
Capsule Networks
Aurélien Géron, November 2017
https://youtu.be/pPN8d0E3900

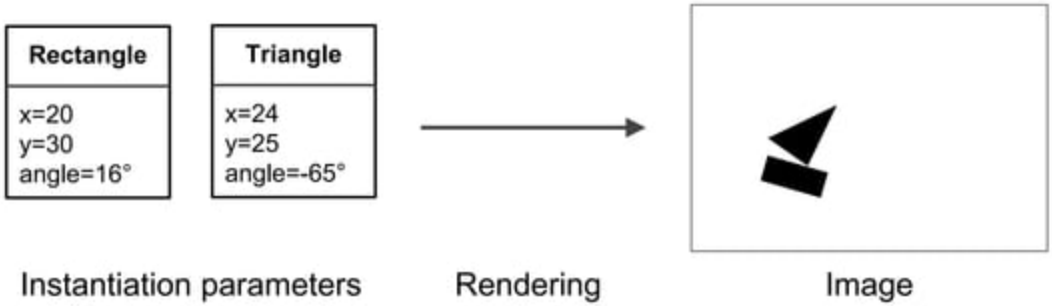
NIPS 2017 Paper

Dynamic Routing Between Capsules

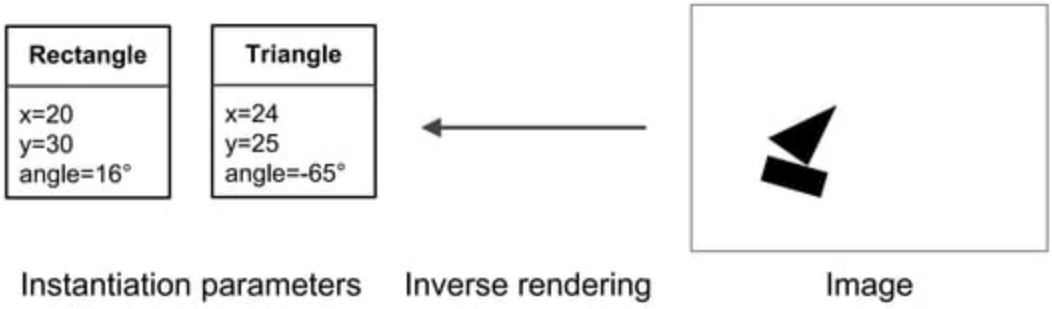
by Sara Sabour, Nicholas Frosst, Geoffrey E. Hinton

October 2017: https://arxiv.org/abs/1710.09829

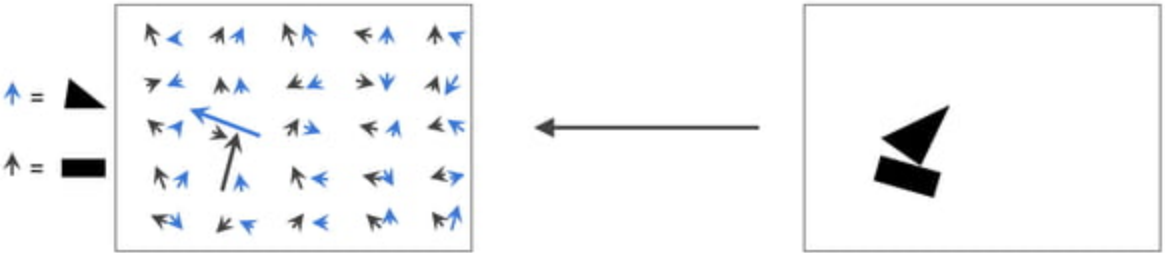
Computer Graphics



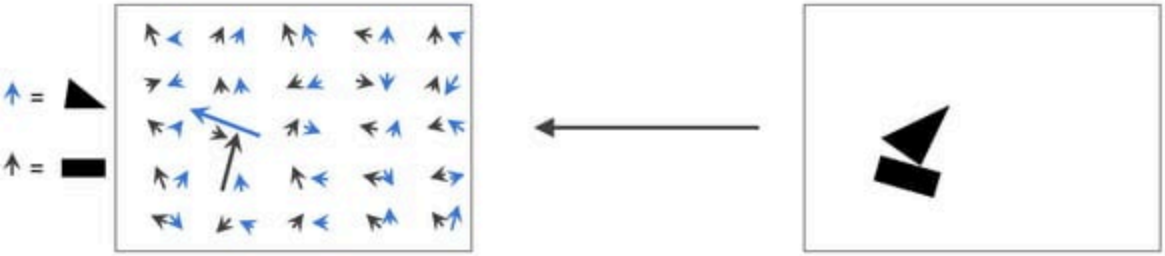
Inverse Graphics



Capsules

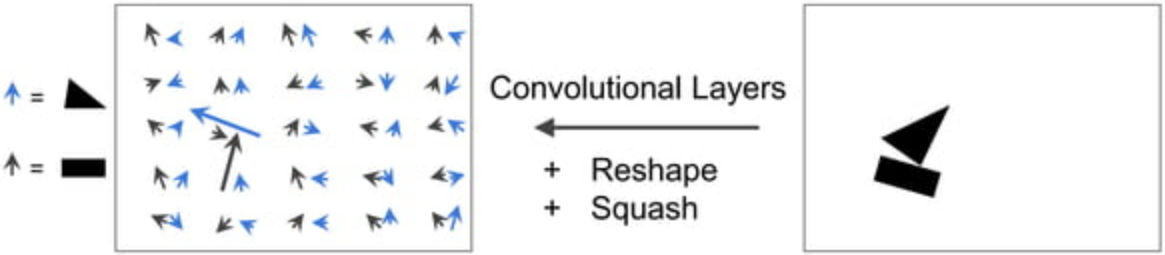


Capsules



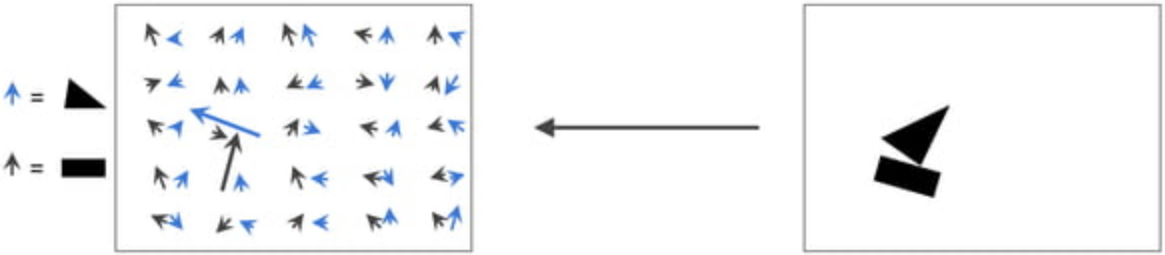
Activation vector:	**Length** = estimated probability of presence **Orientation** = object's estimated pose parameters

Capsules

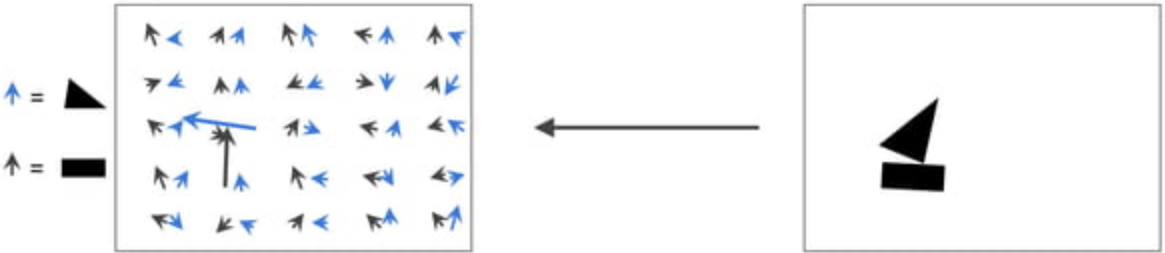


$$\text{Squash}(\mathbf{u}) = \frac{\|\mathbf{u}\|^2}{1 + \|\mathbf{u}\|^2} \frac{\mathbf{u}}{\|\mathbf{u}\|}$$

