ECGAdv Type I Attack Algorithm: Modified version of the Carlini & Wagner (C&W) Attack

The Carlini & Wagner attack is currently the best known attack algorithm to generate adversarial examples, it was published in IEEE S&P 2017 in the paper titled 'Towards Evaluating the Robustness of Neural Networks'. Here, I want to briefly provide a high level overview of the C&W Attack algorithm.

Designing an attack algorithm means formulating an optimization problem to generate adversarial examples and specifying a way to solve that problem followed by experimentally verifying that it actually works. When adversarial examples were first discovered by Szegedy in 2013, the optimization problem to craft adversarial examples was formulated as follows:

$$\textbf{minimize: } D(x, x+\delta)$$
$$\textbf{such that: } C(x+\delta) = t \quad \text{------ constraint 1}$$
$$x + \delta \in [0, 1]^n \quad \text{------ constraint 2}$$

where, x= input image, δ = perturbations, D = distance metric between the adversarial and real image, C = Classifier function, n= dimensions, t = target class. The distance metric is usually specified in terms of L_p norms (L_0, L_2 or L_∞). Constraint 1 makes sure that the image is indeed misclassified and constraint 2 makes sure that the adversarial image is valid i.e. it lies within the normalized dimensions of x.

Traditionally well known way to solve this optimization problem is to define an objective function (generally but not necessarily a loss function that penalizes as we move further from satisfying the constraints) and to perform gradient descent on it; which guides us to a optimal point in the function. However, The formulation above is difficult to solve because $C(x+\delta) = t$ is highly non-linear (the classifier is not a straight forward linear function). In the work by Szegedy in 2013, he resorts to solving the problem approximately, using 2nd order optimization technique known as L-BFGS.

But here, Carlini expresses constraint 1 in a different form as an objective function 'f' (the reasoning behind this is that this form is suited better for optimization) such that when $C(x+\delta) = t$ is satisfied $f(x+\delta) \leq 0$ is also satisfied

Conceptually, the objective function tells us "how close we are getting to being classified as t". One simple example fo the function 'f', one that is not a very good choice but is well suited for explanation is:

$$f = [\, 1 - C(x + \delta)_T \,]$$

Where $C(x + \delta)_T$ is the probability of $x + \delta$ being classified as t. If the probability is low, value of f is closer to 1 whereas when it is classified as t, f is equal to 0.

In the paper, Carlini evaluates 7 different objective functions (all of them are loss functions) 'f' and selects the best one among them that is given by:

$$f(x') = \max(\max\{ Z(x')_i : i \neq t \} - Z(x')_t , -k)$$

Here, $Z(x')$ is the logit (the unnormalized raw probability predictions of the model for each class/ a vector of probabilities) when the input is adversarial example x′.

$\max\{ Z(x')_i : i \neq t \}$ is highest probability for non target class (it represents what is the best prediction among non target classes).

$Z(x')_t$ is the probability of the target class (it represents how confident is the model on misclassifying adversarial example x' as the target class).

$\max\{ Z(x')_i : i \neq t \} - Z(x')_t$ to put it in lay terms, is the difference between "what the model thinks the current image most probably is" and "what we want it to think/ misclassified target". So when the model thinks that this image is what we want it to think, this value is -ve (the probability of target class is higher than any of the non target classes)

The above term is essentially the difference of two probability values, so when we specify another term -k and take a max, we set a lower limit on the value of loss i.e. at least the value given in -k will always hold. Hence, by controlling the parameter '-k' we can specify how confident we want our adversarial example to be classified as.

Again, the criteria for the objective function $f(x + \delta) \leq 0$ is validly held when misclassification occurs. Further, he points out that in the earliest work of Szegedy, the objective function was chosen as the cross entropy loss which he found is the worst performing among the 7 choices in the evaluation done in the paper.

Carlini then reformulates the original optimization problem given by Szegedy using a well known trick in the Optimization problems' literature, that is to move the difficult of the given constraints into the minimization function.

$$\text{minimize: } D(x, x + \delta) + c.f(x + \delta)$$

$$\text{such that: } x + \delta \in [0, 1]^n \quad \text{----- constraint 2}$$

Hence, we only have one constraint. However, a constant 'c' is introduced, when the constraint denoted by f is merged in the minimization function. It is obvious that the

value of c should be greater than 0 (since f is penalizing/ loss and the optimization is minimization). However, the value of c should be suitably chosen, Carlini experimentally finds that the best way to choose value of c is use the smallest value of c for which the misclassification occurs (**f (x + δ) ≤ 0** occurs).

