Exercise 8-1  Generative Adversarial Networks (GANs)

- The loss for $D$ is given as:

$$
J^{(D)} = -\frac{1}{2} \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] - \frac{1}{2} \mathbb{E}_{z \sim p_{z}(z)}[\log(1 - D(G(z)))].
$$

Explain the terms in this loss!

- The generator tries to fool the discriminator, so its loss can be defined as:

$$
J^{(G)} = -J^{(D)}
$$

Write down this optimization problem as a minimax game!

- Why might the usage of $J^{(G)} = -J^{(D)}$ as loss for the generator lead to slow learning? *Hint: What happens to the gradient of the losses if $D(G(z))$ is small? (No calculation required)*

Exercise 8-2  Optimal Solution of a GAN

Recall from the lecture the idea of Generative Adversarial Networks (GANs):

A generator $g$ is trained to produce samples $x = g(h; \theta_g)$. The idea is that the generator performs especially well, i.e. closely approximates the real data distribution $p_{data}(x)$, if a discriminator $d(x; \theta_d)$ is not able to separate the samples produced by $g$ from real samples from $p_{data}(x)$. Both, generator and discriminator, are usually multi-layer perceptrons, such that the $\theta_g$ and $\theta_d$ represent their respective weights. We will omit these parameters in the following for brevity. In a GAN, both networks are trained jointly to perform their respective tasks as good as possible:

$$(\theta_g^*, \theta_d^*) = \arg\min_{\theta_g} \arg\max_{\theta_d} V(g, d)$$

where

$$
V(g, d) = \mathbb{E}_{x \sim p_{data}}[\log d(x)] + \mathbb{E}_{h \sim p_h}[\log(1 - d(g(h)))]
$$

Intuitively, the discriminator attempts to classify the samples correctly as real or fake by maximizing the negative cross-entropy. The generator on the other hand attempts to minimize the negative cross-entropy with the goal of fooling the discriminator into believing that its generated samples are real.

In the following, we want to proof the following statement:

At the unique global optimum, $p_g(x) = p_{data}(x)$ and $d(x) = 1/2$ everywhere.
(a) Given a fixed $g$, what is the optimal discriminator $d^*_g$?

Hint: We can write $V$ as

$$V(g, d) = \int_x p_{\text{data}}(x) \log d(x) dx + \int_h p_h(h) \log (1 - d(g(h))) dh$$

$$= \int_x p_{\text{data}}(x) \log d(x) + p_g(x) \log (1 - d(x)) dx$$

Further, for any $a, b \in \mathbb{R}, a + b > 0$, the function

$$y \rightarrow a \log y + b \log (1 - y)$$

attains its maximum in $[0, 1]$ at $\frac{a}{a+b}$.

(b) Given a fixed generator $g^*$ such that $p_{g^*} = p_{\text{data}}$, what is the optimal discriminator $d^*$? What is the value of $V(g^*, d^*)$?

Exercise 8-3  PyTorch - Variational AutoEncoder

On the course website you find an ipython notebook leading you through the implementation of a Variational AutoEncoder (VAE) in PyTorch. The VAE is learned on the MNIST dataset and as a generative model is able to output self-generated digits.