Capsule Networks

Przemysław Pobrotyn, Sigmoidal
przemek@sigmoidal.io

Main reference for the talk:

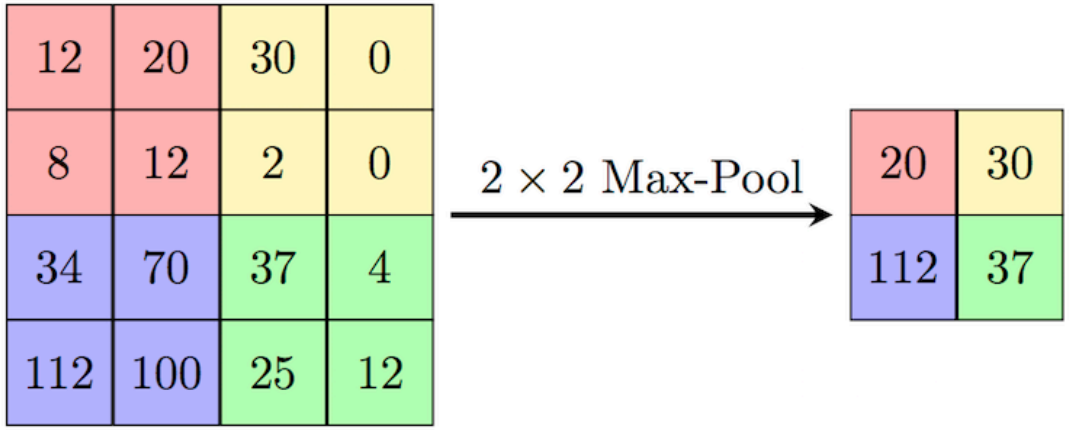
Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In NIPS, 2017.

Motivation: Shortcomings of CNNs

- CNNs are **translation invariant** - they do not capture translations in data.
- CNNs are **poor in learning spatial relationships** among features (perspective, size, orientation, relative position).
- They have to learn different filters for each different viewpoint, and doing so requires **a lot of data**.
- Susceptible to adversarial attacks
- Using data augmentation, we make CNNs also **rotationally invariant** (slight rotation of the object -> same output)

The culprit: MaxPooling ie Primitive Routing

"The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster." - G. Hinton



Dynamic routing:

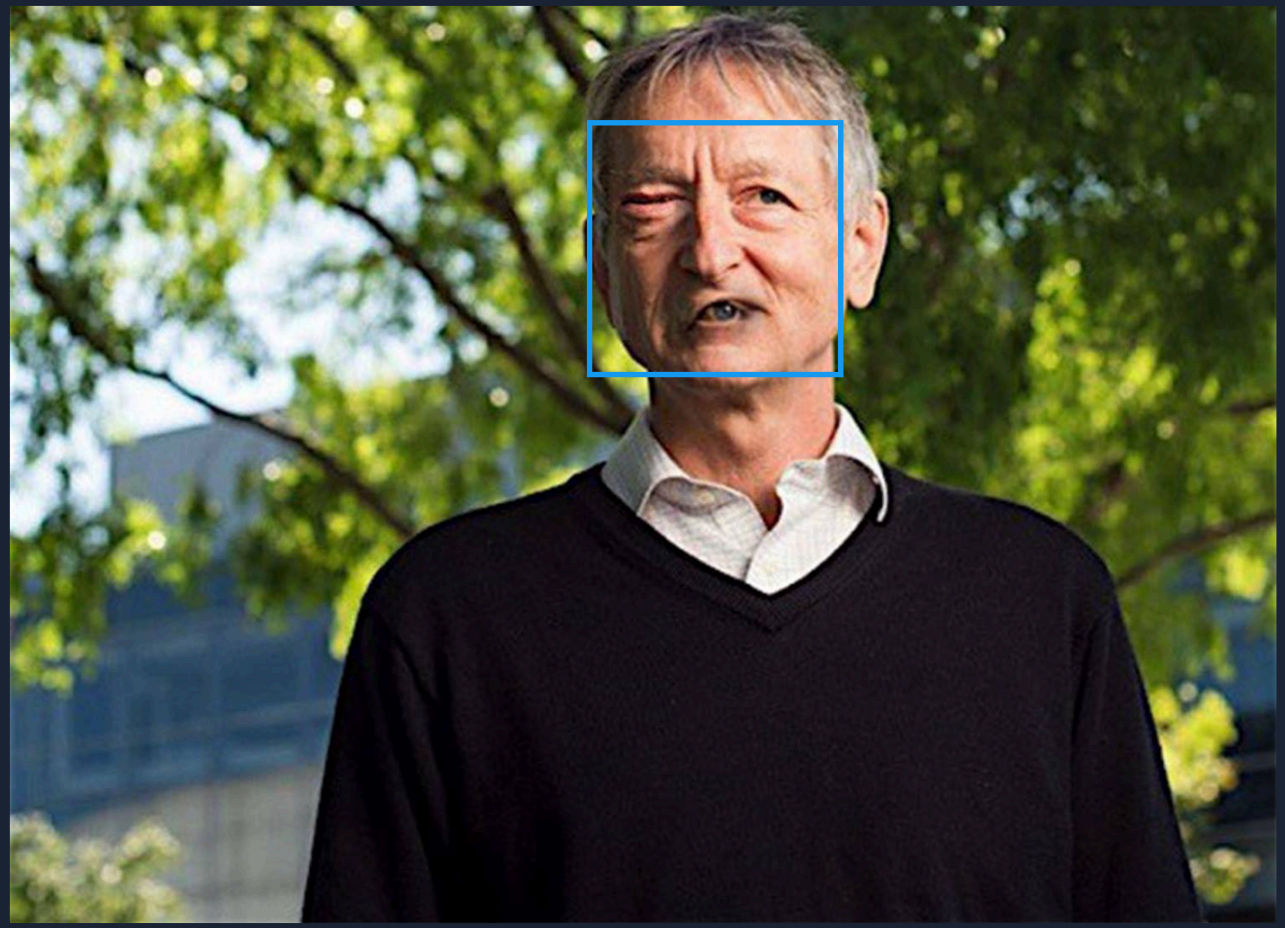
Determine how to route data from lower to higher level layers in real time, based on data (for each forward pass)

