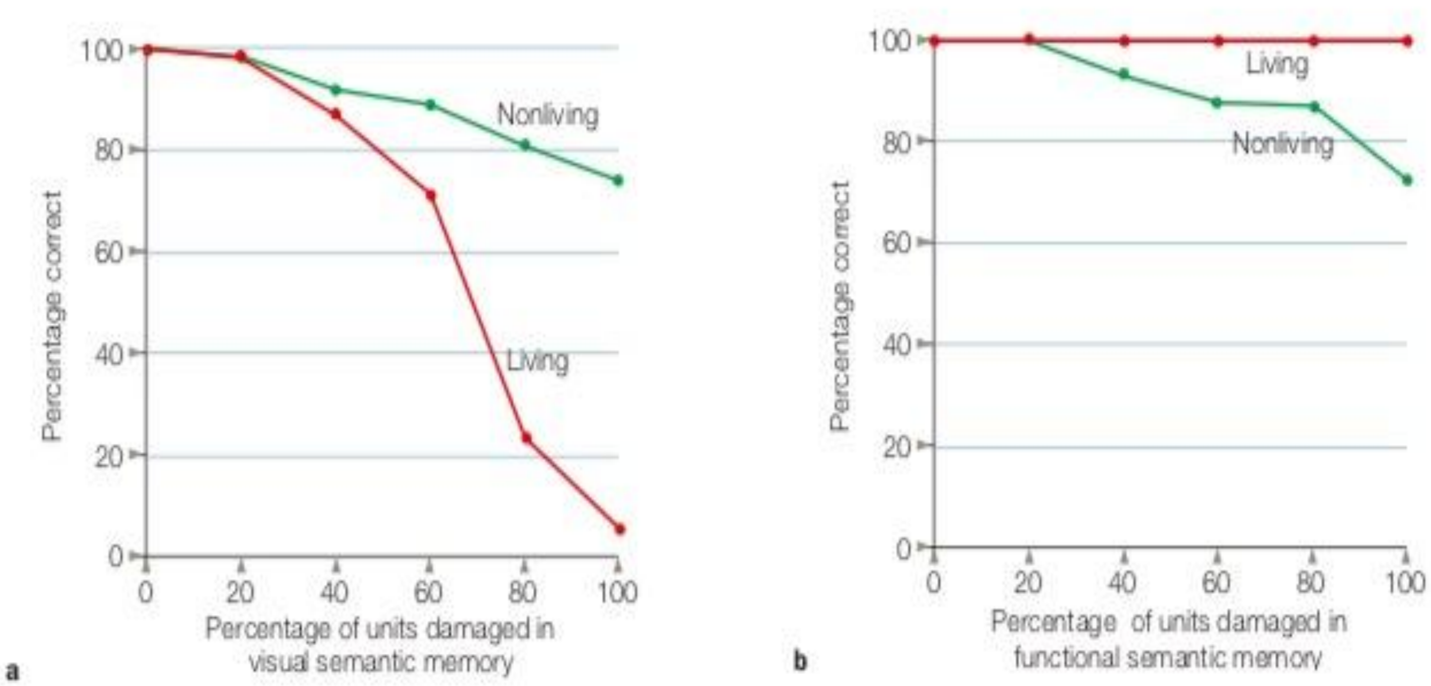


Farah & McClelland lesioning results

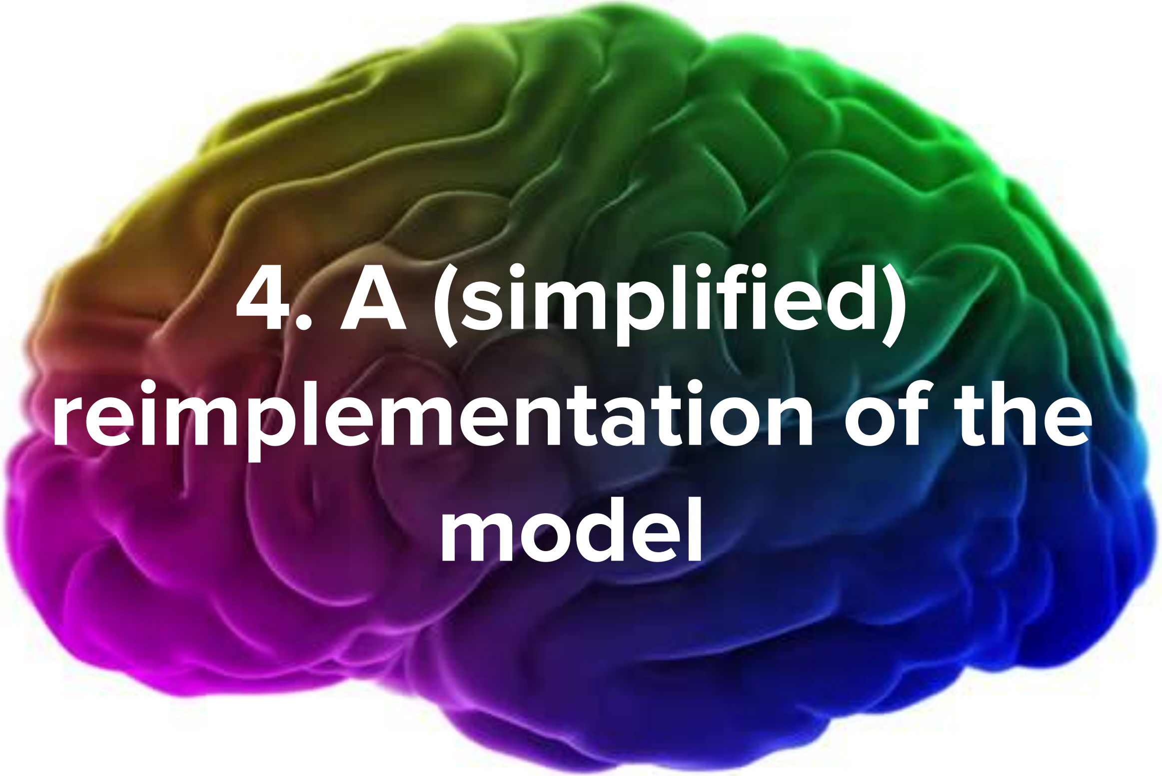


a

Farah & McClelland lesioning results

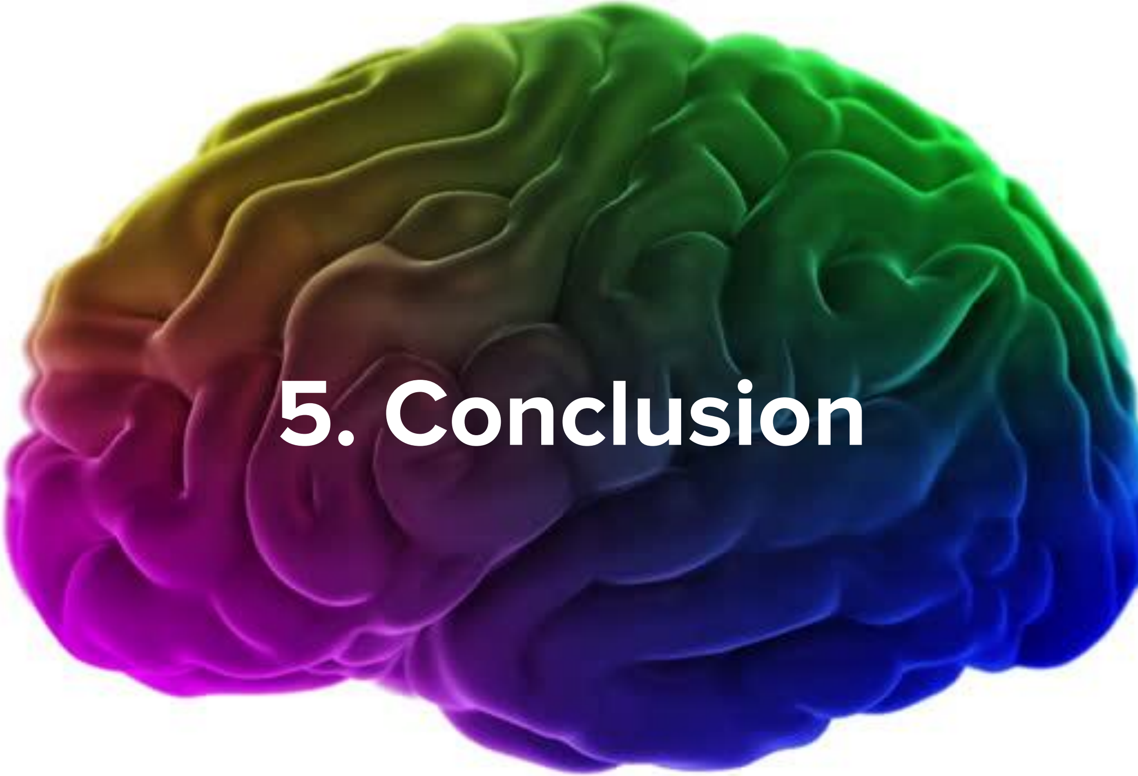


4. A (simplified) reimplementation of the model



https://github.com/miriamschulz/KNP_project

5. Conclusion



Discussion / Conclusions

➔ Complex behavior can be modeled with a relatively simple connectionist model

Discussion / Conclusions

- ➔ Complex behavior can be modeled with a relatively simple connectionist model
- ➔ An even more simplified reimplementation of the model can still reproduce part of the data

Discussion / Conclusions

- ➔ Complex behavior can be modeled with a relatively simple connectionist model
- ➔ An even more simplified reimplementation of the model can still reproduce part of the data
- ➔ Some problems for connectionist modeling:

Discussion / Conclusions

- ➔ Complex behavior can be modeled with a relatively simple connectionist model
- ➔ An even more simplified reimplementation of the model can still reproduce part of the data
- ➔ Some problems for connectionist modeling:
    - ◆ Results dependent on the **statistical structure in the training data**

Discussion / Conclusions

- ➔ Complex behavior can be modeled with a relatively simple connectionist model
- ➔ An even more simplified reimplementation of the model can still reproduce part of the data
