Dynamic Routing Between Capsules

Sara Sabour, *Nicholas Frosst*, and *Geoffrey Hinton*

NIPS 2017

Erfan Loweimi

Centre for Speech Technology Research (CSTR)
University of Edinburgh
23, Apr., 2019

Capsule Papers ...

#Citations ← 21, 04, 2019

[PDF] Transforming Auto-encoders - University of Toronto Computer Science
https://www.cs.toronto.edu/~fritz/absps/transauto6.pdf
by GE Hinton - Cited by 356 - Related articles
Three capsules of a **transforming auto-encoder** that models translations. Each capsule in the figure has 3 recognition units and 4 generation units. The weights ...

ICANN 2011

[PDF] Dynamic Routing Between Capsules - NIPS Proceedings
https://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf
by S Sabour - 2017 - Cited by 551 - Related articles
Dynamic Routing Between Capsules. Sara Sabour. Nicholas Frosst. Geoffrey E. Hinton. Google Brain. Toronto. {sasabour, frosst, geoffhinton}@google.com.

NIPS 2017

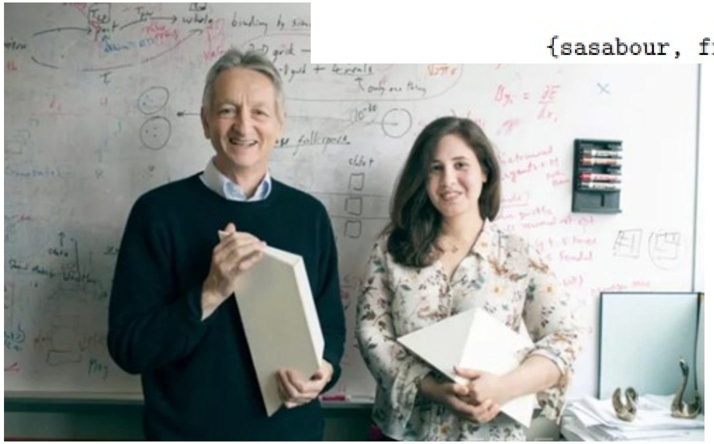
[PDF] matrix capsules with em routing - OpenReview
https://openreview.net/pdf?id=HJWLfGWRb
by GE Hinton - 2018 - Cited by 101 - Related articles
MATRIX CAPSULES WITH EM ROUTING. GeoffreyHinton, SaraSabour, NicholasFrosst. Google Brain. Toronto, Canada. {geoffhinton, sasabour, frosst}@google.

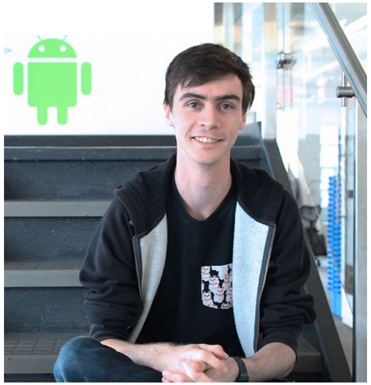
ICLR 2018

Dynamic Routing Between Capsules

Sara Sabour Nicholas Frosst

Geoffrey E. Hinton
Google Brain
Toronto
{sasabour, frosst, geoffhinton}@google.com





[PDF] Dynamic Routing Between Capsules - NIPS Proceedings

https://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf

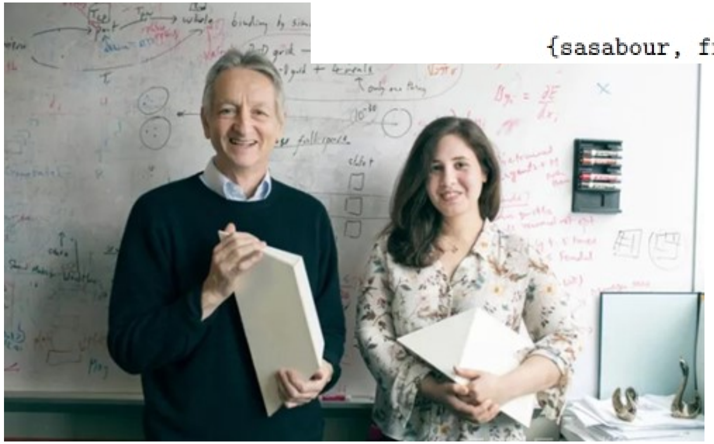
by S Sabour - 2017 - Cited by 557 - Related articles

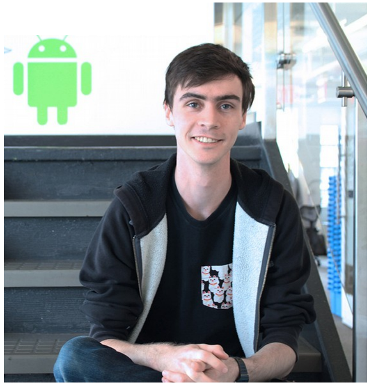
Dynamic Routing Between Capsules. Sara Sabour. Nicholas Frosst. Geoffrey E. Hinton. Google Brain. Toronto. {sasabour, frosst, geoffhinton}@google.com.

Dynamic Routing Between Capsules

Sara Sabour **Nicholas Frosst**

Geoffrey E. Hinton
Google Brain
Toronto
{sasabour, frosst, geoffhinton}@google.com





[PDF] Dynamic Routing Between Capsules - NIPS Proceedings
https://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf
by S Sabour - 2017 - Cited by 557 - Related articles

23, 4, 2019

Dynamic Routing Between Capsules. Sara Sabour. Nicholas Frosst. Geoffrey E. Hinton. Google Brain. Toronto. {sasabour, frosst, geoffhinton}@google.com.

Outlines

- Pros/Cons of CNNs
- CapsNet aims to solving two problems ...
- Routing mechanism
- Experimental Results
- Challenges
- Wrap-up

Convolutional Neural Networks (CNN)

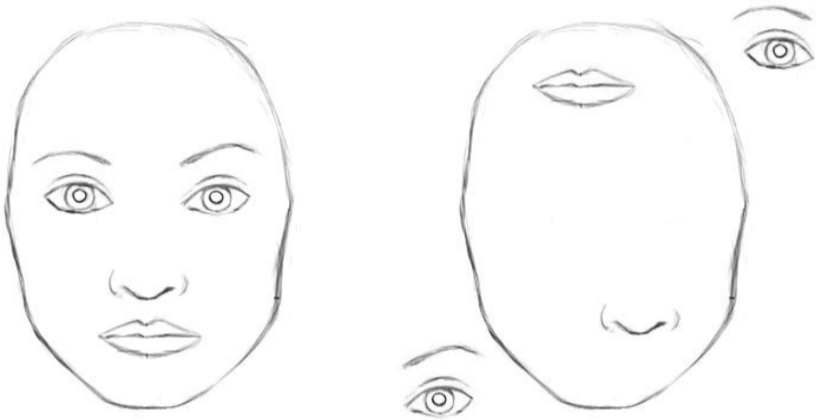
- Main components:
  - Feature detectors, interleaved with subsampling layers
- CNNs work best for recognition
  - Weight sharing
  - Sparsity of connections

Convolutional Neural Networks (CNN)

- Main components:
  - Feature detectors, interleaved with subsampling layers
- CNNs work best for recognition
  - Weight sharing
  - Sparsity of connections
- CNNs afford some translation invariance to small changes
  - Replicating the feature detectors (learned knowledge) across image
  - Max-pooling

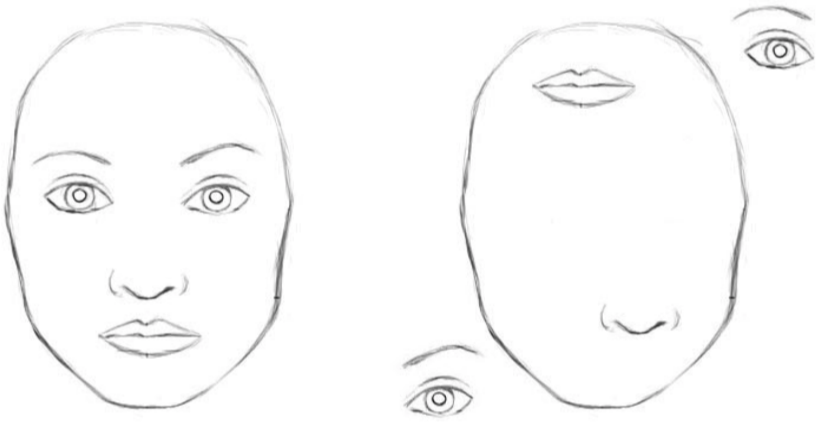
CNNs Problems (1)

- Picasso Problem → Right parts in wrong position
  - Mere existence of parts means whole



CNNs Problems (1)

- Picasso Problem → Right parts in wrong position
  - Mere existence of parts means whole
    - **OK**-ish for classification, **BAD** for segmentation/localisation



CNNs Problems (2)

- No built-in mechanism to extrapolate their understanding (internal representation) to radically new viewpoints

CNNs Problems (2)

- No built-in mechanism to extrapolate their understanding (internal representation) to radically new viewpoints
  - Only can deal with this through a lot of training data

Max-pooling is the Culprit ...

- Max-pooling along with replicating filters (knowledge) leads to some translation/rotation invariance

Max-pooling is the Culprit ...

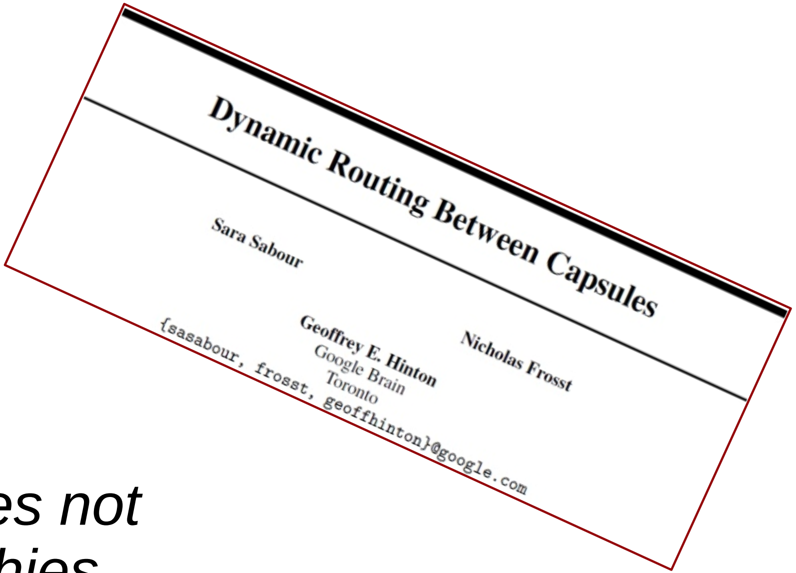
- Along with replicating filters (knowledge) leads to some translation/rotation invariance

- Most active neuron are routed to the higher level ...
  - Without considering the higher level activities (hierarchy)

- Discard information about precise position and relative spatial relationships

Max-pooling is the Culprit ...

“The pooling operation used in CNNs is a big mistake and the fact that it works so well is disaster.”


Dynamic Routing Between Capsules

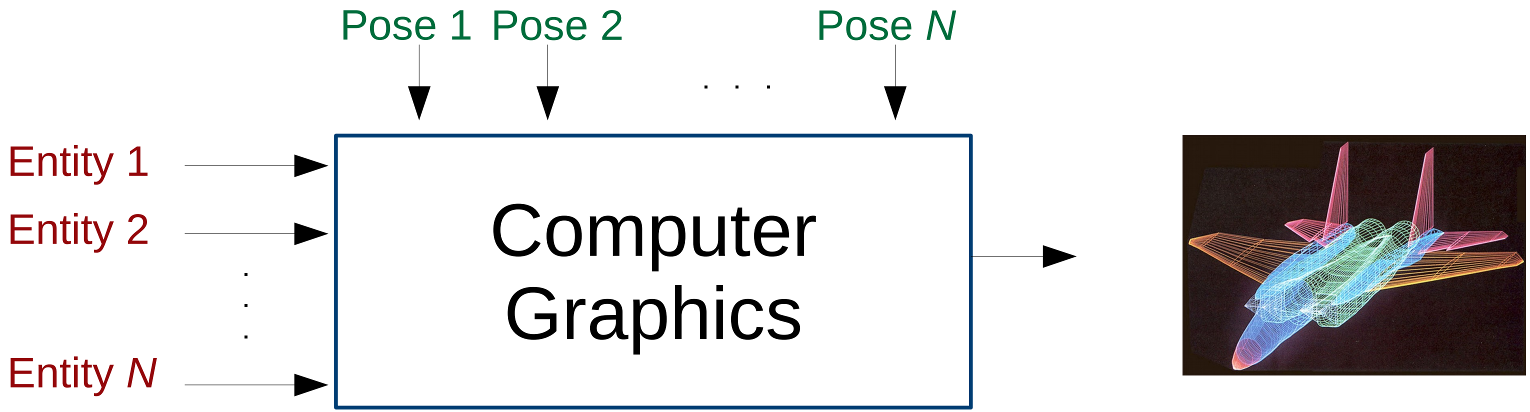
Sara Sabour   Nicholas Frosst   Geoffrey E. Hinton
Google Brain
Toronto
{sasabour, frosst, geoffhinton}@google.com

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


DETOUR

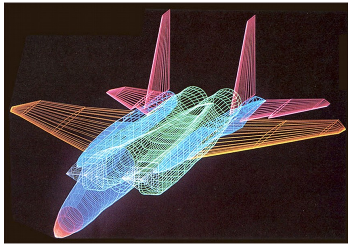
Computer Graphics

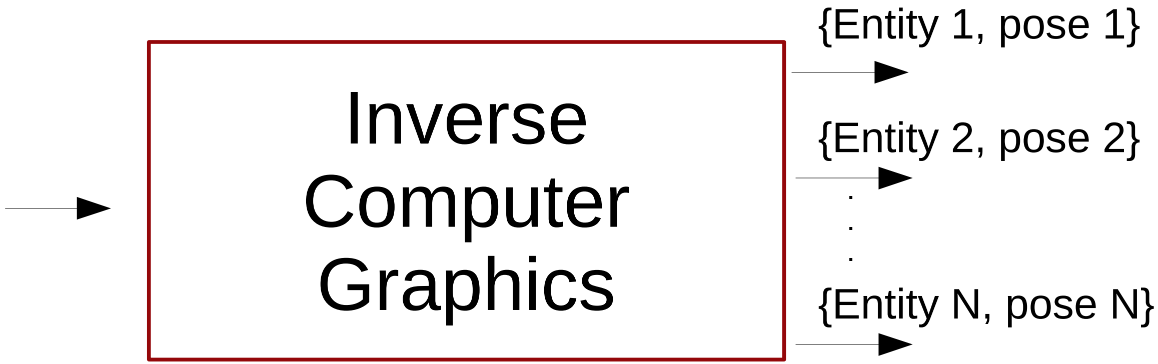
- Entities + Instantiation Parameters → Synthetic Images
  - Entities: basic shapes
  - Instantiation parameters: pose (translation, rotation, etc.)



