How to represent part-whole hierarchies in a neural network

Geoffrey Hinton

Google Research
&
The Vector Institute
&
University of Toronto

Three recent advances in neural networks

- Transformers for modeling natural language.
- Unsupervised learning of visual representations via agreement.
- Generative models of images that use implicit functions.
- 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.

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

- The next seven 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)

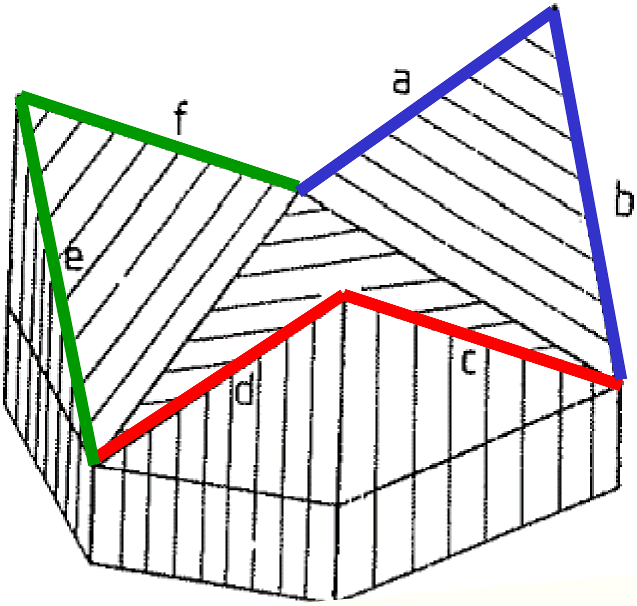
- Imagine a wire-frame cube resting on a table-top.

- Imagine the body diagonal that goes from the front bottom right corner, through the center of the cube to the top back left corner.

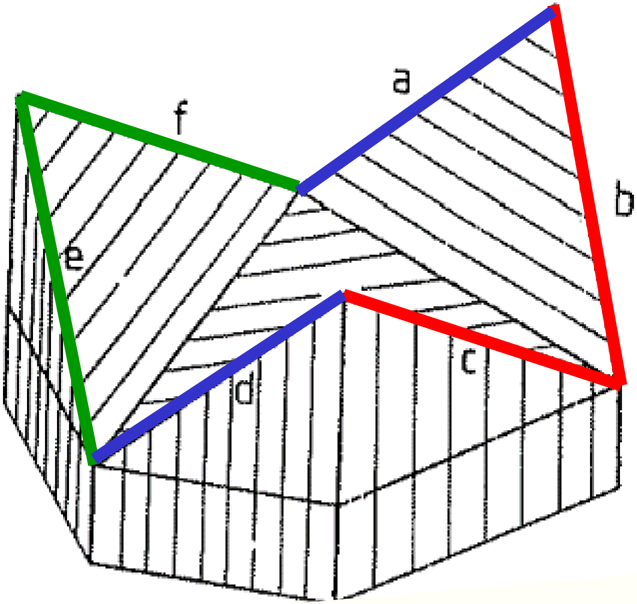
- Keeping the front bottom right corner on the table top, move the top back left corner until it is vertically above the front bottom right corner.

