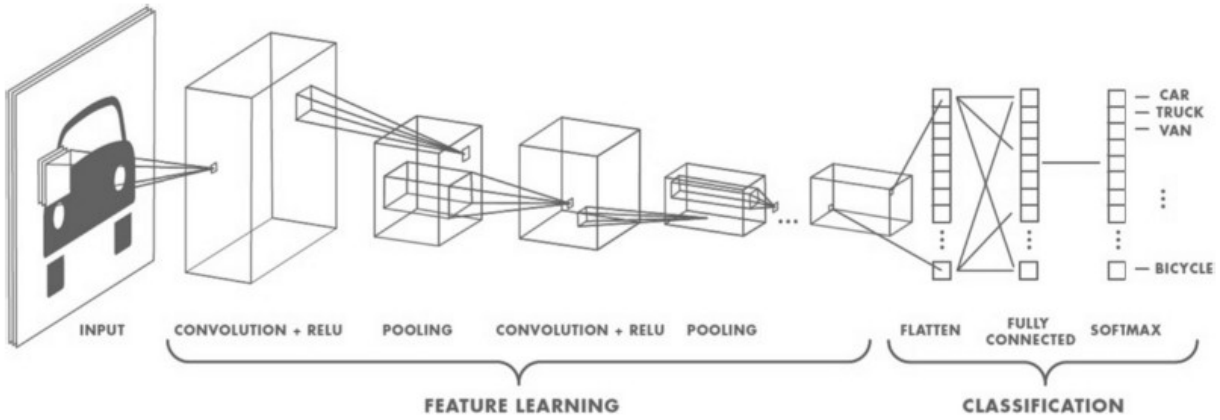
Dynamic Routing Between Capsules

발표자 : MI2RL 은다인

Convolutional Neural Networks

What is the problem with CNNs?



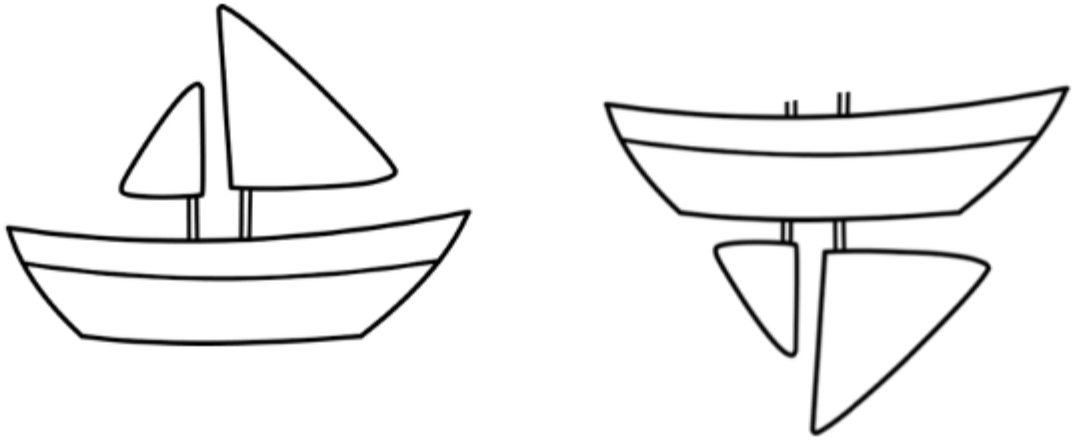
Contents from https://hackernoon.com/what-is-a-capsnet-or-capsule-network-2bfbe48769cc

1) If the images have rotation, tilt or any other different orientation then CNNs have poor performance.
2) In CNN each layer understands an image at a much more granular level (slow increase in receptive field).

→ DATA AUGMENTATION, MAX POOLING

Convolutional Neural Networks

What is the problem with CNNs?

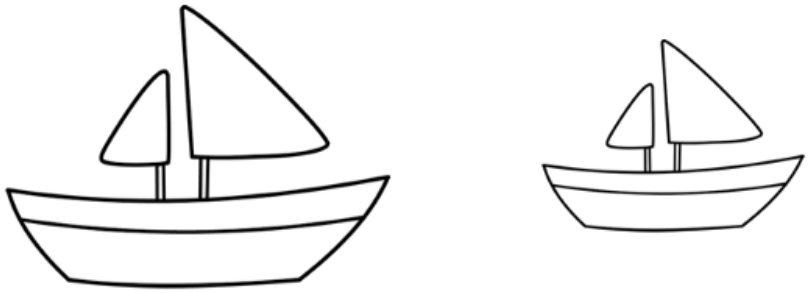


Contents from https://hackernoon.com/what-is-a-capsnet-or-capsule-network

"Pooling helps in creating the positional invariance. Otherwise This inv
also triggers false positive for images which have the components of
but not in the correct order."

Convolutional Neural Networks

What we need : EQUIVARIANCE (not invariance)

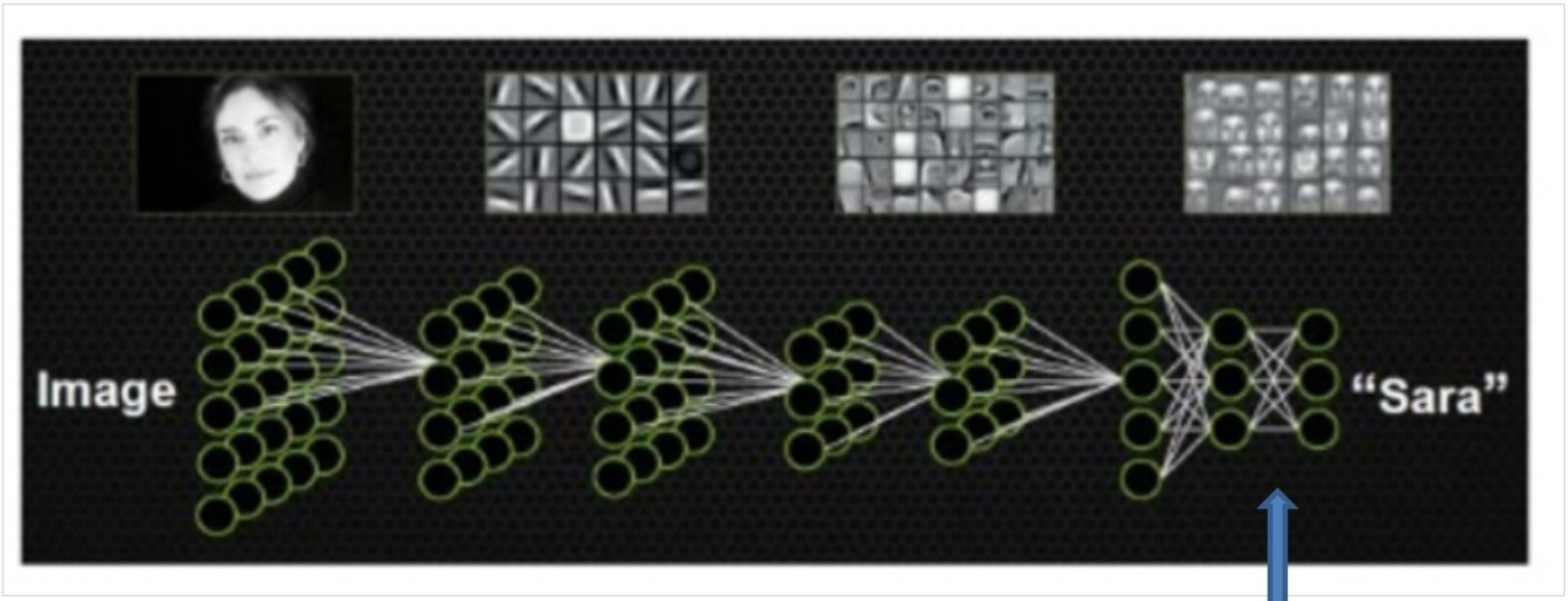


Contents from https://hackernoon.com/what-is-a-capsnet-or-capsule-network-2bfbe48769cc

"Equivariance makes a CNN understand the rotation or proportion change and adapt itself accordingly so that the spatial positioning inside an image is not lost."

Capsule Network

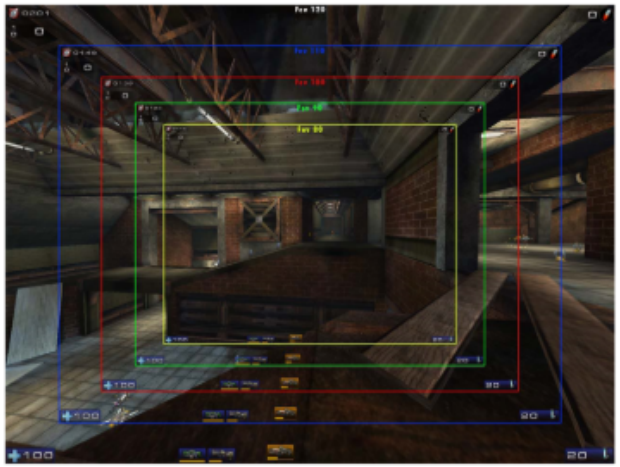
Convolution filter의 한계

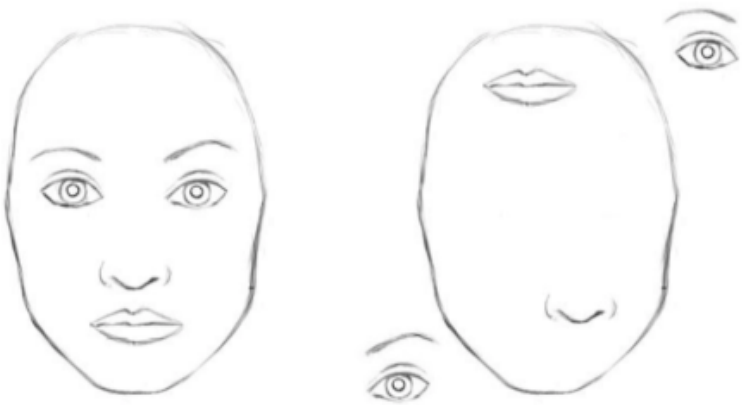


Simple feature와 Complex feature가 합쳐지고 이를 이용해 classification을 하게됨 -> 위치에 대한 정보는 무시하고 pixel이 담고 있는 정보에 집중하게되는 문제가 있다.

Capsule Network

Max-pooling





-> 더 넓은 FOV에서 spatial한 정보를 얻기 위해 MaxPooling을 이용하지만, 여전히 각각의 feature detector에 대한 위치 정보는 잊게된다. 따라서 오른쪽 그림처럼 눈,코,입의 위치가 뒤집어져도 CNN은 같은 얼굴이라고 인식하게된다.

-> Internal data representation of a CNN network does not take into account important spatial hierarchies between simple and complex objects.