Inverse Computer Graphics

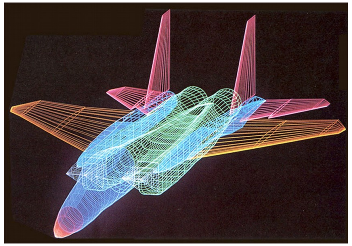
- Image → Entities + Instantiation Parameters

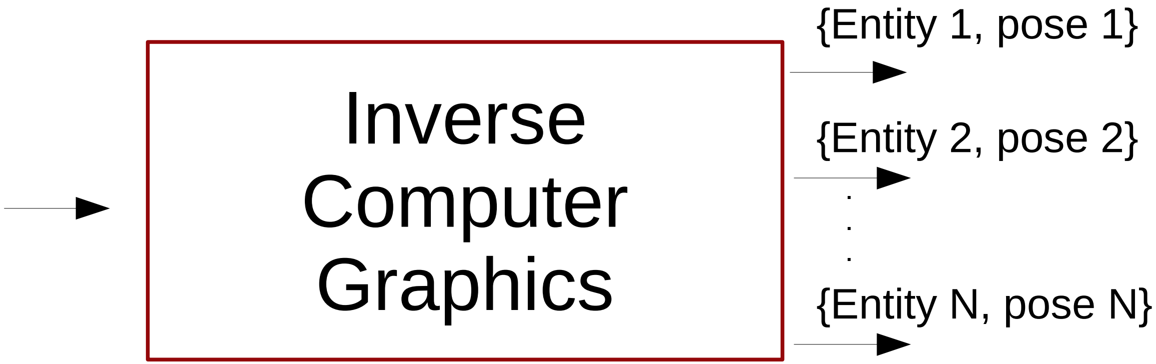




Inverse Computer Graphics

- Image → Entities + Instantiation Parameters





Hinton claim: Human brain performs some inverse graphics.

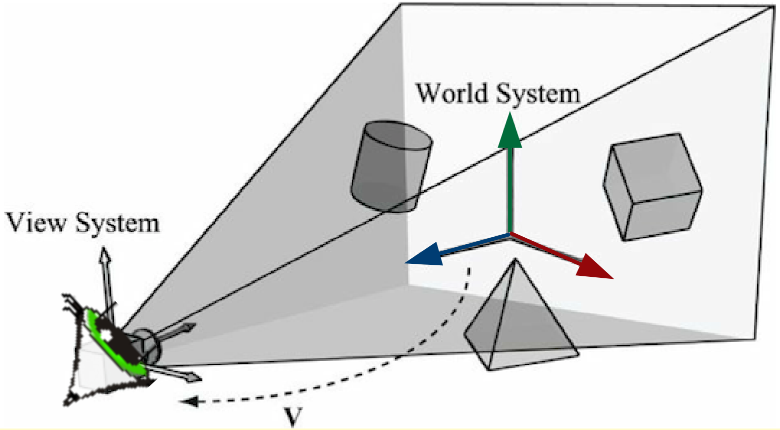
Some Definitions

- Invariant
  - A property that does not change after some transformation
- Equivariant
  - A property that changes predictably under transformation
- Image transformations
  - Shift (translation), scale (size), rotation (orientation), reflection (mirror)

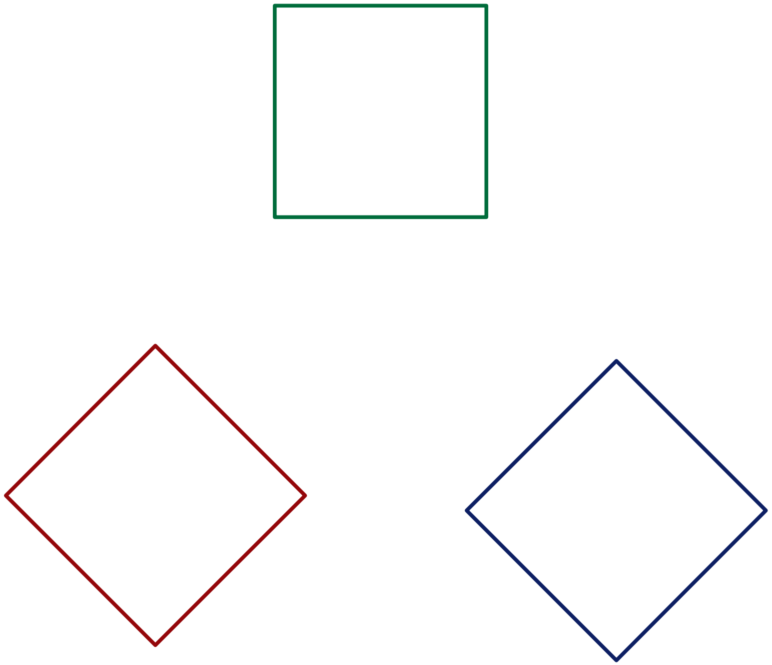
Note that ...

- Invariant
- Equivariant
- Image transformations
- Effect of image transformations on ...
  - Labels → invariant
  - Instantiations parameters → equivariant

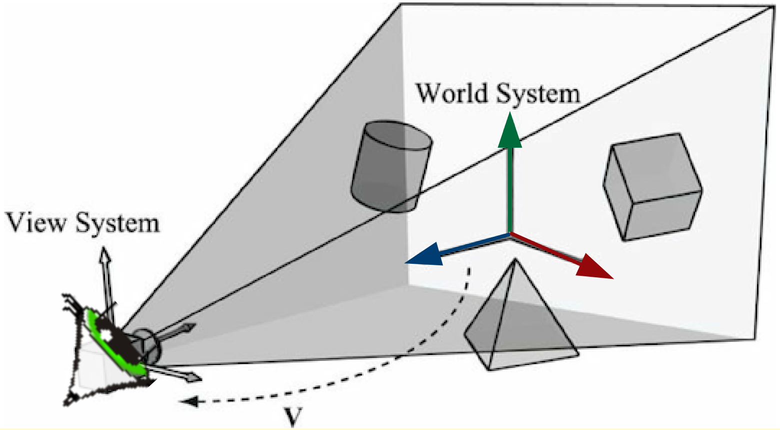
Hinton: Human visual system imposes some coordinate frames in order to represent shapes (after Irvin Rock)



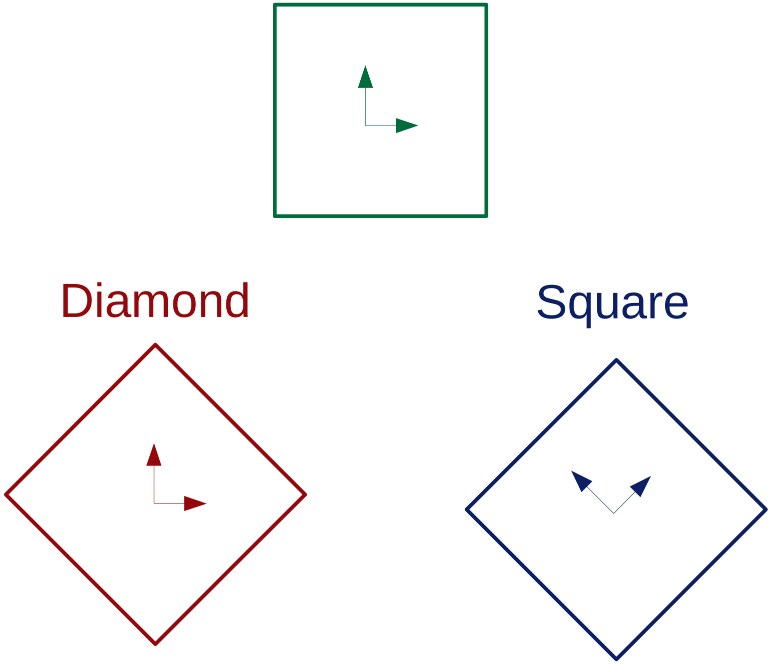
http://ycpcs.github.io/cs470-fall2014/labs/lab07-2.html



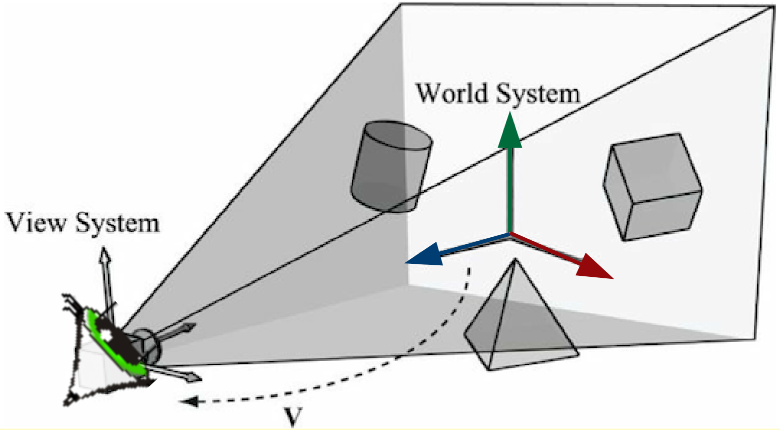
Hinton: Human visual system imposes some coordinate frames in order to represent shapes (after Irvin Rock)



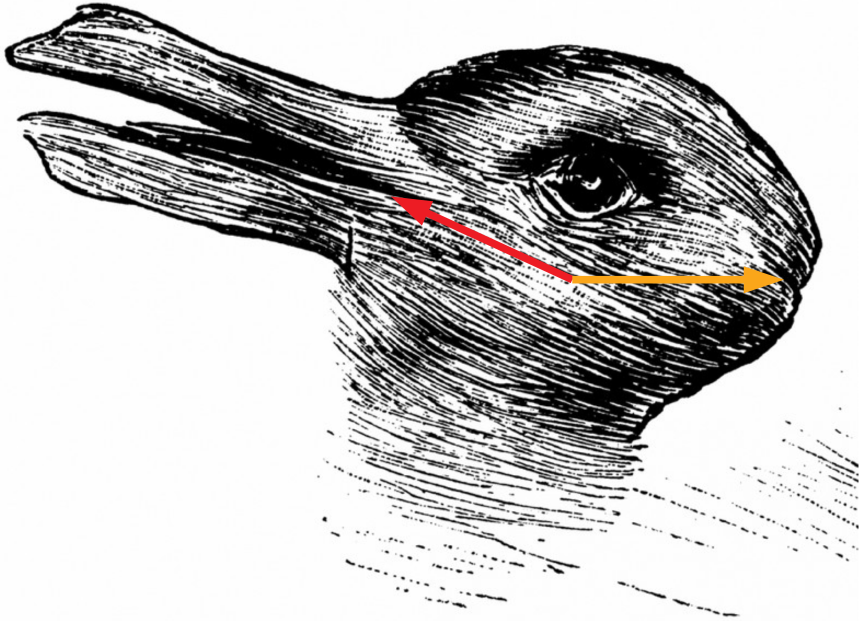
http://ycpcs.github.io/cs470-fall2014/labs/lab07-2.html



Hinton: Human visual system imposes some coordinate frames in order to represent shapes (after Irvin Rock)

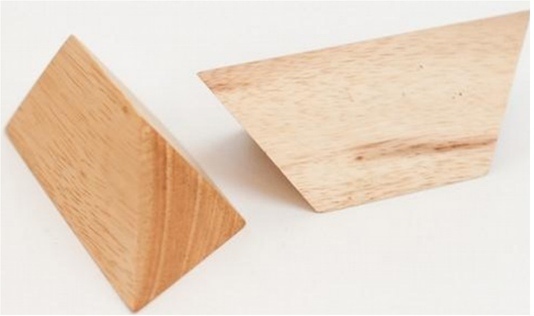


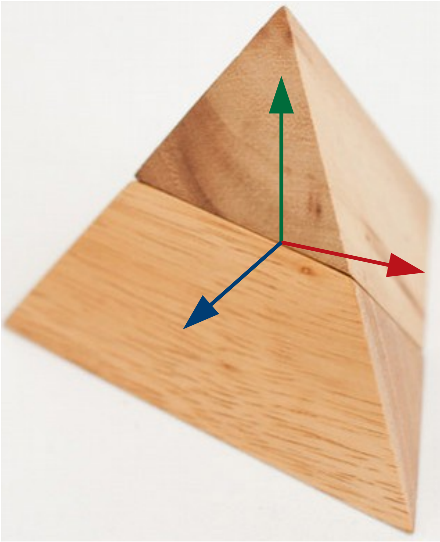
http://ycpcs.github.io/cs470-fall2014/labs/lab07-2.html



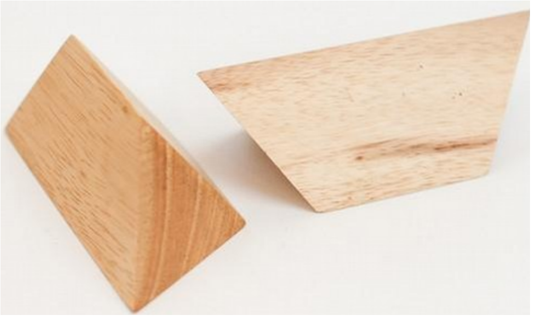
Dock or Rabbit?

