Reminder: Plan for Remaining Weeks

Date	Topic	Hw?
April 8	n-gram models	yes ✓
April 10	Python (functions)	
April 15	n-gram models (cont.)	yes ✓
April 17	Python (tokens)	
April 22	Machine learning	yes ⇐
April 24	Python (ngrams)	
April 29	Human-like models	practice/Final project
May 01	Python (frequencies)	
May 06	Summary	Final project
May 08	Python Q&A session	
May 14		Final Project due (at midnight)

Language & Technology

Lecture 8: Machine Learning

Alëna Aksënova & Aniello De Santo

Stony Brook University
alena.aksenova@stonybrook.edu
aniello.desanto@stonybrook.edu

Why Machine Learning?

- For many tasks, designing models by hand is
  - too expensive (thousands of man hours), and/or
  - too inflexible (cannot adapt to changes), and/or
  - simply impossible

Example

- spam filter
- broad-coverage grammars for hundreds of languages
- machine translation systems

- Computers have to be able to learn from input on their own.
- **Also:** Humans learn, too; we don't get English with our genes.

What is Learning?

All of the following are colloquially called "learning",
but they are **not the same**:

- learning to walk
- learning the names of all US presidents
- learning self-discipline
- learning tennis
- learning addition
- learning French (as a second language)

Parameters for Types of Learning

	instruction	end state	generalization	categorical
walking	×	✓	×	?
presidents	✓	✓	×	✓
discipline	×	×	×	×
Tennis	✓	?	?	?
addition	✓	✓	✓	✓
French	✓	?	✓	×

Learning as Generalization

learning generalization from a finite set of inputs to a (possibly infinite) target class of outputs

The *Gavagai* Problem

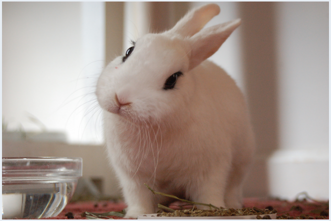
- Suppose you are on a remote island, trying to learn the language of the locals.
- One of them points at a rabbit that just jumped out of the bushes and says "gavagai".
- What does *gavagai* mean?

Learning as Generalization

learning: generalization from a finite set of inputs to a (possibly infinite) target class of outputs

The *Gavagai* Problem

- Suppose you are on a remote island, trying to learn the language of the locals.
- One of them points at a rabbit that just jumped out of the bushes and says "gavagai".
- What does *gavagai* mean?

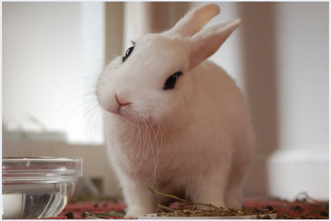


Learning as Generalization

learning generalization from a finite set of inputs to a (possibly infinite) target class of outputs

The *Gavagai* Problem

- Suppose you are on a remote island, trying to learn the language of the locals.
- One of them points at a rabbit that just jumped out of the bushes and says "gavagai".
- What does *gavagai* mean?
  - rabbit
  - animal
  - Look there!
  - Watch out!
  - How cute!
  - There's our dinner!
  - Pull my finger!



Blank Slate Learning is Impossible

- **David Hume (1711-1776)**
  Learning is preconception-free **blank slate generalization**.
- The Gavagai problem shows that Hume cannot be right.
- There's always infinitely many different ways to generalize.
- A learner must have preconceived notions of what makes for a good generalization.

Prior Knowledge: How Babys Learn

- **Importance of Prosody**
  Babys already pay attention to prosody in the womb.
- **Words: They're a Thing**
  Babys quickly learn to probabilistically detect word boundaries.
- **Generalizations Between Words**
  Once a child realizes that *who* and *which* must be at the beginning of questions, they immediately generalize this to other question words like *when* and *how*.

Conclusion

Humans are genetically hard-wired to pay attention to specific aspects of language and generalize then in a specific way.

Prior Knowledge: Lexical Gaps

Many concepts are never lexicalized in any languages as they do not represent common human generalization patterns.

Example

- taller than half the people in the room
- an even number of years old
- a word that is its semantic opposite when read backwards
- more bulky than heavy
- not made in the US

Prior Knowledge: Non-Existent Generalizations

- Artificial language learning experiments reveal how adults reason about language.
- Test subject is given new words and examples of their usage.
- They must then use the word in new examples.
- Words with “natural” meanings are learned correctly, whereas words with unnatural words aren’t acquired even with lots of data.