- 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



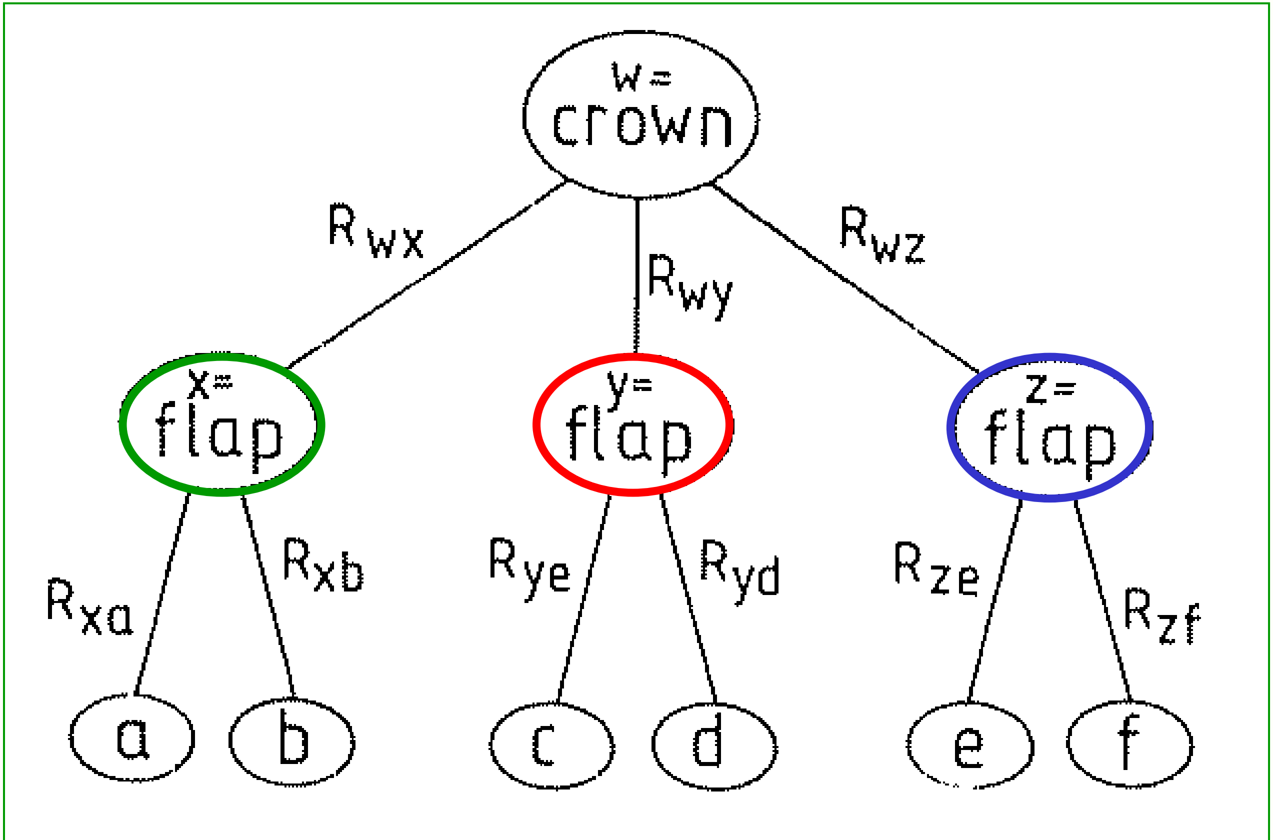
A different percept of the 6 rods



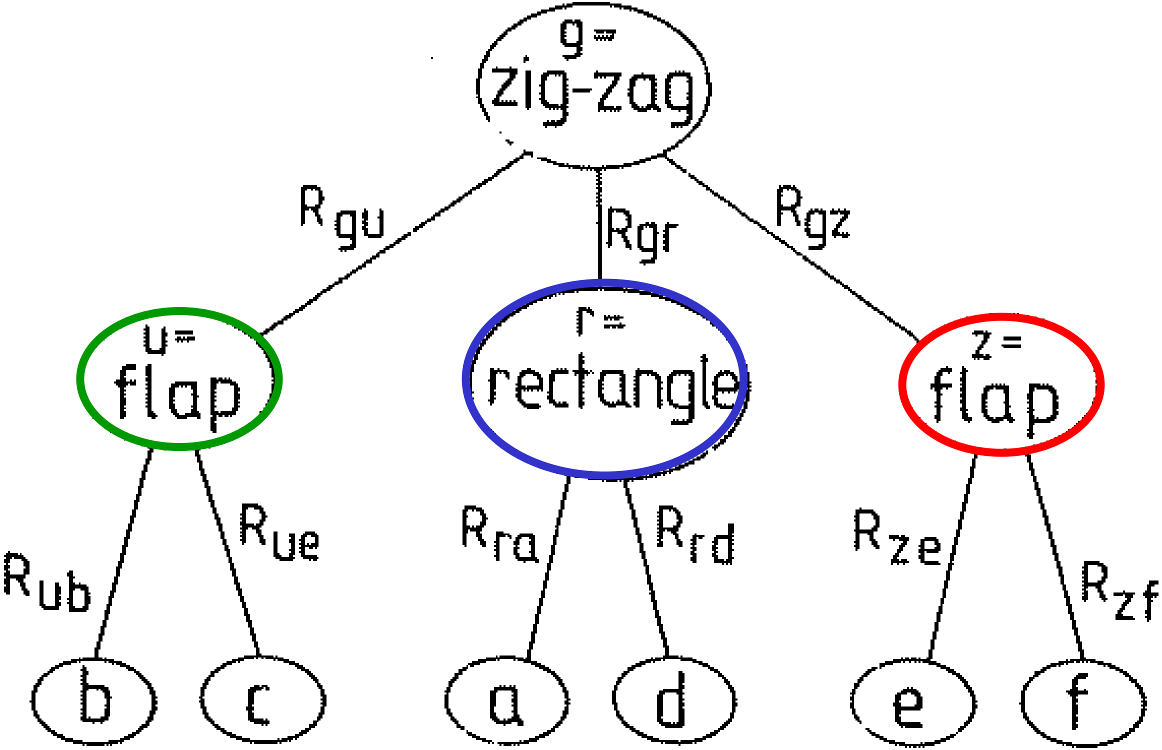
