Psychology of Language

12 Computational models II

Fall 2023
Tues/Thur 5:00-6:15pm

Emma Wing
Drop-in hours:
Wednesdays 3-4pm
& by appointment
Webex link

Thanks to Wesley Leong for content and slide inspiration!

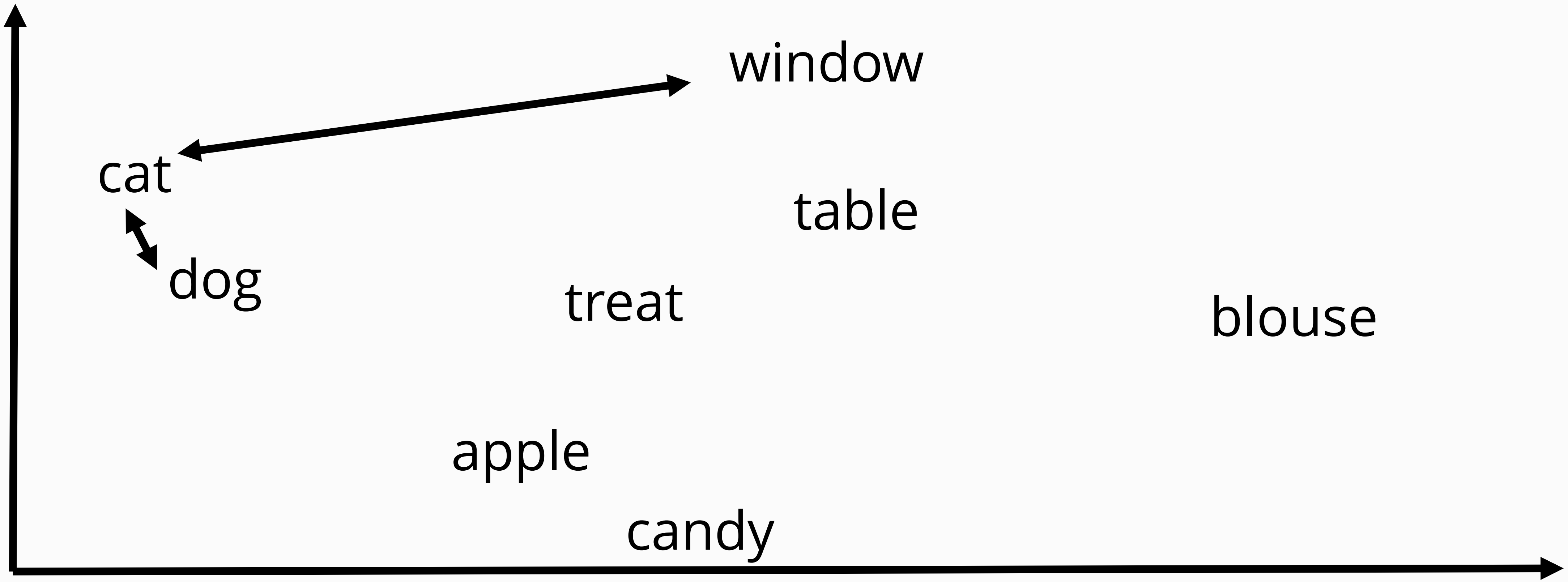
Road map

- Review from **11 Computational models I**
- Unit 2: The Mature System

  12 Computational models II

Review: 11 Computational models I

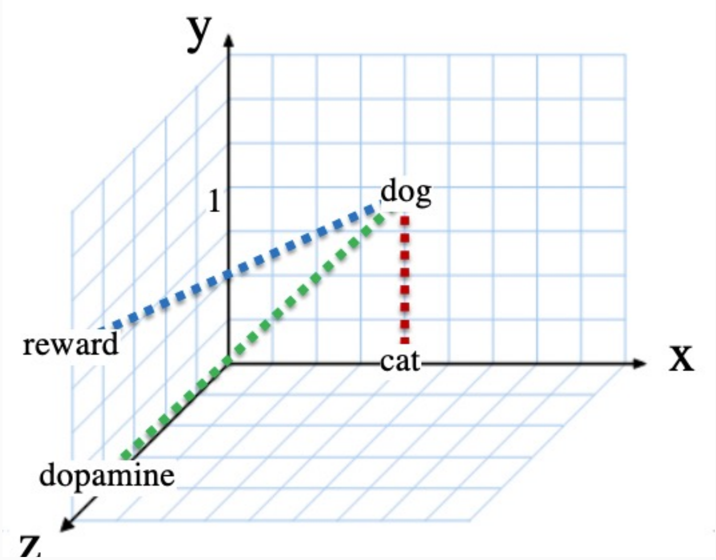
- **Semantic spaces** capture meaning as relationships between words (in terms of distance)



Review: 11 Computational models I

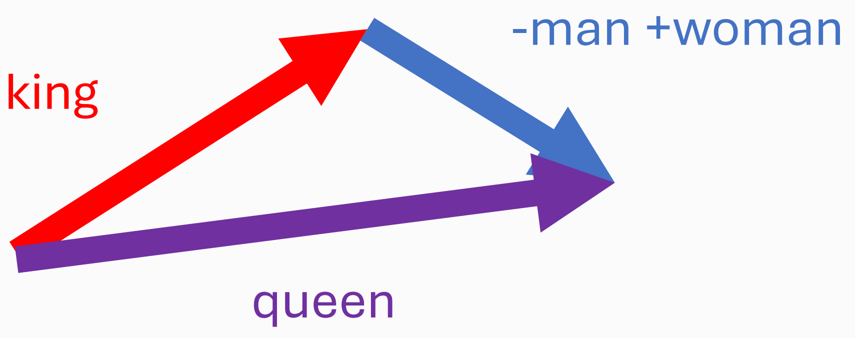
- Model 1: **Latent semantic analysis** - Words that co-occur in bodies of text (global contexts) are often more similar to one another than words that do not.

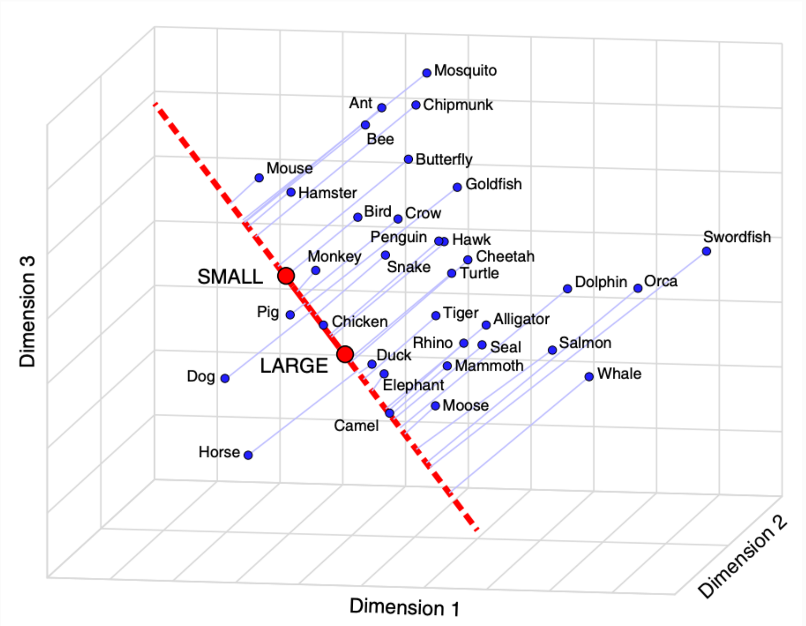
Context (Document)				
	#1	#2	#3	coordinate
CAT	1	0	0	1, 0, 0
DOPAMINE	0	0	1	0, 0, 1
DOG	1	1	0	1, 1, 0
REWARD	0	1	1	0, 1, 1



Review: 11 Computational models I

- Model 2: **Word2Vec** - predicts neighboring words (local contexts); some features map well to human intuitions





Review: 11 Computational models I

- ✓ Models are wrong but useful
- ✓ Meaning can be represented as a semantic space
  - ✓ Semantic spaces can be created using word vectors
- ✓ Context and co-occurrence matters for these models
- ✓ LSA uses global context
- ✓ Word2Vec uses local context
- ✓ Similarities and differences between computational models for word meanings and what humans do
  - ✓ Acquisition (and input!)
  - ✓ Organization
  - ✓ Use for prediction

Learning objectives

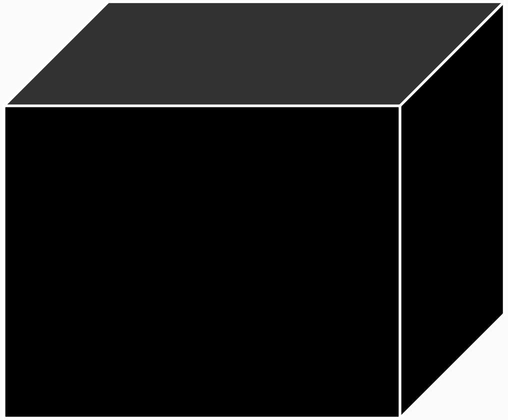
- Describe how a neural network works
- Define ‘black box’
- Describe backpropagation and how it helps the model improve
- Be able to link what the model does to potential insights about how the mind/brain works.

