Representing part-whole hierarchies in a neural network (GLOM)

Léo Gagnon

May 31, 2021

Université de Montréal

Plan of the presentation

1. Human vision vs CNN vision.
2. GLOM overview (high-level description)
3. Deeper look at the parts
4. Discussion : important ideas, potential problems.

Human vision vs CNN vision

Human vision

There is strong psychological evidence that people parse visual scenes into part-whole parse tree where

Human vision

There is strong psychological evidence that people parse visual scenes into part-whole parse tree where

- nodes are entities that have an *identity* (with corresponding *intrinsic frame of reference*) and a *pose*.

Human vision

There is strong psychological evidence that people parse visual scenes into part-whole parse tree where

- nodes are entities that have an *identity* (with corresponding *intrinsic frame of reference*) and a *pose*.
    - representation is viewpoint-equivariant

Human vision

There is strong psychological evidence that people parse visual scenes into part-whole parse tree where

- nodes are entities that have an *identity* (with corresponding *intrinsic frame of reference*) and a *pose*.
    - representation is viewpoint-equivariant
- the link between a part and a whole is the relationship between the entities irrespective of their pose.

Human vision

There is strong psychological evidence that people parse visual scenes into part-whole parse tree where

- nodes are entities that have an *identity* (with corresponding *intrinsic frame of reference*) and a *pose*.
  - representation is viewpoint-equivariant
- the link between a part and a whole is the relationship between the entities irrespective of their pose.
  - the relationship between entities is viewpoint-invariant

Hinton's cube demonstration

Hinton proposes the following experiment

1. Visualize a cube lying flat in front of you.
2. Orient the cube so that a vertical axis passes through two opposing vertex of the cube.
3. Try to point out where the other vertex are.

Hinton's cube demonstration

Hinton proposes the following experiment

1. Visualize a cube lying flat in front of you.
2. Orient the cube so that a vertical axis passes through two opposing vertex of the cube.
3. Try to point out where the other vertex are.

It's very hard for most people!
Imposing a different frame of reference changes the identity of the object in our head

Hinton's cube demo

One could perceive this new solid as having a crown with the following part-whole hierarchy

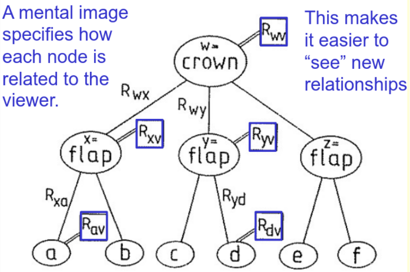

Figure 1: Parse tree for the crown of the tilted cube

The links between node are coordinate transformations between intrinsic coordinate frames of entities. The pose of every node can be deduced just by seeing one.

Hinton's cube demonstration

The "crown" representation is not the only way to look at the edges.

Hinton’s cube demonstration

The "crown" representation is not the only way to look at the edges.

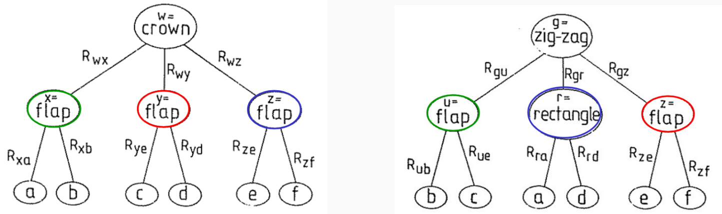

Figure 2: Different parse trees for "same" object

Hinton's cube demonstration

The "crown" representation is not the only way to look at the edges.

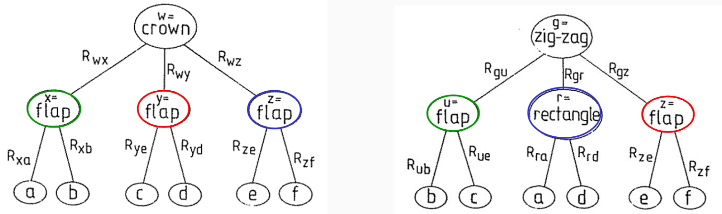

Figure 2: Different parse trees for "same" object

The "crown" representation makes it obvious that there is a three-fold rotational symmetry while the "zig-zag" representation makes it obvious that there is a pair of parallel lines.