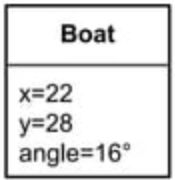
Equivariance



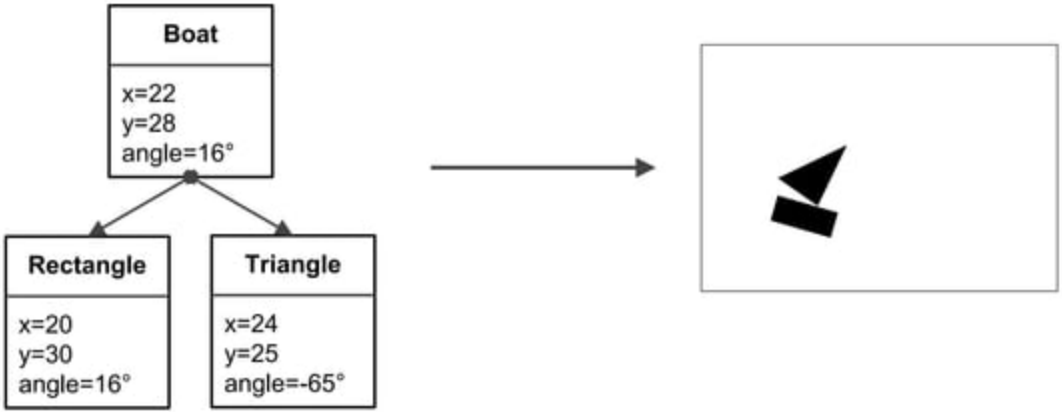
Equivariance



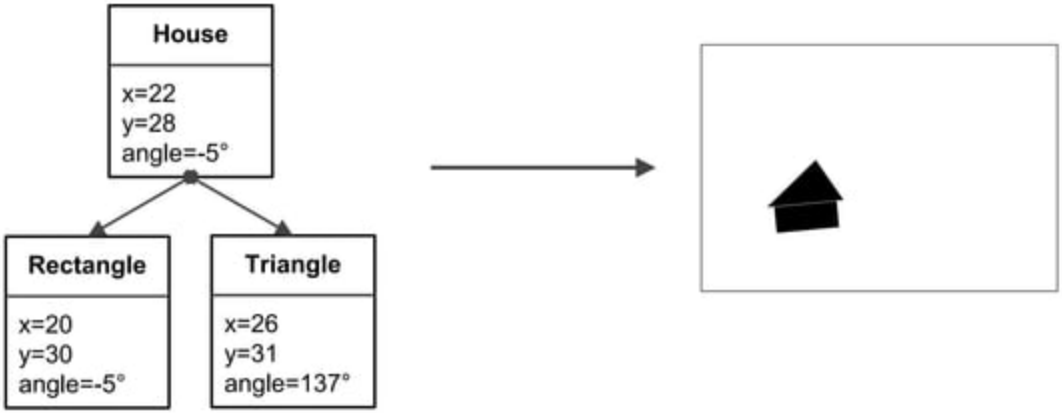
A hierarchy of parts



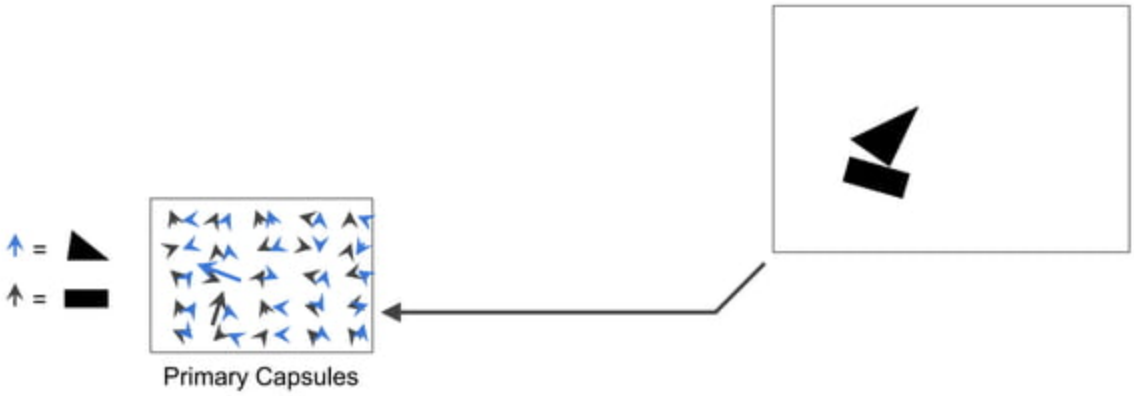
A hierarchy of parts



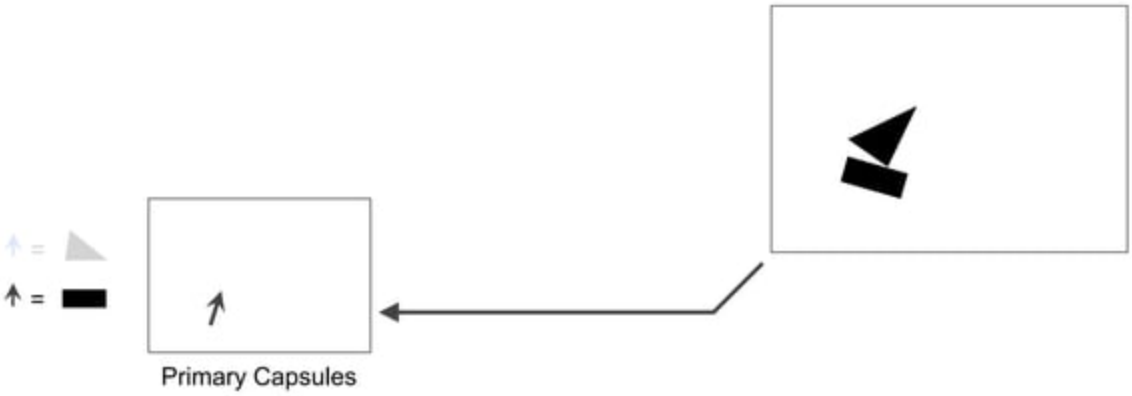
A hierarchy of parts



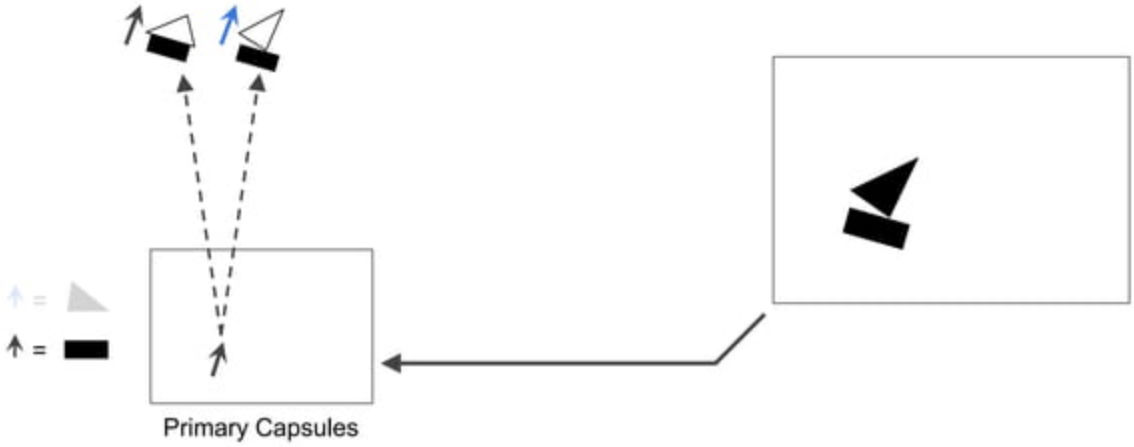
Primary Capsules



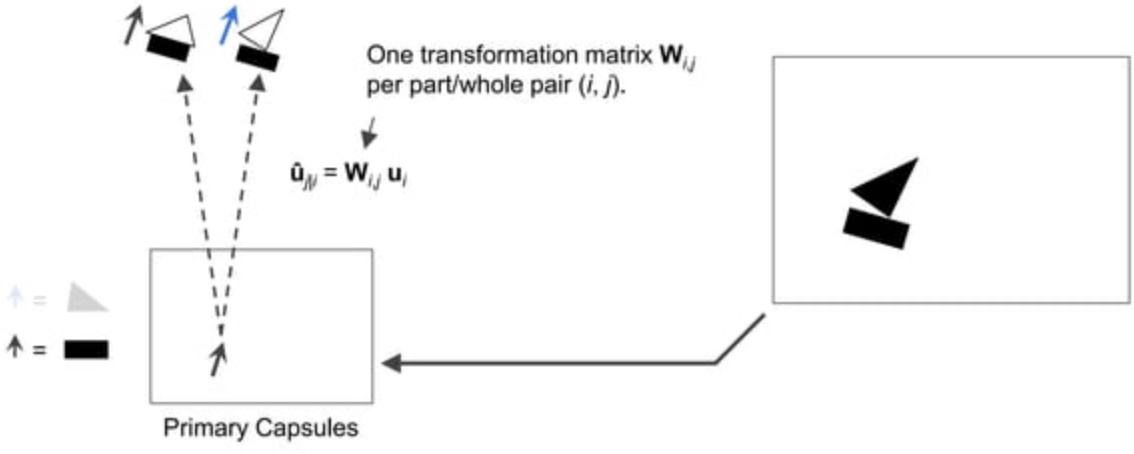
Predict Next Layer's Output