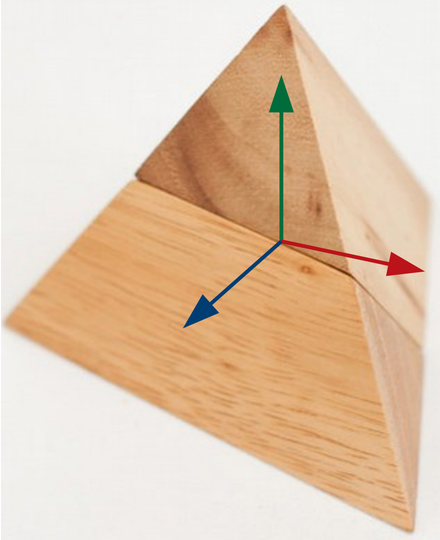
Tetrahedron Jigsaw Puzzle





Tetrahedron Jigsaw Puzzle



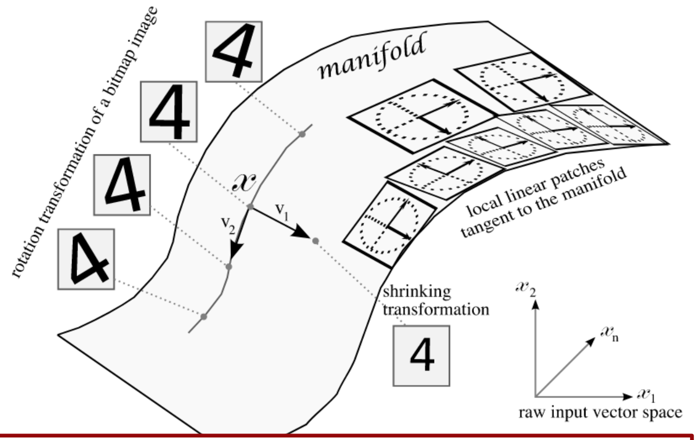


1. Find intrinsic frame of reference
   – Imagine the whole

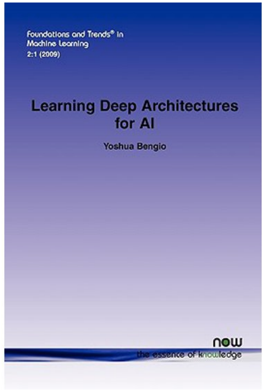
2. Build *part-whole maps* using
   – frame of reference
   – contextual info


END
DETOUR

Invariance and Equivalence



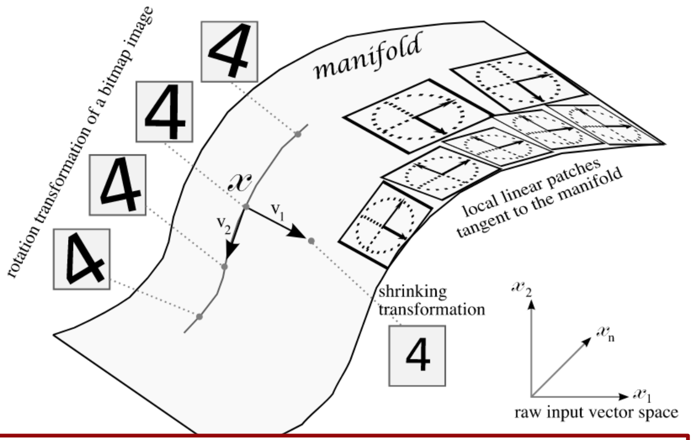
Page 12



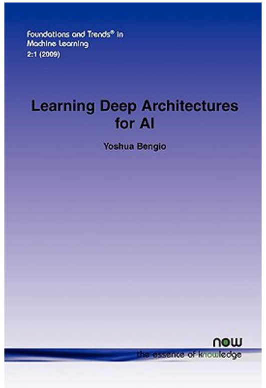
– Label → invariance

– Pose (Instantiation parameters) → equivalence

Invariance and Equivalence



Page 12



– Label → invariance

– Pose (Instantiation parameters) → equivalence

* No built-in disentanglement mechanism in CNNs
  – A lot of data is required for dealing with pose change.

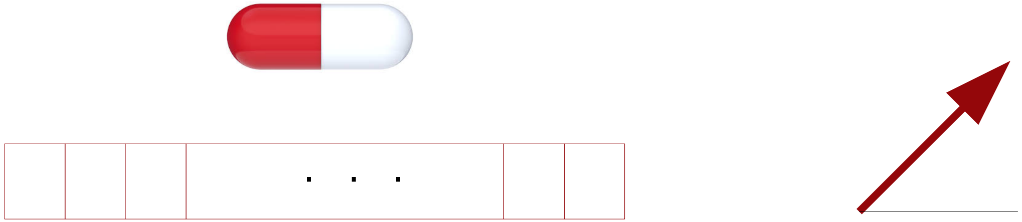
Capsule Networks aim at solving two problems ...

- Disentangling learning mechanisms of invariant (label) and equivariant (pose) properties

- Smarter way for information flow from lower layers to the higher layers in the hierarchy

Capsule and CapsNet

- A set of neurons that collectively produce an activity vector
- Each capsule detects/represents an entity
  - Length: probability of presence/existence
  - Orientation: instantiation parameters, state, properties

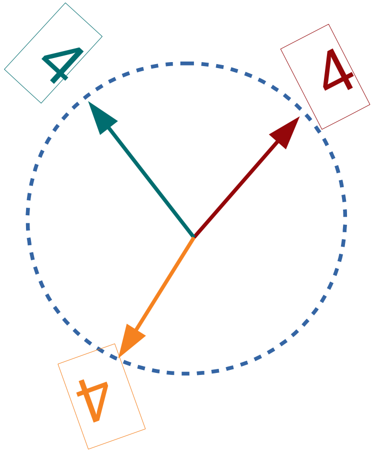


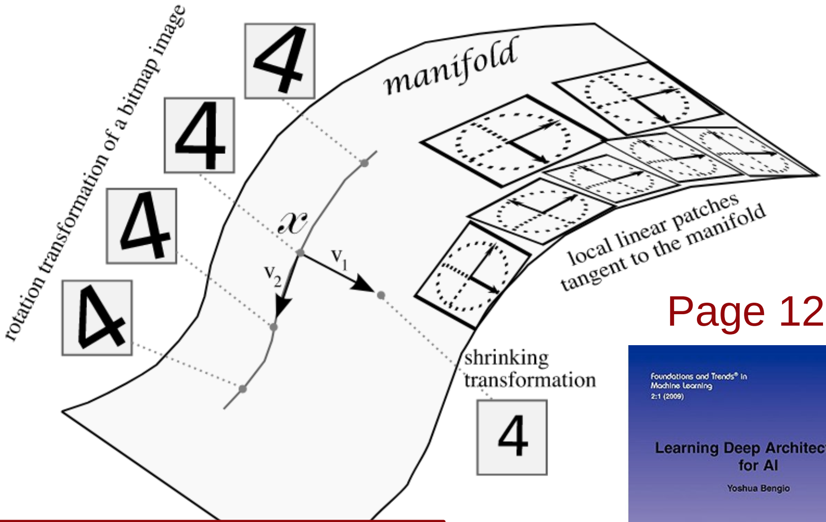
Capsule and CapsNet

- A set of neurons that collectively produce an activity vector
- Each capsule detects/represents an entity
  - Length: probability of presence/existence
  - Orientation: instantiation parameters, state, properties

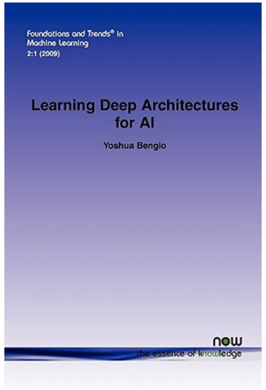
- CapsNets is similar to CNN with two differences
  - Scalar-output nodes are replaced with vector-output capsules
  - Max-pooling is replaced with *routing-by-agreement*

CapsNet Approach to Invariance and Equivalence Properties





Page 12



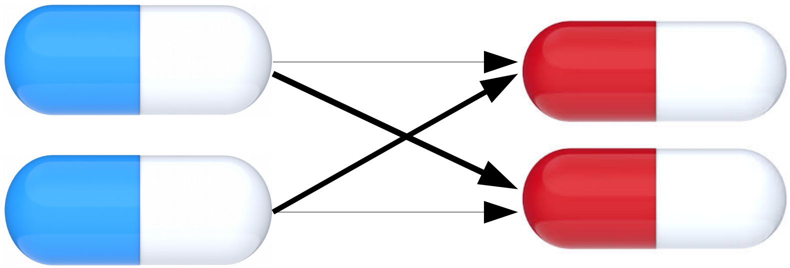
– Entity's presence probability: invariance
– Entity's pose (Instantiation parameters) equivalence

– Built-in separation mechanism

Dynamic Routing
via
Routing-by-agreement

Routing-by-Agreement – Steps

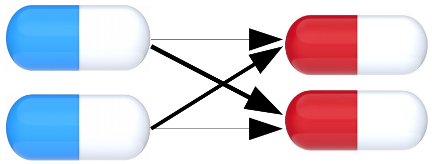
0) Outputs of capsules in lower layer ($\mathbf{u}_i$) are available



Routing-by-Agreement – Steps

0) Outputs of capsules in lower layer ($\mathbf{u}_i$) r available

1) For capsule j in higher layer, make a prediction ($\hat{\mathbf{u}}_{j|i}$)

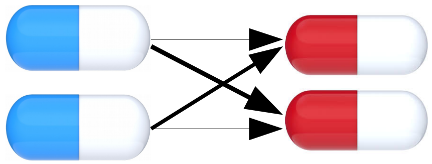


Routing-by-Agreement – Steps

0) Outputs of capsules in lower layer ($\mathbf{u}_i$) r available

1) For capsule *j* in higher layer, make a prediction ($\hat{\mathbf{u}}_{j|i}$)

2) Compare the prediction with actual output ($\mathbf{v}_j$)



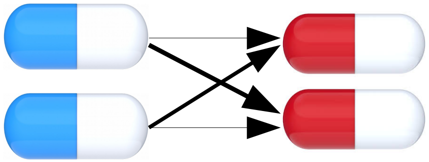
Routing-by-Agreement – Steps

0) Outputs of capsules in lower layer ($\mathbf{u}_i$) r available

1) For capsule *j* in higher layer, make a prediction ($\hat{\mathbf{u}}_{j|i}$)

2) Compare the prediction with actual output ($\mathbf{v}_j$)

3) Based on $\hat{\mathbf{u}}_{j|i}$ & $\mathbf{v}_j$ similarity, adjust the connection strength (routing)



Routing-by-Agreement – Steps

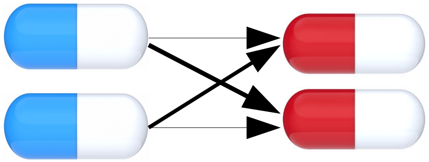
0) Outputs of capsules in lower layer ($\mathbf{u}_i$) r available

1) For capsule *j* in higher layer, make a prediction ($\hat{\mathbf{u}}_{j|i}$)

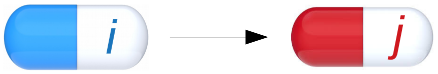
2) Compare the prediction with actual output ($\mathbf{v}_j$)

3) Based on $\hat{\mathbf{u}}_{j|i}$ & $\mathbf{v}_j$ similarity, adjust the connection strength (routing)

4) Go to (2), if not converged



Routing-by-Agreement – Equations



$\hat{\mathbf{u}}_{j|i}$: Prediction of *i* about *j* using W_{ij}

c_{ij}: coupling coef. Between *i* and *j*

$\mathbf{s}_j$: pre-activation of *j*

$\mathbf{v}_j$: activation of *j*

Squashing Non-linearity

b_{ij}: logit (similarity)

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

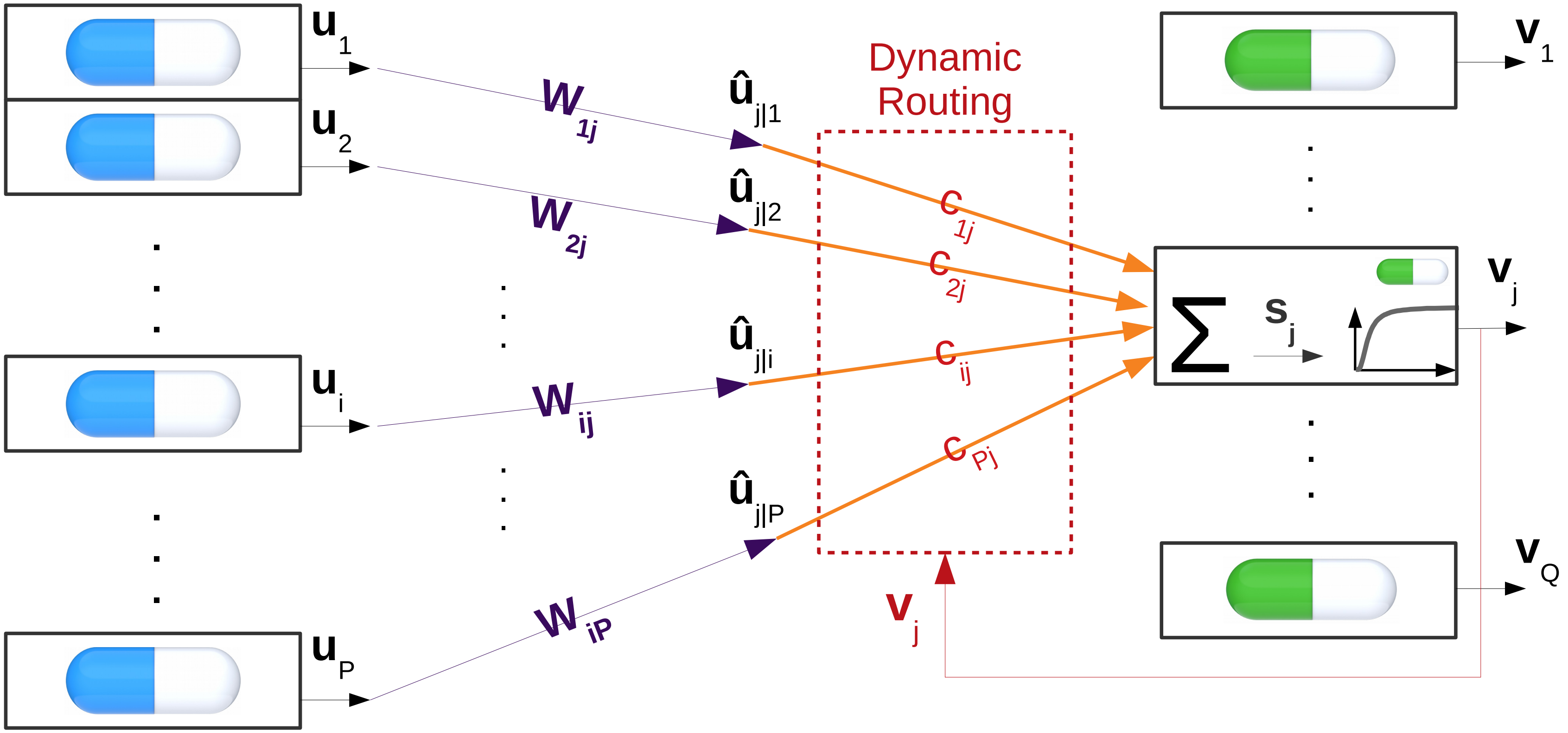
iteration

$$\mathbf{s}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$$

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1 + ||\mathbf{s}_j||^2}\frac{\mathbf{s}_j}{||\mathbf{s}_j||}$$

$$b_{ij} += \mathbf{v_j}.\hat{\mathbf{u}}_{j|i} \longrightarrow c_{ij} = \frac{\exp(b_{ij})}{\sum_{j'}\exp(b_{ij'})}$$

Routing-by-Agreement – WorkFlow



Routing-by-Agreement – Algorithm

Procedure 1 Routing algorithm.