Capsule Network

-> CapsNet에서는 이러한 문제를 해결하기 위해서 Capsule이라는 개념을 도입하였다.
-> 캡슐은 하나의 Entity가 되고 Entity는 여러 property로 구성되어있다.
→ Entity : 자전거를 타고 있는 사람
→ Property : type, scale, position, velocity

Capsule Network

-> 고차원 공간으로 entity를 보낸 후 눈,코,입 등의 공간적인 관계까지 고려하고 이 공간에서조차 제대로 된 얼굴로 인식된다면 '얼굴'이라고 판단함
-> MaxPooling대신 Dynamic Routing을 사용하여 오브젝트 파트 (눈, 코,입 등)들의 상대적 위치까지 조합할 수 있다.

(1) 하위 캡슐은 Dynamic Routing 과정을 통해 이를 가장 잘 처리할 수 있는 상위 캡슐로 연결된다.

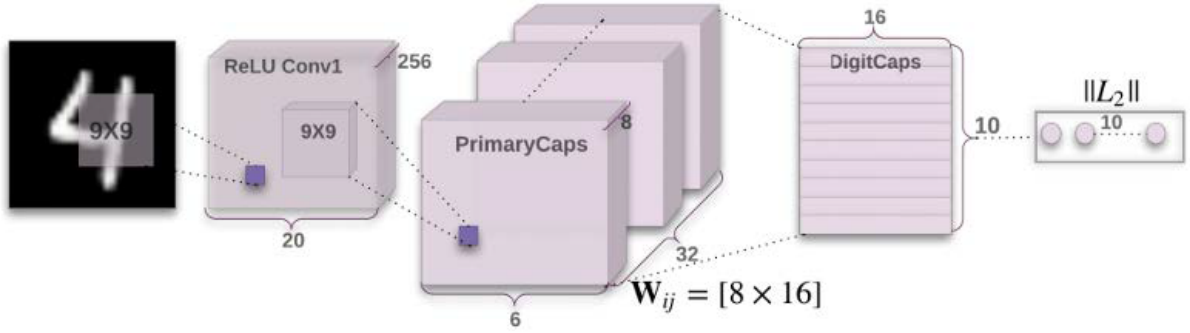
(2) 하위 캡슐에서는 local information이 place-coded 된다.

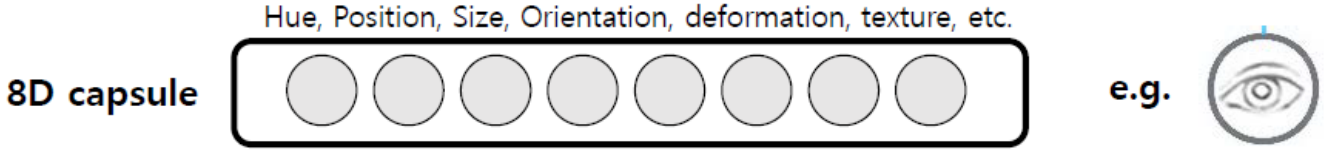
(3) 상위 캡슐에서는 훨씬 더 많은 positional information이 rate-coded 된다 (조합된다).

(4) Dynamic Routing 과정을 통해서 캡슐은 더 복잡한 entity 를 훨씬 자유롭게 표현이 가능하다.

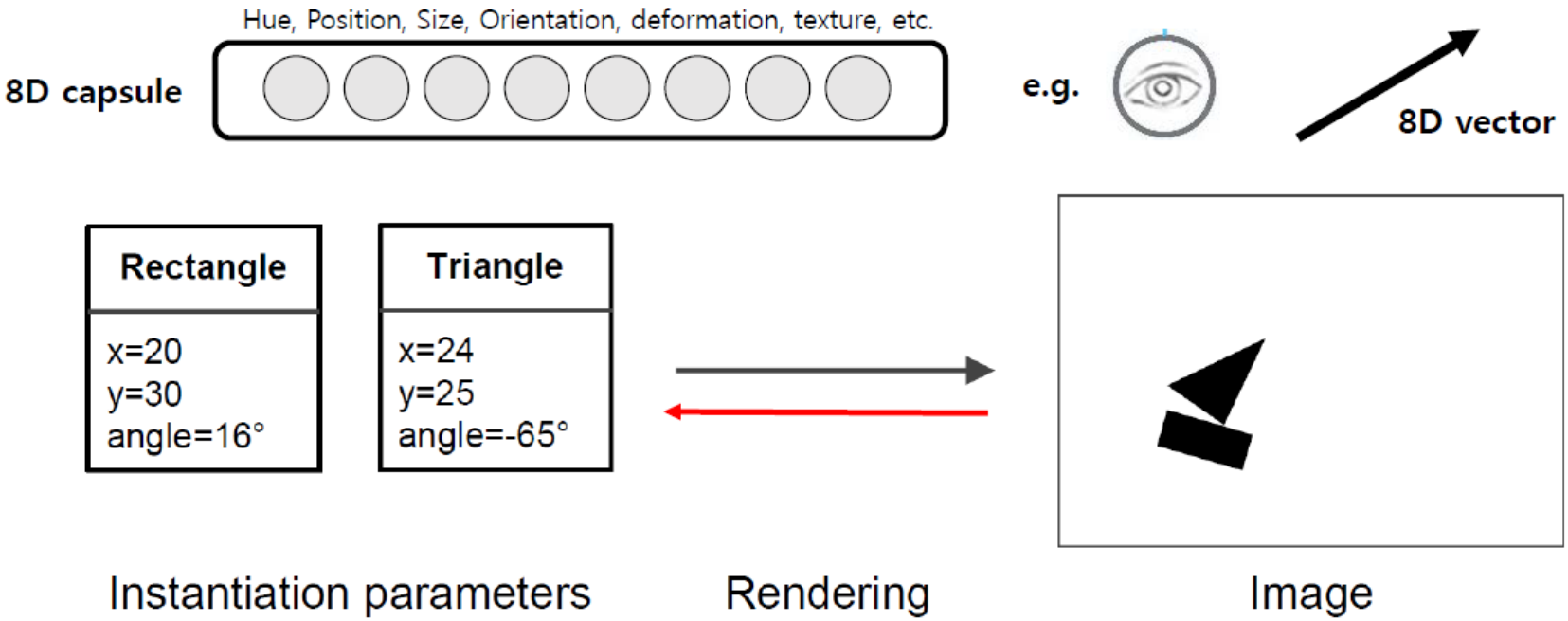
Capsules

"A capsule is a group of neurons whose activity vector represents the instantiation parameters of a specific type of entity such as an object or an object part."



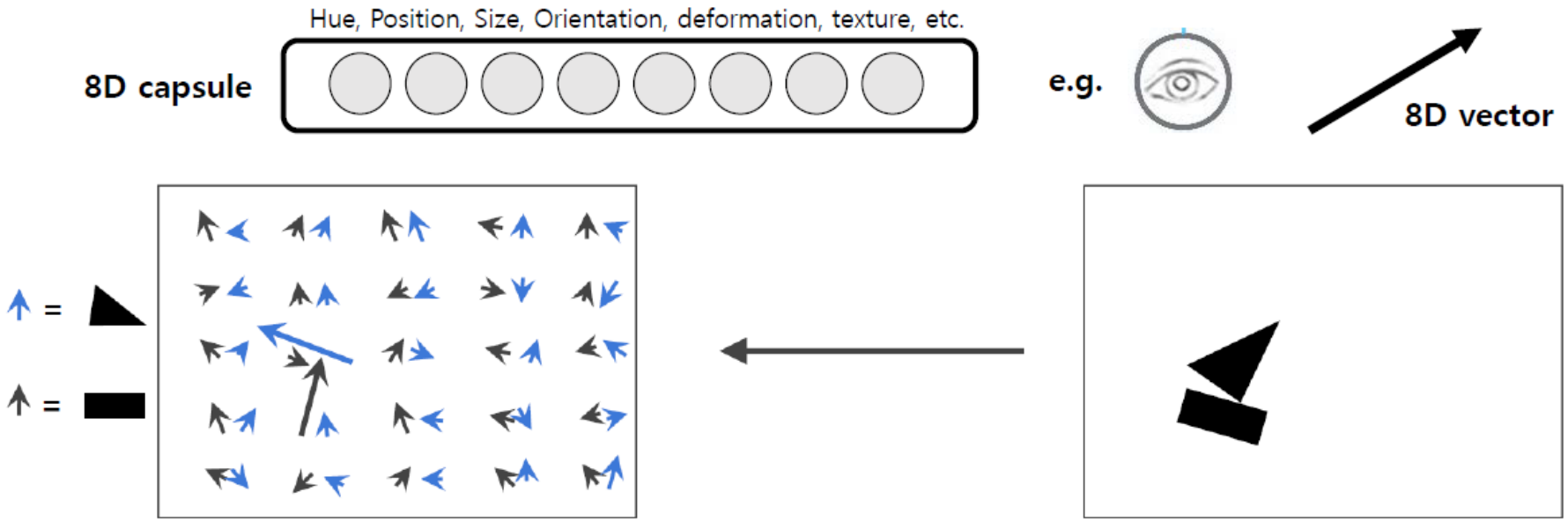


Capsules



Contents from https://www.slideshare.net/aureliengeron/introduction-to-capsule-networks-capsnets

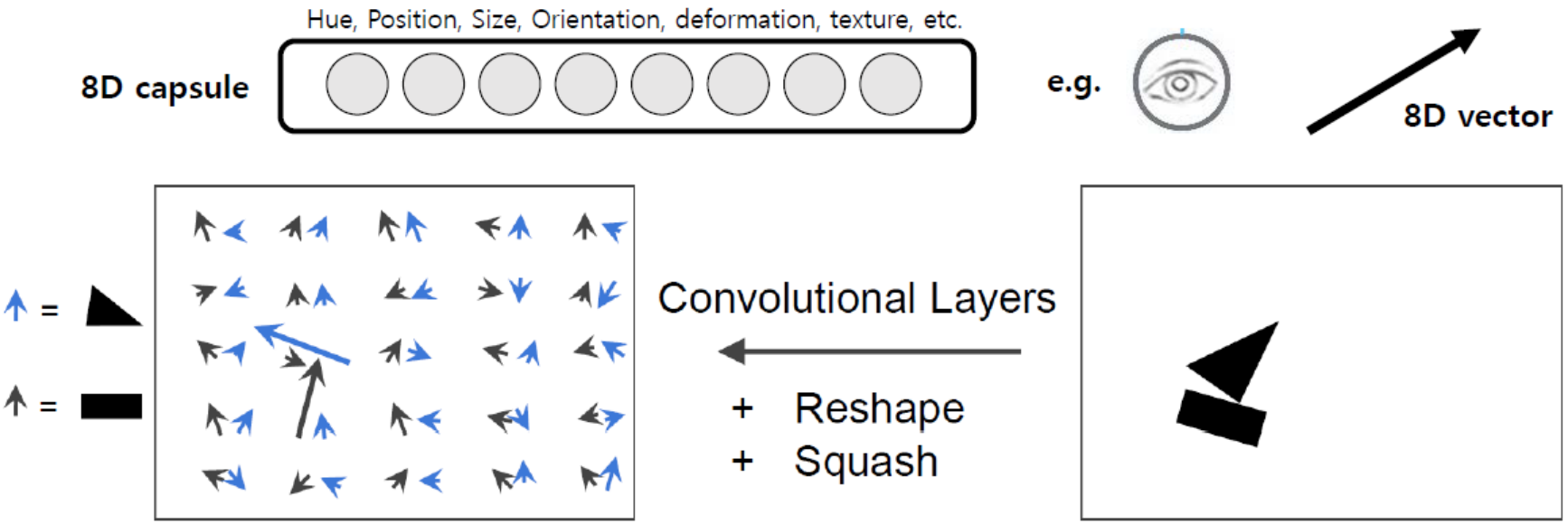
Capsules



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

Contents from https://www.slideshare.net/aureliengeron/introduction-to-capsule-networks-capsnets

