Title: How to represent part-whole hierarchies in a neural net

Speakers: Geoffrey Hinton

Series: Colloquium

Date: May 05, 2021 - 2:00 PM

URL: http://pirsa.org/21050001

Abstract: I will present a single idea about representation which allows several recent advances in neural networks to be combined into an imaginary system called GLOM. GLOM answers the question: How can a neural network with a fixed architecture parse an image into a part-whole hierarchy which has a different structure for each image? The idea is simply to use islands of identical vectors to represent the nodes in the parse tree. The talk will discuss the many ramifications of this idea. If GLOM can be made to work, it should significantly improve the interpretability of the representations produced by neural nets when applied to vision or language.

How to represent part-whole hierarchies in a neural network

Geoffrey Hinton

Google Research
&
The Vector Institute

Overview of talk: Combining three recent advances in neural networks

- Transformers.
- Unsupervised learning of visual representations via contrastive agreement.
- Generative models of images that use neural fields.
- I will combine these three advances to create an imaginary vision system called GLOM that is much more like human perception than current deep nets.

Two goals of neural network research

- Engineering: Most researchers are just trying to design neural nets that work better.
  - Anything that works is allowed.
  - 100 layers is OK. Weight-sharing is OK

- Science: Some researchers investigate neural nets in order to understand how the brain works.
  - We can still learn a lot from the brain.
  - For half a century, the brain was the only reason for believing that big neural nets that learn almost everything from data would ever work.

Some sources of information about how our visual system actually works

- Psychological demonstrations of how it goes wrong.
- Weird effects of brain damage.

- The anatomy of neo-cortex.
- The responses of cortical neurons to stimuli.
- The properties of synapses.

- Computer simulations that show what works and what doesn't.

Why it is hard to make real neural networks learn part-whole hierarchies

- Each image has a different parse tree.
- Real neural networks cannot dynamically allocate neurons to represent nodes in a parse tree.
  - What a neuron does is determined by the weights on its connections and the weights change slowly.
- So how can static neural nets represent dynamic parse trees?
  - I will combine three recent advances to propose an answer to this question.

Ways to represent part-whole hierarchies

- Symbolic AI: For each image, dynamically create a graph in which a node for a whole is connected to nodes for its parts.

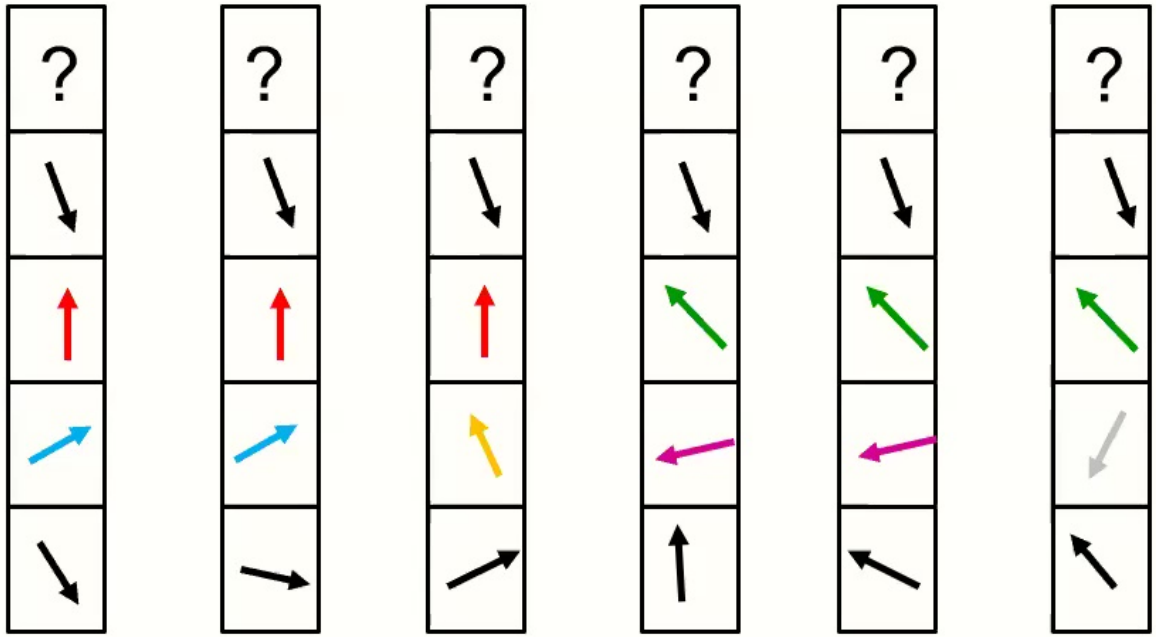
- Capsules: Permanently allocate a piece of neural hardware for each *possible* node. For each image, activate a small subset of the possible nodes and use dynamic routing to activate connections between whole and part nodes.

A novel way to represent part-whole hierarchies

- GLOM: Allocate hardware to columns. Each column contains multiple levels of representation of what is happening in a small patch of the image.
  - A single patch might be represented by embeddings for a nostril, a nose, a face, a person, a party.

- Use islands of agreement to represent the nodes in the parse tree.
  - All the locations that are occupied by the same face should have exactly the same embedding vector at the object level.
  - The embedding vectors act like pointers.