1: **procedure** ROUTING($\hat{\mathbf{u}}_{j|i}$, r, l)
2: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
3: **for** r iterations **do**
4: for all capsule i in layer l: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$ ▷ softmax computes Eq. 3
5: for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
6: for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$ ▷ squash computes Eq. 1
7: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
return $\mathbf{v}_j$

– c_{ij} is learned by *dynamic routing* in forward path

– W_{ij} is learned by *backprop* in backward path

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_{j'} \exp(b_{ij'})}$$

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}$$

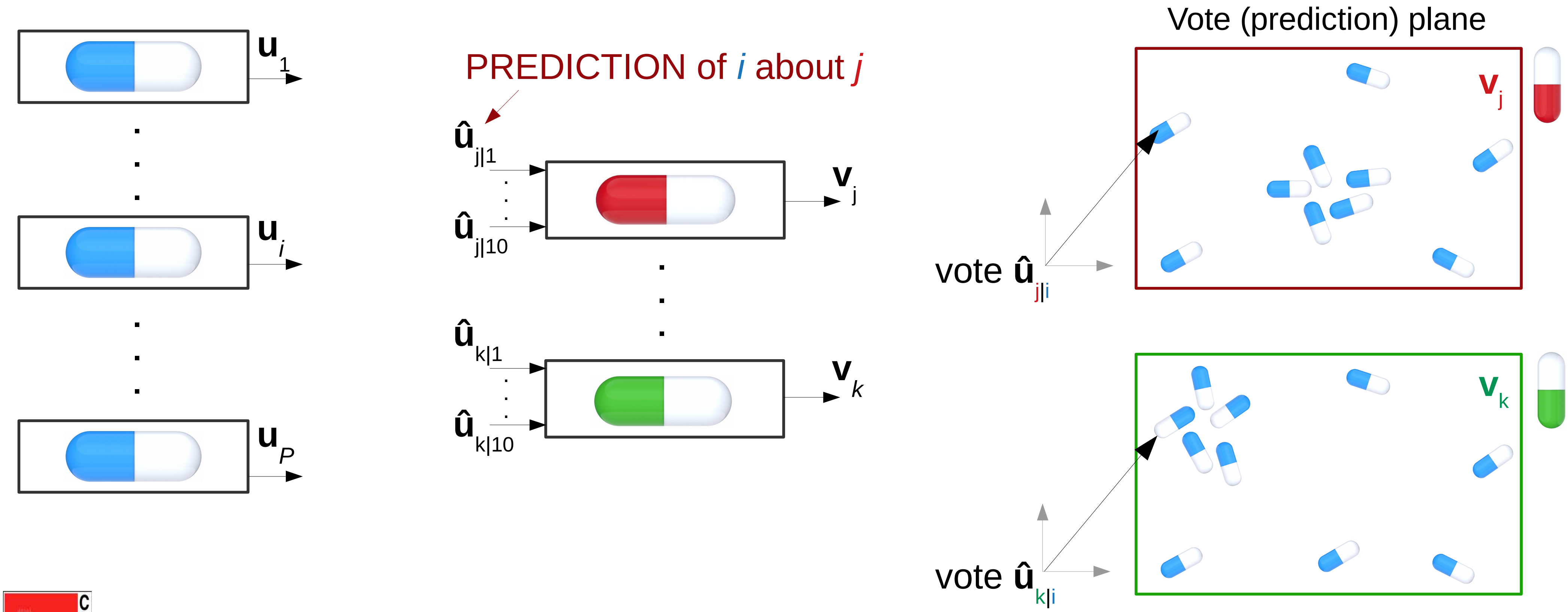
Conventional NN vs CapsNet

	Neurons	**Capsules**
Input/Output	Vector/Scalar	Vector/Vector
Training	Backpropagation	Dynamic Routing & Backpropagation
Pre-activation	$z_j = \sum_i \boxed{w_{ij}} \, x_i + \boxed{b_j}$	$\mathbf{s}_j = \sum_i \boxed{c_{ij}} \, \boxed{\mathbf{W}_{ij}} \, \mathbf{u}_i$
Non-linearity	scalar2scaler	vector2vector
	ReLU, Tanh, etc.	$\mathbf{v}_j = \frac{\lVert\mathbf{s}_j\rVert^2}{1 + \lVert\mathbf{s}_j\rVert^2} \frac{\mathbf{s}_j}{\lVert\mathbf{s}_j\rVert}$

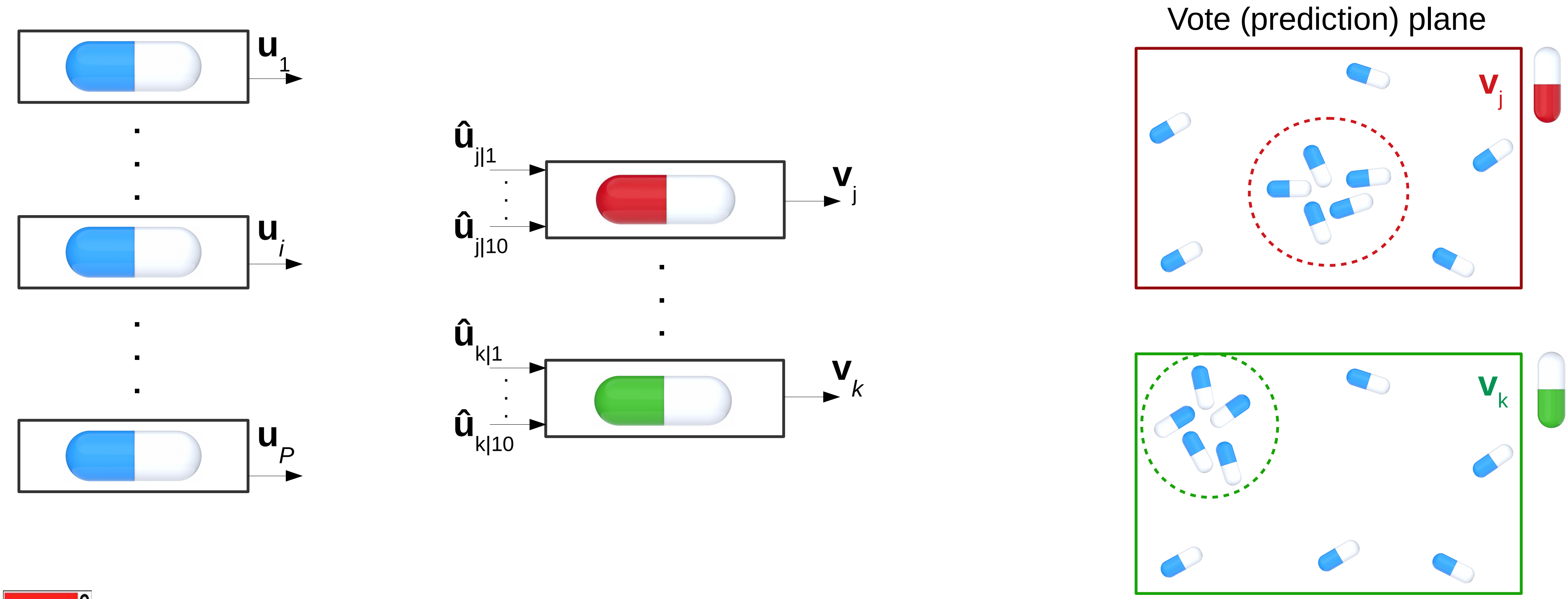
Routing-by-Agreement

Intuitions

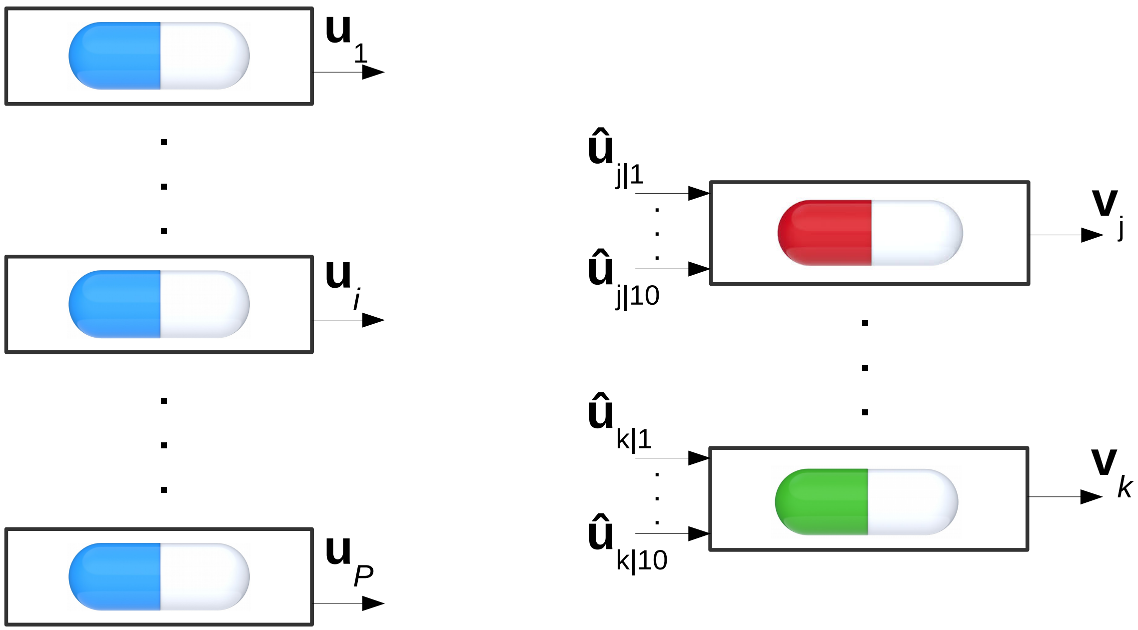
Routing-by-Agreement – Intuition (1)

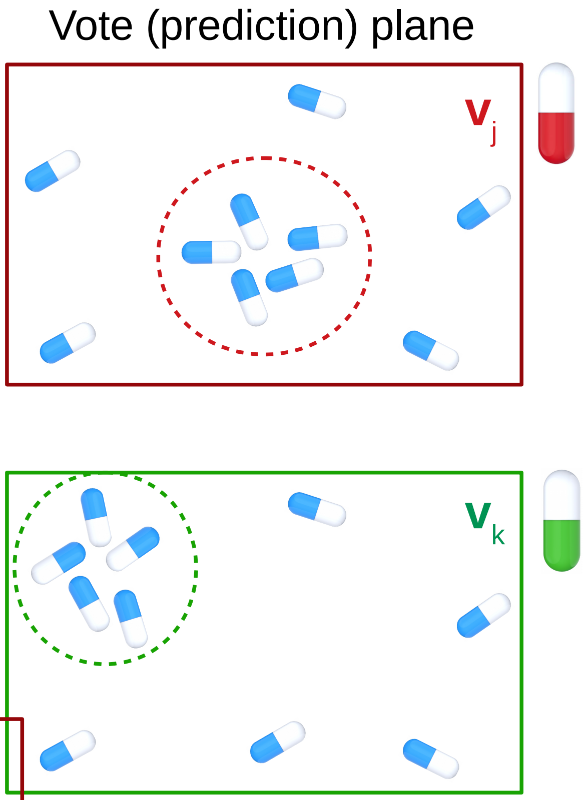


Routing-by-Agreement – Intuition (1)



Routing-by-Agreement – Intuition (1)

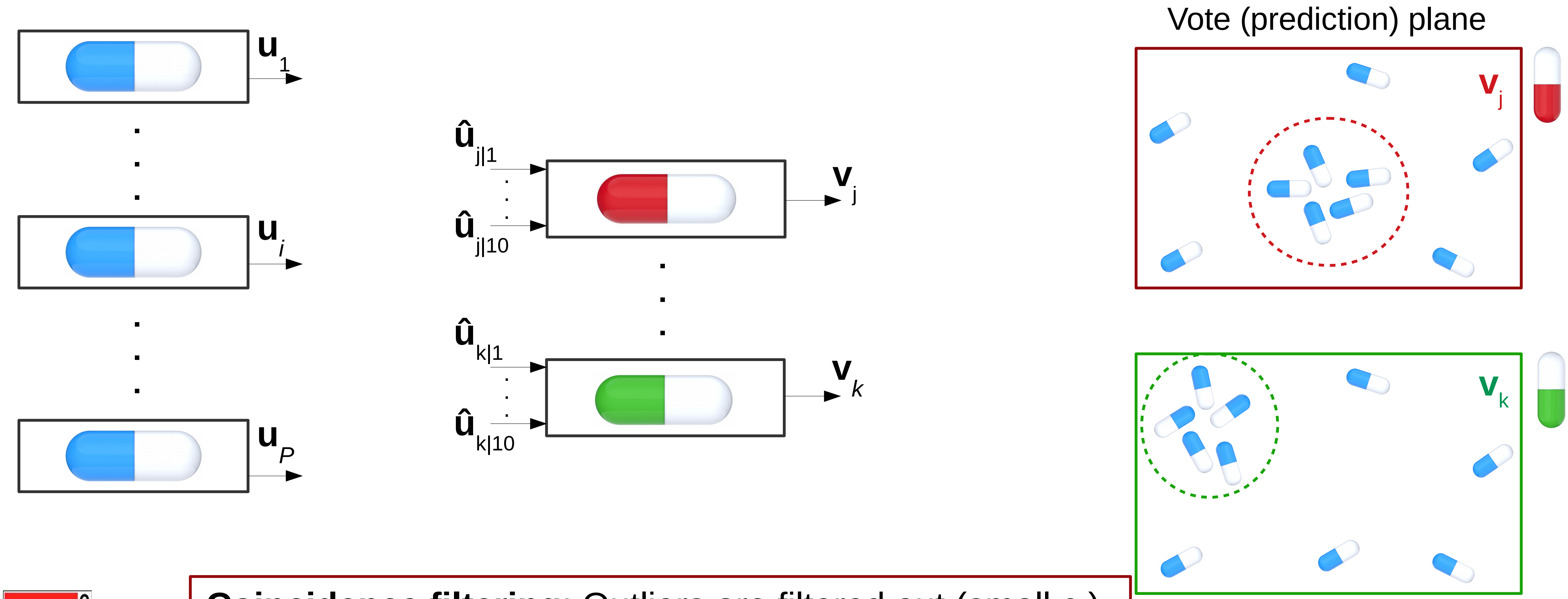




Capsules in dash circle should be routed to

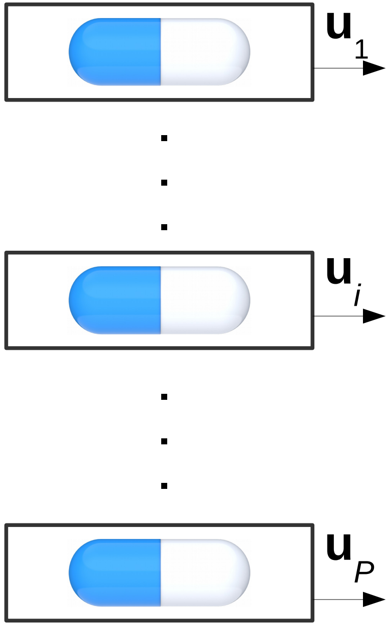
→ larger coupling coefficients, c_{ij}, (**part-whole map**)

Routing-by-Agreement – Intuition (2)



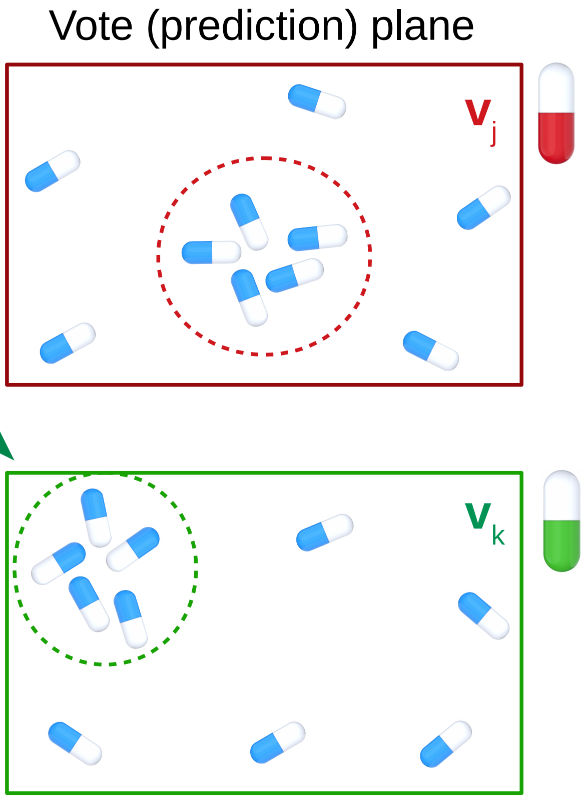
Coincidence filtering: Outliers are filtered out (small c_{ij}).

