Homework 3: Simple Countermeasures

Course: Computational Intelligence and Adversarial Learning

Janzaib Masood
Department of Electrical and Computer Engineering
Auburn University

Question #1: Given: A Simple Neural Network that you can probe,

- **Is it possible to determine the type of neural network being used and reverse engineer it?**
  Yes, it is possible.
- **How would you go about doing this?**
  This process can be done in a step by step fashion, if one knows the working of a neural network in general.
  1. Know your inputs and outputs (2 floats intput,1 float output).
  2. Know the nature of the task (classification or **regression** / sequential or **combinational**).
  3. As the probes can be connected anywhere in network, so we can know how a single forward pass occurs (how an input vector is transformed to produce the answer). Here it is simply a gaussian model where we adjust the standard deviation in the learning process.
  4. By looking at dimensions for every layer, we can draw the architecture of the neural network. But in this case, it is simply a single neuron which takes inputs x and y and hyperparameter sigma to generate each output.

  Note: knowing the type or architecture of the neural network does not necessarily reveal how/why it does the decision making (particularly when networks are deeper).

  This answer to this part lies in the **interpretability methods** which help us identify the semantic understanding of a model's working. The results of this type of analysis generates heatmaps in the end. Heatmaps can help identify which input features help to excite an output, and which features suppress it.

- **What kind of countermeasure can be used to prevent this from happening?**
  To prevent someone from understanding the mechanics/structure of a neural network model, we can simply publish the model in a format which cannot be probed. And avoid releasing information of the layer dimensions, activations, regularizations, parameter matrices and hyperparameters etc.

  Note: There are several model agnostic methods which do not care about the structure of a model, but they can still figure out the input features that trigger the output in positive/negative ways.
  To prevent our models from being looked at, we can also limit the number of forward passes a single user can do through them.

Question #2: A training set T, and an evaluation set E (Note: E consists of the inputs given to the network as well as the outputs that the network generated), and the fact that you cannot probe the Simple Neural Network.

- **Is it possible to determine the value of sigma used to create E?**
  Yes, it is possible.
  1. Make sure input in set T and E are from same distribution.
  2. Start with a random sigma.
  3. Update sigma by reducing a cost which simply looks for an input/output transformation of E that replicates distribution T.
  4. Stop the sigma updates, when you feel that E is similar enough to E.

  Assuming that T and E are both from the same distribution, we can use the standard deviation of T in the Simple Neural Network to create E.

- **If so, what kind of countermeasure can be used to prevent this from happening?**
  A simple countermeasure can be, hiding the input/output values of T in a particular range.
  For example: hide the training examples where inputs are between 0.4 and 0.6