MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

Deep learning-based synthesis of brain fMRI images using a diffusion probabilistic approach

Scientific Students' Association Report

Author:

Levente Oszvald

Advisor:

Dr. Luca Szegletes
Szabolcs Torma

2023

Contents

Kivonat

A modern orvoslásban a funkcionális MRI (fMRI) vizsgálatok az egyik legnépszerűbb képalkotó technológiának számítanak. A mérések az idegrendszeri aktivitással összefüggésben lévő választ mérik az emberi agyban, amellyel képesek a kutatók elváltozásokat észlelni, betegségeket diagnosztizálni és ezeket kezelni. Ennek azonban több kihívása is akad, többek között rendkívül erőforrásigényes, illetve a páciens számára megterhelő. Ennek okán megnő az igény különböző adatdúsító módszerek fejlesztésére.

Napjainkban a generatív modellezés, ha erről kifejezetten nem is szerzünk tudomást, sok helyen jelen van. Legyen szó szintetikus zenékről, politikusok szájába adott mondatokról, vagy akár olyan képekről, amelyeken nem létező emberek mosolyognak, ez a technológia szerves részét képzi, és fogja képezni életünknek az elkövetkező években. Mint oly sok területre, az orvosi képalkotáshoz is elért a hullám, ahol is több, addig megoldatlan feladatra is választ jelenthet.

Dolgozatomban a generatív modellezés családjának egyik legkorszerűbb tagját, a diffúziós valószínüségi modelleket használom fel valósághű fMRI jelek generálására. A munkámban kitérek különböző olyan módszerekre is, amelyekkel a mintavételezést kondícionálom, illetve írányítom bizonyos tulajdonságok (pl. osztályhűség, adathűség) elérése érdekében. A modelleket a látens térben alkalmazom, ezzel csökkentve a futási időt és a hardverigényt, így növelve a hatékonyságot. A szintetikus adatok minőségét az irodalomban gyakran alkalmazott metrikák segítségével kvantitatív, továbbá kvalitatív módon értékelem ki és elemzem.

Munkám eredményeként bemutatom, hogy a diffúziós generálással lehetséges élethű fMRI jelek előállítása zajmintákból, és ezek a modellek képesek ezen adatok komplex karakterisztikájának megtanulására.

Abstract

In modern medicine, functional MRI (fMRI) is one of the most popular imaging technologies. This technology measures the response associated with neural activity in the human brain, enabling researchers to detect lesions, diagnose diseases and treat them. However, it has several challenges, including being extremely resource-intensive and stressful for the patient. This increases the need to develop different data augmentation methods in this domain.

Today, generative modeling, even if not explicitly known, is present in many areas. Whether it is synthetic music, sentences spoken by politicians, or even pictures of non-existent people smiling, this technology is, and will be an integral part of our lives for years to come. As in so many areas, the wave has reached medical imaging, where it could provide solutions to a number of previously unsolved problems.

In my work, I apply diffusion probabilistic models, a state-of-the-art member of the generative modelling family, to generate realistic fMRI signals. I also explore various methods to condition and control the sampling in order to achieve certain properties (e.g. class fidelity, data fidelity). The models operate in the latent space, reducing runtime and hardware requirements, thus increasing efficiency. The quality assessment of the synthetic data involves both quantitative analysis, utilizing metrics commonly employed in the literature, as well as qualitative evaluation.

As a result of my work, I show that it is possible to generate lifelike fMRI signals from noise samples using diffusion generation, and that these models are capable of learning the complex characteristics of these data.

Chapter 1

Introduction

During the course of my work, I dive into the realm of generative modeling, an innovative and emerging technology that has garnered significant attention in recent years. This cutting-edge approach has been driven by the latest scientific advancements, and I explore its application across various data modalities while evaluating the outcomes achieved.

To embark upon this journey, it is necessary to provide a concise yet comprehensive introduction to the field of generative modeling. This introductory section sets the stage for understanding the fundamental principles and key concepts that underpin this specialized area of research. By capturing the essence of generative modeling, we can appreciate its huge potential in generating synthetic data that exhibits striking resemblance to real-world samples.

In recent years, deep learning-based generative modeling has brought a paradigm shift across multiple domains of multimedia, encompassing areas such as speech recognition and text generation. This revolutionary technology has also significantly impacted the realms of video and image generation, reshaping the methods by which we produce and comprehend visual and textual content. However, in the scope of this paper, my focus remains primarily on the exploration of image generation techniques. Through the power of deep learning, generative models have the ability to learn intricate patterns and generate novel, realistic images that resemble those from the real world. This transformative technology has found applications not only in creative fields such as art and design but also in scientific research, including the realm of brain imagery. Brain imaging technologies, such as functional magnetic resonance imaging (fMRI) and electroencephalography (EEG), provide crucial insights into the functioning and structure of the human brain. However, the interpretation and analysis of brain imaging data pose significant challenges due to their complex and high-dimensional nature.

Deep generative models, such as Variational Autoencoders (VAEs)[23] and Generative Adversarial Networks (GANs)[16] have revolutionized the field by leveraging the power of neural networks to model and generate images. These models learn from vast amounts of training data, discovering underlying patterns and features that enable them to generate compelling and diverse visual content.

Recently, new techniques and methods have emerged in the domain of image generation. One such intriguing category is referred to as Diffusion Models, which harness the principles of diffusion processes derived from the fields of statistics and physics. Within this domain, various ideas have emerged, including the Denoising Diffusion Probabilistic Models (DDPMs)[19]. These models have demonstrated exceptional performance, achieving state-of-the-art metric results in generative competitions focused on the CIFAR10 dataset.

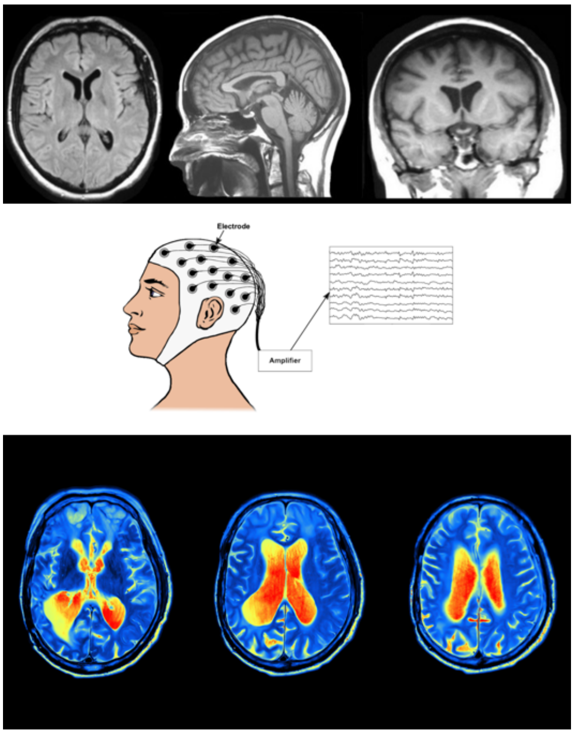

Figure 1.1: MRI[7], EEG[1] and fMRI[3] signals from top to bottom

Building upon this progress, subsequent advancements have been made to enhance sampling and generation efficiency, as demonstrated by the Denoising Diffusion Implicit Models paper[33]. Furthermore, another notable development in this field is the introduction of Stable Diffusion[28], which garnered considerable attention in the past year. These recent advancements showcase the continuous evolution and potential of Diffusion Models in the context of image generation.

Shifting to brain imagery, there have been scientific papers released about the synthesis between EEG and fMRI signals using generative adversarial networks and Autoencoders (AE)[11], as well as about EEG generation with DDPMs [36], however, it is important to note that a widely accepted standard for brain imagery generation has not yet been established in this field.

A comprehensive evaluation is essential when assessing the quality of generated content, regardless of the specific domain, both qualitatively and quantitatively speaking. For such reasons, several techniques have been established as the universal standard for such cases. Two prominent metrics are the Inception Score (IS) [31] - which utilizes the so called InceptionNet [35] model to evaluate the quality of the generated images - and the Frechet Inception Distance (FID)[18] - which measures the similarity between the generated and real images by leveraging the latent features of the InceptionNet model. Additionally, I have employed a self-supervised contrastive learning-based [13] evaluation on the real and generated fMRI samples to ascertain whether the generated images are embedded similarly to real images within the latent space. This method aims to determine the fidelity of the generated samples by examining their proximity to real samples. I thoroughly investigate the effectiveness of the mentioned metrics to extensively assess the quality and fidelity of the generated content.