The embedding vectors for nearby columns at a single time-step as GLOM settles



At each level there are islands of agreement. These islands represent the parse tree for the scene.

It is a multi-level, real-valued Ising model with coordinate transforms between levels.

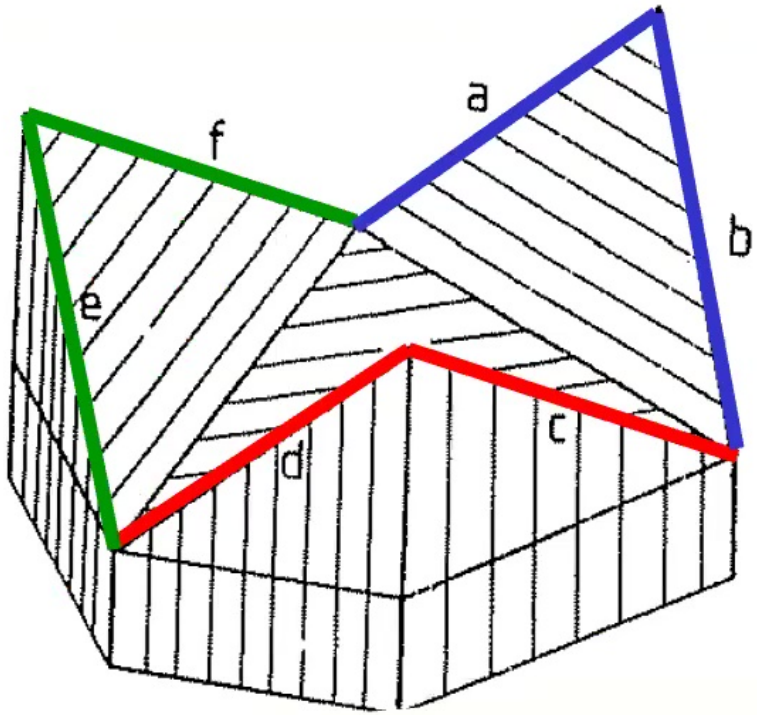
The psychological reality of the part-whole hierarchy and coordinate frames

- The next few slides demonstrate the psychological reality of part-whole hierarchies in vision.
- They also demonstrate the psychological reality of rectangular coordinate frames in human vision.

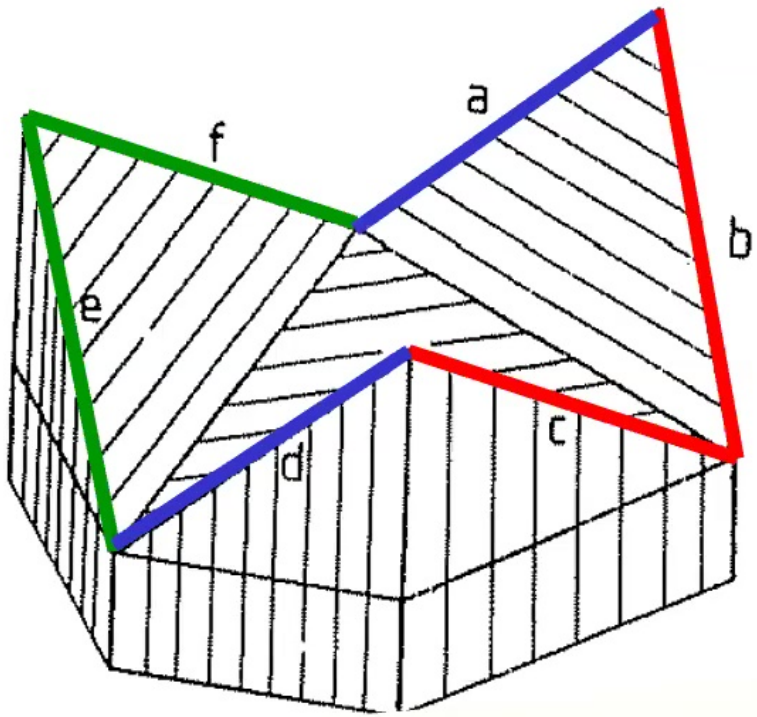
The cube demonstration (Hinton, 1979)

- Imagine a wire-frame cube resting on a table-top.
- Imagine the body diagonal that goes from the front bottom right corner, through the center of the cube to the top back left corner.
- Keeping the front bottom right corner on the table top, move the top back left corner until it is vertically above the front bottom right corner.
- Hold one finger-tip above the table to mark the top corner. With the other hand, point out the other corners of the cube.

