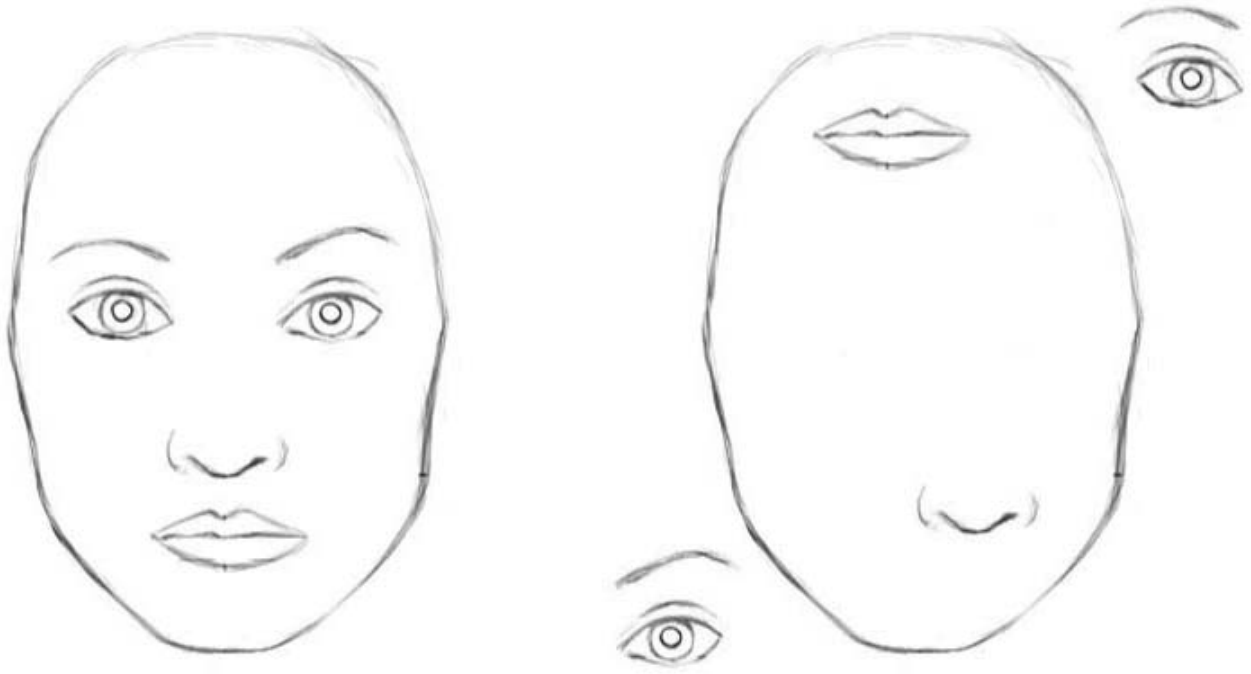
Dynamic Routing Between Capsules

PRESENTATION BY: KEVIN DUARTE

Motivation

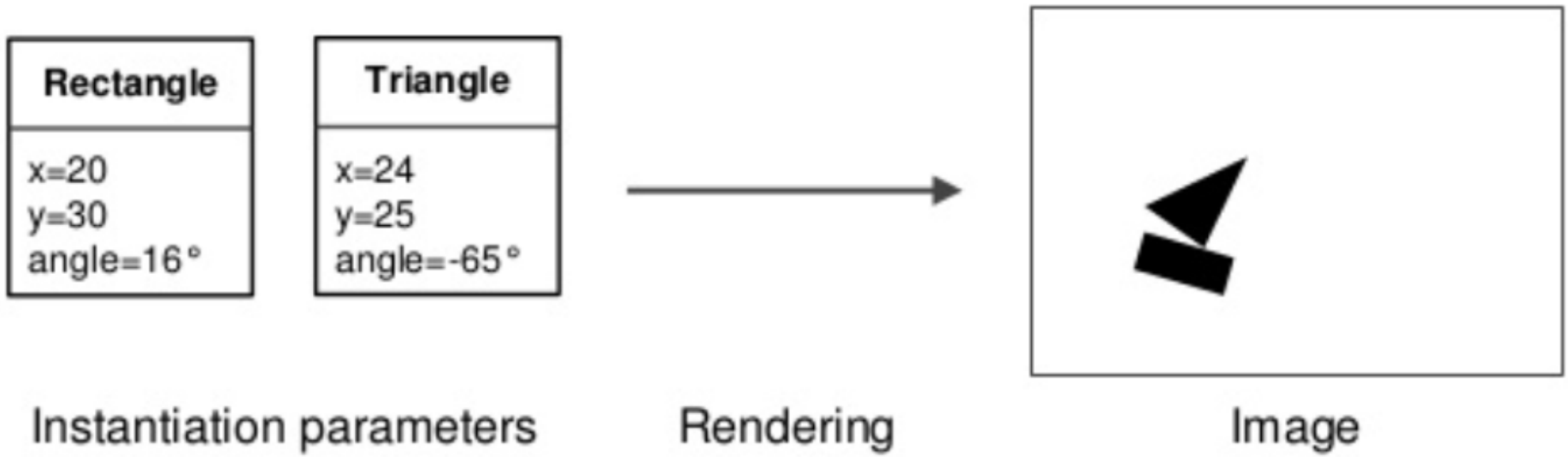
- Max Pooling
  - Reduces spatial size – loses information
- CNNs do not implicitly model entities
- Capsule Networks add extra structure
  - Models entities/features as vectors
  - Routing-by-agreement