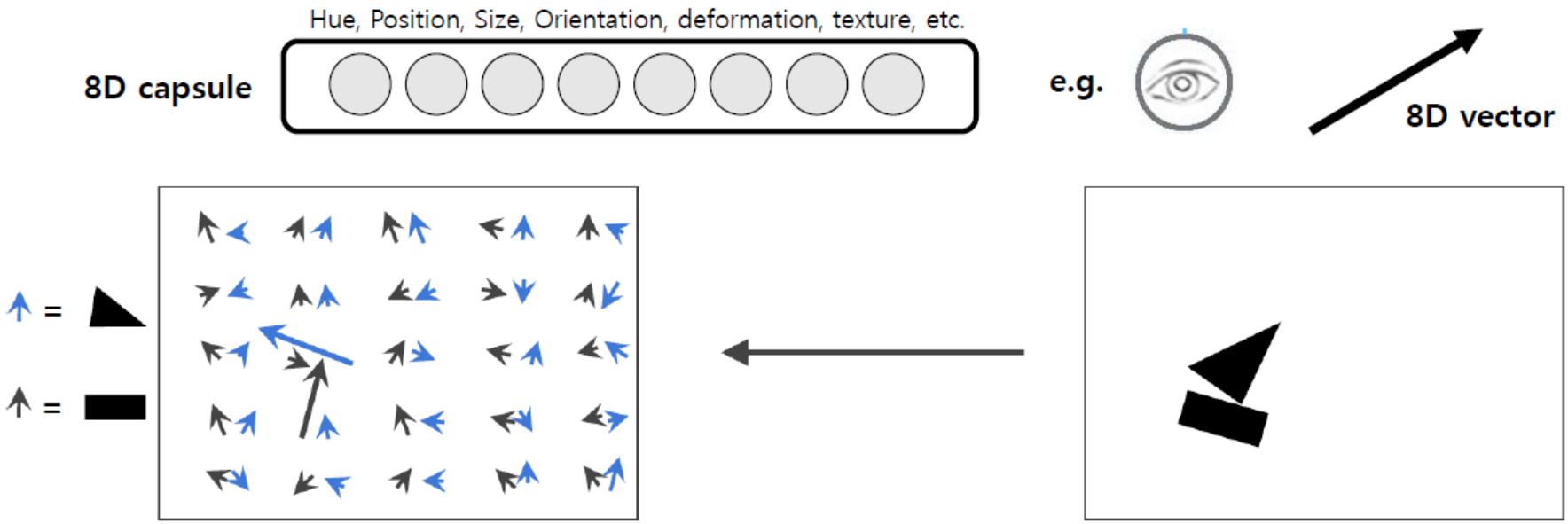
Capsules



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

Contents from https://www.slideshare.net/aureliengeron/introduction-to-capsule-networks-capsnets

Capsules



Equivariance of Capsules

Contents from https://www.slideshare.net/aureliengeron/introduction-to-capsule-networks-capsnets

Dynamic Routing Algorithm

Participants in the Algorithm

- **Prediction vector**
  - With the previous capsule output u_i and transformation matrix W_{ij}, we computer prediction vector as

$$\hat{u}_{j|i} = W_{ij} u_i$$

- **The capsule output of next layer**
  - Then the capsule output of the next layer v_j is computed as

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad \text{, where} \quad s_j = \sum_i c_{ij} \hat{u}_{j|i}$$

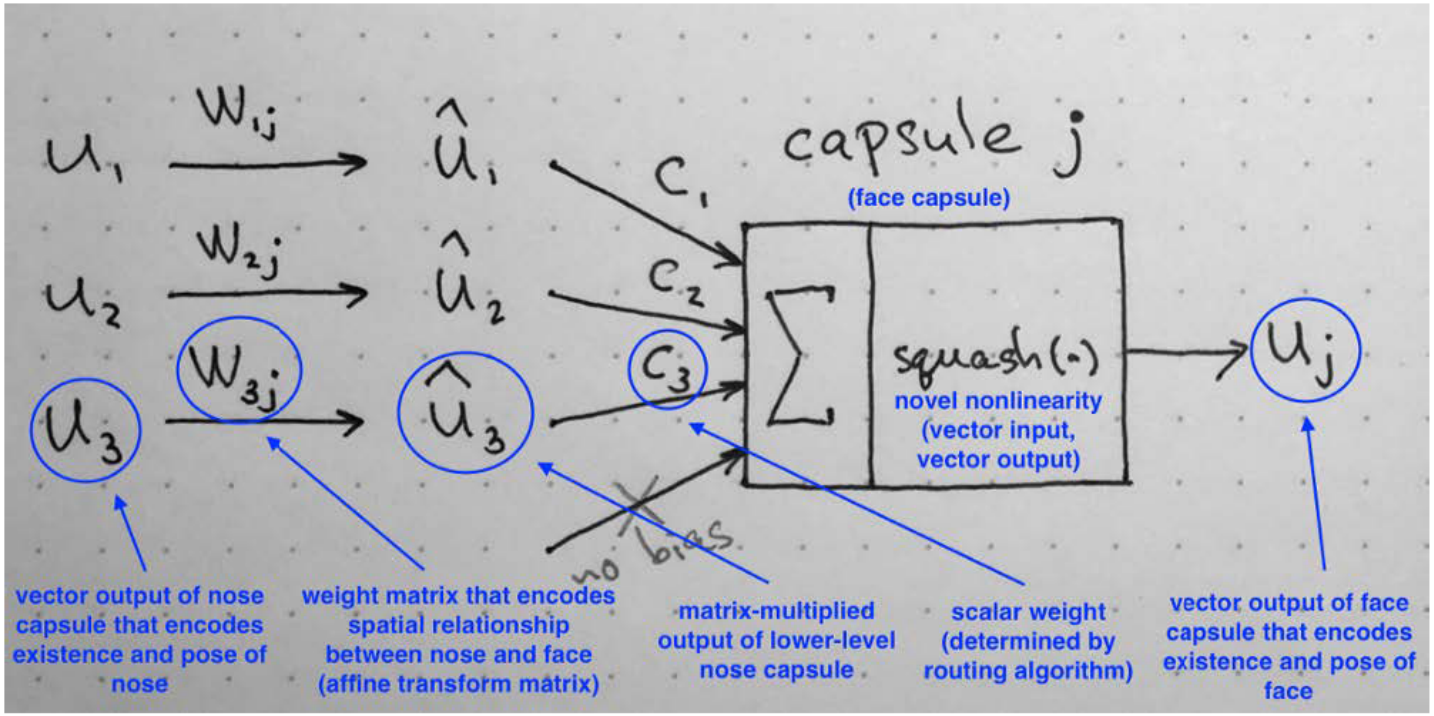
- **Coupling coefficient**
  - Here, c_{ij} are called coupling coefficient which is trained with dynamic routing algorithm.
  - We impose restriction that $\sum_i c_{ij} = 1$ which is achieved by softmax function of relevancy or similarity score b_{ij} which is initialized with zeros and progressively updated as follows(which reminds me of Hebbian learning rule):

$$similarity = \hat{u}_{j|i} \cdot v_j$$

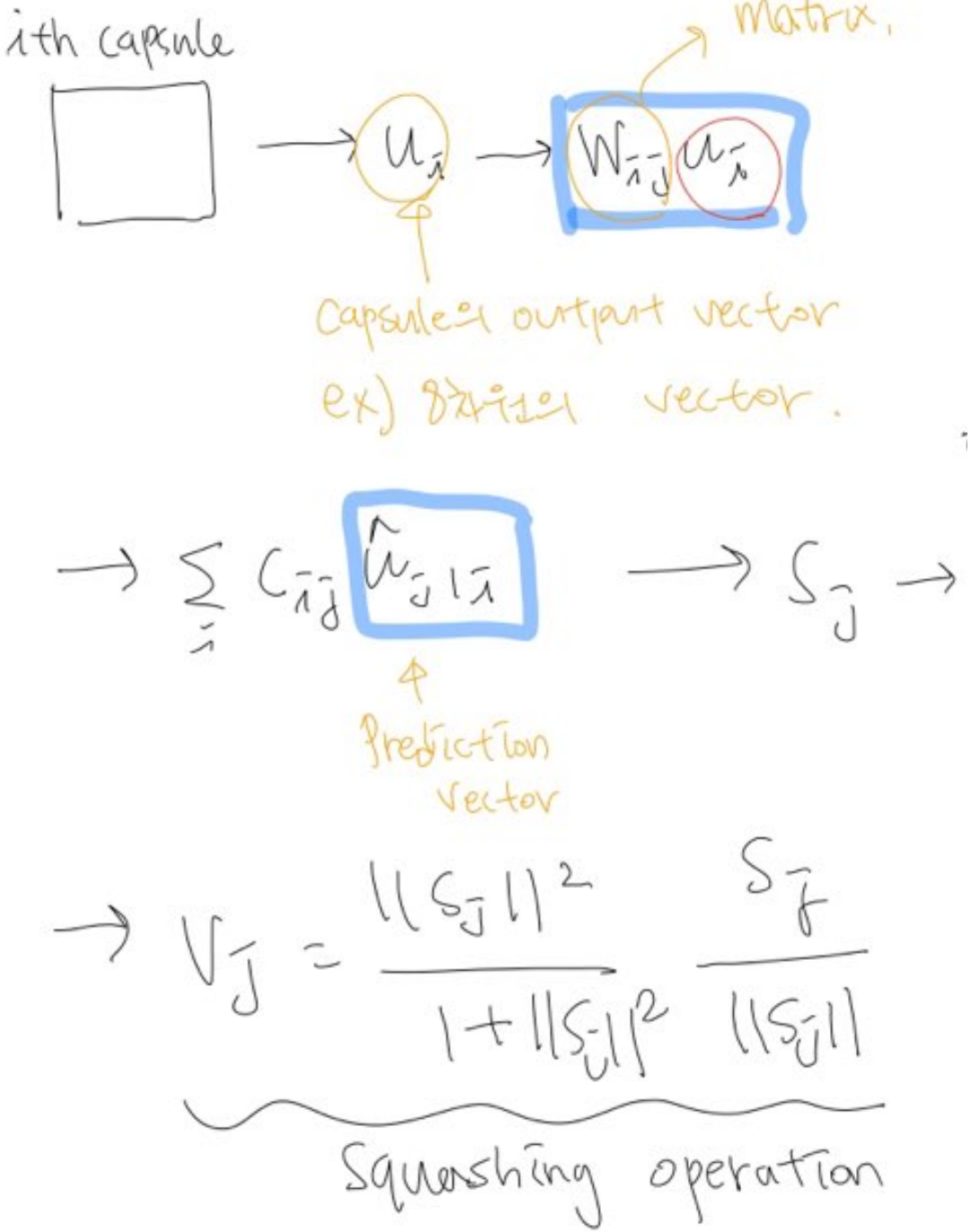
$$b_{ij} \leftarrow b_{ij} + similarity$$

