Capsule Networks

By Modar Alfadly

(October 2nd, 2018)

Papers

- Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton.
  "Dynamic routing between capsules."
  Advances in Neural Information Processing Systems (2017)

- Hinton, Geoffrey E., Sara Sabour, and Nicholas Frosst.
  "Matrix capsules with EM routing."
  International Conference on Learning Representations (2018).

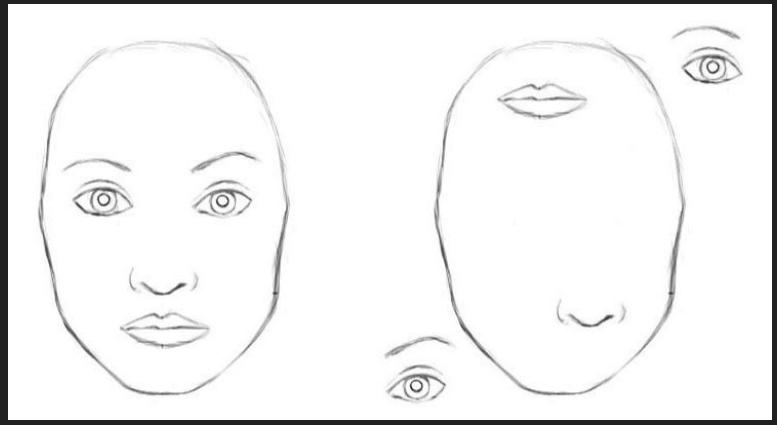
Motivation

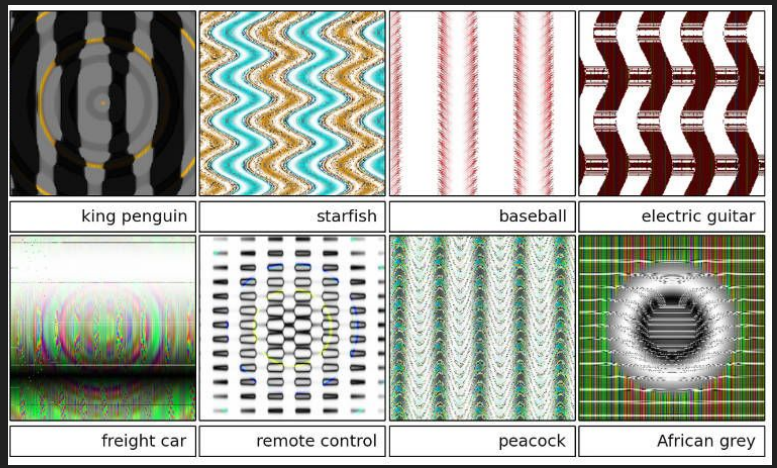
Why Convolutional Neural Networks (CNNs)?

- Spatial coherence of images
- Translation invariance
- Hierarchy of features

So, what is wrong with CNNs?

- Pooling layers
- Objects relationships
- Part-whole relationships



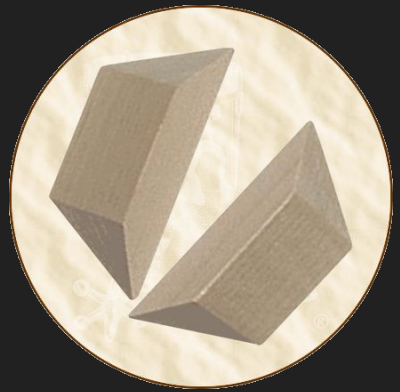


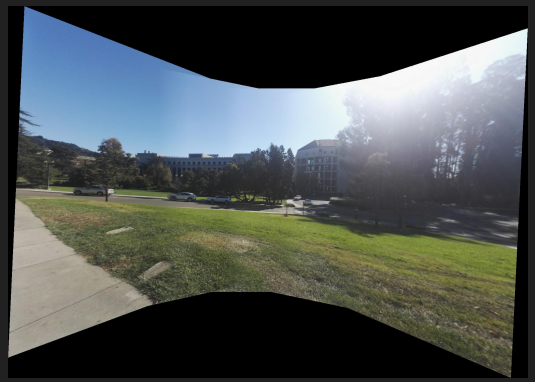
Motivation

What is wrong with pooling? [~40 minutes of Hinton's talk]