Issues with CNNs



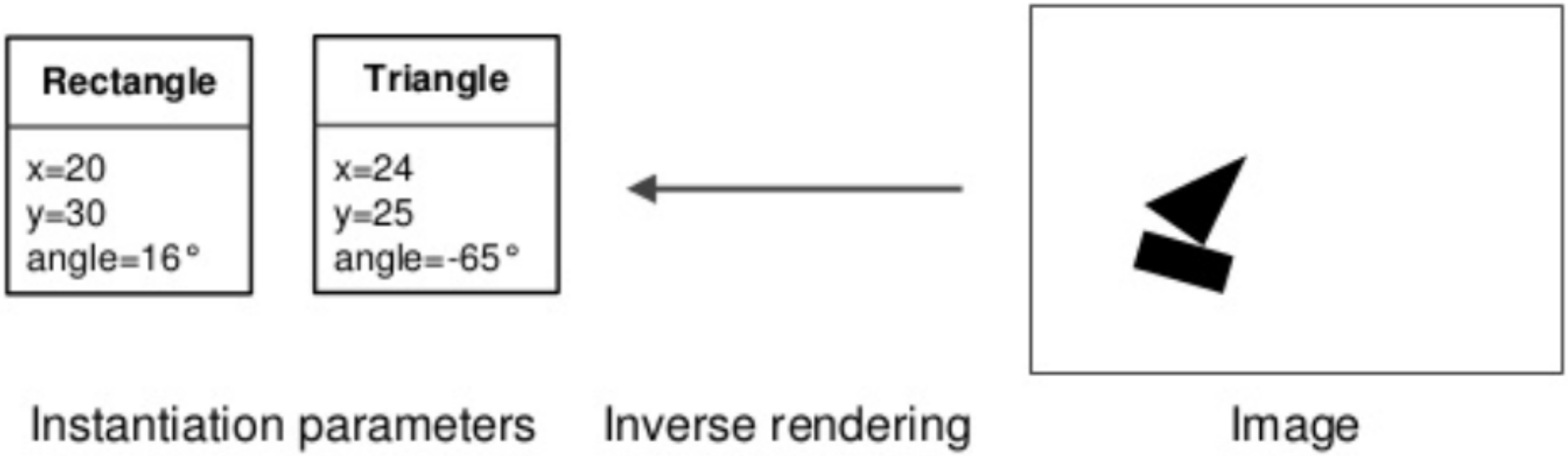
Pechyonkin, M. (2017, November 03). Understanding Hinton's Capsule Networks. Part I: Intuition. from https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks-part-i-intuition-b4b559d1159b