An arrangement of 6 rods



A different percept of the 6 rods

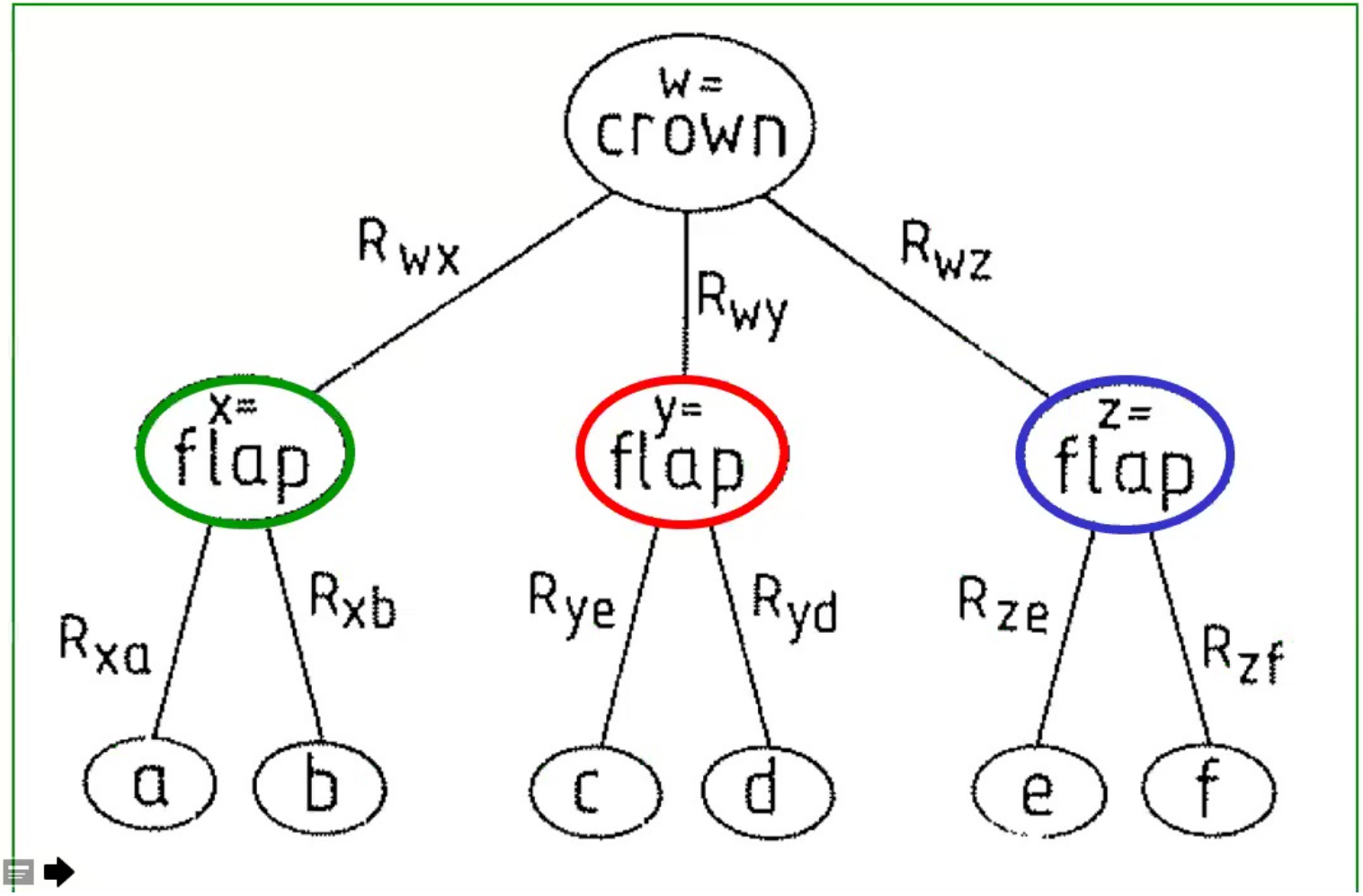


Alternative representations (unlike CNNs)

- The very same arrangement of rods can be represented in quite different ways.
  - Its not like the Necker cube where the alternative percepts disagree on depth.