$$c_{ij} = \frac{\exp b_{ij}}{\sum_k \exp b_{ik}}$$

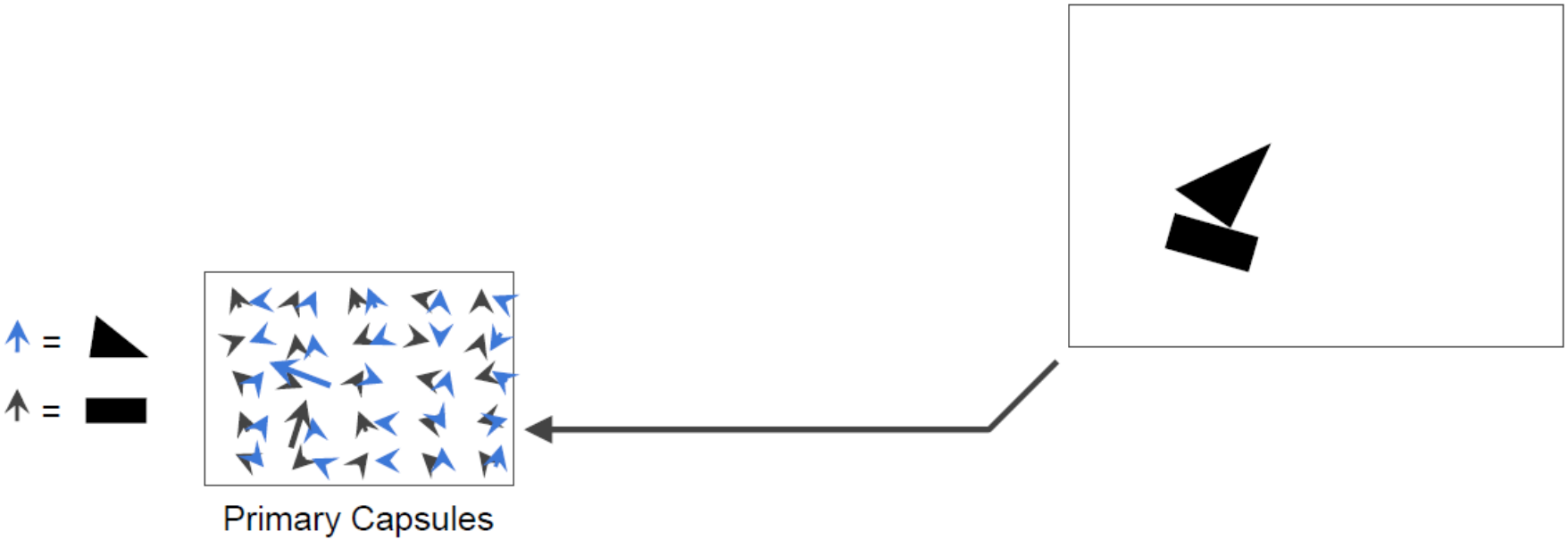
Routing by Agreements



Contents from https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks-part-iii-dynamic-routing-between-capsules-349f6d30418

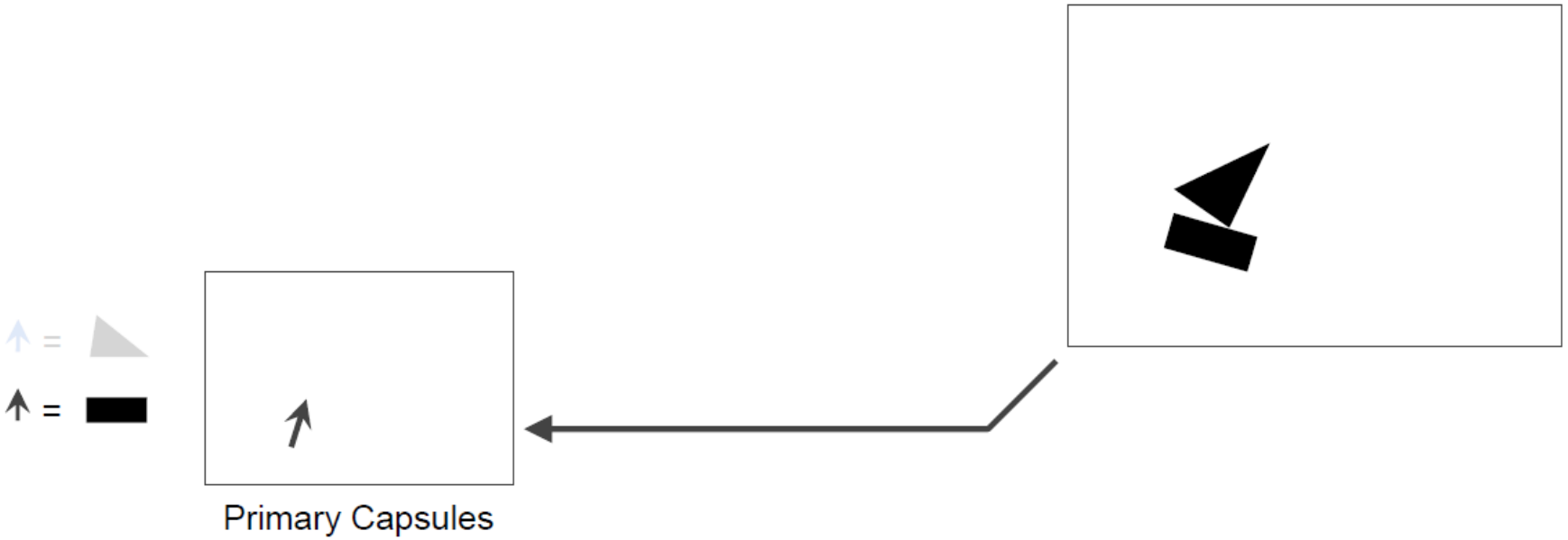

ith capsule
Matrix.
u_i
W_{ij} u_i
Capsule의 output vector
ex) 8차원의 vector.
$\sum_i c_{ij} \hat{u}_{j|i}$
s_j
Prediction vector
$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|}$
Squashing operation