Image source: https://computersciencewiki.org/index.php/Max-pooling_/_Pooling

clarifai

PRODUCTS SOLUTIONS DEVELOPERS COMPANY DEMO PRICING LOG IN

GENERAL FACE NSFW COLOR MORE MODELS



Face VIEW DOCS

1 FACE DETECTED

Face VIEW DOCS

No face detected

CapsNets Goals

- Encode information about pose (translation, rotation, deformation etc) of an entity (instantiation parameters) - this information should be **equivariant**, not invariant
- Preserve hierarchical pose relationships between object parts
- Route information between layers dynamically
- As a result, generalize with less training data

A Capsule

- Capsule = vector
- Length = probability of detection of a feature (invariant)
- Direction = pose of the detected feature, ie instantiation parameters (equivariant)

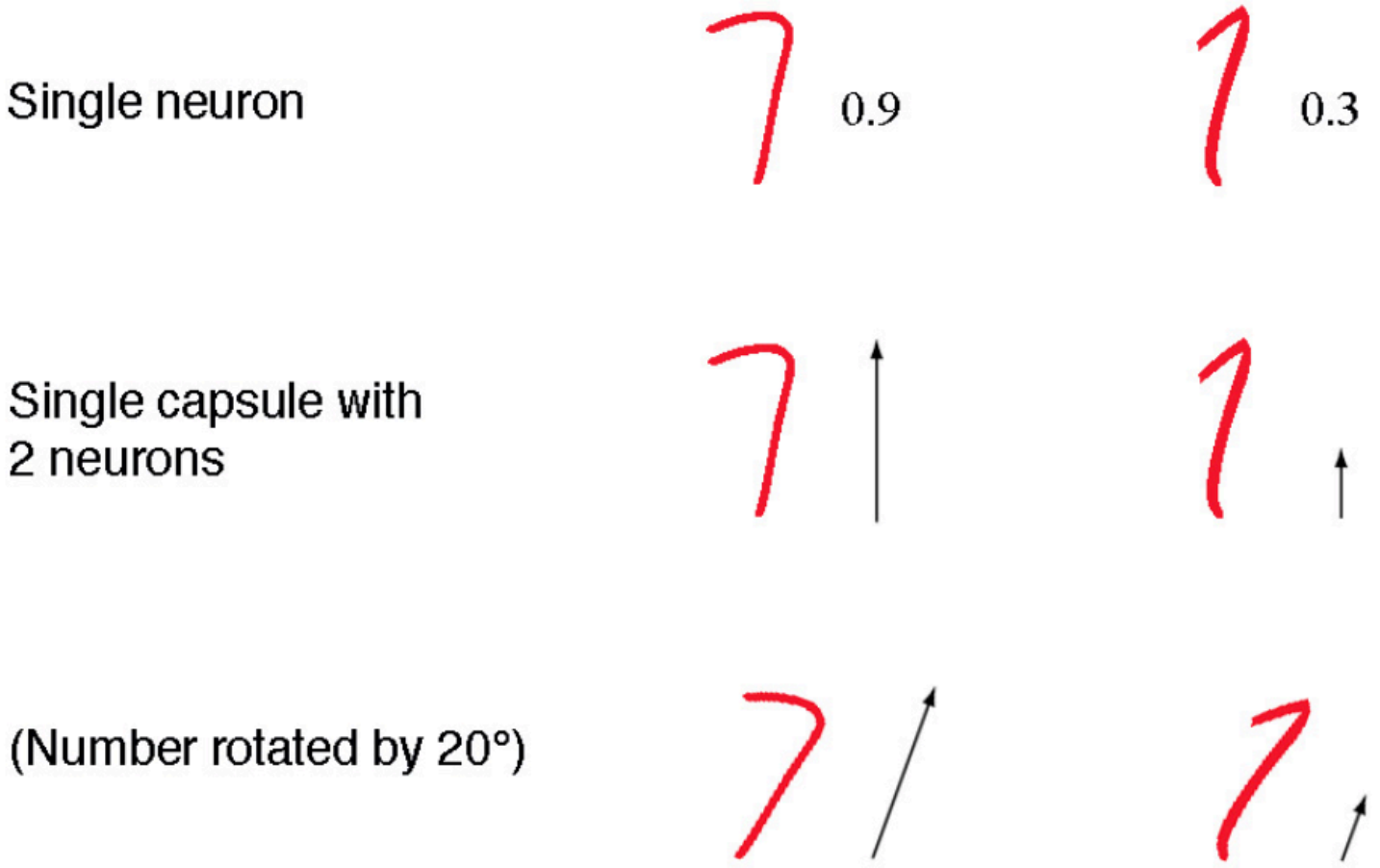


Image source: https://jhui.github.io/2017/11/03/Dynamic-Routing-Between-Capsules/

Forward Pass in a Capsule Network

Capsule vs. Traditional Neuron			
Input from low-level capsule/neuron		vector($\mathbf{u}_i$)	scalar(x_i)