Alternative representations

- 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

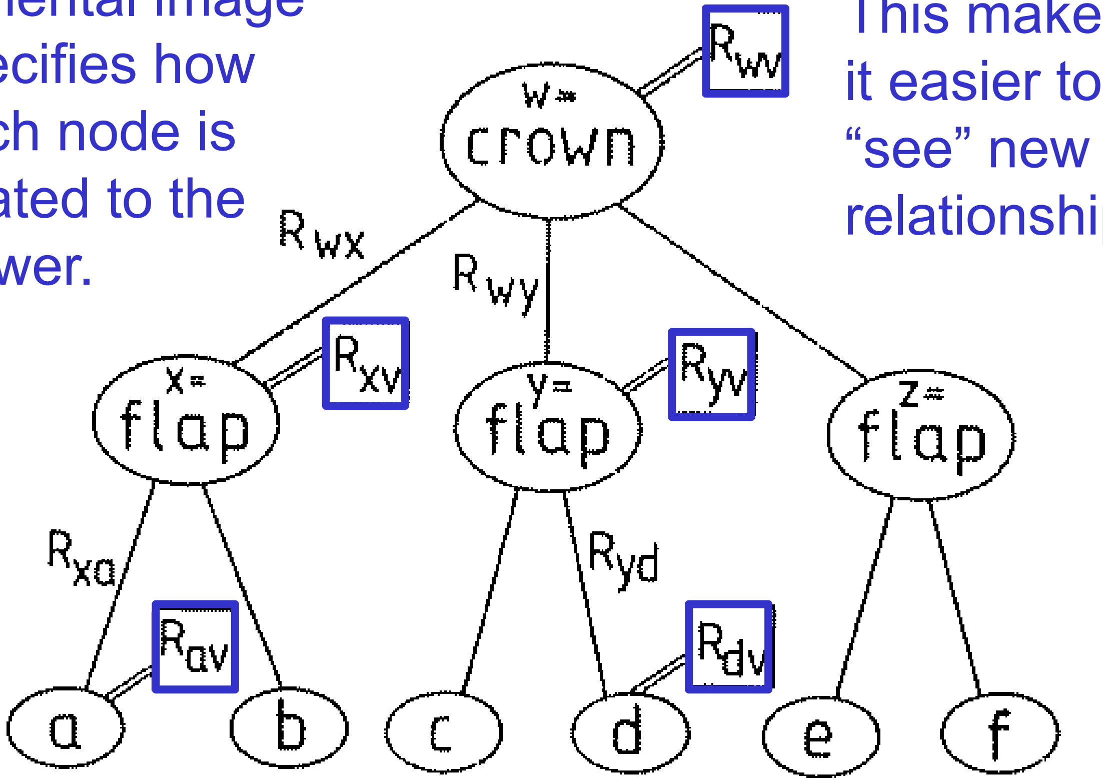


A structural description of the “zig-zag”