Within my thesis, I propose an approach to generate high-quality fMRI signals using deep learning approaches, such as the above mentioned DDPMs, and encoders. With my solution, fMRI signals conditioned on features derived from the datasets are also produced.

Section 2 introduces the fundamental principles of my method, offering an in-depth exploration of the diffusion process, generative modeling, the denoising diffusion model and its sampling techniques. Moving forward, Section 3 provides a concise overview of the dataset utilized in the experiment, encompassing information on acquisition and preprocessing. Subsequently, in Section 4, I present the proposed image generation method, outlining the network training process, with the neural network architecture included. Furthermore, in Section 5, I present a detailed analysis of the results obtained, along with the evaluation metrics - such as contrastive learning - and corresponding scores achieved. Finally, Section 6 concludes with conclusions drawn from the findings, and possible future directions for further advancements.

Chapter 2

Background

2.1 Medical imaging

Medical Imaging is a significant field in the realm of healthcare that has revolutionized the way we analyze and treat various medical conditions. This field includes a diverse range of technologies and techniques designed to visualize the internal structures of the human body, providing invaluable insights into the functioning and abnormalities of some organs or tissues hidden by the skin and bones. Although imaging of removed organs and tissues can be performed for medical reasons, such procedures are usually considered part of pathology instead of medical imaging.

Recently, the advancements in noninvasive[1] techniques have further expanded the horizons of medical imaging. These noninvasive methods hold a valuable advantage in healthcare by eliminating the need for invasive procedures, thus reducing risks, discomfort, and recovery times.

2.1.1 MRI

Magnetic Resonance Imaging [14], also known as MRI, is one of the most widely employed noninvasive medical imaging techniques nowadays. It utilizes powerful magnetic fields and radio waves to produce detailed, high-resolution images of the body's organs and tissues. This method does not use ionizing radiation, thus being safer than other well established imaging modalities, like computed tomography (CT) or positron emission tomography (PET) scans[30].

In contrast to the aforementioned methods, MRI offers improved contrast in soft tissue images. Nevertheless, it might be less comfortable for patients due to lengthier scan times, noisier environments, and less convenient positioning. This positioning is partly due to the design of the MRI scanner (shown in 2.1). The machine has three major components, the outer most being a magnet, which produces a strong homogeneous magnetic field. This magnetic field is about 10000 times the earth's magnetic field. The middle component is the gradient coils, which localizes the radio frequency (RF) signal in three dimensional space. The inner most part is the RF coils, which sends and receives the RF signal to and from the organ or tissue.

[1] The term "noninvasive" refers to medical procedures or techniques that do not require the penetration of the skin or the body's natural barriers, such as the skin

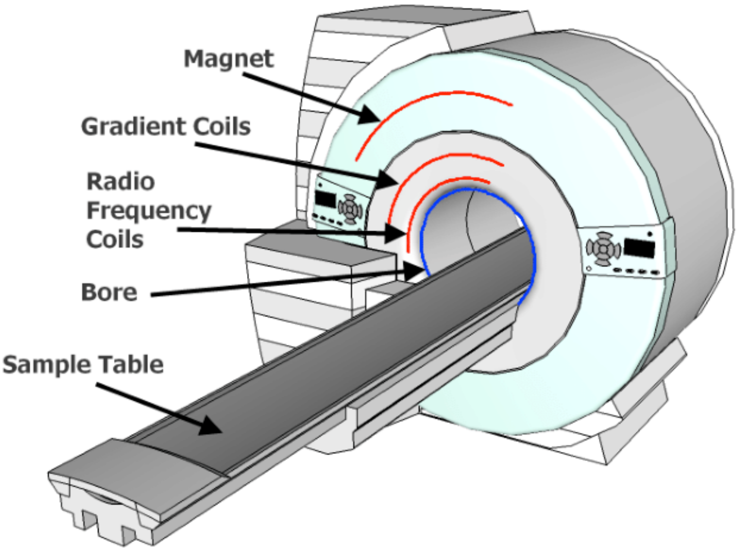

Figure 2.1: Architecture of a MRI machine[6]

Mechanism Certain atomic nuclei are able to absorb and re-emit radiofrequency energy when placed in a magnetic field, such as the one inside the MRI scanner. During measurement, nuclei - mostly hydrogen nuclei, protons - inside a person's body tend to behave like this.

This magnetic field causes the protons in the hydrogen atoms of the body's tissues to align themselves with the magnetic field's direction. After this, radiofrequency (RF) pulses are sent into the body, which leads to the protons "flipping out" of their magnetic alignment. Following this, the nuclei gradually return to their original alignment, emitting radiofrequency signals in the process. Figure 2.2 visualizes the aforementioned steps.

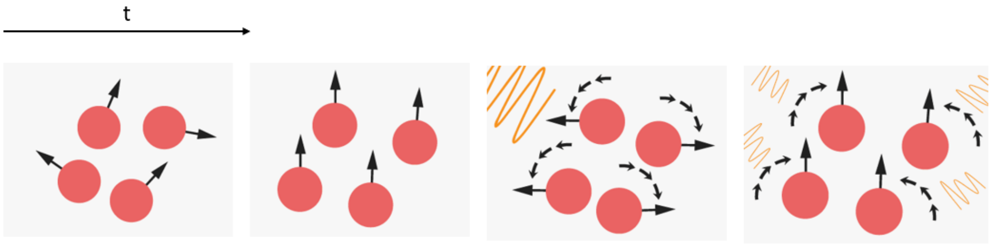

Figure 2.2: Steps of signal detection in MRI. From left to right: no magnetic field, magnetic realignment, RF pulse and flip, realignment with magnetic field, release of RF pulse [9]

Different tissues in the body produce different signals, resulting in contrasts in the created images. These differences originate from two key factors, T1 and T2 relaxation. T2 relaxation refers to the time it takes for the protons in tissues to return to their equilibrium state - meaning the loss of coherence among protons within the same tissue - after being flipped out by RF pulses. On the other hand, T1 relaxation is the process by which the protons return to their equilibrium alignment with the main magnetic field after the flip.

By adjusting various parameters, MRI technicians have the ability to create MRI images of the examined body part with different contrasts (Figure 2.3. This is done by changing the timing and characteristics of radiofrequency pulses and the time intervals between them. One of these parameters is “Echo Time” (or TE), which corresponds to the time the MRI technician waits to detect the signals after the nuclei have been knocked out of alignment with the field. The other significant parameter is called the “Repetition Time” (or TR), which refers to the time interval between successive radiofrequency pulses during an MRI scan.

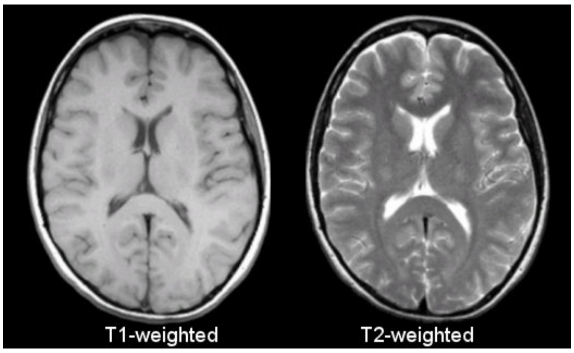

Figure 2.3: T1- and T2-weighted MRI images [7]

The choice of shorter TE and TR parameters results in emphasizing the contrast between tissues with different T1 relaxation times. In such cases, tissues with extended T1 relaxation times appear brighter, while those with shorter T1 relaxation times appear darker. This imaging technique is called T1-weighted MRI.

Likewise, longer TE and TR enhances the contrast between tissues with different T2 relaxation times. In this setup, tissues with longer T2 relaxation times appear brighter, while those with shorter T2 relaxation times appear darker. This is characteristic of T2-weighted images.

T1-weighted images are often used for anatomical imaging and for highlighting the boundaries between different tissue types, while T2-weighted images are valuable for detecting abnormalities involving fluid content, such as inflammation, edema, or lesions.