Operation	Affine Transform	$\hat{\mathbf{u}}_{j\|i} = \mathbf{W}_{ij}\mathbf{u}_i$	–
	Weighting	$\mathbf{s}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j\|i}$	$a_j = \sum_i w_i x_i + b$
	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)

Source: https://medium.com/ai³-theory-practice-business/understanding-hintons-capsule-networks-part-ii-how-capsules-work-153b6ade9f66

Affine Transformation

$$\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i$$

- Matrices W_{ij} encode important spatial and other relationships between lower and higher level features (capsules)
- They are learned using backprop
- Having multiplied a lower level capsule by the W_{ij} matrix, we obtain a predicted position of a higher level feature based on the position of the lower level feature
- Intuition: if lower level features agree to the pose of the higher level feature, it's likely that feature is present

Weighting: Dynamic Routing by Agreement

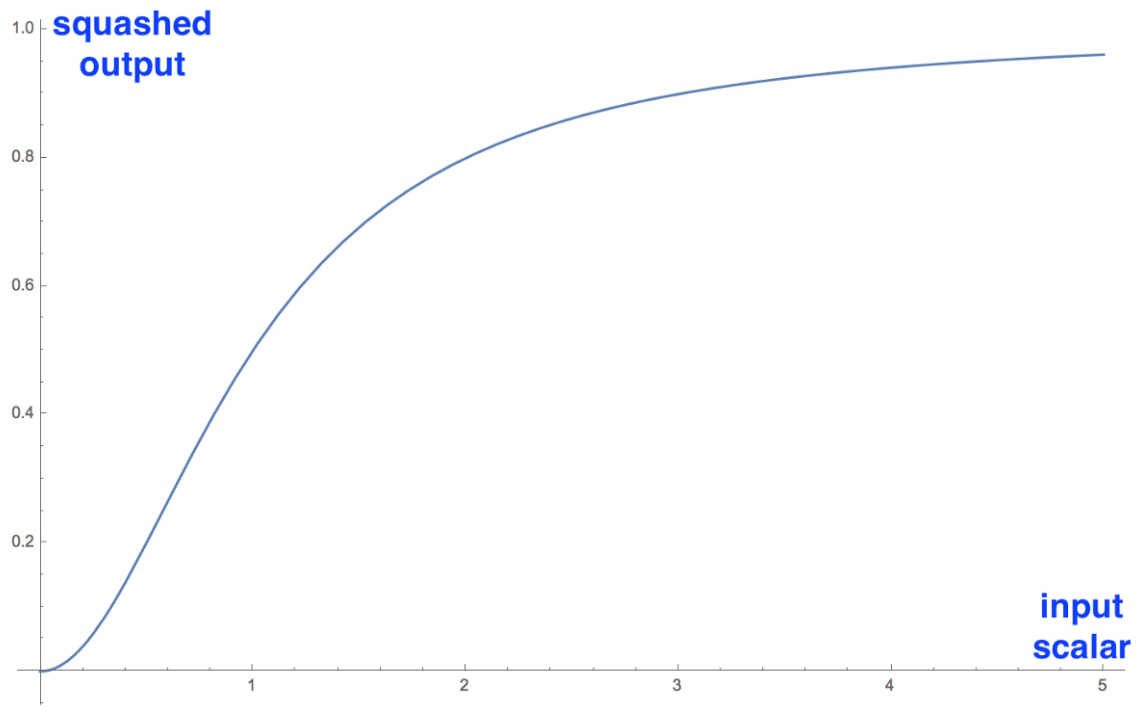
$$\mathbf{s}_j = \sum_i c_{ij} \hat{\mathbf{u}}_{j|i}$$