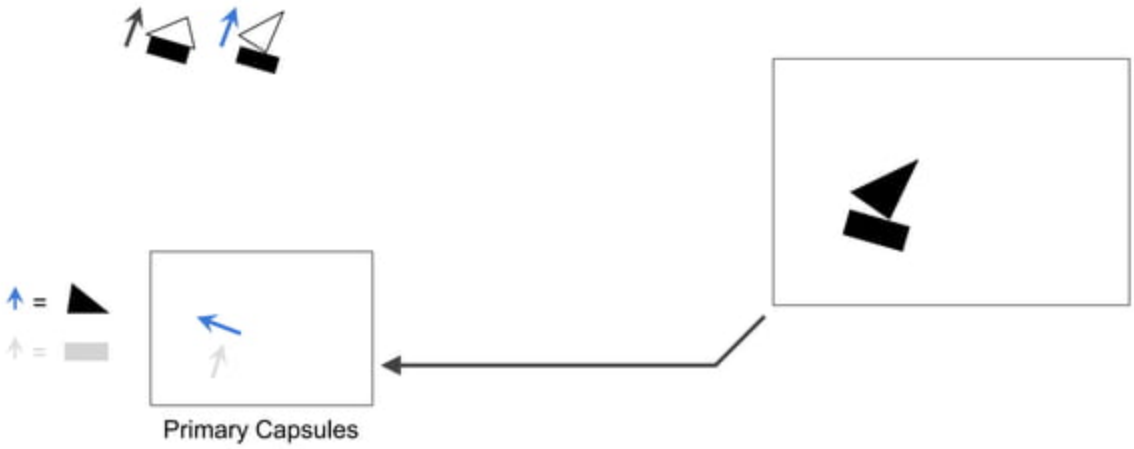
Predict Next Layer's Output



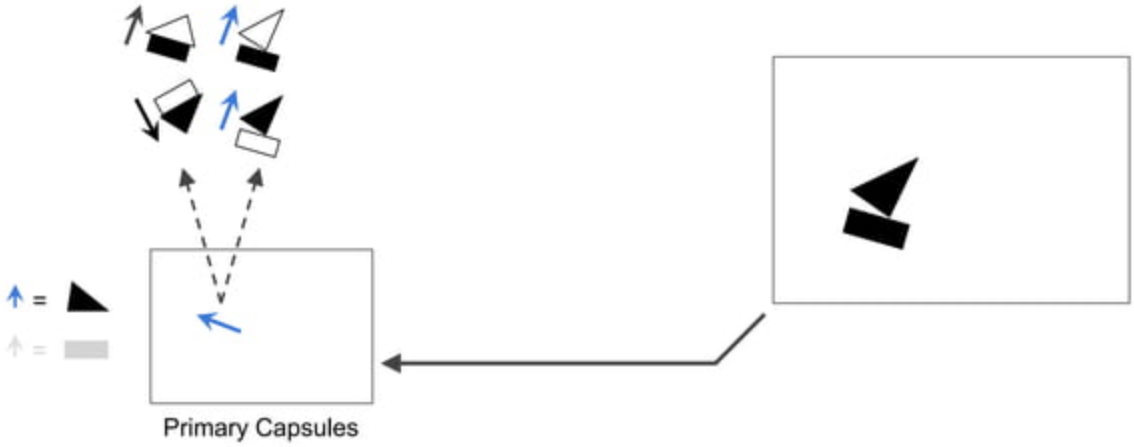
Predict Next Layer's Output



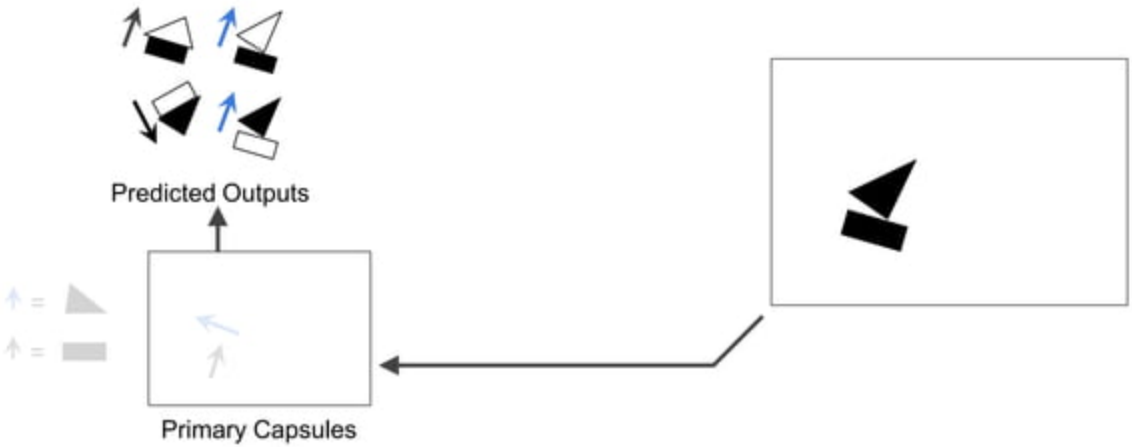
Predict Next Layer's Output



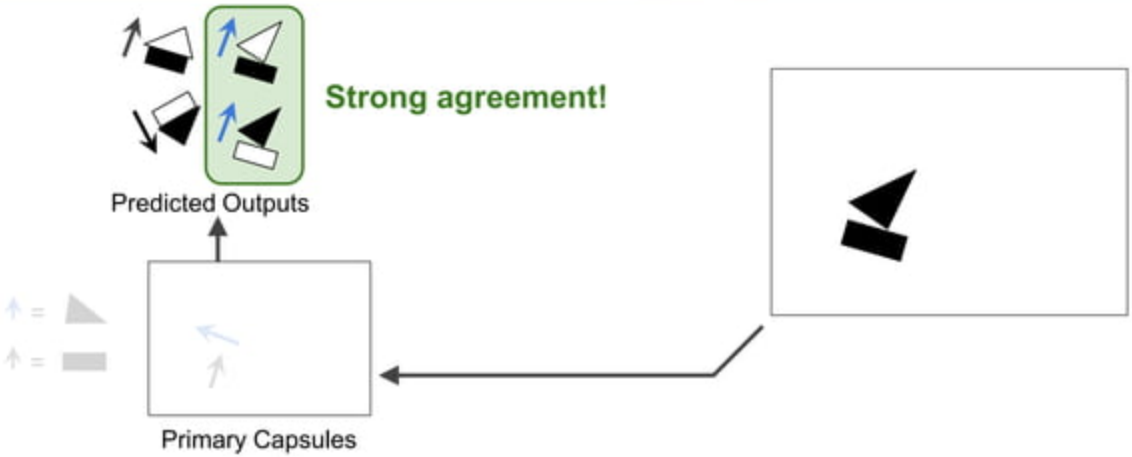
Predict Next Layer's Output



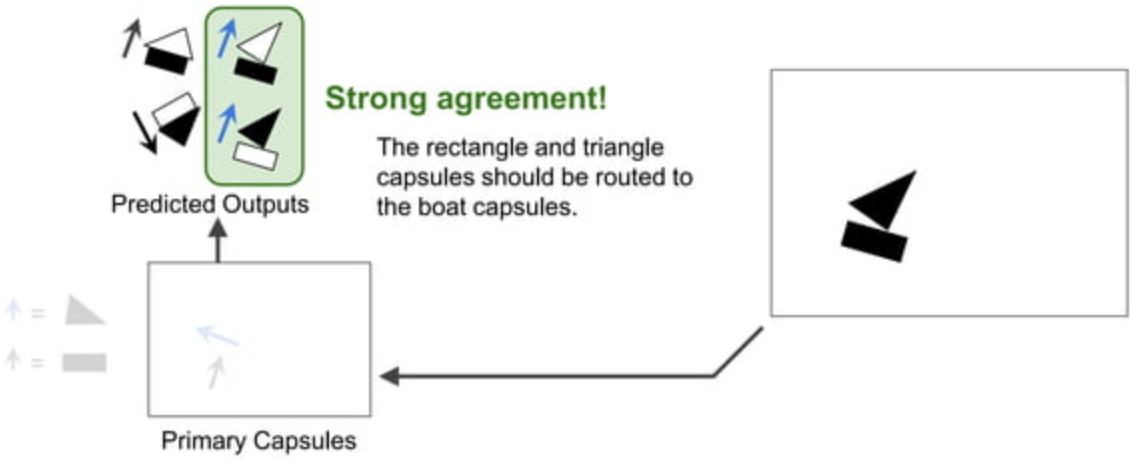
Compute Next Layer's Output



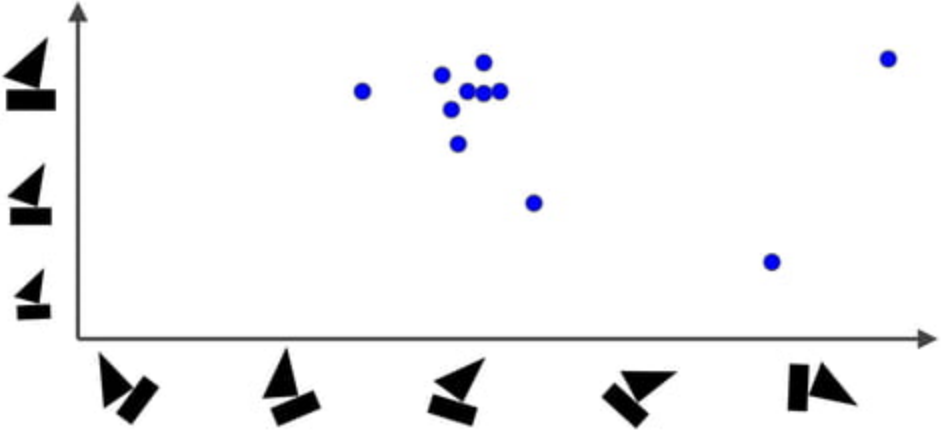
Routing by Agreement



