IFT 6756 - Lecture 5
(Adversarial Examples)

This version of the notes has not yet been thoroughly checked. Please report any bugs to the scribes or instructor.

Scribes **Instructor:** Gauthier Gidel
Winter 2021: [Mathieu Godbout, François Mercier, Sharath Chandra Raparthy]

1 Summary

In the previous lecture we talked about the Empirical Risk Minimization (ERM) framework. In ERM, we consider a predictor f with parameters θ (e.g. a neural network) for which we aim to learn the best parameters according to a training set $\{(x_i, y_i)\}_{i=1}^N$. This yields the ERM objective

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(f_\theta(x_i), y_i\right), \tag{1}$$

where ℓ is a loss function that characterizes the proximity between the predictor's output for x_i and the actual expected output y_i.

In this lecture we take for granted that we have parameters θ that allow for good performance on both the train and test sets. We then ask: how does the predictor behave on examples that are close to those sets, yet are not exactly part of the same distribution ? For instance, if the predictor is an image classifier, is it possible to modify an image so that it remains the same to the human eye but the predictor now makes a mistake when it tries to classify it ?

2 Adversarial examples

The answer to the above questions is affirmative and is best described by an example. As can be seen on figure 1, one can find an image that is exactly the same to the human eye but that fools a state-of-the-art image classifier. Such examples that are close to the distribution of the training dataset but are mispredicted by a trained model are referred to as **adversarial examples**. The notion of adversity comes from the fact that such examples can be seen as attacks against a given predictor, where the goal of the attack is to fool the predictor as much as possible.

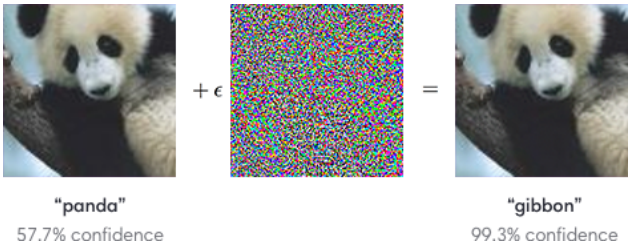

Figure 1: Adversarial example from Goodfellow et al. [8]. Here, a perturbation with $\epsilon = 0.007$ is used, yielding no difference to the human eye for the image but completely fooling a trained GoogLeNet [13] network.

2.1 Formal definition

The attack presented in figure 1 is not random: it is specifically tailored for the image classifier used and the panda image example. Generally speaking, we are interested in computing an optimal attack of a predictor f given an input example (x, y) and a loss function ℓ. Our aim is to find x' that is close enough to x so that the label y should also be an appropriate target output, but that is far enough from x to fool the network. Since the network is trained via ERM for a loss function ℓ, fooling it is represented by making it incur a high loss. This yields the natural definition below of **best attack**.

Definition 1 (Best attack)**.** *For a predictor f, an example pair (x, y), a loss function ℓ and a distance metric d, we define the best attack x' as*

$$\max_{x' \in \mathcal{X}} \ell(f(x'), y) \quad \textit{such that} \quad d(x', x) \leq \epsilon,$$

where $\mathcal{X}$ is the set of all admissible inputs.

2.1.1 Insights

The first thing to note from the definition of best attack is that the choice of d is arbitrary and is only constrained to being a meaningful distance metric between two inputs $x, x' \in \mathcal{X}$. For instance, if the predictor's inputs are images, then any L_p norm can be used as a distance metric if we consider the difference between the images

$$d(x, x') = \|x - x'\|_p = \left(\sum_{i=1}^{d} |x_i - x'_i|^p \right)^{\frac{1}{p}}.$$

Of all L_p norms, L_2 and L_∞ are the most frequently encountered in the context of adversarial examples. For those, we have

$$\|x - x'\|_2 = \sqrt{\sum_{i=1}^{d} (x_i - x'_i)^2}, \quad \|x - x'\|_\infty = \max_{i=1,\dots,d} |x_i - x'_i|,$$

respectively known as the Euclidean and Chebyshev distances. Figure 2 illustrates how different values of p make the L_p norms behave on the unit square. As can be seen on the figure, the more p is increased, the more the L_p norm induced becomes focused on the biggest component of the measured vectors. In particular, in the case of images as inputs, we can view each RGB pixel of an image as 3d vector of pixel values and the L_∞ distance between two images is simply the pixel value that most differs between them.