- Weights c_{ij} are NOT learned during backprop!
- Instead, they are **dynamically** determined in each forward pass
- Weights c_{ij} are computed so that signal from lower level capsules is sent to the higher level capsule for which lower level capsules predict the same pose (routing by agreement)

Squashing

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

additional "squashing" unit scaling



Source: https://medium.com/ai³-theory-practice-business/understanding-hintons-capsule-networks-part-ii-how-capsules-work-153b6ade9f66

Dynamic Routing by Agreement

Procedure 1 Routing algorithm.

1: **procedure** ROUTING($\hat{\boldsymbol{u}}_{j|i}, r, l$)
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: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$
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 $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$
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 $\mathbf{v}_j$

Source: Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In NIPS, 2017.

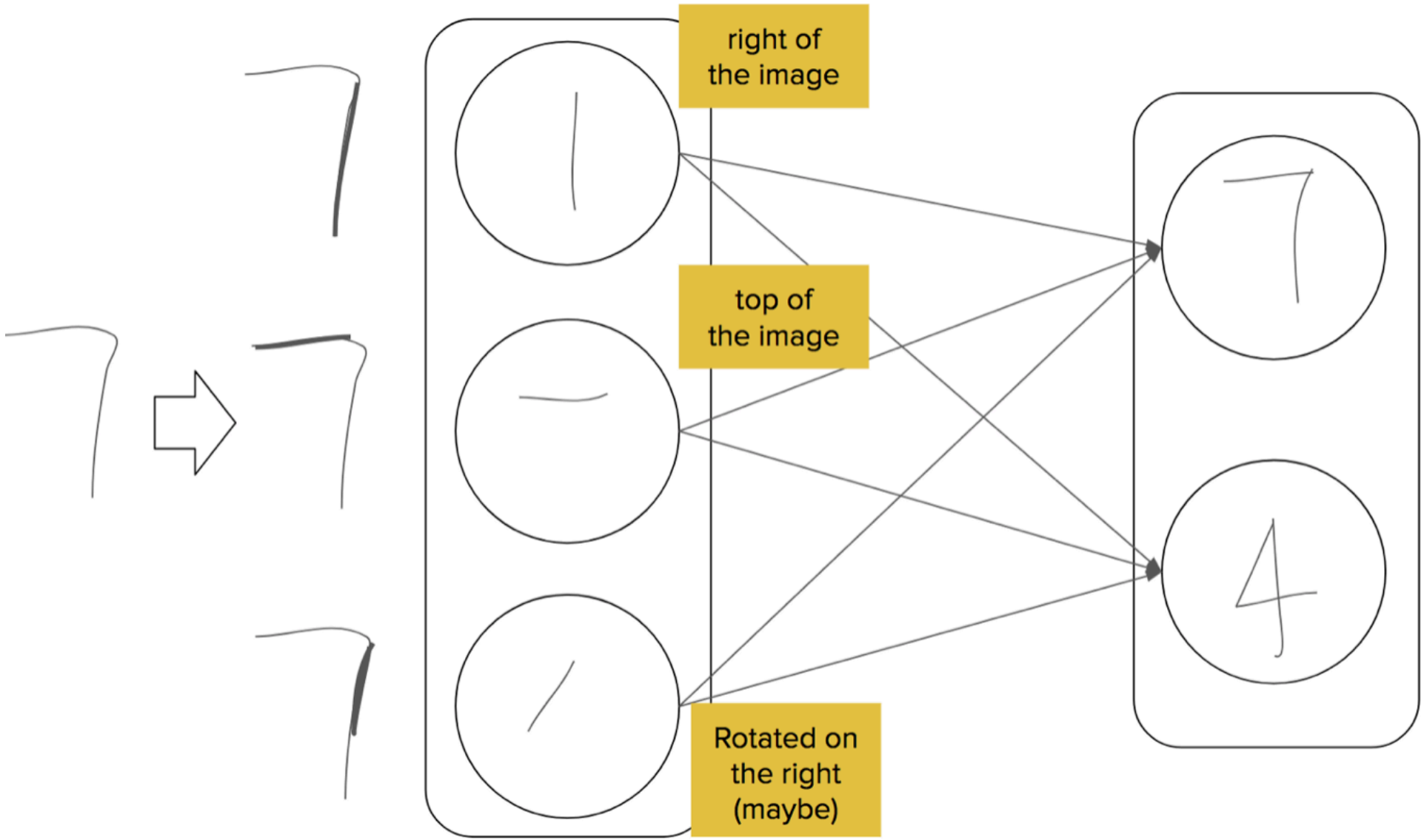


Image source: https://hackernoon.com/uncovering-the-intuition-behind-capsule-networks-and-inverse-graphics-part-i-7412d121798d

Dynamic Routing by Agreement

- Lower level capsules are only routed to appropriate higher level capsules, thus higher level capsules receive less noise and have cleaner pose
- We obtain a clear hierarchy of parts: enough to trace the activations and their routes
- Helps parsing crowded scenes with overlapping objects: information tends to be routed so that it best explains all the entities present in the scene

CapsNet for MNIST - Encoder

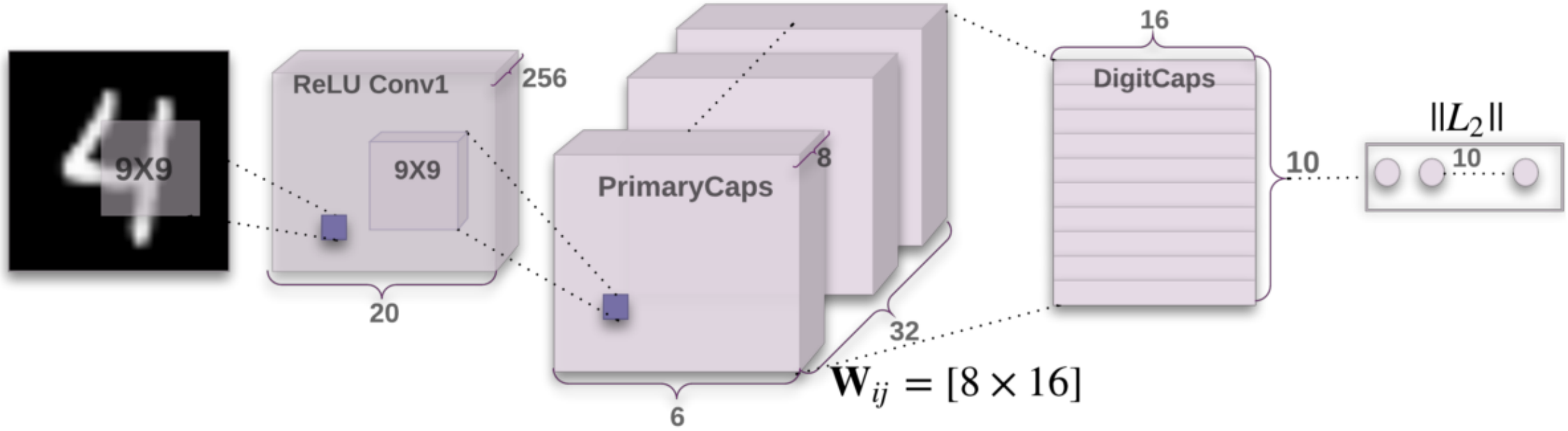


Image source: Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In NIPS, 2017.

