1 Literature Review

1.1 Adversarial examples

An adversarial example is a sample from the original distribution of the data that has some small noise added to it such that the image remains almost identical to the original, but it is able to fool a model that has been trained on a standard dataset. More formally for a sample $x \in \mathbb{R}^n$ with label y and a classification model $f_\theta(x)$ with parameters θ, we can define the adversarial perturbation $r(x) \in \mathbb{R}^n$ as the solution to :

$$\begin{aligned} \arg\max_{\delta} \quad & \ell(f(x+\delta;\theta), y) \\ \text{s.t.} \quad & \|\delta\| \leq \epsilon \end{aligned}$$

Where ℓ is the loss function and the norm of constraint is either infinity or 2.This is a untargeted attack and the type mostly used in my work. The other possibility is targeting a specific label other than the original and minimizing the loss.

Different attacks try to solve this optimization problem with different methods. Two of the most famous ones are FGSM(The fast gradient sign method) and PGD (projected gradient descent) attacks. FGSM uses a simple one step optimization to solve the problem where the perturbation is bounded in a $l_{\inf}$ ball by ϵ. The adversarial example is defined as:

$$x + \varepsilon \operatorname{sgn}\left(\nabla_x L(x, y, \theta)\right) \tag{1}$$

Note that in these optimizations the variable is the input x and not the weights θ. PGD uses multiple steps of projected gradient descent to minimize the negative of loss, which gives it a better chance at finding examples that maximize the loss. This comes at the cost of slowing down the process and choosing an appropriate rate α. The PGD adversarial example is given by:

$$x^{t+1} = \Pi_{x+C}\left(x^t + \alpha \operatorname{sgn}\left(\nabla_x L(x, y, \theta)\right)\right) \tag{2}$$

Here as before C is the set of all possible perturbations.

Robust training: Robust training is the process of training a model such that it is resistant to a set of adversarial attacks. To the best of my knowledge it has not been possible to train a model that is resistant to all possible attacks. The process of robustly training a network with respect to a set of attacks is identical to the standard training of a neural network except each sample needs to be replaced by the adversarial examples that the chosen set of attacks produce. Usually it is best if these are examples that maximize the loss function, since they represent the worst case scenario for the network. This method was first suggested by [**madry2019deep**], it is not trivial why this method works and [**madry2019deep**] uses the Danskin's Theorem to prove the correctness of their method. Formally robust training can be formulated as:

$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{r \in C} L(x + r, y, \theta)\right] \tag{3}$$

Where D is the dataset distribution. We can see that the inner maximization is a set C constrained adversarial example.

1.2 Self-supervised learning

Self-supervised learning is the process of training a model based on a

1.3 Disentanglement

[**isolates_Content_from_Style_2021**]