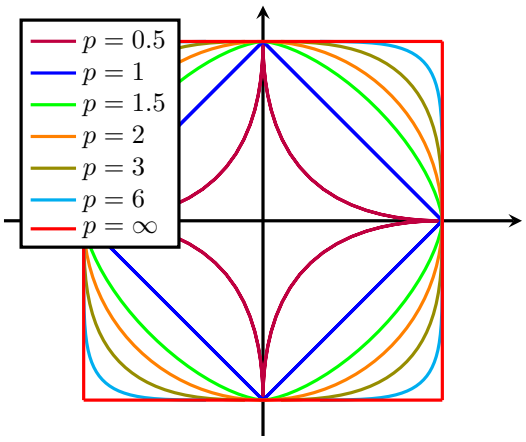

Figure 2: Visualization of the 2d points $\mathbf{x}$ where $\|\mathbf{x}\|_p = 1$ for different values of p on the unit square.

The second thing to note from definition 1 is that the adversarial example x' also has to be in $\mathcal{X}$. This constraint is natural in the sense that it forces the best attack to take place in the observation space $\mathcal{X}$ where we want the predictor to learn. If we come back to the image classification problem, this means that any adversarial input x' has to still be an image. Specifically, if we consider images with pixel values ranging from 0 to 1, then it is important that the pixel values in x' are all still in the $[0, 1]$ interval.

One last important thing to note is that the best attack is defined with respect to a given predictor f. In order to accurately attack a predictor, it is crucial to have access to some kind of information on the attacked predictor to quickly find its weaknesses and exploit them.

2.2 Threat models

The best attack in Definition 1 can be viewed as an optimization problem where the aim is to find the optimal adversarial example x'.

Definition 2 (Adversarial examples as optimization problem)**.** *For a predictor f, an example pair (x, y), a loss function ℓ and a distance metric d, the best adversarial example can be computed by solving*

$$x' \in \arg\max_{x' \in \mathcal{X}} \ell(f(x'), y) \quad \textit{subject to} \quad d(x, x') \leq \epsilon$$

We are interested in solving the maximization problem in Definition 2. To do so, we define different threat models which correspond to the different levels of knowledge of f we assume available. In this section, we will attempt to classify these threat models based on the assumptions they make.

- **White Box Threat Model:** In the case of white box treat model, the attacker assumes access to the model parameters. This assumption is quite strong and hence limits it's applicability to real world systems.
- **Black Box Threat Model:** In the case of black box treat model, the attacker no more has a privilege to access the model's complete information and can only query from the model.
- **Practical Black Box Threat Model:** This is even more realistic model where the assumptions are further reduced. Here there is a limitation on number of queries we can make on the model.
- **NoBox Threat Model:** This was recently introduced by Bose et al. where we assume is not have any access to the threat model $f(\cdot)$ but we only assume to have some knowledge (Ex: model's architecture) about it. This makes the setting even more challenging and hence more realistic.

2.2.1 White box threat model

In this section, we dig deep into solving the optimization problem 2 where we have access to the threat model f_t. The problem is a constrained optimization problem where we want x' close to x and also x' should be a real image in $\mathcal{X}$. Since we are using a norm constraint to keep the images close to real images, we use a gradient method that correspond to the geometry of the constraint. In particular we use projected gradient ascent algorithm. Goodfellow et al. proposed a method to generate the adversarial examples x'. The main idea is to replace the random perturbation with a more meaningful perturbation which corresponds to the sign of the gradient at each pixel.

