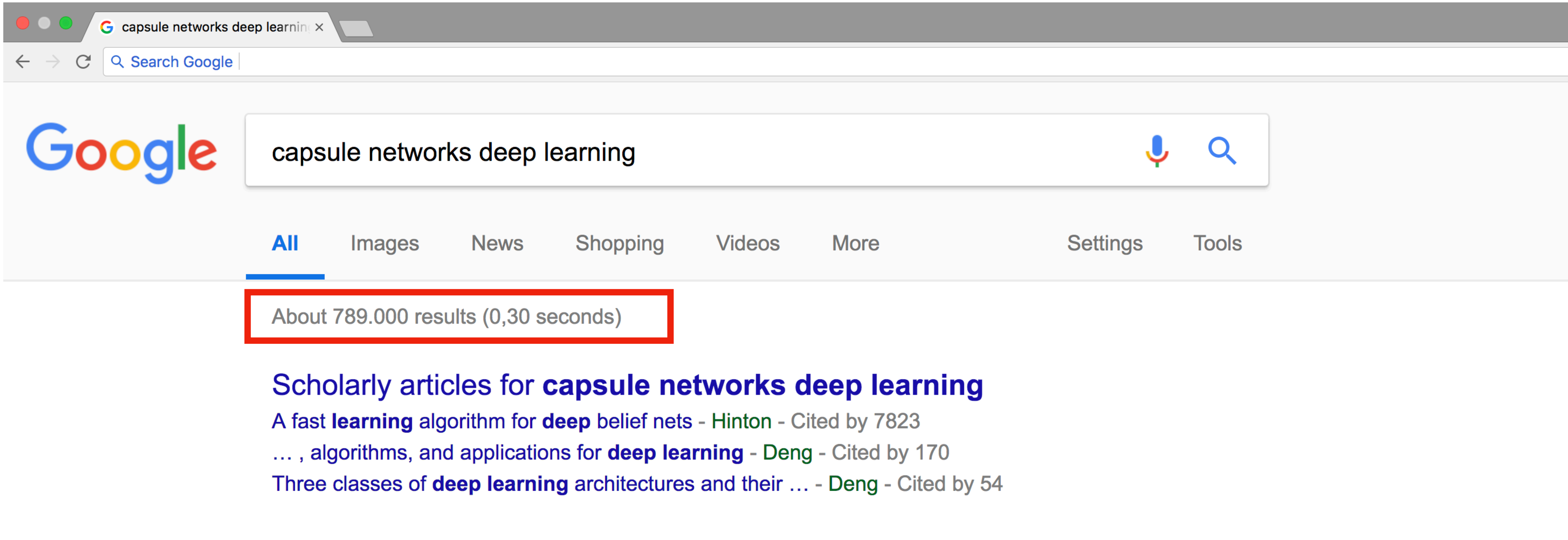
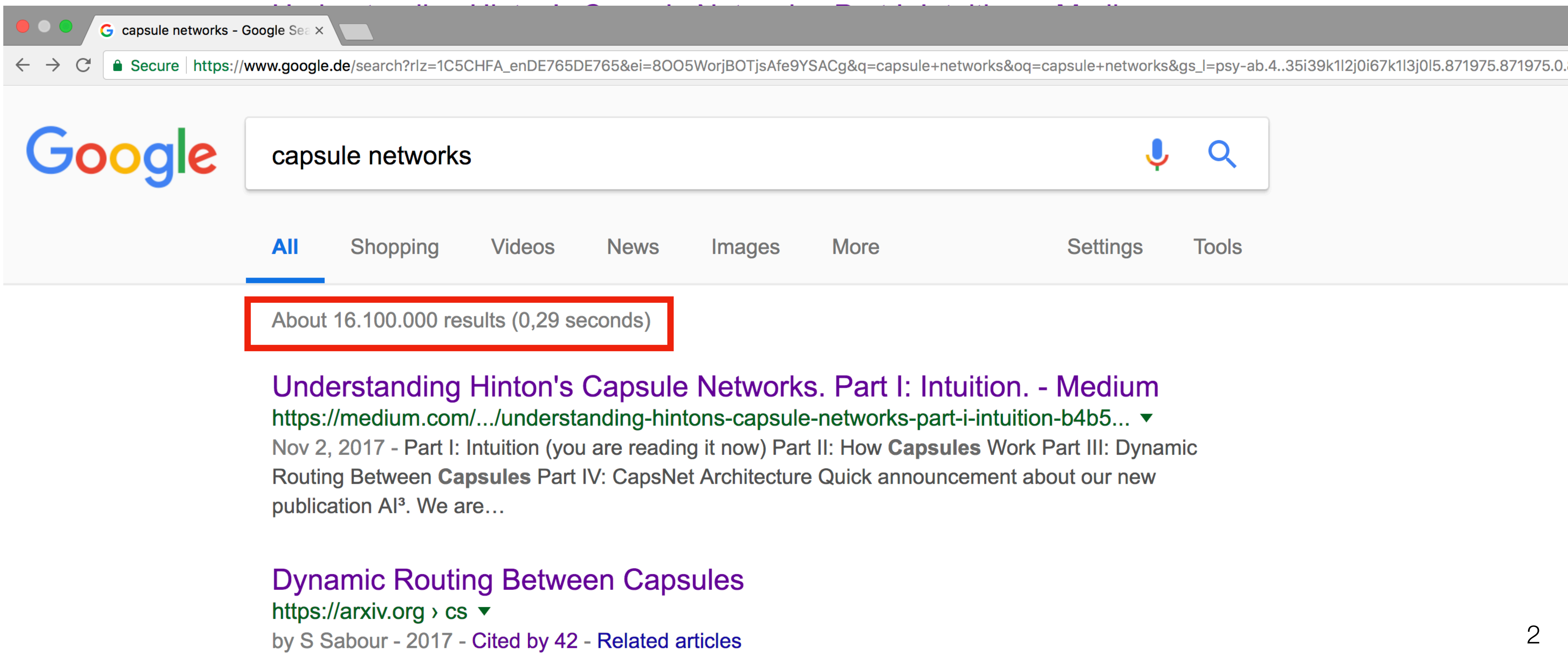
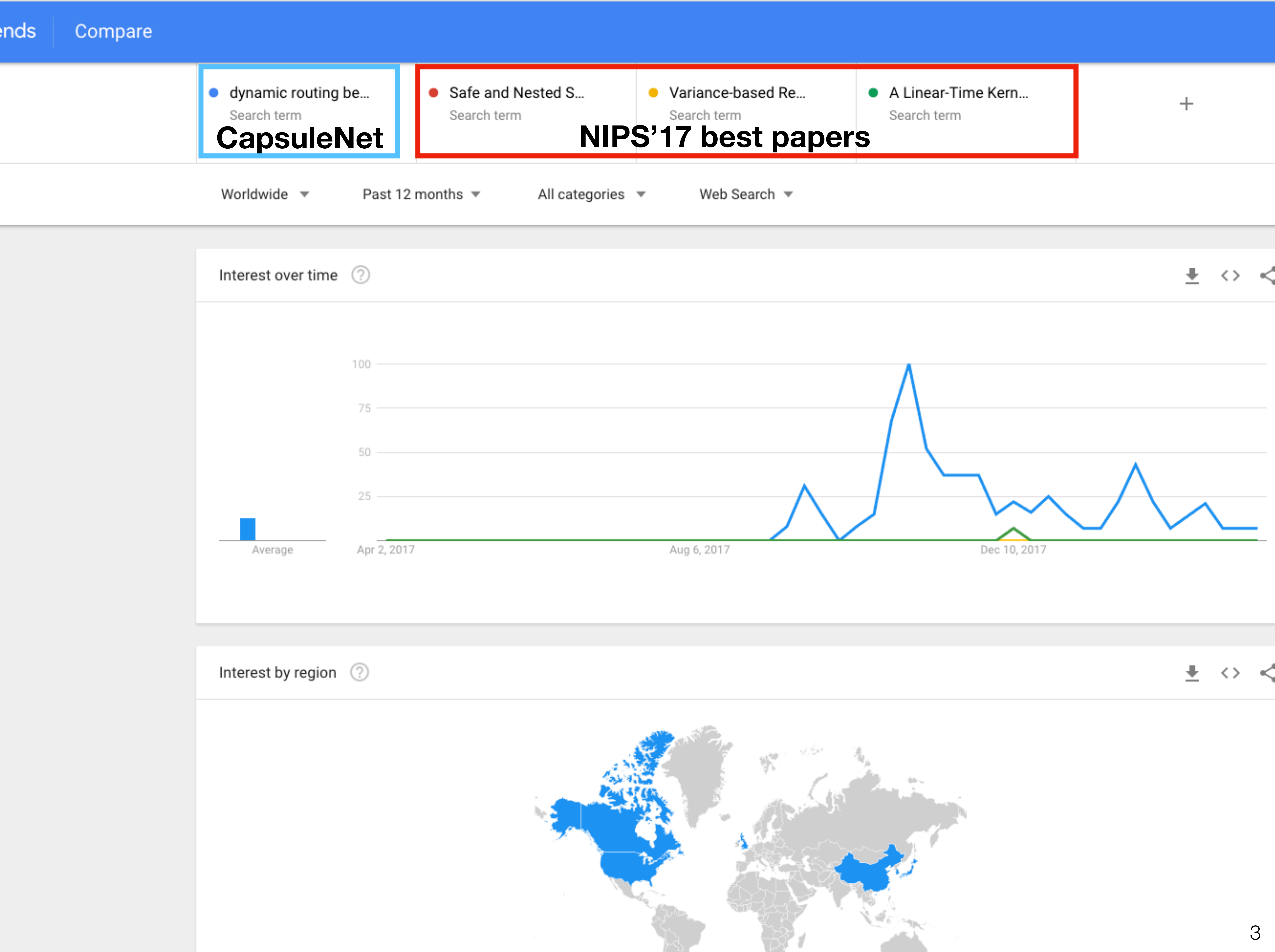
What is Capsule net?

Dr. Narisu Tao
Data Scientist@Advanced Analytics and Data Lab, E.ON
2018-03-29
Munich, Germany

Image source: https://www.artificial-intelligence.blog/news/capsule-networks-explained


capsule networks deep learnin
Search Google
Google
capsule networks deep learning
All
Images
News
Shopping
Videos
More
Settings
Tools
About 789.000 results (0,30 seconds)
Scholarly articles for capsule networks deep learning
A fast learning algorithm for deep belief nets - Hinton - Cited by 7823
… , algorithms, and applications for deep learning - Deng - Cited by 170
Three classes of deep learning architectures and their … - Deng - Cited by 54


capsule networks - Google Se
Secure
https://www.google.de/search?rlz=1C5CHFA_enDE765DE765&ei=8OO5WorjBOTjsAfe9YSACg&q=capsule+networks&oq=capsule+networks&gs_l=psy-ab.4..35i39k1l2j0i67k1l3j0l5.871975.871975.0.
Google
capsule networks
All
Shopping
Videos
News
Images
More
Settings
Tools
About 16.100.000 results (0,29 seconds)
Understanding Hinton's Capsule Networks. Part I: Intuition. - Medium
https://medium.com/.../understanding-hintons-capsule-networks-part-i-intuition-b4b5...
Nov 2, 2017 - Part I: Intuition (you are reading it now) Part II: How Capsules Work Part III: Dynamic
Routing Between Capsules Part IV: CapsNet Architecture Quick announcement about our new
publication AI³. We are…
Dynamic Routing Between Capsules
https://arxiv.org › cs
by S Sabour - 2017 - Cited by 42 - Related articles