Routing-by-Agreement – Note

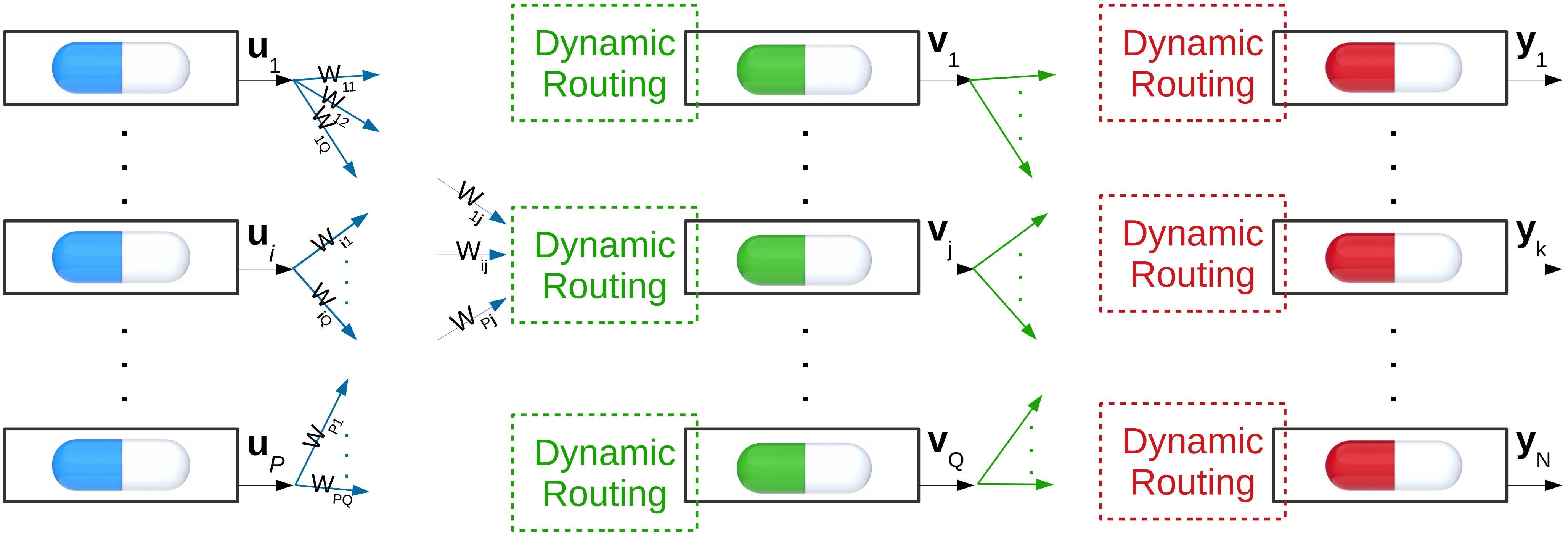


$$\hat{\mathbf{u}}_{j|i} = W_{ij}\mathbf{u_i}$$

Votes distribution in vote plane, i.e. $\hat{\mathbf{u}}_{j|i}$ and $\hat{\mathbf{u}}_{k|i}$, are different because although $\mathbf{u}_i$ is the same, W_{ij} and W_{ik} are different.

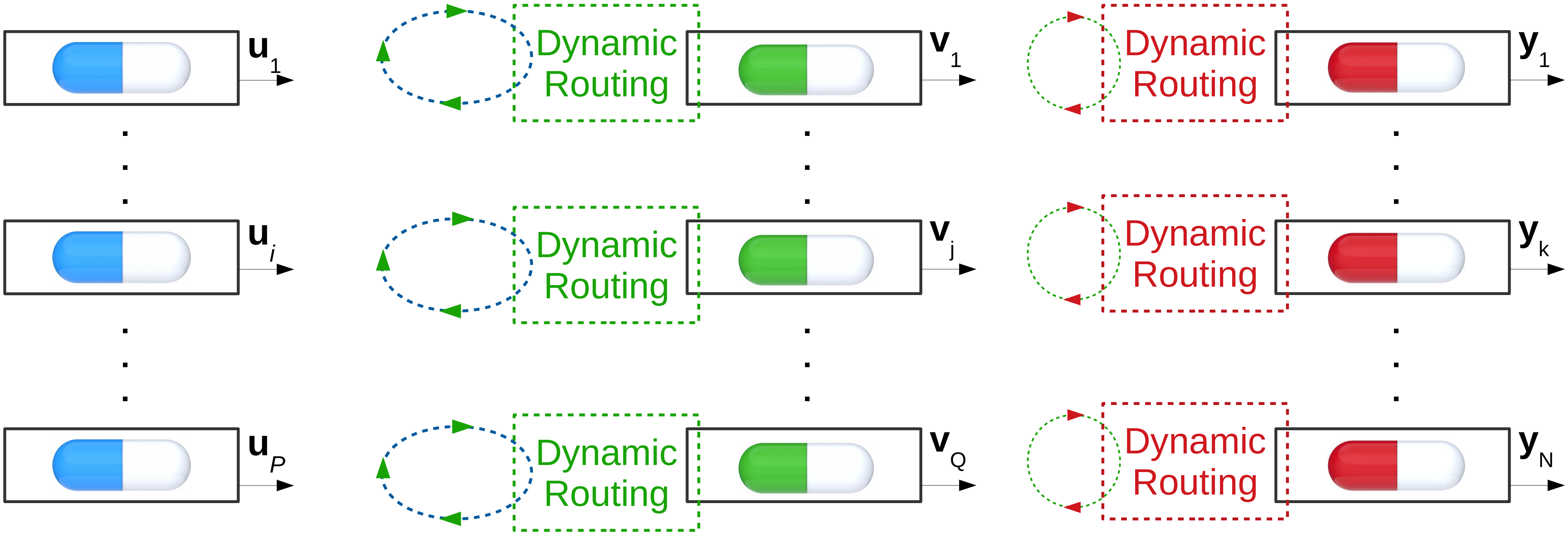


Routing-by-Agreement – Intuition (3)



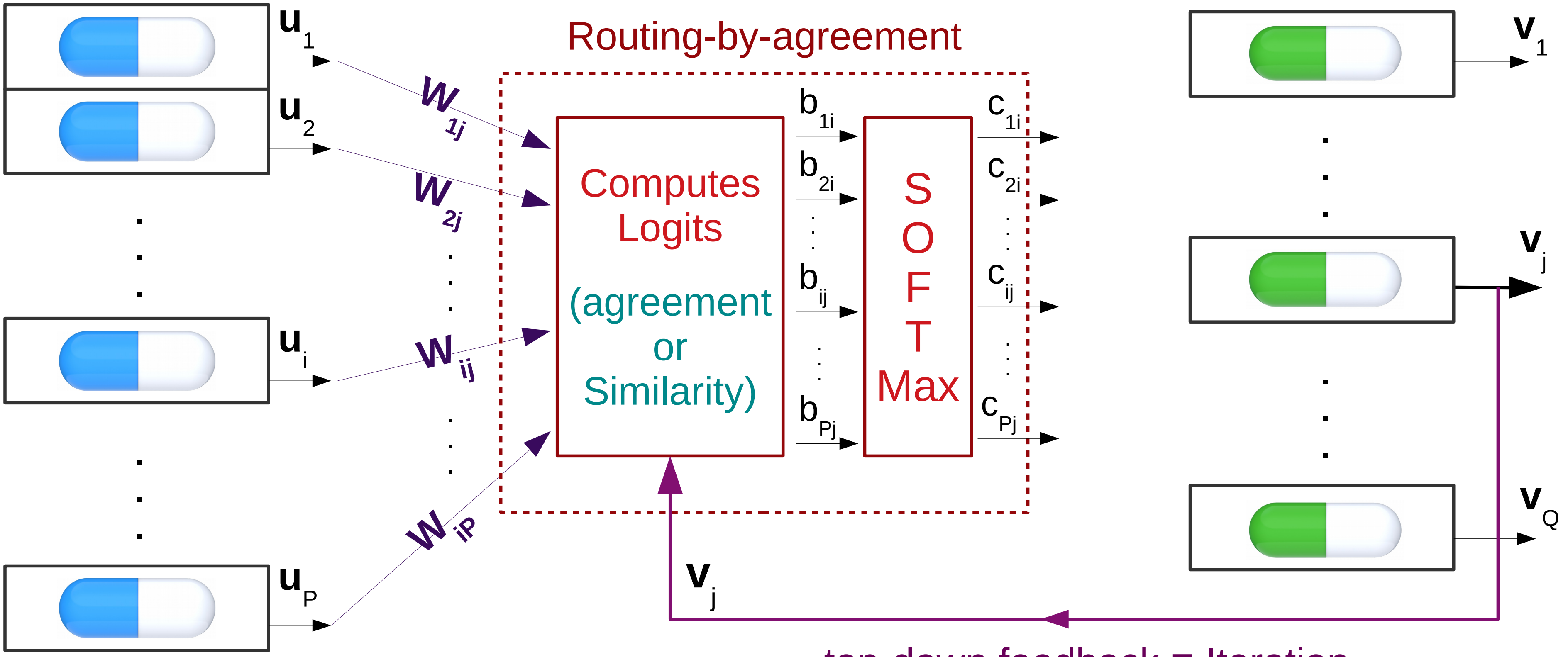
Each higher level capsule has a dynamic routing block.
– **Primary** capsule layer (lowest level) has not.

Routing-by-Agreement – Iterations



Routing-by-agreement to done greedily across layers ...
– When iteration**s** between blue-green finished, move to green-red.

Routing-by-Agreement – Intuition (3)



Capsule Network in NIPS 2017

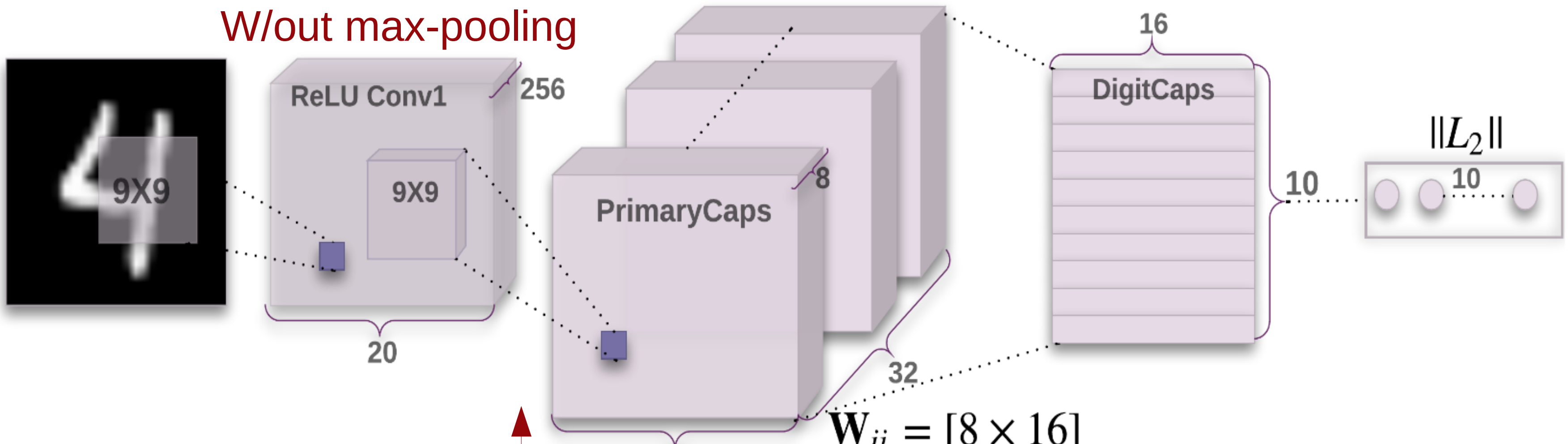
Dynamic Routing Between Capsules

Sara Sabour **Nicholas Frosst**

Geoffrey E. Hinton
Google Brain
Toronto
{sasabour, frosst, geoffhinton}@google.com

Capsule Network in NIPS 2017

W/out max-pooling



$$\mathbf{W}_{ij} = [8 \times 16]$$

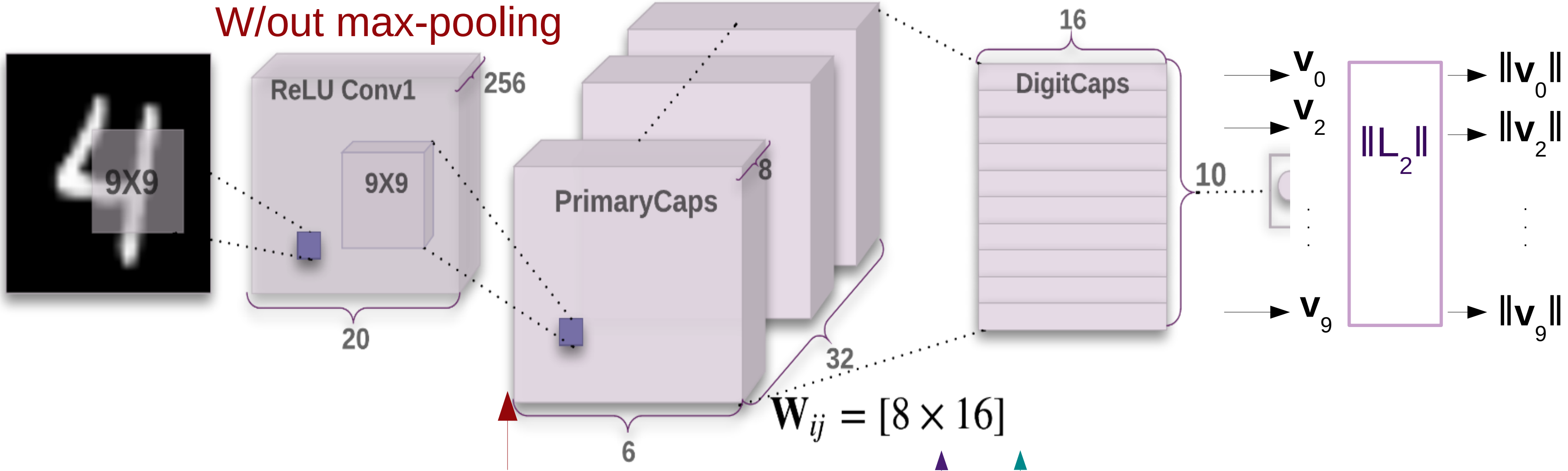
reshape(-1, capsule_size)