Computer Graphics



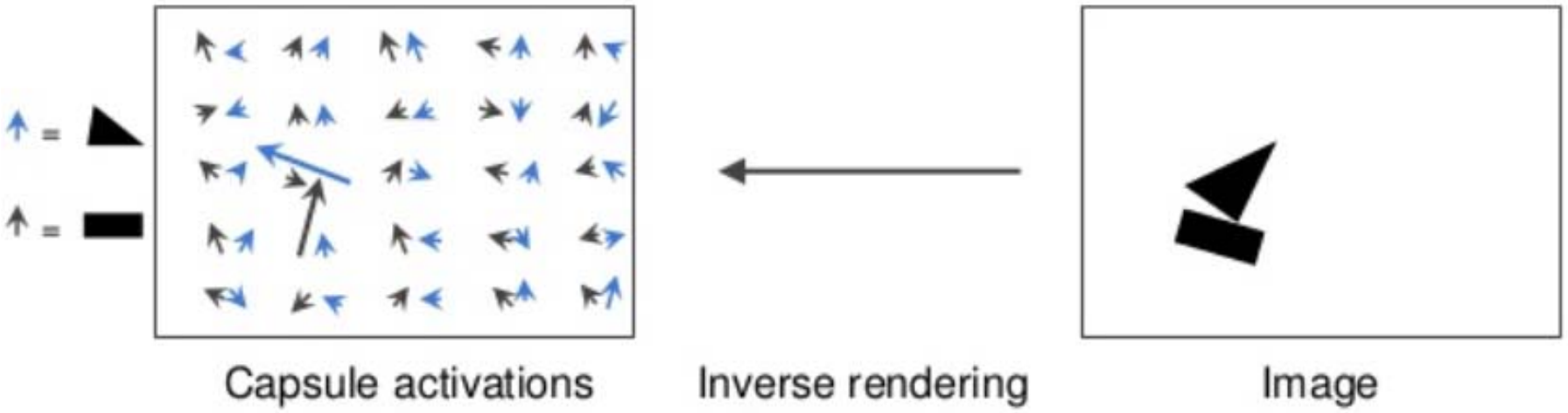
Aurélien Géron (2017). Introduction to Capsule Networks (CapsNets).
https://www.slideshare.net/aureliengeron/introduction-to-capsule-networks-capsnets

Inverse Graphics



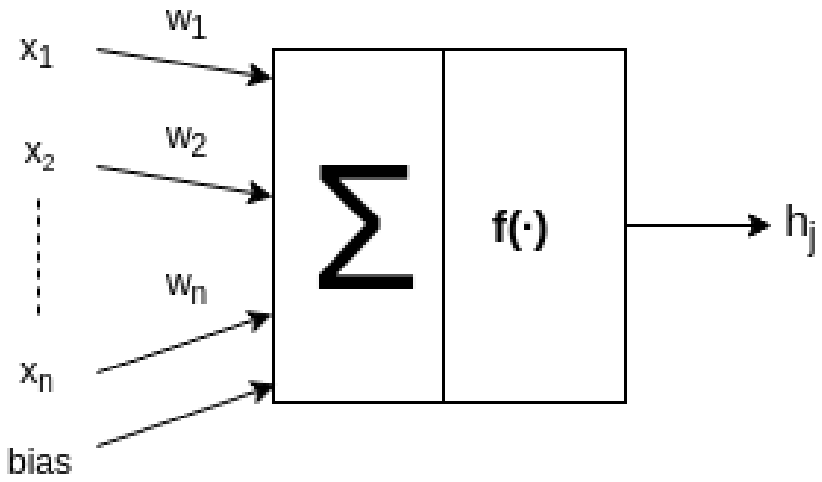
Aurélien Géron (2017). Introduction to Capsule Networks (CapsNets).
https://www.slideshare.net/aureliengeron/introduction-to-capsule-networks-capsnets

Capsules



Aurélien Géron (2017). Introduction to Capsule Networks (CapsNets).
https://www.slideshare.net/aureliengeron/introduction-to-capsule-networks-capsnets