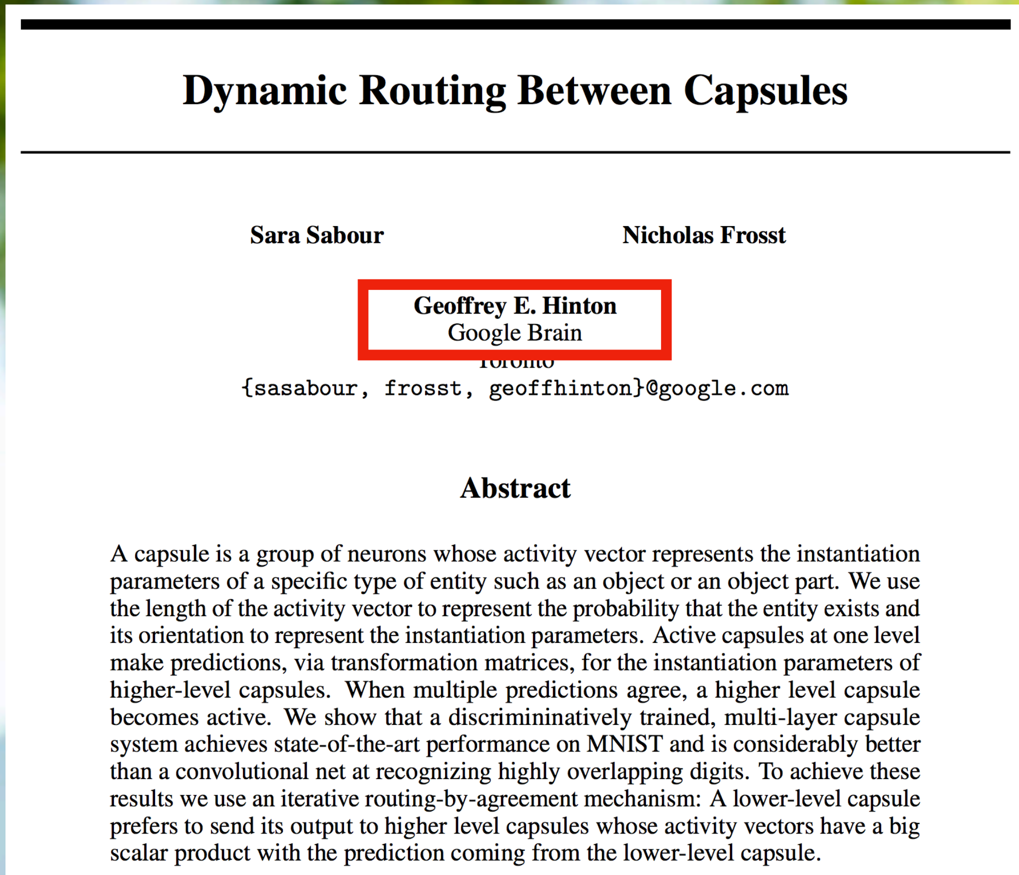


ends
Compare
dynamic routing be...
Search term
CapsuleNet
Safe and Nested S...
Search term
Variance-based Re...
Search term
A Linear-Time Kern...
Search term
NIPS’17 best papers
Worldwide
Past 12 months
All categories
Web Search
Interest over time
100
75
50
25
Average
Apr 2, 2017
Aug 6, 2017
Dec 10, 2017
Interest by region

Why people are so interested?

Do you know who is this man?

Image source: https://www.thestar.com/content/dam/thestar/news/world/2015/04/17/how-a-toronto-professors-research-revolutionized-artificial-intelligence/geoffrey-hinton-3.jpg.size.custom.crop.1086x723.jpg



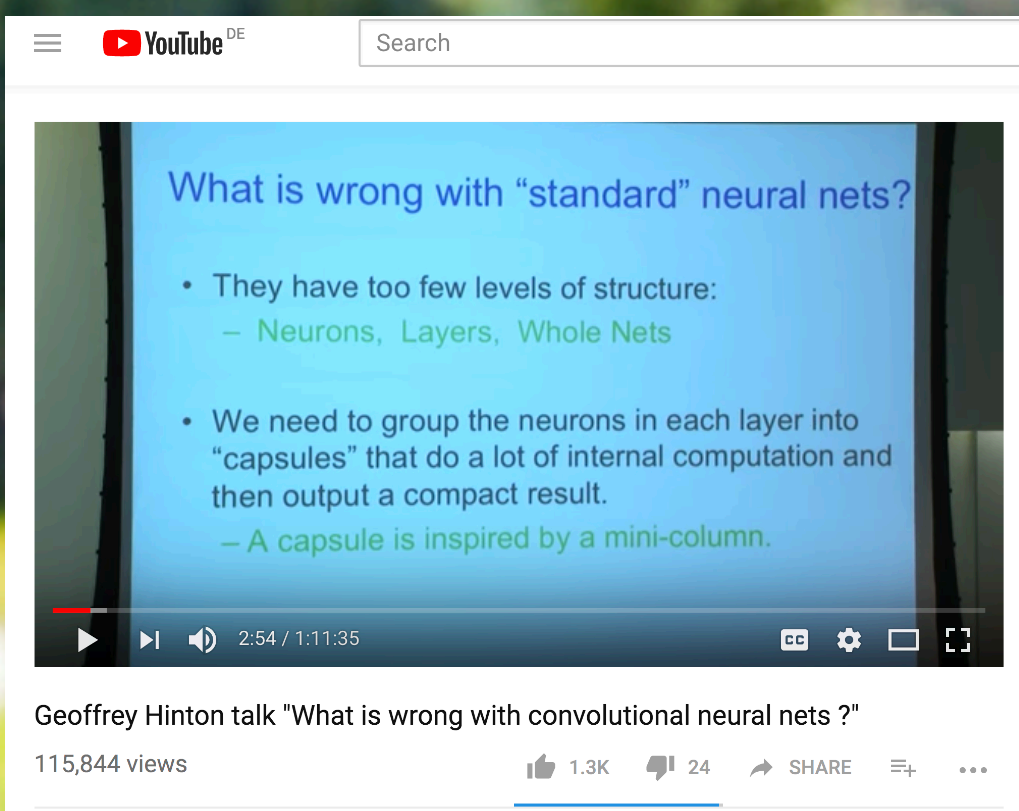
Dynamic Routing Between Capsules

Sara Sabour   Nicholas Frosst

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

Abstract

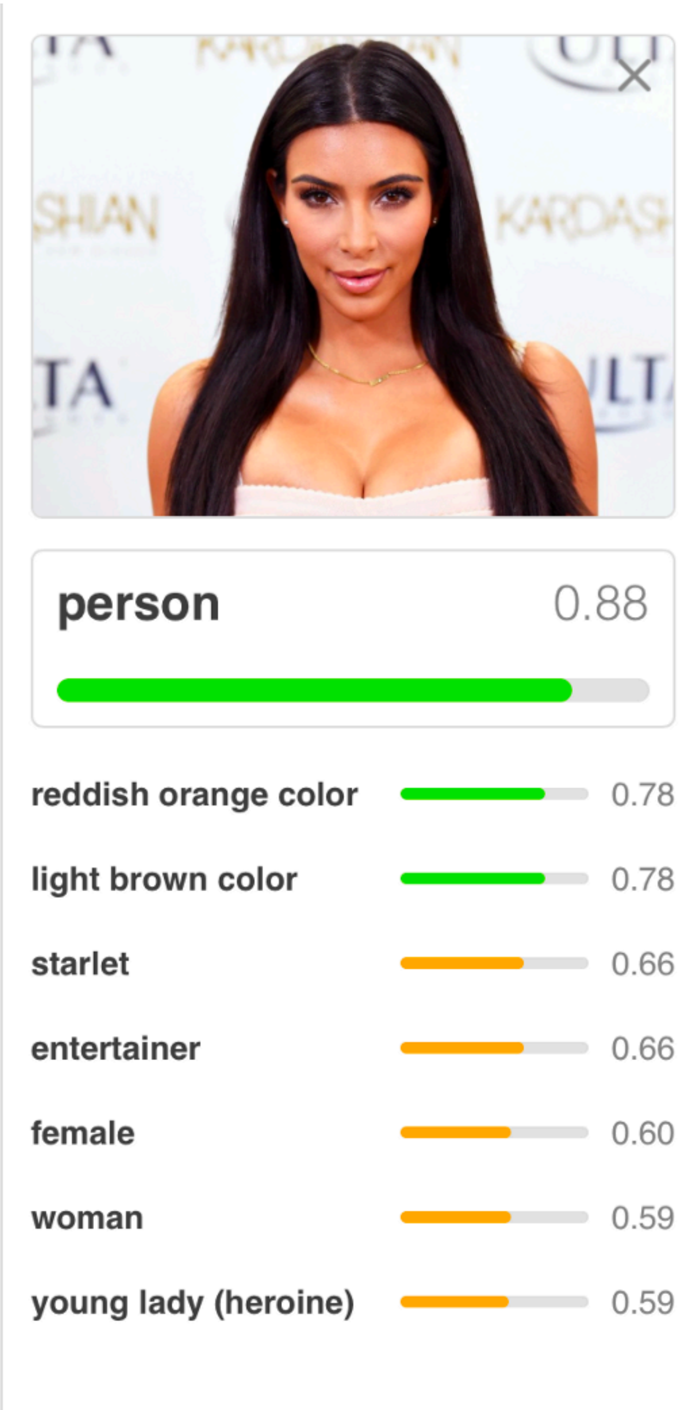
A capsule is a group of neurons whose activity vector represents the instantiation parameters of a specific type of entity such as an object or an object part. We use the length of the activity vector to represent the probability that the entity exists and its orientation to represent the instantiation parameters. Active capsules at one level make predictions, via transformation matrices, for the instantiation parameters of higher-level capsules. When multiple predictions agree, a higher level capsule becomes active. We show that a discrimininatively trained, multi-layer capsule system achieves state-of-the-art performance on MNIST and is considerably better than a convolutional net at recognizing highly overlapping digits. To achieve these results we use an iterative routing-by-agreement mechanism: A lower-level capsule prefers to send its output to higher level capsules whose activity vectors have a big scalar product with the prediction coming from the lower-level capsule.

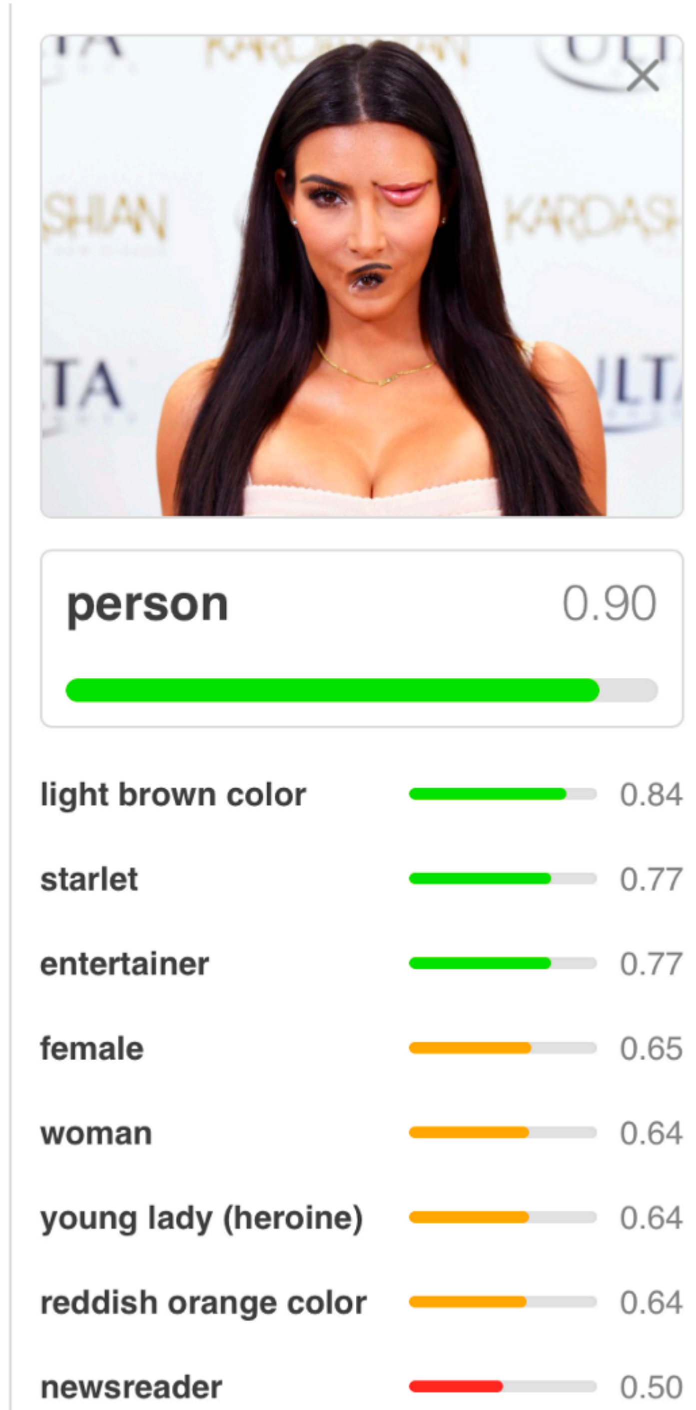


Prof. Dr. Geoffrey E. Hinton
God father of Deep Learning

Image source: https://www.thestar.com/content/dam/thestar/news/world/2015/04/17/how-a-toronto-professors-research-revolutionized-artificial-intelligence/geoffrey-hinton-3.jpg.size.custom.crop.1086x723.jpg

