

# Homework 1 : Unsupervised Deep Learning

João Ramos da Silva Tabuaço Freitas  
Università Degli Studi di Padova  
Student ID: 2009440

18 July 2022

## Introduction

Unsupervised learning takes an agnostic approach in learning features in data. This distinguishes it from supervised learning, where models attempt to form a connection between an input and a ground-truth target. The performance of unsupervised learning models is normally assessed by how faithfully the data can be reproduced from the features learned.

**Autoencoders** and **generative-adversarial networks** (GANs) are two models used in unsupervised learning. In both cases, these models are composed by a perception module along with a reproduction module. These two work in tandem to achieve faithful reproduction of data with large number of features.

The goal of the autoencoder is to study inputs and tune itself to embed input data in a latent space of reduced dimensionality, and subsequently reproduce it. To achieve this, an encoder transforms an input into a single vector in said latent space. A decoder then takes in the latent vector output of the encoder and attempts to retrieve the initial input as faithfully as possible. This motivates the use of mean-squared error (MSE) to quantify the deviation, with the ultimate goal of minimizing it.

Autoencoders may be undercomplete or overcomplete, depending on whether they have a smaller, or larger number of hidden neurons than the input dimensions. The former performs dimensionality reduction, by learning only data's most obvious features. This proves useful in denoising input, for example. While the latter has the capability of capturing larger detail from the inputs, its learning converges to the identity. In a denoising scenario, this means the output will be just as noisy as the input. This is a disadvantage of using deviation from input as a criterion.

An undercomplete autoencoder essentially learns to perform dimensionality reduction. Thus, its performance can be compared with principal component analysis (PCA), or t-distributed stochastic neighbor embedding (t-SNE). In fact, an autoencoder without non-linearities is expected to learn the structure of a PCA subspace. Naturally, non-linearity

aids in capturing a more diverse set of features, which arguably endows the autoencoder with a larger descriptive power. To test this, its performance may be compared to t-SNE, which is well-established as a standard for performing dimensionality reduction, also being non-linear.

The structure of a GAN differs slightly from that of an autoencoder. It starts with a generator which produces samples  $\mathbf{x} = g(\mathbf{z}, w_g)$  from a randomly generated latent vector  $\mathbf{z}$ , where  $w_g$  are the generator parameters. These samples are then fed into the discriminator, whose goal is to observe an input, and reach a conclusion  $y = d(\mathbf{x}, w_d) = d(g(\mathbf{z}, w_g), w_d)$  about said input (with  $w_d$  being the discriminator parameters). The learning task can be formulated as a zero-sum game solved by a minimax decision.<sup>1</sup> The discriminator observes a payoff  $v(g, d)$ , while the generator receives  $-v(g, d)$ . Thus, the goal is to find  $g^*$  such that:

$$g^* = \arg \min_g \max_d v(g, d) \quad (1)$$

Normally, the choice of  $v(g, d)$  is:

$$v_{\text{mm}}(g, d) = -\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}}[\log(d(\mathbf{x}))] - \mathbb{E}_{\mathbf{x} \sim p_g}[\log(1 - d(\mathbf{x}))] \quad (2)$$

In other words, the aim of the minimax game is for the discriminator to maximize its payoff by classifying fake/real inputs correctly, while the generator attempts to minimize the discriminator's payoff by tricking it into believing that the inputs are real.

One issue with the minimax game formulation is that the gradient quickly diverges away from  $d(\mathbf{x}) = 1/2$ . This can be addressed by modifying the generator loss, as is done in non-saturating GAN (NSGAN) formulation.<sup>2</sup> In this scenario, the discriminator loss is  $v_{\text{mm}}$ , while the generator loss is:

$$v_{\text{NS}}(g, d) = -\mathbb{E}_{\mathbf{x} \sim p_g}[\log(d(\mathbf{x}))] \quad (3)$$

Both the minimax and the non-saturating formulations will be explored.

## 1 Methods

The autoencoder task consists in minimizing the MSE between input and output.

$$\{\hat{w}\} = \arg \min_w \frac{1}{n} \sum_{i=1}^n (\hat{\mathbf{x}}_i - \mathbf{x}_i)^2 \quad (4)$$

The model design consists of three convolutional layers, followed by an unflatten layer, and two fully connected layers. The last layers

---

<sup>1</sup>Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014c). Generative adversarial networks. In NIPS'2014.

<sup>2</sup>Lucic, M., Kurach, K., Michalski, M., Bousquet, O., Gelly, S., (2018). Are GANs Created Equal? A Large-Scale Study. In NIPS'2018.

## 2 Results