Hinton's Capsule Networks

An introduction

Capsules?

Two fundamental papers:

- *Transforming Auto-Encoders* - Hinton, Krizhevsky, Wang (2011)
- *Dynamic Routing Between Capsules* - Sabour, Frosst, Hinton (2017)

Capsules?

To understand the concept, let's go back to *distributed representations* for a moment

Distributed Representations (reprise)

Hinton's idea of using deep neural networks to do pattern matching can be traced back to this concept

Distributed Representations, 1984

Distributed Representations (reprise)

Distributed Representations, 1984:

Each active unit represents a 'micro-feature' of an item [...]

Many people [...] can rapidly retrieve the item that satisfies the following partial description: It is an actor, it is intelligent, it is a politician.

Distributed Representations (reprise)

The idea that *features* of an object are useful in pattern recognition is not surprising

The core strength in deep networks is that these features can be learned from the data

The unit (or group of units) that identifies an actor can then be useful to recognize multiple patterns and **generalize** to combination of features which did not occur together specifically

Capsule Networks

The core idea in capsule networks is changing the fundamental unit of computation so that it is more reliable and resistant to transformations (think scaling, translation, rotation in Computer Vision) - from a neuron to a **capsule**

Capsule Networks

From **Transforming Auto-Encoders**:

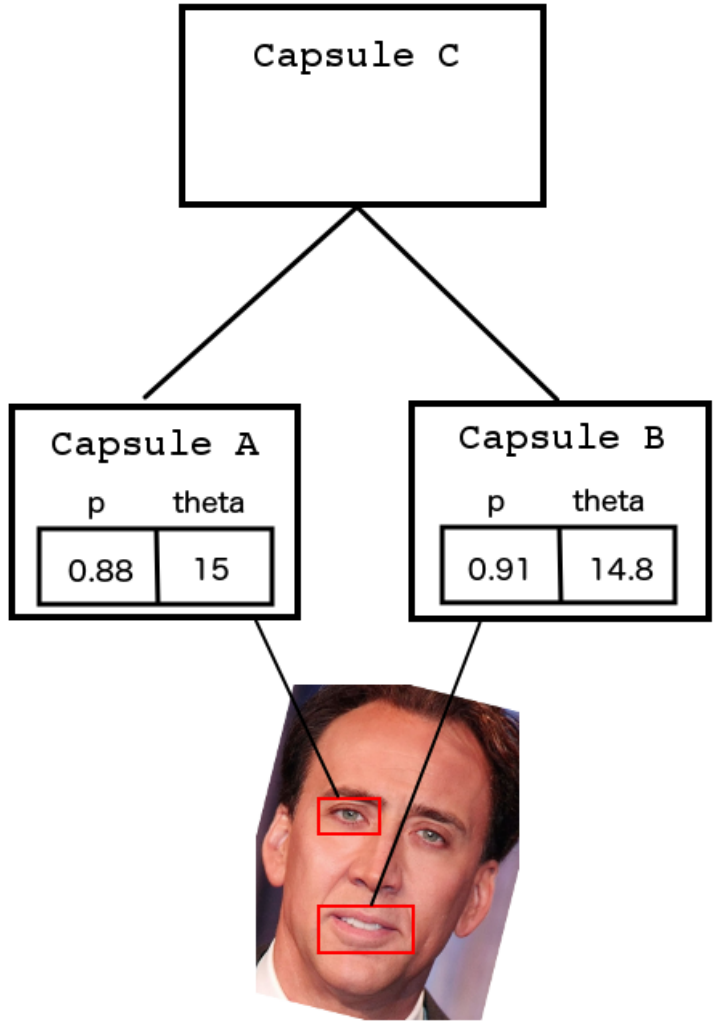
*[...] artifical neural networks should use local "capsules" that perform some quite complicated internal computations on their inputs and then encapsulate the results of these computations into a **small vector of highly informative outputs** [...]*