Since the 80s, MRI has established itself as a versatile imaging technique, it is widely used in hospitals all around the world, having roughly more than 50 000 machines in operation [8].

2.1.2 fMRI

Functional Magnetic Resonance Imaging (often referred to as fMRI) extends the use-cases of MRI by incorporating brain activity measurements. This technique has, in less than two decades, become the most commonly used method for the study of human brain function [30].

When neurons in the brain become active - as a consequence of some physical or mental task - local blood flow through that area is increased. This activity related increase in blood flow leads to a relative surplus in local blood oxygen. The signal, which is measured in fMRI depends on this change, and is called Blood Oxygenation Level Dependent, or

BOLD (Figure 2.5). This is a type of specialized brain scan used to map the neural activity in the brain by imaging the change in blood flow that follows a brief period of neuronal activity. This phenomenon is also recognized as the hemodynamic response [30], which exhibits exciting characteristics. It is very slow, taking approximately 5 seconds to reach its peak, and features an even slower undershoot phase (Figure 2.4).

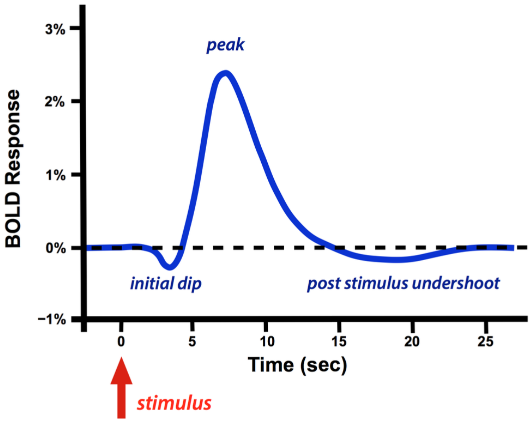

Figure 2.4: Hemodynamic response [5]

Moreover, fMRI is also able to measure resting state, which helps in identifying the patient's baseline BOLD activity [25]. It follows a similar approach to Positron Emission Tomography (PET) but surpasses it by being non-radioactive, faster, and more widely accessible.

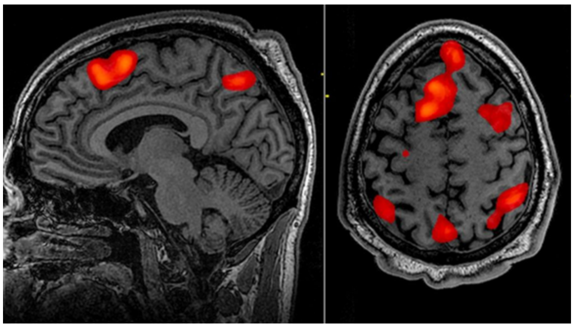

Figure 2.5: fMRI signals[4]

Imaging The analysis of fMRI data includes the manipulation and processing of images. An fMRI image is constructed from voxels, which are three dimensional pixels, having a third, Z axis along with the familiar X and Y axis. Thus, when assembled, in an fMRI image X represents the left–right dimension, Y represents the anterior–posterior dimension, and Z represents the inferior–superior dimension.

While (structural) MRI images are displayed as three-dimensional matrices (Figure 2.6), fMRI data have an additional fourth, time axis, making it a time series of three dimensional images. This four-dimensional matrix is usually stored in a single entity. These images are most commonly stored as unsigned 16-bit values, meaning that they can take integer values from 0 to 65535.

Finally, the problem of having different kind of shapes, and sizes of brain images introduced a common space in which different individuals can be aligned, creating a standardized

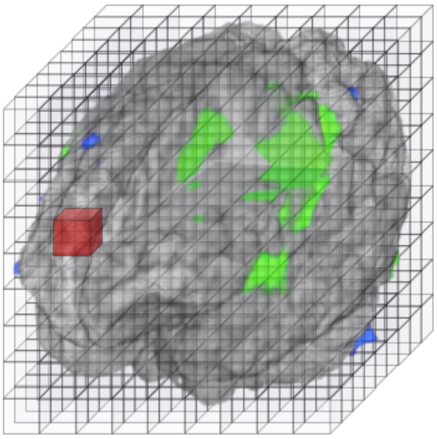

Figure 2.6: 3-dimensional, matrix nature of the data

database of samples. The most well known of these is the approach developed by Jean Talairach [24], which method was also applied during my experiments.

2.2 Deep learning

Deep learning has emerged as a groundbreaking field within the broader domain of machine learning, revolutionizing the way computers learn and perform complex tasks[10]. This powerful approach to artificial intelligence is inspired by the structure and functioning of the human brain, specifically the interconnected network of neurons. By utilizing deep neural networks with multiple layers of interconnected nodes, deep learning algorithms can automatically learn hierarchical representations of data, enabling them to effectively extract intricate patterns, features, and relationships from large and complex datasets. Deep learning is a subset of both machine learning and artificial intelligence, while building on the foundation of both, it introduces additional capabilities.

One of the key advantages of deep learning lies in its ability to handle raw, unstructured data, such as images, text, and audio, without the need for explicit feature engineering. Instead, deep neural networks can automatically learn relevant features directly from the data, eliminating the need for manual feature extraction and significantly reducing the reliance on domain-specific knowledge.

Furthermore, deep learning has demonstrated remarkable performance in a wide range of applications, including computer vision, natural language processing and speech recognition. With its remarkable capacity to learn from vast amounts of data and uncover complex patterns, deep learning has achieved breakthroughs in tasks such as image classification, object detection, machine translation, and voice synthesis, pushing the boundaries of what was previously thought possible.

The fundamental framework powering this entire approach is called the neural network.

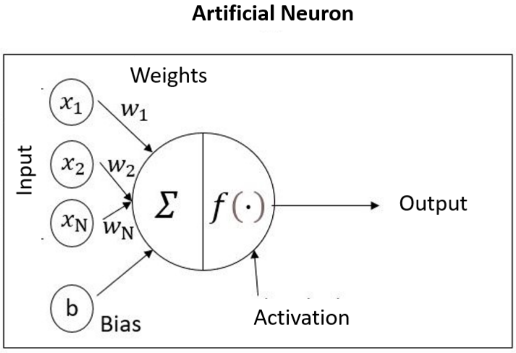

Figure 2.7: Architecture of one artificial neuron

2.2.1 Neural Network

A neural network is a powerful computational model inspired by the structure and functioning of the human brain. It is composed of interconnected nodes, known as neurons, which work collaboratively to process and analyze complex patterns in data. Each neuron receives input signals, performs calculations, and generates an output signal that is passed on to other neurons [40].

The connection between biological and artificial neurons lies at the fundamental principles of information processing. Biological neurons receive electrical signals from other neurons through specialized structures called dendrites. These signals are integrated within the neuron and, if the accumulated input reaches a certain threshold, an electrical signal known as an action potential is generated and transmitted through the axon to other connected neurons. This process forms the basis of communication and information flow in the brain.

Similarly, artificial neurons in neural networks receive inputs from other neurons or external sources (in the first "layer"). Each artificial neuron applies a mathematical function to the weighted sum of its inputs, determining whether it should produce an output signal or "fire". The weights assigned to the inputs play a crucial role in controlling the strength of the connections between neurons.

By looking at 2.7, one can inspect the elements of one neuron. As mentioned the weighted inputs and the bias term are summed up, and then fed through an activation function to produce the output of the neuron. This bias term acts similarly to the intercept term in linear regression. It allows the neuron to adjust its output independently of the input values. This helps the network to learn and represent patterns and relationships that are not strictly dependent on the input values alone, resulting in better outcomes. The activation function helps in introducing non-linearity to the system, which is essential for solving complex problems, like finding the connection between non-linear input variables. There are several types of activation functions commonly used in neural networks, each with its own characteristics and suitability for different tasks. Some of the popular activation functions are *Sigmoid*, *Rectified Linear Unit (ReLU)* or *Softmax*, each with it's own advantages and use-cases.