"The pooling operation used in CNNs is a big mistake
and the fact that it works so well is a disaster" -Geoffrey Hinton

- Bad fit to the psychology of shape perception
  We assign intrinsic coordinate frames to objects (think homography)

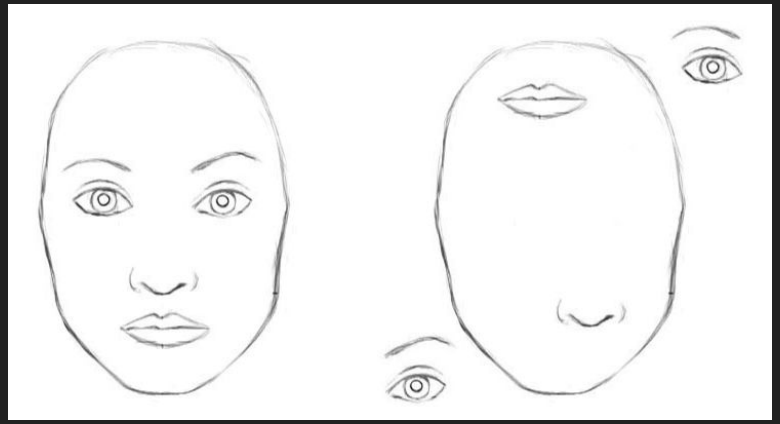




Motivation

What is wrong with pooling? [~40 minutes of Hinton's talk]

- It gives us invariance where we want equivariance
  Small translations are fine but big ones are not (place-coded vs rate-coded)



Motivation

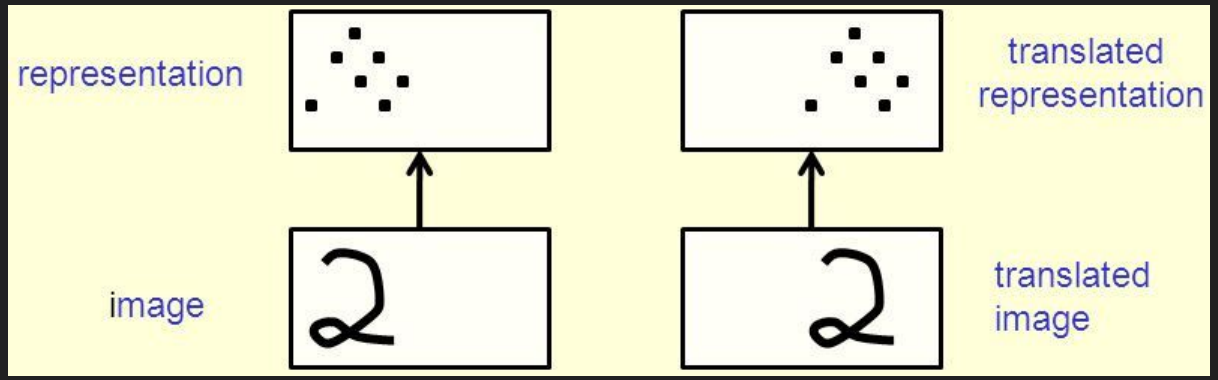
What is wrong with pooling? [~40 minutes of Hinton's talk]

- It fails to use the underlying linear structure (Inverse Graphics)

Motivation

What is wrong with pooling? [~40 minutes of Hinton's talk]

- It is a poor way to do dynamic routing
  Translated pixels should be processed by the same neurons (or Capsules)