Traditional Neuron



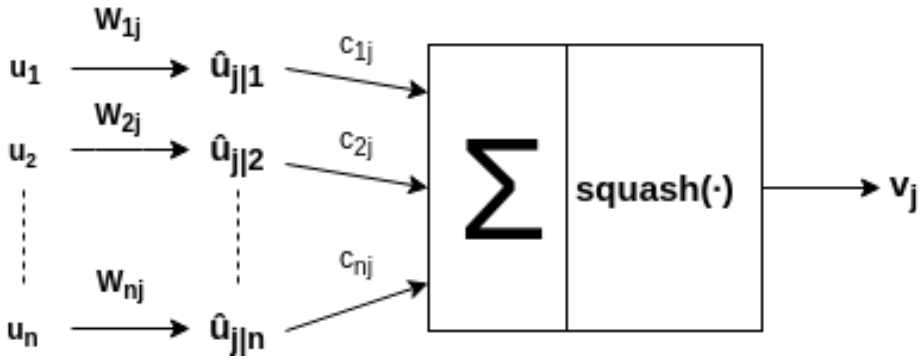
Given input scalars x_i

$$a_j = \sum_i x_i w_i + b$$

$$h_j = f(a_j)$$

Output is a scalar h_j

Capsules



Given input vectors $\boldsymbol{u_i}$

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

$$\boldsymbol{s_j} = \sum_i c_{ij}\hat{\boldsymbol{u}}_{\boldsymbol{j|i}}$$

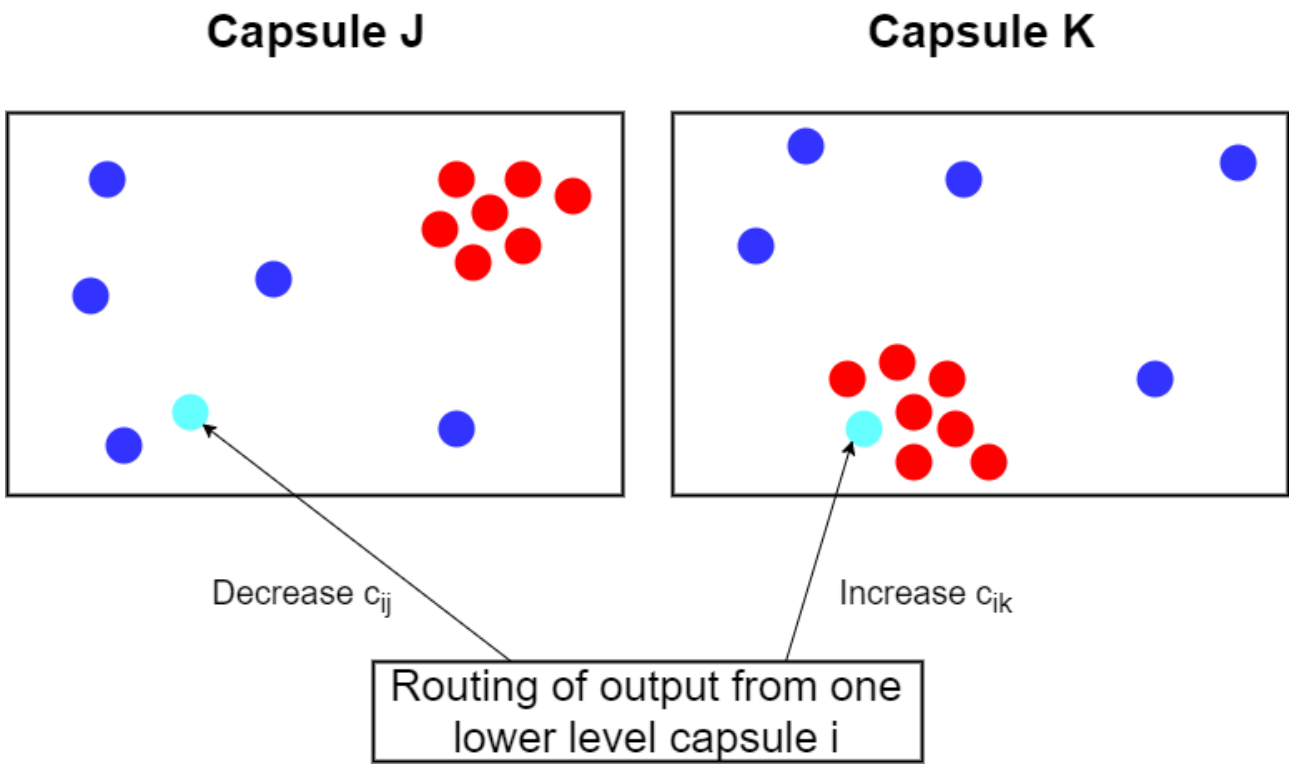
$$\boldsymbol{v_j} = \frac{||\boldsymbol{s_j}||^2}{1+||\boldsymbol{s_j}||^2}\frac{\boldsymbol{s_j}}{||\boldsymbol{s_j}||}$$

Output is a vector $\boldsymbol{v_j}$

Capsule vs. Traditional Neuron			
Input from low-level capsule/neuron		vector($\mathbf{u}_i$)	scalar(x_i)
Operation	Affine Transform	$\widehat{\mathbf{u}}_{j\|i} = \mathbf{W}_{ij}\mathbf{u}_i$	–
	Weighting	$\mathbf{s}_j = \sum_i c_{ij}\widehat{\mathbf{u}}_{j\|i}$	$a_j = \sum_i w_i x_i + b$
	Sum		
	Nonlinear Activation	$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}$	$h_j = f(a_j)$
Output		vector($\mathbf{v}_j$)	scalar(h_j)

Pechyonkin, M. (2017, November 15). Understanding Hinton's Capsule Networks. Part II: How Capsules Work. from https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks-part-ii-how-capsules-work-153b6ade9f66