The limitation of CNNs





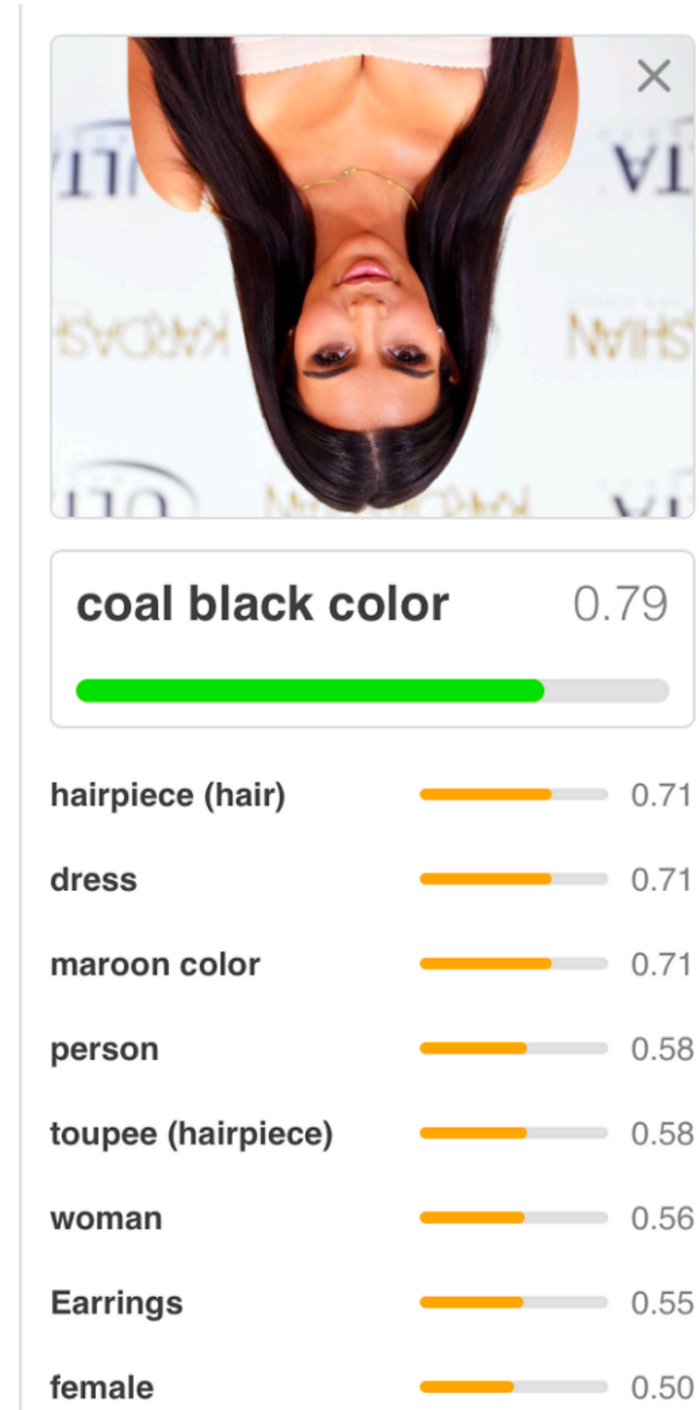


Image source:
https://cdn-images-1.medium.com/max/1540/1*h-Be4Uj-a7N6_Qx-K03KkQ.png
https://cdn-images-1.medium.com/max/1540/0*6Uepc8ierhzwBIVN.

"For thirty years, the state-of-the-art in speech recognition used hidden Markov models …. These models … had a **representational limitation** that was ultimately fatal: The one-of-n representations they use are **exponentially inefficient** compared with, say, a recurrent neural network that uses **distributed representations**. To double the amount of information that an HMM can remember …, we need to square the number of hidden nodes. For a recurrent net we only need to double the number of hidden neurons.

Now that convolutional neural networks have become the dominant approach to object recognition, it makes sense to ask whether there are any exponential inefficiencies that may lead to their demise.

A good candidate is the difficulty that convolutional nets have in generalizing to novel viewpoints. The ability to deal with translation is built in, but for the other dimensions of an affine transformation we have to chose between replicating feature detectors on a grid that grows exponentially with the number of dimensions, or increasing the size of the labelled training set in a similarly exponential way. **Capsules (Hinton et al. [2011]) avoid these exponential inefficiencies …**

Capsules make a very strong representational assumption: …, which …, eliminates the binding problem (Hinton [1981a]) and allows a capsule to use a distributed representation …. This distributed representation is **exponentially more efficient** than encoding the instantiation parameters by activating a point on a high-dimensional grid and with the right distributed representation, capsules can then take full advantage of the fact that spatial relationships can be modelled by matrix multiplies.

Research on capsules is now at a similar stage to research on recurrent neural networks for speech recognition at the beginning of this century. …"

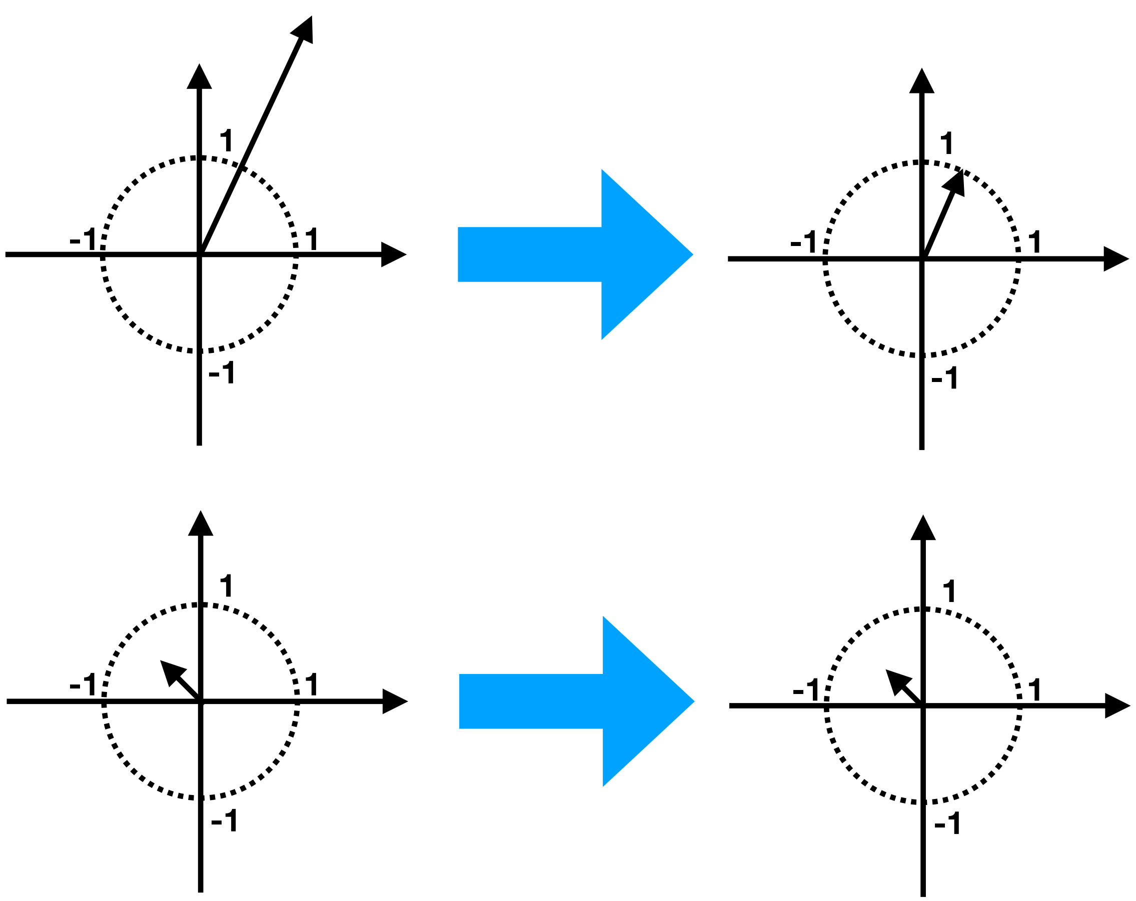
— *Discussion and previous wor*k section
from the paper *Dynamic Routing Between Capsules*

How it actually works?

Squash function
versus
sigmoid activation function

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

$$f(x) = \frac{1}{1 + e^{-x}}$$



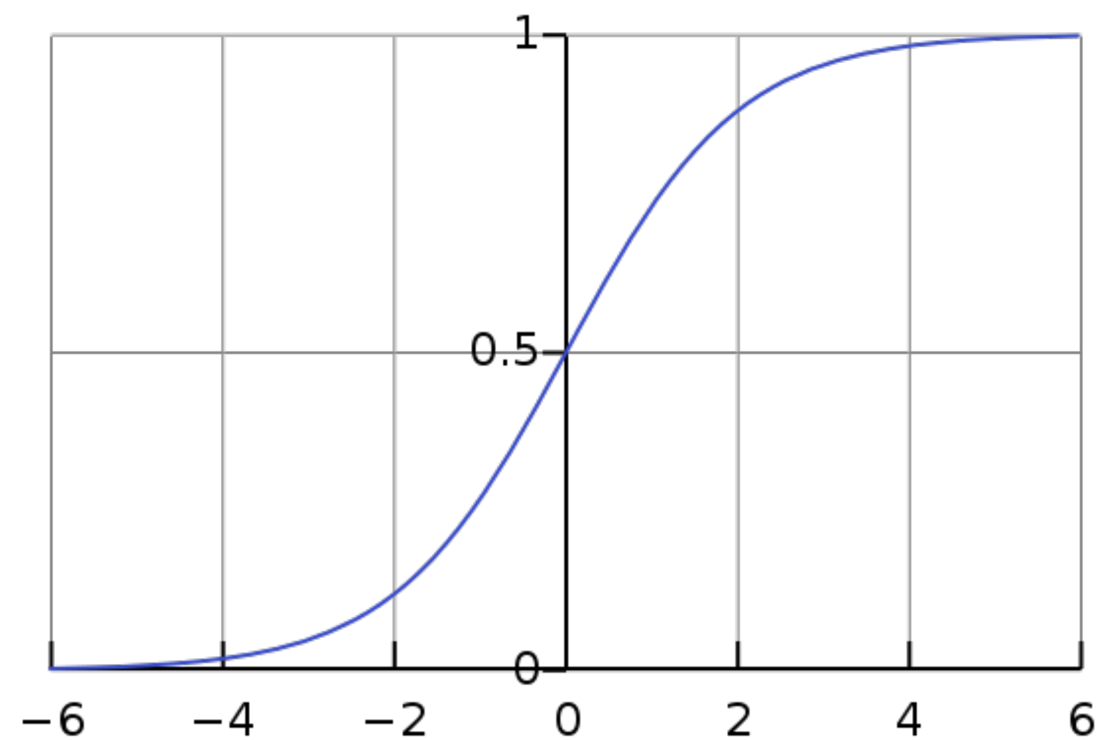
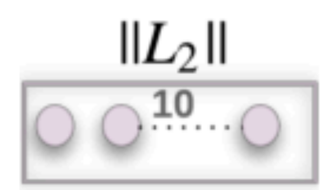


Image source:
https://upload.wikimedia.org/wikipedia/commons/thumb/8/88/Logistic-curve.svg/600px-Logistic-curve.svg.png