By organizing these neurons into layers and leveraging mathematical algorithms, neural networks are capable of learning from data and making accurate predictions or decisions. A simple neural network is depicted on figure 2.8. The strength of a neural network lies in its ability to automatically extract relevant features from raw data, enabling it to tackle a

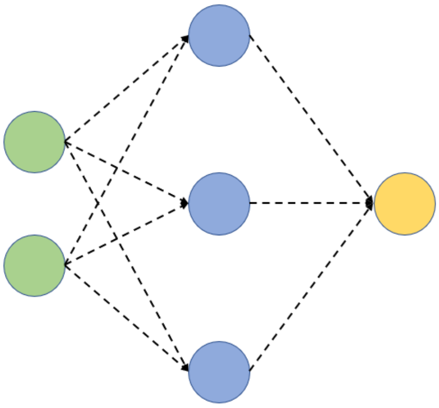

Figure 2.8: A simple network with two inputs and one hidden layer

wide range of tasks such as image recognition, natural language processing, and predictive modeling.

2.2.1.1 Operation of the network

A fully functional neural network comprises two essential components: the feedforward pass and the backward pass. The feedforward pass involves the propagation of input data through the network, starting from the input layer and proceeding to the output layer. At each neuron, the weighted sum of inputs is computed, followed by the application of an activation function, which generates the neuron's activation value. This process continues until the network produces its final output. This output typically represents a numerical variable, such as an image matrix, a vector, or a single value. The produced output is then compared with the desired output, leading to the calculation of an error or discrepancy between the two.

This error is then propagated backwards in the network, this is called the backward pass. This component of the network computes the gradients of the error with respect to the weights in each layer (2.1) and uses these gradients to adjust the weights in a way that reduces the error. Adjusting the weights can happen in different manners also, but this in not in the scope right now.

$$\frac{\delta error}{\delta weight_i} \tag{2.1}$$

Upon examining the revised neural network illustrated in Figure 2.9, one can observe the backward propagation of the error rate. The iterative process - which involves the feedforward and backward pass in succession - is called the *training* of the neural network. During this training phase, the forward pass calculates the output of the network, while the backward pass tries to minimize the error coming from this output by updating the weights of the system accordingly. This process continues until the network's performance reaches a satisfactory level. Once the network's performance is sufficient, the prediction of the system can be initiated on yet unseen input data.

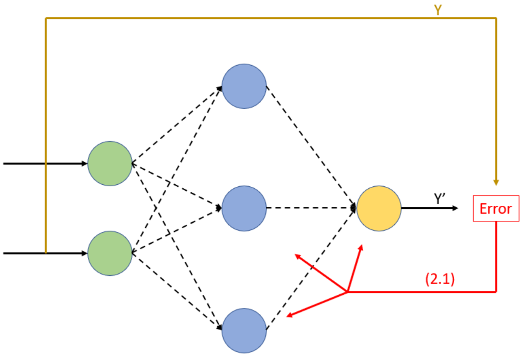

Figure 2.9: A simple network with backpropagation

The feedforward pass, and its complementary, the backward pass is also dependant on the architecture of the network, which can easily differ from the proposed simple architecture in 2.8.

2.2.1.2 Common architectures

Neural networks can take various architectural forms, each designed to address specific problem domains and achieve specific learning objectives. Here, I shortly present an overview of some commonly used neural network architectures.

Feedforward Neural Networks, also known as Multi-Layer Perceptrons (MLPs)[27], are the simplest and most widely used neural network architecture. They consist of an input layer, one or more hidden layers, and an output layer. Information flows only in one direction, from the input layer through the hidden layers to the output layer, without any loops or feedback connections.

Convolutional Neural Networks (CNN)[26] are primarily employed for analyzing visual data, such as images. They leverage convolutional layers, pooling layers, and fully connected layers to extract meaningful features hierarchically from input data. CNNs are known for their ability to capture spatial relationships and translational invariance, making them well-suited for tasks like image classification, object detection, and image segmentation. Other kinds of architectures can also be formulated based on convolutional layers. One such network is called the U-net architecture, which I will cover in more details.

Recurrent Neural Networks (RNN)[32] are designed to handle sequential data by introducing recurrent connections within the network. This architecture allows information to be processed not only based on current inputs but also on previous inputs and internal states. RNNs exhibit temporal dynamics, making them suitable for tasks such as speech recognition, language modeling, and sequence generation.

Generative Adversarial Networks (GAN) [16] consist of two neural networks, a generator and a discriminator. The generator aims to generate realistic data samples, such as images, while the discriminator attempts to distinguish between real and generated samples. Through an adversarial training process, GANs can learn to generate high-quality synthetic data that closely resembles the training data distribution. GANs have found applications in image synthesis, style transfer, and data augmentation. These will be covered in a later section.

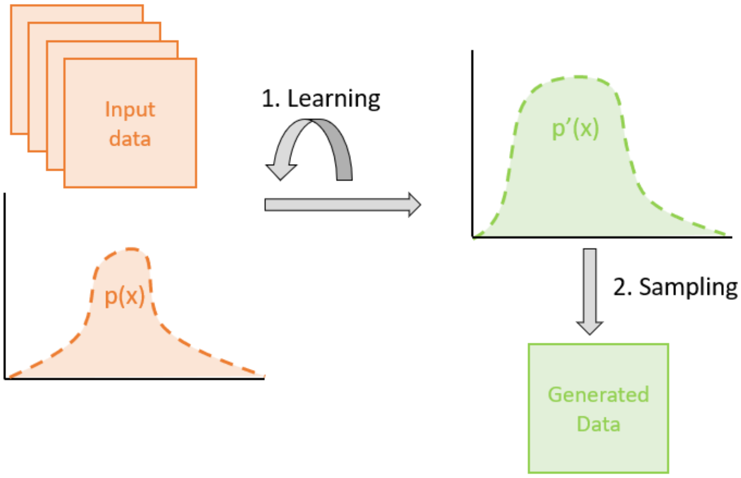

Figure 2.10: Structure of generative modeling

The above mentioned architectures are just a few examples and there exist many other variations and specialized architectures tailored for specific tasks.

2.3 Generative Modeling

Generative modeling is an emerging field that aims to create new data that closely resemble a given dataset. It revolves around the idea of learning and understanding the underlying patterns and structures of the data in order to generate new instances that exhibit similar characteristics and features. The goal is not to merely replicate existing data, but rather to capture the essence of the data distribution and generate novel samples that possess inherent variability and creativity [34].

The process of this modeling can be split into to sections, as shown in Figure 2.10. First, the learning of the true underlying patterns is initiated. In this section, the model tries to learn some hidden or latent features of the input data, usually its distribution $p(x)$. Learning (or approximating) this distribution can be achieved in several ways. Variational autoencoders do this, by mapping the data into a low-dimensional feature- or latent-space through an encoder network, which learns a probabilistic distribution over the latent variables given the data, and later VAEs use this latent space to sample (generate) the new images. Generative Adversarial Networks, on the other hand, achieve this by trying to fool the discriminator part of the network, as mentioned earlier.

It is important to note that learning the true underlying distribution of complex high-dimensional data is often infeasible or even impossible. Instead, generative models aim to approximate the true distribution as closely as possible, capturing the essential characteristics and variability of the data.

After completing the learning phase, the next step involves sampling. This process relies on the learned (approximated) distribution, denoted as $p'(x)$, which represents the model's understanding of the underlying data distribution. By leveraging this learned distribution, the generative model can generate novel samples that capture the essential characteristics of the training data.

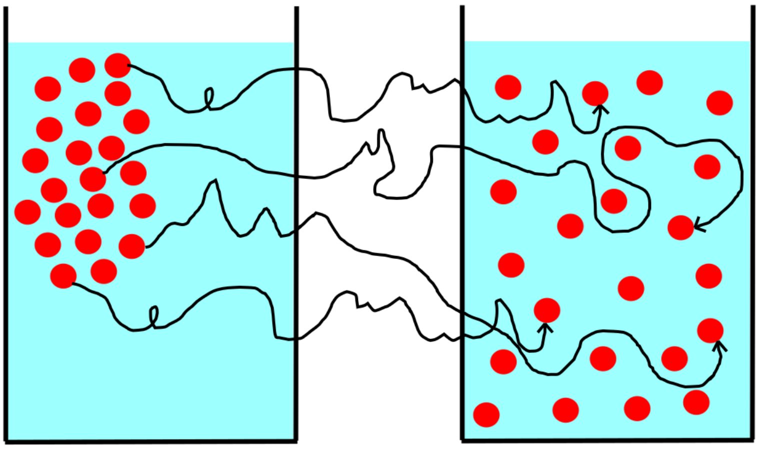

Figure 2.11: Diffusion Process, stochastic flow of particles

In addition to GANs and VAEs, a novel approach has emerged in the field of generative modeling known as Diffusion-Based generative networks [19][33].