An analogy

People have compared neural networks to a black box

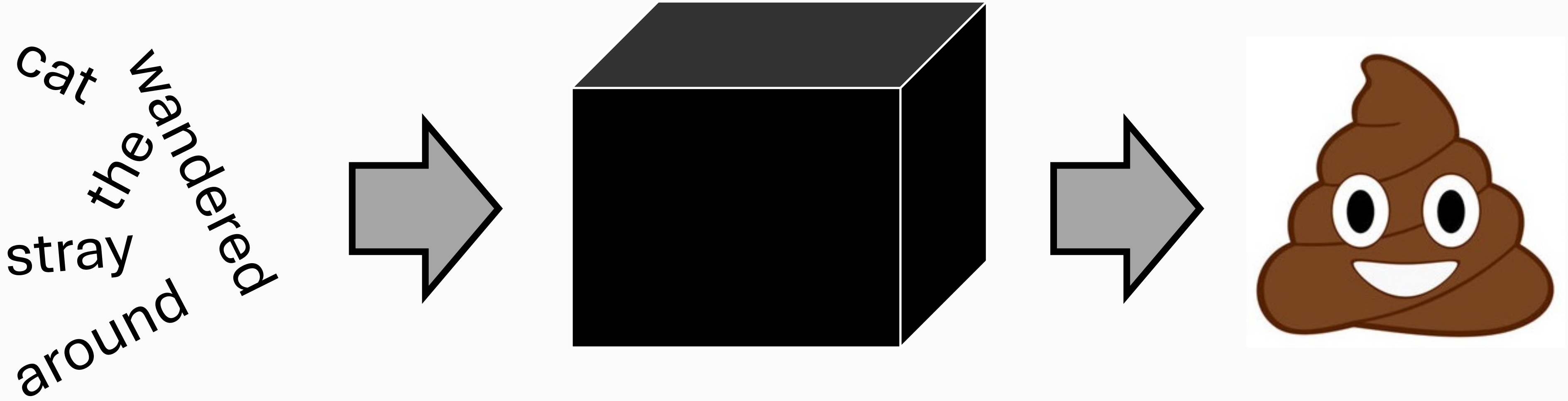
An analogy

People have compared neural networks to a black box



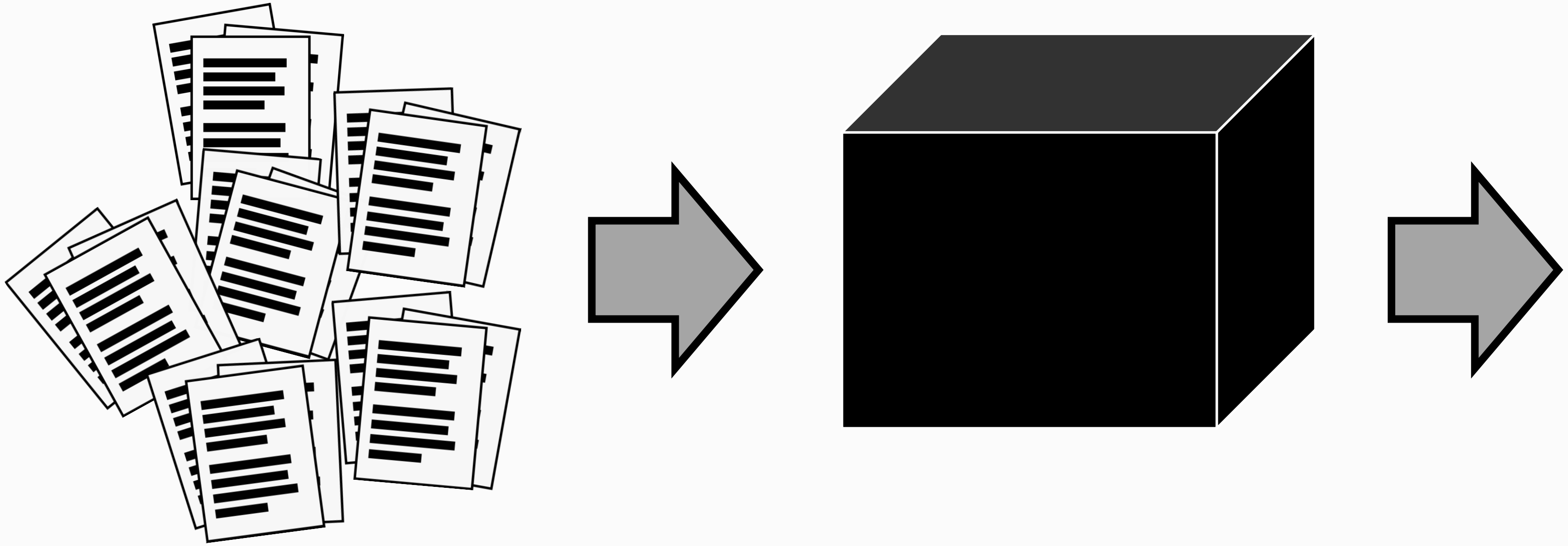
An analogy

At first its internal workings aren't organized, and it produces gibberish



An analogy

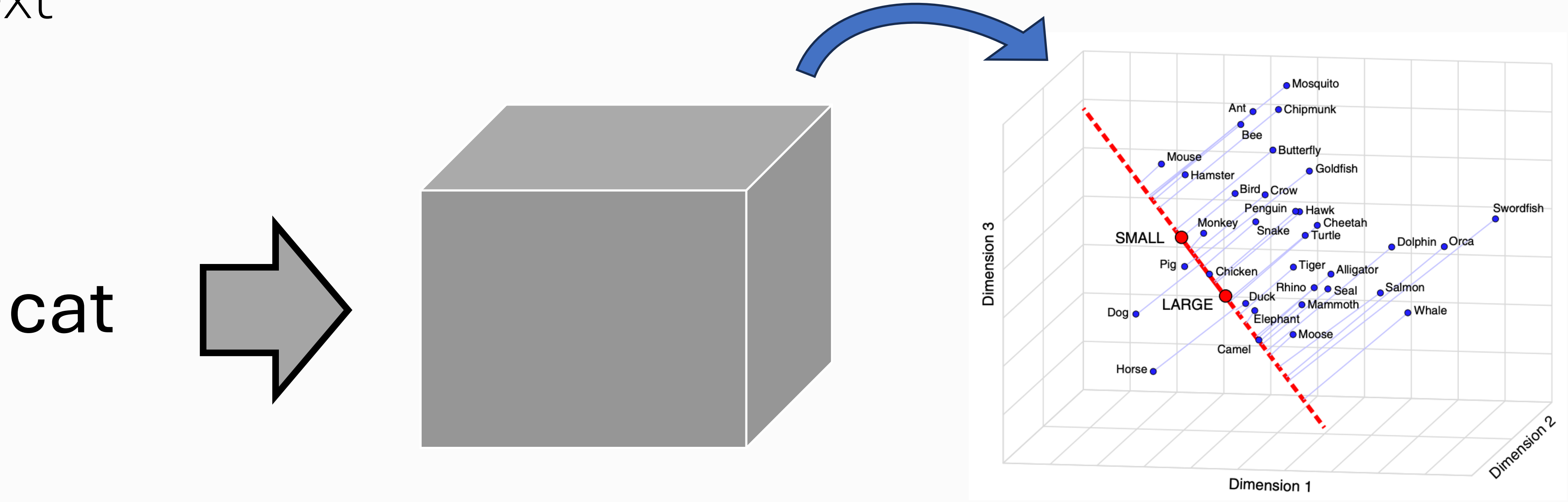
You give it lots of text, some magic happens, and suddenly it "does" language



The stray cat
wandered around
the neighborhood
looking for food

An analogy

Last class, we looked at the internal representations that models* like word2vec contain after they've been trained on lots of text

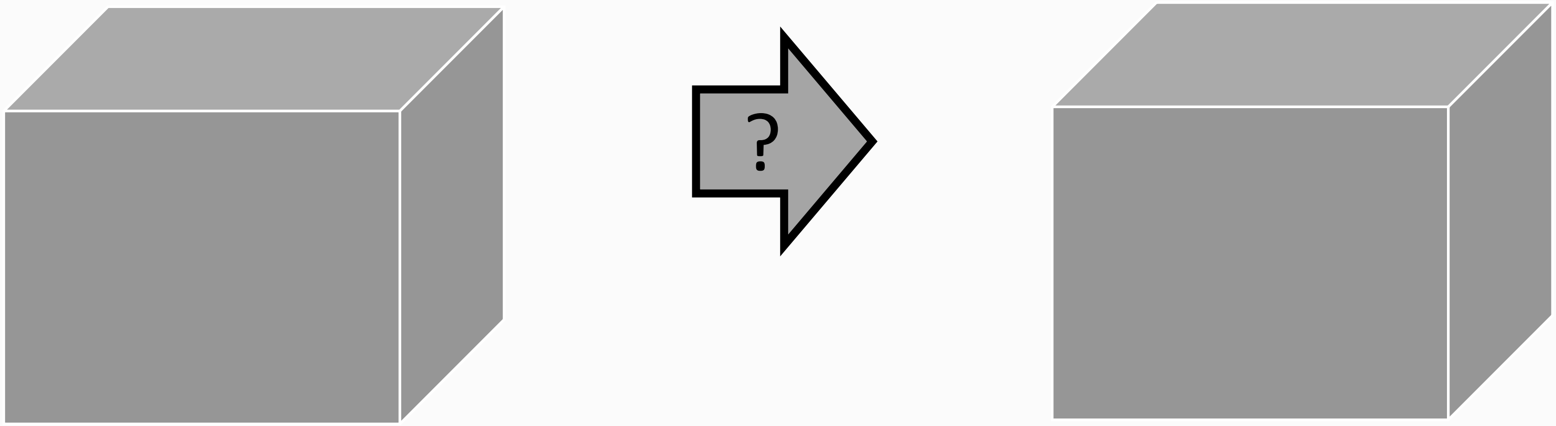


*note that LSA is not a black box in the same sense, since we know exactly what it's doing (counting word co-occurrences across documents)

An analogy

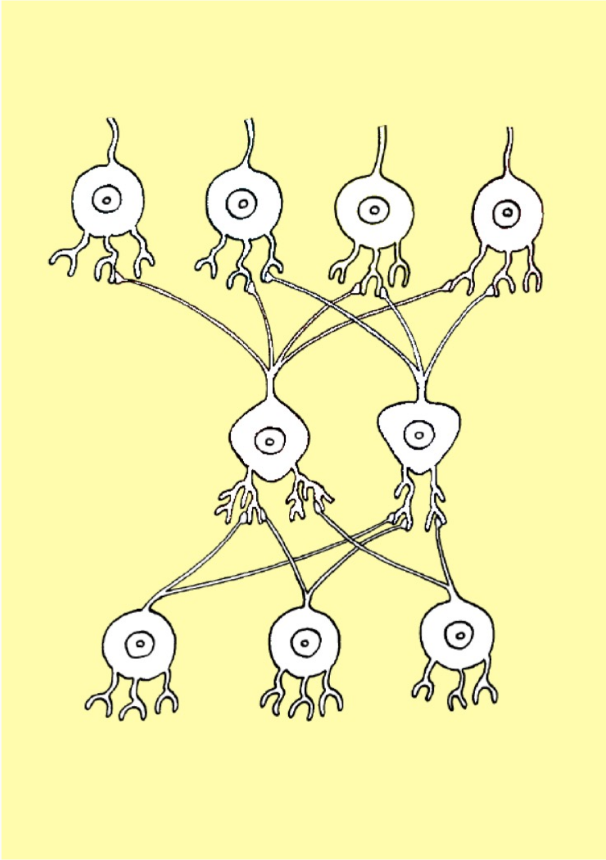
This class, we learn a little about how the models form this internal machinery

We'll also see how a very straightforward task and simple model architecture can lead to quite sophisticated internal representations