Motivation

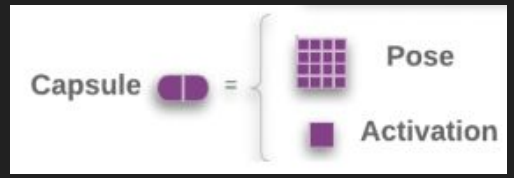
What is wrong with pooling? [~40 minutes of Hinton's talk]

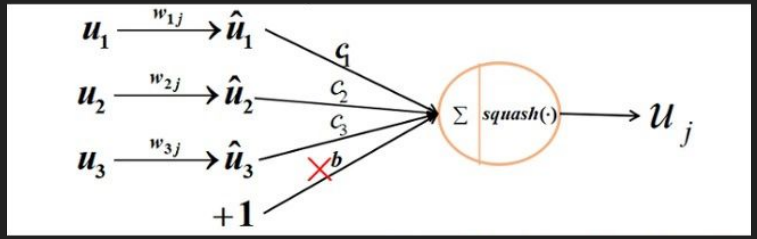
"The pooling operation used in CNNs is a big mistake
and the fact that it works so well is a disaster" -Geoffrey Hinton

- Bad fit to the psychology of shape perception:
  We assign intrinsic coordinate frames to objects (think homography)
- It gives us invariance where we want equivariance:
  Small translations are fine but big ones are not (place-coded vs rate-coded)
- It fails to use the underlying linear structure (Inverse Graphics)
- It is a poor way to do dynamic routing:
  Translated pixels should be processed by the same neurons (or capsules)

What is a Capsule?

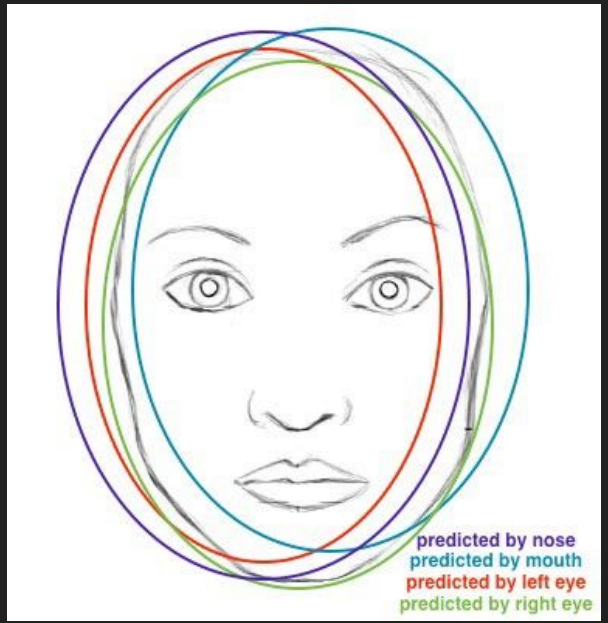
- Capsules encode features as vectors that contain the pose and the activation, so, instead of scalar neurons, now we have vectors
- By applying learned viewpoint invariant projection, on the capsules of a certain layer we are projecting these capsules/features to the coordinate frame of the next layer (low to high)
- The projected capsules are then weighted using dynamic routing
- Finally, a non-linear activation function is applied to the weighted sum





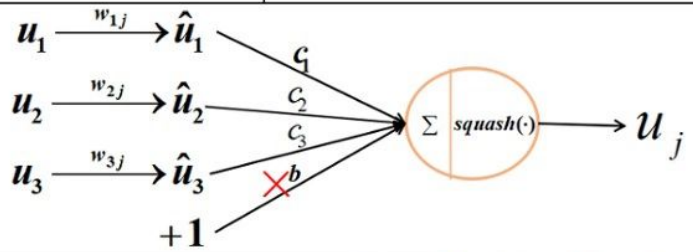
What is a Capsule?

