CSC321

Lecture on Distributed Representations and Coarse Coding

Geoffrey Hinton

Localist representations

- The simplest way to represent things with neural networks is to dedicate one neuron to each thing.
  - Easy to understand.
  - Easy to code by hand
    - Often used to represent inputs to a net
  - Easy to learn
    - This is what mixture models do.
    - Each cluster corresponds to one neuron
  - Easy to associate with other representations or responses.
- But localist models are very inefficient whenever the data has componential structure.

Examples of componential structure

- Big, yellow, Volkswagen
  - Do we have a neuron for this combination
    - Is the BYV neuron set aside in advance?
    - Is it created on the fly?
    - How is it related to the neurons for big and yellow and Volkswagen?
- Consider a visual scene
  - It contains many different objects
  - Each object has many properties like shape, color, size, motion.
  - Objects have spatial relationships to each other.

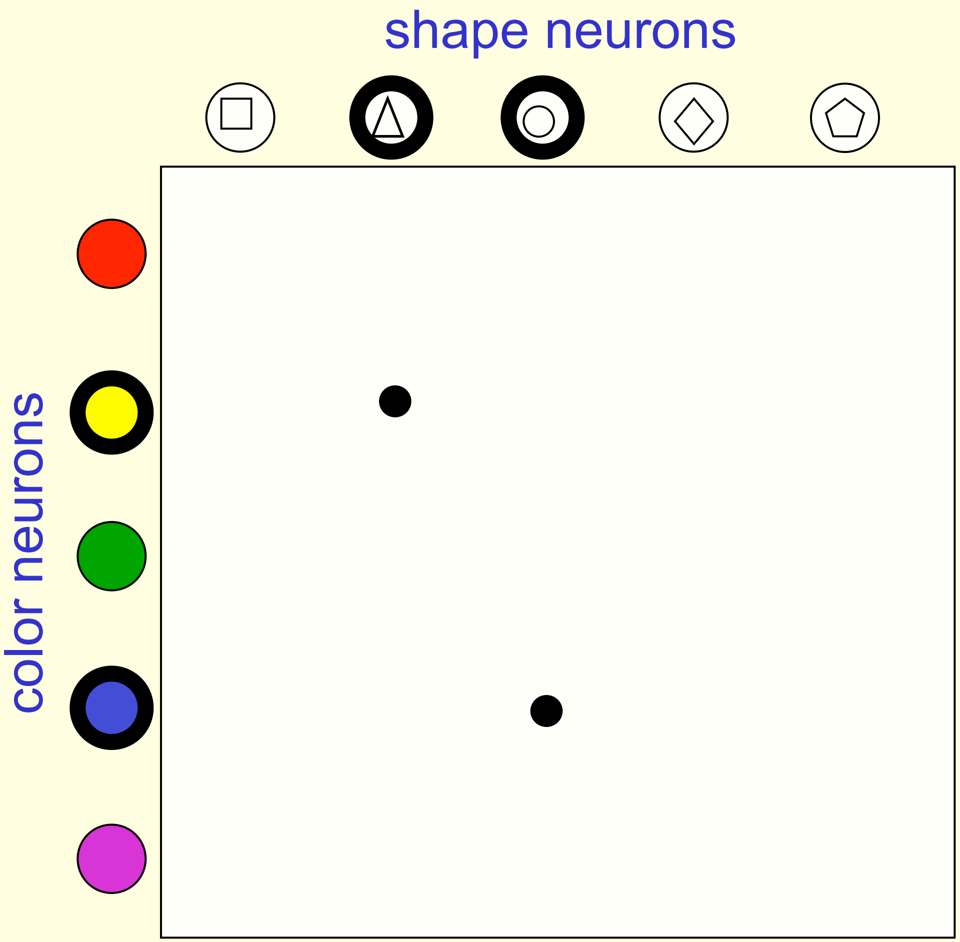
Using simultaneity to bind things together

Represent conjunctions by activating all the constituents at the same time.

- This doesn't require connections between the constituents.
- But what if we want to represent yellow triangle and blue circle at the same time?

Maybe this explains the serial nature of consciousness.

- And maybe it doesn't!



Using space to bind things together

- Conventional computers can bind things together by putting them into neighboring memory locations.
  - This works nicely in vision. Surfaces are generally opaque, so we only get to see one thing at each location in the visual field.
    - If we use topographic maps for different properties, we can assume that properties at the same location belong to the same thing.

The definition of "distributed representation"

- Each neuron must represent something, so this must be a local representation.
- "Distributed representation" means a many-to-many relationship between two types of representation (such as concepts and neurons).
  - Each concept is represented by many neurons
  - Each neuron participates in the representation of many concepts
- Its like saying that an object is "moving".