Formally, let x be the input to the model f, y be the corresponding target and $\ell(f(x), y)$ be the associated loss function. Then the perturbation is generated by calculating the gradient of the loss with respect to the **input x** (but not the model parameters) $\nabla_x \ell(f(x), y))$. We take the sign of this gradient $\text{sign}(\nabla_x \ell(f(x), y))$ to ensure that the gradient is well adapted to the $L_\infty-$ball. We finally generate the adversarial example x' by adding the perturbation $\text{sign}(\nabla_x \ell(f(x), y))$ proportional to the input x.

$$x' = x + \epsilon \,\text{sign}(\nabla_x \ell(f(x), y))$$

Here intuitively we are adding a perturbation to the original image x which would increase the loss. We can show that the scaled perturbation $\epsilon \,\text{sign}(\nabla_x \ell(f(x), y))$ we do here is equivalent to the steepest ascent direction that maximizes

the inner product $p^T \nabla_x \ell(f(x), y)$ that stays within the constraint $||p||_\infty \leq \epsilon$ (here p is the distance between x and x'). Formally this is written as

$$\epsilon \operatorname{sign}(\nabla_x \ell(f(x), y)) = \max_{||p||_\infty \leq \epsilon} p^T \nabla_x \ell(f(x), y)$$

To prove this let $g = \nabla_x \ell(f(x), y)$. Now we can write $\max_{||p||_\infty \leq \epsilon} p^T \nabla_x \ell(f(x), y)$ as

$$\begin{aligned} \max_{||p||_\infty \leq \epsilon} p^T \nabla_x \ell(f(x), y) &= \max_{||p||_\infty \leq \epsilon} p^T g \\ &= \max_{|p_i| \leq \epsilon} \sum_i p_i g_i \quad (\text{since } ||p||_\infty \leq \epsilon \iff |p_i| \leq \epsilon \; \forall i) \\ &\leq \sum_i |g_i| \epsilon \quad (\text{since } |p_i| \text{ can be at max } \epsilon) \end{aligned} \tag{2}$$

This is maximized only when $p_i = \epsilon \operatorname{sign}(g_i)$. This is a single step method, meaning we are generating the adversarial example with only single gradient update. This is hence computationally very efficient. One can also think of generating fancier attacks by taking more gradient steps [10] or by adding momentum [7]. (Note: While taking more gradient steps we need to make sure that we respect the both these constraints; $x' \in [0, 1]^d$ and $||x - x'|| \leq \epsilon$)

2.2.2 Black box threat model

As explained earlier, in case of the black box threat model we don't have access to the gradients of f_t but we can only query from the model to obtain the probability scores. But from what we have seen so far we need to estimate the gradients of the target model f_t in order to produce an adversarial image. Chen et al. proposes a method called Zeroth-Order Optimization (ZOO), to approximate the gradient by doing multiple forward passes on the model with some perturbation δe_i on the input. The gradient is approximated as

$$\nabla \ell(f(x)) \approx \frac{\ell(f(x)) - \ell(f(x + \delta e_i))}{\delta}$$

This by first principles is an approximation of the gradient. Black-box threat models will not be extensively discussed in this course so if you are curious to explore, please refer this survey by [2]

2.3 Defences

Similar to research on adversarial examples attacks, several defensive strategies have been elaborated over the past years. At their core, they try to make a model f to be robust to adversarial examples attacks. The following list provides some examples of these defenses:

- **Gradient masking.** The idea is to have a model f with discontinuities, and therefore no gradient information for some ranges. Intuitively, this defense makes harder the optimization problem for attacking the model f. However, this defense can be overcome, see Athalye et al. [1] for further information.
- **Preprocessing of inputs.** The idea is to transform the original image in such way that finding a small noise to fool the model f will be harder. For example, one way could be to round pixel values. Thus, the distance between the original images and a perturbed image will be even smaller. However, designing such preprocessing steps is hard in practise and it contradicts a key benefit from machine learning to leverage learning over handcrafted knowledge.
- **Adversarial training.** The idea for this defense is to train the model f on the original dataset (x, y) but also on perturbed variations of x. This approach is detailed further in section 2.3.2.
- **Many more.** Defense strategies are an open and active research topic. For further pointers to other types of defense strategies, please refer to the list from www.robust-ml.org.

2.3.1 Targeted adversarial attacks

Before diving further in defensive strategies, we cover targeted adversarial attacks. Targeted adversarial attacks are adversarial examples attacks with the addition constraint that we explicitly attempt to fool the model with a wanted target class.