Human vision

Recap :

- We chop up the visual world into *things* that have an intrinsic reference frame.
- We mentally represent these *things* with viewpoint-invariant hierarchical relationship with other *things*.
- When we look at a scene we try to fit a parse tree that explains part-whole relationships between *things-at-a-pose* we see.

How it differs from CNN vision

CNN don't seem to be doing that.

- The pooling operation make the network locally invariant to the position of a feature (not viewpoint-equivariant)
- They don't appear to be using viewpoint-invariant entities representations. They seem to learn viewpoint-invariance by looking at more data.
- They lack explicit hierarchical structure. They seem to be mostly looking at small patterns/texture at different scales, not part-whole hierarchies.

GLOM overview (high-level description)

GLOM overview

"GLOM answers the question: How can a neural network with a fixed architecture parse an image into a part-whole hierarchy which has different structure for each image?"

GLOM overview

Figure 3: Doge

GLOM overview

Figure 4: Image partitioned in locations

GLOM overview

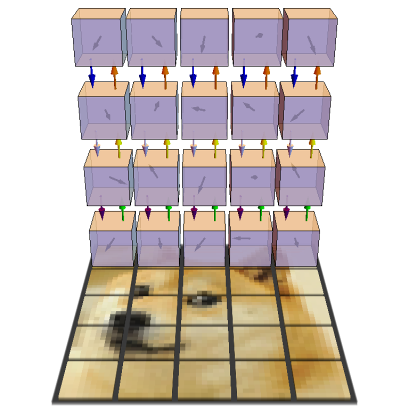

Figure 5: Stacks of activity vectors at each locations. Shared weights. Inter-level interaction are not shown.

GLOM overview

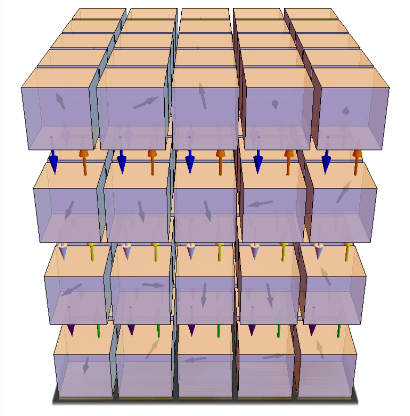

Figure 6: Stacks of activity vectors at each locations. Shared weights. Inter-level interaction are not shown.

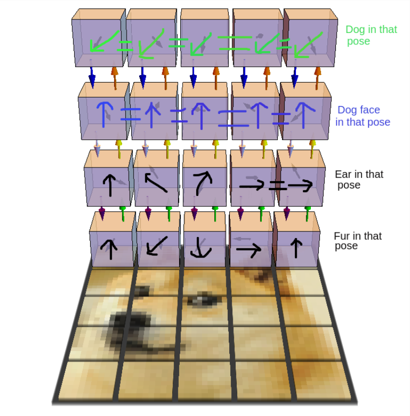

Figure 7: Each activity vector represent an *object-at-a-pose*. Bottom-up and top-down neural networks try to predict other levels. Inter-level interactions act as echo-chambers.