DigitCaps size

Primary capsule size

Capsule Network in NIPS 2017

W/out max-pooling

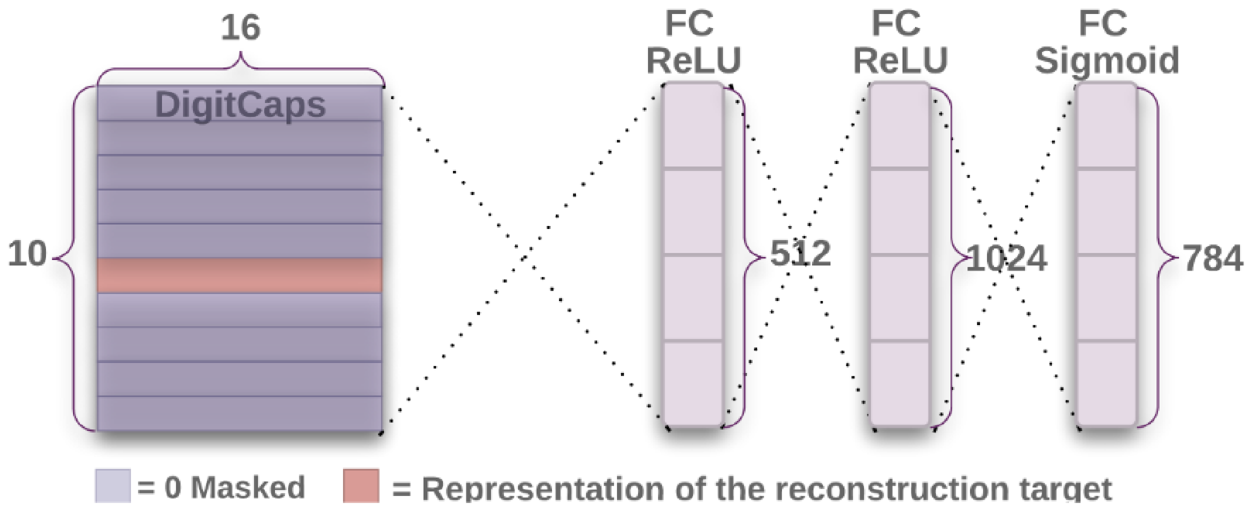


reshape(-1, capsule_size)

DigitCaps size

Primary capsule size

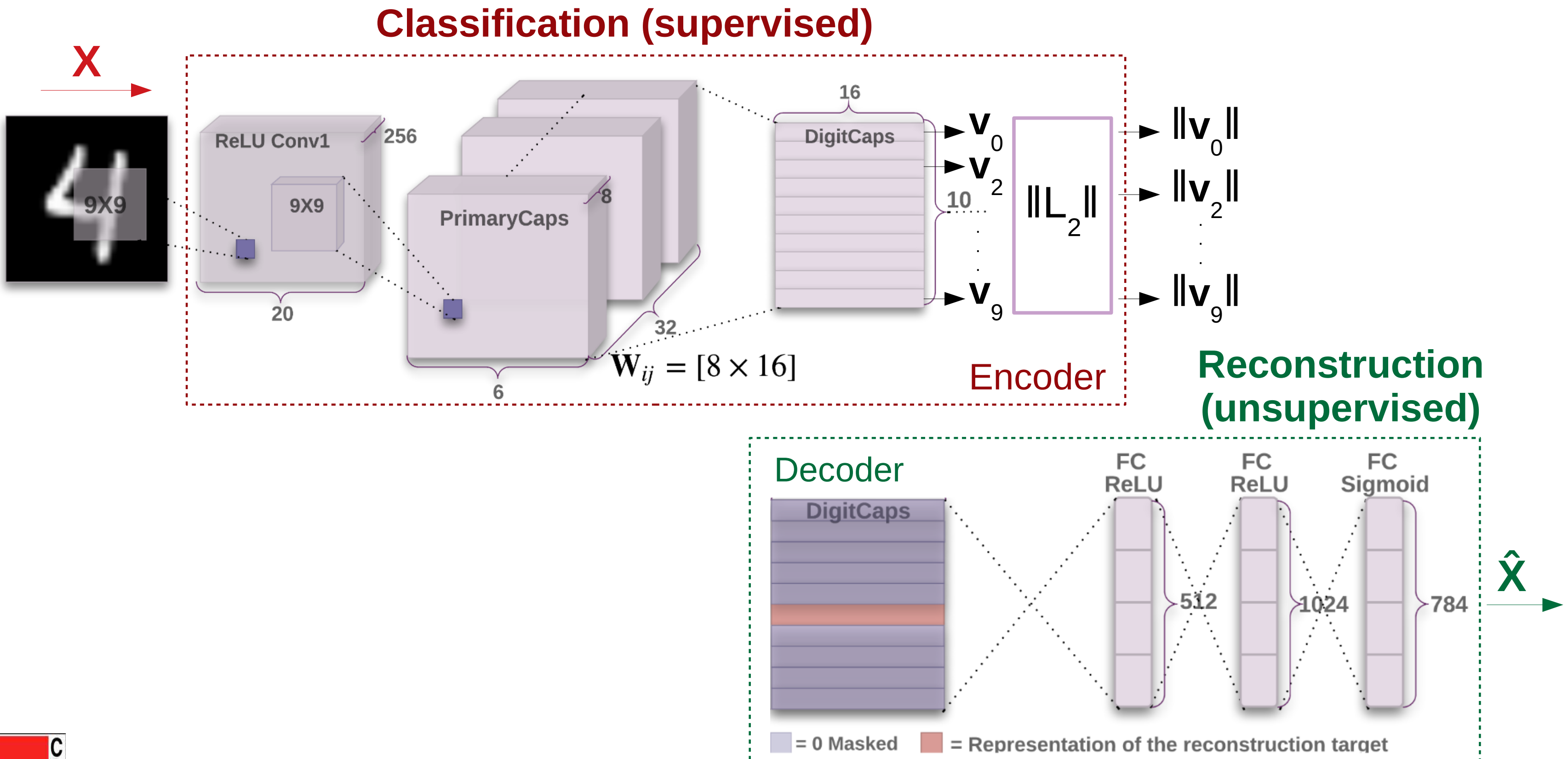
Reconstruction Network (Decoder)



– Target capsule is kept; rest is **0** masked

– Reconstruction from a vector → ~ auto-encoder

Unsupervised Reconstruction Loss



Loss Function

- Loss function = supervised + α unsupervised

Loss Function

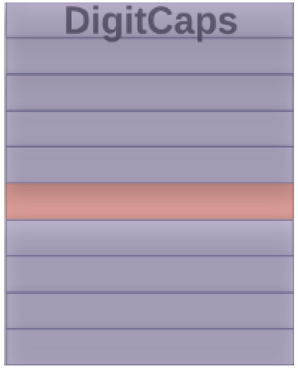
- Loss function = supervised + α unsupervised
- Supervised → classification
  - margin loss
- Unsupervised (decoder) → Reconstruction
  - MSE
  - Down-scaled by α = 5e-4
    - Adjusting scales + making the supervised part dominant

Supervised Margin Loss

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda \; (1 - T_k) \max(0, \|v_k\| - m^-)^2$$

$$L = \sum_k L_k$$

T_k = **1** if (digit of class k is present) else **0**



Supervised Margin Loss

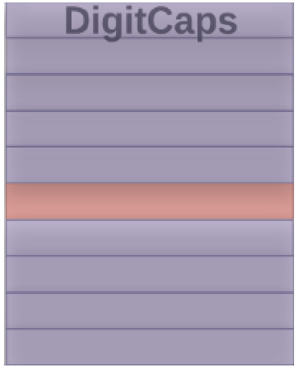
$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda\ (1 - T_k) \max(0, \|v_k\| - m^-)^2$$

$$L = \sum_k L_k$$

T_k = **1** if (digit of class k is present) else **0**

$Z = T_k X + (1-T_k) Y$

- T_k \in (0,1) → weighted mean
- T_k \in {0,1} → Z = X if T_k==1 else Y

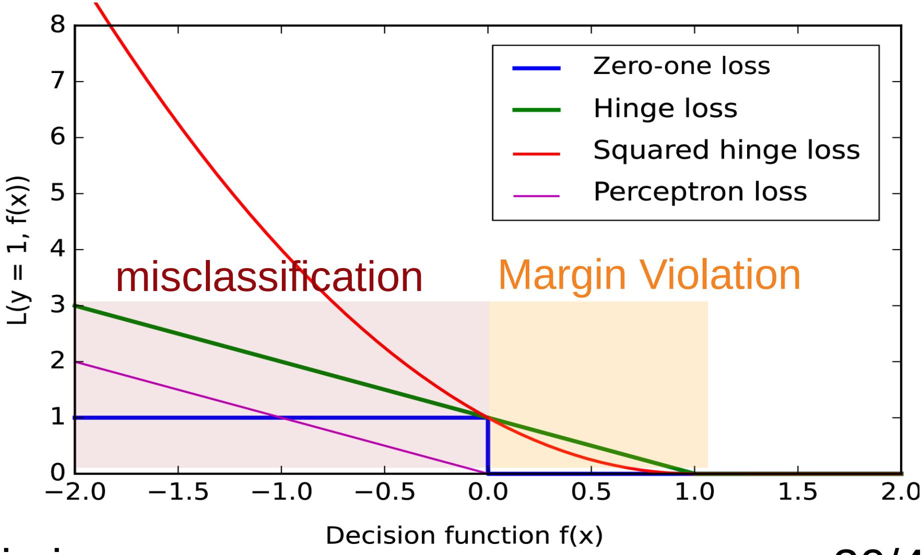


Supervised Margin Loss

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda\ (1 - T_k) \max(0, \|v_k\| - m^-)^2$$

$$L = \sum_k L_k$$

* Hinge (max-margin) loss:
  - max(0, m^+-x)
    ==>> min loss: x > m^+
  - max(0, x-m^-)
    ==>> min loss: x < m^-
  - m^+=0.9, m^-=0.1



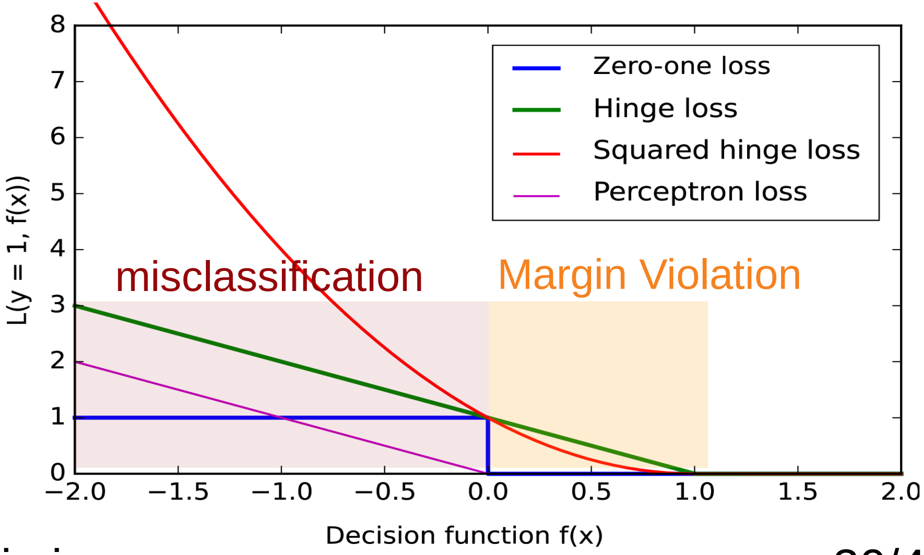
Supervised Margin Loss

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda\ (1 - T_k) \max(0, \|v_k\| - m^-)^2$$

$$L = \sum_k L_k$$

For minimum loss

- If $T_k == 1$: $\|v_k\| > m^+$
- If $T_k == 0$: $\|v_k\| < m^-$



Supervised Margin Loss

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda\ (1 - T_k) \max(0, \|v_k\| - m^-)^2$$

$$L = \sum_k L_k$$

* $\lambda = 0.5$
  – down-weighs $T_k = 0$ case
  – Purpose: Numerical stability

Experimental Results

CapsNet Classification Error

Table 1 STOA: 0.21% Trials for STD: 3

Method	Routing	Reconstruction	MNIST (%)	MultiMNIST (%)
Baseline	-	-	0.39	8.1
CapsNet	1	no	$0.34_{\pm 0.032}$	-
CapsNet	1	yes	$0.29_{\pm 0.011}$	7.5
CapsNet	3	no	$0.35_{\pm 0.036}$	-
CapsNet	3	yes	$\mathbf{0.25}_{\pm 0.005}$	**5.2**

CapsNet Classification Error

Table 1 STOA: 0.21% Trials for STD: 3

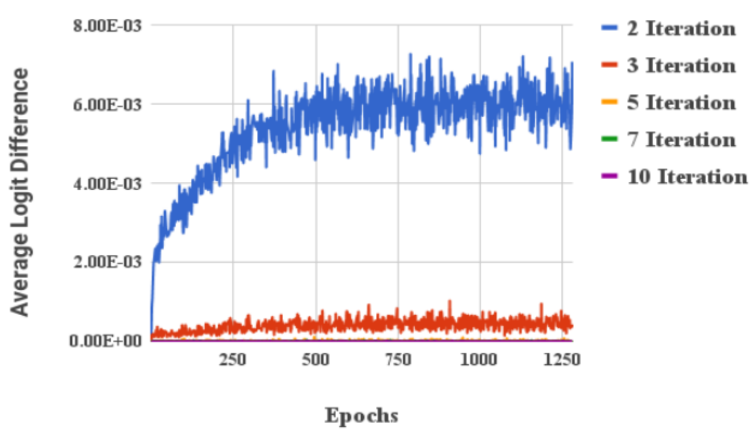
Method	Routing	Reconstruction	MNIST (%)	MultiMNIST (%)
Baseline	-	-	0.39	8.1
CapsNet	1	no	$0.34_{\pm 0.032}$	-
CapsNet	1	yes	$0.29_{\pm 0.011}$	7.5
CapsNet	3	no	$0.35_{\pm 0.036}$	-
CapsNet	3	yes	$\mathbf{0.25}_{\pm 0.005}$	**5.2**

* Routing iterations: 3 to 5 is enough $\leftarrow$ computational cost + overfitting

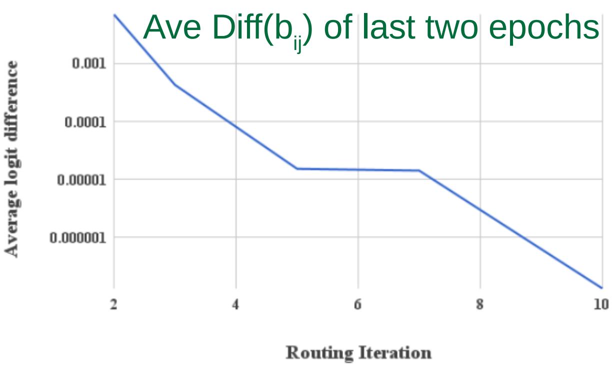
* Adding reconstruction term to loss is useful.