Inference

Input



Output

Inference

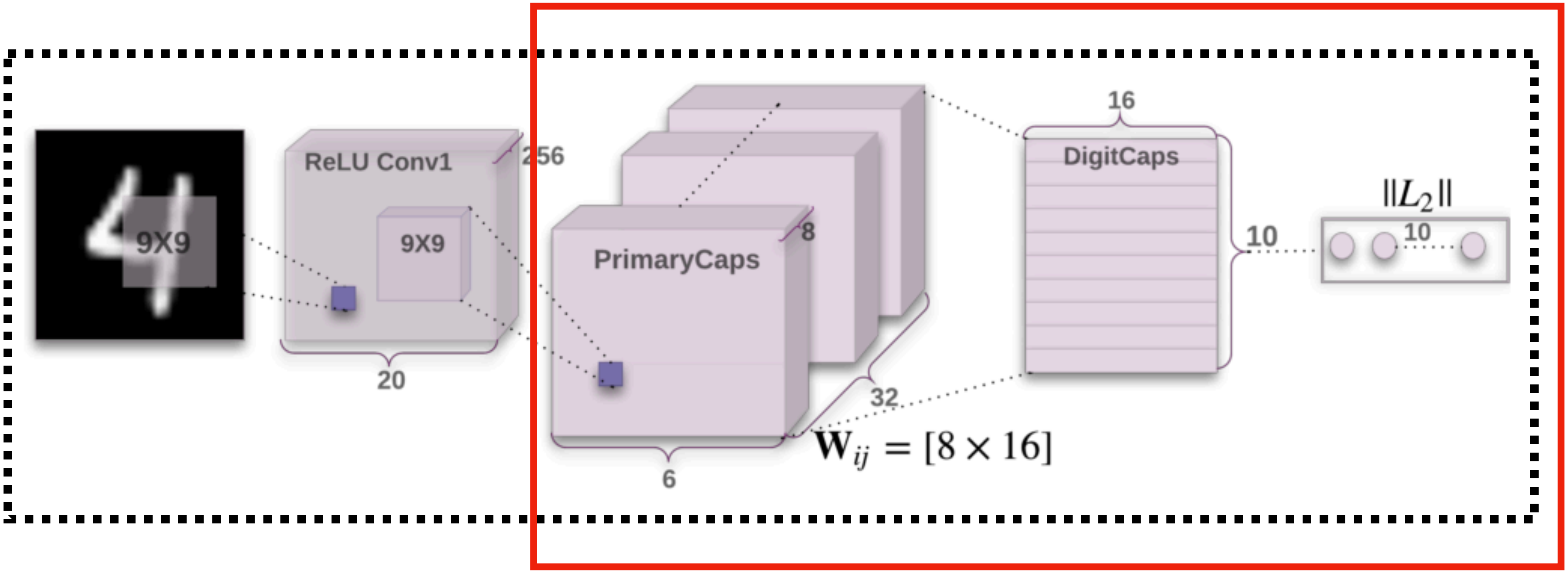


Image source: https://arxiv.org/pdf/1710.09829.pdf

Inference

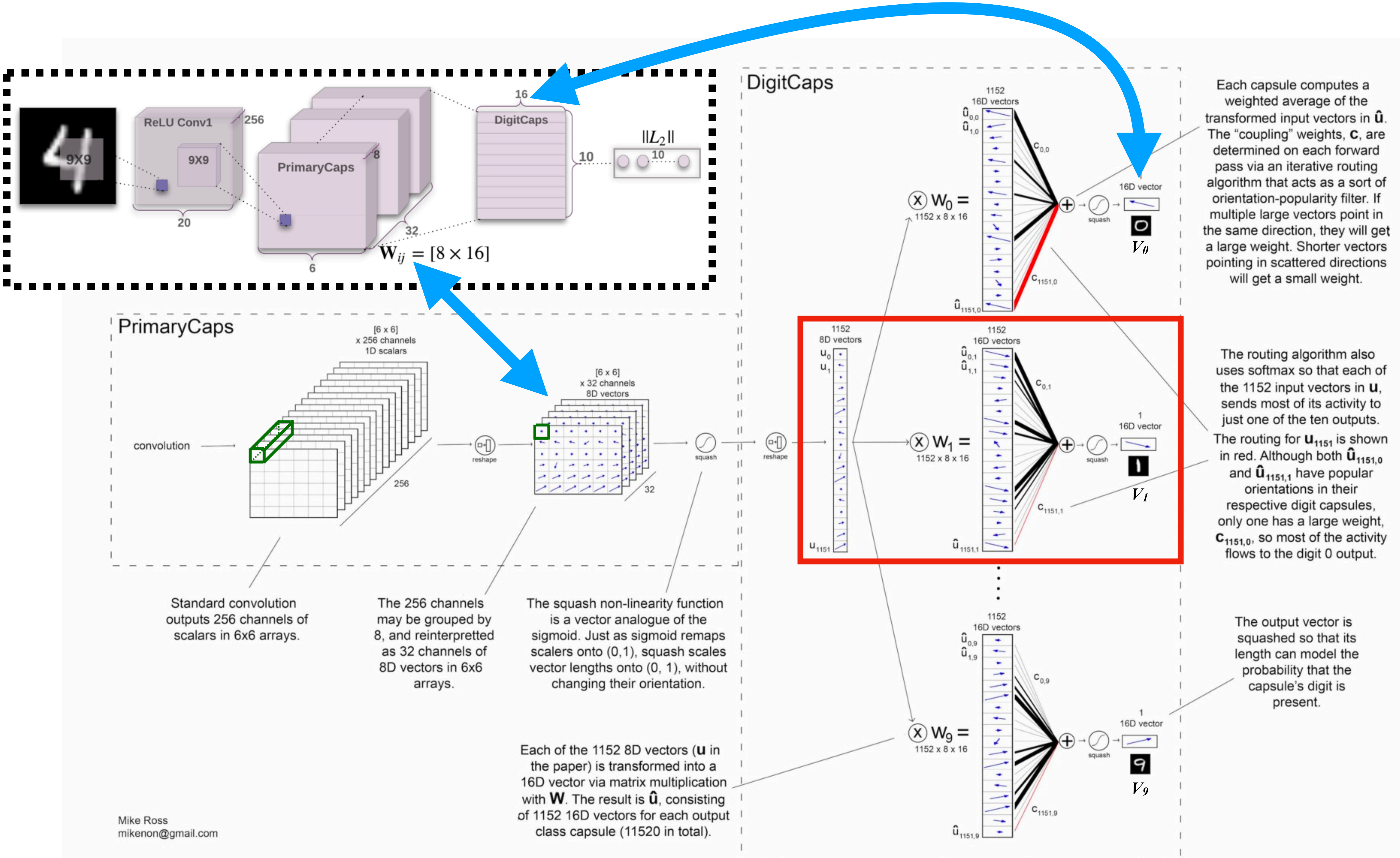
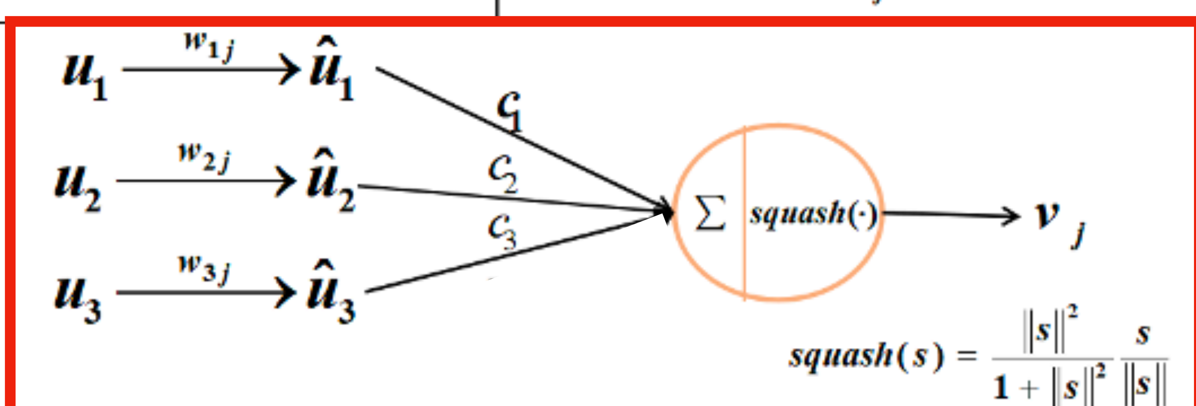
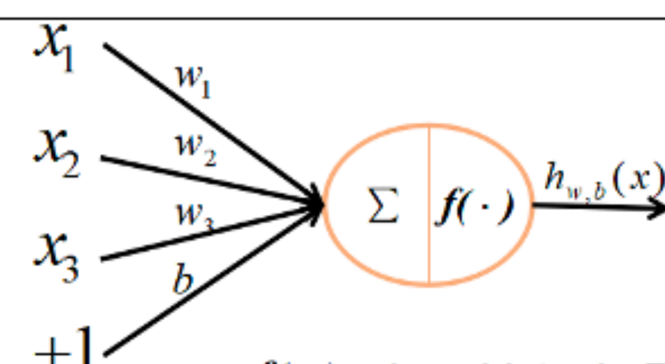


Image source:
https://cdn-images-1.medium.com/max/1800/1*UPaxEd1A3N5ceckB85RIRg.jpeg
https://arxiv.org/pdf/1710.09829.pdf

Inference

		capsule VS.	traditional neuron
Input from low-level neurons/capsules		vector(u_i)	scalar(x_i)
Operations	Linear/Affine Transformation	$\hat{u}_{j\|i} = W_{ij} u_i$ (Eq. 2)	$a_{j\|i} = w_{ij} x_i + b_j$
	Weighting Summation	$s_j = \sum_i c_{ij} \hat{u}_{j\|i}$ (Eq. 2)	$z_j = \sum_{i=1}^{3} 1 \cdot a_{j\|i}$
	Non-linearity activation	$v_j = squash(s_j)$ (Eq. 1)	$h_{w,b}(x) = f(z_j)$
output		vector(v_j)	scalar(h)





$f(\cdot)$: sigmoid, tanh, ReLU, etc.

Capsule = New Version Neuron!
vector in, vector out VS. scalar in, scalar out

Image source: https://raw.githubusercontent.com/naturomics/CapsNet-Tensorflow/master/imgs/capsuleVSneuron.png