Capsule Networks

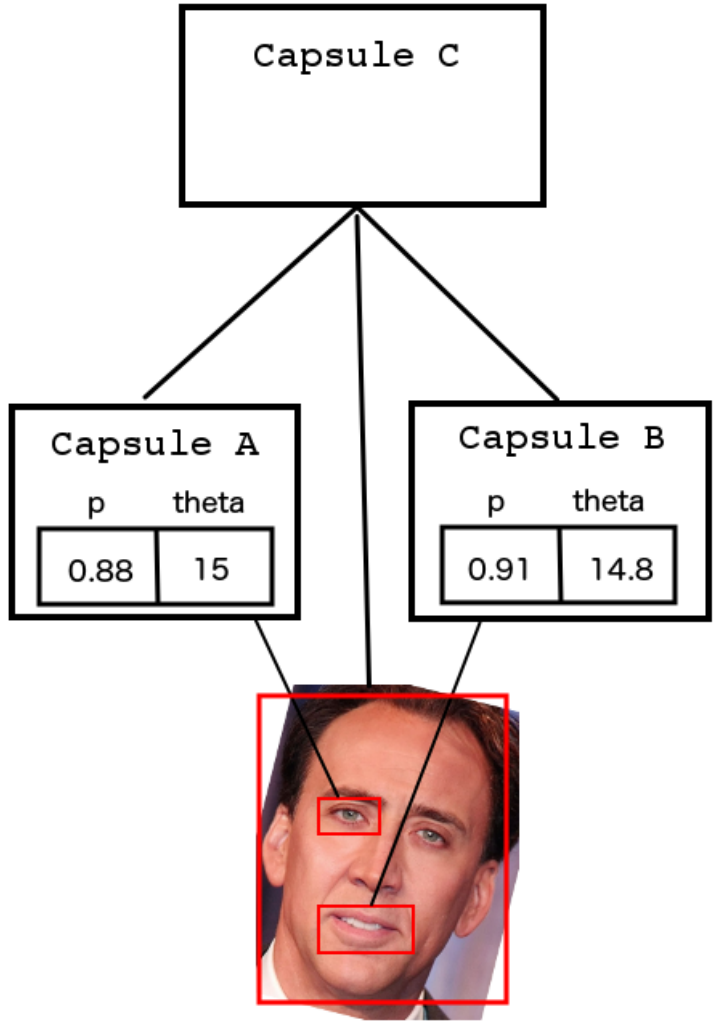
To Hinton, a capsule can be trained to recognize both a "visual entity" and the transformations it has been subjected to

This can be useful to "recognize wholes by recognizing their parts"

Part-whole pattern recognition



Part-whole pattern recognition



Part-whole pattern recognition

If we have reliable part detectors, many CV tasks become trivial

The issue is building those! Hinton knows it, of course:

"But where do the first-level capsules come from? How can an artificial neural network learn to convert the language of pixel intensities to the language of pose parameters?"

Transforming Auto-Encoder

A transforming auto-encoder is an auto-encoder which has also learned to model a few transformations

It uses a number of independent capsules, each one made of a feedforward network with two types of units (neurons):