CapsNet for MNIST - Decoder

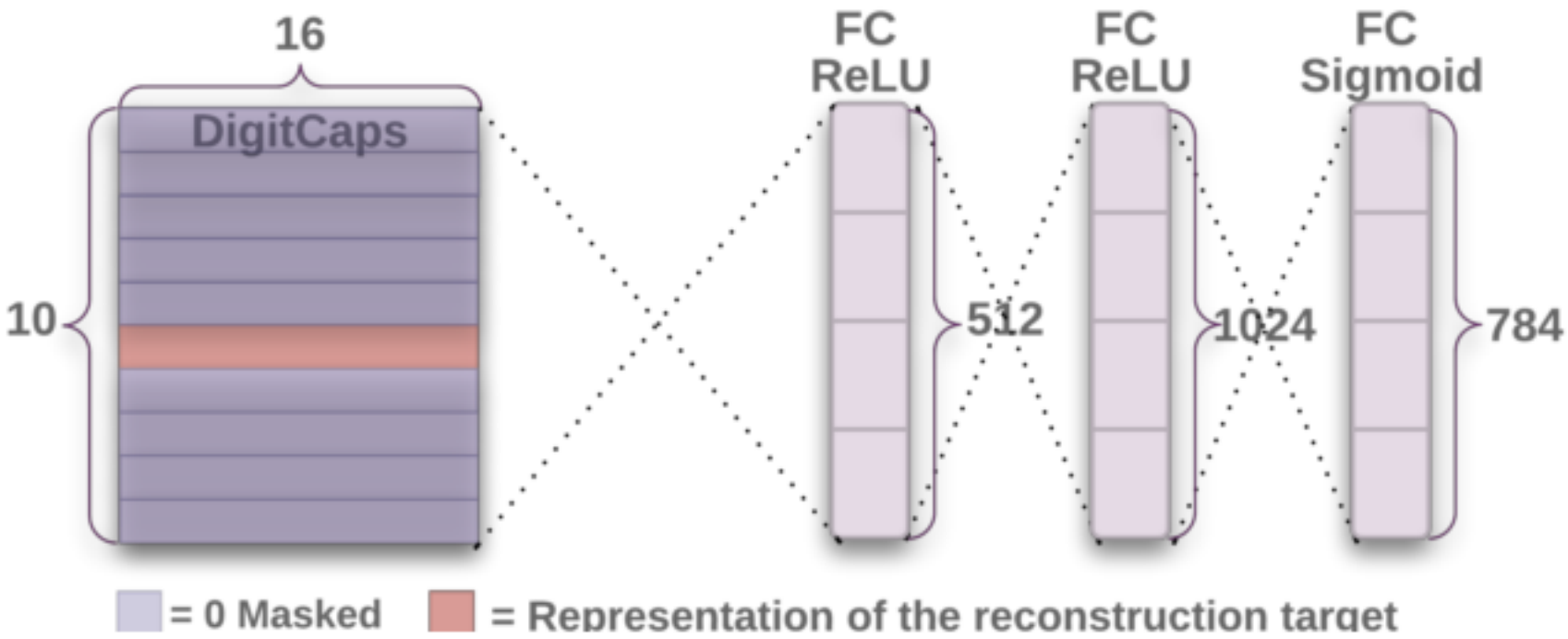


Image source: Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In NIPS, 2017.

Results

Table 1: CapsNet classification test accuracy. The MNIST average and standard deviation results are reported from 3 trials.

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**

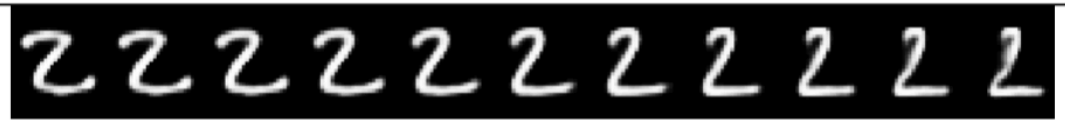
Training was performed on 28x28 MNIST images that have been shifted by up to 2 pixels in each direction with zero padding. No other data augmentation/deformation was used.

Wan et al. [2013] achieved 0.21% test error with ensembling and augmenting the data with rotation and scaling. They achieved 0.39% without them.