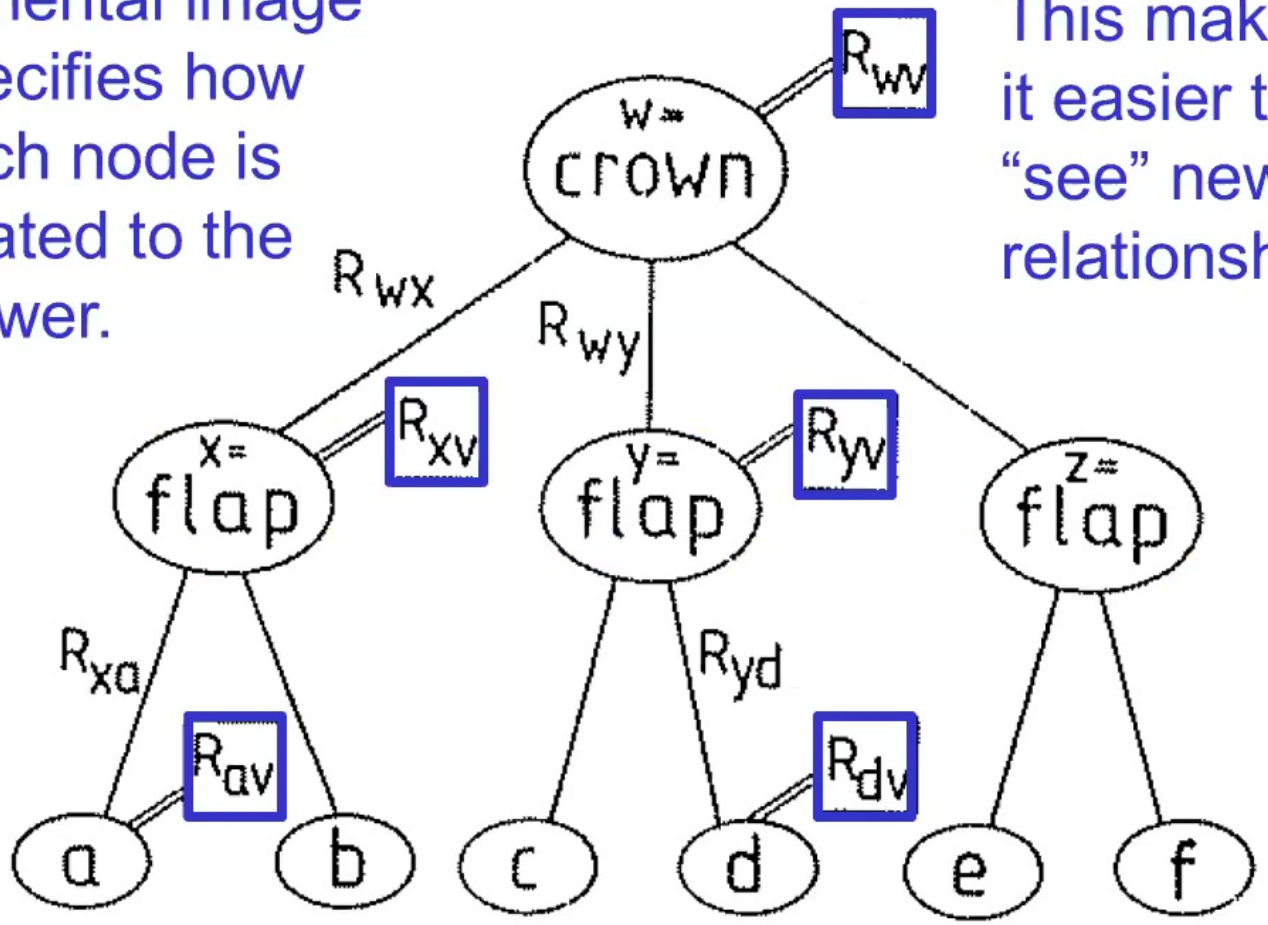
- The alternative percepts do not disagree, but they make different facts obvious.
  - In the zig-zag representation it is obvious that there is one pair of parallel edges.
  - In the crown representation there are no obvious pairs of parallel edges because the edges do not align with the intrinsic frame of any of the parts.

A structural description of the “crown” formed by the six rods



A mental image of the crown

A mental image specifies how each node is related to the viewer.



This makes it easier to "see" new relationships

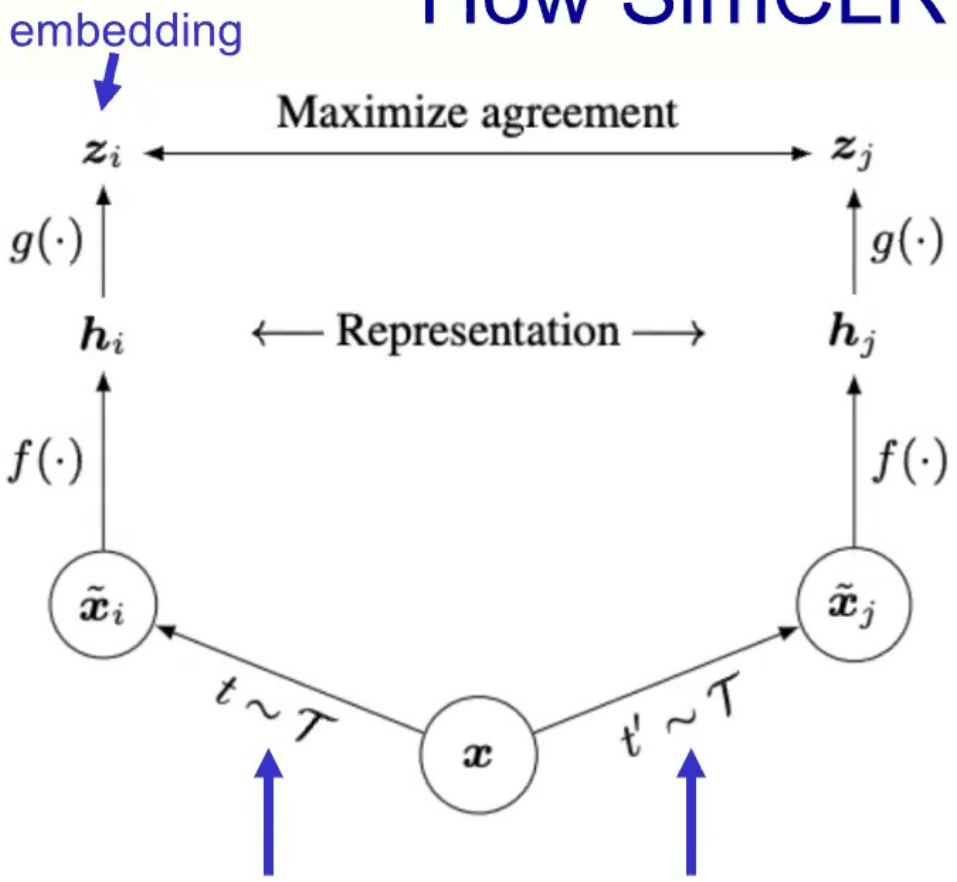
A brief introduction to contrastive learning of visual representations

- Contrastive self-supervised learning uses the similarity between activity vectors produced from different patches of the same image as the objective.