A Quick Experiment

blip	not *blip*	*blip*	not *blip*			
not *gnok*	not *gnok*	*gnok*	*gnok*			
not *bnik*	not *bnik*	*bnik*	not *bnik*			
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*		
not *gnok*	not *gnok*	*gnok*	*gnok*			
not *bnik*	not *bnik*	*bnik*	not *bnik*			
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	
not *gnok*	not *gnok*	*gnok*	*gnok*			
not *bnik*	not *bnik*	*bnik*	not *bnik*			
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*			
not *bnik*	not *bnik*	*bnik*	not *bnik*			
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*			
not *bnik*	not *bnik*	*bnik*	not *bnik*			
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

blip not red

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*		
not *bnik*	not *bnik*	*bnik*	not *bnik*			
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

blip not red

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	
not *bnik*	not *bnik*	*bnik*	not *bnik*			
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

blip not red

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*			
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

blip not red

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*			
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

blip not red

gnok brown or rectangular

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*		
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

blip not red

gnok brown or rectangular

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

blip not red

gnok brown or rectangular

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	*bnik*
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

blip not red

gnok brown or rectangular

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	*bnik*
glop	*glop*	not *glop*	*glop*			
blok	*blok*	*blok*	*blok*			

blip	not red
gnok	brown or rectangular
bnik	blip and gnok

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	*bnik*
glop	*glop*	not *glop*	*glop*	*glop*		
blok	*blok*	*blok*	*blok*			

blip not red

gnok brown or rectangular

bnik blip and gnok

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	*bnik*
glop	*glop*	not *glop*	*glop*	*glop*	*glop*	
blok	*blok*	*blok*	*blok*			

blip not red

gnok brown or rectangular

bnik blip and gnok

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	*bnik*
glop	*glop*	not *glop*	*glop*	*glop*	*glop*	*glop*
blok	*blok*	*blok*	*blok*			

blip not red

gnok brown or rectangular

bnik blip and gnok

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	*bnik*
glop	*glop*	not *glop*	*glop*	*glop*	*glop*	*glop*
blok	*blok*	*blok*	*blok*			

blip not red

gnok brown or rectangular

bnik blip and gnok

glop bnik and if not brown, then not rectangular

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	*bnik*
glop	*glop*	not *glop*	*glop*	*glop*	*glop*	*glop*
blok	*blok*	*blok*	*blok*	not *blok*		

blip not red

gnok brown or rectangular

bnik blip and gnok

glop bnik and if not brown, then not rectangular

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	*bnik*
glop	*glop*	not *glop*	*glop*	*glop*	*glop*	*glop*
blok	*blok*	*blok*	*blok*	not *blok*	not *blok*	

blip not red

gnok brown or rectangular

bnik blip and gnok

glop bnik and if not brown, then not rectangular

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	*bnik*
glop	*glop*	not *glop*	*glop*	*glop*	*glop*	*glop*
blok	*blok*	*blok*	*blok*	not *blok*	not *blok*	not *blok*

- blip: not red
- gnok: brown or rectangular
- bnik: blip and gnok
- glop: bnik and if not brown, then not rectangular

A Quick Experiment

blip	not *blip*	*blip*	not *blip*	*blip*	*blip*	*blip*
not *gnok*	not *gnok*	*gnok*	*gnok*	*gnok*	*gnok*	*gnok*
not *bnik*	not *bnik*	*bnik*	not *bnik*	*bnik*	*bnik*	*bnik*
glop	*glop*	not *glop*	*glop*	*glop*	*glop*	*glop*
blok	*blok*	*blok*	*blok*	not *blok*	not *blok*	not *blok*

- blip: not red
- gnok: brown or rectangular
- bnik: blip and gnok
- glop: bnik and if not brown, then not rectangular
- blok: bnik or glop, but not both

Why Machine Learning is Hard

- The Gavagai problem is why machine learning is so hard.
- No matter how much data you have, it is not enough to tell you the correct way to generalize.
- Humans have to tell computers what generalizations they should entertain.
- But for many problems, humans don't know the answer either!