Definition 3 (Targeted adversarial attacks)**.** . *Let f a model, x an original input, x' a perturbed input, t the target class that we expect from the model f with the perturbed input x'.*

$$x' \in \underset{||x-x'|| \leq \epsilon}{\arg\min}\ \ell(f(x'), t)$$

In the paper Carlini and Wagner [4], authors explored different loss functions l for the definition 3. Let F the output from softmax, Z the logits.

$$\begin{aligned}
\ell_1(\boldsymbol{x'}) &= -cross_entropy_{F,t}(\boldsymbol{x'}) + 1 \\
\ell_2(\boldsymbol{x'}) &= (\max_{i \neq t}(F(\boldsymbol{x'})_i) - F(\boldsymbol{x'})_t))^+ \\
\ell_3(\boldsymbol{x'}) &= softplus\left(\max_{i \neq t}(F(\boldsymbol{x'})_i) - F(\boldsymbol{x'})_t\right) - \log(2) \\
\ell_4(\boldsymbol{x'}) &= (0.5 - F(\boldsymbol{x'})_t)^+ \\
\ell_5(\boldsymbol{x'}) &= -\log(2F(\boldsymbol{x'})_t - 2) \\
\ell_6(\boldsymbol{x'}) &= (\max_{i \neq t}(Z(\boldsymbol{x'})_i) - Z(\boldsymbol{x'})_t)^+ \\
\ell_7(\boldsymbol{x'}) &= softplus\left(\max_{i \neq t} Z(\boldsymbol{x'})_i) - Z(\boldsymbol{x'})_t\right) - \log(2)
\end{aligned}$$

Note: The most effective loss appears to be ℓ_6. Please refer to Carlini and Wagner [4] for further information.

2.3.2 Adversarial training

In this section, we cover further the defense strategy related to adversarial training. The core idea is to train a model f on the original dataset (x, y) but also on perturbed variations of x. Over time, different perspectives have been used for this problem.

Regularization perspective Adversarial training was originally proposed by Goodfellow et al. [8]. In this paper, the perspective was about regularization. From the original setting for training logistic regression (see definition 4), authors derived an adversarial training setting (see definition 5).

Definition 4 (Original training of logistic regression)**.**

$$\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim p_{data}} \log(1 + e^{-y \cdot \boldsymbol{w}^T \boldsymbol{x}})$$

Note: In the original paper Goodfellow et al. [8], the bias term is present in the equation. By considering original weight $\boldsymbol{w}$ and bias b, we can rewrite $\boldsymbol{w}^T\boldsymbol{x} + b = \tilde{\boldsymbol{w}}^T\tilde{\boldsymbol{x}}$ where $\tilde{w} = (\boldsymbol{w}, b)$ and $\tilde{x} = (\boldsymbol{x}, 1)$.

$$
\begin{aligned}
\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), y) &= \nabla_{\boldsymbol{x}} \log(1 + e^{-y \cdot \boldsymbol{w}^T \boldsymbol{x}}) \\
&= -y\boldsymbol{w} e^{-y \cdot \boldsymbol{w}^T \boldsymbol{x}} \frac{1}{1 + e^{-y \cdot \boldsymbol{w}^T \boldsymbol{x}}} \\
\operatorname{sign}(\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), y)) &= \operatorname{sign}(-y\boldsymbol{w}) \quad as \ \frac{e^{-y \cdot \boldsymbol{w}^T \boldsymbol{x}}}{1 + e^{-y \cdot \boldsymbol{w}^T \boldsymbol{x}}} > 0 \\
&= -\operatorname{sign}(y\boldsymbol{w}) \\
&= -y * \operatorname{sign}(\boldsymbol{w}) \quad as \quad y \in \{-1, 1\}
\end{aligned} \tag{3}
$$