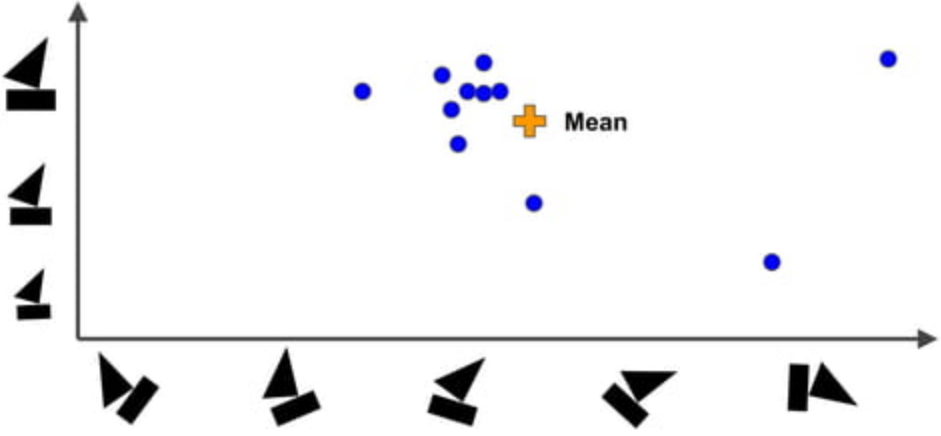
Routing by Agreement



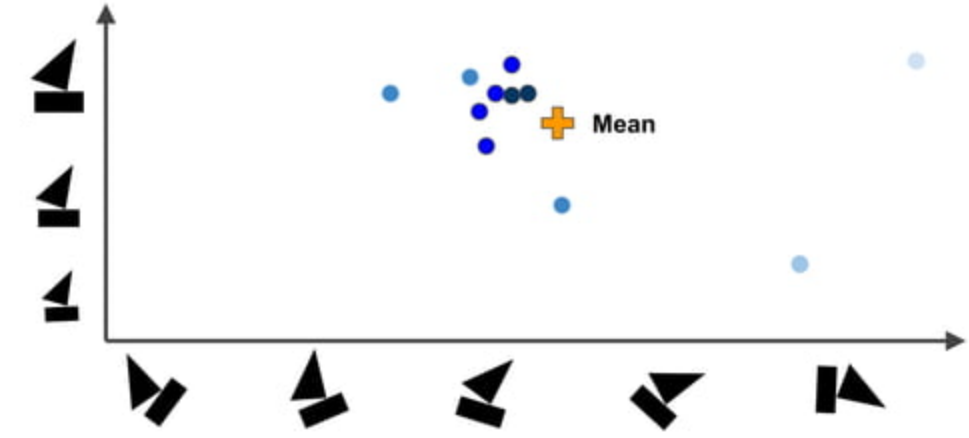
Clusters of Agreement



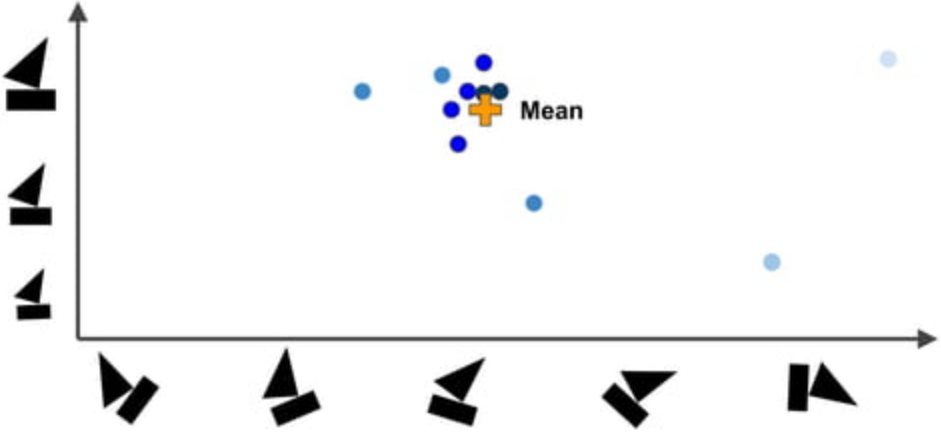
Clusters of Agreement



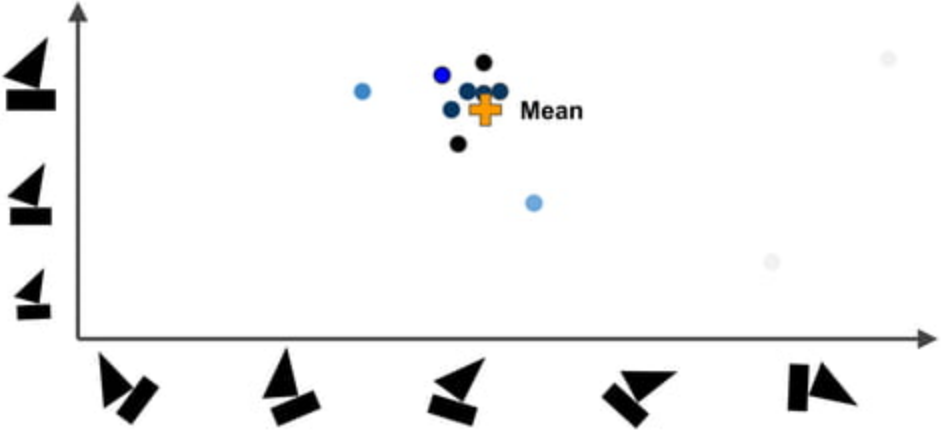
Clusters of Agreement



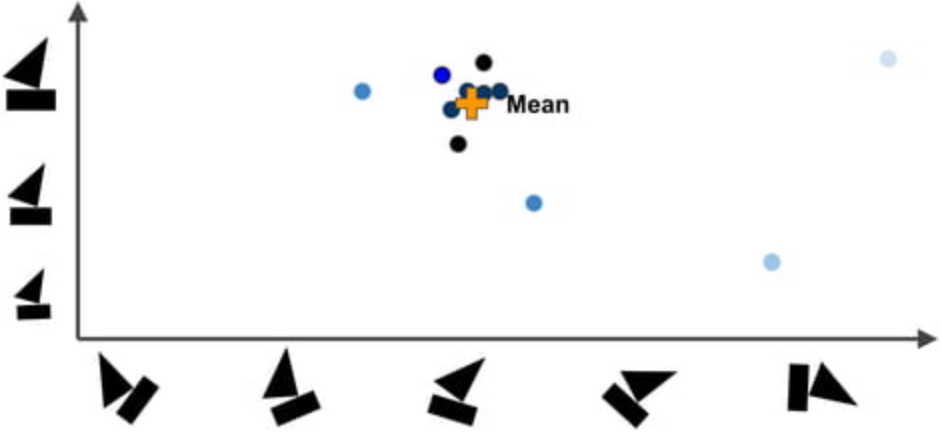
Clusters of Agreement



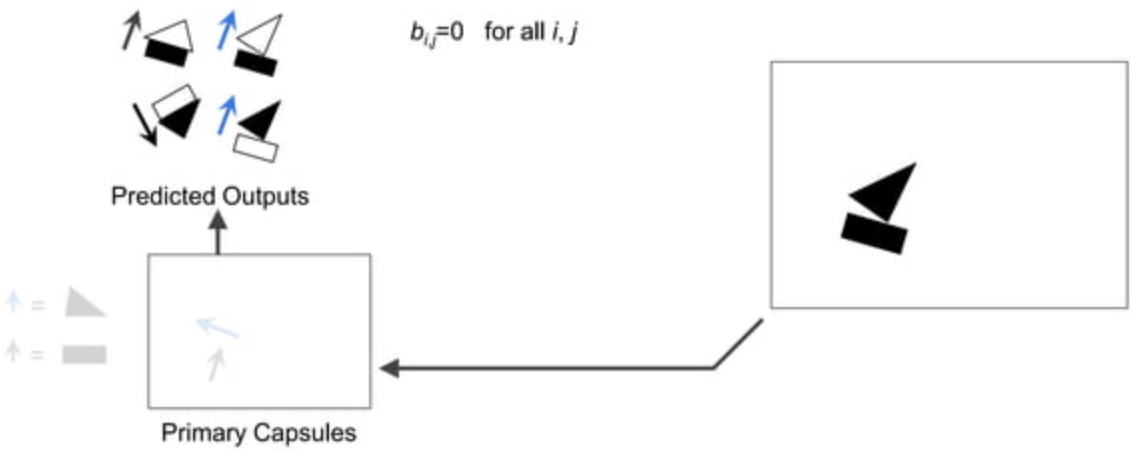
Clusters of Agreement



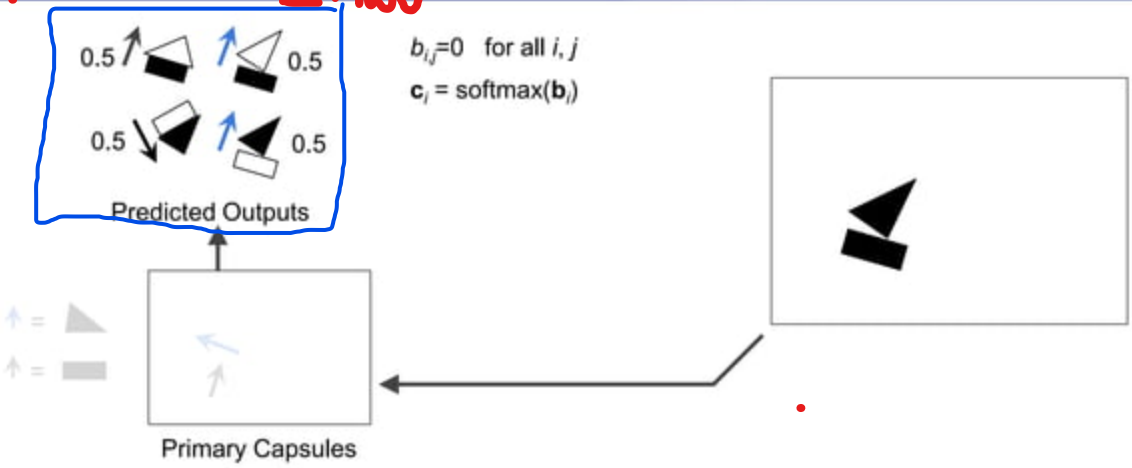
Clusters of Agreement