2.3.1 Diffusion-based generative modeling

The subsequent sections will introduce the principles of the diffusion-based methodology for generative modeling, starting from its foundational components.

2.3.1.1 Diffusion Process

To capture the essence of the diffusion based generative modeling approaches, we have to understand what is diffusion itself. Diffusion is the net movement of anything - for example, atoms - generally from a region of higher concentration to a region of lower concentration. Figure 2.11 represents this specific feature. Brownian motion is also classified as a diffusion process.

Diffusion processes are a specific class of stochastic processes known as Markov processes [12] [21]. Markov processes are characterized by the Markov property. The Markov property states that the future state of the process depends only on its current state and is independent of its past states. In other words, given the present state, the future behavior of the process is not influenced by the history of the process. In the case of diffusion processes, the random motion and spreading of particles or quantities occur in a continuous manner, with the probability of transitioning from one state to another determined by the local environment and the concentration gradient.

When discussing generative modeling, diffusion processes are usually mentioned as the building blocks - or the main foundation behind the idea - of the diffusion-based generative models. By simulating the gradual spreading and mixing of information in a continuous manner, diffusion-based generative models can generate high-quality synthetic data that closely resembles the characteristics of the original data distribution. I present here one of these models.

2.3.1.2 Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models (DDPM) [19] have gained significant attention as a powerful tool for modeling complex data distributions. In this section, I present the core ideas behind this approach.

DDPMs leverage the principles of diffusion processes and probabilistic modeling to progressively refine noisy or corrupted data samples, ultimately generating high-quality and diverse samples. The model consists of a forward and a backward diffusion process. In the **forward process**, the model iteratively transforms an initial x_0 data point sampled from the original data distribution towards a target distribution. This target distribution is usually a Gaussian (Normal) distribution. These iterative steps are formulated as the following:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathbf{I}), \tag{2.2}$$

where x_{t-1} is the previous state of the data point, which is perturbed with some Gaussian noise according to a β_t scaling variable. The whole forward process consists of multiple, T repeated steps creating the following equation:

$$q(x_{1:T} \mid x_0) = q(x_1, x_2, \ldots, x_T \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}). \tag{2.3}$$

Using a large enough T, sampling from the $q(x_{1:T} \mid x_0)$ distribution will be completely independent of the original distribution of our data, i.e. $q(x_T) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$ is our target.

The term sampling might be a bit abstract in the case of $x_t \sim q(x_t \mid x_{t-1})$, but it can be formalized using the reparameterization trick

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t} \epsilon, \tag{2.4}$$

where $\alpha_t = 1 - \beta_t, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This introduction is needed to show how the sampling for any t between time 0 and T is expressed:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\tilde{\alpha_t}} x_0, (1 - \tilde{\alpha_t}) \mathbf{I}), \tag{2.5}$$

where $\tilde{\alpha_t} = \prod_{i=0}^{t} \alpha_i$. This equation will be in use during the training phase of the network in each iteration.

The **backward diffusion** process is formulated in a way, that it can complement the forward process by transforming the $x_T \sim q(x_T)$ noisy sample back into an “initial” sample, which looks as if it was sampled from the original distribution, $q(x_0)$:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t), \tag{2.6}$$

where

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2.7}$$

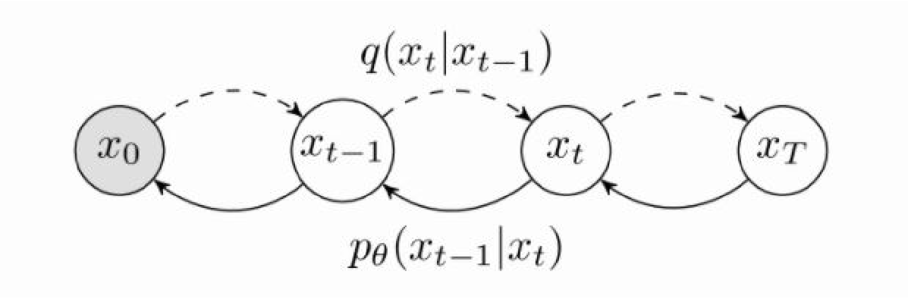

Figure 2.12: Visualization of the processes

The forward and backward processes are shown on 2.12, This $p_\theta(x_{0:T})$ is intractable, which brings the evidence lower bound (ELBO)[2] into the equation and thus, a minimization task is formulated, which is also suitable for a neural network. Through multiple mathematical simplifications, the ELBO equation becomes a "denoising" task, meaning that the neural network's only task is to match the noise, which is added to the datapoint in the forward process. If the neural network is able to successfully reconstruct this noise term, then this up-until-now intractable formula $p_\theta(x_{0:T})$ becomes predictable, and thus the backward process is feasible. The loss function of the neural is network is the following:

$$\mathcal{L} = \mathbf{E}_t[\| \epsilon - \epsilon_\theta(x_t, t) \|^2], \tag{2.8}$$

Assuming that this task succeeds, the next step is incorporating this information into the backward process, which leads to the iterative sampling procedure, from x_T up until X_0:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\tilde{\alpha}_t}}\epsilon_\theta(x_t, t)) + \sigma_t z, \tag{2.9}$$

where $t = T, ..., 0$, $\beta, \tilde{\alpha}$ are scalers and σ is some arbitrary Gaussian noise.

2.3.1.3 Conditional DDPM

The above mentioned diffusion model can be further evolved, one of the options is the introduction of conditional generation. In this case, the model is not only conditioned on the t-th time of the forward diffusion, but also on a class-label or vector-like variable, let's name it **c**. This modification does change some of the previously declared equations, but for the sake of clarity, only the sampling is re-introduced:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\tilde{\alpha}_t}}\epsilon_\theta(x_t, t, \mathbf{c})) + \sigma_t z, \tag{2.10}$$

where $\epsilon_\theta(x_t, t, \mathbf{c}))$ is the newly formulated neural network.

With this approach, the generation could be guided in a concrete direction depending on the **c** variable.

In scientific papers [19][33], the conditioning property was mostly a class label, depicting the type of the to-be-generated image, e.g. a *gorilla.* To bring this idea into this use-case, which is brain imagery, one can choose between some options. The connection (or synthesis) between EEG and fMRI signals could be investigated, futhermore, how conditioning the generation on preprocessed (or latent) EEG signals could effect the generated fMRI images' features. In the case of fMRI images, conditioning the generation on purely attached class-labels was feasible, which will be further evaluated later on.

2.3.1.4 Denoising Diffusion Implicit Models

Denoising Diffusion Implicit Models (DDIMs) are a variant of DDPMs, with a twist of introducing a non-markovian generative process, which allows for a deterministic generation. This is achieved by using an implicit sampling that allows for fewer timesteps in the backward process, resulting in a faster and more efficient sampling. This approach was used during the my generative experiments.

Chapter 3

Datasets

During the experiments, one publicly accessible dataset was used in this work to investigate the capabilities of the denoising neural networks.

3.1 ABIDE

The Autism Brain Imaging Data Exchange (ABIDE) set is formed by a collaboration of sixteen imaging sites sharing neuroimaging data. I used the preprocessed version of the data. [15] The dataset includes observations from altogether 1112 subjects: 539 autism spectrum disorder-suffering (ASD) individuals and 573 typical control participants. This nature of the subjects is referred to as "ASD vs non-ASD" throughout the rest of the paper. The subjects' resting state and structural fMRI were recorded. Multiple versions exist of the preprocessed data, as five different teams made their own versions with varying tools, pipelines and strategies. In the current work, I used the version which was produced by using the Connectome Computation System pipeline and strategy with filtering but no global signal correction.

3.1.1 ABIDE Preprocessing

As noted, the ABIDE dataset was previously preprocessed to a certain degree, however for the special use-cases of the paper, further preprocessing was necessary.