- Many different groups have developed contrastive, self-supervised learning since Becker and Hinton introduced one version of the idea in 1992.

- I will only mention one model called SimCLR developed in Toronto.

How SimCLR works



Different crops and color distortions of the same image.

Minimize the differences between embeddings of patches from the same image.

Maximize the differences between similar embeddings of patches from different images.

How good are the representations found by SimCLR?

- After unsupervised learning, take the layer before the learned embeddings and fit a linear classifier (i.e a softmax).
  - The linear classifier does very well.

A problem with contrastive learning of visual representations

- It works, but it is not intuitively satisfying.
  - What if one patch in an image contains parts of objects A and B, and the other patch contains parts of objects A and C.
    - Do we really want to get the same output vector for both patches?
- GLOM is designed to overcome this problem by using attention.

Spatial coherence

- The original motivation for using agreement of the output vectors from different patches as an objective function was not classification.
  - The aim was to find properties that are coherent across space or time (Becker and Hinton, 1992).

- GLOM is a new way of discovering spatial coherence that relies on performing coordinate transforms to make very different parts like a nose and a mouth very similar at the next level up.

A Biological Inspiration

- Every cell has a complete set of instructions for making proteins.
- The environment of the cell determines which proteins are actually expressed.
  - So cells all have the same knowledge, but differ in their vector of protein expressions.
  - Those vectors are similar within an organ.
- It seems wasteful to duplicate all of the knowledge in every cell, but it is very convenient.

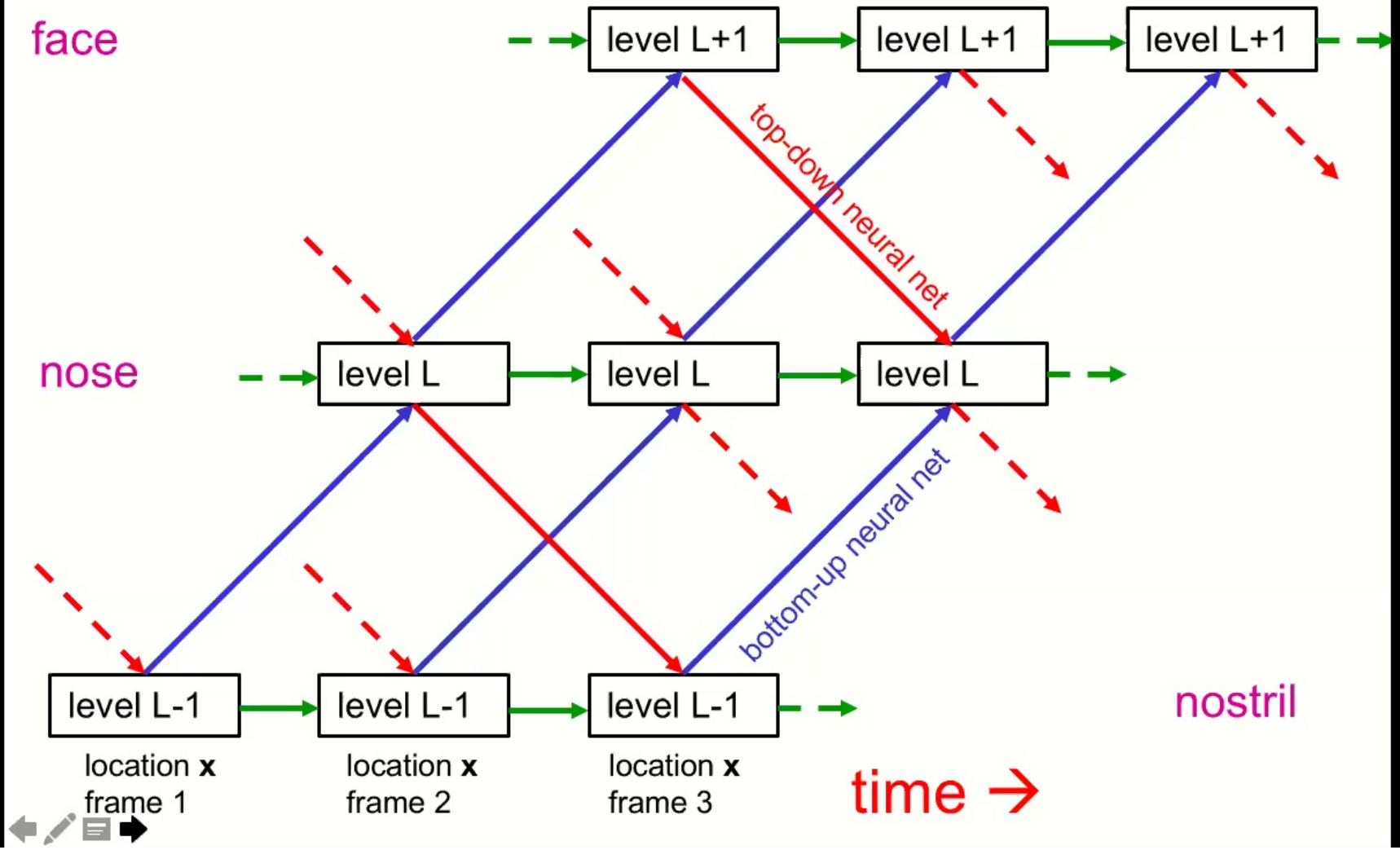
The analogy with vision

- A column of hardware that is dedicated to what is happening at one image location is like a cell.
- The complete vector of embeddings at multiple levels in a column is like the vector of protein expressions in a cell.
  - Objects are like organs. Organs are collections of cells with similar gene expression vectors.