The Machine Learning Recipe

All machine learning follows the same procedure:

1. **Problem**
   produce outputs for inputs based on properties p, q, ...
   spam filter, face recognition, machine translation
2. **Supervised training**
   sample of predefined input-output pairings
   spam & ham emails, photos with names, original & translated text
3. **Testing**
   test performance of model on new test sample
4. **Rinse and repeat**
   change some parameters of model, train and test again

The Machine Learning Recipe

All machine learning follows the same procedure:

1. **Problem**
   produce outputs for inputs based on properties p, q, ...
   spam filter, face recognition, machine translation
2. **Supervised training**
   sample of predefined input-output pairings
   spam & ham emails, photos with names, original & translated text
3. **Testing**
   test performance of model on new test sample
4. **Rinse and repeat**
   change some parameters of model, train and test again

But what properties are relevant?
What parameters should be changed?

The Moral of the Story

- Any finite amount of input allows for an infinite number of different generalizations.
- But humans mostly generalize in the same ways.
- Due to our genetic endowment we are hardwired to generalize in certain ways but not in others.
- A **computer has no genetic hardwiring**, we must include it in the algorithm.

Machine Learning in a Nutshell

The hard part of machine learning is figuring out the right generalization mechanisms for a given problem.

The Moral of the Story

- Any finite amount of input allows for an infinite number of different generalizations.
- But humans mostly generalize in the same ways.
- Due to our genetic endowment we are hardwired to generalize in certain ways but not in others.
- A **computer has no genetic hardwiring**,
  we must include it in the algorithm.

Machine Learning in a Nutshell

The hard part of machine learning is figuring out the right generalization mechanisms for a given problem.

Interim Summary

- Learning always involves prior knowledge (the generalization strategy).
- The reason that we learn what we learn is that we are genetically hardwired to generalize only in specific ways.
- Computers must be given an effective generalization strategy for every new problem.
- This is **hard and takes lots of time**.

Interim Summary

- Learning always involves prior knowledge (the generalization strategy).
- The reason that we learn what we learn is that we are genetically hardwired to generalize only in specific ways.
- Computers must be given an effective generalization strategy for every new problem.
- This is **hard and takes lots of time**.

Whenever things get hard, people start looking for shortcuts. . .

A Current Hype: Deep Learning

- One learning model is all over the media right now: **deep learning**
- Deep learning = very large and complex neural networks
- Neural networks were inspired by the human brain.

Standard Model of the Human Brain

- connected network of neurons
- input activates neurons, which start "firing" (= emitting electrical current)
- current activates other neurons $\Rightarrow$ activation patterns
- learning = strengthening connection between specific neurons

A Current Hype: Deep Learning

- One learning model is all over the media right now: **deep learning**
- Deep learning = very large and complex neural networks
- Neural networks were inspired by the human brain.

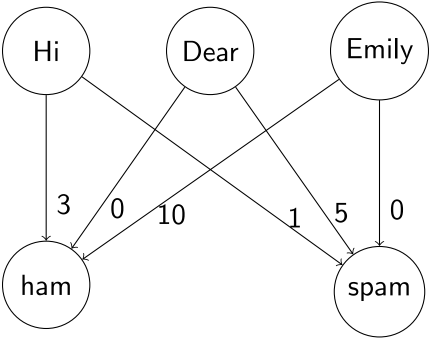
Standard Model of the Human Brain

- connected network of neurons
- input activates neurons, which start "firing" (= emitting electrical current)
- current activates other neurons $\Rightarrow$ activation patterns
- learning = strengthening connection between specific neurons

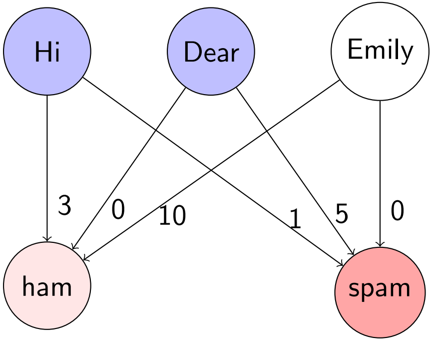
The Perceptron

The Perceptron: A Mini-Version of a Neural Network

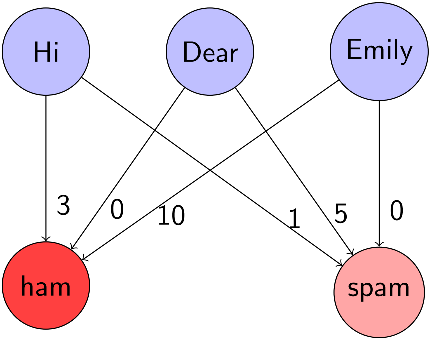
- **input layer:** neurons that are sensitive to input
- **output layer:** neurons that represent output values
- **connections:** weighted links between input and output layer
- most activated output neuron represents decision



