Identifying potential hardening techniques for image classifiers

Autonomous systems and machine learning is an ever accelerating field. In the specific case of object detection and image classification, research into neural networks and their applications rapidly became a main focus within the open source community. The famous ImageNet Large Scale Visual Recognition Challenge (ILSVRC) set the precedent in the field for developing networks with high levels of accuracy and as such, the security of these networks was often a secondary concern. These types of networks can be easily exploited using adversarial imagery. In which, perturbations are added to an image that will then cause the network to misclassify an image.

Often, these images are crafted to exploit specific types of classifier i.e. trained on certain types of data or use different architecture. Largely, this becomes an issue when the training of these types of classifiers is outsourced – how do we trust the system? Therefore, investigations into protecting these models are incredibly important. For this tasking, the possible different hardening techniques fall largely into two categories; data manipulation and network manipulation.

Data Manipulation:

- Can training data be manipulated sufficiently that a classifier trained on that data will be robust to a variety of adversarial methods?
- Can extra, statistical information be extracted about the data and fed in at the training stage (i.e. extra information that an adversary cannot access).

Network Manipulation:

- Can hardening be introduced into the pipeline before the training stages? I.e. can the architecture of the network be altered such that adversarial images introduced in training have little to no effect on the network?
- Is there a way to tell if a network has been compromised by bad data after it has been trained? E.g. “badnets” [1].

It is likely that combinations of the above will be required to harden a neural network sufficiently against the majority of attacks. However, insight into why individual components of the system i.e. data, architecture, functions have certain effects on the behaviour of the system would be extremely beneficial.

References:

[1] https://arxiv.org/abs/1708.06733