CapsNet - 8.2M params, Baseline - 35.4M params

Interpretation of instantiation parameters

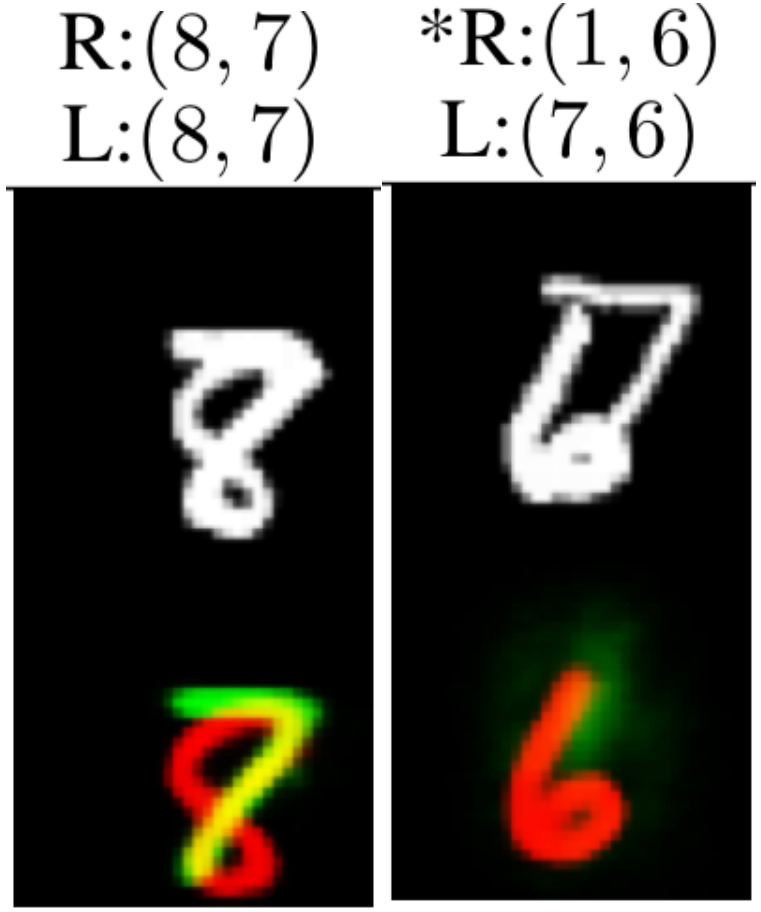
Figure 4: Dimension perturbations. Each row shows the reconstruction when one of the 16 dimensions in the DigitCaps representation is tweaked by intervals of 0.05 in the range $[-0.25, 0.25]$.

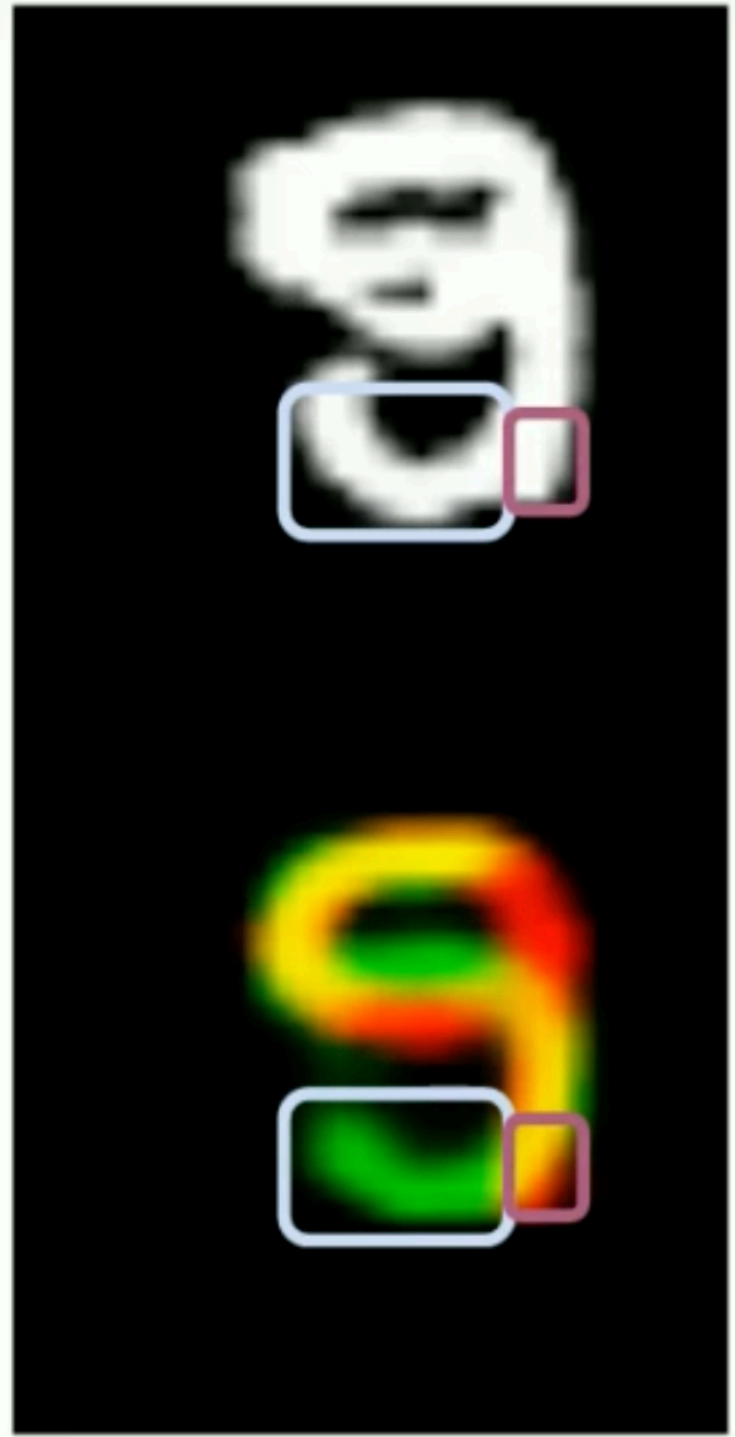
Scale and thickness	
Localized part	
Stroke thickness	
Localized skew	
Width and translation	
Localized part	