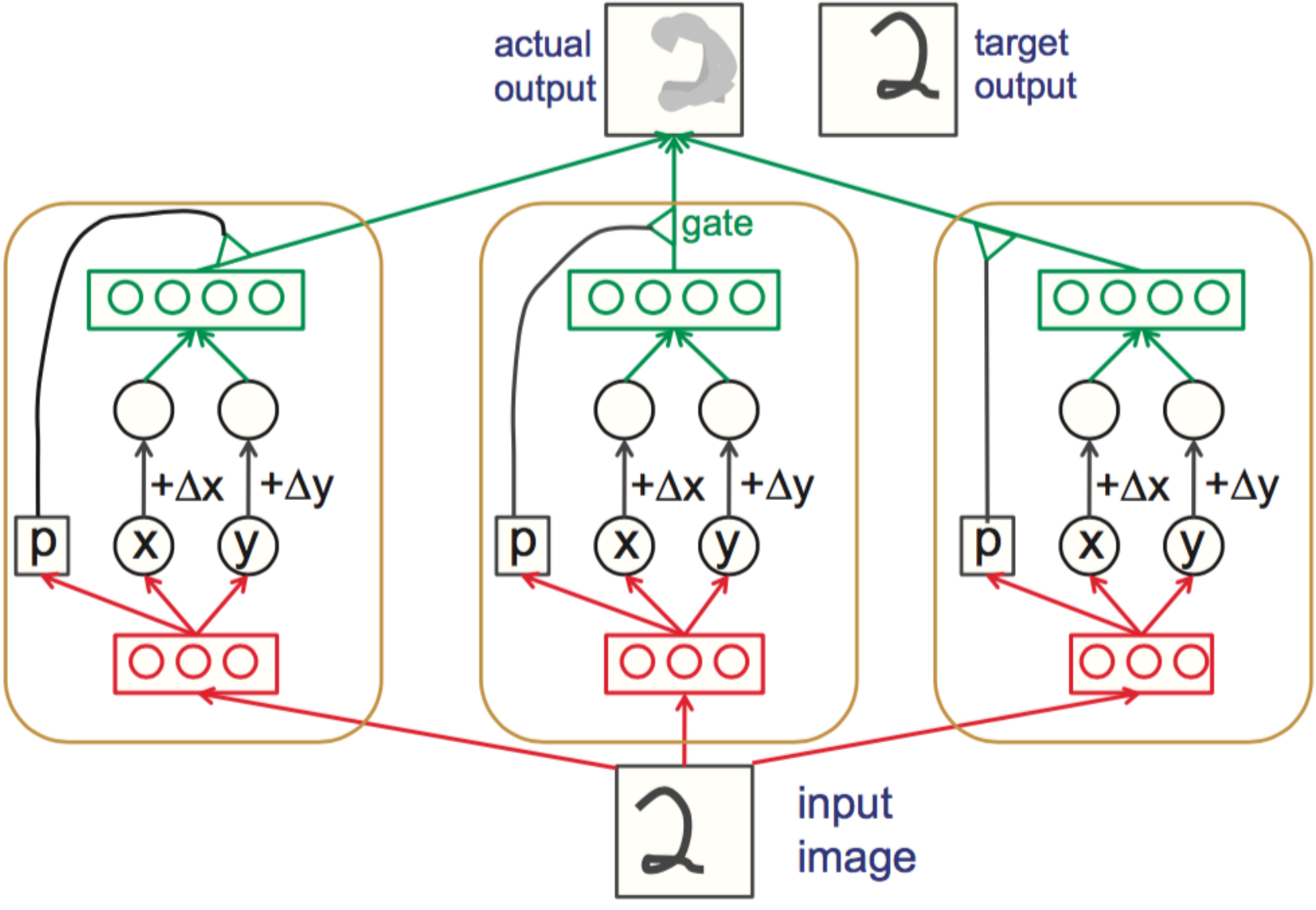
- *recognition units* model the probability that the visual entity is appearing and the transformation parameters;
- *generation units* compute the capsule's contribution to the generated image - this is weighted by the probability of the visual entity appearing, so inactive capsules do not contribute to the generated images

Transforming Auto-Encoder

An easy example is a "translation auto-encoder", which is given the images and the desired translation parameters Δx and Δy as input