Disclaimer

- The outer loop of vision is a sequence of intelligently chosen fixations that sample the optic array to provide the information required to perform a task.
- For each fixation we reuse the same neural net to produce a multi-level representation of the retinal image produced by that fixation.
- This talk is only about what happens on the first fixation.

Three adjacent levels of GLOM in a single column



Interactions between and within levels

- The level L embedding at location x is an average of four contributions:

1. The bottom-up contribution from the level L-1 embedding in the same column at the previous time-step.

2. The top-down contribution from the level L+1 embedding in the same column at the previous time-step.

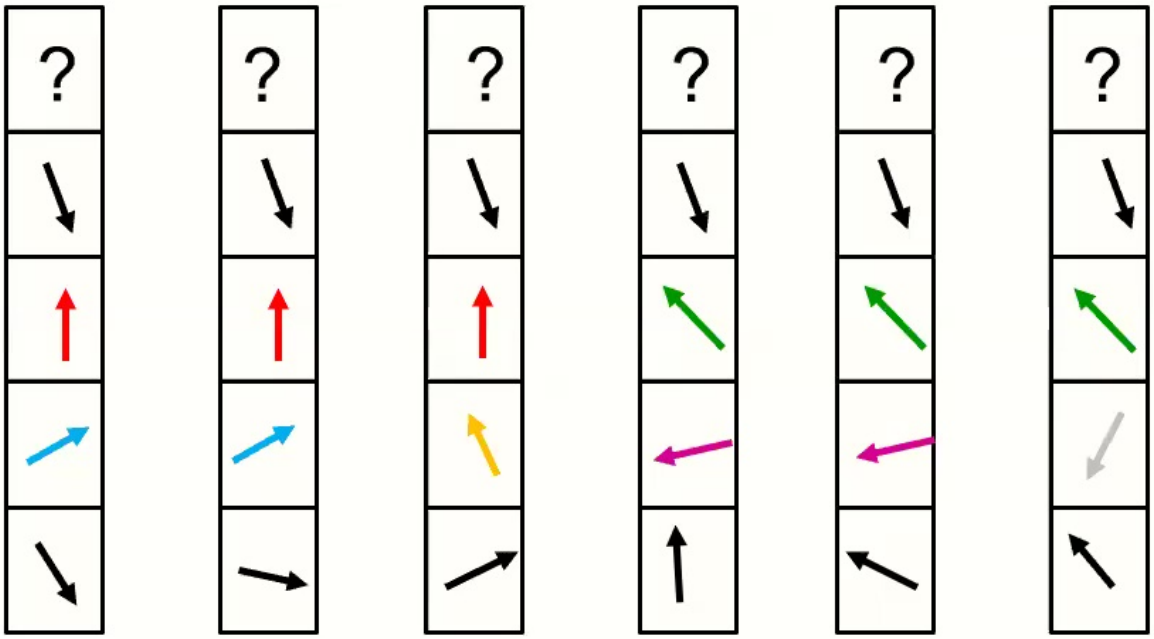
3. The level L embedding at the previous time-step.

4. The attention-weighted average of the level L embeddings in other nearby columns at the previous time step.

The attention-weighted average

- The level L embedding at location **x** tries to agree with similar level L embeddings at other locations.
  - The attention weighted average of the level L embeddings at other locations, **y**, uses weights proportional to exp[L(**x**) . L(**y**)]
  - This causes the level L embeddings to form islands of similar embeddings.
    - Islands are echo chambers.

The embedding vectors for a row of locations at a single time-step as GLOM settles



At each level there are islands of agreement. These islands represent the parse tree for the scene.

A problem with making an object vector the same at all the locations occupied by the object