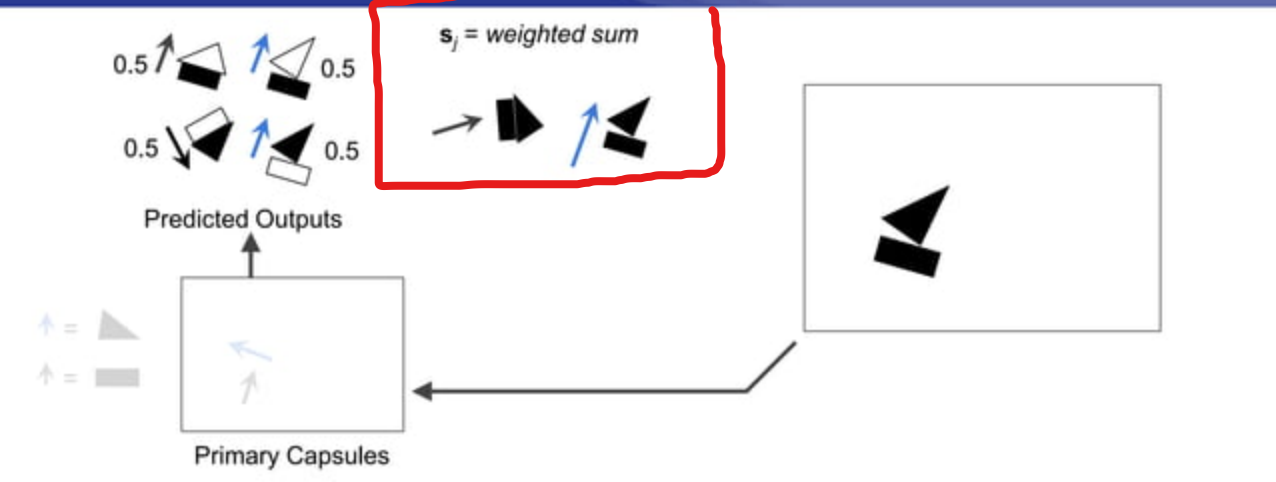
Routing Weights



Routing Weights

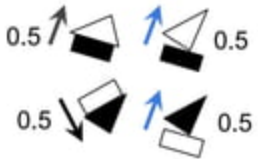


Compute Next Layer's Output



Compute Next Layer's Output

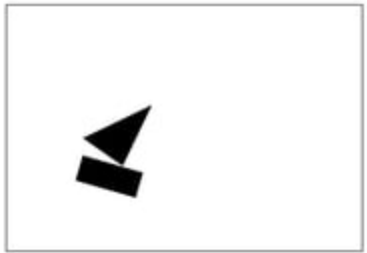
$v_j = squash(s_j)$



Predicted Outputs

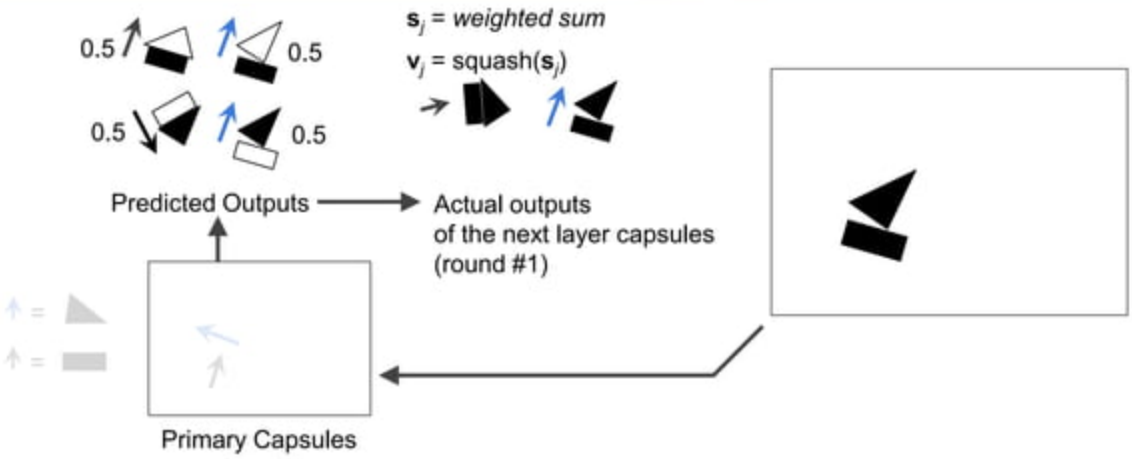
s_j = *weighted sum*

$v_j = squash(s_j)$

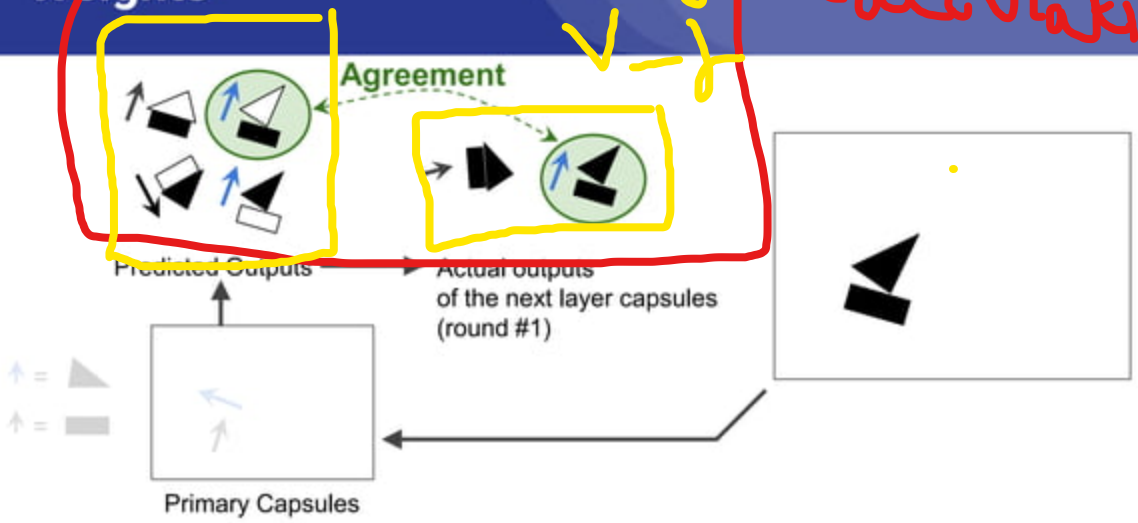


Primary Capsules

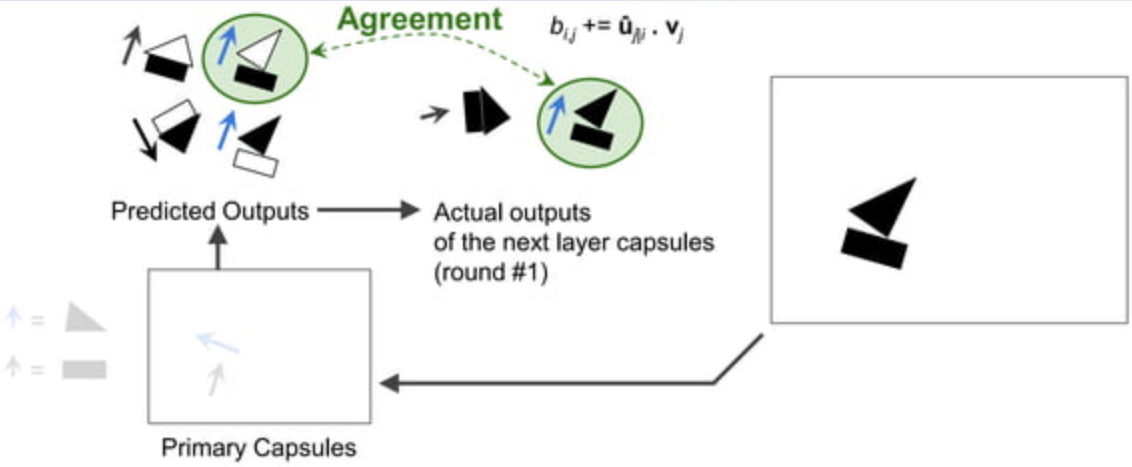