Coarse coding

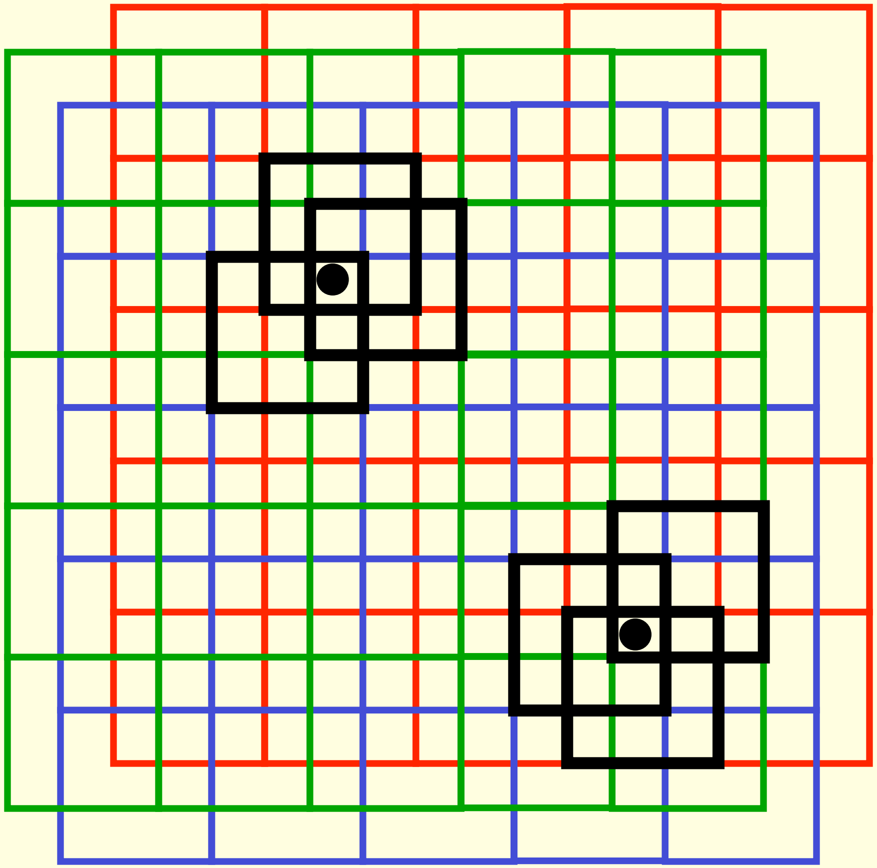
- Using one neuron per entity is inefficient.
  - An efficient code would have each neuron active half the time (assuming binary neurons).
    - This might be inefficient for other purposes (like associating responses with representations).
- Can we get accurate representations by using lots of inaccurate neurons?
  - If we can it would be very robust against hardware failure.

Coarse coding

Use three overlapping arrays of large cells to get an array of fine cells

- If a point falls in a fine cell, code it by activating 3 coarse cells.

- This is more efficient than using a neuron for each fine cell.
  - It loses by needing 3 arrays
  - It wins by a factor of 3x3 per array
  - Overall it wins by a factor of 3



How efficient is coarse coding?

- The efficiency depends on the dimensionality
  - In one dimension coarse coding does not help
  - In 2-D the saving in neurons is proportional to the ratio of the fine radius to the coarse radius.
  - In k dimensions , by increasing the radius by a factor of r we can keep the same accuracy as with fine fields and get a saving of:

$$saving = \frac{\#\,fine\ neurons}{\#\,coarse\ neurons} = r^{k-1}$$

Coarse regions and fine regions use the same surface

- Each binary neuron defines a boundary between k-dimensional points that activate it and points that don't.
  - To get lots of small regions we need a lot of boundary.

fine → $c n r^{k-1}$; coarse → $C N R^{k-1}$

$$total\ boundary = cnr^{k-1} = CNR^{k-1}$$

$$\frac{n}{N} = \left(\frac{C}{c}\right)\left(\frac{R}{r}\right)^{k-1}$$

saving in neurons without loss of accuracy → $\frac{n}{N}$

constant → $\left(\frac{C}{c}\right)$

ratio of radii of fine and coarse fields → $\left(\frac{R}{r}\right)^{k-1}$

Limitations of coarse coding

- It achieves accuracy at the cost of resolution
  - Accuracy is defined by how much a point must be moved before the representation changes.
  - Resolution is defined by how close points can be and still be distinguished in the represention.
    - Representations can overlap and still be decoded if we allow integer activities of more than 1.
- It makes it difficult to associate very different responses with similar points, because their representations overlap
  - This is useful for generalization.
- The boundary effects dominate when the fields are very big.

