Last updated on Wednesday, September 27, 2017

2017 IFN680 - Assignment Two (Siamese network)

Assessment information

- Code and report submission due on **Monday 30th October, 08.30am**
- Use **Blackboard** to submit your work
- Group size: three people per submission. Smaller group sizes allowed (1 or 2 people).

Overview

- You will implement a deep neural network classifier to predict whether two images belong to the same class. The dataset you will use is a set of images of handwritten digits.
- The approach you are asked to follow is quite generic and can be apply to problems where we seek to determine whether two inputs belong to the same equivalence class.
- You are provided with scaffolding code that you will need to complete with your own functions.
- You will also perform experiments and report on your results.

Introduction

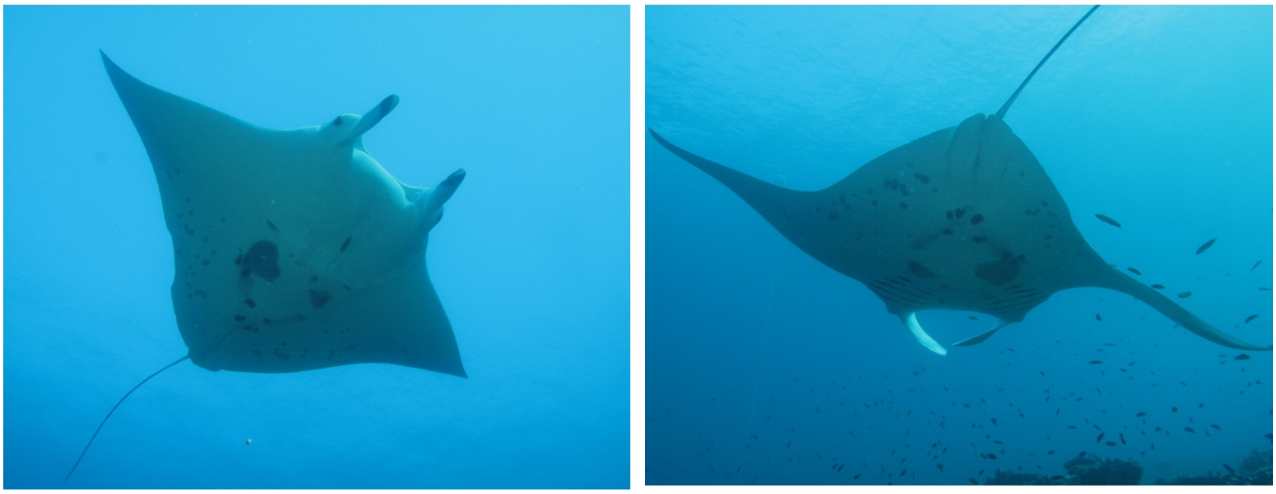
Despite impressive results in object classification, verification and recognition, most deep neural network based recognition systems become brittle when the view point of the camera changes dramatically. Robustness to geometric transformations is highly desirable for applications like wild life monitoring where there is no control on the pose of the objects of interest. The images of different objects viewed from various observation points define *equivalence classes* where by definition two images are said to be equivalent if they are views from the same object.

These equivalence classes can be learned via *embeddings* that map the input images to vectors of real numbers. During training, equivalent images are mapped to vectors that get pulled closer together, whereas if the images are not equivalent their associated vectors get pulled apart.

Background

Common machine learning tasks like classification and recognition involve learning an appearance model. These tasks can be interpreted and even reduced to the problem of learning manifolds from a training set. Useful appearance models create an invariant representation of the objects of interest under a range of conditions. A good representation should combine *invariance* and *discriminability*. For example, in facial recognition where the task is to compare two images and determine whether they show the same person, the output of the system should be invariant to the pose of the heads. More generally, the category of an object contained in an image should be invariant to viewpoint changes.