Routing-by-Agreement



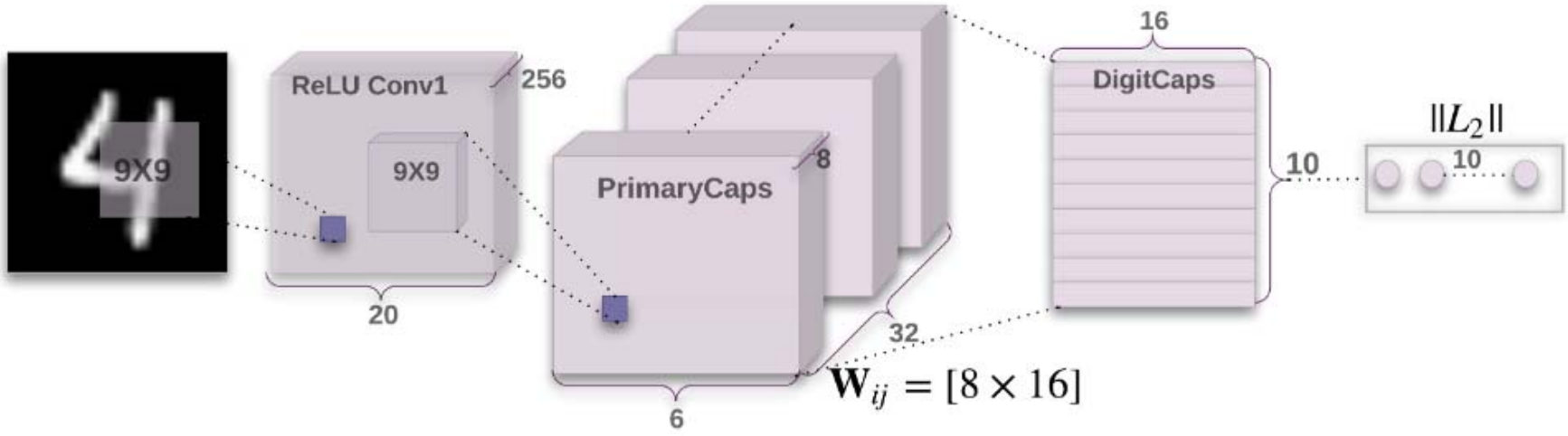
Routing Algorithm

Procedure 1 Routing algorithm.

1: **procedure** ROUTING($\hat{u}_{j|i}$, 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 $\mathbf{v}_j$

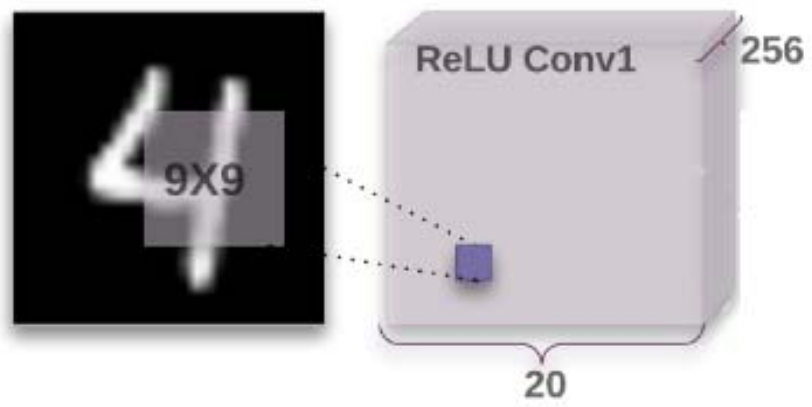
Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules."

CapsNet Architecture



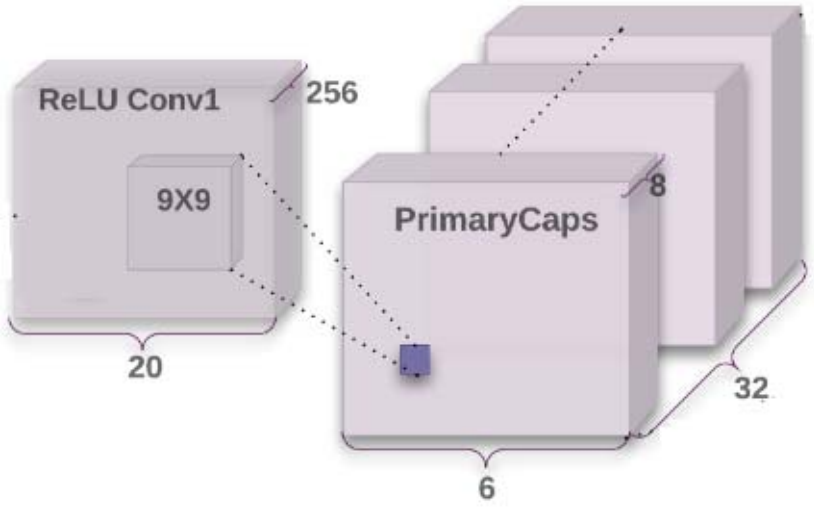
Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules."