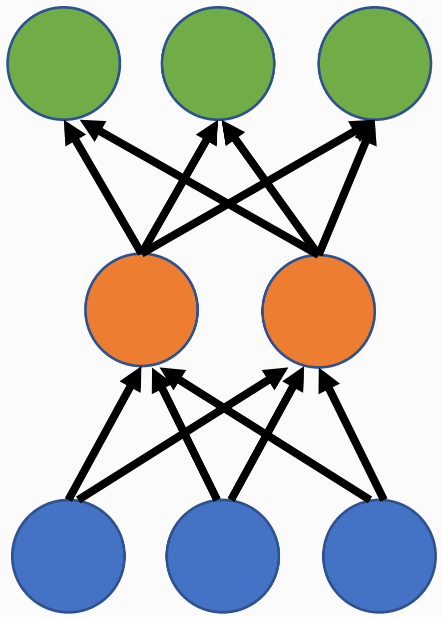


Neural networks

Biological neural network



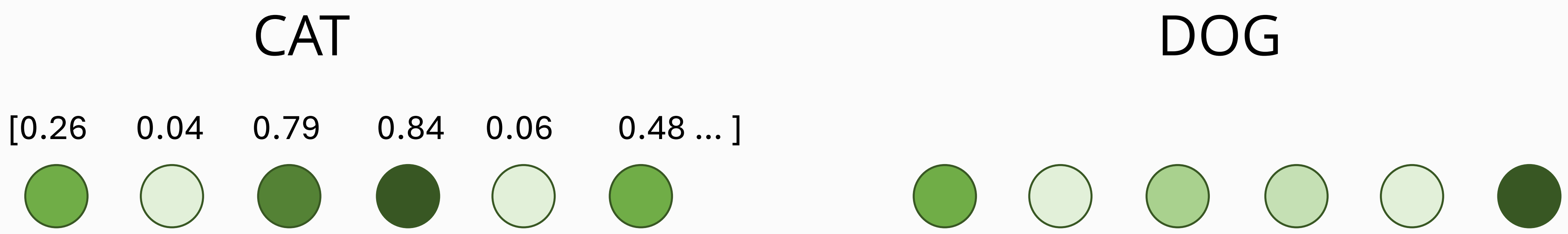
Artificial neural network



Neural networks

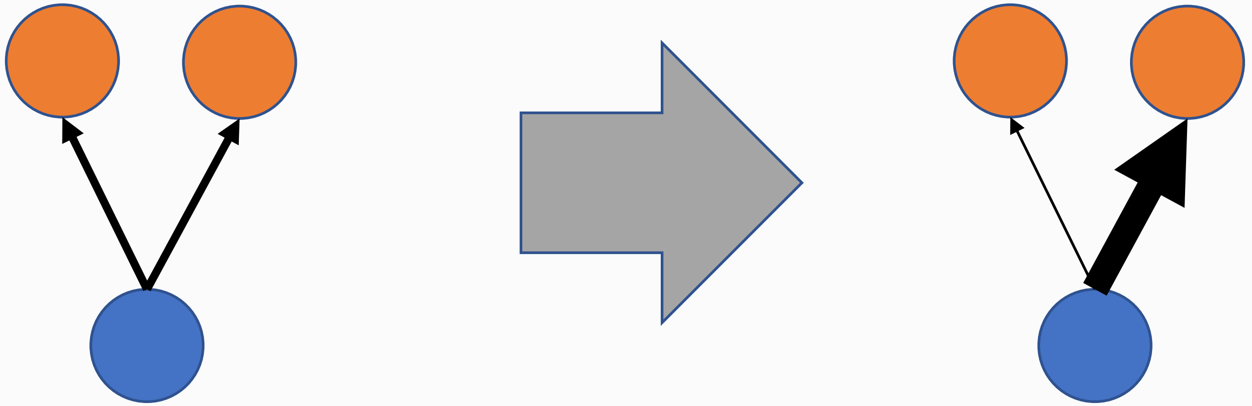
Artificial NNs can ‘represent’ concepts through patterns of activation (distributed representation)

	#1	#2	#3	coordinate
CAT	✔			1, 0, 0



Neural networks

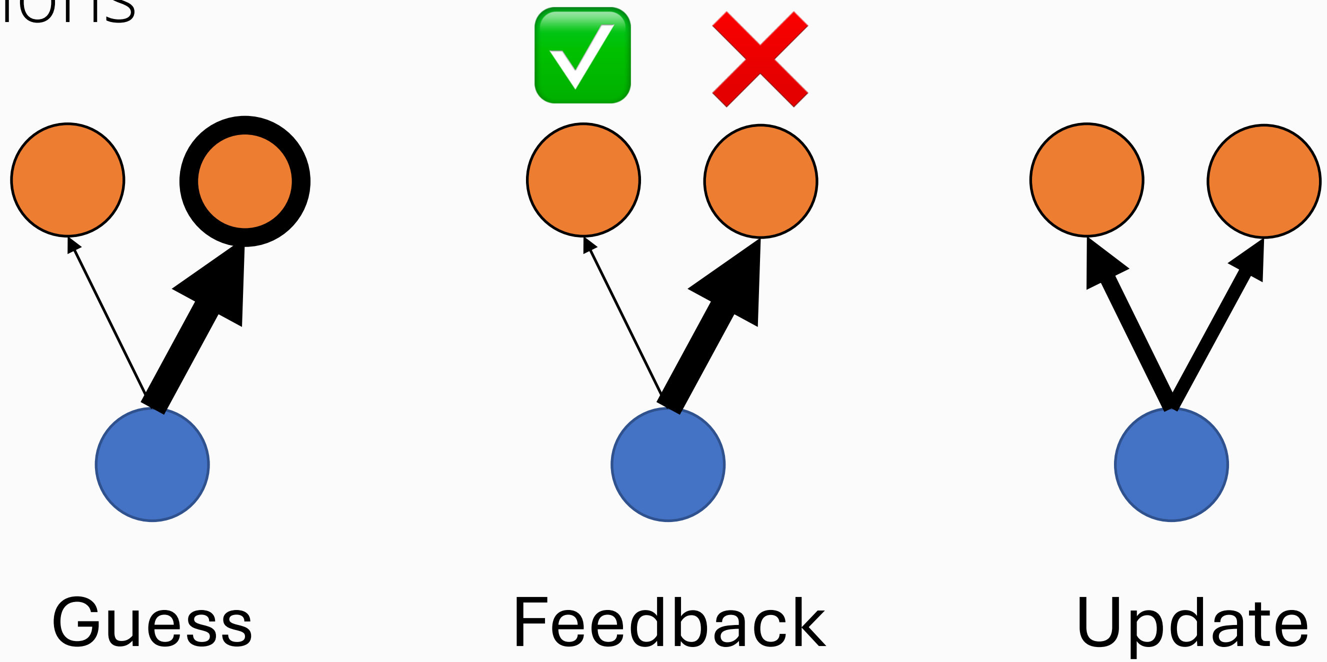
Artificial neural networks learn by making adjustments to connection strengths (also called weights)



The size of the arrow illustrates the strength of the connection

Neural networks

They can use **error-driven learning**: adjusting their connection strengths to better match the desired output over many iterations

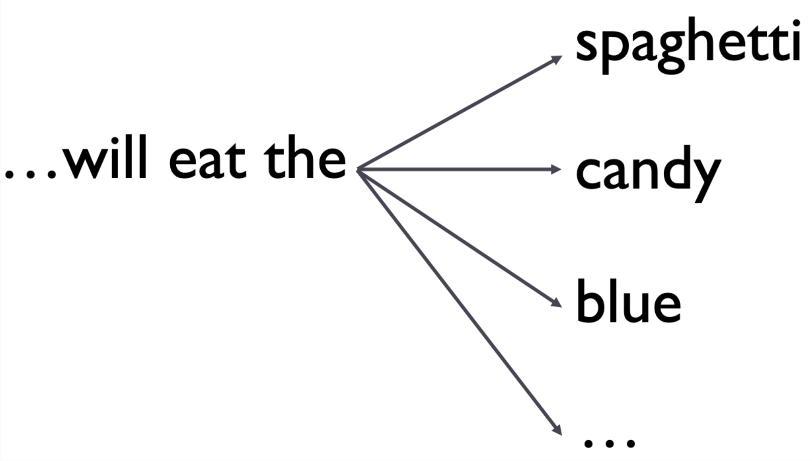


Building an artificial neural network

How might we build an artificial neural network that can do things we seem to do with language?

Model 3: Simple Recurrent Network (SRN)

Its task: predict what word will come next in the sentence



Elman (1990)

Simple Recurrent Network (SRN)

Its task: predict what word will come next in the sentence

The input:

- Toy vocabulary of 29 words (man, cat, rock, sleep, break, etc.)
- forming 10,000 valid sentences
  - *The man broke the rock*
  - *The cat chased the mouse*
  - etc.

Elman (1990)

Simple Recurrent Network (SRN)

Its task: predict what word will come next in the sentence

How is this all input into the model?

- Just a bunch of 1s and 0s for each word

0000000000000000000000000000010 (woman)

0000000000000000000000000010000 (smash)

Elman (1990)

Simple Recurrent Network (SRN)

The punch line: over time, the model adjusts its internal weights to better match the intended target

- Remember the arrows and how thick they are? It's actually a bunch of math instead.

- Importantly, it starts off with no idea what a "man" is, or "sleep", or "rock".
  - It just has a string of numbers corresponding to each word, which the researcher has input into the model.

- It ultimately figures out how to predict the next work in the sentence. How does it do this?

Elman (1990)

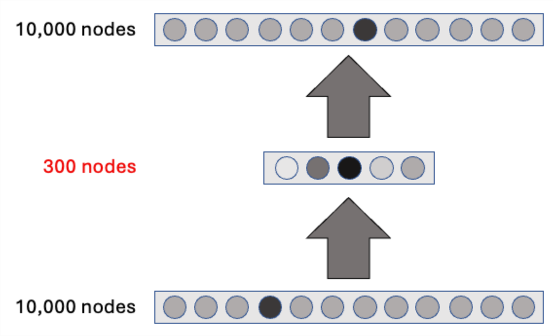
How an SRN works

Model architecture

How an SRN works

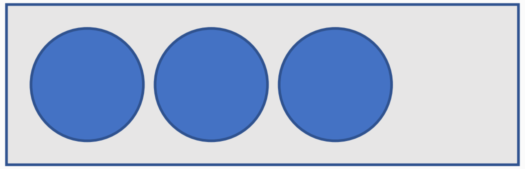
Model architecture

(recall word2vec)