This assignment borrows ideas we developed for a manta ray recognition system. The motivation for our research work is the lack of fully automated identification systems for manta rays. The techniques developed for such systems can also potentially be applied to other marine species that bear a unique pattern on their surface. The task of recognizing manta rays is challenging because of the heterogeneity of photographic conditions and equipment used in acquiring manta ray photo ID images like those in the figures below.



Two images of the same Manta ray

Many of those pictures are submitted by recreational divers. For those pictures, the camera parameters are generally not known. The state of the art for manta ray recognition is a system that requires user input to manually align and normalize the 2D orientation of the ray within the image. Moreover, the user has to select a rectangular region of interest containing the spot pattern. The images have also to be of high quality. In practice, marine biologists still prefer to use their own decision tree that they run manually.

In order to develop robust algorithms for recognizing manta spot patterns, a research student and I have considered the problem of recognizing artificially generated patterns subjected to projective transformations that simulate changes in the camera view point.

Artificial data allowed us to experiment with a large amount of patterns and compare different network architectures to select the most suitable for learning geometric equivalence. Our experiments have demonstrated that Siamese[1] convolutional neural networks are able to discriminate between patterns subjected to large homographic transformations.

In this assignment, you will work with a simpler dataset. Namely the MNIST dataset. You will build a classifier to predict whether two images are warped views from digits of the same class or not.

[1] Siamese network are defined in the next section.

Learning equivalence classes

A *Siamese* network consists of two identical subnetworks that share the same weights followed by a distance calculation layer. The input of a Siamese network is a pair of images *Pi* and *Pj*. If the two images are deemed from the same equivalence classes, the pair is called a *positive pair*, whereas for a pair of images from different equivalence classes, the pair is called *a negative pair.*

The input images *Pi* and *Pj* are fed to the twin subnetworks to produce two vector representations *f(Pi)* and *f(Pj)* that are used to calculate a proxy distance. The training of a Siamese network is done on a collection of positive and negative pairs. Learning is performed by optimizing a *contrastive loss* function (see code documentation). The aim is to minimize the distance between a pair of images from the same equivalence class while maximizing the distance between a pair of different equivalence classes.

Your tasks

- You are provided with scaffolding code and an example of a Siamese network based on a multi-layer perceptron network. You need to familiarize yourself with this code, then adapt it to your need.
- Write code to create a Siamese network based on a convolutional network with at least 3 convolutional layers, and 2 fully connected layers. No need to use more than 5 convolutional layers!
- Create a dataset of 100000 warped images using the provided function *random_deform*. Use the call *im2 = assign2_utils.random_deform(im1, 45, 0.3) to* warp the image *im1* with a rotation of at most 45 degrees and a projective transformation with a “strength” of 0.3.
- Find a suitable architecture for the base convolutional neural network of your Siamese network.
- Train your Siamese network on the original dataset (without warping) and report the performance of your network.
- Train your Siamese network on the warped dataset and report the performance of your network.
- Investigate whether starting training the network on easier pairs improve the classifier. That is, start training the network on images that have a deformation significantly smaller than 45 degree and with the strength parameter smaller than 0.3.
- Describe your findings in the report. Use tables and figures to support your arguments.

Submission

You should submit via Blackboard a zip file containing
A **report in pdf format** strictly **limited to 8 pages in total.**

- explain clearly your methodology for your experiments
- present your experimental results using tables and figures

Your Python file **my_submission.py**

Marking Guide Focus

- **Report**:
  - Structure (sections, page numbers), grammar, no typos.
  - Clarity of explanations.
  - Figures and tables (use for explanations and to report performance).
- **Code quality**:
  - Readability, meaningful variable names.
  - Proper use of Python constructs like numpy arrays, dictionaries and list comprehension.
  - Header comments in classes and functions.
  - Function parameter documentation.
  - In-line comments.
- **Experiments**
  - Soundness of the methodology
  - Evidence based discussion/conclusion

Final Remarks

- Do not underestimate the workload. Start early. You are strongly encouraged to ask questions during the practical sessions.
- Email questions to f.maire@qut.edu.au