GLOM overview

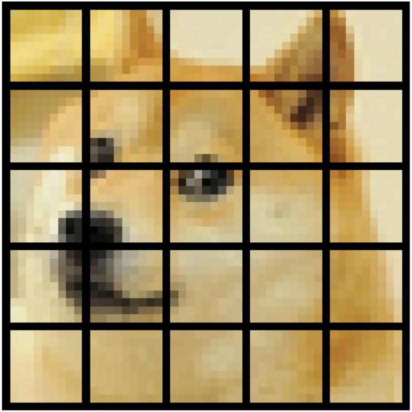

Figure 8: Level 0

GLOM overview

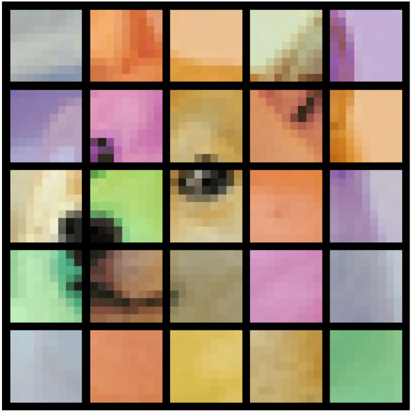

Figure 8: Level 1

GLOM overview

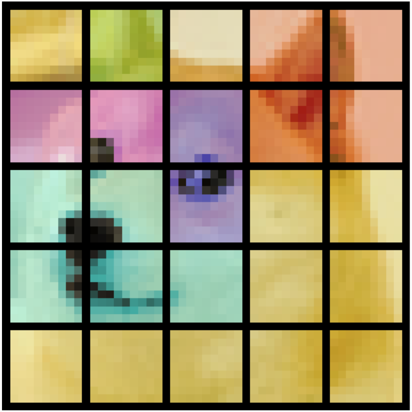

Figure 8: Level 2

GLOM overview

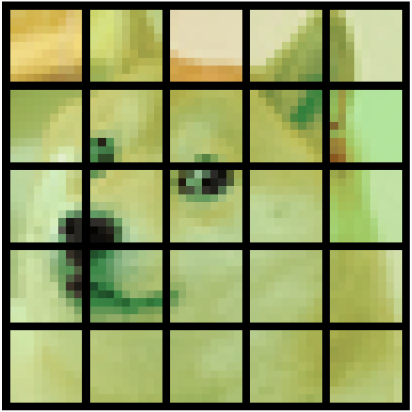

Figure 8: Level 3

GLOM overview

The GLOM architecture processes an image by repeatedly **updating** the activity vectors (representations) with 3 contributions (weighted average) :

- Bottom-up prediction from cell bellow
- Top-down prediction from cell above (also take position as input, more on this later)
- Echo-chamber by looking at nearby cells on the same level

GLOM overview

The GLOM architecture processes an image by repeatedly **updating** the activity vectors (representations) with 3 contributions (weighted average) :

- Bottom-up prediction from cell bellow
- Top-down prediction from cell above (also take position as input, more on this later)
- Echo-chamber by looking at nearby cells on the same level

Cellular automaton!

Main idea

The main idea is the following :

- At the beginning, the activity vectors are very uncertain about what they represent : they are a superposition of possible objects-at-a-pose.

Main idea

The main idea is the following :

- At the beginning, the activity vectors are very uncertain about what they represent : they are a superposition of possible objects-at-a-pose.
- The top-down and bottom-up prediction are therefore also uncertain : superposition of predictions.

Main idea

The main idea is the following :

- At the beginning, the activity vectors are very uncertain about what they represent : they are a superposition of possible objects-at-a-pose.
- The top-down and bottom-up prediction are therefore also uncertain : superposition of predictions.
- The update operation (averaging) increases the confidence that into predictions that are popular and spatially coherent.

Main idea

The main idea is the following :

- At the beginning, the activity vectors are very uncertain about what they represent : they are a superposition of possible objects-at-a-pose.
- The top-down and bottom-up prediction are therefore also uncertain : superposition of predictions.
- The update operation (averaging) increases the confidence that into predictions that are popular and spatially coherent.
- After repeated updates, the activity vectors collectively converge to the representation of a parse tree where the nodes are islands of identical vectors.

GLOM overview

Recap :

- Activity vectors represent what is present the location at a certain level of description.
- Within a column, adjacent activity vectors try to predict each other.
- Within a level nearby activity vectors make echo-chambers.
- Nodes of the parse tree are represented by island of identical vectors.
- The parse tree **emerges** over the repeated interactions between cells.

Deeper look

Activity vectors

Activity vectors

What exactly does the vector represent?

How could it capture uncertainty?

- Naive way would be to take vectors as a point in a identity-pose latent space (some neurons for pose, some for identity).

Activity vectors

What exactly does the vector represent?

How could it capture uncertainty?

- Naive way would be to take vectors as a point in a identity-pose latent space (some neurons for pose, some for identity).
    - This cannot account for uncertainty!

Activity vectors

What exactly does the vector represent?

How could it capture uncertainty?

- Naive way would be to take vectors as a point in a identity-pose latent space (some neurons for pose, some for identity).
  - This cannot account for uncertainty!
- Instead, Hinton proposes that activity vectors represent log-probability distributions in that space.