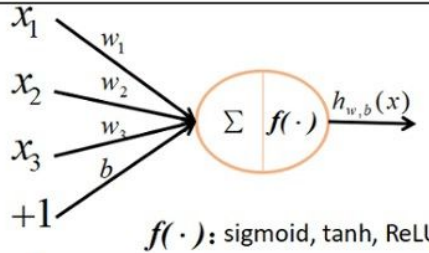
For example, lower capsules can be part detectors (eye, mouth, nose) and higher ones can be whole detectors (face)



Capsule vs Neurons

		capsule	vs.	traditional neuron
Input from low-level neuron/capsule		vector($\boldsymbol{u}_i$)		scalar(x_i)
Operation	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)		$a_j = \sum_{i=1}^{3} W_i x_i + b$
	Sum			
	Non-linearity activation fun	$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|}$ (Eq. 1)		$h_{w,b}(x) = f(a_j)$
output		vector(v_i)		scalar(h)





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

Dynamic Routing (A modern Hough transform)

A chance of 1 in a million for two 6 dimensional vectors to agree on every dimension within 10%.

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 $\mathbf{v}_j$

The number of iterations is r and 3 is found to be good in practice (more overfits)

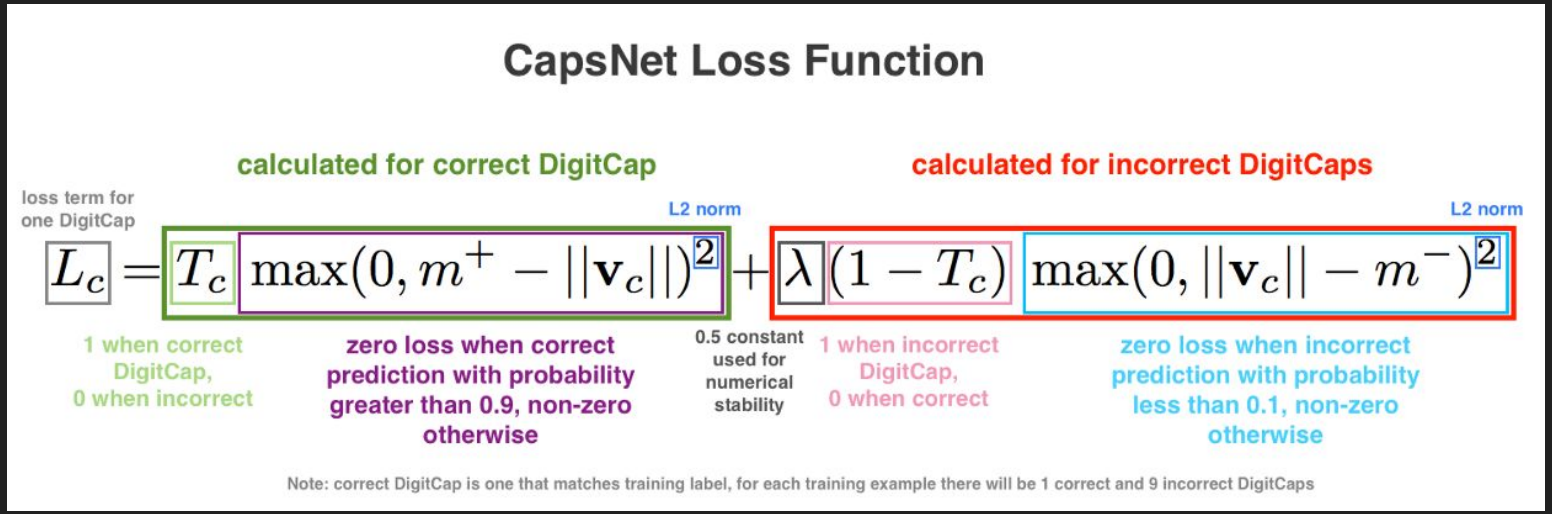
Dynamic Routing (A modern Hough transform)

Expectation-Maximization (EM) procedure

$$\text{score} = \sum_i \log p(\mathbf{x}_i | \textit{unifom-gauss-mixutre}) - \sum_i \log p(\mathbf{x}_i | \textit{uniform})$$