CapsNet Logit Change (MNIST)

After 500 epochs, average change in logit (b_{ij}) is stabilised.



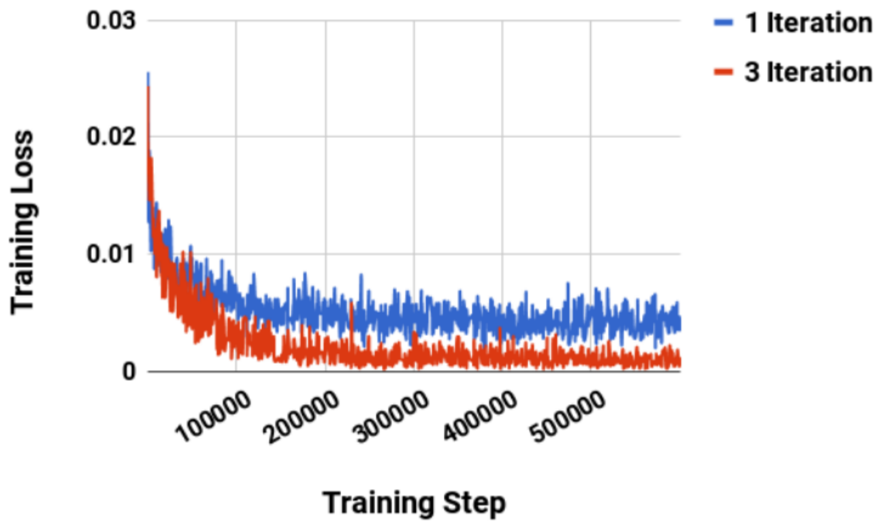
3 iterations of routing is enough.



CapsNet Training Loss (CIFAR 10)

3 iterations of routing optimise the loss faster and converges to a lower loss at the end.

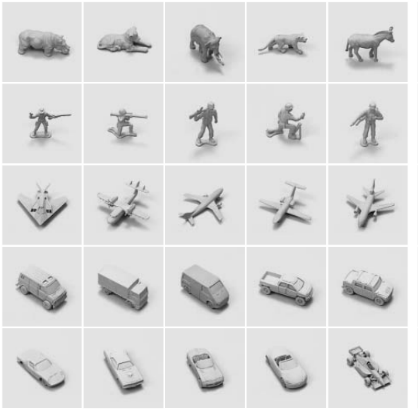
More routing iterations increases the network capacity → overfitting



CapsNet Error on CIFAR 10

- CapsNet: 10.6%
  - About what standard CNNs achieved when first tried
    - *Zeiler and Fergus 2013* → 19.4, 15.1%
  - State-of-the-art: 3.47% (*Graham 2015*)

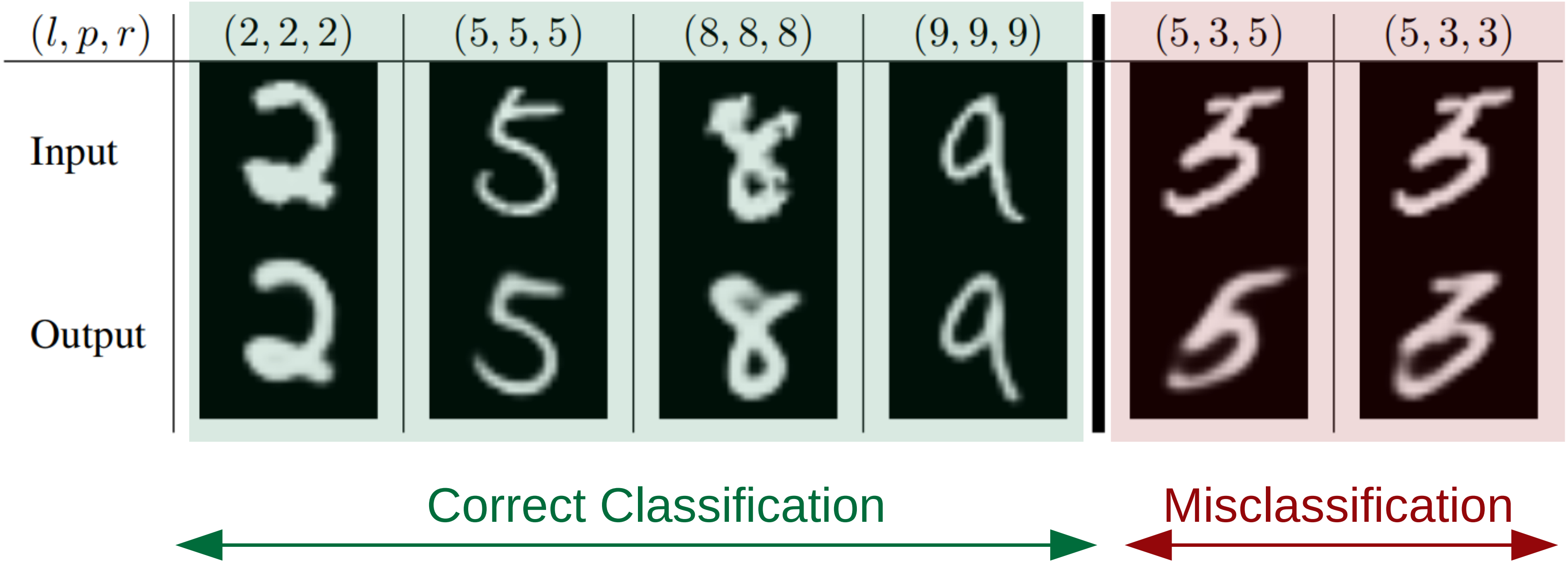
CapsNet Error on Small NORB



- CapsNet error: 2.7%
  - Best task for CapsNet (Appendix B)
  - On-par with state-of-the-art (2.56%)
    - *Ciresen et al., 2011*
  - New CapsNet with EM routing, ICLR 2018 → 1.4%

CapsNet Reconstruction

(l,r,p) = (target **l**abel, **p**rediction, **r**econstruction target)



Reconstruction from the target capsule (~ auto-encoder)

CapsNet Reconstruction

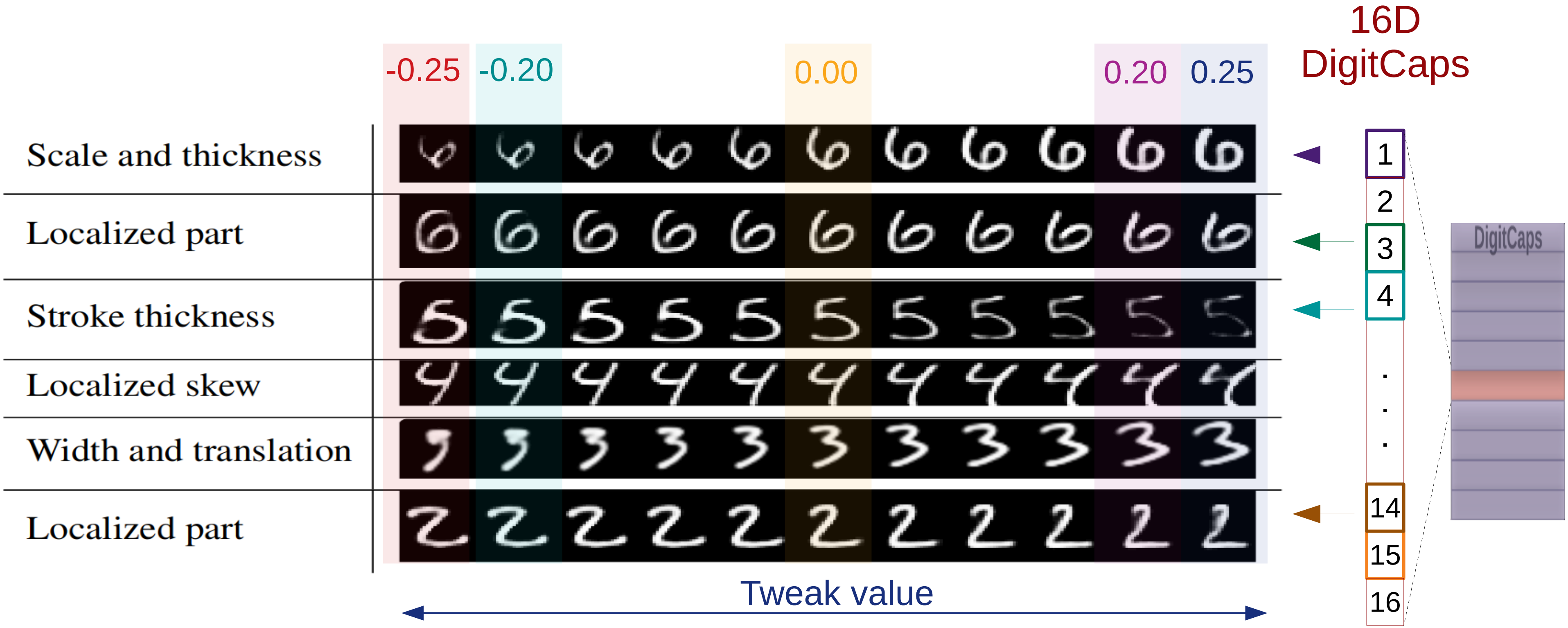
(l,r,p) = (target **l**abel, **p**rediction, **r**econstruction target)

Correct Classification

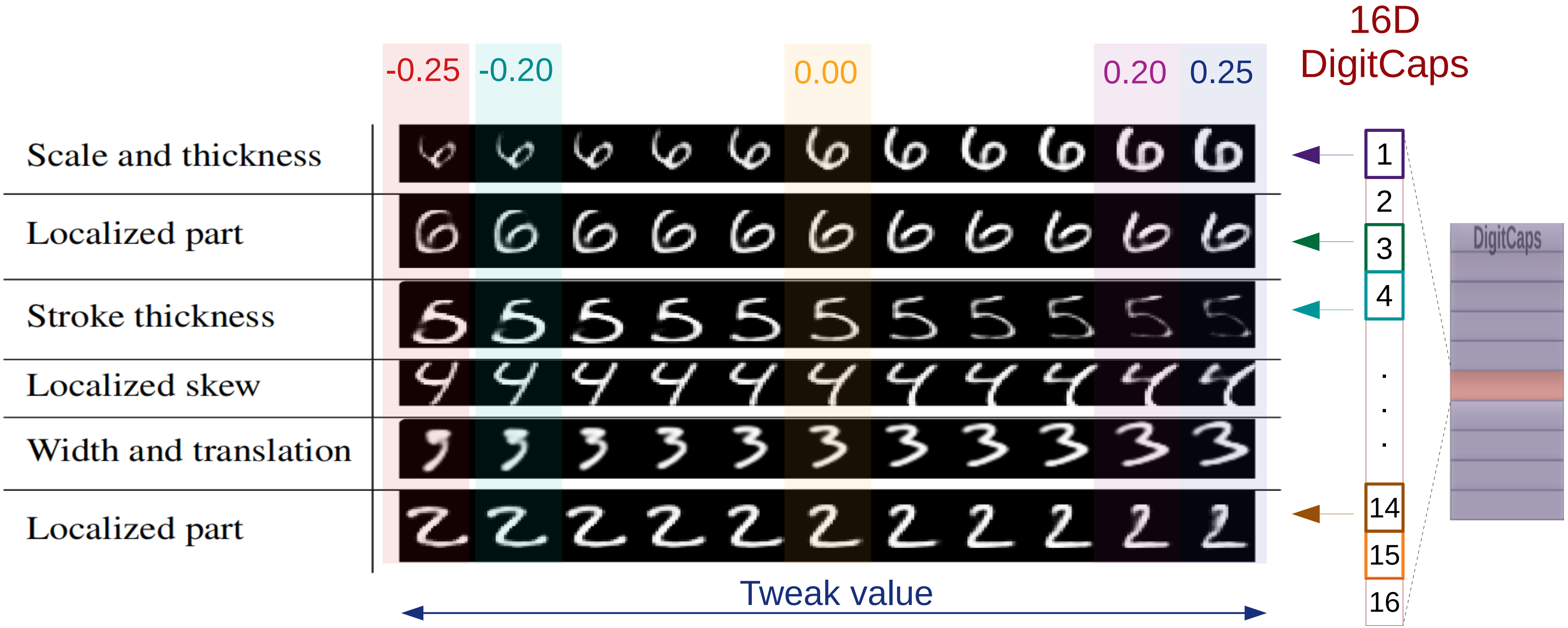
Misclassification

The model preserves many of the details while smoothing the noise.

Effect of Dimension Perturbation on Recon.