- ➔ Some problems for connectionist modeling:
  - Results dependent on the **statistical structure in the training data**
  - **Hyperparameters**: just twitch the hyperparameters until we get the desired results!

Discussion / Conclusions

- ➔ Complex behavior can be modeled with a relatively simple connectionist model
- ➔ An even more simplified reimplementation of the model can still reproduce part of the data
- ➔ Some problems for connectionist modeling:
  - Results dependent on the **statistical structure in the training data**
  - **Hyperparameters**: just twitch the hyperparameters until we get the desired results!
  - How **cognitively plausible** are such models really?

Discussion / Conclusions

- ➔ Complex behavior can be modeled with a relatively simple connectionist model
- ➔ An even more simplified reimplementation of the model can still reproduce part of the data
- ➔ Some problems for connectionist modeling:
  - ◆ Results dependent on the **statistical structure in the training data**
  - ◆ **Hyperparameters**: just twitch the hyperparameters until we get the desired results!
  - ◆ How **cognitively plausible** are such models really?
- ➔ Implications: use such models as **proof-of-theory** and **complement** for neuropsychological theories?

Discussion / Conclusions

➔ **Symbolic** vs. **distributed** representations

- Distributed representations can go a long way in modeling double dissociation data

Discussion / Conclusions

→ **Symbolic** vs. **distributed** representations

- Distributed representations can go a long way in modeling double dissociation data
- No need for a separate semantic memory system for living vs. non-living things

Discussion / Conclusions

➔ **Symbolic** vs. **distributed** representations

- Distributed representations can go a long way in modeling double dissociation data
- No need for a separate semantic memory system for living vs. non-living things
- But: recent evidence of specific cells in large deep neural networks that reply to very specific concepts! (see e.g. Bowers, 2017)

References

Bowers, J. S. (2017). Parallel distributed processing theory in the age of deep networks. *Trends in cognitive sciences, 21*(12), 950-961.

Farah, M. J., & McClelland, J. L. (1991). A computational model of semantic memory impairment: Modality specificity and emergent category specificity. *Journal of Experimental Psychology: General, 120*(4), 339–357.

Hebb, D. O. (2005). *The organization of behavior: A neuropsychological theory*. Psychology Press. Chicago.

McCulloch, W.S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*(5), 115–133.

Minsky, M. L., & Papert, S. A. (1988). *Perceptrons: expanded edition*. Chicago.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review, 65*(6), 386-408. Chicago.

Questions & Discussion