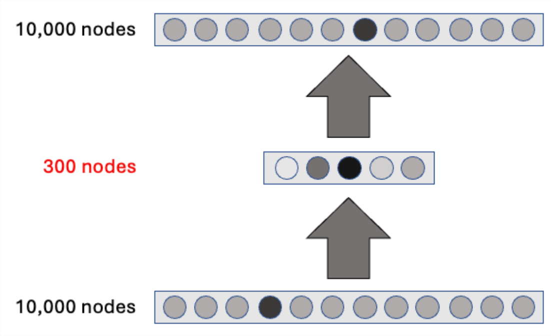
How an SRN works

Model architecture



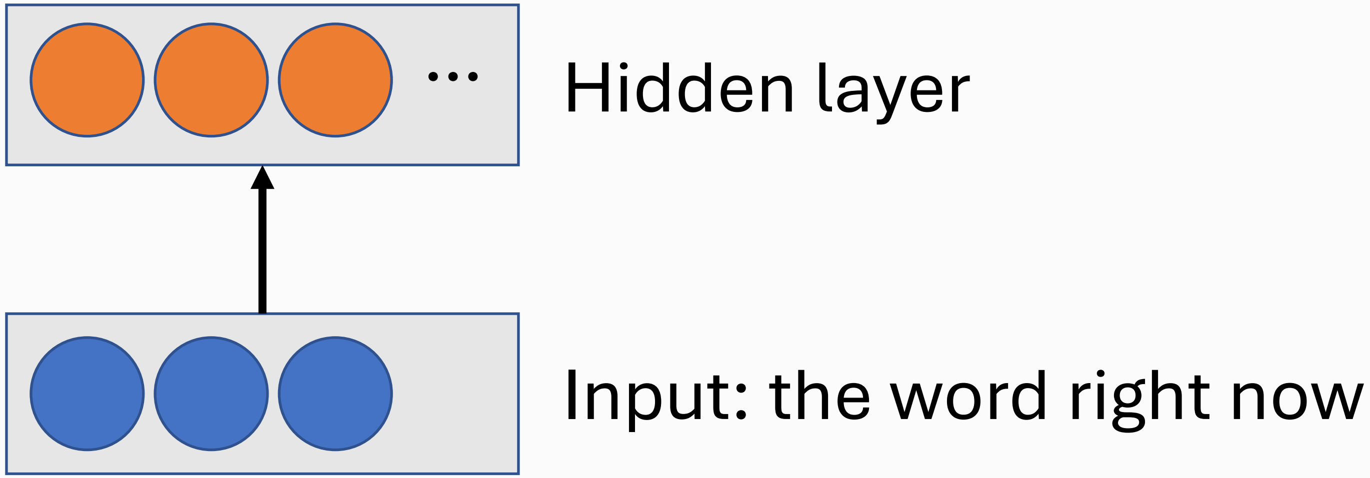
Input: the word right now

(recall word2vec?)

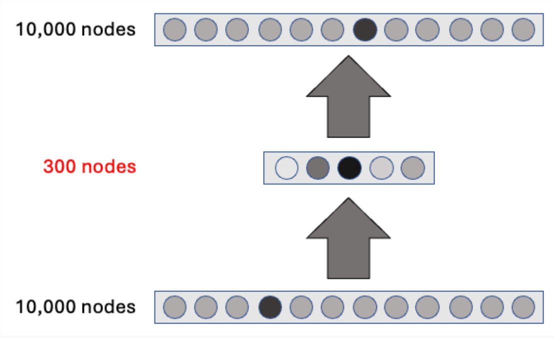


How an SRN works

Model architecture

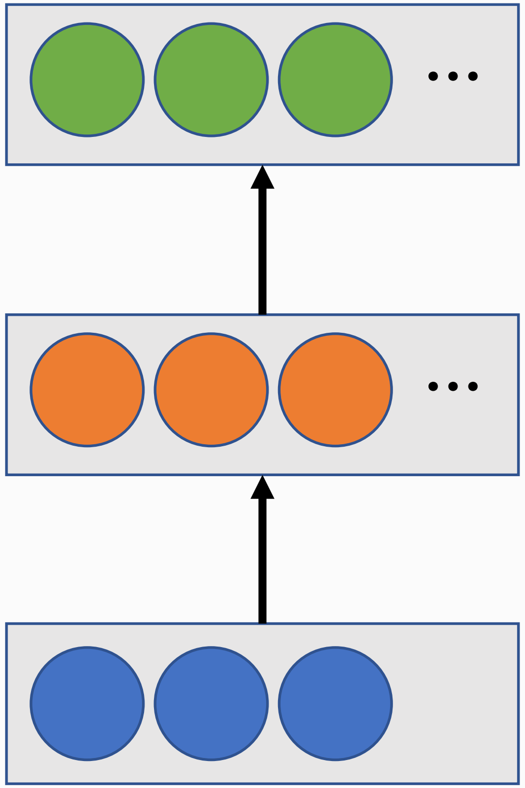


(recall word2vec?)



How an SRN works

Model architecture

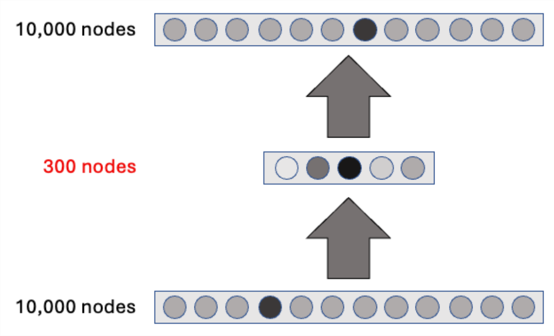


Output: guess what word comes after

Hidden layer

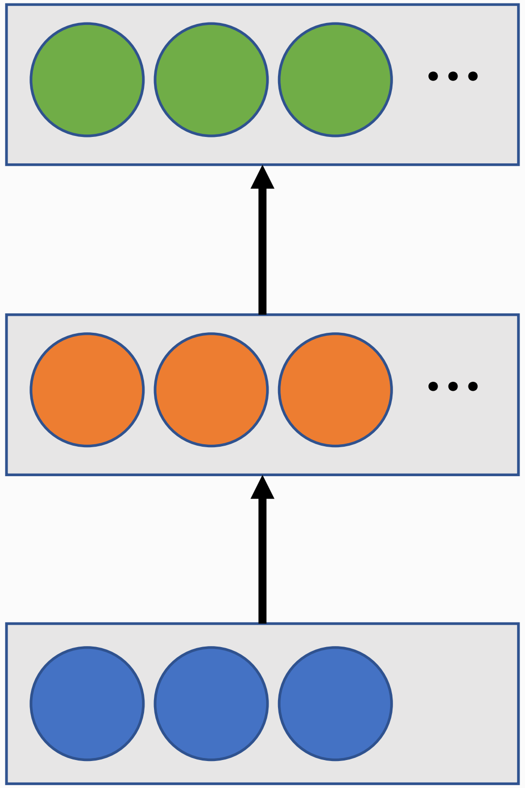
Input: the word right now

(recall word2vec?)



How an SRN works

Model architecture: three layers with a number of nodes

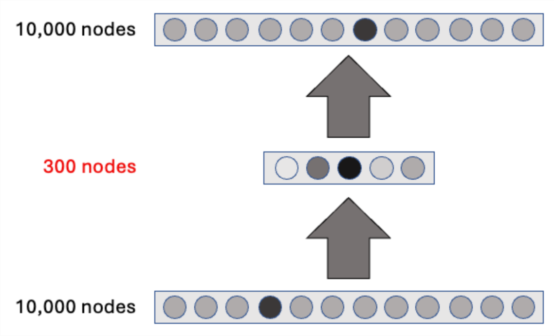


Output: guess what word comes after

Hidden layer

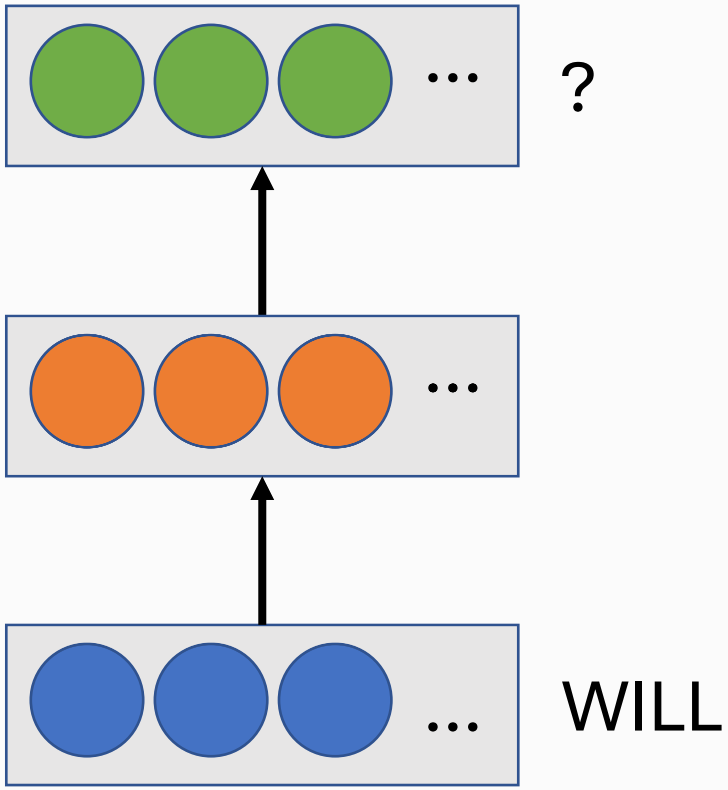
Input: the word right now

(recall word2vec?)