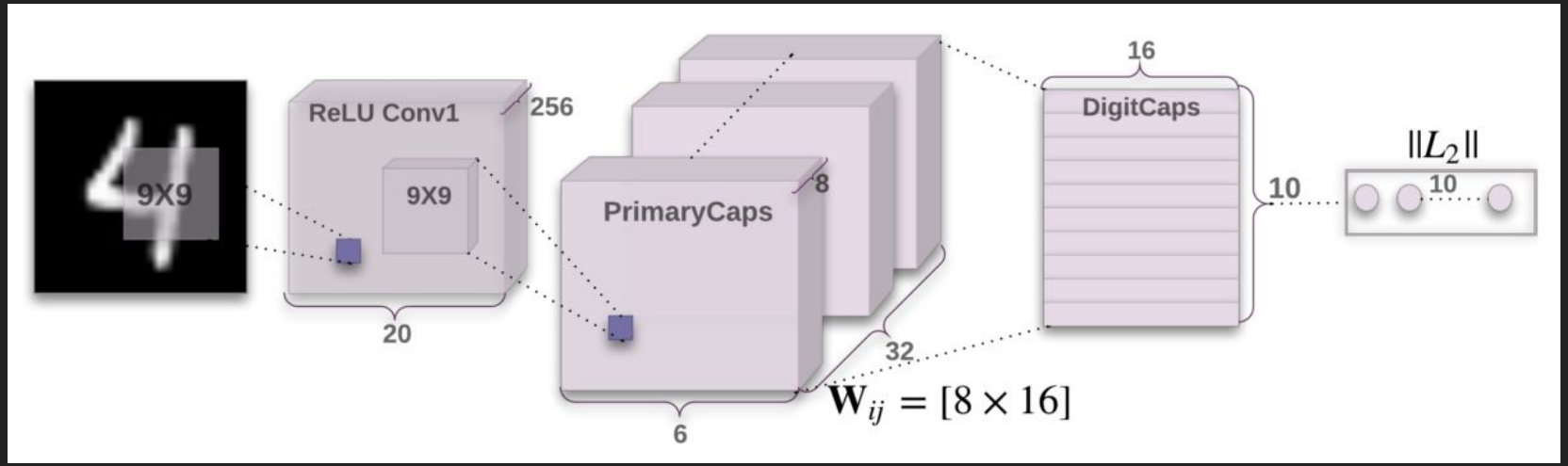
Loss function

Spread loss (similar to hinge loss)