Perceptron Activation for *Hi Dear*



Perceptron Activation for *Hi Dear Emily*



Training the Perceptron

The perceptron learns in a strange way:

1. **Rewiring Step**
   if output is wrong, randomly change weights of links
2. **Evaluation Step**
   - if output is closer to intended result, keep new weights
   - otherwise keep previous weights
3. **Iteration Step**
   if output is still wrong, return to first step

Training the Perceptron

The perceptron learns in a strange way:

1. **Rewiring Step**
   if output is wrong, randomly change weights of links
2. **Evaluation Step**
   - if output is closer to intended result, keep new weights
   - otherwise keep previous weights
3. **Iteration Step**
   if output is still wrong, return to first step

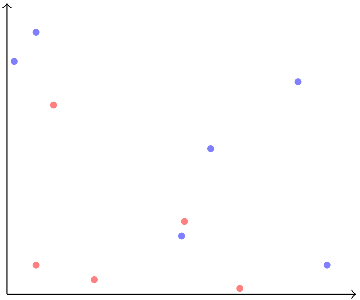
Learning = Iterated Trial and Error

What Can be Learned: Linear Separability

Theorem (Perceptron Learning)

A target class can be learned perfectly by the perceptron if and only if it is **linearly separable**.

linearly separable target class can be delineated by a single line

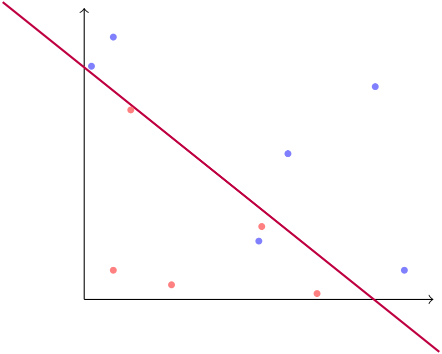


What Can be Learned: Linear Separability

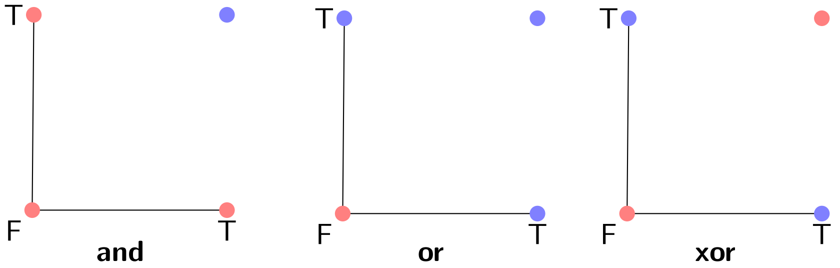
Theorem (Perceptron Learning)

A target class can be learned perfectly by the perceptron if and only if it is **linearly separable**.

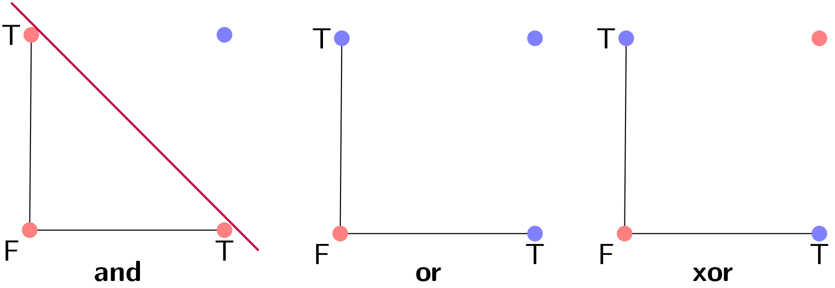
linearly separable target class can be delineated by a single line