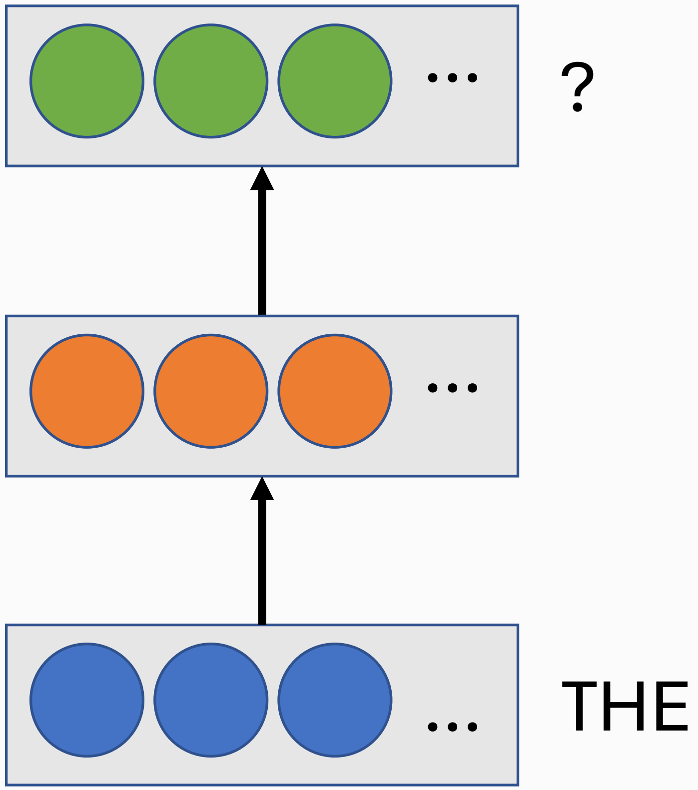
How an SRN works

Example



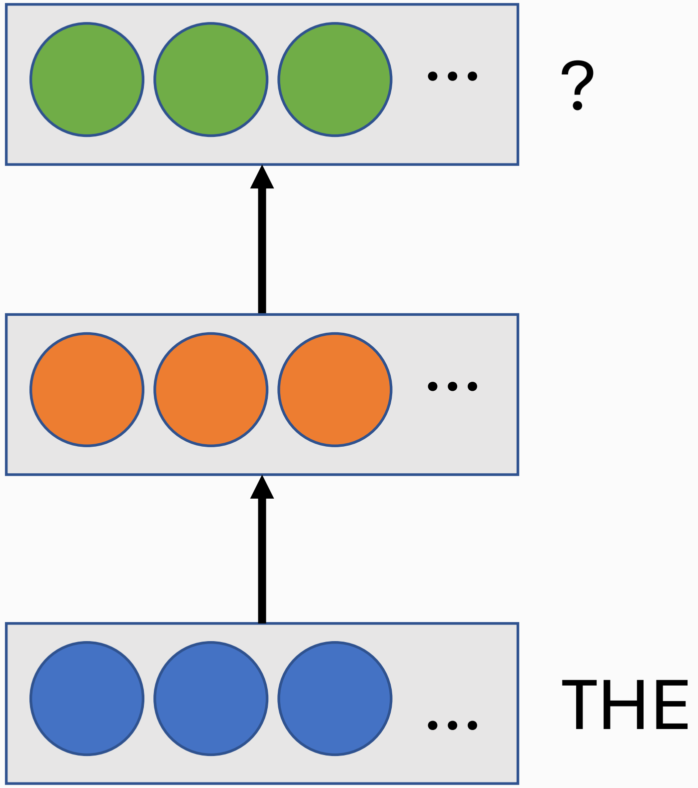
How an SRN works

Example



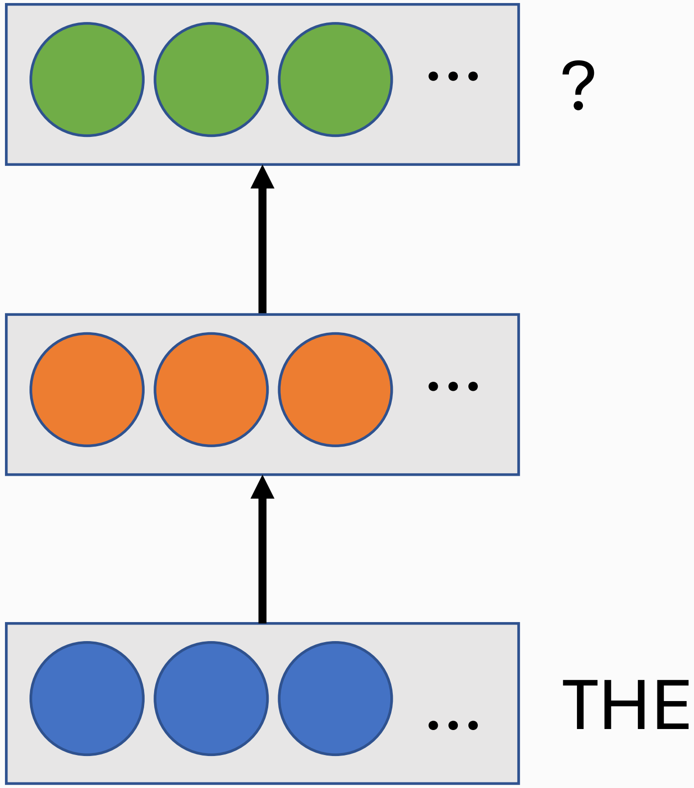
How an SRN works

Problem! If it only sees one word at a time, it's difficult to predict what comes next



How an SRN works

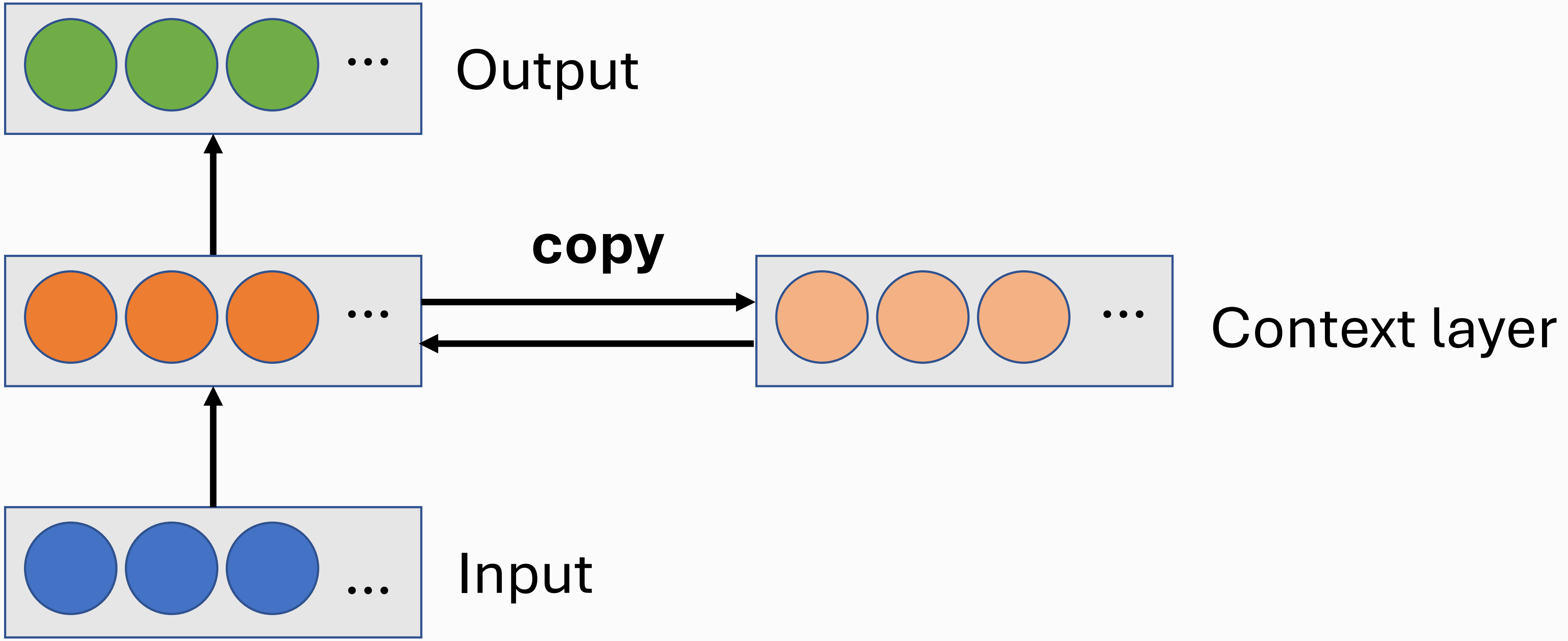
What do we need to add?