Let $f_{softplus}(\boldsymbol{x}) := \log(1 + e^{\boldsymbol{x}})$

$$
\begin{aligned}
\textit{Adversarial training objective} &:= \min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y) \sim p_{data}} f_{softplus}(-y \cdot \boldsymbol{w}^T (\boldsymbol{x} + \epsilon \operatorname{sign}(\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), y))) \\
&= \min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y) \sim p_{data}} f_{softplus}(-y \cdot \boldsymbol{w}^T (\boldsymbol{x} + \epsilon(-y * \operatorname{sign}(\boldsymbol{w}))) \quad \textit{using (3)} \\
&= \min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y) \sim p_{data}} f_{softplus}(-y \cdot \boldsymbol{w}^T \boldsymbol{x} + \epsilon y^2 \cdot \boldsymbol{w}^T \operatorname{sign}(\boldsymbol{w})) \\
&= \min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y) \sim p_{data}} f_{softplus}(-y\boldsymbol{w}^T \boldsymbol{x} + \epsilon \cdot \boldsymbol{w}^T \operatorname{sign}(\boldsymbol{w}))) \quad \textit{as } y^2 = 1 \\
&= \min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y) \sim p_{data}} f_{softplus}(-y\boldsymbol{w}^T \boldsymbol{x} + \epsilon ||\boldsymbol{w}||_1) \quad \textit{as } \boldsymbol{w}^T \operatorname{sign}(\boldsymbol{w})) = ||\boldsymbol{w}||_1 \\
&= \min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y) \sim p_{data}} f_{softplus}(\epsilon ||\boldsymbol{w}||_1 - y\boldsymbol{w}^T \boldsymbol{x}) \\
&= \min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y) \sim p_{data}} \log(1 + e^{\epsilon ||\boldsymbol{w}||_1 - y\boldsymbol{w}^T \boldsymbol{x}})
\end{aligned}
$$

Definition 5 (Adversarial training of logistic regression Goodfellow et al. [8])**.**

$$\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y) \sim p_{data}} \log(1 + e^{\epsilon ||\boldsymbol{w}||_1 - y\boldsymbol{w}^T \boldsymbol{x}})$$

From the derivation for the linear setting, Goodfellow et al. [8] derived this setting for deep neural network models.

Definition 6 (Adversarial training of deep neural network models Goodfellow et al. [8])**.** . $J(\theta, \boldsymbol{x}, y) := \ell(f_\theta(\boldsymbol{x}, y)$

$$\tilde{J}(\theta, \boldsymbol{x}, y) := \alpha J(\theta, \boldsymbol{x}, y) + (1 - \alpha) J(\theta, \boldsymbol{x} + \epsilon \operatorname{sign}(\nabla_{\boldsymbol{x}} J(\theta, \boldsymbol{x}, y)), y)$$

If $\alpha = 1$: Standard Empirical Risk Minimization.
If $\alpha = 0$: Minimization only on adversarial examples.

Note: In their experiments, Goodfellow et al. [8] used $\alpha = 0.5$ as their initial guess worked well. They didn't explore further tuning this hyperparameter.

Minimax perspective In Madry et al. [10], authors took a Minimax perspective and shown to improve the performance compared to Goodfellow et al. [8]. Madry et al. [10] will be presented on Friday 5th February 2021.

Note: As fun fact, Madry et al. [10] published their methods with 2 challenges, MNIST and CIFAR, to allow the community to attempt breaking their approach. For MNIST, no attack has fooled the classifier more than 9.5% points. (original model accuracy: 98.8%, accuracy with the best attack during the challenge: 89.3%). For CIFAR, the degradation is about 41% points (original model accuracy: 87.3%, accuracy with the best attack during the challenge: 45.8%). These two challenges are still considered are benchmarks in the community to test an new attack strategy.

2.3.3 Transferability of adversarial training

Originally introduced by Goodfellow et al. [8] with the terminology generalization of adversarial examples, the community has then used the terminology transferability of adversarial training.

The general idea is to attempt to fool a model A and to use this learning to another model B. This type of attacks is more realistic, as attackers may have a limited access to the target model B. In the paper Papernot et al. [12], which will be presented on Friday 5th February 2021, this concept is used to approximate the model B with a model A, and then learning to fool model A for an attack on model B.

In Moosavi-Dezfooli et al. [11], authors have shown the existence of fixed noise across all training inputs which can fool the targeted model. See figure 3 for an example.