Dynamic Routing Between Capsules

Dynamic Routing

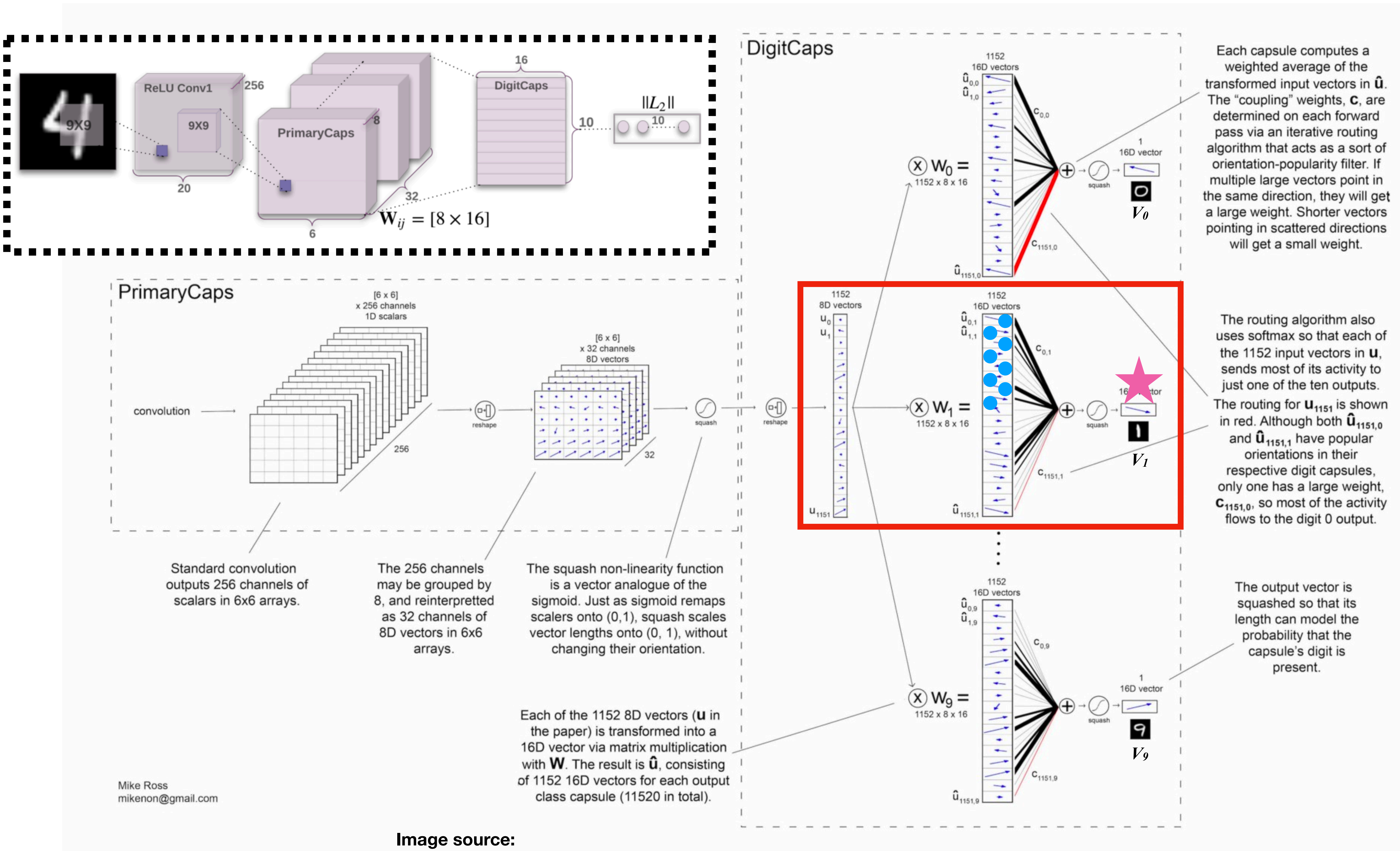
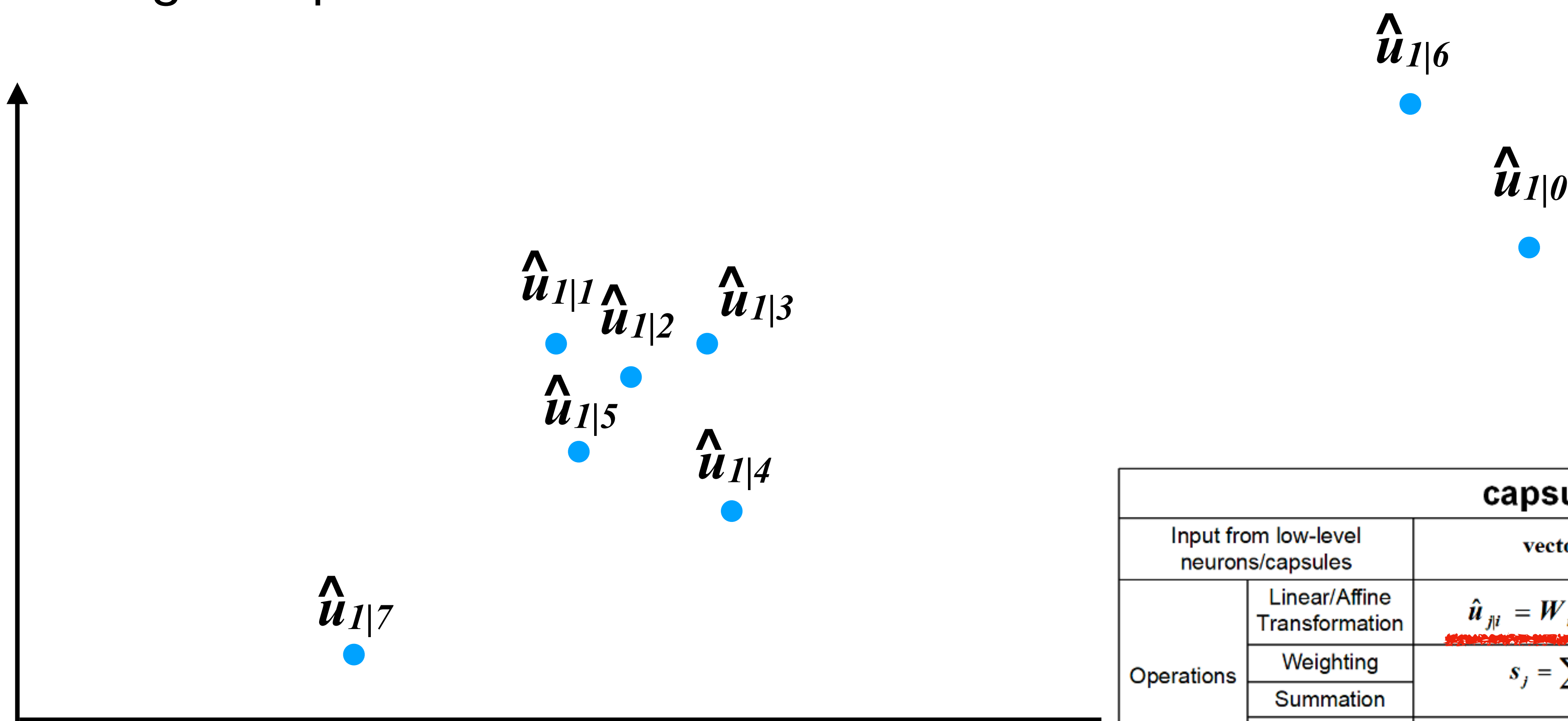


Image source:
https://cdn-images-1.medium.com/max/1800/1*UPaxEd1A3N5ceckB85RIRg.jpeg
https://arxiv.org/pdf/1710.09829.pdf

Dynamic Routing

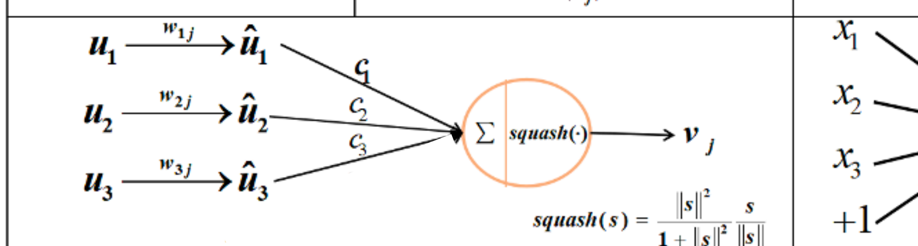
Assuming that the digits capsule is 2-D vector rather than 16-D vector.



Procedure 1 Routing algorithm.

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

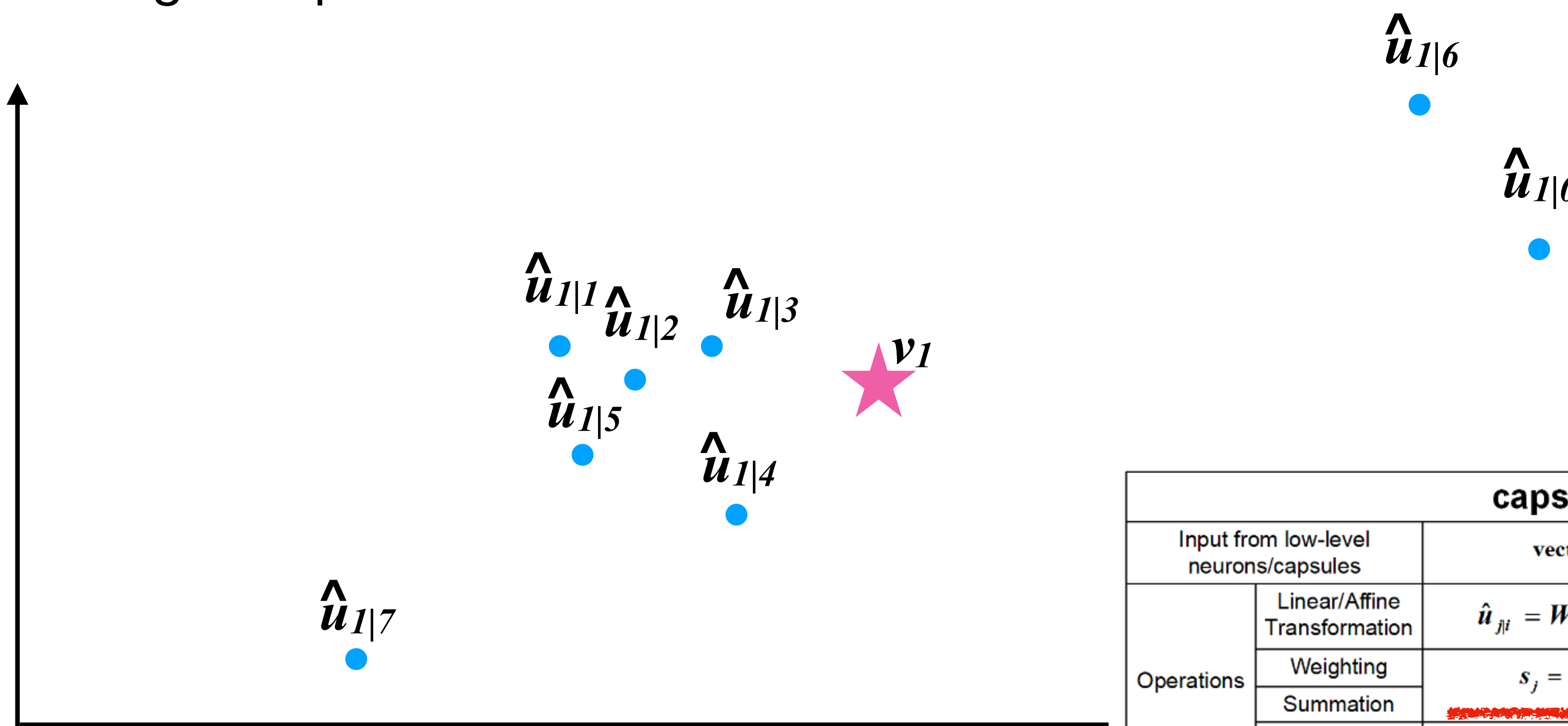
		capsule	VS.
Input from low-level neurons/capsules		vector(u_i)	
Operations	Linear/Affine Transformation	$\hat{u}_{j|i} = W_{ij} u_i$ (Eq. 2)	
	Weighting Summation	$s_j = \sum_i c_{ij}\hat{u}_{j|i}$ (Eq. 2)	
	Non-linearity activation	$v_j = squash(s_j)$ (Eq. 1)	
output		vector(v_j)	



Capsule = New Version Neuron
vector in, vector out VS. scalar in, sc

Dynamic Routing

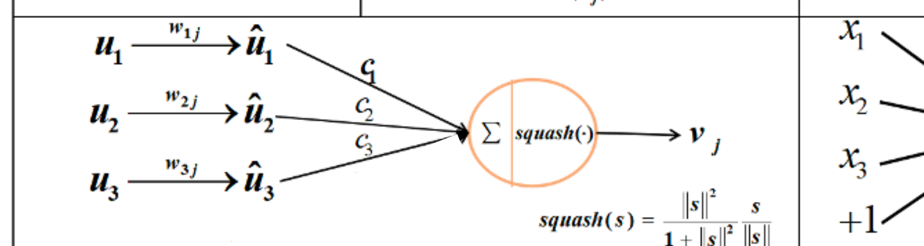
Assuming that the digits capsule is 2-D vector rather than 16-D vector.



Procedure 1 Routing algorithm.