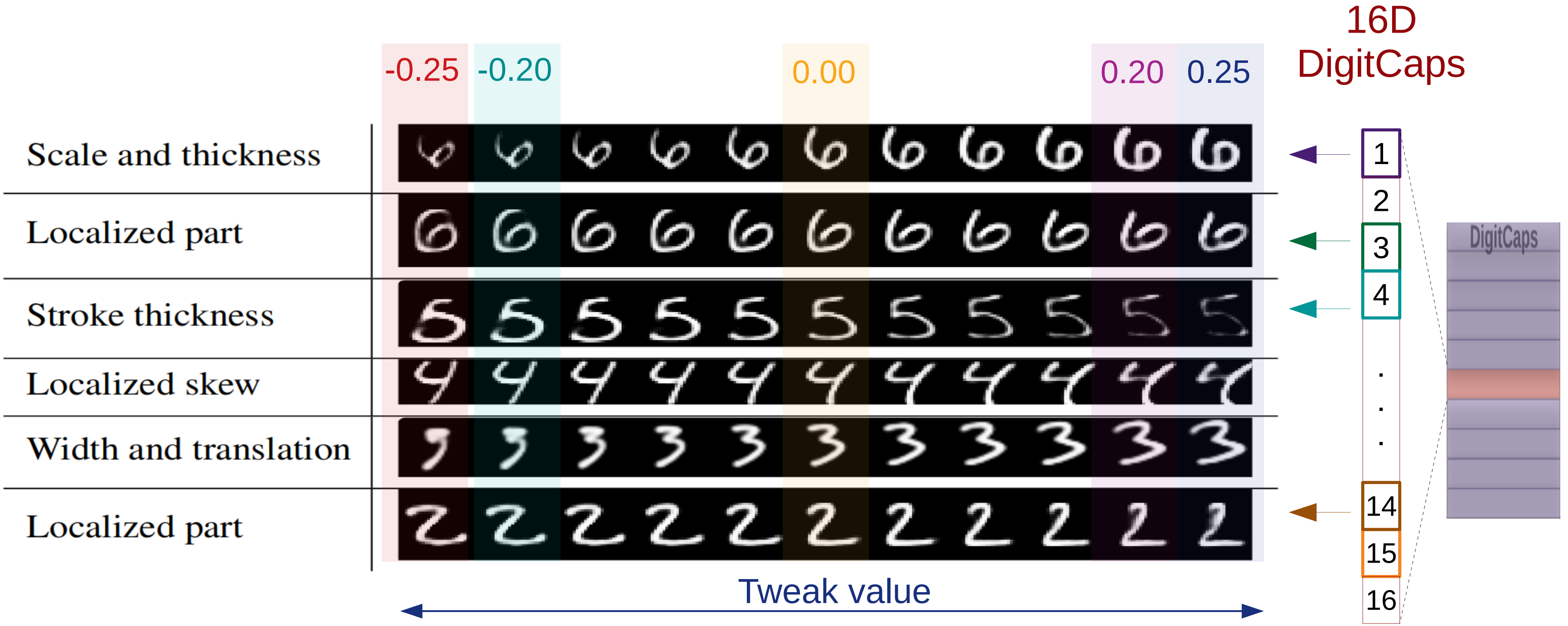


Effect of Dimension Perturbation on Recon.



Each dimension of capsule learns to span the space of variation of an instantiation parameter, e.g. scale, translation, thickness

Effect of Dimension Perturbation on Recon.

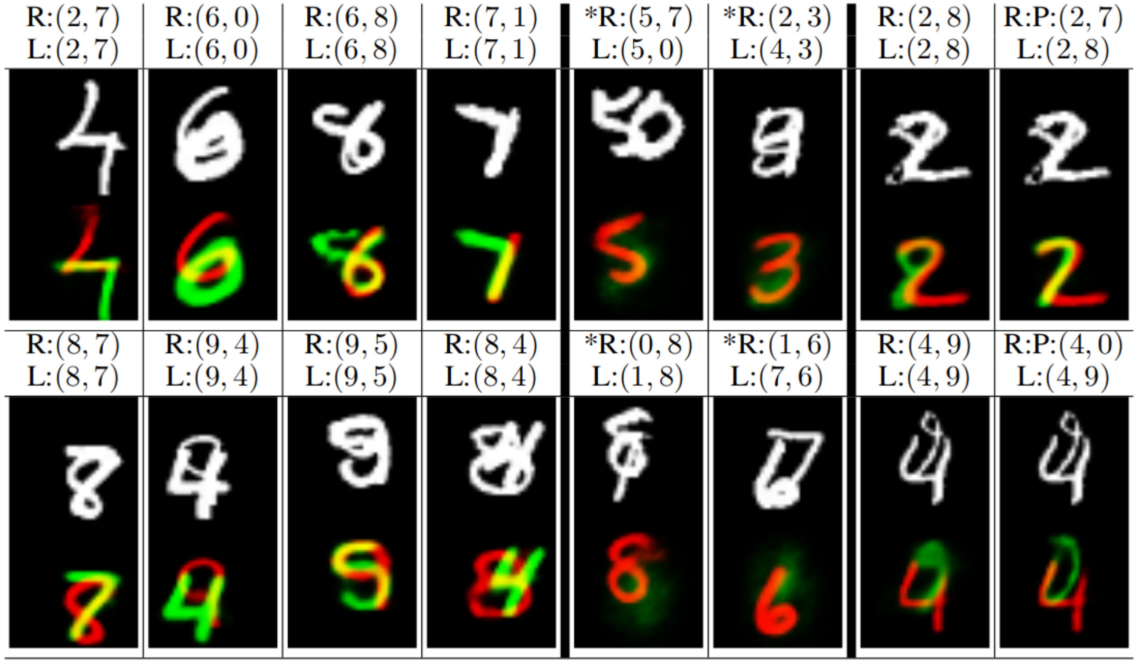


Higher interpretability & controllability

MultiMNIST Reconstruction

- MultiMNIST
  - Each X has two labels (l_1,l_2)
- L: (l_1,l_2)
  - Target classification labels
- R: (r_1,r_2)
  - Target reconstruction label
- P: predicted label
- *: reconstruction from a digit that is neither the label nor the prediction.

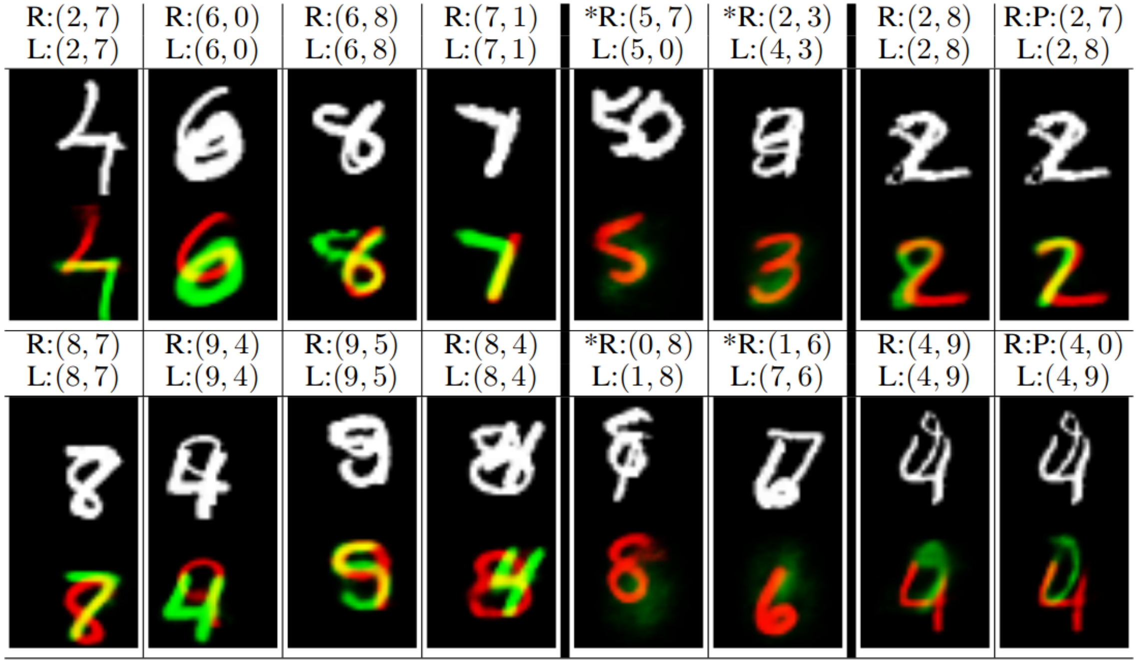
3 routing iterations



Red and **Green** are reconstructed digits (yellow: overlap)

MultiMNIST Reconstruction

3 routing iterations



CapsNet successfully deals with overlapping objects.

Red and Green are reconstructed digits (yellow: overlap)

Challenges Ahead CapsNet

- Not state-of-the-art in tasks like CIFAR 10 (good start!)
- Not tested yet on larger databased (e.g. ImageNet) due to technical issues
  - Slow training → Routing iterations
  - Memory problem
- A CapsNet cannot see two very close identical objects
  - “crowding” ↔ similar to human vision system

Wrap-up (1)

- Each capsule is a group of neurons
  - Expand artificial scalar neuron to vector
- Capsule represents an entity through a vector (inverse graphics)
  - Magnitude → probability of the entity presence → invariant
  - Phase → state of the entity → equivariant
- Dynamic routing: how capsules of two layers should communicate
- Parameters & Learning
  - Coupling coefficients (c_{ij}) → routing-by-agreement
  - Affine transformations (W_{ij}) → backpropagation

Wrap-up (2)

- Advantages:
  - Built-in disentanglement between entity’s pose (equivariant) and presence probability (invariant)
  - Dynamic hierarchical modelling, smarter than static max-pooling
  - Requires less data, higher robustness (viewpoint), interpretability
- Challenges:
  - Technical difficulties in scaling up (e.g. memory problem)
  - Performance is still not in the state-of-the-art level
    - e.g. CIFAR 10 (Error: 10.6% vs 3.47%)

That's it!

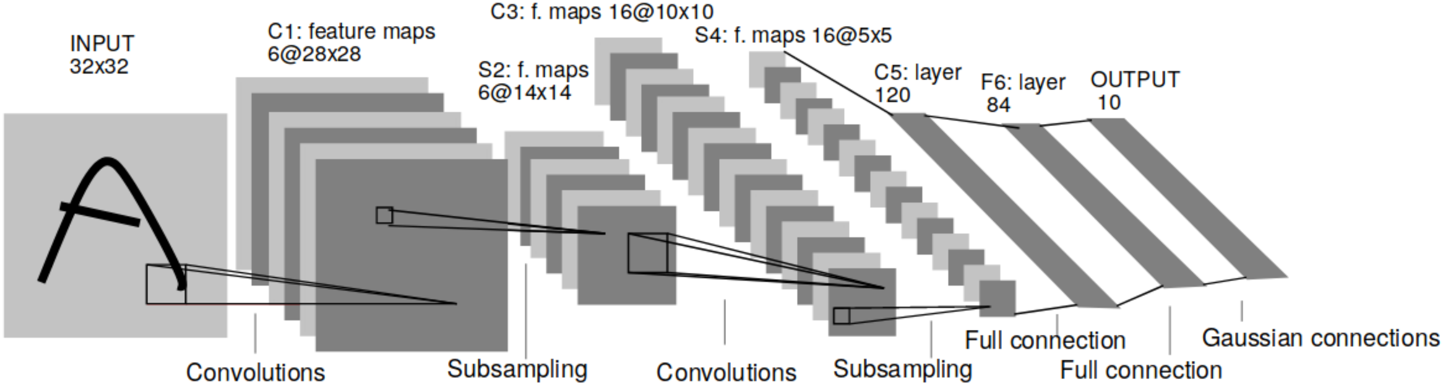
- Thanks for Your Attention
- Q/A
- Appendices
  - Appendix A: MNIST Database & its variants
  - Appendix B: (Small) NORB Database

MNIST Database

PROC. OF THE IEEE, NOVEMBER 1998

Gradient-Based Learning Applied to Document Recognition

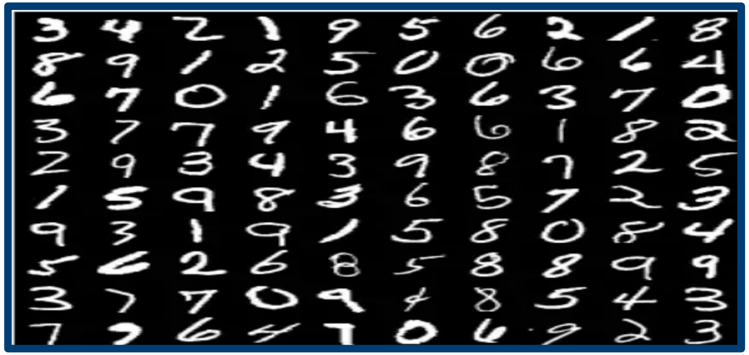
Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

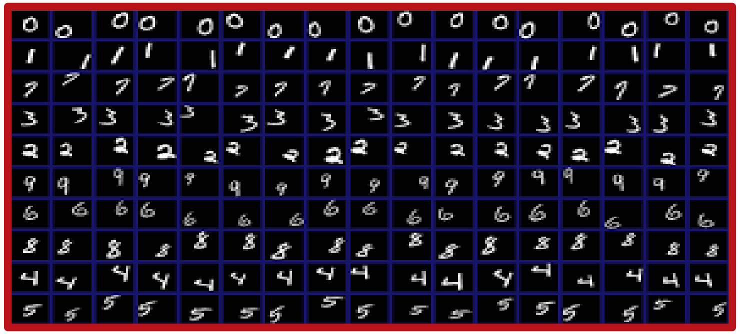


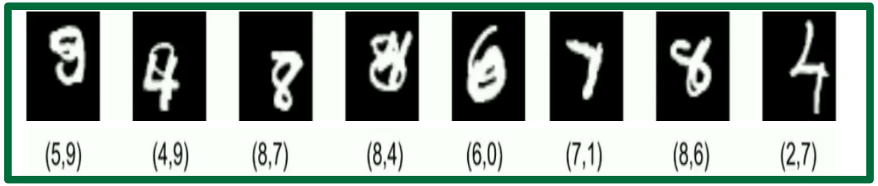
LeNet-5 Architecture

MNIST Database

- *Yan LeCun et al., 1998*
- Handwritten digits
  - 28 x 28
  - Training: 60 k
  - Test: 10 k
- Variants
  - affMNIST
  - MultiMNIST
  - EMNIST: letters+digits
    - train: 240k, test: 40k







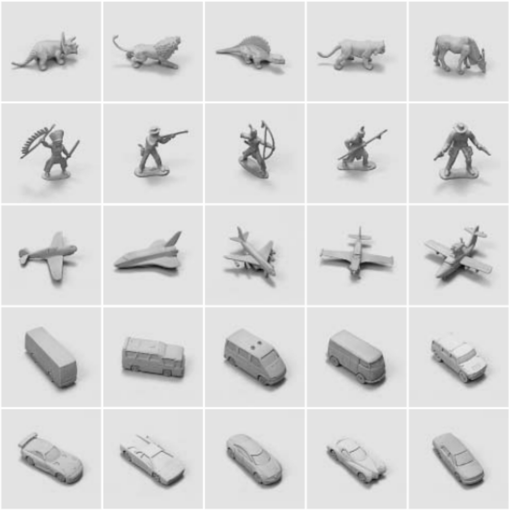
(Small) NORB Database

Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting

Yann LeCun, Fu Jie Huang,
The Courant Institute, New York University
715 Broadway, New York, NY 10003, USA
http://yann.lecun.com

Léon Bottou
NEC Labs America,
4 Independence Way, Princeton, NJ 08540
http://leon.bottou.org

Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)
1063-6919/04 $20.00 © 2004 **IEEE**



(Small) NORB Database

- *Y. LeCun et al., 2004*
- 3D object recognition task
  - 96 x 96 images of 50 toys, 5-generic categories
    - Animal, human, airplane, car, truck
- Objects where imaged by 2 cameras under ...
  - 6 Lighting conditions, 9 elevations, 18 azimuths
- Download
  - NORB → 29160 images
  - Small NORB→ 24300 images
    - Normalised object sizes and uniform background