The Limitations of Deep Learning in Adversarial Settings

Ravi Raju

5-2-2019

1 Motivation

The paper goes into background about the different settings that an adversary is subject under to attack a neural network model. They want to solve the problem of how Carlini-Wagner posed their threat model. That is, given the neural network F, an image sample x and an adverserial label, Y^*, does there exist a perturbation vector δ_x such that $F(X + \delta_x) = Y^*$. This can be modeled as an optimization problem:

$$\arg\min ||\delta_x||_p \, s.t. \, F(X + \delta_x) = Y^* \tag{1}$$

Their approach to attacking models is that they want to look at what input changes would cause large output variations whereas previous attacks looked at output variations to induce the corresponding input perturbations.

To this end, they essentially do this by finding the Jacobian of the function of the network. They basically state these are more effective than gradient-based attacks. We obviously know that this false since we have the CW-attack which has not been defeated yet. The adverserial saliency map enables an efficient exploration of the adversarial-sample space.

2 Solution

In Section III, the paper introduces an example network which learns the XOR function and how analyzing the Jacobian matrix of this function demonstrates the vulnerability of the network. Consider Figure 5. It is clear that a small change in the input at x_2 will cause a misclassification. The Jacobian being small is the reason why the search space for adverserial examples is reduced since you don't really care about the regions when the derivative is small. The basic procedure for finding the salincy map is as follows:

1. Compute the forward derivative $\nabla F(X^*)$, where X^* is an adverserial sample.
2. Construct a saliency map S based on this derivative.

3. Modify one input feature, i_{max} by some θ.

This process is iterated until a sample is found or the maximum distance (distortion) factor is reached. The first item is essentially an artifact of forward autodiff. The saliency map is effectively a tool to which input feature is the one to be altered for the misclassification to occur. The rule is basically trying to increase the output probability of your target class while decreasing the probabilities of the other classes. For the final step, the θ that is introduced is problem specific and should be discussed with respect to the results.

3 Results

The question is that should the parameter θ increase or decrease the pixel intensity? This question is explored in section IV. For the saliency map, a pair of pixels is considered since very few pixels would meet the heuristic search criteria in Equation 8. The pixel values can compensate if one the other pixels has minor flaws. Decreasing the pixel intensity also induces adverserial samples.

The model that was evaluated was the LeNet Architecture on MNIST. The main result from the paper is that 97.10% misclassification based on changing 4.02% of the input features.

4 Discussion/Takeaway

They find that decreasing pixel intensity is less effective at finding adverserial examples as opposed to increasing the intensity since removing pixels makes it harder to extract information to classify the input. Later in the paper, they mention a metric known as adverserial distance which they empirically find that values that are close to one are harder to misclassify. In the last discussion portion of the paper, they mention adding adverserial samples to training set as a form of regularization.