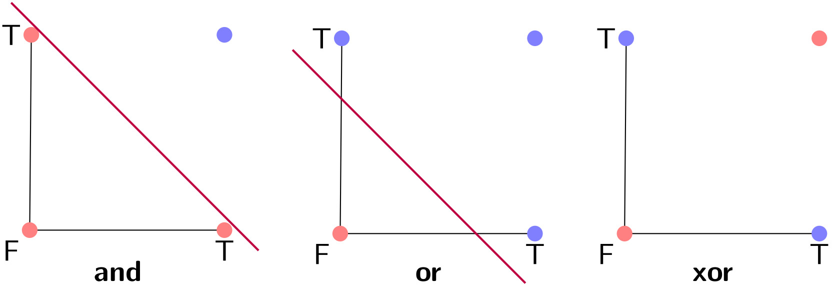
Example: Calculating Truth



Example: Calculating Truth



Example: Calculating Truth



Improving the Perceptron: Neural Network

Perceptron is too weak for classes that are not linearly separable.

Neural Networks and Deep Learning

Neural networks extend the perceptron with

- intermediate layers, and
- feedback loops between layers.

Deep learning uses neural networks with dozens of layers
$\Rightarrow$ no revolutionary techniques, just super-sizing old ideas

Why Neural Networks are Loved and Hated

Neural networks are a **tremendous improvement** because

- they work surprsingly well with no explicit guidance;
- they work across many different domains;
  machine translation, melanoma diagnosis, self-driving car
- instead of understanding the problem,
  we can brute-force it with trial and error.

Neural networks are **unsatisfying** because

- they are too complex to tell what is going on;
- it is all trial-and-error with shaky theoretical foundation;
- there are no safeties or learning guarantees;
- minor changes in data or model can have huge effects.

Is the Brain a Perceptron/Neural Network?

A Nasty Truth

We have no idea how the brain computes!

- The physiology of the brain has limited bearing on computation:
  - a program can have very different hardware instantiations
  - a hardware activation can compute very different things
- The physiology of the brain is open to interpretation:
  - Do neurons have activation levels?
  - Computer transistors only have two states (on and off) despite the range of transistor voltages being almost infinite.

The Real-World Failure

- We have a full "brain map" of roundworm *C. elegans*.
- 302 neurons and links between them
- Yet we have no idea how this brain works!

Is the Brain a Perceptron/Neural Network?

A Nasty Truth

We have no idea how the brain computes!

- The physiology of the brain has limited bearing on computation:
  - a program can have very different hardware instantiations
  - a hardware activation can compute very different things
- The physiology of the brain is open to interpretation:
  - Do neurons have activation levels?
  - Computer transistors only have two states (on and off) despite the range of transistor voltages being almost infinite.

The Real-World Failure

- We have a full “brain map” of roundworm *C. elegans*.
- 302 neurons and links between them
- Yet we have no idea how this brain works!

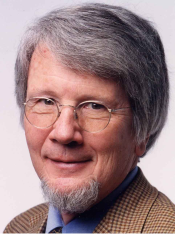
A Different View of Neural Computation

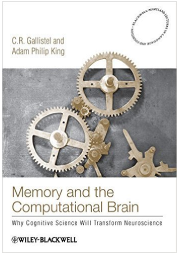
Randy Gallistel (Psychologist@Rutgers)

- Neuroscience has it all wrong.
- We need theory of computation.
- Good starting point:
  Memory data structures?
  How are they instantiated?

Check out the **Brain Science Podcast**

`https://www.youtube.com/playlist?list=PLUSRfoOcUe4bf0ly1WSvxagPiG3569Al1`





Zipf's Law Strikes Again!

- Neural networks have a bigger problem than cognitive reality: **Zipf's Law!**
- Many constructions in language are excessively rare.
- Neural networks rely on iterated trial and error
  $\Rightarrow$ stabilization is driven by common inputs
- When push comes to shove, a neural network will maximize performance over common inputs at the expense of rare inputs.

Moral of the Story

- Deep learning is all the rage now.
- But it is unlikely that it will solve the problem of language learning.

Zipf's Law Strikes Again!

- Neural networks have a bigger problem than cognitive reality: **Zipf's Law!**
- Many constructions in language are excessively rare.
- Neural networks rely on iterated trial and error
  $\Rightarrow$ stabilization is driven by common inputs
- When push comes to shove, a neural network will maximize performance over common inputs at the expense of rare inputs.

Moral of the Story

- Deep learning is all the rage now.
- But it is unlikely that it will solve the problem of language learning.