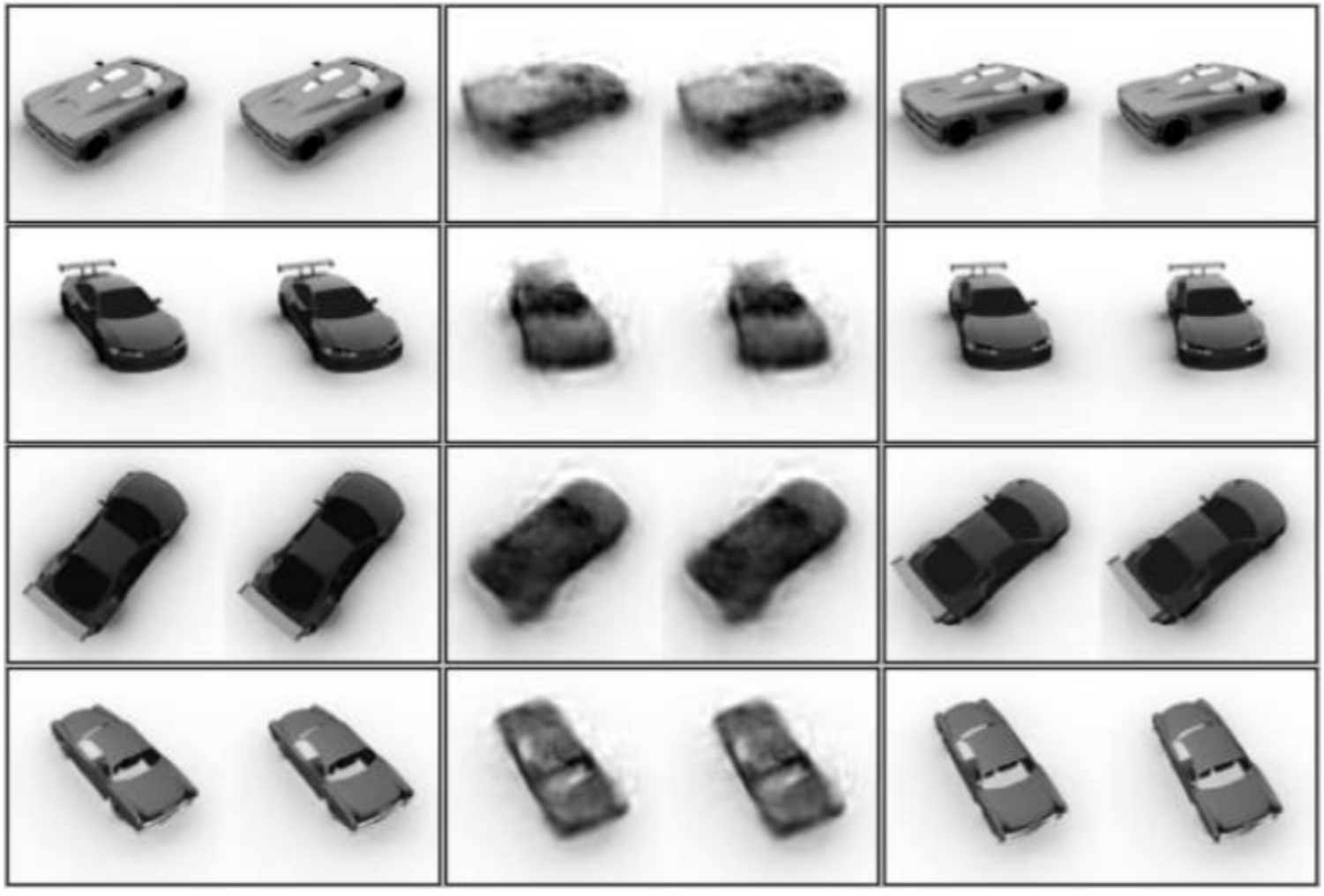
In *Transforming Auto-Encoders*, such a model is trained on a "translated MNIST" dataset, and manages to generate translated digits correctly

Transforming Auto-Encoder



Transforming Auto-Encoder

Harder experiment: learning a viewpoint transformation matrix



Discussion (1)

Limitations of the capsule concept in general:

*[...] it is not possible for a capsule to represent more than one instance of its visual entity at the same time**

Discussion (2)

Generality of the transforming auto-encoder:

*A transforming auto-encoder can force the outputs of a capsule to represent any property of an image that we can manipulate in a known way. It is easy, for example, to scale up all of the pixel intensities.**

Discussion (3)

Why probabilities?

Thank you!