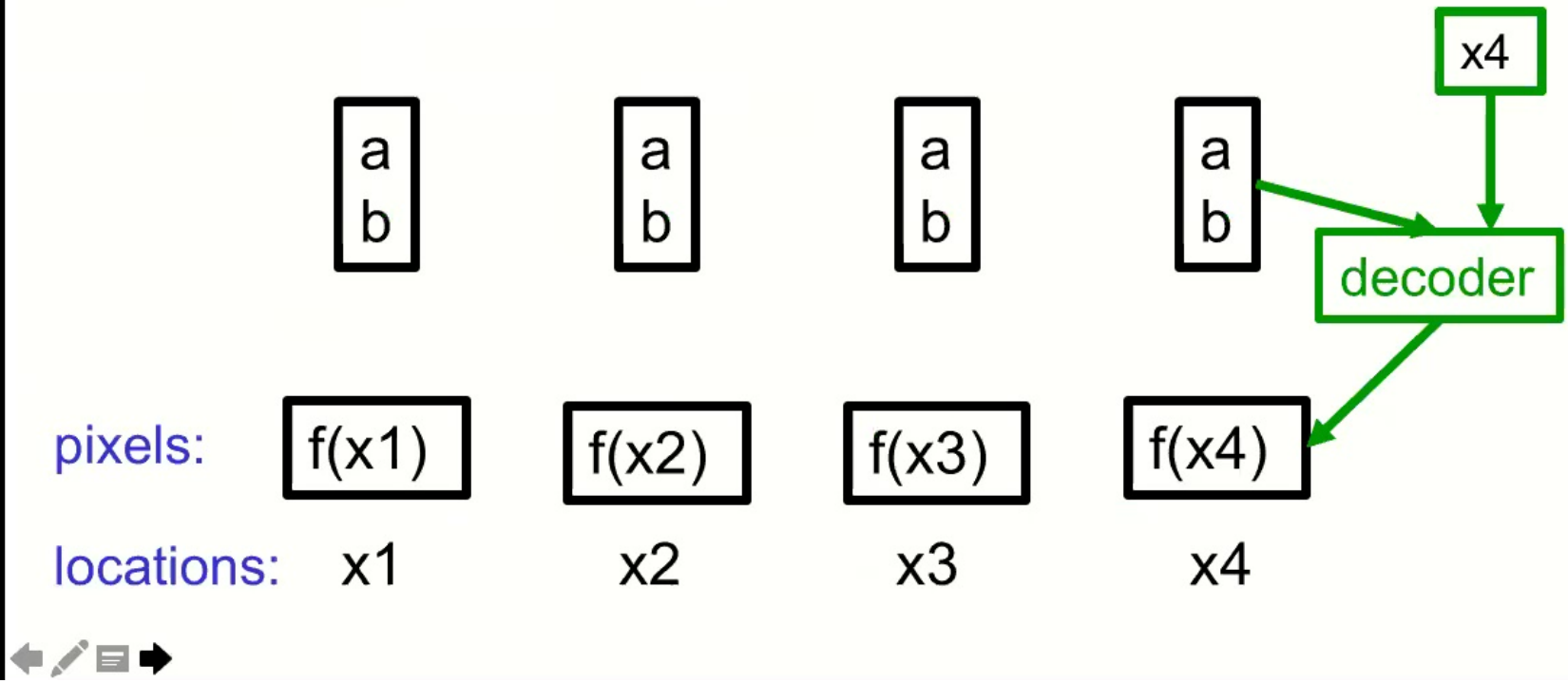
- When a face vector makes top-down predictions for the parts of the face, how can the same face vector make different predictions for locations occupied by the nose and locations occupied by the mouth?

- The answer is to use hierarchical neural fields.
  - Instead of predicting a whole image from a code vector, a neural field predicts one small location of the image when given the code vector and a representation of the coordinates of the location.

A very simple example of a neural field decoder

- Suppose we have a row of pixels in which the intensity increases linearly along the row as in $f(x) = ax + b$

- We can give every pixel an identical code.



Top-down prediction of the parts of a face

- The object level embedding vector for a face contains viewpoint information about the spatial relationship between the intrinsic coordinate frame of the face and the coordinate frame of the camera or retina.
- Given the coordinates of a location in the image, the top-down neural net can compute where that image location is within the intrinsic coordinate frame of the face.
  - So the top-down net can compute which part goes at that image location.
  - This allows it to predict the nose vector for locations within the nose and the mouth vector for locations within the mouth.

One way to deal with ambiguous parts: disambiguation at the part level

- A possible nose could interact directly with a possible mouth.
- They disambiguate each other if they have the right spatial relationship.
- So we need a “transformational random field” in which the pose of the nose predicts the pose of the mouth via a nose->mouth coordinate transform (and vice versa).
- N interacting parts need O(N^2) coordinate transforms between parts.

A different way to deal with ambiguous parts: The Hough transform

- Instead of allowing the parts to interact directly, allow each part to make an ambiguous multimodal prediction for the identity and pose of the whole object in the same column.
  - Unlike capsules, no dynamic routing is required.
- The whole is present if many multimodal predictions from different columns can agree on a mode.

- If each column predicts an unnormalized log probability distribution over the space of possible object instances, we can simply add the predicted distributions in different columns.
  - But we should only add similar distributions.

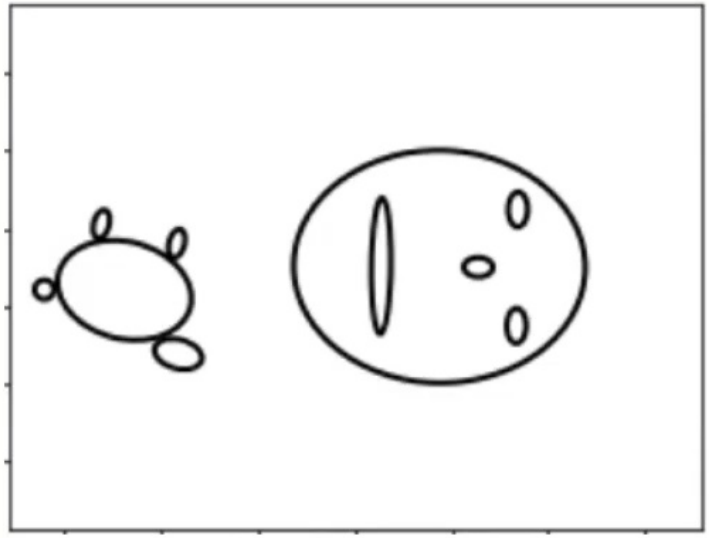
How to implement multimodal predictions in the joint space of identity and pose

- Each neuron in the embedding vector for the object is a basis function that represents a vague distribution in the log probability space.
- The activity of the neuron scales this log distribution.
- The full object embedding vector represents the sum of these scaled log distributions.
- The individual distributions can be very vague because they only need to represent one thing at a time: the object occupying that location.

