A Summary of Capsule Networks with an application on FashionMNIST

Cristóbal Silva Pérez
Universidad de Chile
crsilva@ing.uchile.cl

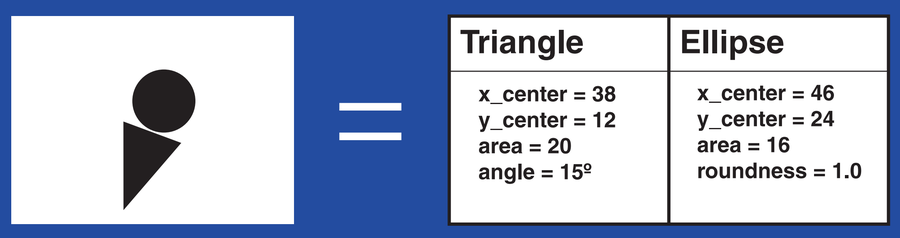
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS UNIVERSIDAD DE CHILE

GAMES
Grupo de Aprendizaje de Máquinas, infErencia y Señales
Centro de Modelamiento Matemático

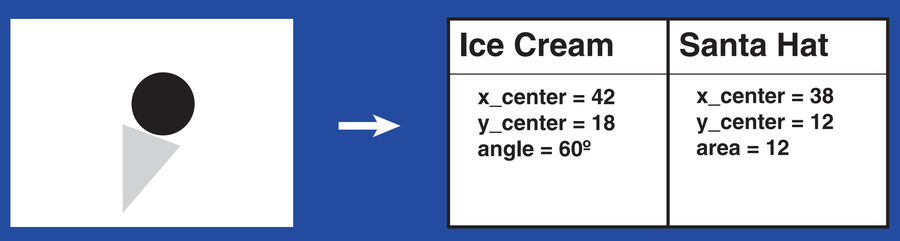
1. What are Capsules?

- Capsules are neurons whose **output is a vector** instead of a scalar value.

- Output **dimensions represent instantiation parameters** of a single entity. The length of the vector asserts whether such entity exists or not (activation).



- Each capsule in a layer **makes a prediction** via learned transformation matrices for the instantiation parameters of capsules in the layers above.



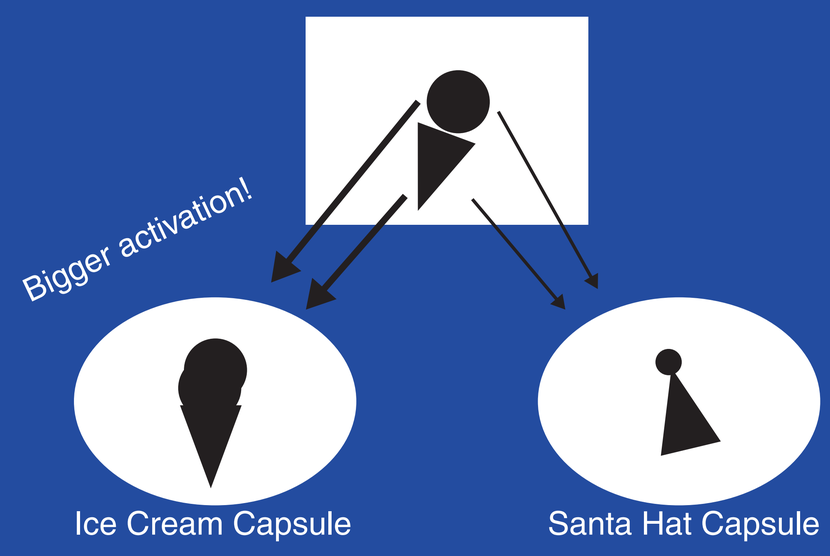
2. Dynamic Routing

- We can use the agreement of the predictions to route the output of capsules to the appropiate capsule a level above.