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

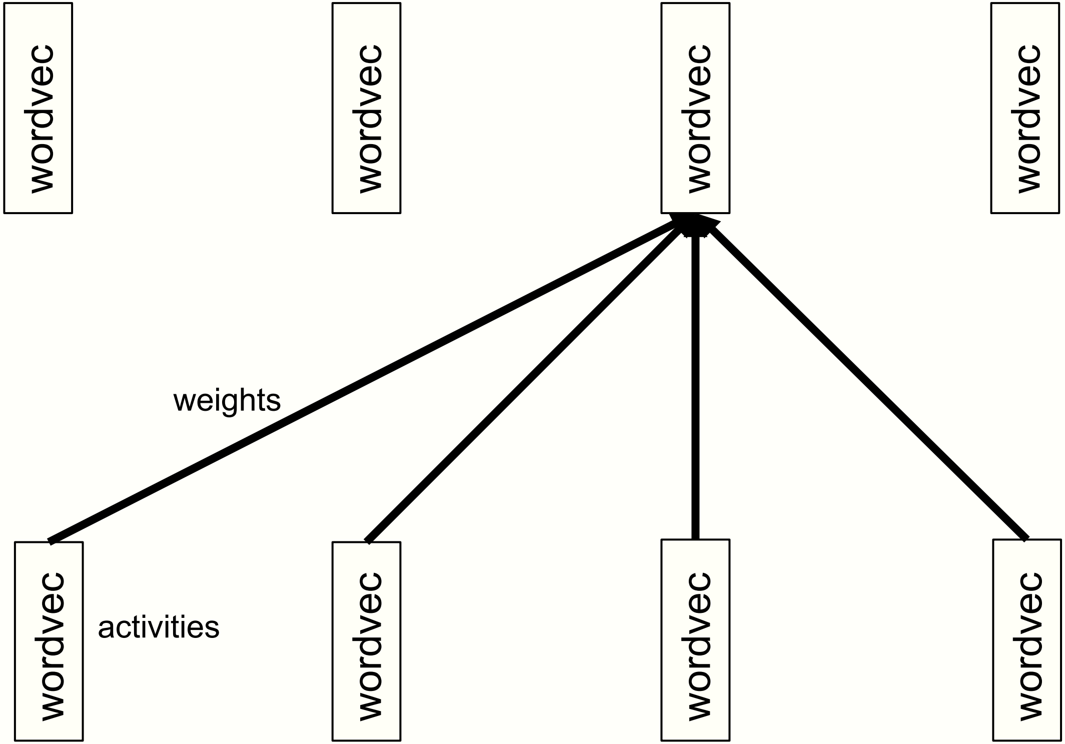
Why it is hard to make real neural networks learn part-whole hierarchies

- 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.