The fMRI samples were stored as 4-dimensional vectors, the first 3-dimensions being the spatial dimensions, and the fourth being the time dimension. These vectors were sliced into 3D samples along the time dimension and transposed into the order of $A \times S \times C$ (axial, saggital and coronal, in image space, channel, height and width (Figure 2.6)). Rescaling was also applied on the samples into the $[-1, 1]$ range during the experiments. Furthermore, in my experiments, I excluded measurements for which at least one of the reviewers of the fMRI measurement noted failure. I padded the slices along the saggital and the coronal dimension for, resized the images for easier processing and less computational power. The resulting dimensions of each 3-dimensional samples were $61 \times 48 \times 56$ during the denoising diffusion modeling. For each 3D fMRI sample, a number of numerical and text based features were available, some of those are

- The ASD vs non-ASD feature
- The site of the measurement

- The subject of the measurement (the person)
- The time slice of the given 3D sample[1]

During my experiments, the feature that I have dealt with the most was the **ASD vs non-ASD** feature, this was the basis of my class-conditional generation.

Splitting Having a look at several 3D samples from the same subject revealed, that "close" samples - in time manner - are not distinguishable, they carry the same structural features. This is not a surprise, since only a few seconds pass between connecting time slices.

This finding resulted in two different spliting of the train, valid and test sets during my work. These were the following:

1. Splitting to keep the ASD vs non-ASD balance across the splits, not paying attention to the site/subject cross section
2. Splitting the data in a way that no subject is present in two splits, keeping the ASD vs non-ASD balalnce as good as possible

The first split was applied at the start of the work, during the generation process and at the first iteration of the evaluations, which is detailed in 5.2.1. The second split was applied after this, which greatly improved the evaluation metrics, but no generation was done with this split yet.

[1] for one subject multiple 3D samples are available, since slicing along the temporal axis was done beforehand

Chapter 4

Methods

4.1 Background on applied neural networks

In the previous sections, the need for a neural network-based model was justified. It's task was set to be the noise "segmentation", reconstruction based on the input image and the volume of the noising, i.e. the t-th time of the forward process. In a formulated manner: $\epsilon_\theta(x_t, t)$. For such case, the U-Net architecture is commonly used[29]. The U-Net architecture is a convolutional neural network (CNN) that has revolutionized the field of medical image segmentation. The architecture exhibits a unique encoder-decoder structure that enables learning complex latent features of the input, such as - in this case - the added noise to an image.

The name "U-Net" originates from the U-shaped architecture formed by its symmetrical structure. The network architecture comprises two key components: the contracting path (encoder) and the expansive path (decoder). The contracting path consists of a series of convolutional and pooling layers, which capture the spatial information from the input image. This encoding process progressively reduces the spatial dimensions while increasing the number of feature channels. The expansive path, on the other hand, is responsible for recovering the spatial information lost during the encoding process. It employs transposed convolutions, also known as upsampling, to gradually increase the spatial resolution while reducing the number of feature channels. Skip connections are a distinctive feature of the U-Net architecture, connecting corresponding layers in the encoder and decoder paths. These connections enable the network to utilize both local and global contextual information during the segmentation process. On Figure 4.1, this skip connection is visualized.

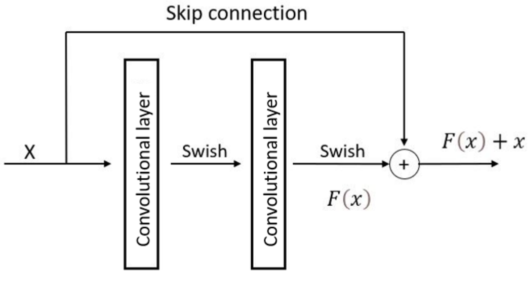

Figure 4.1: Skip connection in the U-net model

In addition to this, in my model, incorporating - embedding - the timing information t from the **forward process** is a necessity, in order to learn the appropriate size of the noise added to the input image. Time embedding involves encoding temporal information into the input data, allowing the network to learn and exploit temporal dependencies during the segmentation process. For this, I used the positional encoding introduced in the Attention Is All You Need paper[39]. 4.1 shows how the positional embedding from the position - in my case t - is calculated. The authors stated, that using this the would allow the model to learn the relative positions.

$$\begin{aligned} P_E(\text{pos}, 2i) &= \sin\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right) \\ P_E(\text{pos}, 2i+1) &= \cos\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right) \end{aligned} \tag{4.1}$$

This extension enhances the U-Net's ability to handle time-varying patterns, motion, and temporal context. In our case, this time embedding is existent at each convolutional (skip) connection, allowing the network to fuse both spatial and temporal features at different scales.

My model has 4 downsampling and 4 corresponding upsampling blocks, with each block having residual skip-connections. Each upsampling step makes use of the concatenation with the matching downsampling state, which preserves some context of the original input. Therefore, the final output of the model is the exact same in case of dimensionality as the input image.

4.2 Problem formulation

The generation of the fMRI samples involve two phases, namely the training of the neural network, and the sampling part, which features this trained network for its denoising capabilities

Training Having sampled a batch of fMRI samples from the dataset $\mathcal{S} = \{x_1, x_2, ...\}$ with a distribution of $p(\mathcal{S})$, where each sample has a shape of $61 \times 48 \times 56$, these samples are randomly perturbed with t steps of Gaussian noise, resulting in a x_t sampled from the 2.5 distribution. This x_t, t pair is fed into the neural network ϵ_θ, - where θ denotes the trainable parameters. The neural network's goal is to give an estimate about the noise added to the x_t sample in each forward t step. The objective is to minimize the distance between the estimate $\epsilon_\theta(x_t, t)$, and the gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

Sampling If trained, the neural network with θ learned parameter is used in this process. Starting from a random noise sample $\tilde{x_T} \sim \mathcal{N}(0, \mathbf{I})$, the sampling process iterates from T to 0, while gradually removing the noise learned in the training phase. If succeeded, the final product of the sampling $\tilde{x_0} \sim p(\mathcal{S})$, thus the neural network is able to generate samples, which could belong in the original dataset.

4.3 2D fMRI generation

The experiments started with the generation of the 2-dimensional fMRI samples, meaning only the saggital and coronal axis of the originally 4-dimensional data was used.

This means, that the slices of the axial dimension were separated, and included in the dataset as normal different samples. With this, the size of the dataset greatly increased. The training was performed on a machine with GPU capabilities, namely Nvidia Geforce 3090 (24 gigabytes of VRAM), and lasted for 16 hours. Results obtained from the sampling are presented on Figure 4.3.

Conditioning on class labels In the case of the ABIDE dataset, the conditioning was done using the ASD versus non-ASD features of the samples, which - for the human eye - is invisible, thus evaluating this condition is up to the metrics mentioned in 5.

Conditioning on class and index labels Initially, the concept of generating three-dimensional samples emerged from a conditional perspective. Namely, that conditioning the neural network - just as in the case of class-conditioning - on the indexes of the slices might work as expected, resulting in samples which are in succession.

Taking a look at the real images in 4.2, this nature of the data is visible, i.e. the three images representing axial dimension slices 50, 52, and 54 appear to show a progressive reduction in the size of the brain signals. In the case of the generated images, the following section summarises the findings.

4.3.1 Generating power of 2D approach

If we take a look at the some consecutive fMRI slices from a real sample, it is obvious that some kind of connection is present between the images. In our case, the successive slices are getting smaller and smaller, which of course does not come as a surprise, since the axial view of the fMRI does shrunk with superior slice numbers.

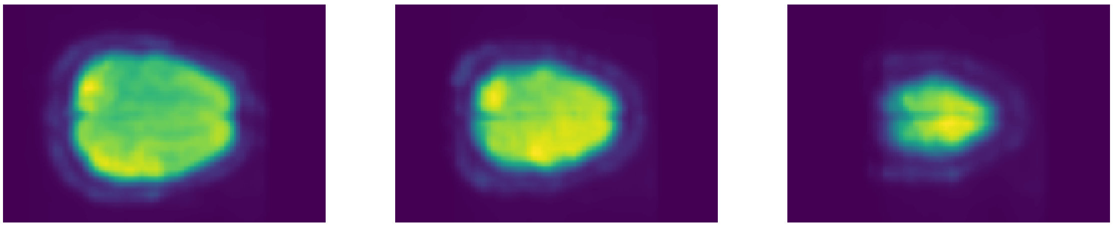

Figure 4.2: Real 2d samples

On the other hand, the index-conditioned 2 dimensional fMRI generation can not take into account information from the previous slice, which can be explained rather simply. Since the number of training samples and subjects in the dataset is so large, it is not feasible to have every - let's say - 50th axial slice to carry the same features. This means, that if the neural network is given the index 50 and 52, it might produce slices closer to two different subjects, resulting in cases shown on Figure 4.3. In this particular case, the 52nd slice likely corresponds to a subject whose original brain images were not adequately captured, or it is possible that the 52nd slice has exceeded the upper boundary of the brain.