Compute Next Layer's Output



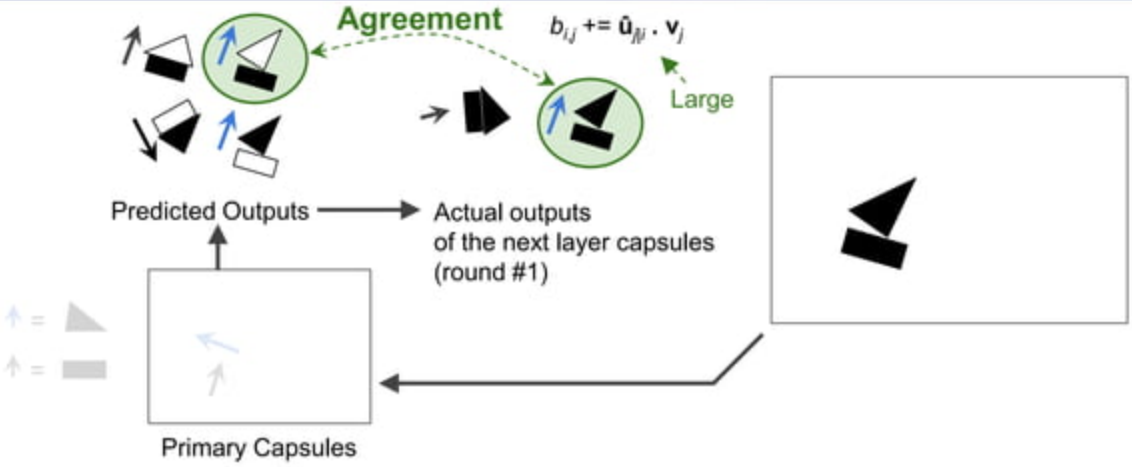
Update Routing Weights



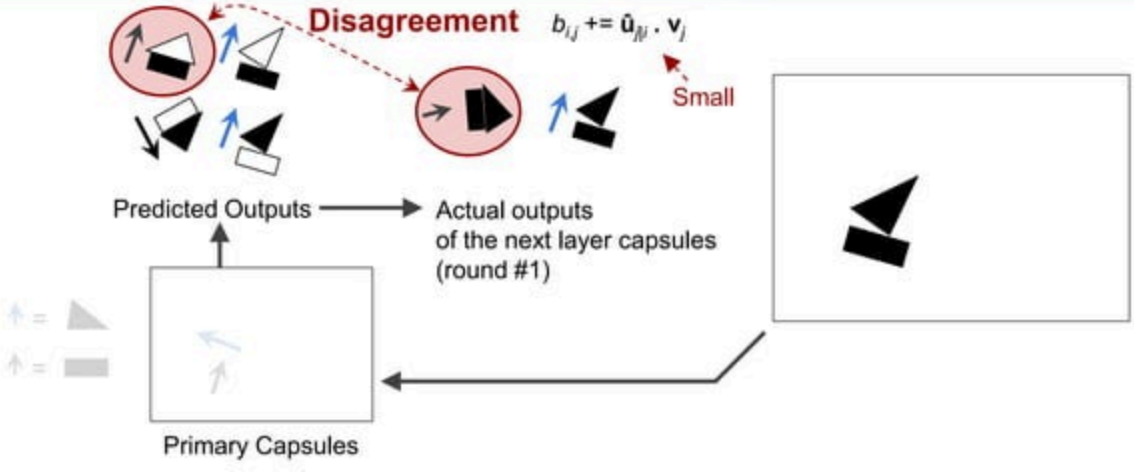
Update Routing Weights



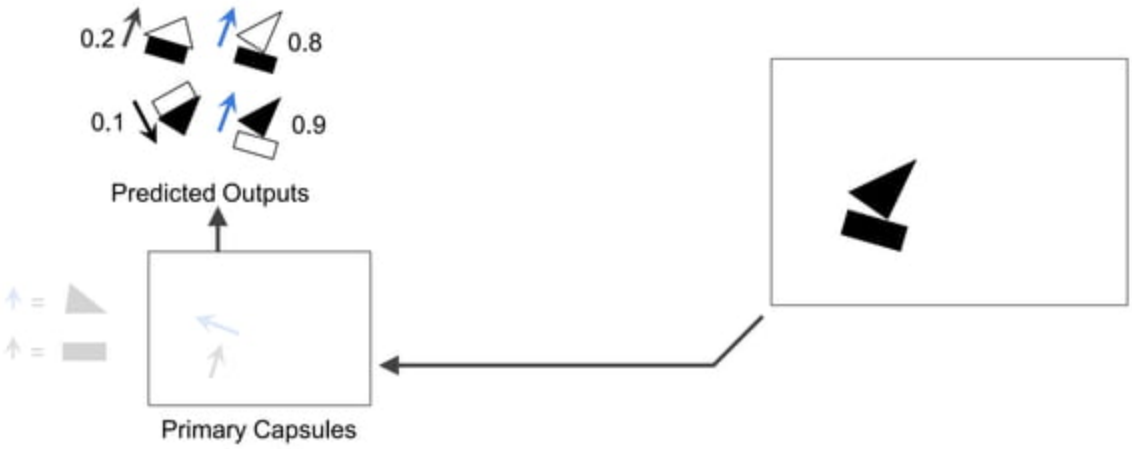
Update Routing Weights



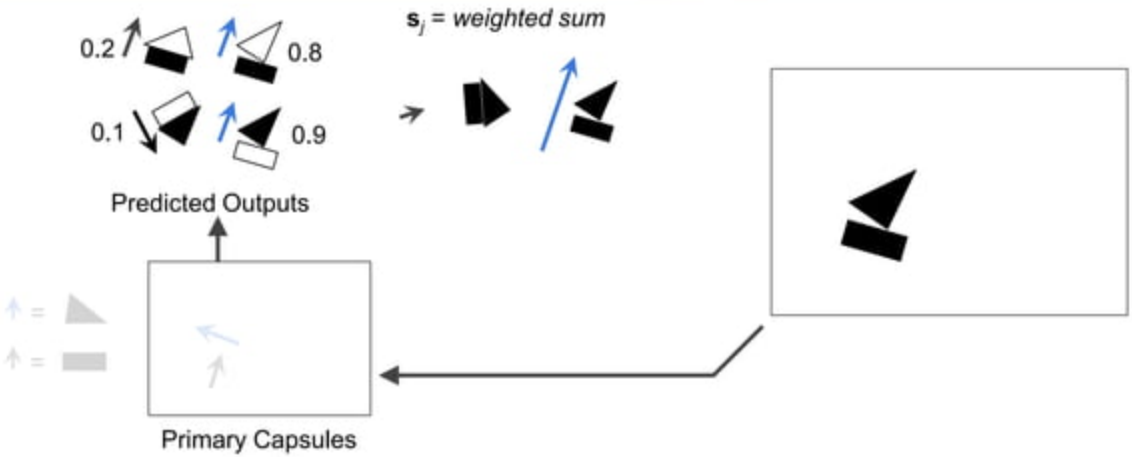
Update Routing Weights



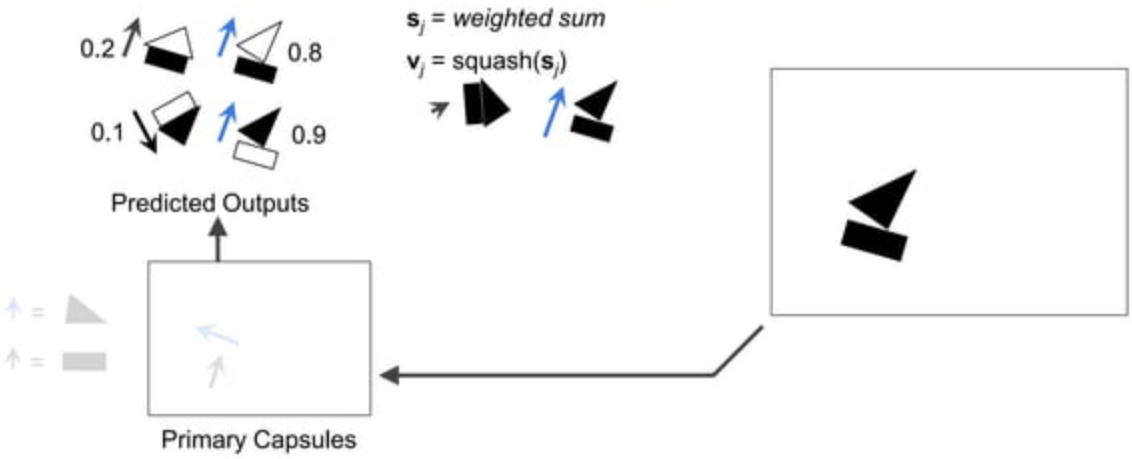
Compute Next Layer's Output



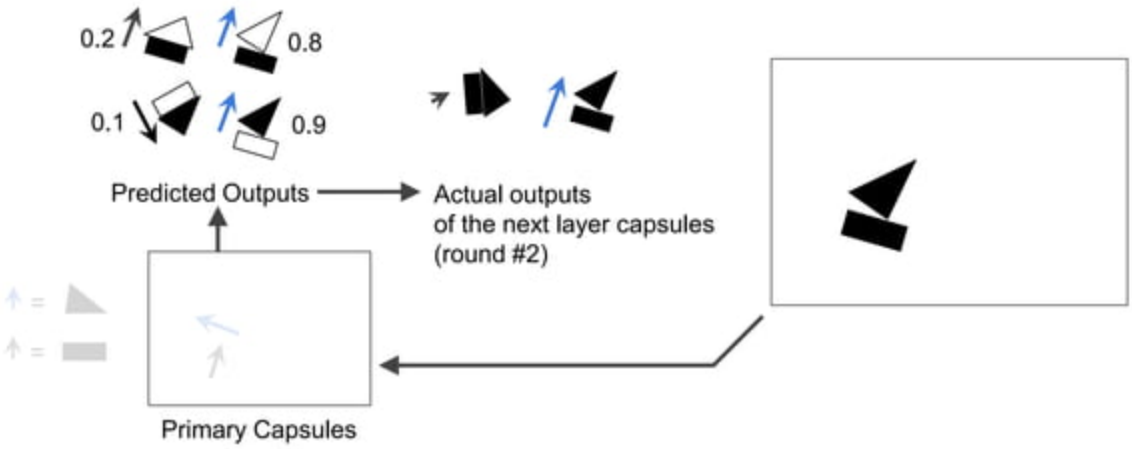
Compute Next Layer's Output