How an SRN works

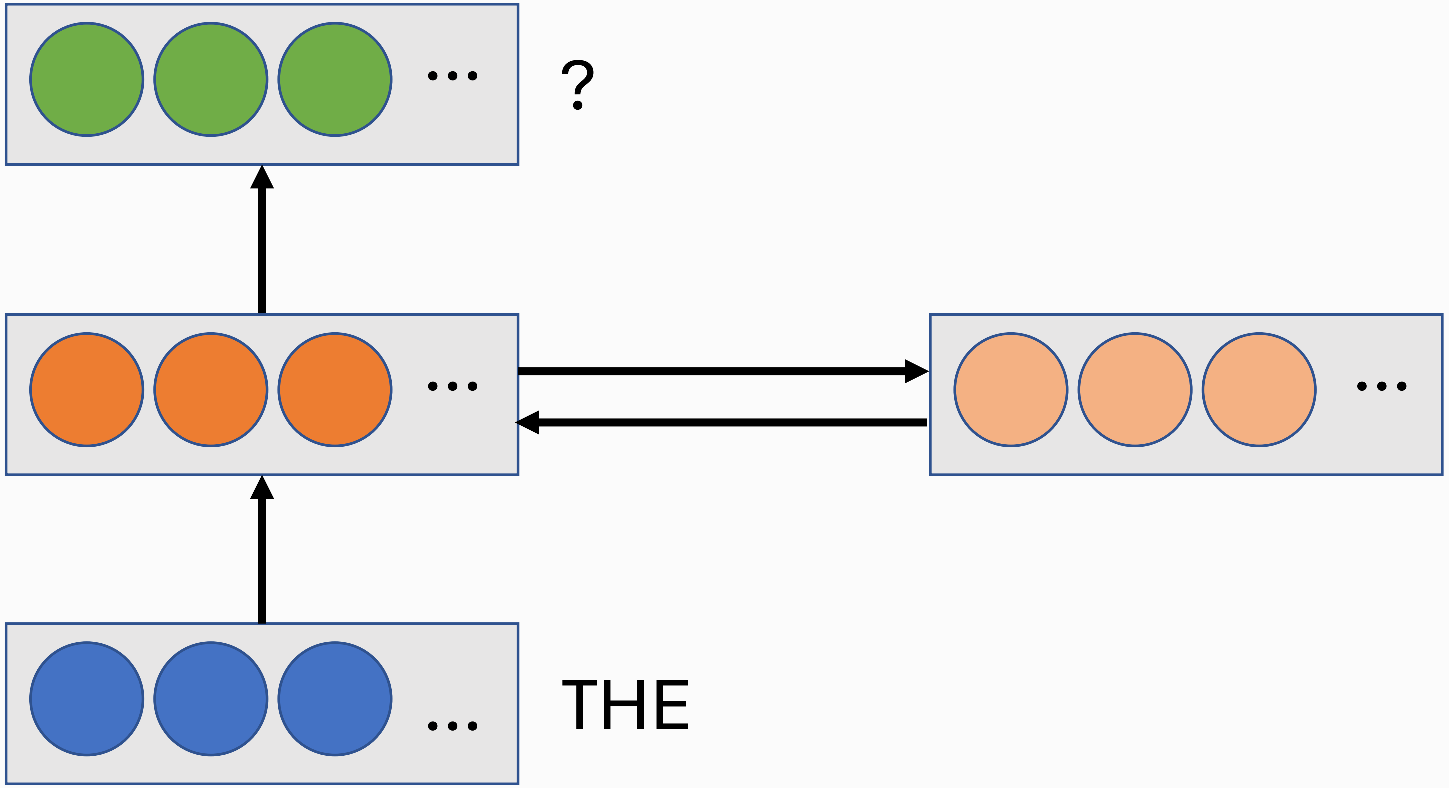
Model architecture update

Solution: give it memory of what came before



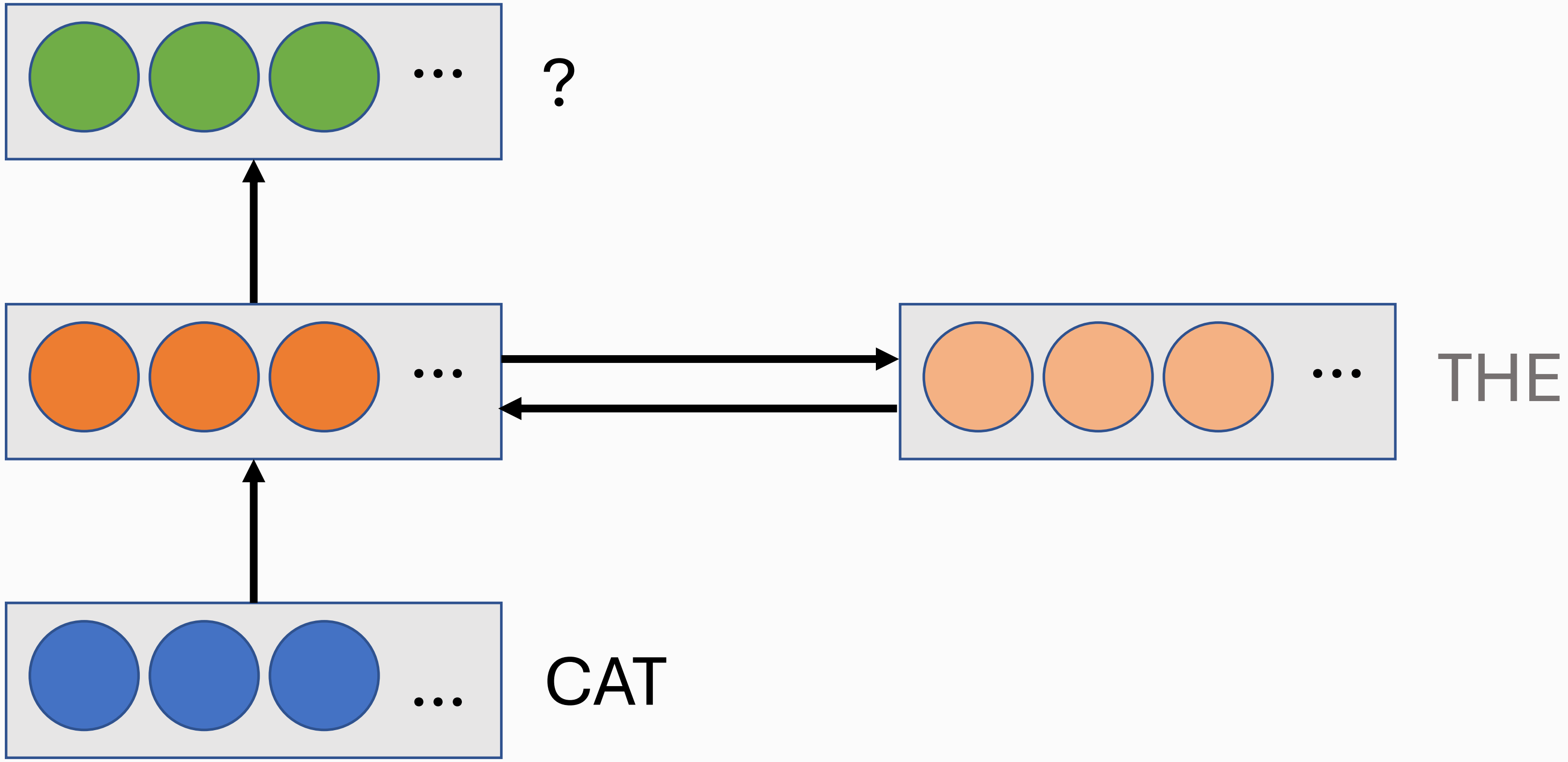
How an SRN works

Solution: give it memory of what came before



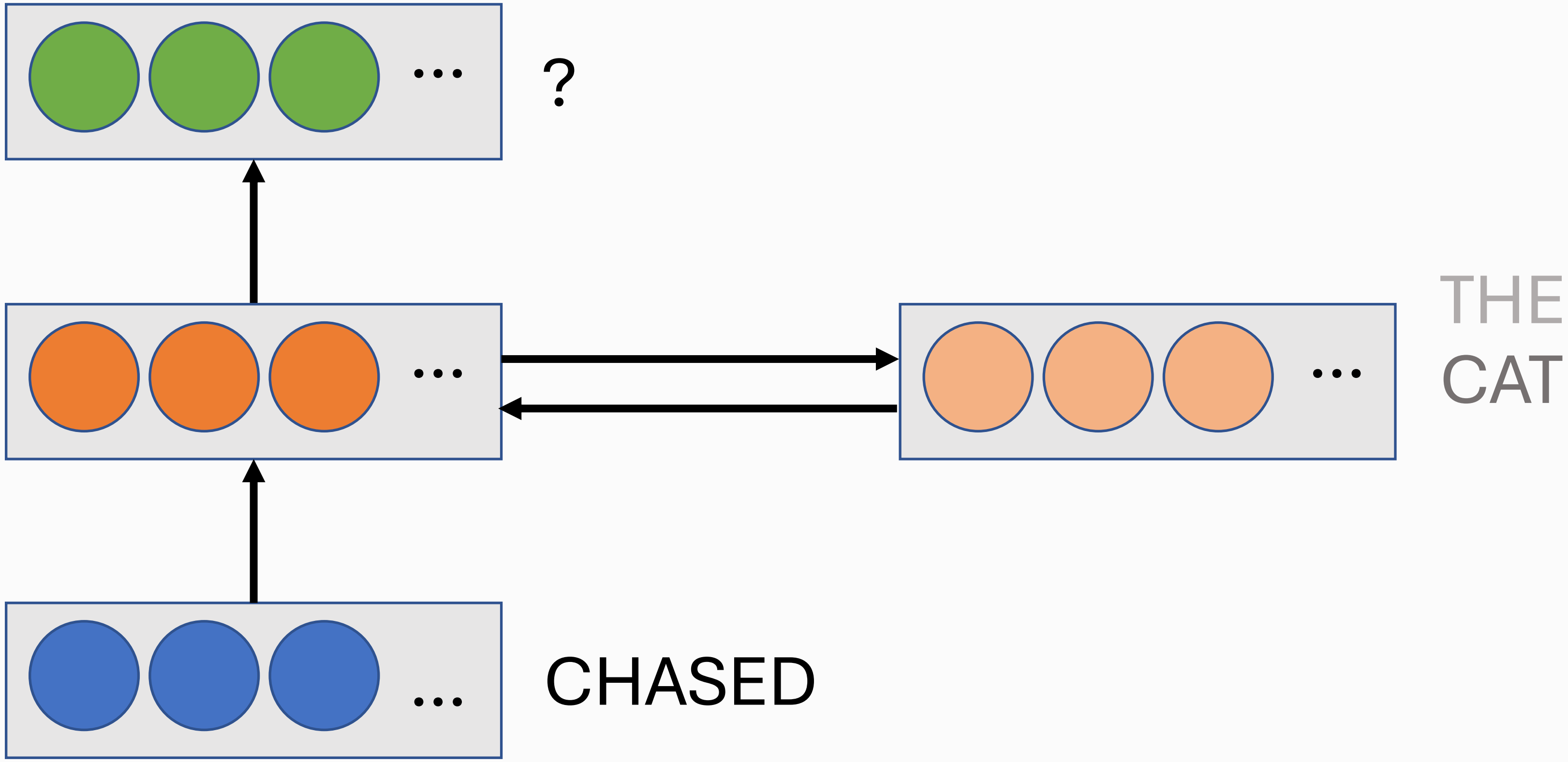
How an SRN works

Solution: give it memory of what came before



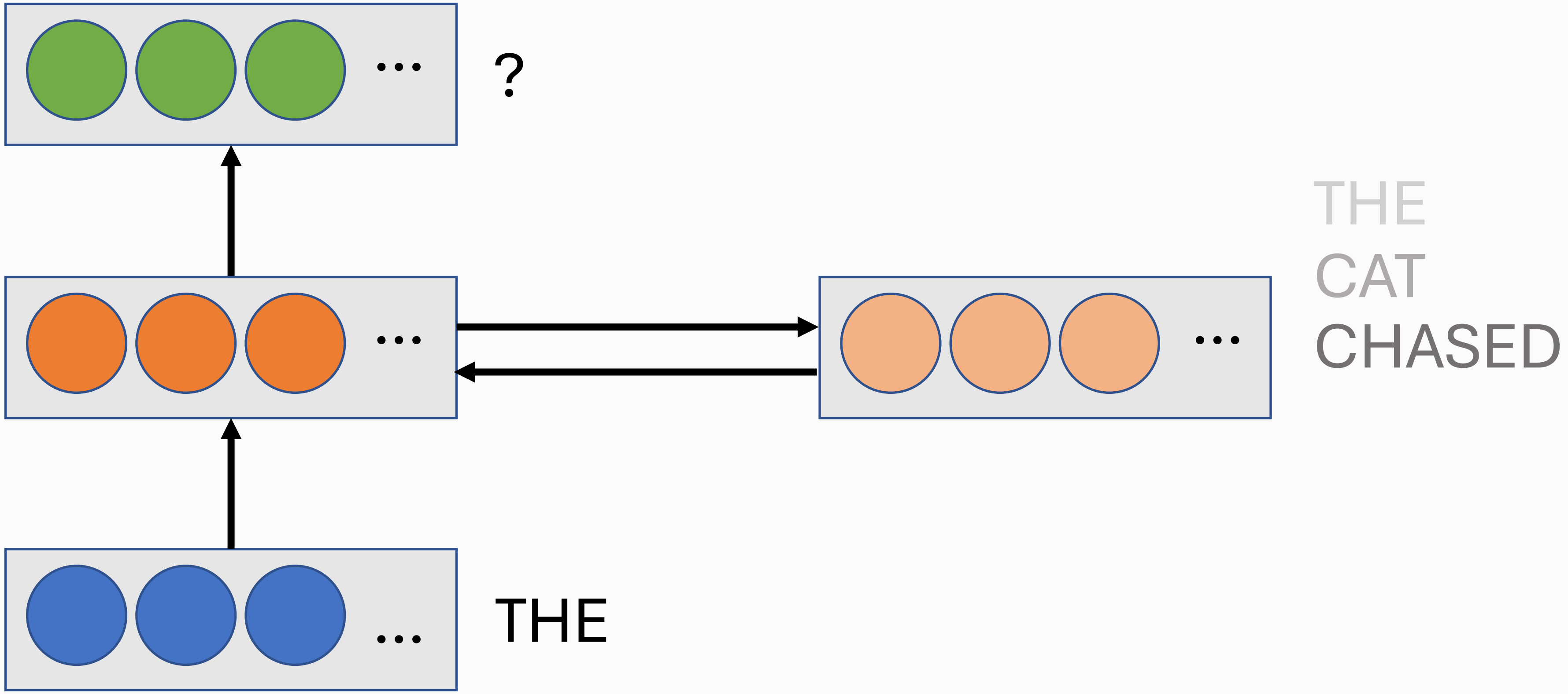
How an SRN works

Solution: give it memory of what came before



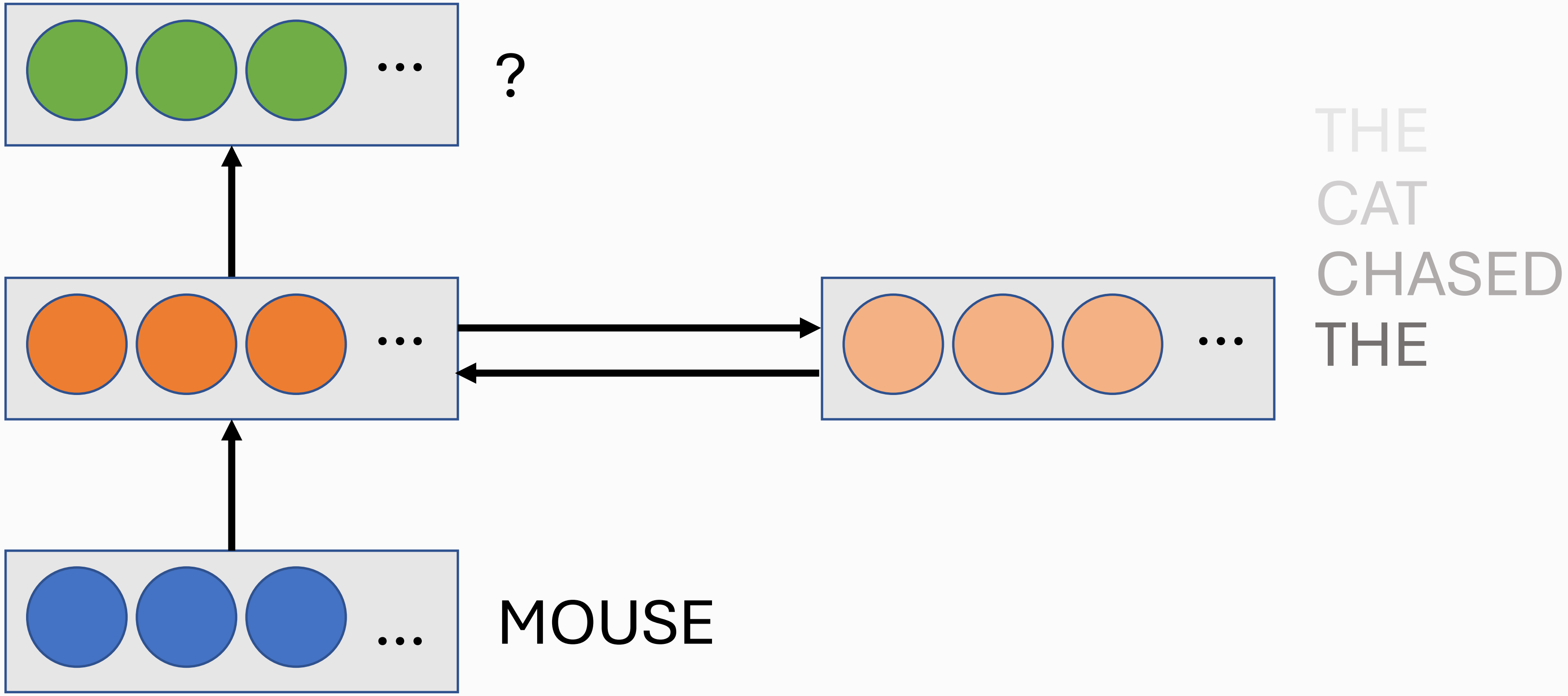
How an SRN works

Solution: give it memory of what came before



How an SRN works

Solution: give it memory of what came before

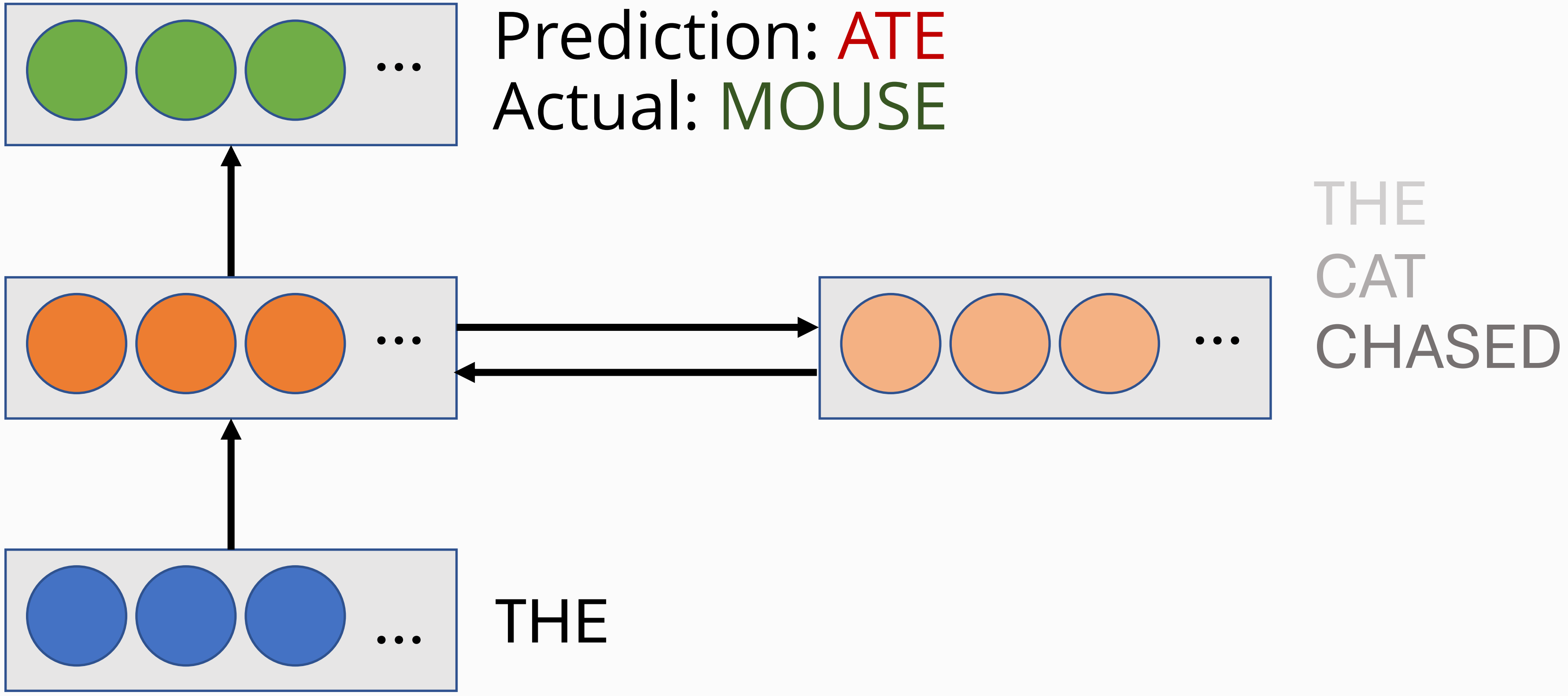


How an SRN works

- The SRN is now able to use prior context to make predictions about upcoming words – but how does it learn what the right predictions are?