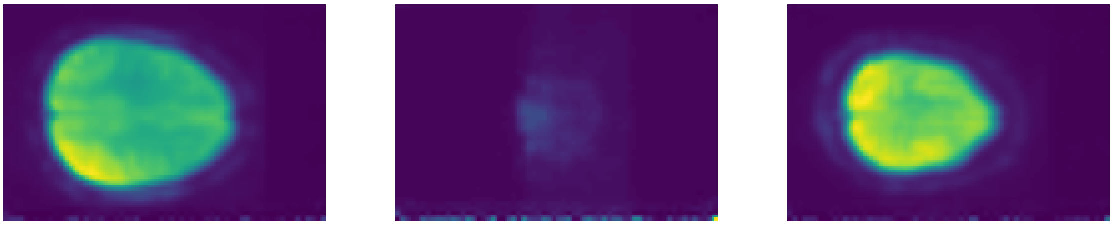

Figure 4.3: Index conditioned 2d generated samples

Learning from these experiences, I turned to 3-dimensional - including the axial dimension - generation.

4.4 3D fMRI generation

In the case of 3-dimensional generation, the only change included was the modification of the neural network. This meant the inflation of the thus far 2-dimensional convolutional layers to 3-dimensional ones. The sampling of the generated samples did not change, only the new dimension was added.

My experimentation encompassed various sampling methods, as outlined in 2, including DDPMs and DDIMs, each exhibiting distinct characteristics in terms of speed and memory usage.

The most prominent trials were the following, each with included generated samples - the images show the axial slices starting from upper left corner:

- The usage of DDIM sampler, with a step size of 8[1]. This sampling was the fastest, but resulted in less meaningful images. Figure 4.4 shows the results.
- The usage of DDIM sampler with a bigger - yet compared to DDPM still a small - number of steps, 64. Here, the generation produced more visible and easier-to-see samples[2].Figure 4.5 shows the results.
- The usage of the DDPM sampler, despite being the most time-consuming among the three mentioned due to its 1000 timesteps in the reverse phase, produces results that are significantly more realistic, as evidenced by Figure 4.6.

To have a strong base of qualitative evaluation of the generated samples, Figure 4.7 depicts how a 3-dimensional sample, originating from the ABIDE dataset looks like.

[1] this number represents the steps taken in the backward - generating - process

[2] this saturation of the images is one downside of DDIM samplers

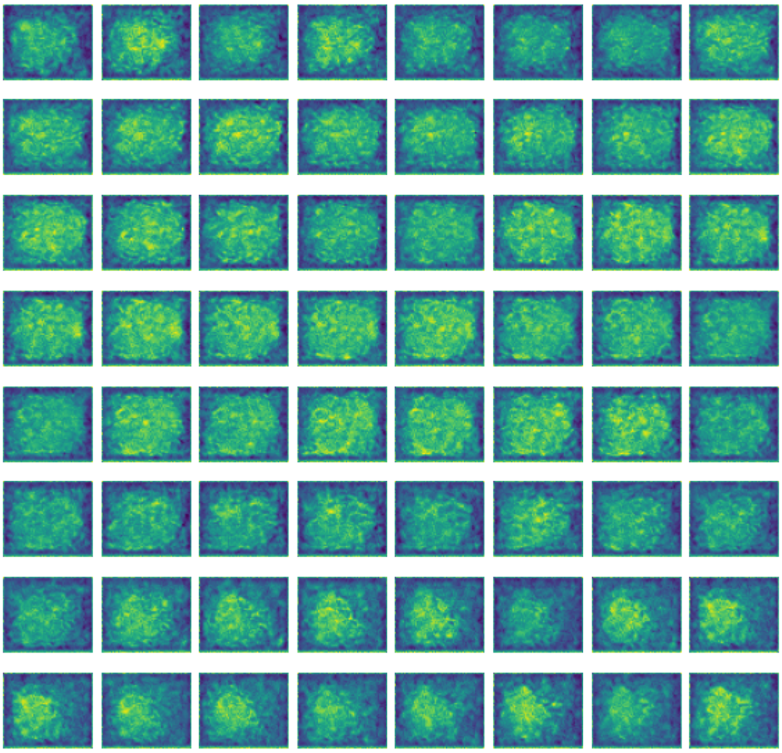

Figure 4.4: Generated ABIDE samples with DDIM (8 steps)

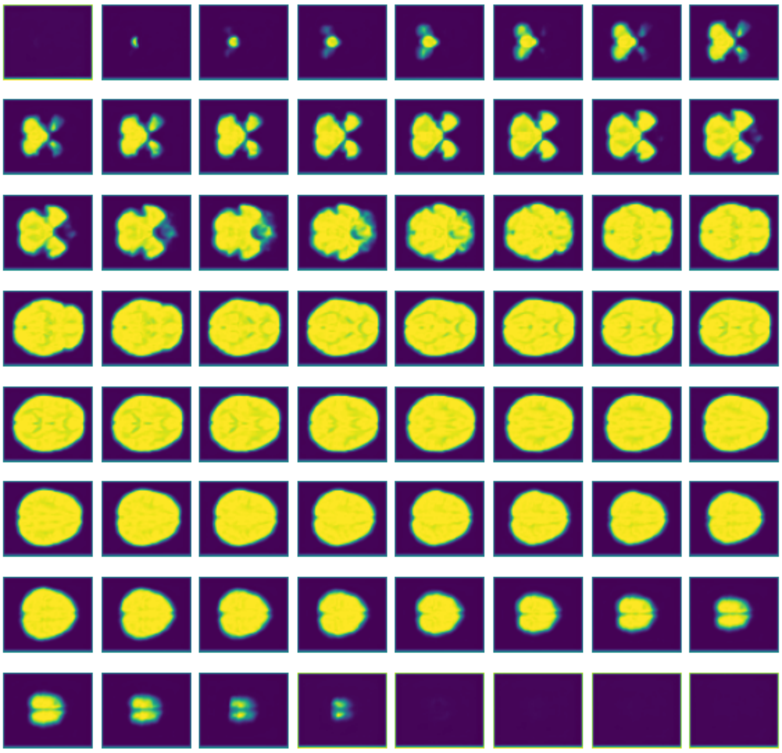

Figure 4.5: Generated ABIDE samples with DDIM (64 steps)