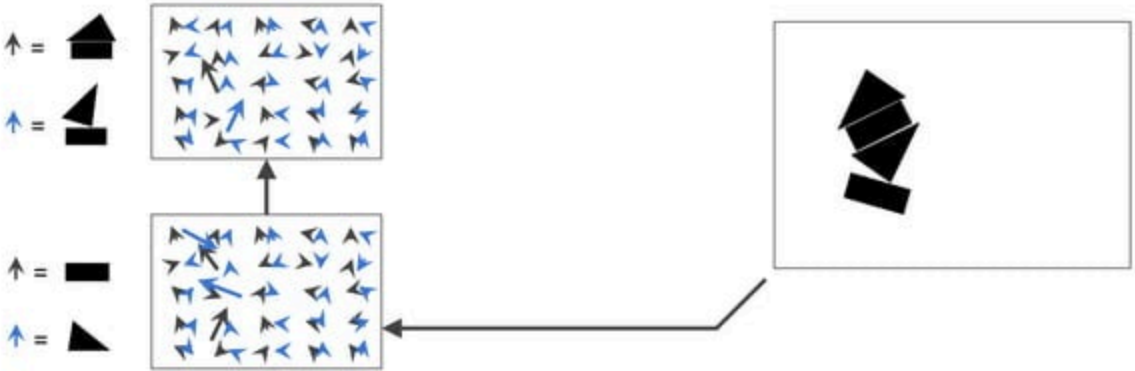
Compute Next Layer's Output



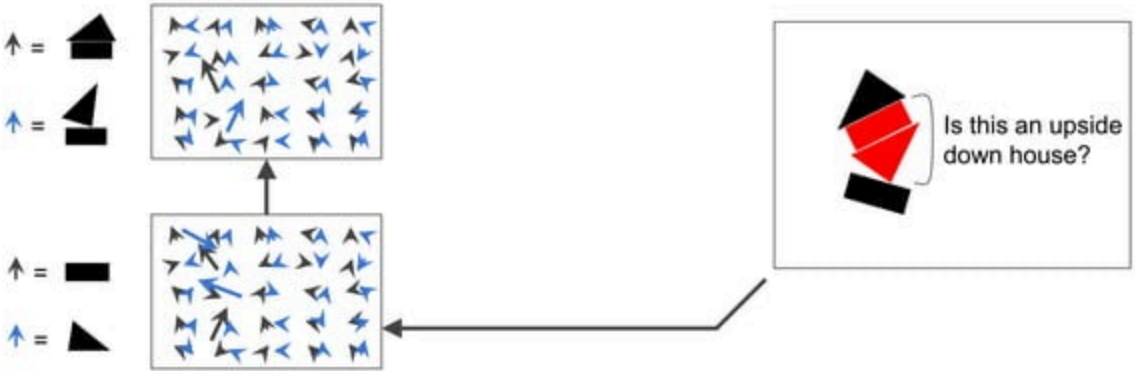
Compute Next Layer's Output



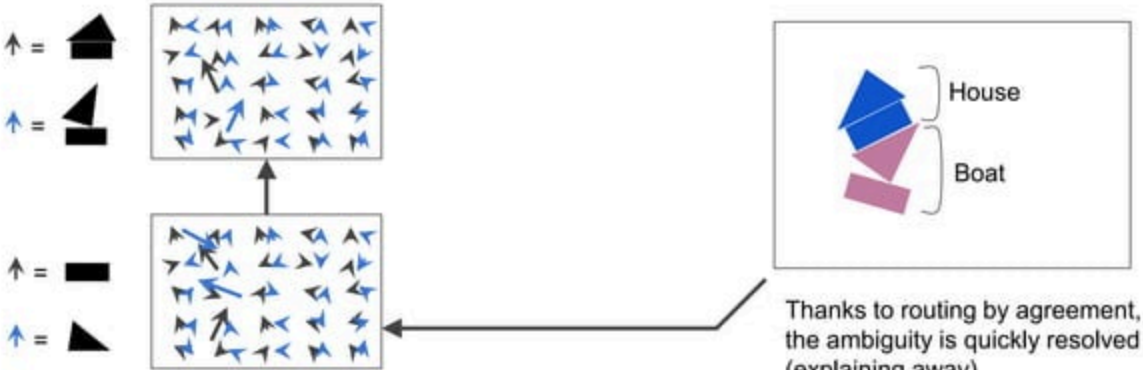
Handling Crowded Scenes



Handling Crowded Scenes

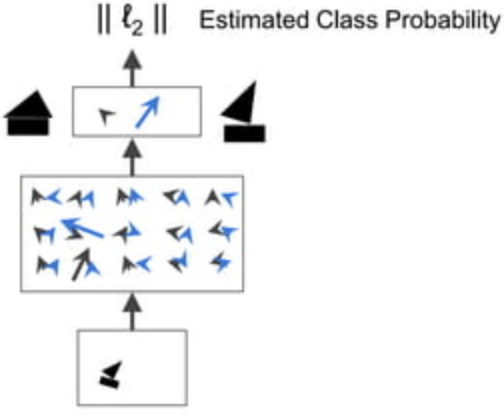


Handling Crowded Scenes

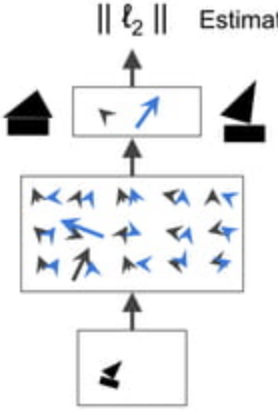


Thanks to routing by agreement, the ambiguity is quickly resolved (explaining away).

