1 Adversarial Examples

Source Paper
Explaining and Harnessing Adversarial Examples. (wyjaśnienie i wykorzystanie przykładów adwersalnych) *ICLR, 2015.*

1.1 What is motivation of adversal examples?

Machine learning models, especially deep neural networks, have achieved remarkable (niezwykłą) accuracy in many domains. However, **adversarial examples** demonstrate a surprising vulnerability of (podatność) these models: tiny, carefully chosen (starannie dobrane) perturbations (zakłócenie) to an input can cause confident misclassifications (błędnych klasyfikacji z wysokim stopniem pewności). Importantly (co istotne), these adversarial perturbations are often *unnoticeable (niewidoczne) or nearly unnoticeable* to the human eye.

This phenomenon is intriguing (intrygujące) because it reveals (ujawnia) that high-performing models do not necessarily learn robust (solidny), human-like concepts of their input data.

In this lab, we will focus on a single, seminal (przełomowy) paper:

> *Explaining and Harnessing Adversarial Examples*
> Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy. *ICLR, 2015.*

1.2 Why This Paper?

1. It introduced a clear explanation of why adversarial examples exist, attributing (przypisując) the phenomenon (to zjawisko) largely to (w dużej mierze) the near-linear behavior of modern networks in high-dimensional input space.

2. It proposed an efficient and straightforward method — the **Fast Gradient Sign Method (FGSM)** — to generate adversarial examples. This method leverages (wykorzystuje) the gradient of the loss function with respect to the input (względem danych wejściowych) to create minimal perturbations that cause misclassification. FGSM demonstrates how optimization techniques can be applied beyond (nie tylko do) model training, serving as (służące jako) versatile tools (uniwersalne narzędzia) in a machine learning researcher's arsenal.

Because we are focusing on a *single paper lab*, our goal is to understand **only** these core ideas and replicate some simplified (uproszczony) experiments without delving into (zagłębienia się) other adversarial attack or defense strategies proposed later in the literature.

2 Core Concept: Adversarial Examples

2.1 Provide the definition

An **adversarial example** is an input that has been *intentionally perturbed* so that a target model (model docelowy) misclassifies it, typically with **very high confidence**. The perturbation is usually small and is often visually (or otherwise (w inny sposób)) difficult for humans to detect.

In mathematical terms (w ujęciu matematycznym), let (przyjmijmy):

- x be a “clean” input (e.g., an original image),
- y be the correct label of x,
- θ be the parameters of the model $f_\theta(\cdot)$.

An adversarial example x_{adv} is created by applying a small perturbation η to x:

$$x_{\text{adv}} = x + \eta$$

such that (tak aby) the model predicts the wrong label:

$$\arg\max f_\theta(x_{\text{adv}}) \neq y$$

while keeping the perturbation η small enough.

2.2 Linear Explanation in High-Dimensional Spaces

A key insight (kluczowa obserwacja) of Goodfellow *et al.* is that nonlinear neural networks often behave *locally* in ways (sposób) that are close to linear. Even more crucially (co jeszcze ważniejsze), high-dimensional spaces allow (pozwalają) many small, correlated changes (skolerowanych zmian) to add up (sumowało się) — leading to a significant change in the final classification output. In other words (innymi słowy), while each individual pixel may be altered by (zmodyfikowany) only a tiny amount (małą wartość) (small enough to be imperceptible (niezauważalna)), those changes, when combined in (połączone z) a specific direction aligned with (zgodnym z) the gradients of the model, effectively push the decision score toward a wrong class.

3 Describe Fast Gradient Sign Method (FGSM)

One of the paper's main practical contributions (praktycznych wkładów) is a fast, simple method to generate adversarial examples. The method is based on the gradient of the loss function with respect to (względem) the input:

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon \cdot \text{sign}\left(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y)\right)$$

where:

- $J(\theta, \mathbf{x}, y)$ is the training loss (e.g., cross-entropy),
- $\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y)$ is the gradient of that loss with respect to the input,
- $\text{sign}(\cdot)$ is the element-wise (element po elemencie) sign function,
- ϵ is a small scalar controlling the amount of perturbation.

This simple, single-step approach (jednokrokowe podejście) often suffices (wystarcza) to produce (wywołać) high-confidence misclassifications on neural networks that have been trained in a "standard" (non-adversarial) manner (sposób).

Recommended Reading: Neural Network Fundamentals

If you need to refresh your understanding of neural networks before diving into adversarial examples, here are some excellent resources:

1. **PyTorch Quickstart Tutorial** — A practical introduction to implementing neural networks with PyTorch.
2. **Micrograd** by Andrej Karpathy — A minimal neural network library built from scratch. Accompanied by an explanatory video that walks through the implementation.
3. **3Blue1Brown's Neural Network Series** — An intuitive visual explanation of neural networks — But what is a neural network? (Part 1 of the series).