Conv1



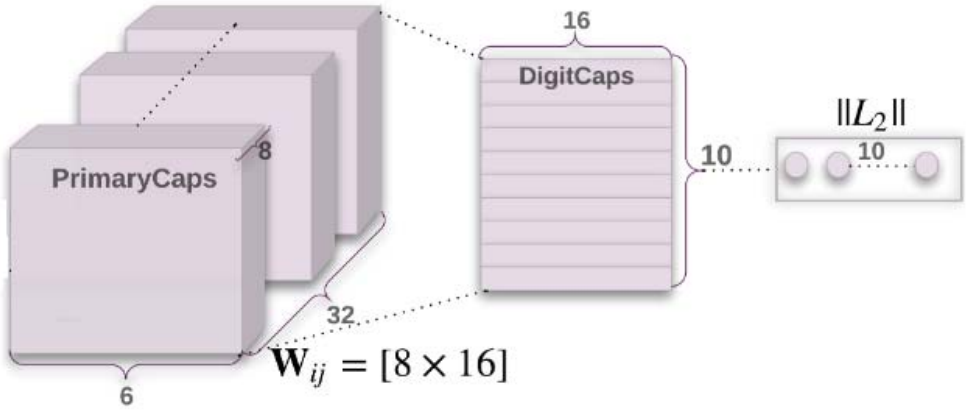
- Input: 28x28x1 image
- Conv1: Convolutional Layer
  - ReLU activation
  - 9x9 Filter
  - Stride of 1
  - 256 feature maps
  - Result has shape (20, 20, 256)

PrimaryCaps



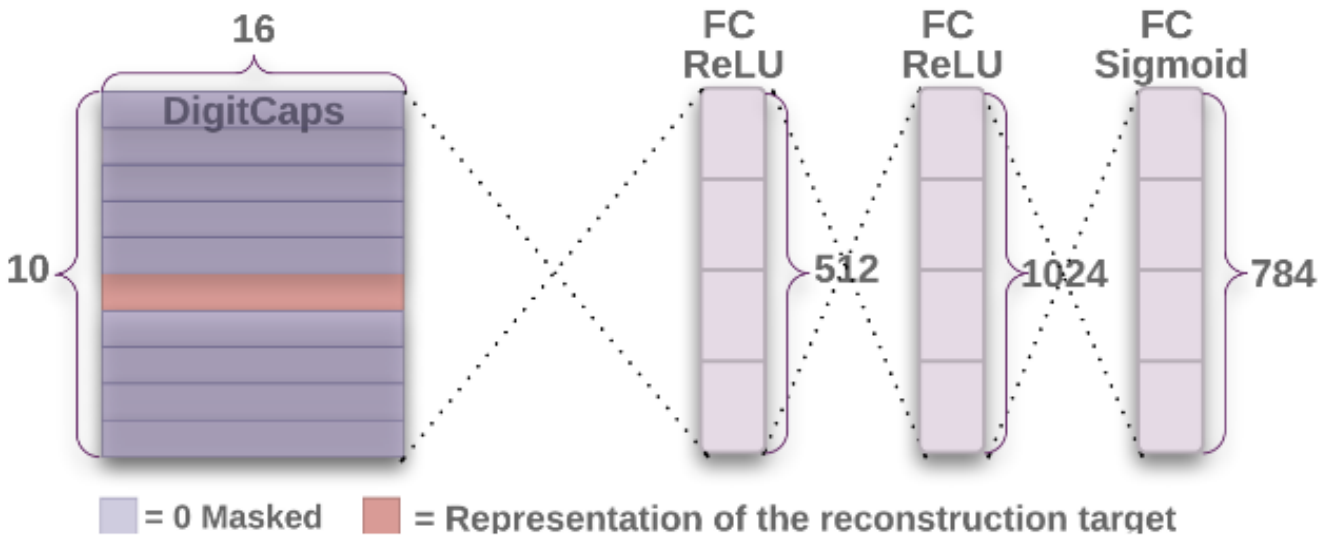
- Add another Convolutional Layer
  - ReLU activation
  - 9x9 filter
  - Stride of 2
  - (32*8)=256 feature maps
  - Result has shape (6, 6, 256)
- Reshape into capsules with 8-d vectors
  - 32 channels of capsules
  - Result has shape (6, 6, 32, 8)
  - Squash the capsule vectors
- Total of (6*6*32)=1152 different capsules