A brief introduction to transformers

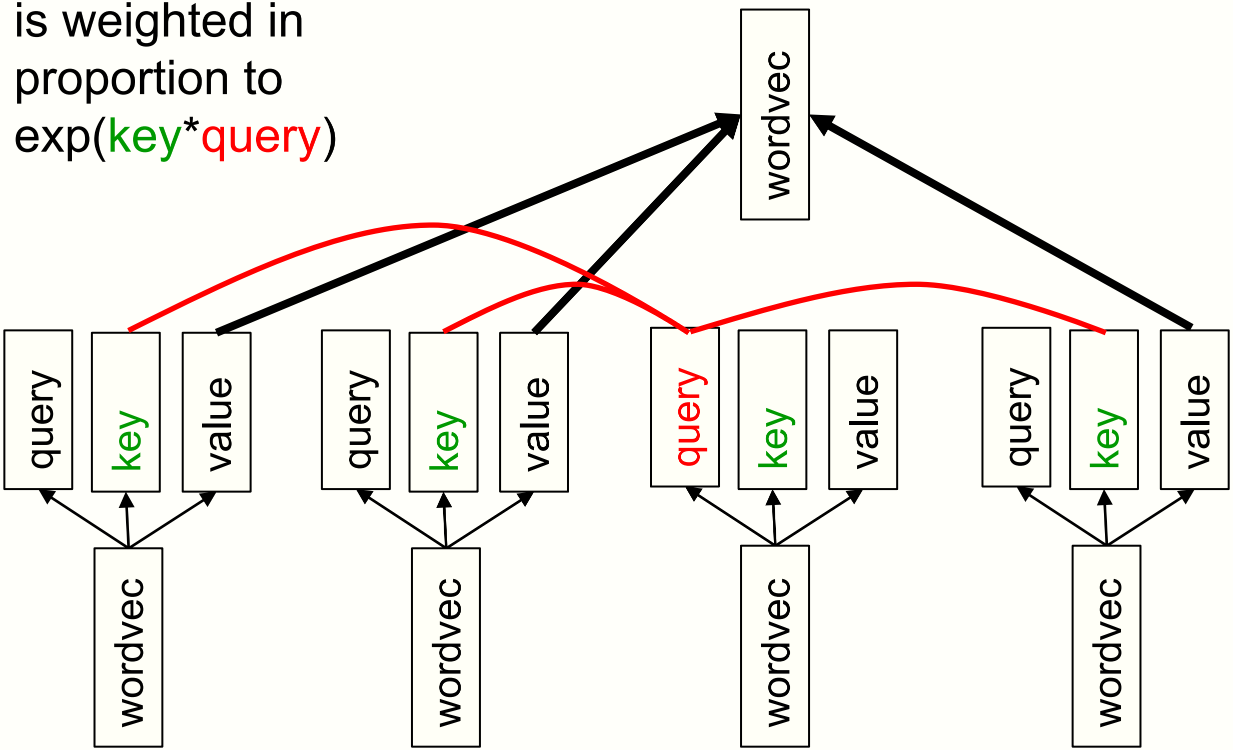
- Attention models (including transformers) make activations depend on the pairwise similarities between *activity* vectors.
  - This contrasts with earlier neural nets that only made activations depend of the similarity between an activity vector and a weight vector.

Standard convolutional neural network for refining word representations based on their context



How transformers work (roughly)

Each value vector is weighted in proportion to exp(key*query)



A brief introduction to contrastive learning of visual representations

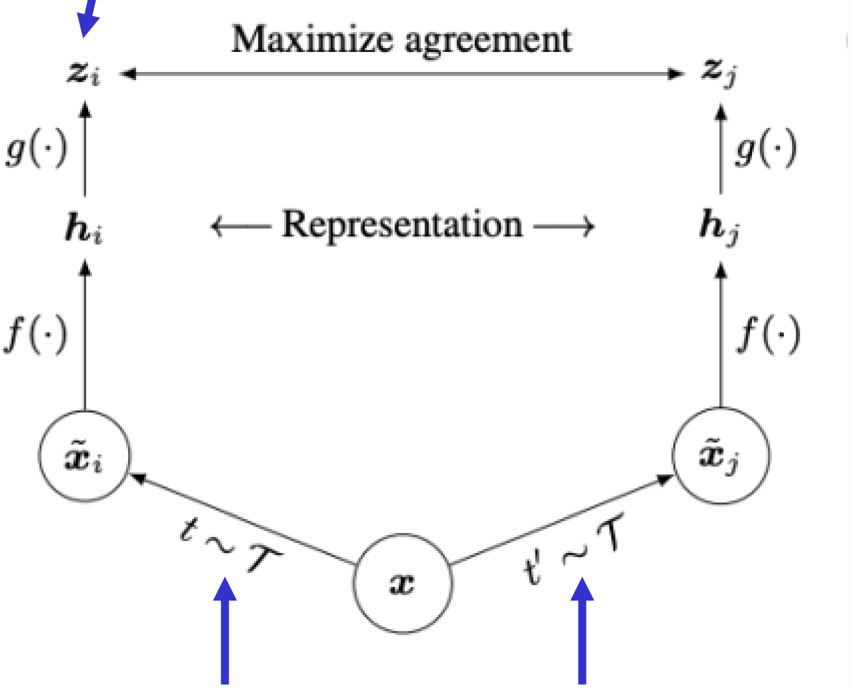
- Contrastive self-supervised learning uses the similarity between activity vectors produced from different patches of the same image as the objective.

