Approximate Regular Expressions: A Comparison of Exact and Approximate Matching Methods

Umme Salma Gadriwala, Tasnim Noshin, Rumsha Siddiqui

{gadriway, noshint, siddiqur}@mcmaster.ca

Department of Computing and Software, McMaster University
1280 Main St. W, Hamilton, Ontario, Canada L8S 4L8

December 2018

Tweet us
your thoughts:
[worth pursuing?]
[achievable?]
@TeamDeepCheck
#MacCSCapstone
#MacEng

Motivation

Security is of utmost importance given the fundamental use of neural networks in safety-critical systems. Major industries such as autonomous vehicles [**?**], medicine [**?**] and military defense [**?**] are at risk of being compromised through neural network attacks.

Problem

Education in neural network security is uncharted territory. It is additionally difficult to identify neural network trade-offs, limitations, and vulnerabilities.

Solution

Help neural network developers uncover potential vulnerabilities, trade-offs and limitations, producing a report of findings for the general public.

Project Objectives

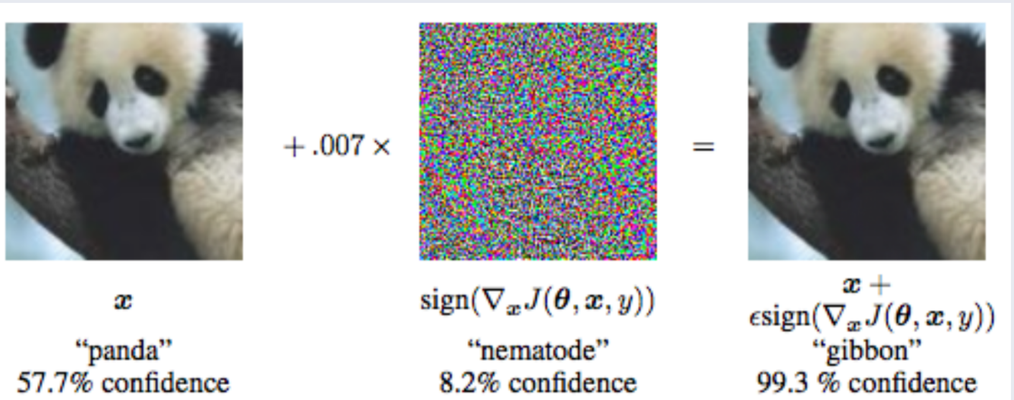
We propose a tool to help developers of neural networks uncover vulnerabilities to a set of known attacks. DeepCheck will consist of two main parts:

- Susceptibility measurements of a neural network in response to different categories of attacks. For the scope of this project, we will focus on adversarial attacks from IBM's Adversarial Robustness Toolbox [**?**] and Cleverhans [**?**].
- A configurable task that can be added to a build system (i.e. GitLab CI) so that deep learning developers can test and identify vulnerabilities in their neural nets before deployment.

Background Study

Neural Networks: Inspired by the human brain, a neural network aims to recognize relationships in a data set using a series of calculations & algorithms. Data is inputted into a neural network as a Euclidean vector and propagates through network layers until the final layer is reached. The node with the heaviest weight is outputted as the classification prediction, with a percentage of certainty.

Adversarial Samples: Neural networks are vulnerable to adversarial samples. Small changes to the input vector can result in misleading activations and a final misclassification. This makes it difficult to apply neural networks in security-critical areas. Development of neural network defenses is a growing area of research.



The above demonstrates how adding a miniscule amount of carefully constructed noise leads to an image that still looks like a "panda" to a human, but that the network thinks is a "gibbon". The *Fast Gradient Sign Method* is one way to generate adversarial samples. Let θ be the parameters of the model (i.e. the weights of the neural network), x the input, y the target label associated with x, and $J(\theta, x, y)$ the loss function (i.e. the objective function used to train the network). The neural network outputs an incorrect label with high confidence (a "gibbon", vs. what is clearly a "panda" to the human eye). [**?**].

Defenses: Thus far, no defense is fool-proof. *Adversarial Training* is one attempt to generalize a model by generating adversarial samples for training; but not all possible samples are generated.

Testing Mechanism: In this recently growing field, it is an arms race between developing attacks and robust defences. While there are libraries of attacks and adversarial sample detection, there is no standard testing mechanism in place to evaluate novel algorithms and approaches.

Methodology

In our proposed evaluation metric, we compare the accuracy of a given neural network's output of an untampered sample set to that of an adversarial sample set. For each adversarial attack, we plan to conduct the following:

1. Determine the accuracy of a given neural network on an untampered test set by identifying the percentage of outputs that are true positive (TP), false positive (FP), true negative (TN) and false negative (FN).
2. Run an adversarial attack on the untampered test set to generate an adversarial sample set. Determine the accuracy of the neural net when running the adversarial sample set by identifying the percentage of outputs that are TP, FP, TN and FN.
3. Report the difference in accuracy percentages between the adversarial and untampered test sets in a confusion matrix.
4. Run a defense on the neural network. Determine the accuracy percentages of the defended neural net when inputting the adversarial sample set.
5. Report the difference in accuracy percentages between the defended and non-defended neural network when running the adversarial samples.

Attack	**Confusion Matrix**		**Confusion Matrix After Defenses**		
NewtonFool	**TP:** 30% --> 25% = -5%	**FP:** 5% --> 17% = +12%	**Defense**	**Type**	**Resulting Confusion Matrix**
	FN: 5% --> 8% = +3%	**TN:** 60% --> 50% = -10%	Feature Squeezing	Input Transformation	**TP:** 25% --> 28% = +3% / **FP:** 17% --> 10% = -7% / **FN:** 8% --> 6% = -2% / **TN:** 50% --> 56% = +6%
			Label Smoothing	Output Transformation	**TP:** 25% --> 29% = +4% / **FP:** 17% --> 10% = -7% / **FN:** 50% --> 55% = +5% / **TN:** 8% --> 6% = -2%
			...	...	...
Universal Perturbation	...		...		
...	...		...		

The reported percentages in each quadrant of the matrix can be used by deep learning developers to analyze potential changes in accuracy, sensitivity, specificity and precision.

Conclusions & Future Work

The proposed metric allows us to evaluate the effects of various adversarial attacks and potential neural network robustness using known defences. In the next phase of our work, we will implement this metric to explore the vulnerabilities of neural networks using Keras and Tensorflow. We further hope to conduct a survey of deep learning researchers' perspectives on neural net verification to explore the sociological views of using deep learning models in security critical systems.

Acknowledgements

We thank Dr. R. Samavi and Yifan Ou for the opportunity and support in exploring security in deep learning machine models, and Dr. C. Anand for his continued support and guidance in our learning and endeavours.

References