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

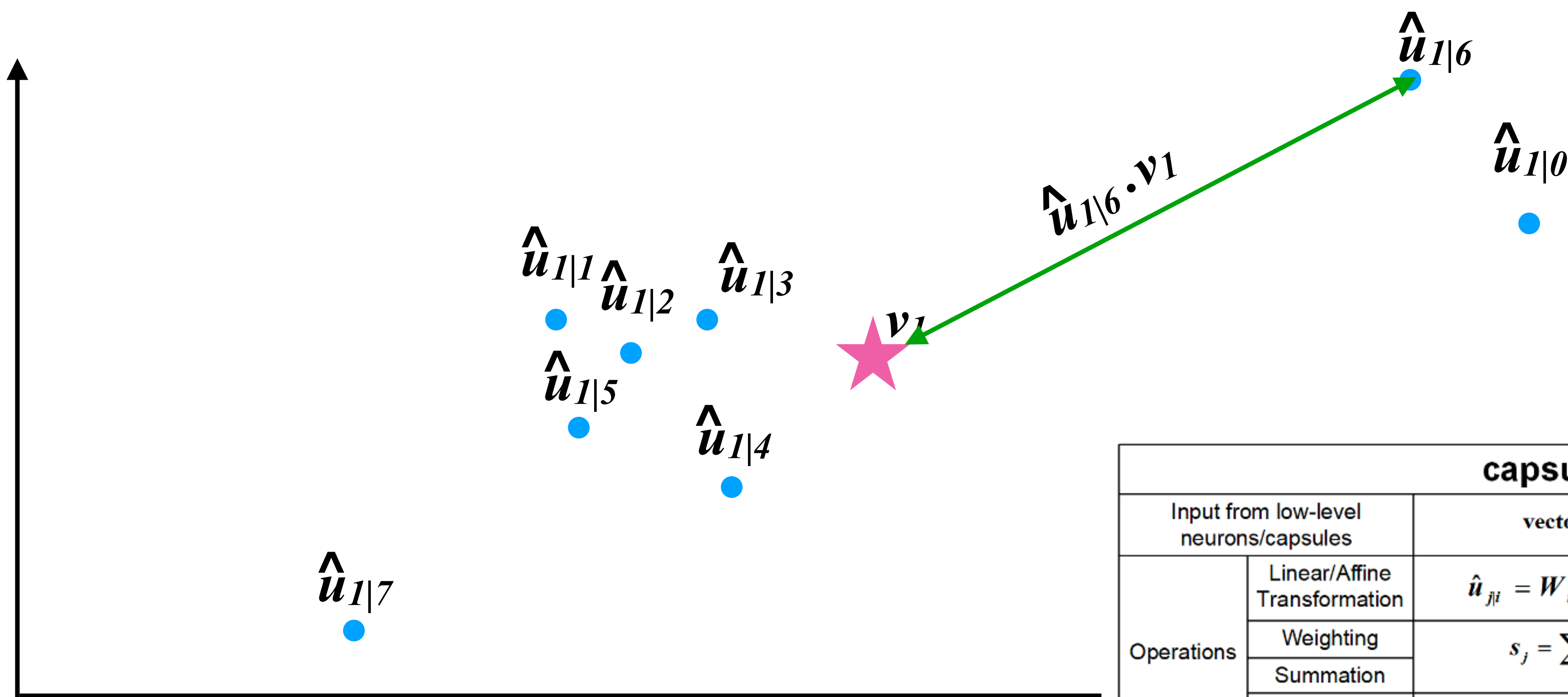
		capsule	VS.
Input from low-level neurons/capsules		vector(u_i)	
Operations	Linear/Affine Transformation	$\hat{u}_{j\|i} = W_{ij} u_i$ (Eq. 2)	
	Weighting	$s_j = \sum_i c_{ij}\hat{u}_{j\|i}$ (Eq. 2)	
	Summation		
	Non-linearity activation	$v_j = squash(s_j)$ (Eq. 1)	
output		vector(v_j)	



Capsule = New Version Neuron
vector in, vector out VS. scalar in, s

Dynamic Routing

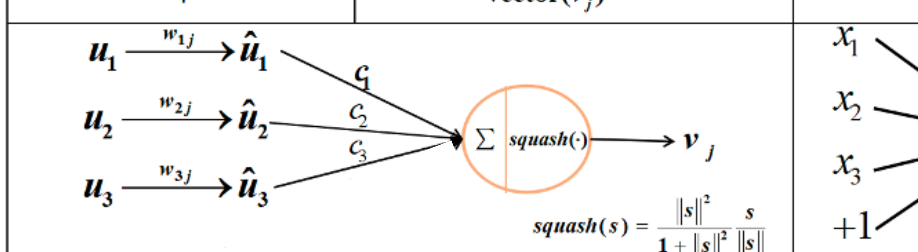
Assuming that the digits capsule is 2-D vector rather than 16-D vector.



Procedure 1 Routing algorithm.

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

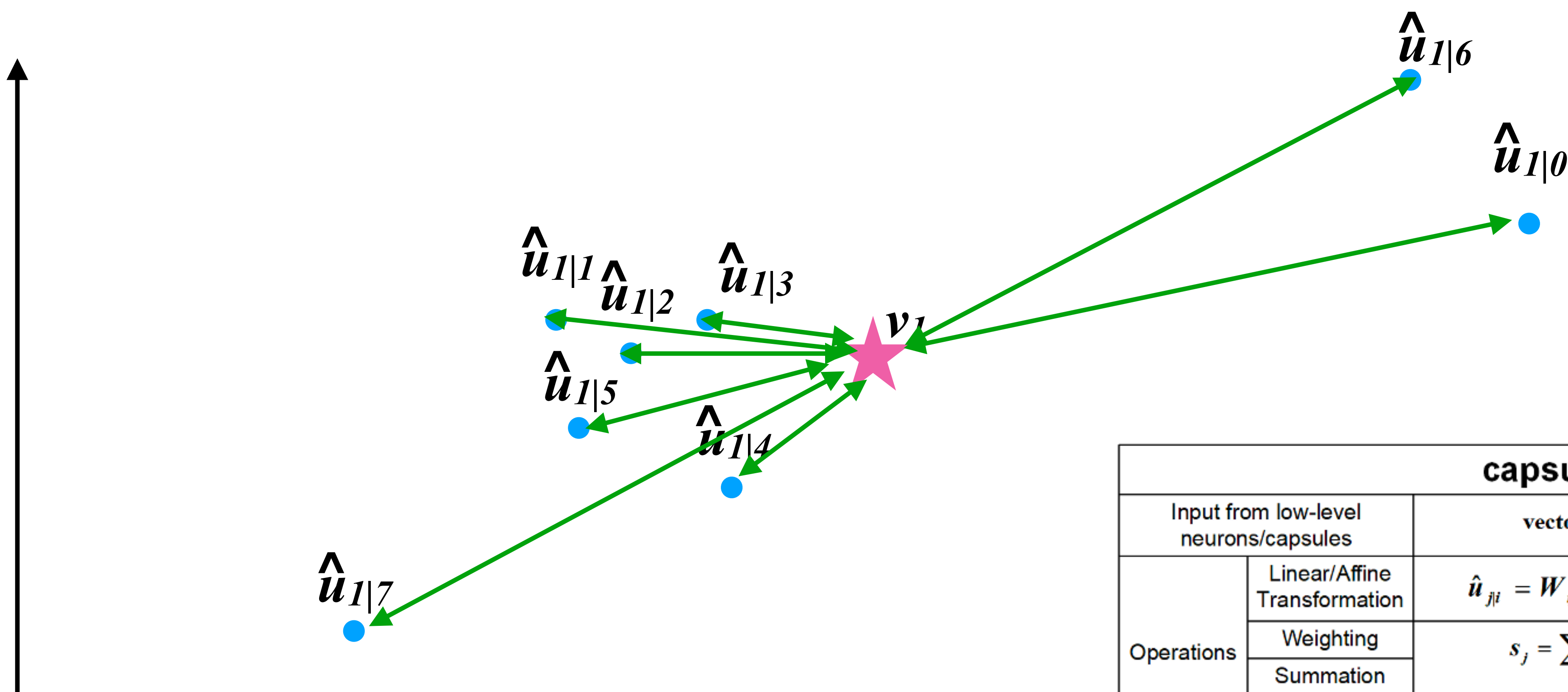
		capsule	VS.
Input from low-level neurons/capsules		vector(u_i)	
Operations	Linear/Affine Transformation	$\hat{u}_{j|i} = W_{ij} u_i$ (Eq. 2)	
	Weighting	$s_j = \sum_i c_{ij} \hat{u}_{j|i}$ (Eq. 2)	
	Summation		
	Non-linearity activation	$v_j = squash(s_j)$ (Eq. 1)	
output		vector(v_j)	



Capsule = New Version Neuron
vector in, vector out VS. scalar in, s

Dynamic Routing

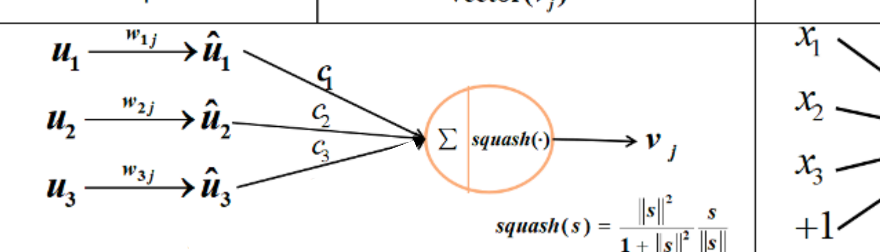
Assuming that the digits capsule is 2-D vector rather than 16-D vector.



Procedure 1 Routing algorithm.

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

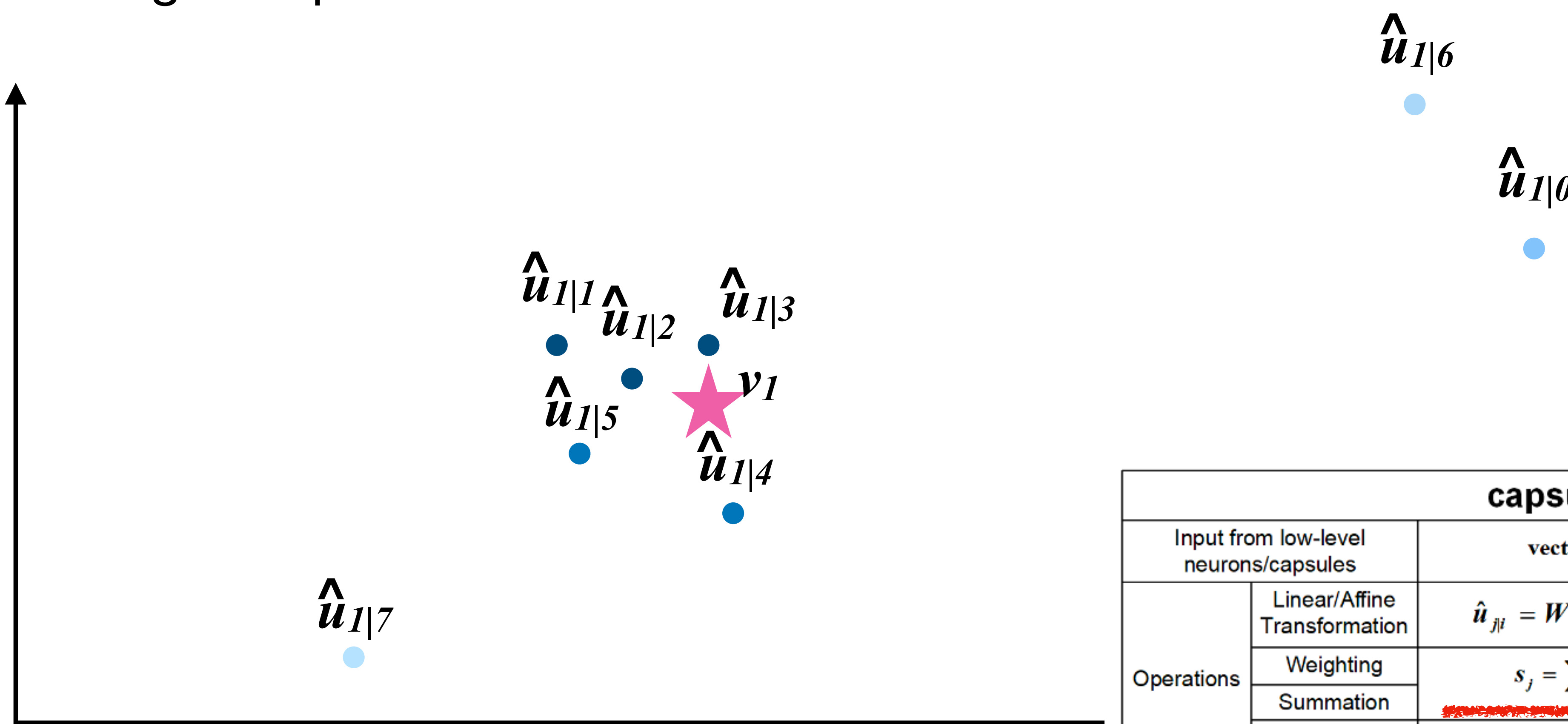
		capsule		VS.
Input from low-level neurons/capsules		vector(u_i)		
Operations	Linear/Affine Transformation	$\hat{u}_{j\|i} = W_{ij}u_i$	(Eq. 2)	
	Weighting	$s_j = \sum_i c_{ij}\hat{u}_{j\|i}$	(Eq. 2)	
	Summation			
	Non-linearity activation	$v_j = squash(s_j)$	(Eq. 1)	
output		vector(v_j)		