Primary Capsules



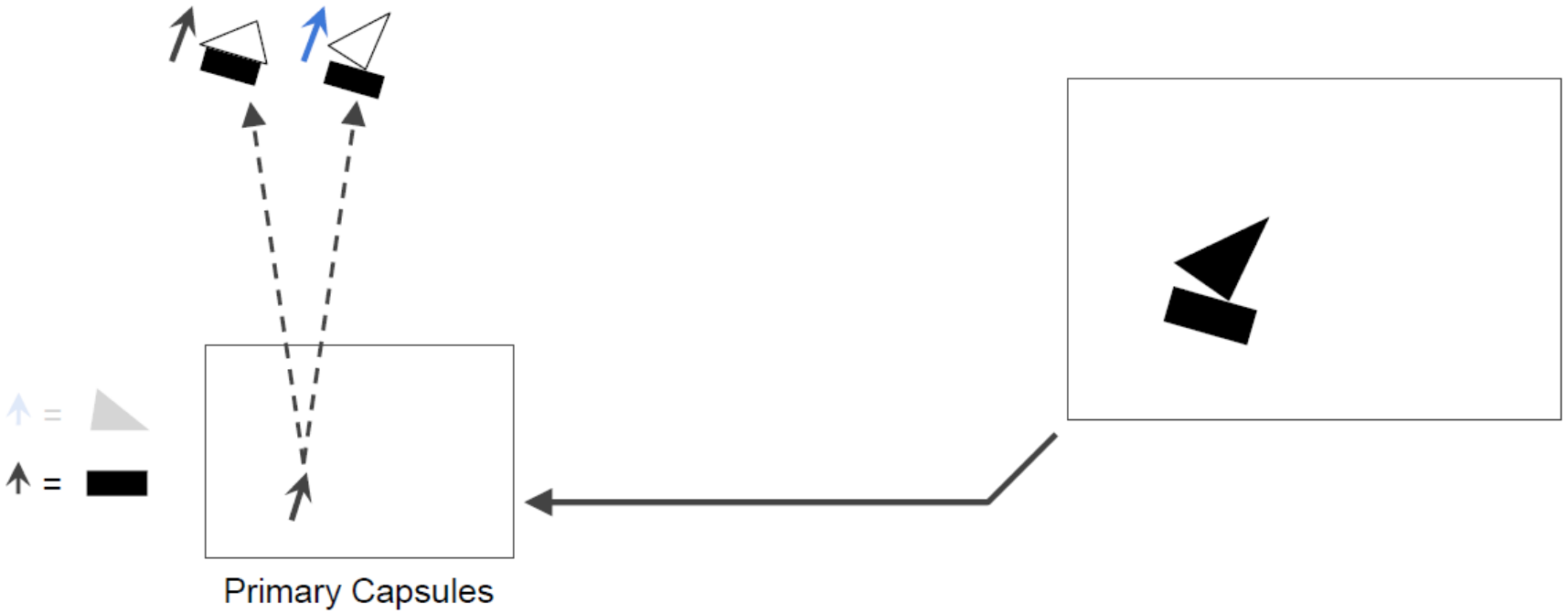
Aurélien Géron, 2017

Predict Next Layer's Output



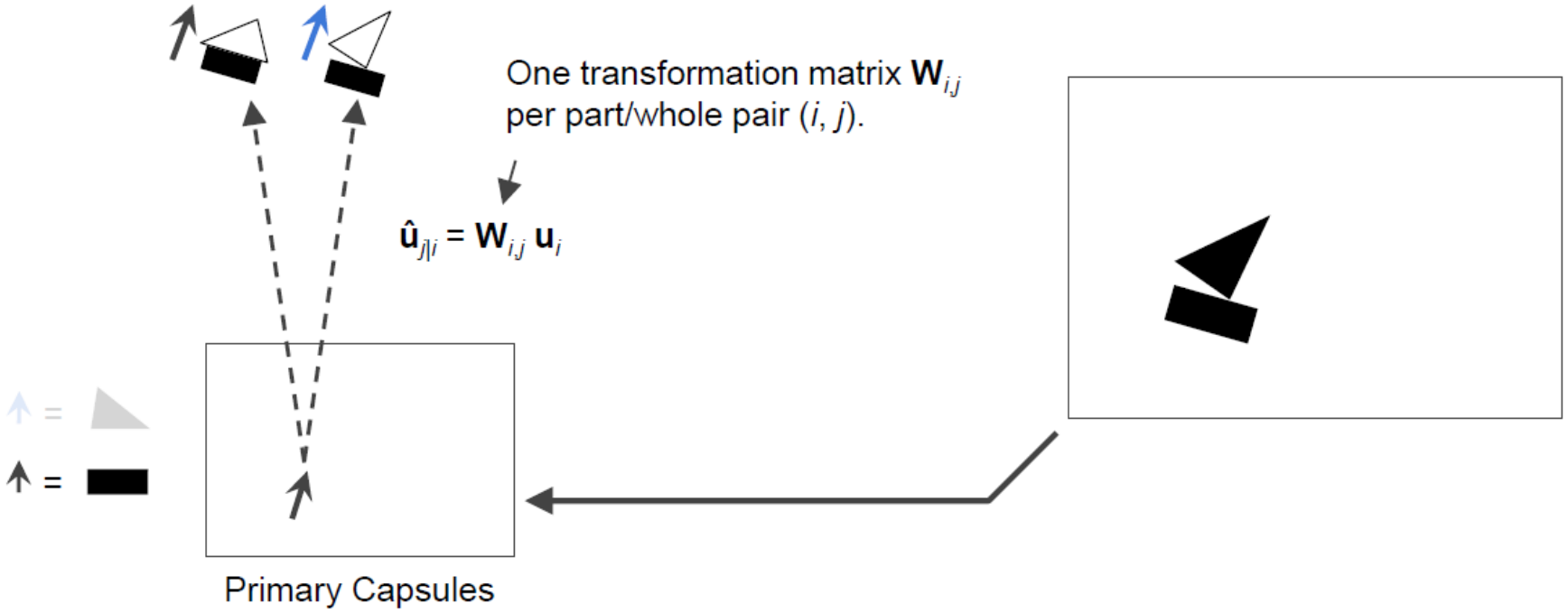
Aurélien Géron, 2017

Predict Next Layer's Output



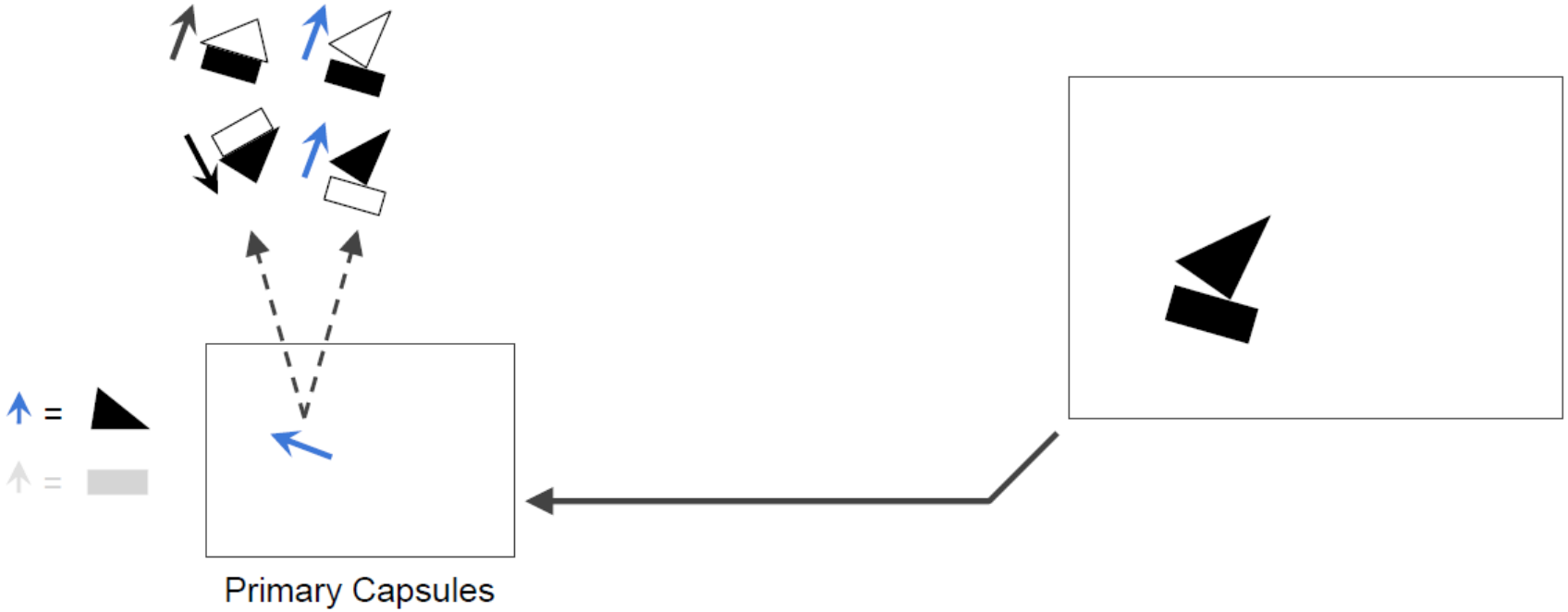
Aurélien Géron, 2017

Predict Next Layer's Output