In the experiment, they found that having a small value of c results in the attack rarely succeeding and having a large value of c results in attack being less effective (large value of L2 distance) but always succeeding. They resort to using binary search to figure out a value of c.

Now, expressing the formulation in terms of L_p norm instead as distance D, it becomes:

minimize: $\| \delta \|_p$ + c.f(x + δ)
such that: x + $\delta \in$ [0, 1]^n ––––– constraint 2

Now we have an optimization problem expressed in a way that is easy to solve and understand but we still have one problem with constraint 2. It is expressed in this particular form known "box constraint" which means that there is an upper and lower bound set to the constraint 2. In the work by Szegedy, they use a solver (algorithms that solve optimization problems) known as L-BFGS (2nd order solver) which natively supports box constraints. This box constraints are not natively supported by solvers like Stochastic Gradient Descent (SGD) and its variants.

Hence, Carlini evaluates 3 methods that the use box constraints in such solvers and select one method known as "change of variables" in which instead of optimizing over variable δ in constraint 2, we optimize over w, which is given by:

δ **= (1/2)* { tanh(w) + 1} - x**
or, x + δ **= (1/2)*{ tanh(w) + 1 }** ––––– constraint 2

Where, tanh is the hyperbolic tangent function. So when tanh(w) varies from -1 to +1, x + δ varies from 0 to 1. The advantage of using this formulation is it allows them to use solver like Adam (introduced in 2014). Adam is a first order iterative algorithm that is better (computationally efficient, less memory) than the classical SGD algorithm. The main difference is that Adam has an adaptive learning rate whereas SGD has it fixed.

The final form of the optimization problem using L_2 distance metric as used in the Type I attack for ECGAdv is:

minimize: $\|$ (1/2)*{ tanh(w) + 1- x $\|_2$ + c.f[(1/2)*{ tanh(w) + 1 }]
such that: tanh(w) varies from -1 to +1

$$\text{where, } f(x') = \max(\max\{ Z(x')_i : i \neq t \} - Z(x')_t , -k)$$

Therefore, the C&W attack is the solution to the optimization problem (optimized over w) given above using Adam optimizer. To avoid gradient descent getting stuck, they use multiple starting point gradient descent as the solver.

The modifications performed in ECGAdv to original C&W attack are:

1. Change the dimensions of the input to (1, 9000, 1) - 3 dimensions [they mention that this is a requirement by Keras]
2. The similarity metrics defined for images in C&W attack is 3 L_p norms. In ECGAdv, they evaluate the C&W attack for 3 metrics (L_2 norm, D_{smooth} and $D_{smooth,L2}$)

Some of the extra caveats of the original C&W attack are:

1. They demonstrated that the best known defense mechanism 'Defensive Distillation' by Nicholas Papernot, published in IEEE S&P 2016 failed for the C&W attack. They showed for previous known weak attacks, the success rate of this defense was high (less than 1% Adversarial examples could bypass this defense)
2. This attack algorithm can generate stronger/ higher quality adversarial examples but the cost to generate them is also higher (because of the way optimization problem is formulated). Even though, Carlini suggests some relaxations in the paper to reduce the cost, it is still costly.
3. The general advice given in the paper to evaluate any new defense mechanism is to evaluate it against strong attacks like the C&W and not others. (Although I found that in his blog Carlini suggests that this advice is outdated). And that robustness against weak attacks is useless because even when defenses worked for weak attacks, they failed for a strong attack with a 100% probability.
4. Compared to attack techniques like the Fast Gradient Sign Method (FGSM), another major difference here is that in the original formulation of the C&W attack problem, the authors do not specify a threshold that sets a limits the maximum distortion that is allowed. This is done in FGSM with a parameter 'ellipson'. What this means is that, the attack here always succeeds.
5. Carlini showed that adversarial examples are classified with a higher confidence than real images for C&W attack. This might be true for other attack algorithms as well.
6. This paper has over 2000 citations and is one of the most popular papers in Adversarial Machine Learning field.