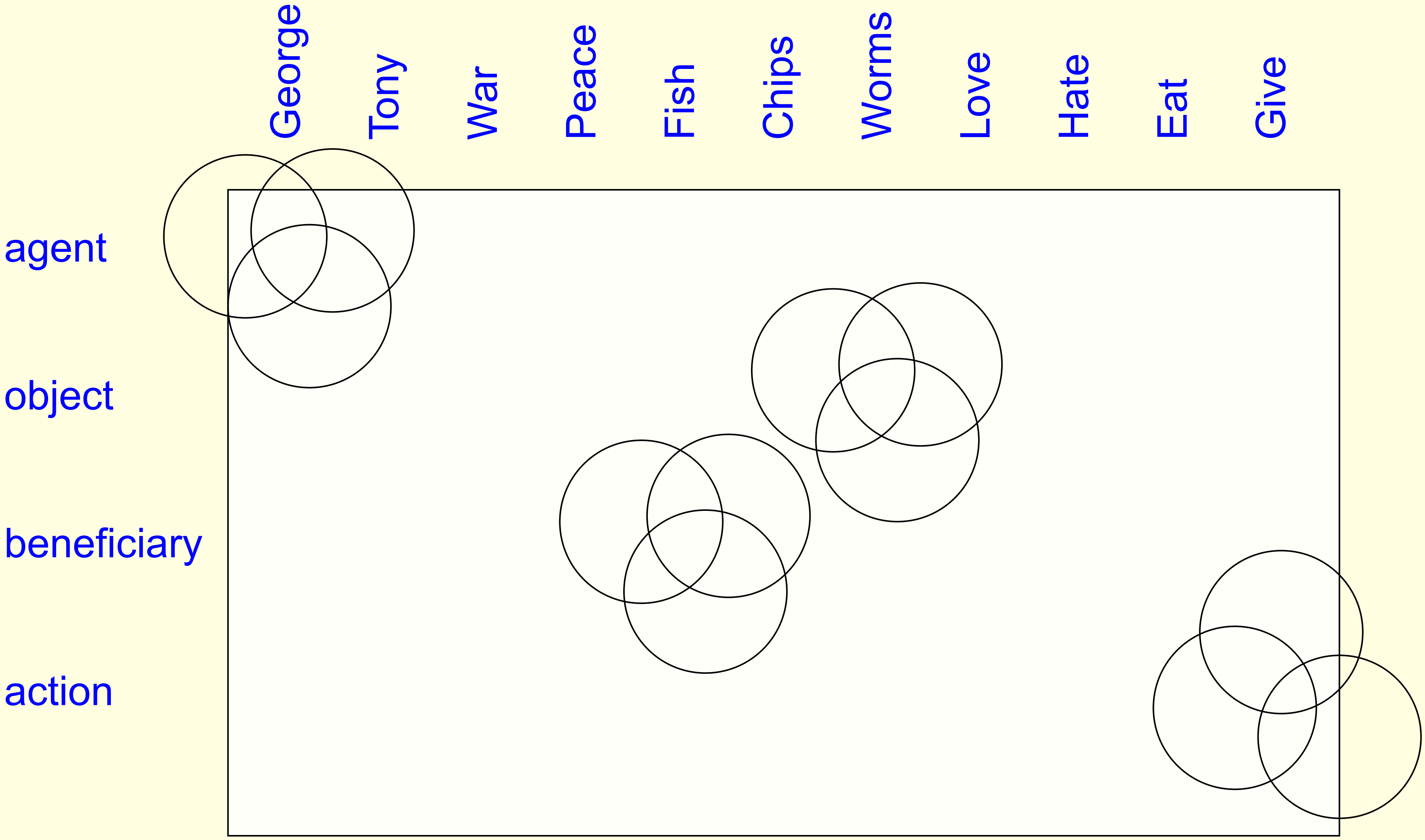
Coarse coding in the visual system

- As we get further from the retina the receptive fields of neurons get bigger and bigger and require more complicated patterns.
  - Most neuroscientists interpret this as neurons exhibiting invariance.
  - But its also just what would be needed if neurons wanted to achieve high accuracy
  - For properties like position orientation and size.
- High accuracy is needed to decide if the parts of an object are in the right spatial relationship to each other.

Representing relational structure

- "George loves Peace"
  - How can a proposition be represented as a distributed pattern of activity?
  - How are neurons representing different propositions related to each other and to the terms in the proposition?
- We need to represent the role of each term in proposition.

A way to represent structures



The recursion problem

- Jacques was annoyed that Tony helped George
  - One proposition can be part of another proposition. How can we do this with neurons?
- One possibility is to use "reduced descriptions". In addition to having a full representation as a pattern distributed over a large number of neurons, an entity may have a much more compact representation that can be part of a larger entity.
  - It's a bit like pointers.
  - We have the full representation for the object of attention and reduced representations for its constituents.
  - This theory requires mechanisms for compressing full representations into reduced ones and expanding reduced descriptions into full ones.