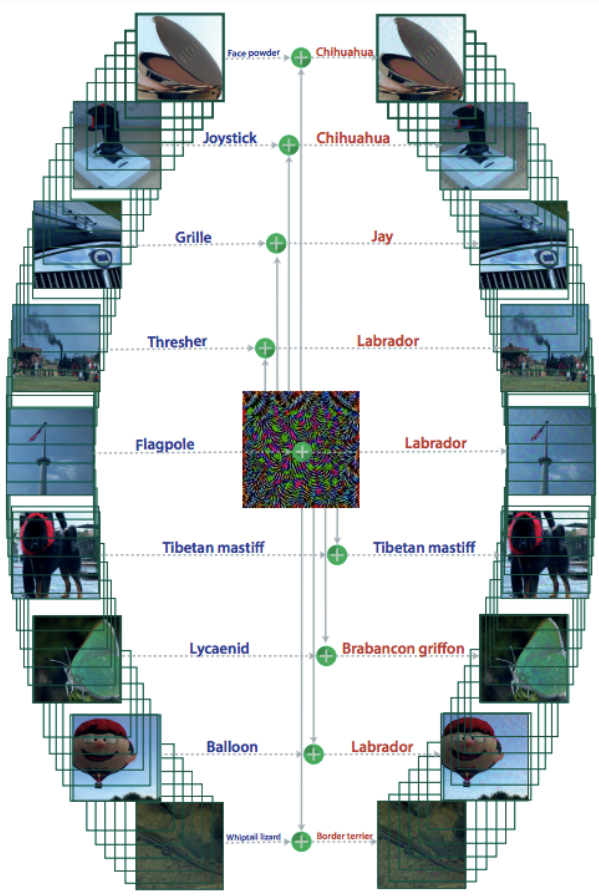

Figure 3: Universal adversarial perturbations (source Moosavi-Dezfooli et al. [11])
On the left, the predictions from the original images. In the middle, the fixed noise added to all original images. On the right, the new predictions from the perturbed images.

3 Discussions

1. Is the white box threat model is always better than the black box threat model?
   A: The answer is yes. This is because we have enough information about f to carry out the optimization. That said, this is not applicable in real world scenarios because of the strong assumptions it carrys.

2. Related to adversarial training for deep neural networks, see definition 6, has it been explored to use a stochastic α ?
   A: Probably, but if not, this will be a good project idea.

3. What is the intuition behind using negative momentum?

A: In optimization, we want to accelerate towards the optima and one way to do this is to have a momentum(positive usually) to ensure the constant push in the right direction. However in minimax optimization, we usually use "negative momentum" and the intuition is to slow down the oscillation near the optima. This is the active research area and could be a project on itself potentially.

4. Is there a relation between the definition 6 and the mix up paper (Zhang et al. [14]) ?
   A: Indeed, we could interpret definition 6 as training a model on a mix of the two images, the original one and the perturbed one.

5. Are GAN more robust to adversarial examples ?
   A: This question is more relevant to conditional GAN which generates samples from noise and target class, unlike traditional GAN which only uses noise. It's unclear if this is the case. It's probably a good project idea. Furthermore, a recent paper under review, More related to conditional GAN (because you need the condition), probably it has been done before. If not project. Furthermore, a recent paper Chen et al. [5], to be presented at ICLR 2021, indicates that a model with a more structured output (ex: soundwave) is more robust. Intuitively, this is due to the optimization problem which is harder in structured output, rather than the categorical labels.

6. Can you find a adversarial examples which are universal to all models ?
   A: In the paper Ilyas et al. [9], authors argue some adversarial examples are non robust features, which is more a characteristic of datasets than models.

References

[1] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, 2018.

[2] S. Bhambri, S. Muku, A. Tulasi, and A. B. Buduru. A survey of black-box adversarial attacks on computer vision models, 2020.

[3] A. J. Bose, G. Gidel, H. Berard, A. Cianflone, P. Vincent, S. Lacoste-Julien, and W. L. Hamilton. Adversarial example games, 2021.

[4] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks, 2017.

[5] B. Chen, Y. Li, S. Raghupathi, and H. Lipson. Beyond categorical label representations for image classification. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=MyHwDabUHZm.

[6] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. Zoo. *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, Nov 2017. doi: 10.1145/3128572.3140448. URL http://dx.doi.org/10.1145/3128572.3140448.

[7] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum, 2018.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples, 2015.

[9] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features, 2019.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[11] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations, 2017.

[12] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning, 2017.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions, 2014.

[14] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.