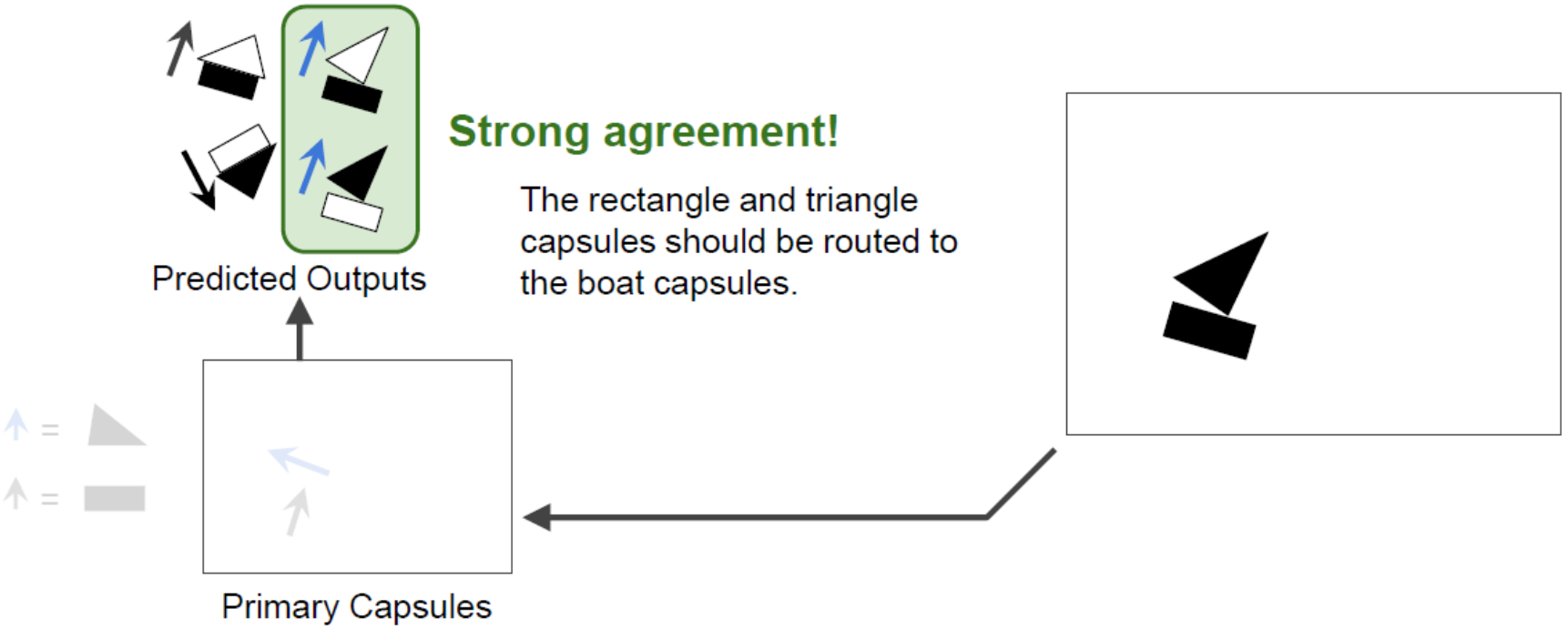


Aurélien Géron, 2017

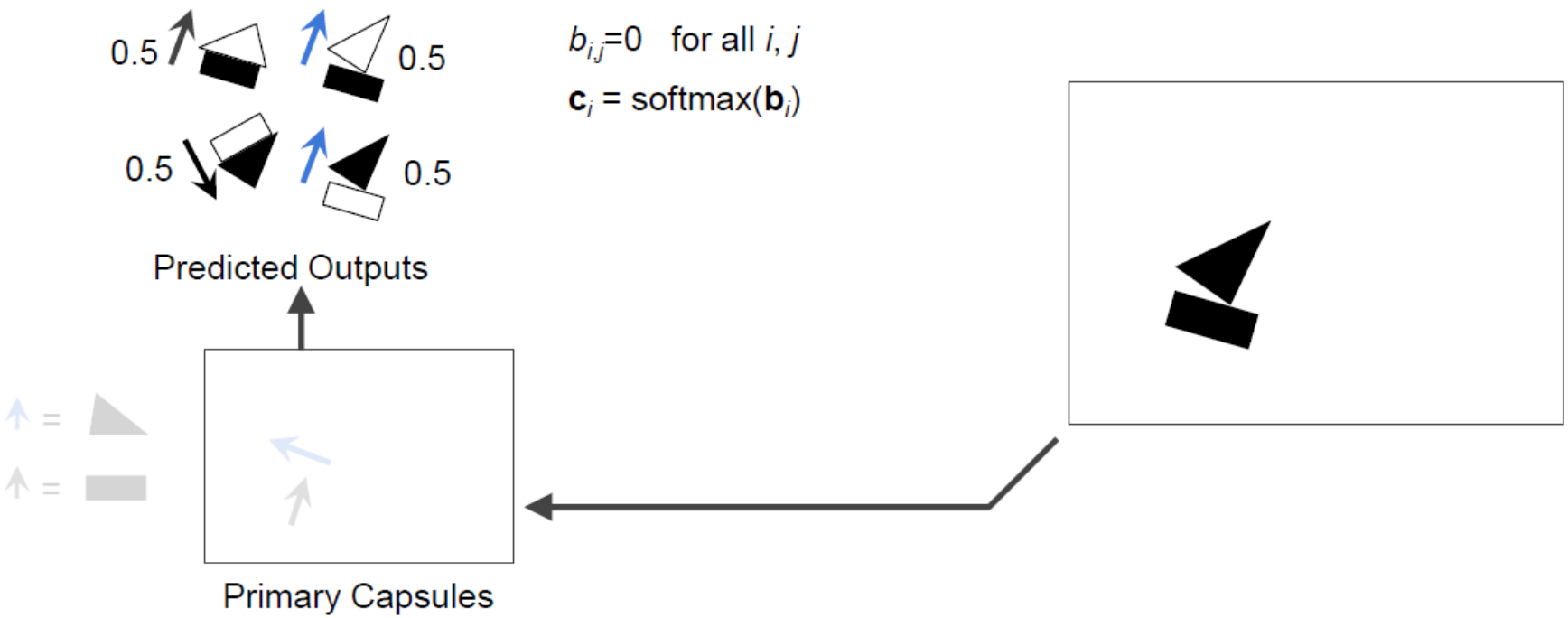
Predict Next Layer's Output



Routing by Agreement



Routing Weights

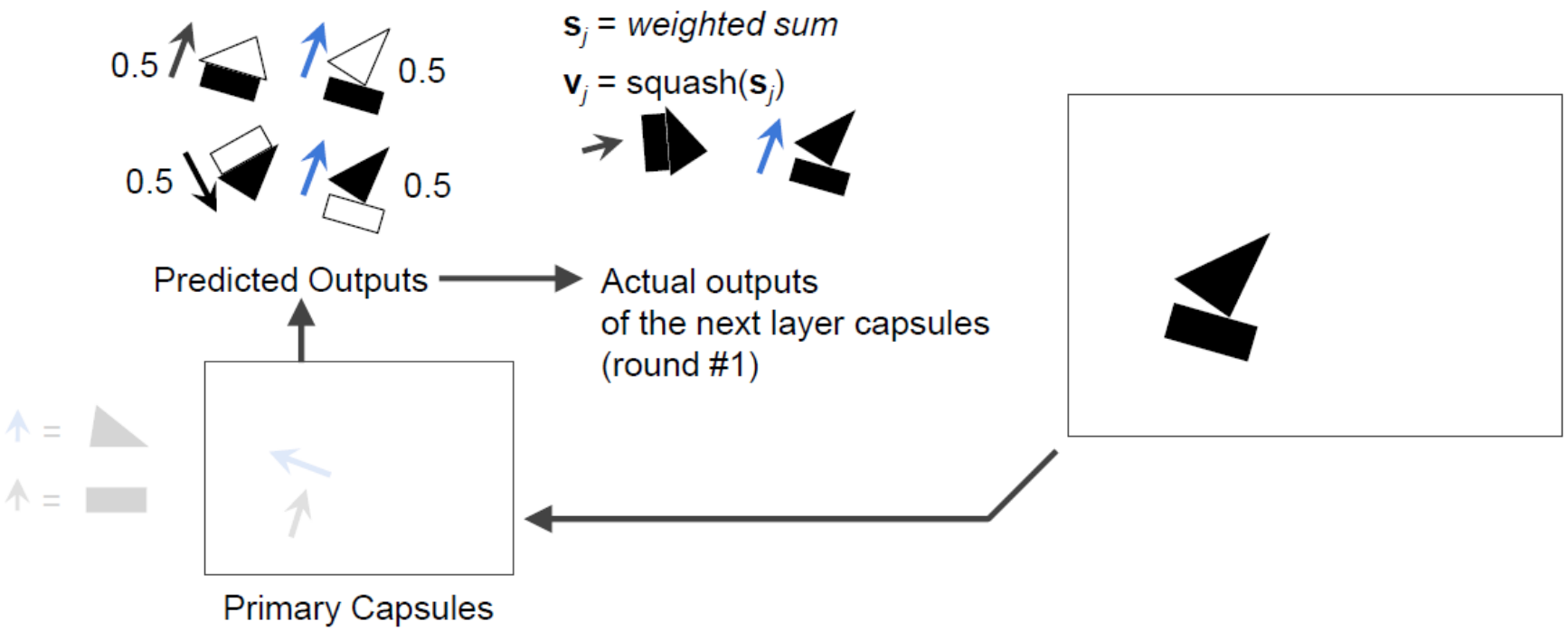


$b_{i,j}=0$ for all i, j

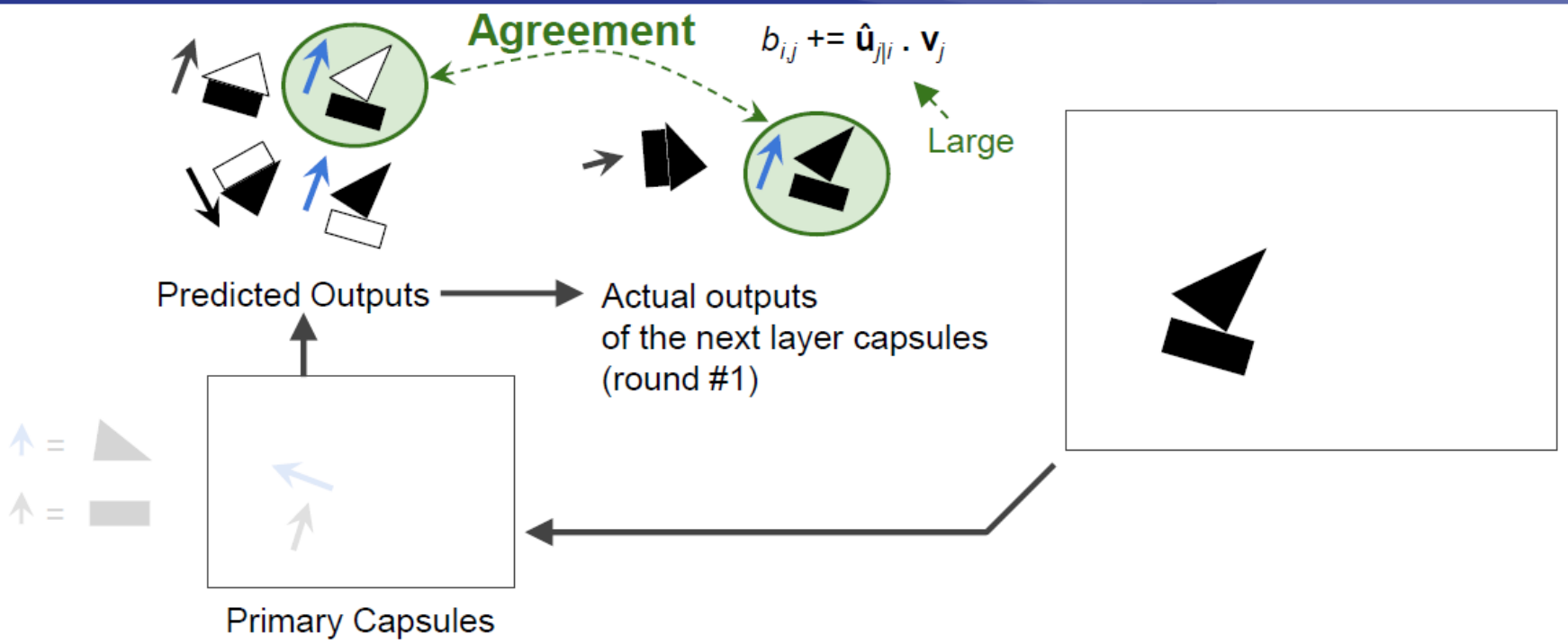
$\mathbf{c}_i = \text{softmax}(\mathbf{b}_i)$