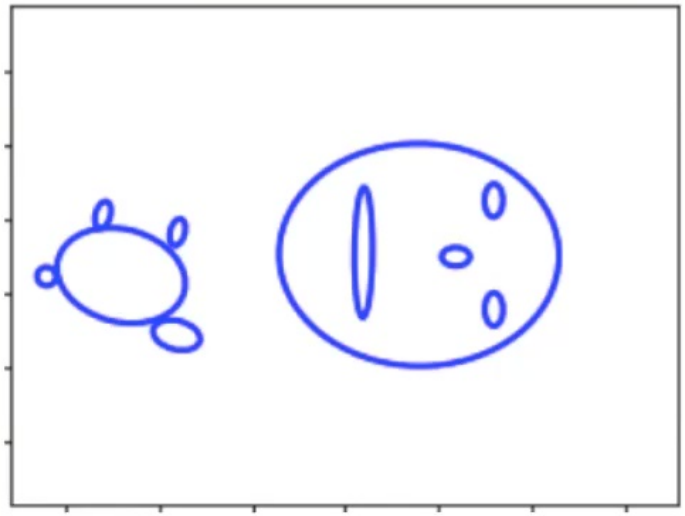
A simple test of GLOM's way of combining multimodal predictions

(by Laura Culp)

The data: 10 ellipses

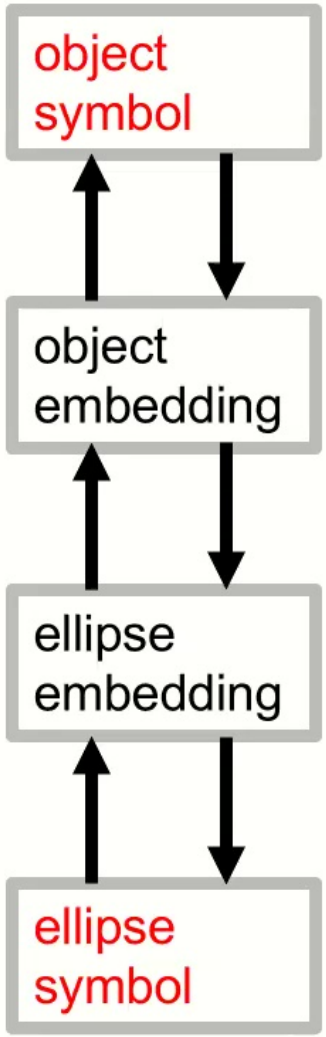


reconstructions of the ellipses from the two final object vectors



A column of GLOM with only 2 embedding levels

6 linear units plus class softmax



Hundreds of basis functions in the log probability space

Hundreds of basis functions in the log probability space

6 linear units

The initial object embedding represents uncertainty across 2 identities and 5 poses

As the network iterates, the ellipse embedding becomes more specific and stops predicting bad modes.

Deep end-to-end training

- Given an image with missing regions at the input, GLOM could be trained to predict the uncorrupted image at its final time-step.
  - This is how BERT is trained to learn good embeddings for word fragments.

- But this objective function alone will not encourage the embeddings to form islands of very similar vectors at different locations.
  - That is where contrastive learning becomes relevant.

An extra term to make the bottom-up and top-down neural nets produce islands of similar predictions

- Each neural net makes predictions for the embeddings at the level above and the level below at the next time-step.
- The actual “consensus” embedding is a weighted average of the predictions from above and below in the same column plus the attention-weighted average of the same level embeddings in other columns.
- If we train the predictions to agree with the consensus, we will increase the agreement between embeddings that are similar.

How can columns share weights in a brain?

- In GLOM, the bottom-up and top-down neural nets between two adjacent levels are the same for all columns.
  - But a brain cannot share weights.

- All we actually need to share is the **knowledge** in different columns.
  - We can do this by co-distillation.
  - Each net provides training signals for the other nets via the attentional interactions between columns.
  - Attention-gated averaging between columns implements CNN's in a brain!

Isn't it wasteful to replicate the object-level embedding vector for every location in an object?

- After the forward pass has settled on how to bind locations to object instances, it seems very wasteful to replicate the object-level embedding vectors for every location.

- But during the search for how to segment the locations into objects, it is very helpful to have a separate object-level embedding vector for each location.
  - It aloows each location to hedge its bets about which other locations it goes with.
  - Similar embedding vectors for different locations can support each other. This should create clusters better and faster than mixtures of Gaussians can discover them.

Replicating object embeddings for every location is less expensive than you might think

- The longer range interactions in an image should be between higher-level embeddings of locations.
- It is fine to only sample these embeddings sparsely because there will be big islands of almost identical higher-level embeddings.
  - This kind of sampling is already used in transformers for language processing.

Summary

- I showed how to combine three important advances in neural networks in a system called GLOM:
  - A simplified version of transformers; contrastive representation learning; neural fields.

- GLOM solves the problem of how to represent parse trees in a neural net without doing dynamic allocation of neurons to nodes in the parse tree.

THE END

long paper about this talk at
arxiv:2102.12627