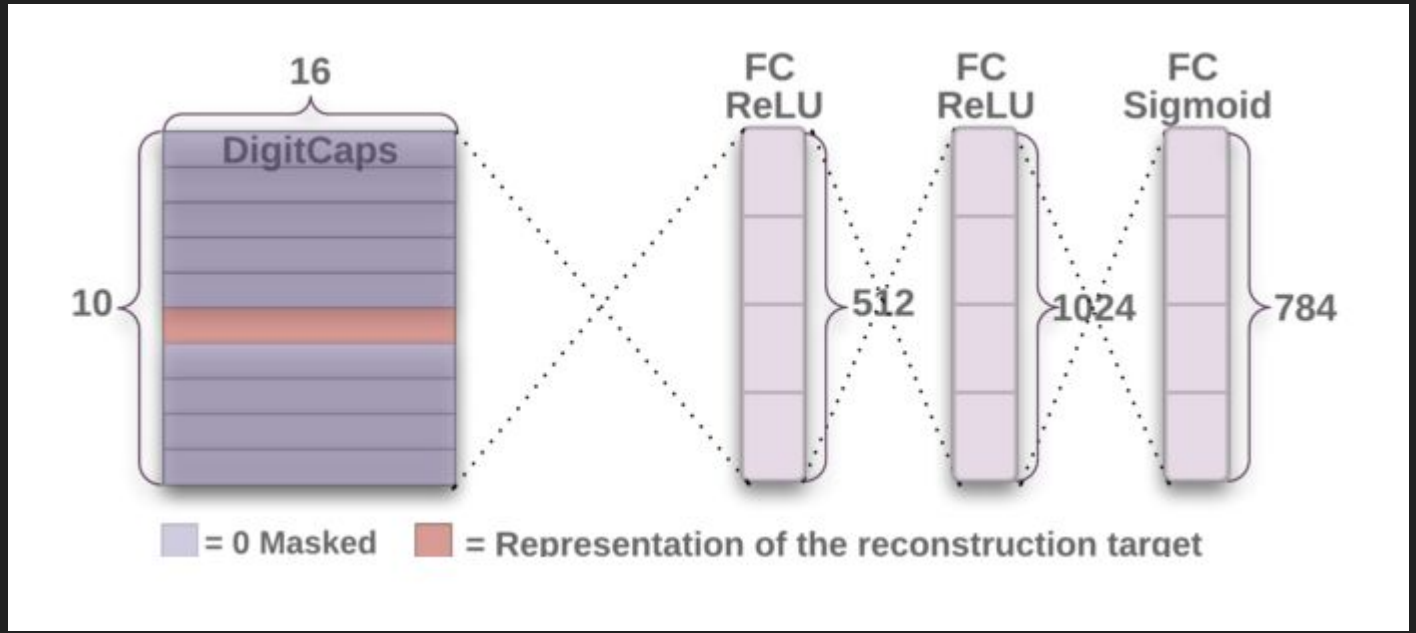
Architecture

Encoder Part

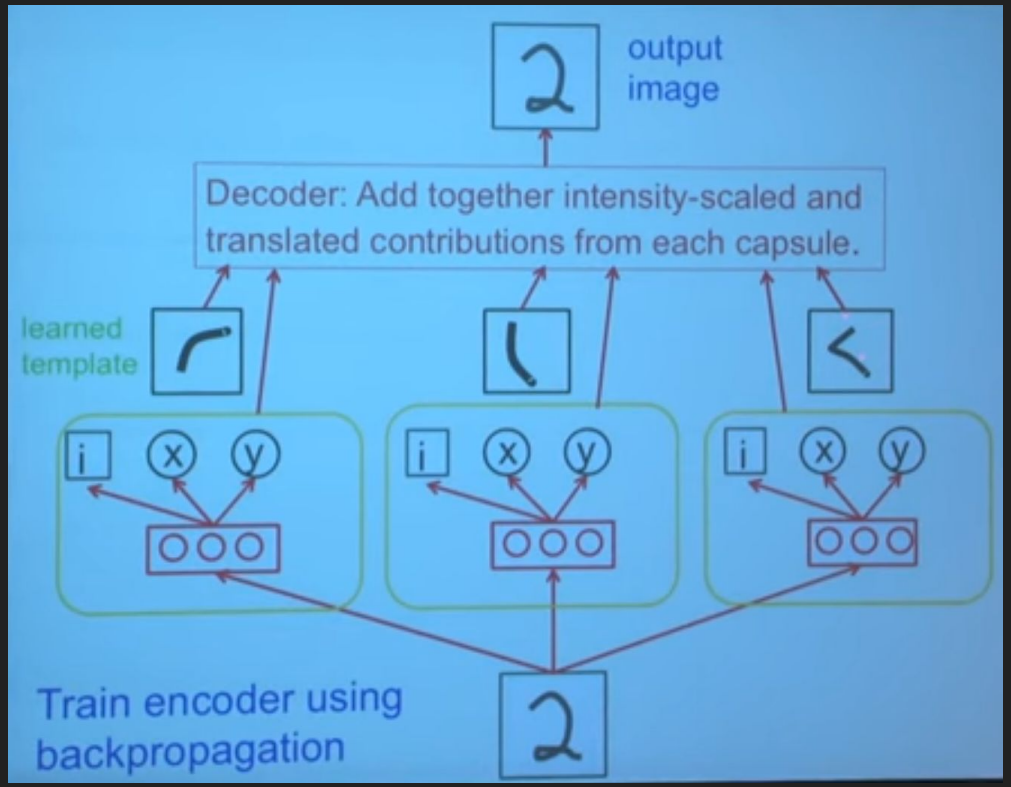


Architecture

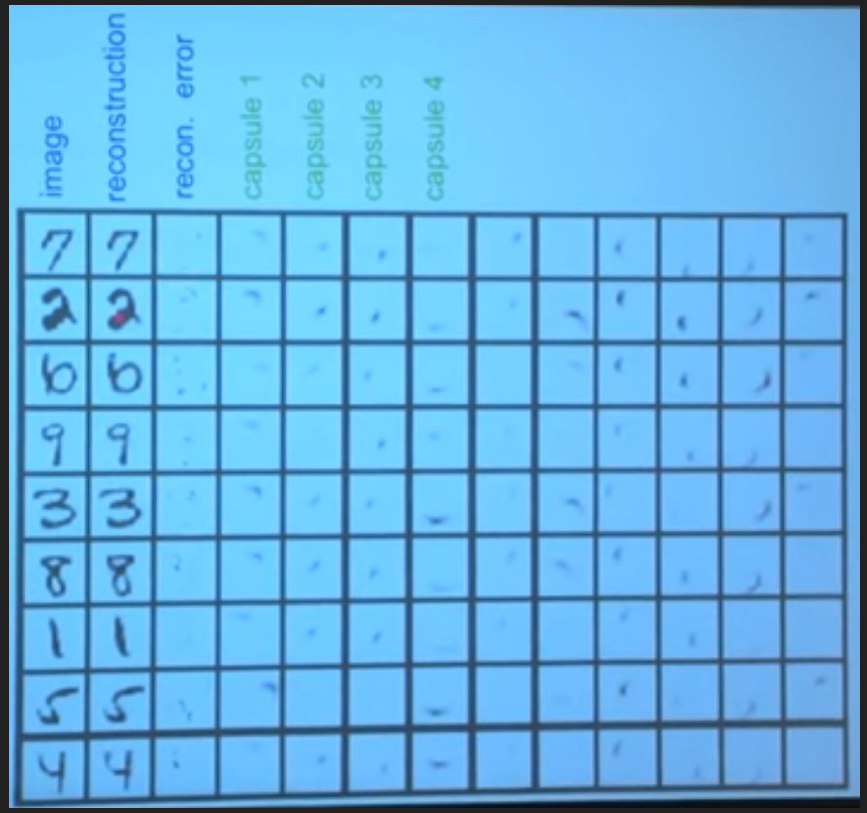
Decoder Part

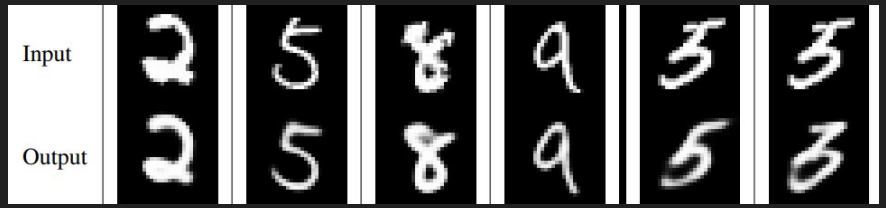


Proof of Concept on MNIST



Proof of Concept on MNIST





Thank You!

See the list of references

1. Dynamic routing between capsules [NIPS17]
2. Matrix capsules with EM routing [ICLR18]
3. What is wrong with ConvNets? [Video]
4. Does the brain do inverse graphics? [Video]
5. Nice series of blogs by Max Pechyonkin
6. Compilation of CapsNets references
7. PyTorch Tutorial by Dulat Yerzat [Code]