Classification
CapsNet



Training



To allow multiple classes, minimize margin loss:

$L_k = T_k \max(0, m^+ - \|\mathbf{v}_k\|^2)$

$+ \lambda (1 - T_k) \max(0, \|\mathbf{v}_k\|^2 - m^-)$

$T_k = 1$ iff class k is present

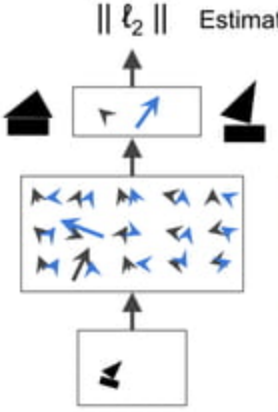
In the paper:

$m^- = 0.1$

$m^+ = 0.9$

$\lambda = 0.5$

Training



Translated to English:

"If an object of class *k* is present, then $||\mathbf{v}_k||^2$ should be no less than 0.9. If not, then $||\mathbf{v}_k||^2$ should be no more than 0.1."

To allow multiple classes, minimize margin loss:

$$L_k = T_k \max(0, m^+ - ||\mathbf{v}_k||^2) + \lambda (1 - T_k) \max(0, ||\mathbf{v}_k||^2 - m^-)$$

$T_k = 1$ iff class *k* is present

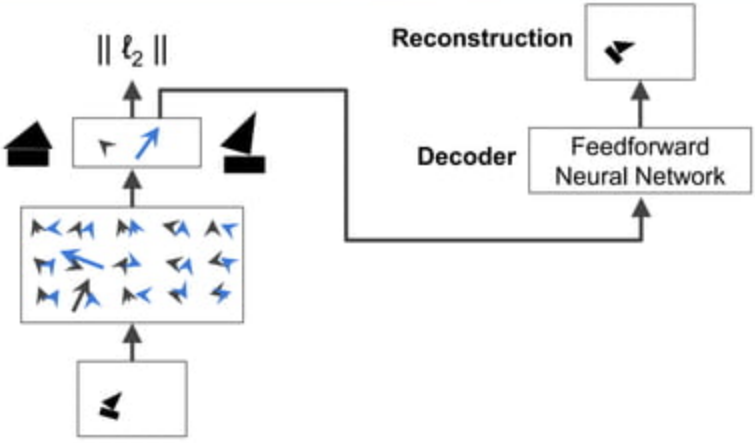
In the paper:

$m^- = 0.1$

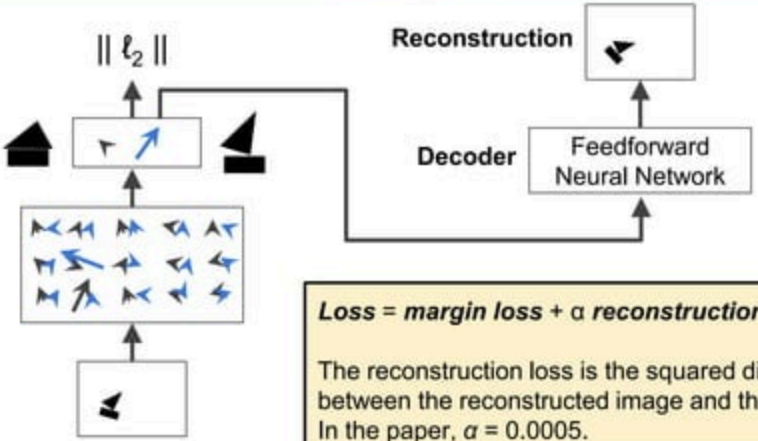
$m^+ = 0.9$

$\lambda = 0.5$

Regularization by Reconstruction



Regularization by Reconstruction

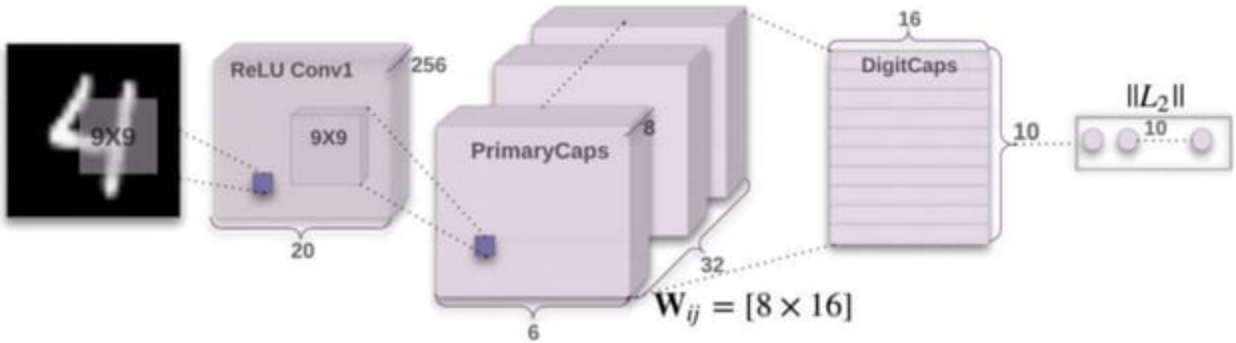


Loss = ***margin loss*** + α ***reconstruction loss***

The reconstruction loss is the squared difference between the reconstructed image and the input image.
In the paper, $\alpha = 0.0005$.

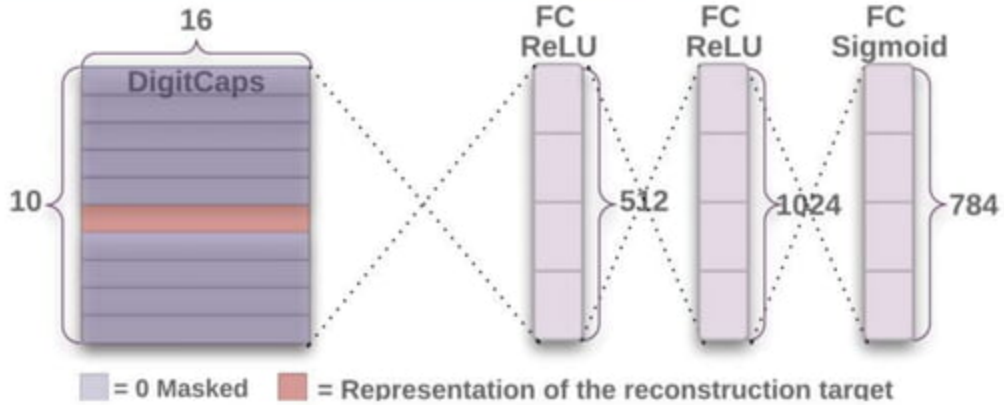
Aurélien Géron, 2017

A CapsNet for MNIST



(Figure 1 from the paper)

A CapsNet for MNIST – Decoder



(Figure 2 from the paper)

Interpretable Activation Vectors

Scale and thickness	
Localized part	
Stroke thickness	
Localized skew	
Width and translation	
Localized part	

(Figure 4 from the paper)

Pros

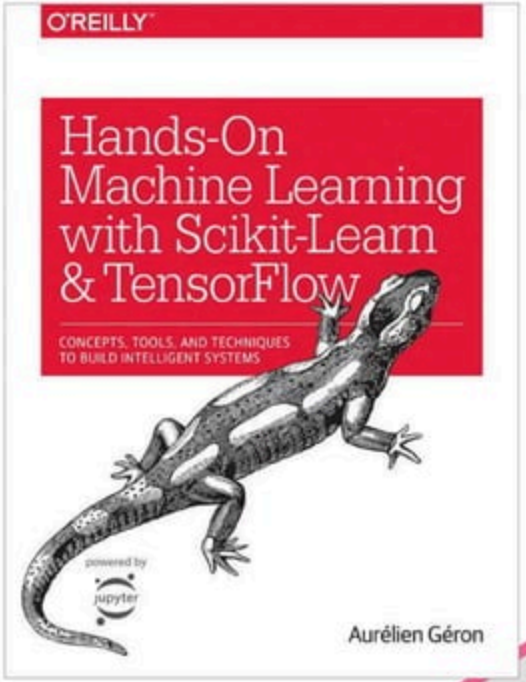
- Reaches high accuracy on MNIST, and promising on CIFAR10
- Requires less training data
- Position and pose information are preserved (equivariance)
- This is promising for image segmentation and object detection
- Routing by agreement is great for overlapping objects (explaining away)
- Capsule activations nicely map the hierarchy of parts
- Offers robustness to affine transformations
- Activation vectors are easier to interpret (rotation, thickness, skew…)
- It's Hinton! ;-)

Cons

- Not state of the art on CIFAR10 (but it's a good start)
- Not tested yet on larger images (e.g., ImageNet): will it work well?
- Slow training, due to the inner loop (in the routing by agreement algorithm)
- A CapsNet cannot see two very close identical objects
  - This is called "crowding", and it has been observed as well in human vision

Implementations

- Keras w/ TensorFlow backend: https://github.com/XifengGuo/CapsNet-Keras
- TensorFlow: https://github.com/naturomics/CapsNet-Tensorflow
- PyTorch: https://github.com/gram-ai/capsule-networks



116 customer reviews

#1 Best Seller in Computer Vision & Pattern Recognition

Amazon: https://goo.gl/IoWYKD
Twitter: @aureliengeron
github.com/ageron