- 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

embedding



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 of class A and B, and the other patch contains parts of objects of class A and C.
        - Do we really want to get the same output vector for both patches?

- GLOM is designed to overcome this problem.

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 a novel way of representing the part-whole hierarchy in a neural net.

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.

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.

- GLOM: Use islands of agreement to represent nodes in the parse tree.

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 differ in their vector of protein expressions. The vectors are similar within an organ.

- It seems wasteful to duplicate all of the knowledge in every cell, but it is very convenient.

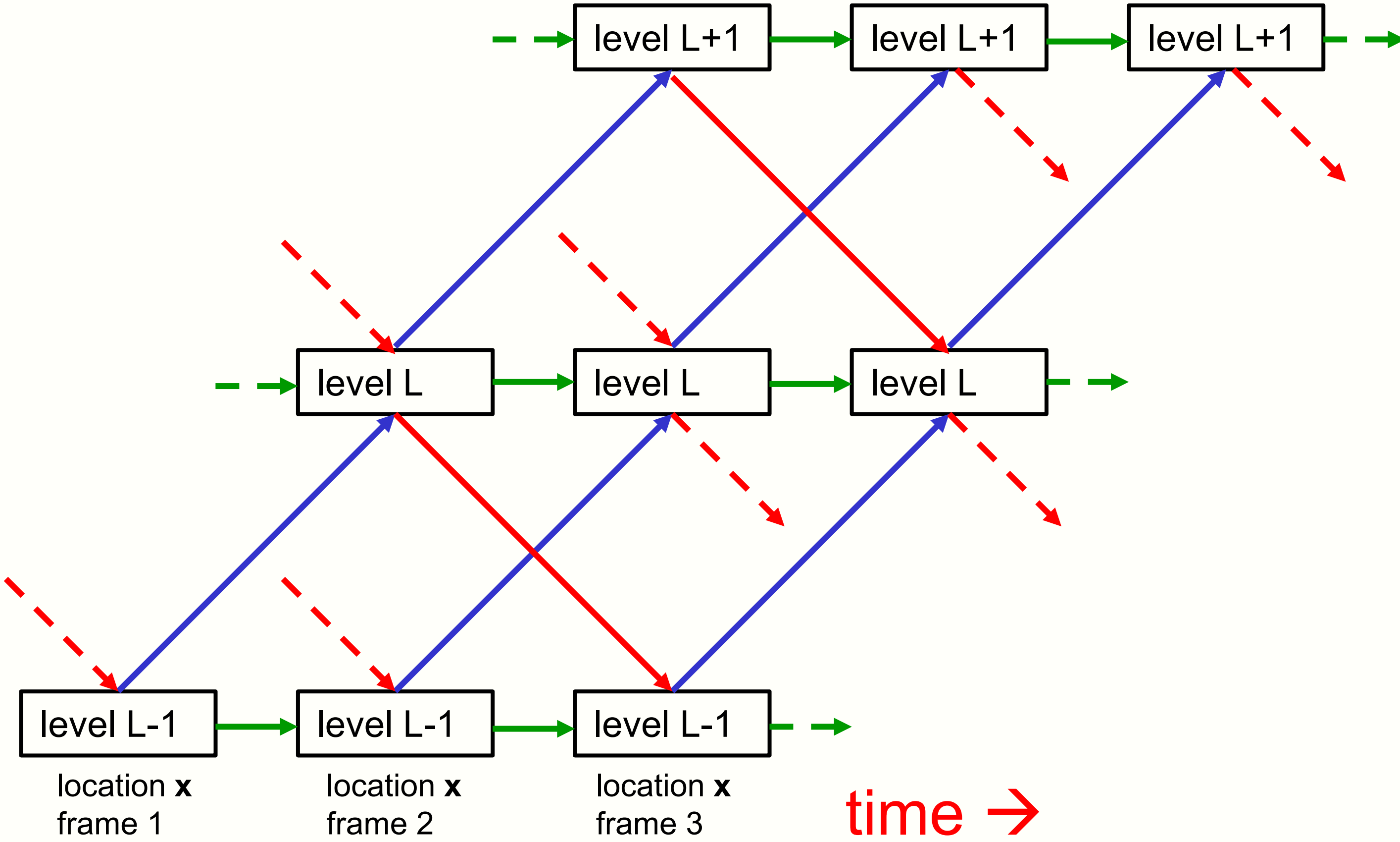
The analogy with vision

- Image locations are like cells.

- Weights are like DNA
  - In a convolutional neural net, the weights are duplicated at every location.

- The complete vector of neural activities centered on a location is like the vector of protein expressions in a cell.
  - Objects are like organs. They are collectons of cells with similar gene expression vectors.

Three adjacent levels of GLOM for one location



Interactions between and within levels

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

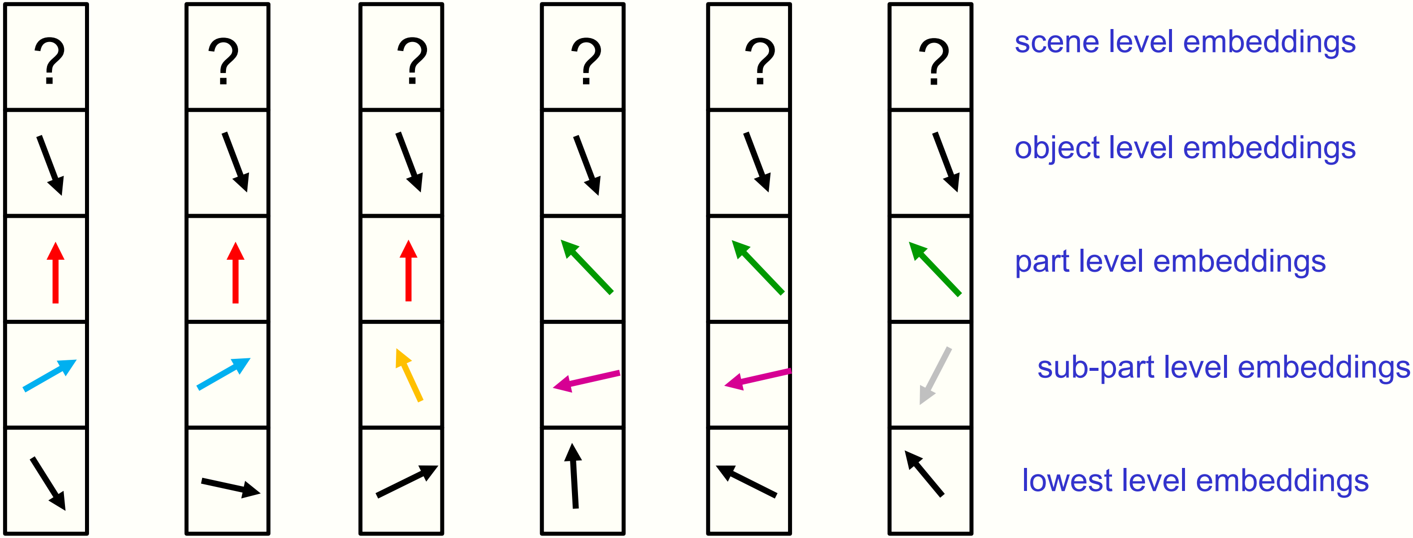
1. The bottom-up contribution from the level L-1 embedding at location x in the previous layer.