DigitCaps



- Multiply PrimaryCaps with matrices W_{ij}
  - $\hat{\boldsymbol{u}}_{\boldsymbol{j}|\boldsymbol{i}} = W_{ij}\boldsymbol{u}_{\boldsymbol{i}}$
  - 8-d vectors transform into 16-d vectors
- Iterative routing algorithm
  - $\boldsymbol{s_j} = \sum_i c_{ij}\hat{\boldsymbol{u}}_{\boldsymbol{j}|\boldsymbol{i}}$
  - $\boldsymbol{v_j} = \frac{||\boldsymbol{s_j}||^2}{1+||\boldsymbol{s_j}||^2}\frac{\boldsymbol{s_j}}{||\boldsymbol{s_j}||}$
- Result has shape (10, 16)
  - 10 capsules, with 16 dimensions

CapsNet Architecture - Reconstruction



Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules."

Loss Function

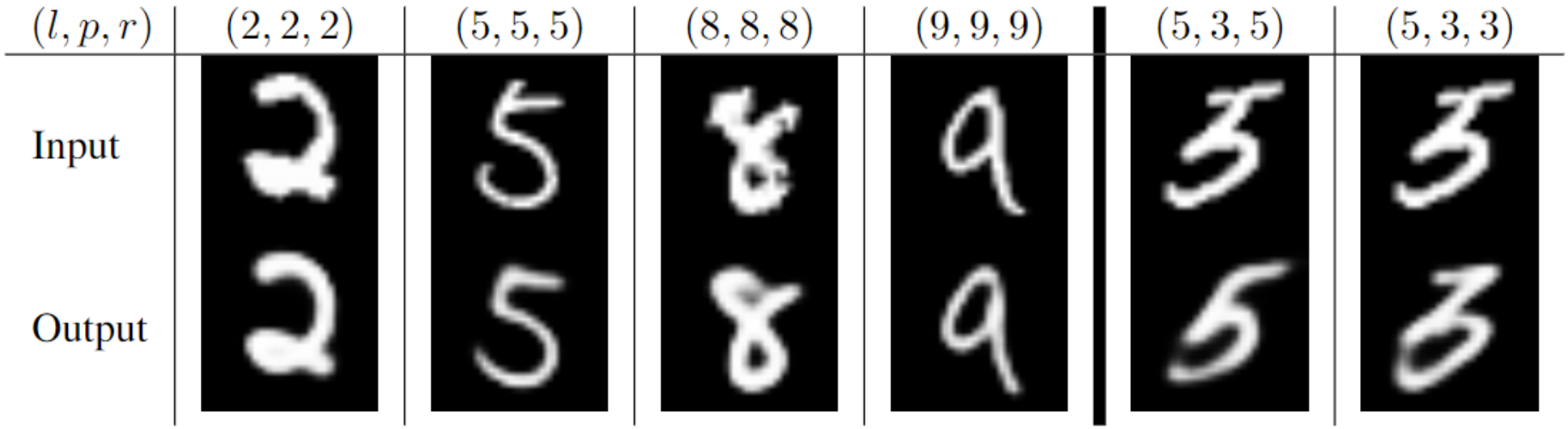
- Margin Loss
  - $L_c = T_c \max(0, m^+ - ||\boldsymbol{v_c}||)^2 + \lambda(1 - T_c)\max(0, ||\boldsymbol{v_c}|| - m^-)^2$
- Reconstruction Loss – L2 Loss
  - $L_r = \sum_{i=0}^{n}(y_i - h(x_i))^2$
- Total Loss
  - $L = \sum_c L_c + 0.0005 L_r$

Results on MNIST

Method	Routing	Reconstruction	MNIST (%)	MultiMNIST (%)
Baseline	-	-	0.39	8
CapsNet	1	no	$0.34_{\pm 0.032}$	-
CapsNet	1	yes	$0.29_{\pm 0.011}$	7
CapsNet	3	no	$0.35_{\pm 0.036}$	-
CapsNet	3	yes	$\mathbf{0.25}_{\pm 0.005}$	**5**

Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules."

Results on MNIST - Reconstruction



Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules."