Capsule = New Version Neuron
vector in, vector out VS. scalar in, sc

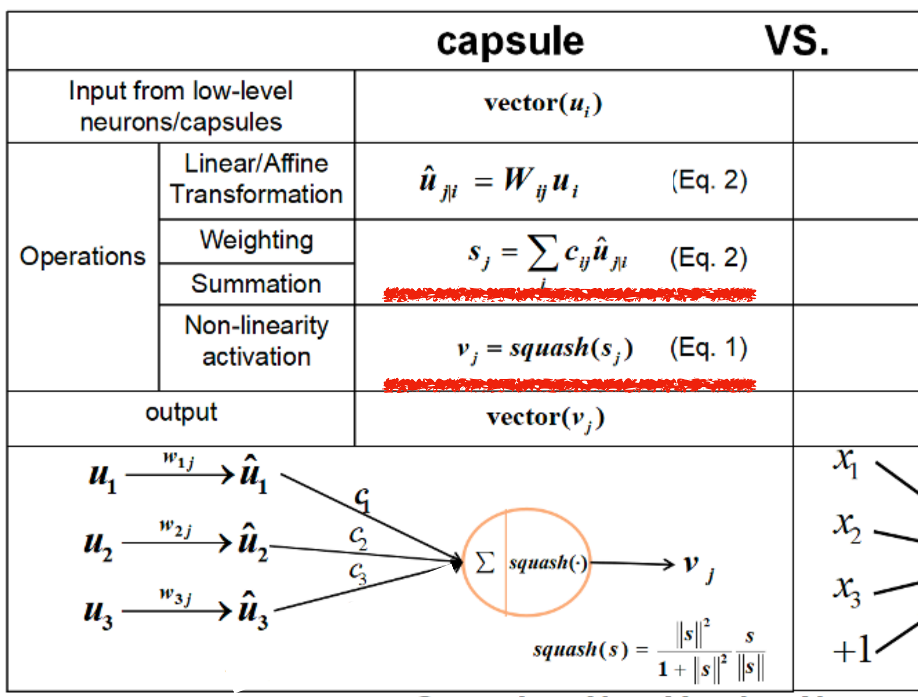
Dynamic Routing

Assuming that the digits capsule is 2-D vector rather than 16-D vector.



Procedure 1 Routing algorithm.

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

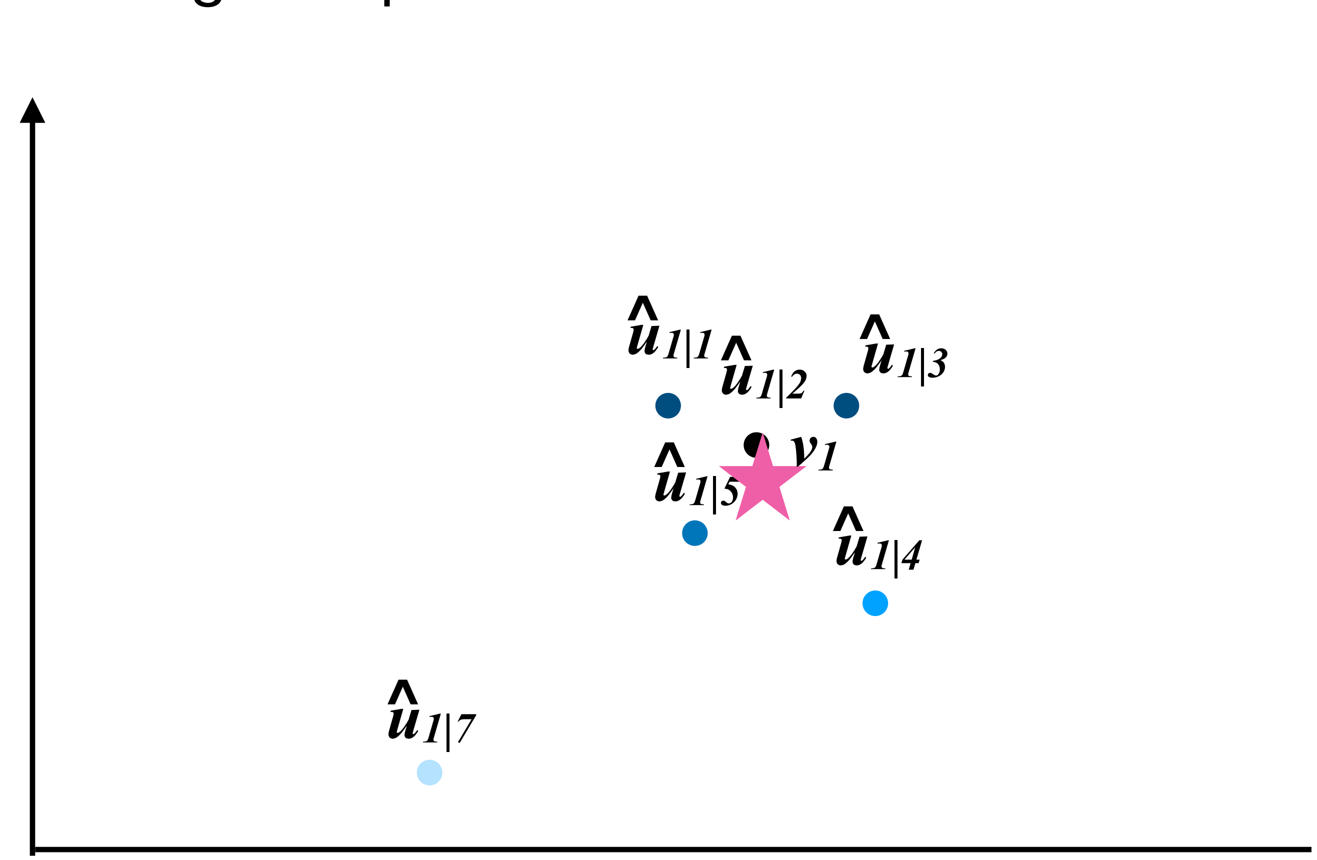


		capsule	VS.
Input from low-level neurons/capsules		vector(u_i)	
Operations	Linear/Affine Transformation	$\hat{u}_{j\|i} = W_{ij} u_i$ (Eq. 2)	
	Weighting	$s_j = \sum_i c_{ij} \hat{u}_{j\|i}$ (Eq. 2)	
	Summation		
	Non-linearity activation	$v_j = squash(s_j)$ (Eq. 1)	
output		vector(v_j)	

Capsule = New Version Neuron
vector in, vector out VS. scalar in, sc

Dynamic Routing

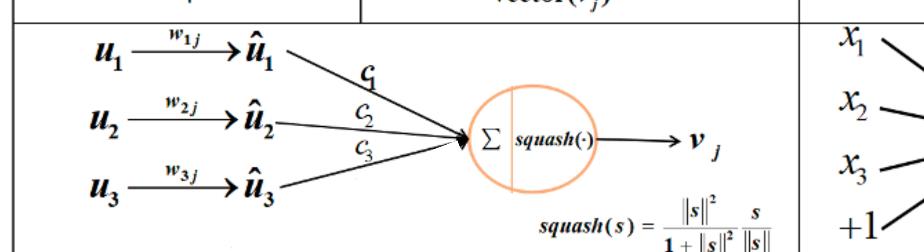
Assuming that the digits capsule is 2-D vector rather than 16-D vector.



Procedure 1 Routing algorithm.

```
procedure ROUTING(û_{j|i}, r, l)
    for all capsule i in layer l and capsule j in layer (l + 1): b_ij ← 0.
    for r iterations do
        for all capsule i in layer l: c_i ← softmax(b_i)            ▷ softmax computes Eq.
        for all capsule j in layer (l + 1): s_j ← Σ_i c_ij û_{j|i}
        for all capsule j in layer (l + 1): v_j ← squash(s_j)       ▷ squash computes Eq.
        for all capsule i in layer l and capsule j in layer (l + 1): b_ij ← b_ij + û_{j|i}.v_j
    return v_j
```

		capsule	VS.
Input from low-level neurons/capsules		vector(u_i)	
Operations	Linear/Affine Transformation	$\hat{u}_{j\|i} = W_{ij} u_i$ (Eq. 2)	
	Weighting	$s_j = \sum_i c_{ij} \hat{u}_{j\|i}$ (Eq. 2)	
	Summation		
	Non-linearity activation	$v_j = squash(s_j)$ (Eq. 1)	
output		vector(v_j)	



Capsule = New Version Neuron
vector in, vector out VS. scalar in, sc

Learning

Learning: loss function

m^{+} = 0.9
m^{-} = 0.1
T_k = 1 if digit of class k is present
$\lambda = 0.5$

Margin loss function

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

Translation to English:

“For digit capsule k,

If an input image is digit k, then $||V_k||^2$ should be no less than 0.9.

If not, then $||V_k||^2$ should be no more than 0.1.”