Cool visualisation tool for CapsNet: https://github.com/bourdakos1/CapsNet-Visualization

Segmenting highly overlapping digits

MultiMNIST - generated by overlaying a digit on top of another digit from the same set (training and testing) but different classes. For each digit in MNIST, 1K MultiMNIST examples were generated. Thus there were 60M training and 10M testing examples.





Summary: Pros and Cons

- Need less data to be trained
- Model the hierarchy of entities
- Keep the information about the pose of the entity
- Promising for segmentation
- Fewer parameters for the same performance as ConvNets
- Still early in development

- Slow training (mostly due to dynamic routing)
- at most one instance of the type of entity that a capsule represents at each location of the image
- capsules like to account for everything in the image, thus they struggle if images have very varied backgrounds (CIFAR-10)
- Still early in development

State-of-the-art research

Matrix capsules with EM routing

Hinton, G. E., Sabour, S. and Frosst, N. (Feb 2018)

- Capsules are now matrices (pose) and logistic units (probability)
- More complex routing algorithm, using Expectation-Maximizaiton

Capsules for Object Segmentation

LaLonde, R. and Bagci, U. (Apr 2018)

- Introduced *deconvolutional capsules*
- Improved routing algorithm to reduce number of parameters
- State-of-the-art performance, 95% (!) fewer parameters than U-net

VideoCapsuleNet: A Simplified Network for Action Detection

Duarte, K., Rawat, Y.S. and Shah, M. (May 2018)

- Introduced a 3D capsule network for videos
- VideoCapsuleNet for action segmentation along with action classification
- Improvements to routing algorithm, introduction of *capsule-pooling*

References

- Papers:
  - Dynamic Routing between Capsules, Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. (2017)
  - Transforming Auto-encoders - Hinton, G. E., Krizhevsky, A. and Wang, S. D. (2011)
  - Matrix capsules with EM routing, Hinton, G. E., Sabour, S. and Frosst, N. (2018)
  - Capsules for Object Segmentation, LaLonde, R. and Bagci, U. (2018)
  - VideoCapsuleNet: A Simplified Network for Action Detection, Duarte, K., Rawat, Y.S. and Shah, M. (2018)
- Talks:
  - Geoffrey Hinton's talk: What is wrong with convolutional neural nets? - Talk given at MIT. Brain & Cognitive Sciences - Fall Colloquium Series., www.youtube.com/watch?v=rTawFwUvnLE
  - Capsule Networks (CapsNets) – Tutorial, Aurelien Geron, www.youtube.com/watch?v=pPN8d0E3900
  - Sara Sabour, Dynamic Routing Between Capsules www.youtube.com/watch?v=gq-7HgzfDBM
- Blog posts/ tutorials:
  - Understanding Hinton's Capsule Networks - Max Pechyonkin's series, medium.com/ai^3-theory-practice-business/understanding-hintons-capsule-networks-part-i-intuition-b4b559d1159b
  - hackernoon.com/uncovering-the-intuition-behind-capsule-networks-and-inverse-graphics-part-i-7412d121798d
  - https://blog.acolyer.org/2017/11/14/matrix-capsules-with-em-routing/
  - A Visual Representation of Capsule Connections in Dynamic Routing Between Capsules - Mike Ross's diagram medium.com/@mike_ross/a-visual-representation-of-capsule-network-

Awesome collection of Capsule resources: github.com/aisummary/awesome-capsule-networks