Aurélien Géron, 2017

Compute Next Layer's Output

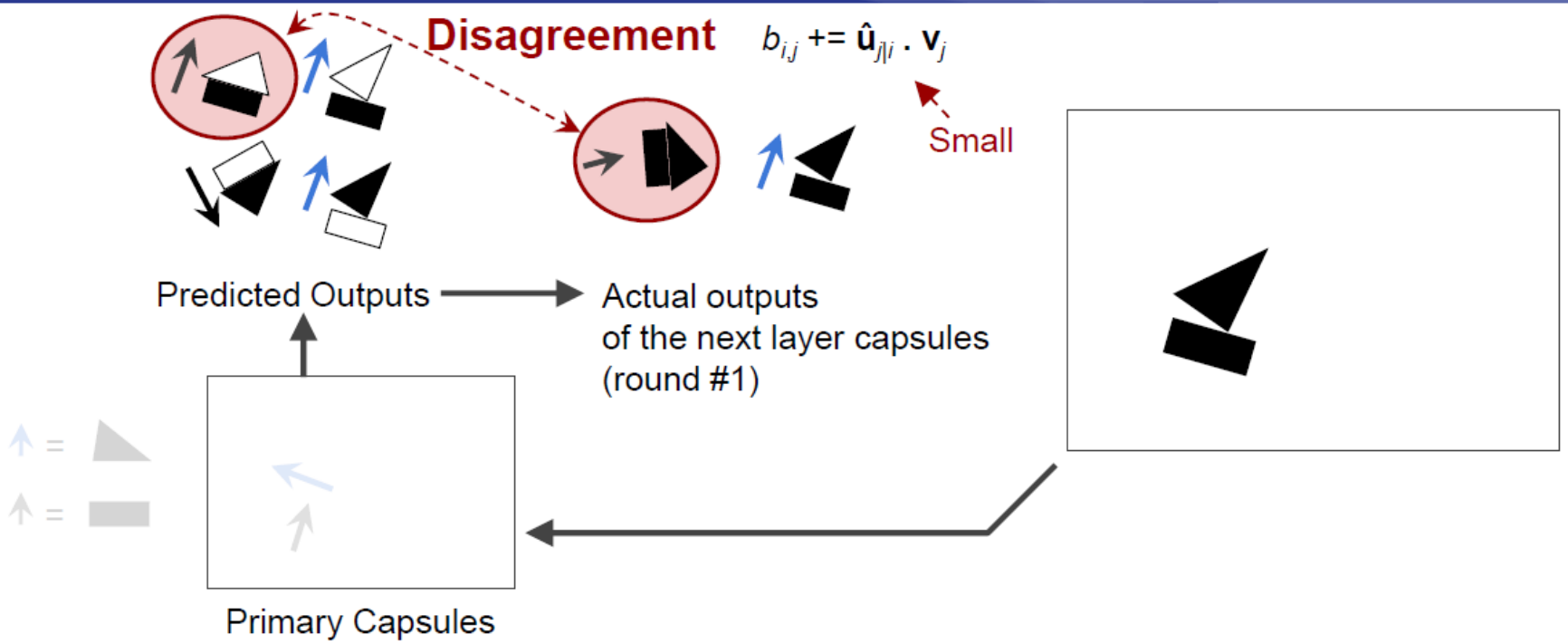


Update Routing Weights

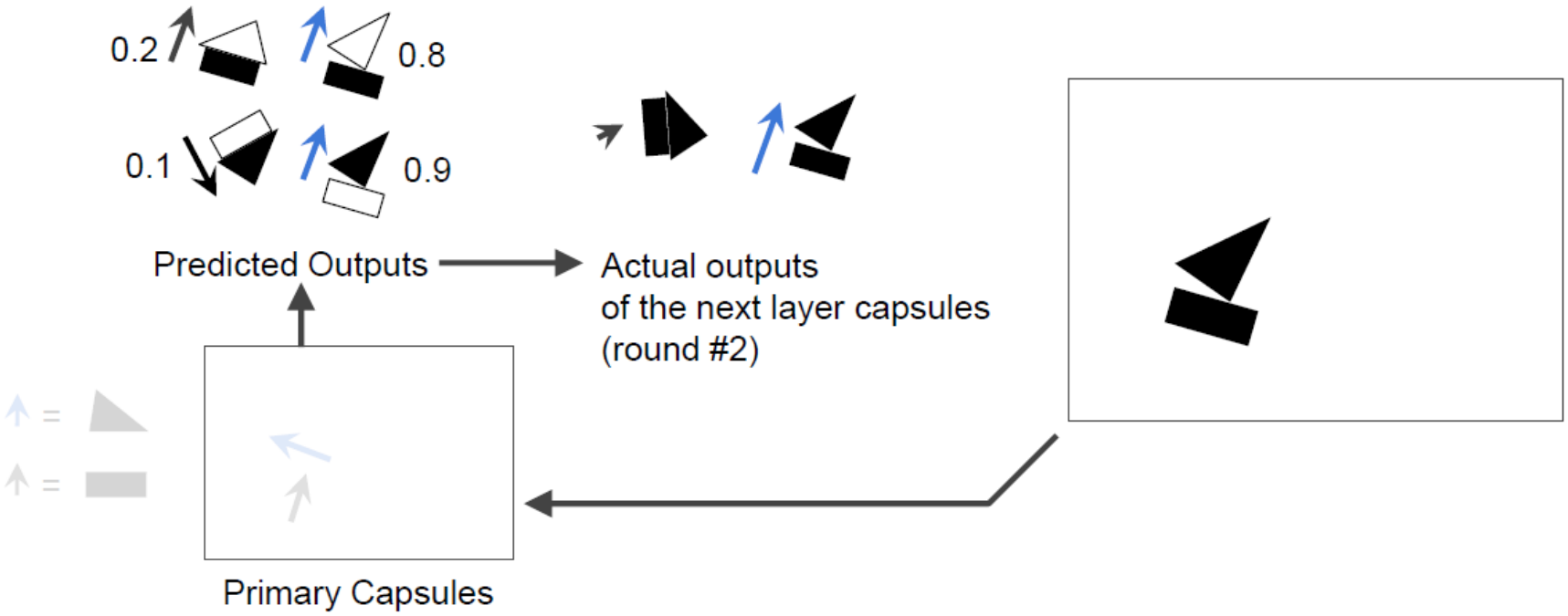


Aurélien Géron, 2017

Update Routing Weights

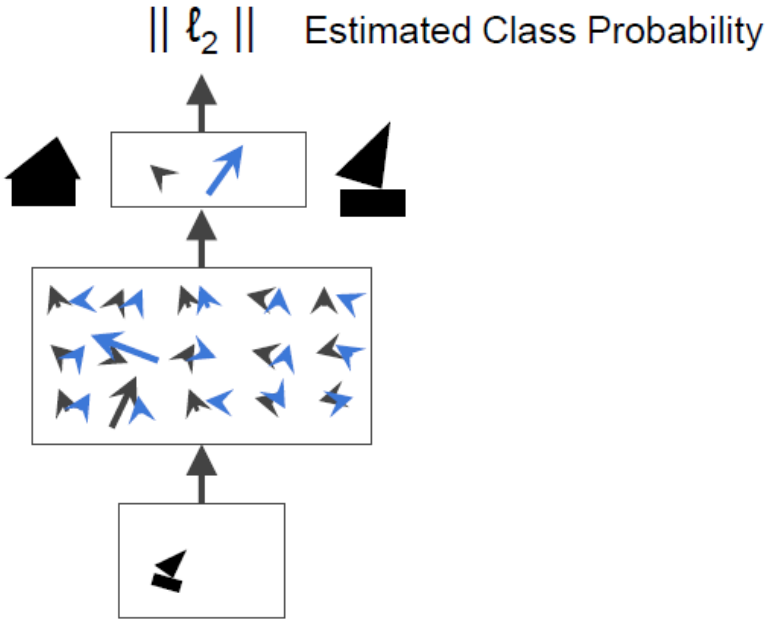


Compute Next Layer's Output



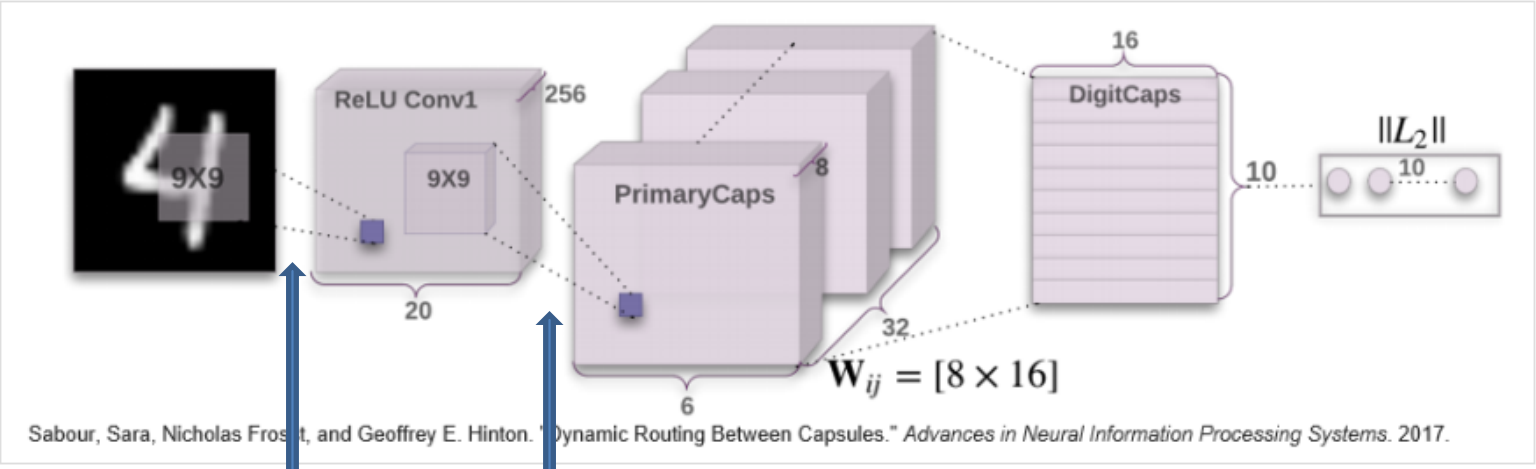
Aurélien Géron, 2017

Classification
CapsNet



Capsule Network

CapsNet architecture

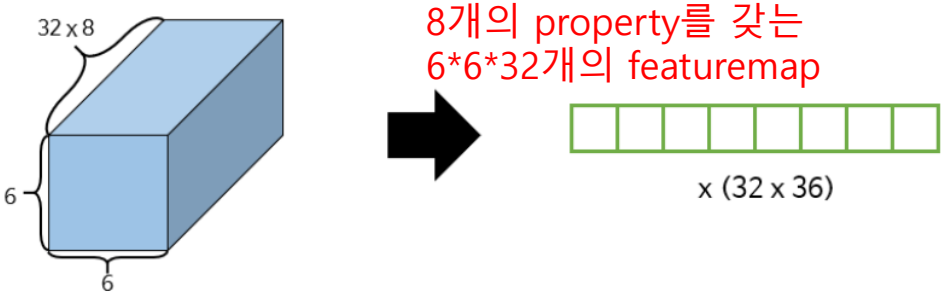


Sabour, Sara, Nicholas Frost, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules." *Advances in Neural Information Processing Systems*. 2017.

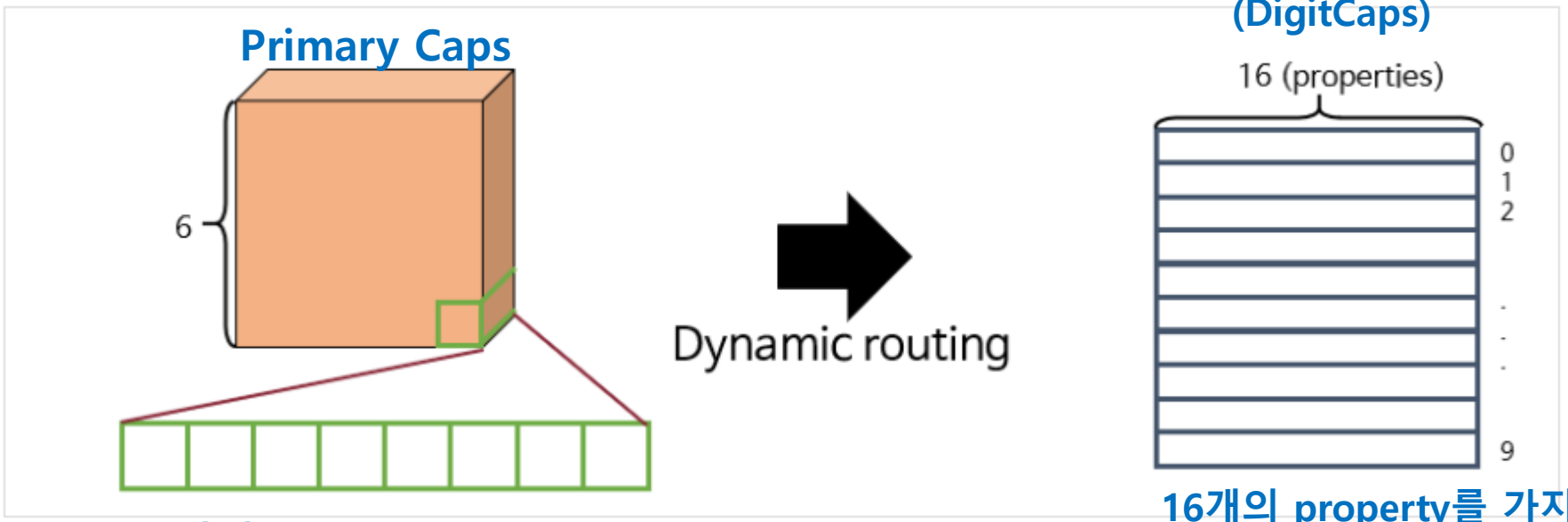
Convolution

Convolution
& Reshape

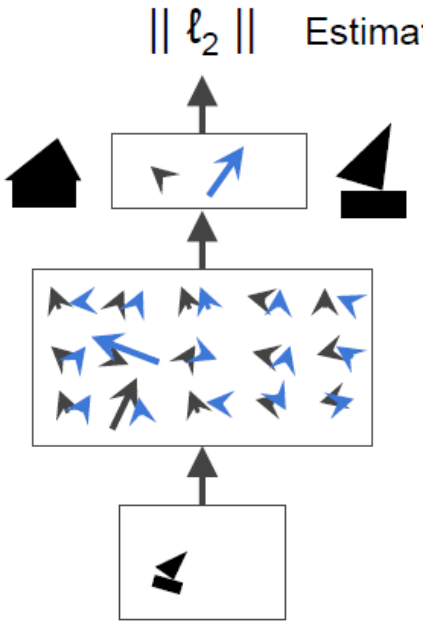
- Reshape output : [6, 6, 32 x 8] → [6 x 6 x 32, 8, 1]



Capsule Network



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:

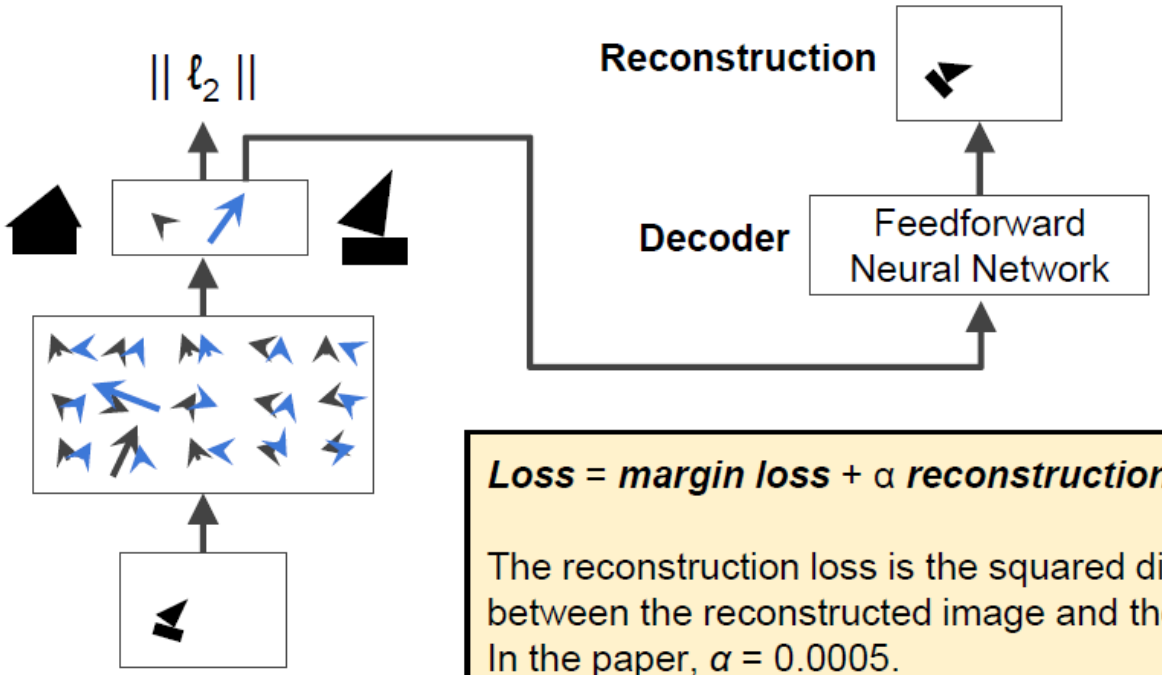
$m^- = 0.1$

$m^+ = 0.9$

$\lambda = 0.5$

Aurélien Géron, 2017

Regularization by Reconstruction

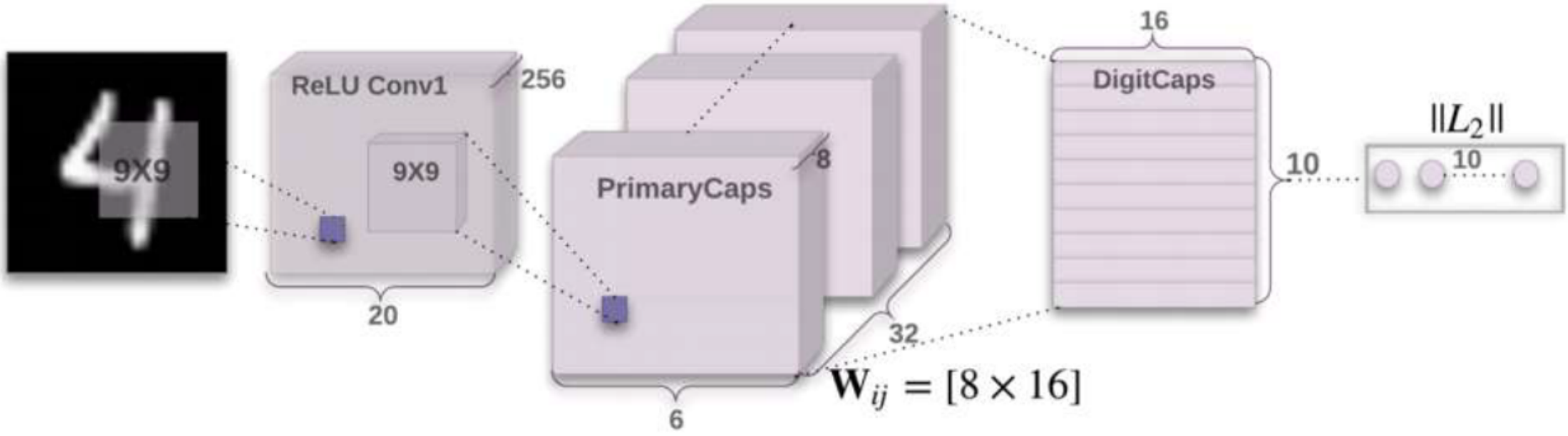


Loss = ***margin loss*** + α ***reconstruction loss***

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

Aurélien Géron, 2017

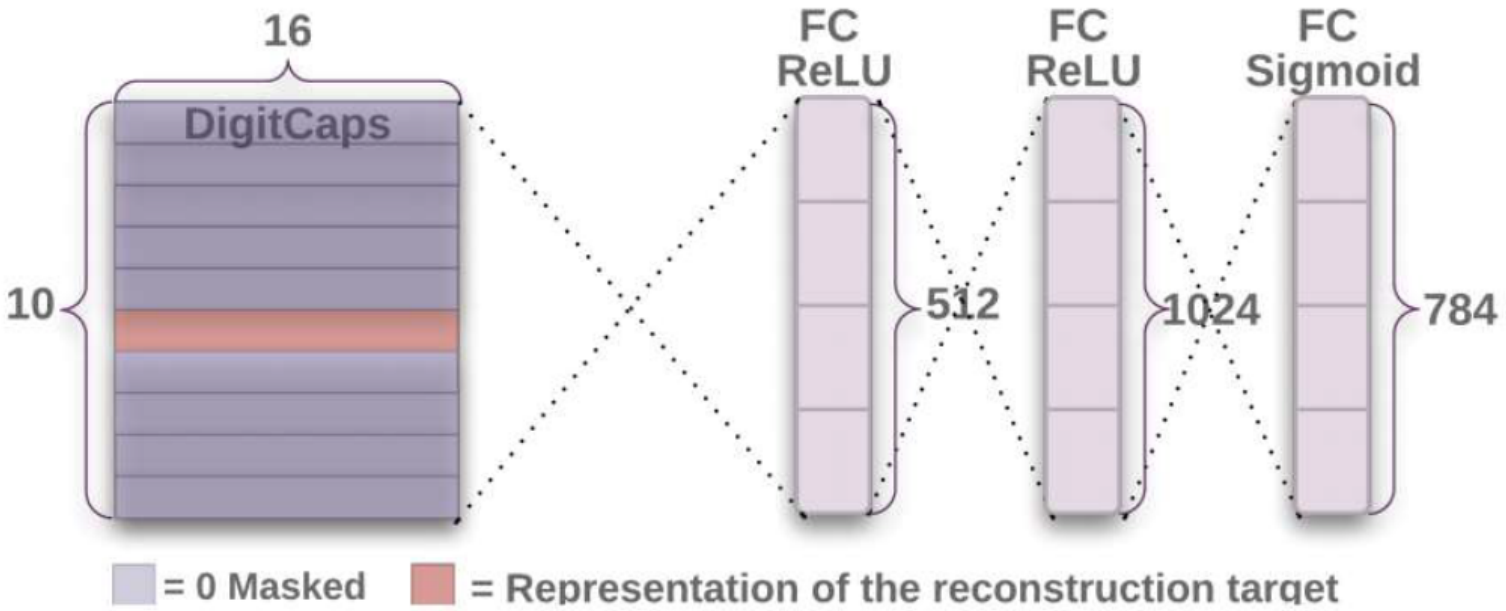
A CapsNet for MNIST



(Figure 1 from the paper)

Aurélien Géron, 2017

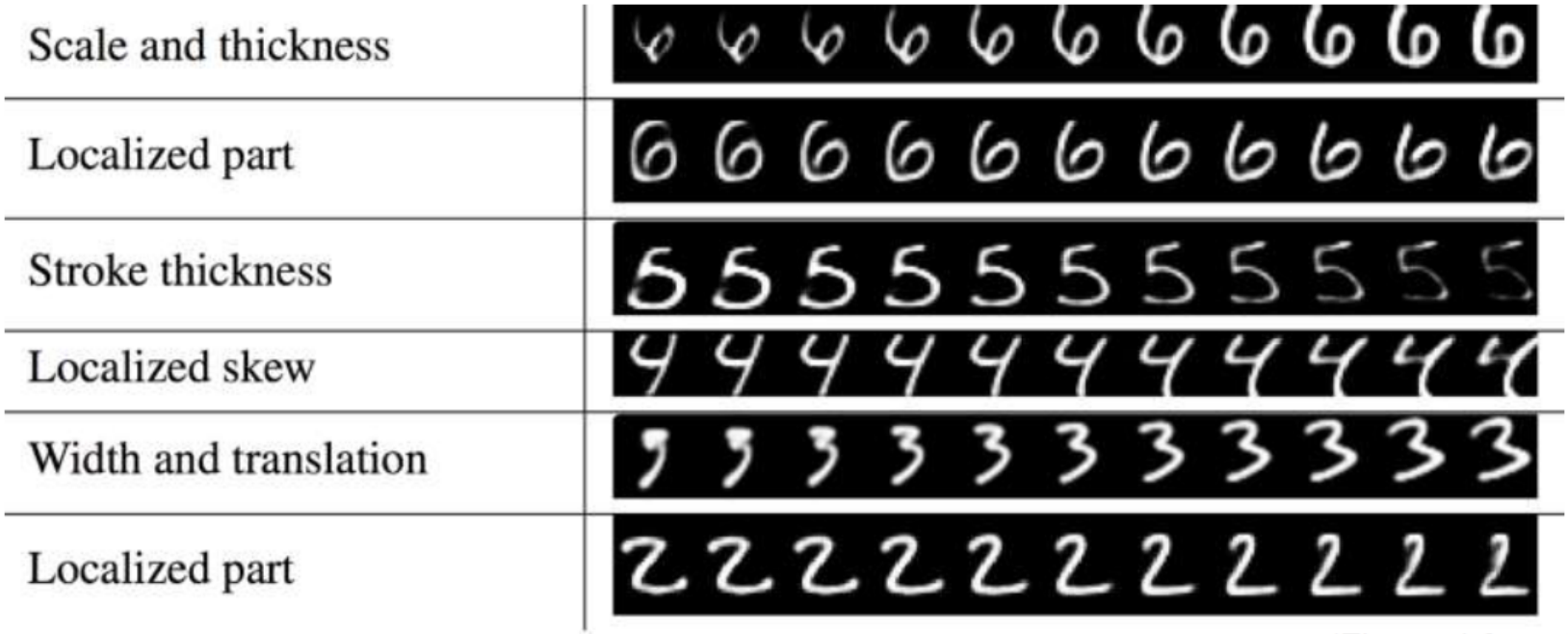
A CapsNet for MNIST – Decoder



(Figure 2 from the paper)

Aurélien Géron, 2017

Interpretable Activation Vectors



(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

References

(1) Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems* (pp. 3856-3866).
(2) https://www.youtube.com/watch?v=_YT_8CT2w_Q (PR12 : Capsule Network 발표영상)
(3) https://www.youtube.com/watch?v=pPN8d0E3900 (Capsule Network 튜토리얼 영상)