Activity vectors

An activity vector $u = [u_1, \ldots, u_N]$ represents the unnormalized log-probability distribution

$$\log(u(x)) = \sum_{i=1}^{N} u_i \log(p_i(x)) = \log\left(\Pi_{i=1}^{N} p_i(x)^{u_i}\right)$$
$$\implies u(x) = \Pi_{i=1}^{N} p_i(x)^{u_i}$$

where the $\log(p_i)$ are basis functions (remain unchanged within a level).

$u(x)$ is an unnormalized product of experts (PoE) and the u_i determine the reliability of the experts.

Activity vectors

An activity vector $u = [u_1, \ldots, u_N]$ represents the unnormalized log-probability distribution

$$\log(u(x)) = \sum_{i=1}^{N} u_i \log(p_i(x)) = \log\Big(\Pi_{i=1}^{N} p_i(x)^{u_i}\Big)$$
$$\implies u(x) = \Pi_{i=1}^{N} p_i(x)^{u_i}$$

where the $\log(p_i)$ are basis functions (remain unchanged within a level).

$u(x)$ is an unnormalized product of experts (PoE) and the u_i determine the reliability of the experts.

Distributed representation!

Activity vectors

Let $u = [u_1, \ldots, u_N]$ and $v = [v_1, \ldots, u_N]$ be activity vectors. Then if we do a weighted average with $\alpha + \beta = 1$ we get the following distribution

$$\log((\alpha u + \beta v)(x)) = \sum_{i=1}^{N} (\alpha u_i + \beta v_i) \log(p_i(x)) = \log\left(\Pi_{i=1}^{N} p_i(x)^{\alpha u_i + \beta v_i}\right)$$

$$\implies (u + v)(x) = \Pi_{i=1}^{N} p_i(x)^{\alpha u_i + \beta v_i} = \left(\Pi_{i=1}^{N} p_i(x)^{u_i}\right)^{\alpha} \left(\Pi_{i=1}^{N} p_i(x)^{v_i}\right)^{\beta}$$

$$= u(x)^{\alpha} v(x)^{\beta}$$

Activity vectors

Let $u = [u_1, \ldots, u_N]$ and $v = [v_1, \ldots, u_N]$ be activity vectors. Then if we do a weighted average with $\alpha + \beta = 1$ we get the following distribution

$$\log((\alpha u + \beta v)(x)) = \sum_{i=1}^{N} (\alpha u_i + \beta v_i) \log(p_i(x)) = \log\left(\Pi_{i=1}^{N} p_i(x)^{\alpha u_i + \beta v_i}\right)$$

$$\implies (u+v)(x) = \Pi_{i=1}^{N} p_i(x)^{\alpha u_i + \beta v_i} = \left(\Pi_{i=1}^{N} p_i(x)^{u_i}\right)^{\alpha} \left(\Pi_{i=1}^{N} p_i(x)^{v_i}\right)^{\beta}$$

$$= u(x)^{\alpha} v(x)^{\beta}$$

Adding the embedding makes another unnormalized PoE with α and β specifying the reliability of each distribution. This will pick out the modes (the common predictions)!

Activity vectors

Another way to see it is that averaging $u = [u_1, \ldots, u_N]$ and $v = [v_1, \ldots, u_N]$ averages the confidence they put on each experts.

Activity vectors

Some things to note/reemphasize :

- The job of the top-down/bottom-up neural networks becomes MUCH harder (everything is indirect and PoE are hard to learn).

Activity vectors

Some things to note/reemphasize :

- The job of the top-down/bottom-up neural networks becomes MUCH harder (everything is indirect and PoE are hard to learn).
- Distributed representation!

Activity vectors

Some things to note/reemphasize :

- The job of the top-down/bottom-up neural networks becomes MUCH harder (everything is indirect and PoE are hard to learn).
- Distributed representation!
- Individual neurons may be hard to interpret.

Top-down neural network

Top-down neural network

We want the activity vectors to form island of identical vectors but at the same time we want the top-down neural networks to predict the content of the level bellow. This is a problem.

Top-down neural network

We want the activity vectors to form island of identical vectors but at the same time we want the top-down neural networks to predict the content of the level bellow. This is a problem.

Solution : Give as input to the top-down neural network the location of the cell. This ANN can then be thought of as a function $f_h(x, y)$, indexed by the activity vector h that return the object-at-a-pose at position (x, y). This is the idea behind *neural fields*

Top-down neural network

We want the activity vectors to form island of identical vectors but at the same time we want the top-down neural networks to predict the content of the level bellow. This is a problem.