source: https://youtu.be/pPN8d0E3900

Learning: loss function

Reconstruction loss function

$$L_{re} = ||X - \hat{X}||$$

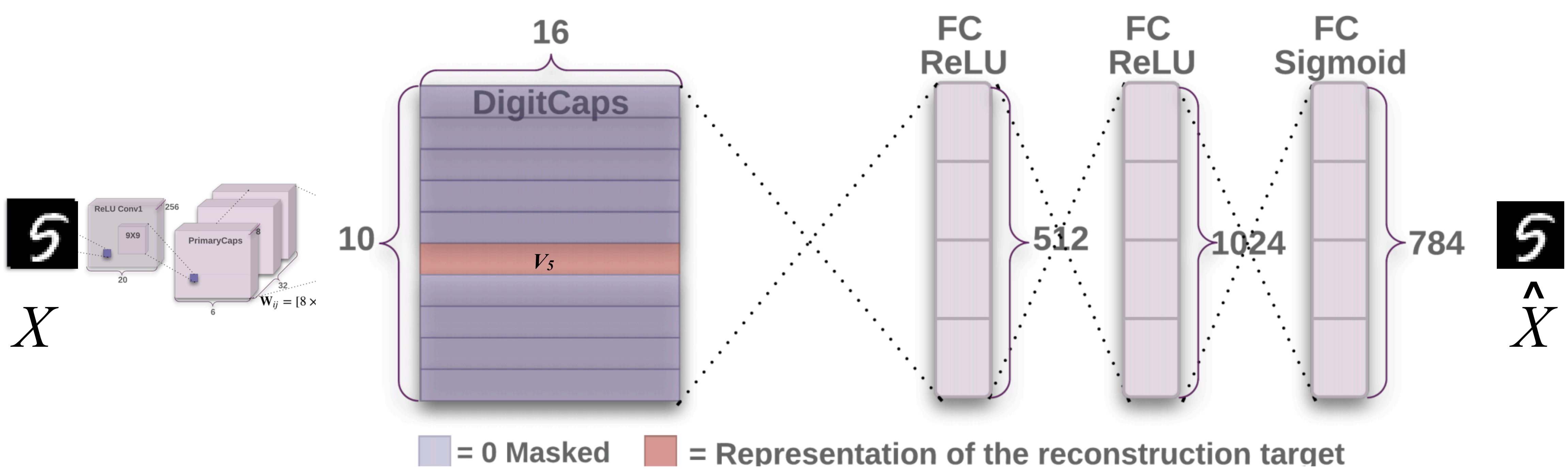


Image source: https://arxiv.org/pdf/1710.09829.pdf

Learning: loss function

Loss function

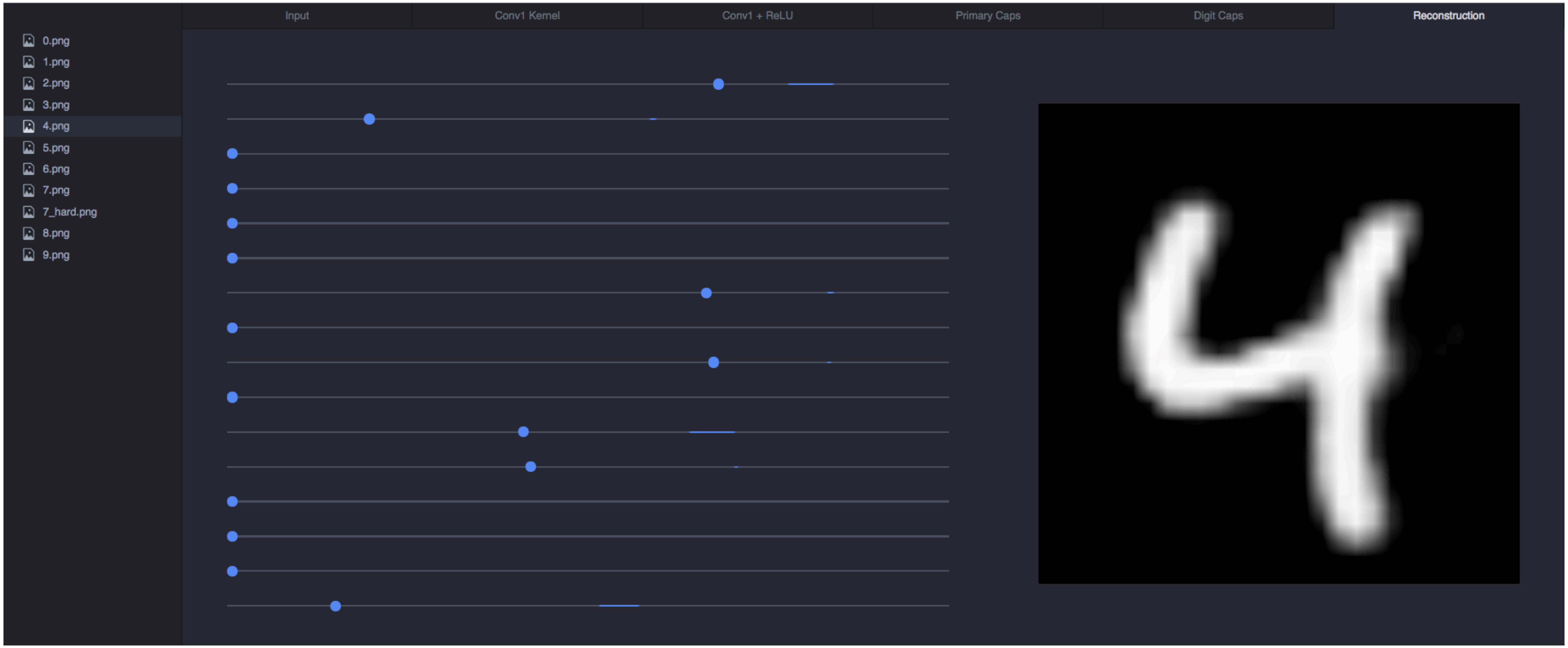
$$L = \sum L_k + 0.0005 \cdot L_{re}$$

Updated weights:

All the weights expect C_{ij} ,*because* C_{ij} *is not weights of the model.*

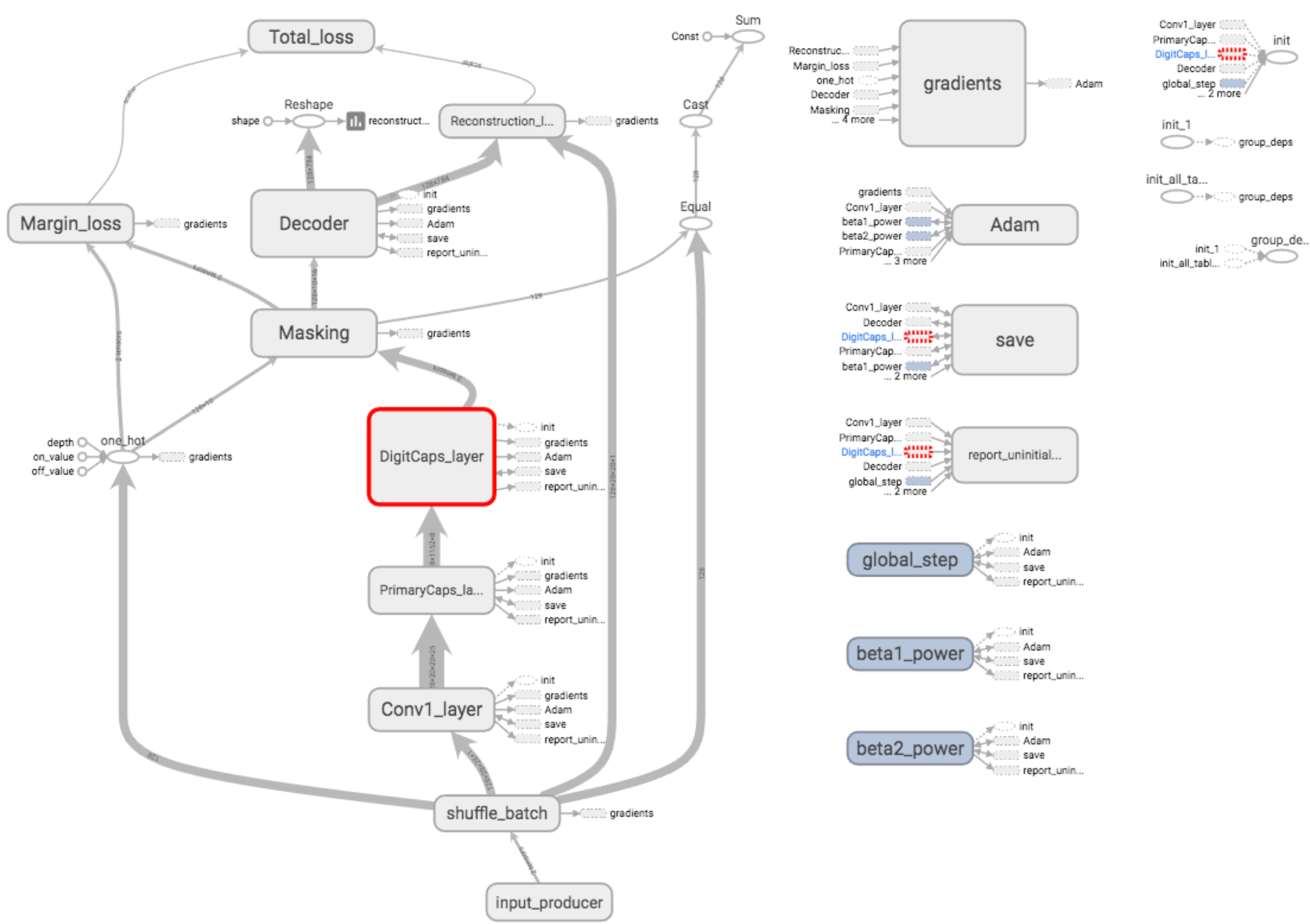
Demo on the reconstruction from digit capsules

Credit goes to
https://github.com/bourdakos1/CapsNet-Visualization



Computation graph of capsule net in TensorBoard

Credit goes to https://github.com/naturomics/CapsNet-Tensorflow



More Resource

aisummary / **awesome-capsule-networks**

Watch 10 | Star 134 | Fork 36

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights

A curated list of awesome resources related to capsule networks

capsule-networks capsule-network computer-vision deep-learning neural-networks awesome-list

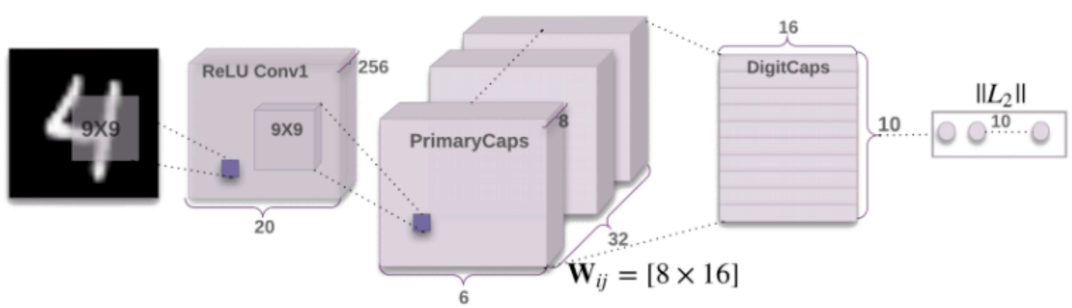
39 commits | 1 branch | 0 releases | 3 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

aisummary Added the missing link to the Keras subsection in the table of contents		Latest commit de8594a 11 days ago
README.md	Added the missing link to the Keras subsection in the table of contents	11 days ago
diagram.png	Added figure 1 from Sabour et al. (2017)	2 months ago

README.md

Awesome Capsule Networks awesome



A curated list of awesome resources related to capsule networks maintained by AI Summary.

Contributing

Please pull a request if you are aware of additional resources.

Your feedback and contributions are always welcome!

Sharing

- Share on Twitter
- Share on Facebook
- Share on Google Plus
- Share on LinkedIn

Table of Contents

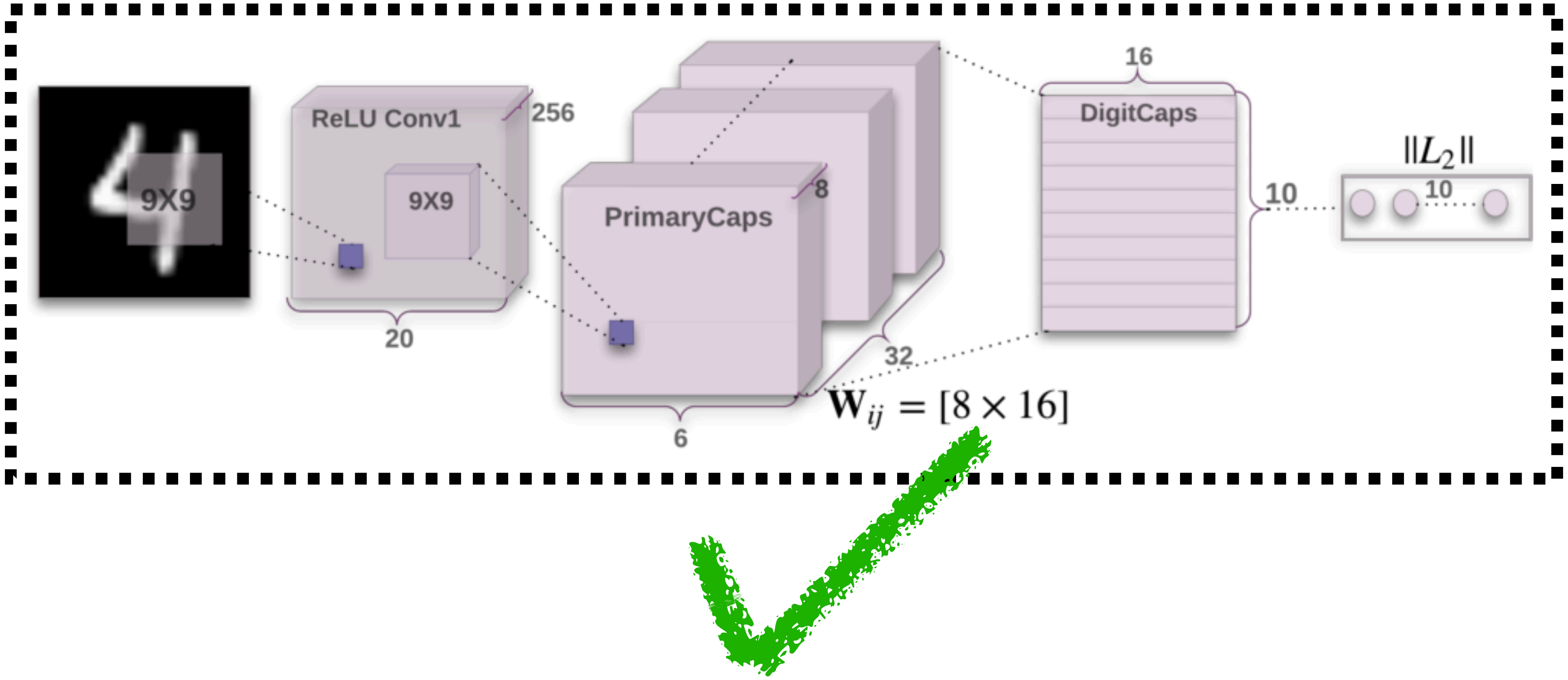
- Papers
  - Papers by Geoffrey Hinton and colleagues
  - Under review
  - Preprints
- Videos
- Blogs
- Dynamic routing implementations
  - Official implementation
  - Chainer
  - CNTK
  - Keras
  - Matlab
  - MXNet
  - PyTorch
  - R
  - TensorFlow
  - Torch
- EM routing implementations
  - PyTorch
  - TensorFlow
- Other resources
- License

https://github.com/aisummary/awesome-capsule-networks

Capsule net should say it's not a person

Image source: https://cdn-images-1.medium.com/max/1980/1*Nr1dKBMQElEoH4AhfVbitw.png

Thank you!



9X9
ReLU Conv1
256
9X9
20
PrimaryCaps
8
32
6
16
DigitCaps
10
$||L_2||$
10
$\mathbf{W}_{ij} = [8 \times 16]$