What Capsule Dimensions Represent

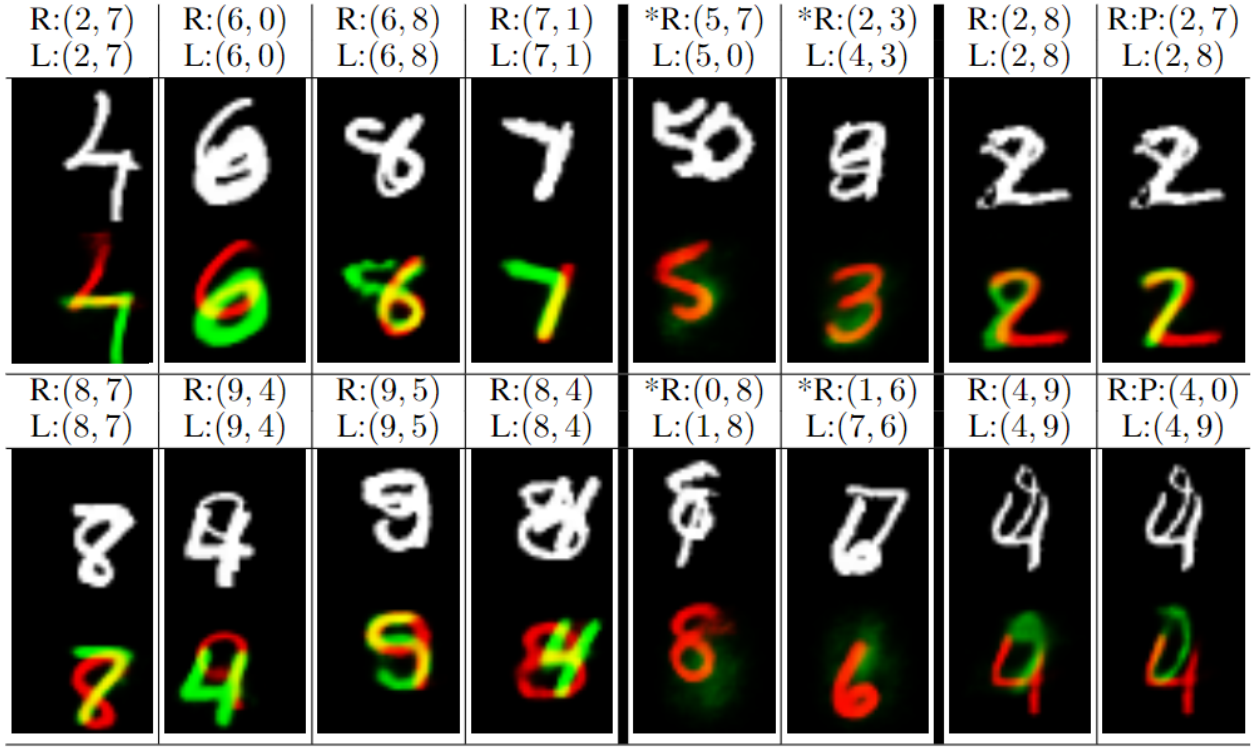
Scale and thickness	
Localized part	
Stroke thickness	
Localized skew	
Width and translation	
Localized part	

Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules."

Results on affNIST

- More robust to small affine transformations
  - 79% on affNIST test set
  - Traditional CNN achieved 66%

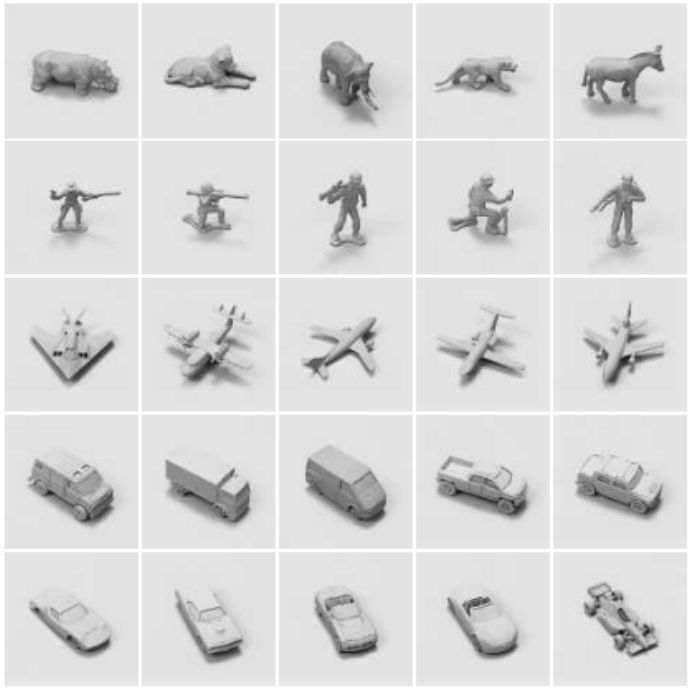
Results on multiMNIST



Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules."

Other Dataset

- CIFAR10: 10.6% error
  - Ensemble of 7 models
  - Not state of the art
- SVHN: 4.3% error
- smallNORB: 2.7% error



Y. LeCun, F.J. Huang, L. Bottou, “Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting.”