- Through a combination of **error-driven learning** and **backpropagation**: Over time, the model adjusts its internal weights to better match the actual next input
- At each word, the SRN makes a guess what the next word will be, then it compares it to the actual next word (the target) to see if it guessed correctly or incorrectly

How an SRN works

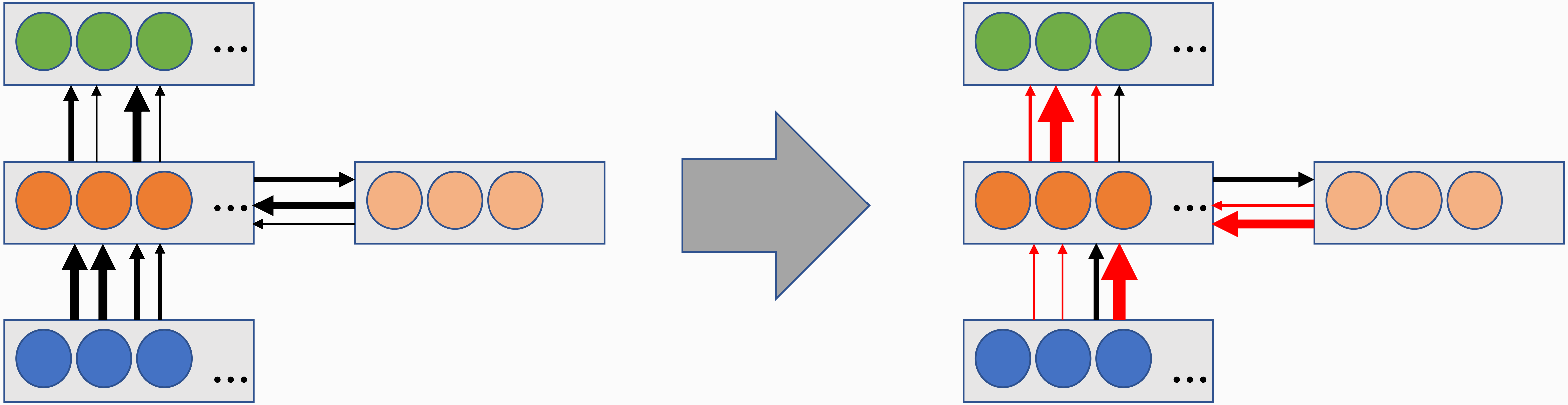
Error-driven learning: The model figures out what changes to make to its connection strengths so it gets a closer answer next time



How an SRN works

Model architecture update

It does this via **backpropagation**: The model figures out what changes to make to its connection strengths so it gets a closer answer next time