Solution : Give as input to the top-down neural network the location of the cell. This ANN can then be thought of as a function $f_h(x, y)$, indexed by the activity vector h that return the object-at-a-pose at position (x, y). This is the idea behind *neural fields*

Not trivial

Echo-chamber

Echo-chamber/attention

The "echo-chamber" contribution is a weighted average of nearby activity vectors on the same level. The weights are given by

$$w_{xy} = \frac{e^{\beta L_x \cdot L_y}}{\sum_z e^{\beta L_x \cdot L_z}}$$

where L_x is the embedding vector at level L and position x, z indexes *nearby* locations and β is an inverse temperature.

Echo-chamber/attention

The "echo-chamber" contribution is a weighted average of nearby activity vectors on the same level. The weights are given by

$$w_{xy} = \frac{e^{\beta L_x \cdot L_y}}{\sum_z e^{\beta L_x \cdot L_z}}$$

where L_x is the embedding vector at level L and position x, z indexes *nearby* locations and β is an inverse temperature.

This assure spacial coherence at the object level.

Training

Training

Hinton proposes to train GLOM in an unsupervised manner like BERT : mask the image and ask it to reconstruct the image. To that he adds a regularizer that encourages the top-down and bottom-up net to predict the consensus opinion. This is maybe a problem because it could encourage the nets to all predict the same thing.

Training

Hinton proposes to train GLOM in an unsupervised manner like BERT : mask the image and ask it to reconstruct the image. To that he adds a regularizer that encourages the top-down and bottom-up net to predict the consensus opinion. This is maybe a problem because it could encourage the nets to all predict the same thing. **Solution (maybe) :** Add another reguarizer that penalizes the neural networks when they predict something that does not agree with the consensus (contrastive learning).

Other design decisions

Other

- How many levels?

Other

- How many levels?
  - Few, about 5. To deal with more just change the scene level.

Other

- How many levels?
    - Few, about 5. To deal with more just change the scene level.
- The visual input?

Other

- How many levels?
  - Few, about 5. To deal with more just change the scene level.
- The visual input?
  - Use a CNN.

Other

- How many levels?
  - Few, about 5. To deal with more just change the scene level.
- The visual input?
  - Use a CNN.
- Inefficient use of memory?

Other

- How many levels?
  - Few, about 5. To deal with more just change the scene level.
- The visual input?
  - Use a CNN.
- Inefficient use of memory?
  - Yes but it is necessary. Also enables sparse connectivity.

Other

- How many levels?
  - Few, about 5. To deal with more just change the scene level.
- The visual input?
  - Use a CNN.
- Inefficient use of memory?
  - Yes but it is necessary. Also enables sparse connectivity.
- How to deal with videos?

Other

- How many levels?
  - Few, about 5. To deal with more just change the scene level.
- The visual input?
  - Use a CNN.
- Inefficient use of memory?
  - Yes but it is necessary. Also enables sparse connectivity.
- How to deal with videos?
  - Just change the image every timesteps (or a few).

Other

- How many levels?
  - Few, about 5. To deal with more just change the scene level.
- The visual input?
  - Use a CNN.
- Inefficient use of memory?
  - Yes but it is necessary. Also enables sparse connectivity.
- How to deal with videos?
  - Just change the image every timesteps (or a few).
- Biologically plausible?

Other

- How many levels?
  - Few, about 5. To deal with more just change the scene level.
- The visual input?
  - Use a CNN.
- Inefficient use of memory?
  - Yes but it is necessary. Also enables sparse connectivity.
- How to deal with videos?
  - Just change the image every timesteps (or a few).
- Biologically plausible?
  - Tree potential problems : contrastive negative examples, weight-sharing and back-propagation

Final thoughts

Things I like :

- Aligned with my introspective sense of how the brain works.
- Cellular automaton and emergent behavior.
- The idea that we could reuse the same GLOM network for different things that have the same hierarchical structure.

Main challenges :

- The bottom-up and top-down neural networks have a hard job. Not obvious how they would work.
- The training with contrastive learning.
- Size of the embedding vectors/memory usage.