2. The top-down contribution from the level L+1 embedding at location x in the previous layer.

3. The attention-weighted average of the level L embeddings at other locations in the previous layer.

4. The previous embedding.

The embedding vectors for a row of locations in a single mid-level layer of GLOM



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

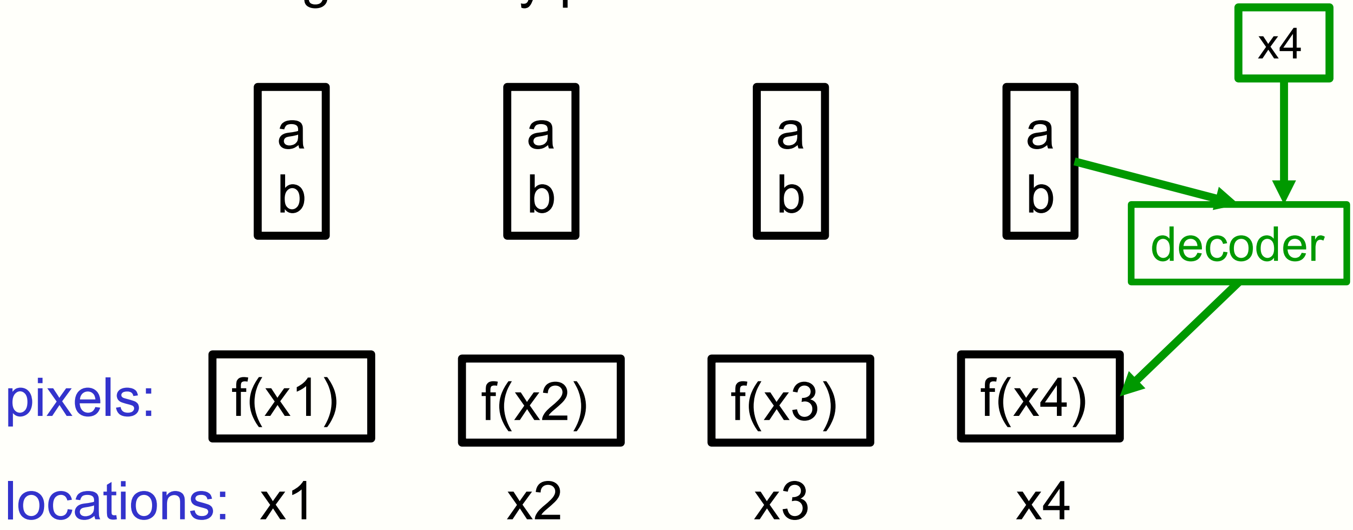
A problem with making an object vector the same at all locations in the object

- 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 in the nose and locations in the mouth?

- The answer is to use implicit functions.
  - Instead of predicting a whole image from a code vector, an implicit function 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 an implicit function 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 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.

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.

Deep end-to-end training

- Given an image with missing regions at the input, GLOM is trained to predict the uncorrupted image at its output.
  - This is how BERT is trained to learn good embeddings for word fragments.
- But this objective function alone will not make the embeddings form islands of similar embeddings 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 a prediction for an embedding at an adjacent level in the next layer.

- The actual “consensus” embedding is a weighted average of two predictions from adjacent levels at the same location plus the attention-weighted average of the same level embedding at other locations.

- If we train the predictions to agree with the consensus, we will increase the agreement between embeddings that are similar.

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 an object-level embedding vector for each location.
  - Each location can 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 briefly explained three important advances in neural networks: transformers; SimCLR, implicit functions.

- I showed how to combine these three advances to design GLOM which 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.

- Nobody else is interested in solving this problem, but they should be.

- The main idea in the talk was complicated. I will put a long paper on arxiv soon.

THE END