1. A low level capsule **weakly sends its output** to all the capsules in the level above.

2. Capsules above send information back to the low level capsules so they **redistribute the output**.

3. Iterate until convergence. The result is that capsules learn to **segment parts of a whole**.

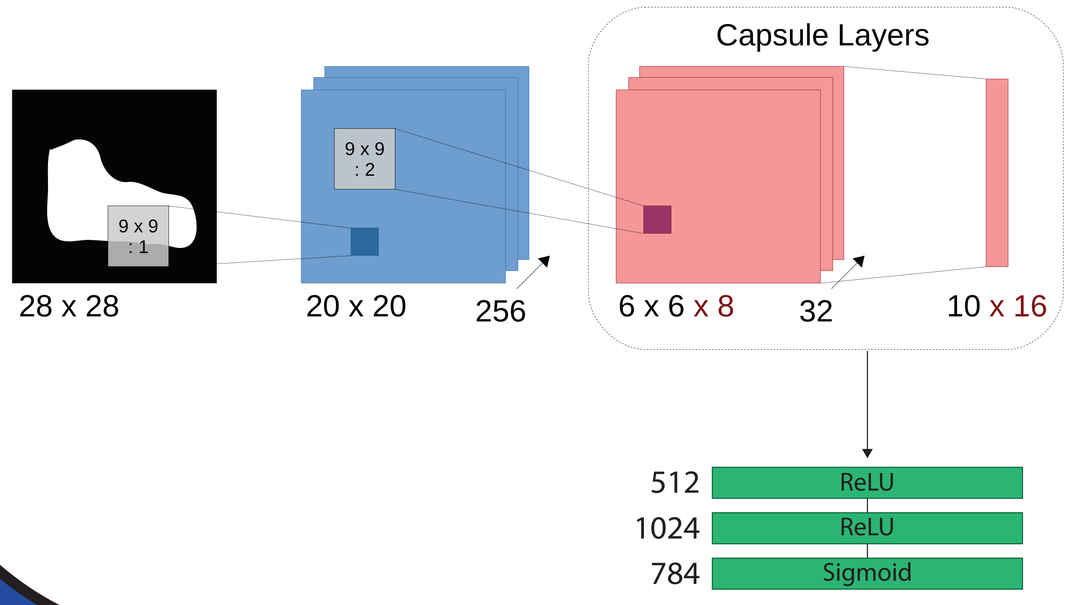


3. CapsNet Architecture

- Originally designed to classify MNIST digits, now trained with the FashionMNIST dataset to test previous findings.
  - Rather than digits, classes represent different clothes

- Instead of large hidden layers, we take each hidden layer and turn them into small capsules.

- Architecture details:
  - 1 convolutional layer
  - 1 capsule convolutional layer (8D capsules)
  - 1 capsule fully connected layer (16D capsules)
  - 1 decoder network



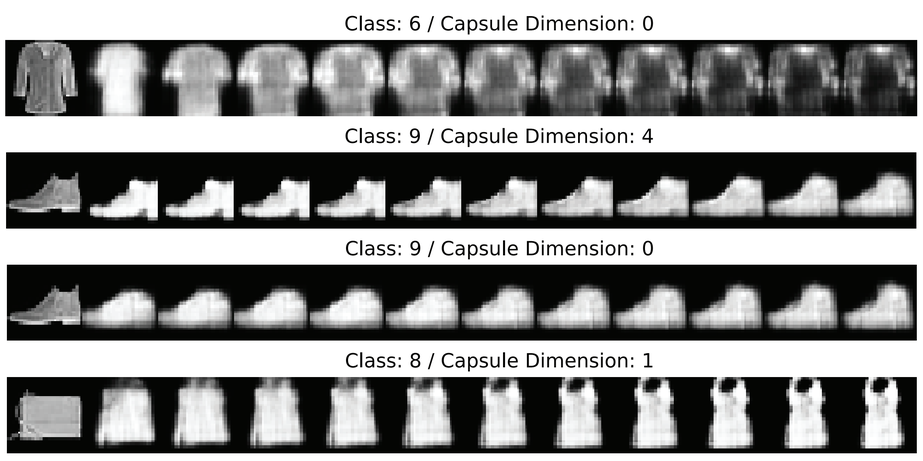
4. Training Results

- Training over FashionMNIST continues the trend of better results against convolutional architectures:

Table 1: Test Set results for the FashionMNIST dataset

Method	Test Accuracy
2 Conv Layers with max pooling	0.876
2 Conv Layers with 3 FC 1.8M parameters	0.932
WRN-28-10 + Random Erasing	**0.963**
CapsNet + 1 routing iteration	0.897
CapsNet + 3 routing iterations	0.927

- As per the original work, the last capsule layer is perturbed to check some of the instantiation parameters learned:



- Results are encouraging, with potential applications in scene segmentation in which objects can be found from various viewpoints and overlappings.

Based on the work of S. Sabour, N. Frosst, G. E. Hinton, *Dynamic Routing Between Capsules*, Advances in Neural Processing Systems 30 (pre-proceedings), 2017
PyTorch code available soon at http://github.com/canas. I would like to thank Xifeng Guo for the reference implementation in Keras.