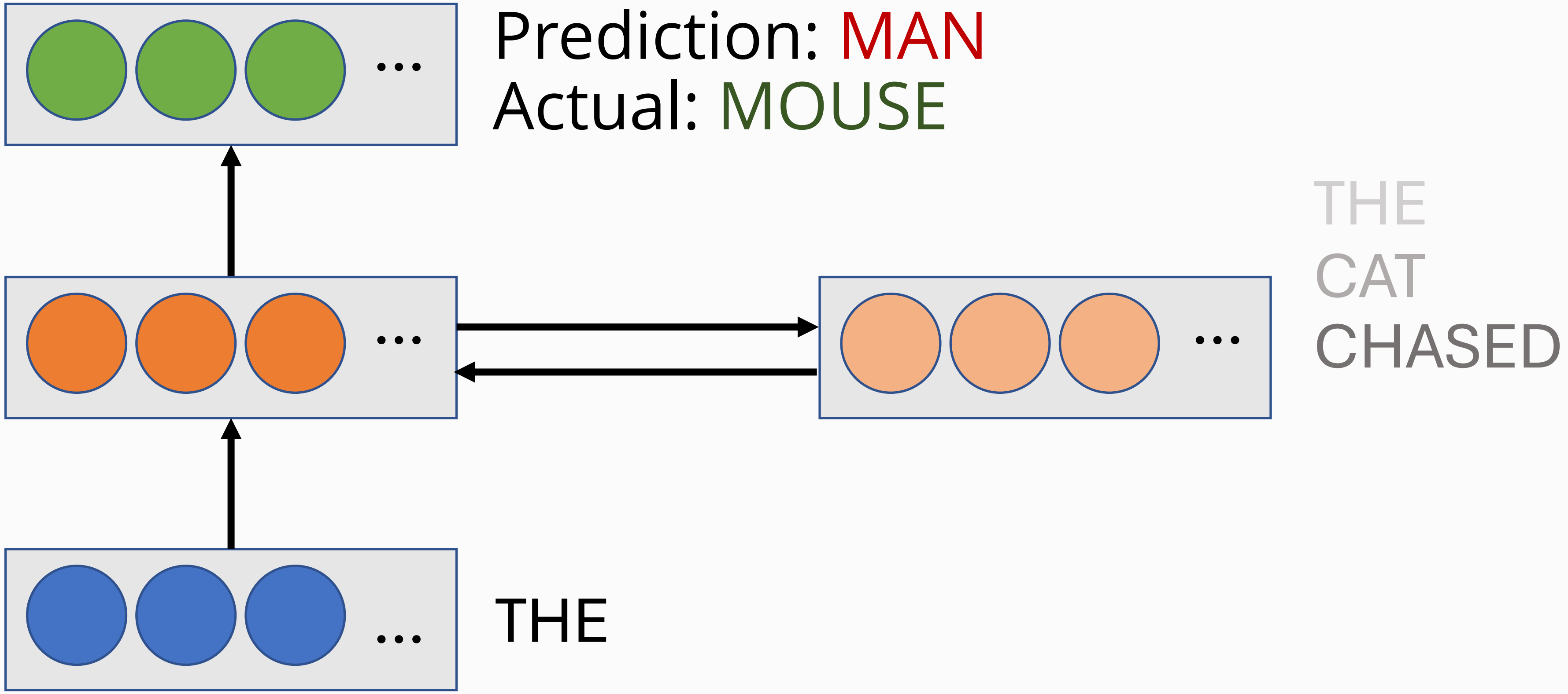


Original weights (shown as arrow thickness)

Updated weights in red (shown as arrow thickness)

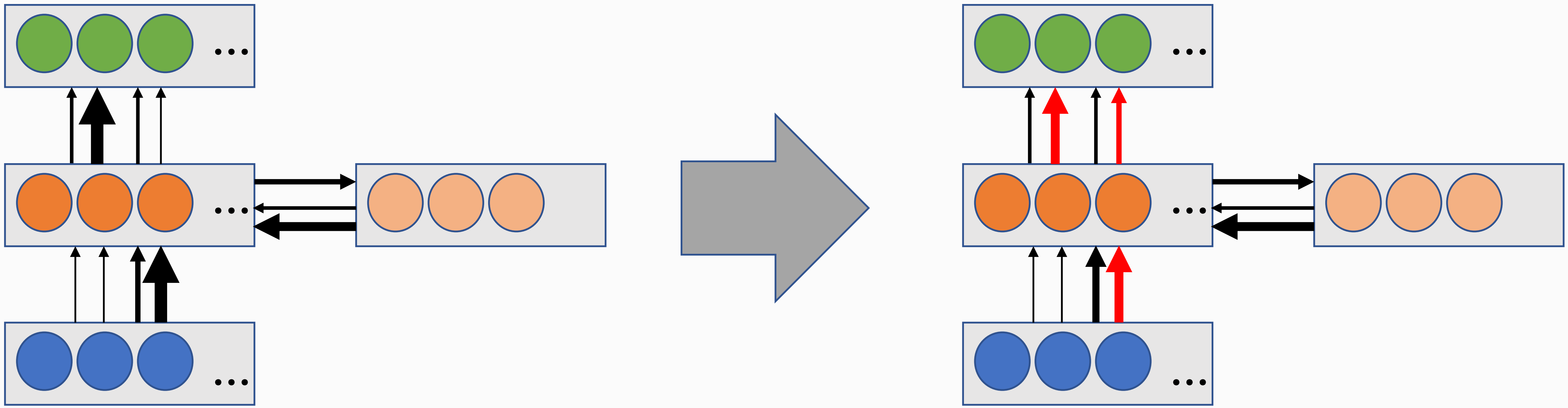
How an SRN works

When it is a closer guess, it doesn't have to make as many changes.



How an SRN works

When it is a closer guess, it doesn't have to make as many changes.



Original weights (shown as arrow thickness)

Updated weights in red (shown as arrow thickness)

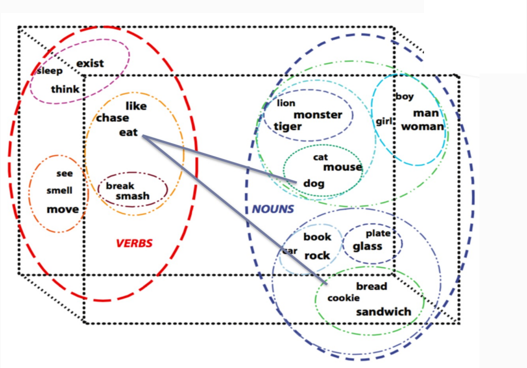
What can we learn from an SRN?

- What did the SRN learn about language from getting better at prediction?
    - Elman (1990) looked at the activation patterns for each word in the hidden layer
- Then he compared the similarity between these activation patterns through hierarchical clustering analysis

Elman (1990)

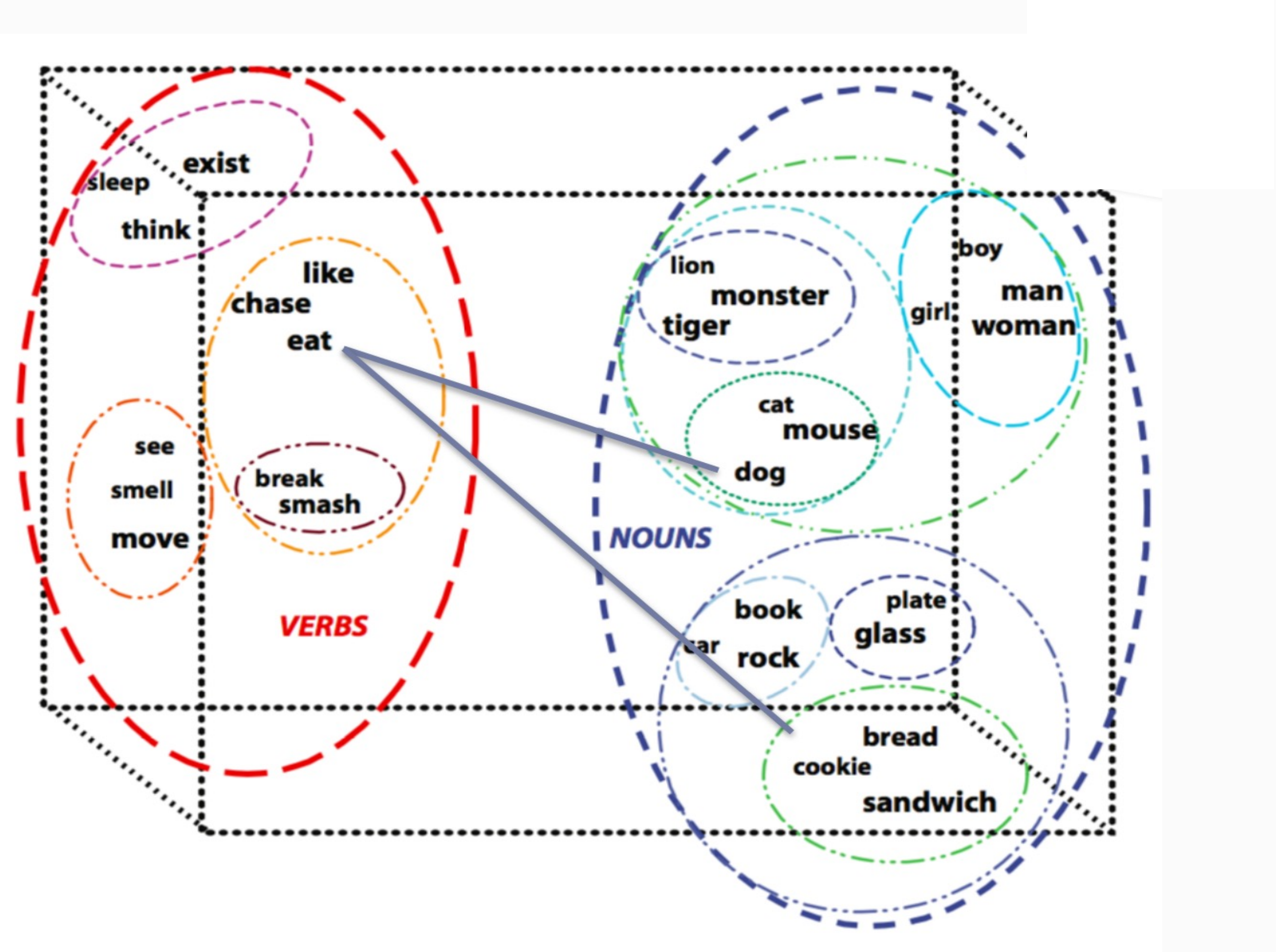
What can we learn from an SRN?

- What did the SRN learn about language from getting better at prediction?
  - Elman (1990) looked at the activation patterns for each word in the hidden layer
- Then he compared the similarity between these activation patterns through hierarchical clustering analysis
  - The SRN developed hierarchical categories!
    - Nouns vs verbs
    - Animates vs inanimates
    - Human vs nonhuman
    - Transitives vs intransitives
    - Etc.



Elman (1990)

What can we learn from an SRN?



Elman (1990)

What can we learn from an SRN?

The task of the model: to predict the next word
The result of the model: categories based on words' contexts

The SRN learned about the contexts in which words could occur
- 'Verbs' come after what we call 'nouns' (syntactic knowledge!)
- 'Edible' things are likely to be mentioned after words like eat (semantic knowledge!)
- 'Animate' things will precede words like eat or chase (semantic knowledge!)

This knowledge is referred to as **emergent representation** because the categories emerged from the architecture of the model.

Elman (1990)

What can we learn from an SRN…about the brain?

Well…

- **Some of the model's architecture matches pretty well with what happens with humans**
  - For example, when hearing a sentence, we are good at predicting what words come next
  - We rely on category information, such as whether something is edible, to make better predictions
- **But how would backpropagation work in the mind?**
  - This part of the model is *not a good model* of the human mind
- There are also some syntactic structures that these models cannot learn, but humans use them!

Wrapping up

Model	Develops semantic space	Task
Latent Semantic Analysis	Explicitly	Count words in documents
Word2Vec	Implicitly	Predict neighboring word
Simple Recurrent Network	Implicitly	Predict upcoming word from context

Wrapping up

- Why models?
  - To think about human behavior in more concrete ways
  - Models take an input and give us an output, and the way it does it can offer insight into human behavior (or can it?!)

Key concepts

- ✓Artificial neural network
- ✓Simple Recurrent Network
- ✓Hidden layer (and why we call it a black box)
- ✓Error-driven learning
- ✓Backpropagation
- ✓Emergent representations
- ✓Similarities and differences between models and the mind/brain
- ✓Why models?