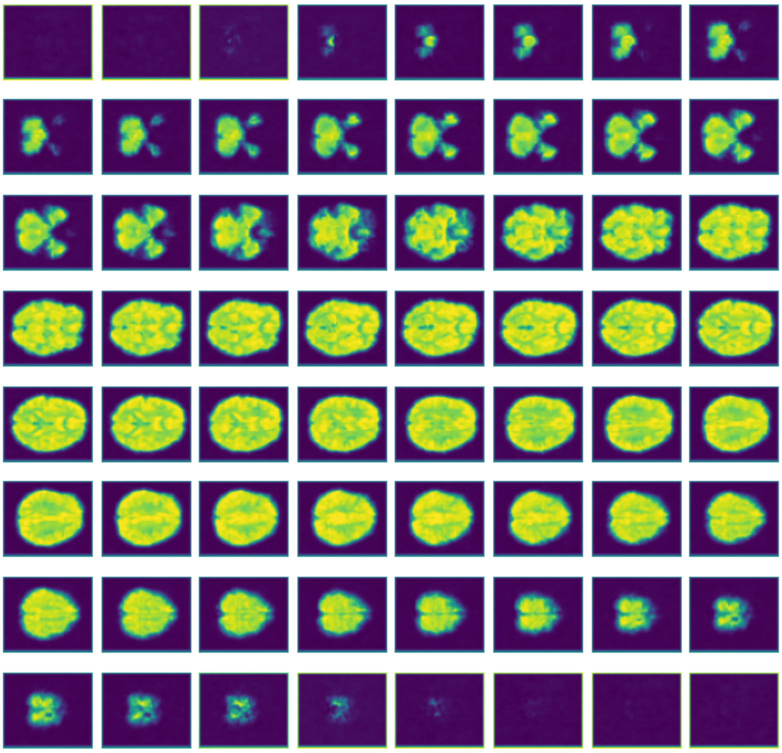

Figure 4.6: Generated ABIDE samples with DDPM

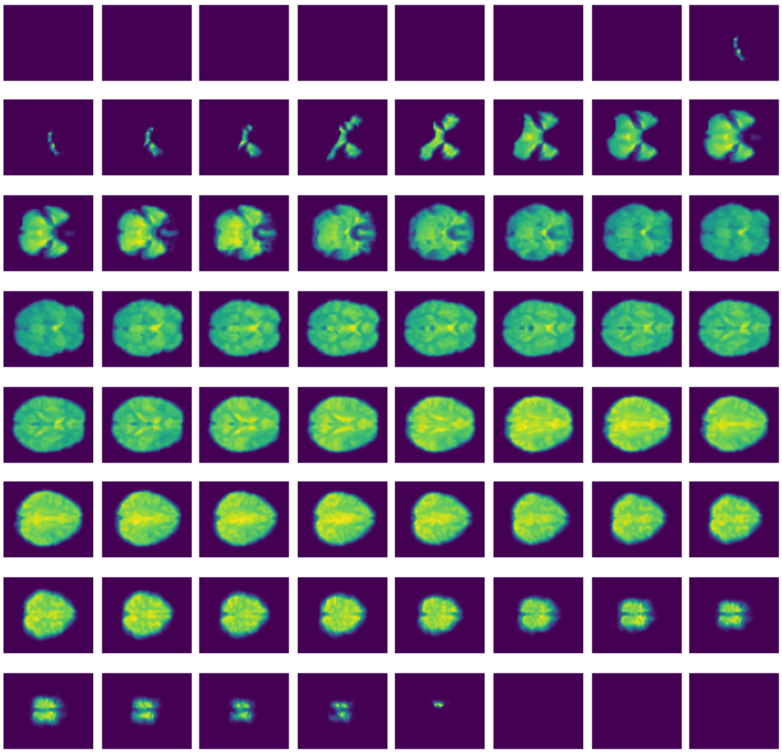

Figure 4.7: Real ABIDE samples

Chapter 5

Evaluation

Evaluation involves the systematic process of assessing the value, quality, and significance of a subject, guided by a predefined set of criteria and standards. This method is universally employed across various domains, including human resources, engineering, and healthcare, to gauge the effectiveness, performance, and impact of initiatives and projects. In the field of deep learning it is no different, evaluating the product, the output of the experiments is a must. In my case, the generated MRI and fMRI signals are assessed in this manner.

5.1 Qualitative and Quantitative Evaluation

Evaluation, broadly speaking, can be done in multiple ways. Two common sides of the process are qualitative and quantitative evaluations, both these methods play a huge role in the final verdict. Quantitative analysis of data is primarily concerned with the measurement and quantification of various attributes, emphasizing numerical values and statistical techniques. It is often used to answer questions related to "how much" or "how many" and is highly suitable for an objective verdict. Qualitative analysis on the other hand, aims to explore and understand complex, non-quantifiable phenomena, often through the analysis of patterns. In this context, it involves finding insights in the generated data, which can be justified by the human eye.

5.1.1 Quantitative Evaluation Metrics

Before evaluating the results of my research, it is best to introduce the metrics used most commonly in the generative modeling domain. This could also help the individual to have an understanding of the capability of my approach, by comparing my results on these common metrics to other public scores in the field.

Root-Mean-Square Error (RMSE) RMSE is the standard deviation of the prediction errors. By using s for the generated data and o for the original data, the formula of RMSE can be written in the following way:

$$RMSE_{so} = \sqrt{\frac{\sum_{i=1}^{N}(\mathbf{x_{pi}} - \mathbf{x_{oi}})^2}{N}}, \tag{5.1}$$

where Σ is summation ("add up"), $(x_{pi} - x_{oi})^2$ are the differences, squared and N is the sample size. The underlying assumption when presenting the RMSE is that the errors are unbiased and follow a normal distribution. Thus, using the RMSE or the standard error (SE) helps to provide a complete picture of the error distribution (Chai et al., 2014).

Structural Similarity Index (SSIM) This universal image quality assessment was developed to measure the difference of a degraded image from the reference image. [**?**] Similarly to the human perception, changes in structural information are easily quantified between two images. The SSIM considers three factors, which are luminance, contrast, and structure.

From these SSIM can be formulated as

$$SSIM(\tilde{\mathbf{x}}, \mathbf{x}) = [l(\tilde{\mathbf{x}}, \mathbf{x})]^\alpha * [c(\tilde{\mathbf{x}}, \mathbf{x})]^\beta * [s(\tilde{\mathbf{x}}, \mathbf{x})]^\gamma \tag{5.2}$$

As originally SSIM was developed for the comparison of 2-dimensional images, I split the generated 3-dimensional fMRI samples along the axial dimension into 2D images and calculate the average of the SSIMs of these slices for the whole sample.

Support Vector Machine (SVM) Support Vector Machines are supervised machine learning models used for classification or regression problems. By training an SVM on actual fMRI data points and their associated labels, we can achieve accurate predictions of classes for previously unseen signals. This method serves as a means to evaluate the performance of my conditional generation approach, as it tests whether the trained SVM can effectively differentiate between the different conditional classes. Accuracy, precision and recall are calculated.

Learned Perceptual Image Patch Similarity (LPIPS) LPIPS essentially computes the similarity between the activations of two image patches for some pre-defined network. Typically, This predefined network is a pre-trained neural network on a large and diverse dataset. This measure has been shown to match human perception well. A low LPIPS score means that image patches are perceptual similar, thus indicating good reconstruction, while a higher score indicates the opposite.

Inception Score (IS) The Inception Score is particularly used to assess the capabilities of generative models, GANs most frequently. The score is calculated based on the output of a separate, pre-trained InceptionV3 image classification model (5.1). This model was trained on more than 30 000 image samples belonging to different genres. The Inception Score is maximized when the following conditions are true:

- The entropy of the distribution of labels predicted by the Inceptionv3 model for the generated images is minimized. In other words, the classification model confidently predicts a single label for each image. Intuitively, this corresponds to the desideratum of generated images being "sharp" or "distinct".
- The predictions of the classification model are evenly distributed across all possible labels. This corresponds to the desideratum that the output of the generative model is "diverse".

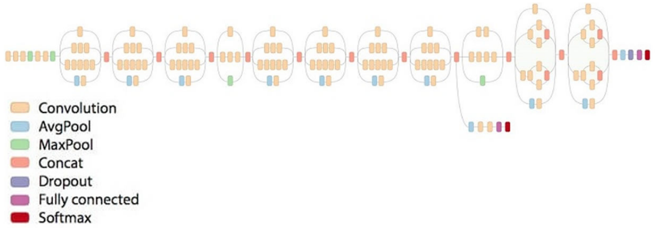

Figure 5.1: InceptionV3 network

Frechet Video Distance (FVD) A relatively new metric to measure the performance of generative models for video data, based on the principles of Frechet Inception Distance (FID) that uses feature representations that capture the temporal coherence of video-contents and taking quality into consideration of each frame [**?**]. The metric is calculated as follows:

$$d(I(x_o), I(x_s)) = \|\mu_o - \mu_s\|^2 + Tr(\Sigma_o + \Sigma_s - 2\sqrt{\Sigma_o\Sigma_s}) \tag{5.3}$$

, where $I(x_o)$ and $I(x_s)$ are the feature representations of the original and generated samples, respectively, while $I()$ is a pre-trained model producing the features.

For a pre-trained network, we trained a version of the Inception3D network for binary classification of neurological condition on the *ABIDE* set and used its features. When calculating the metric on our generated *ABIDE* set, the image slices along the axial dimension are considered as frames.

5.2 Evaluation results on fMRI

As presented, fMRI data is a rather complex modality, which does make the quantitative evaluation a challenging task. Specifically, the characteristics of the ABIDE dataset, the primary source of training samples, present certain difficulties:

1. For now, only 3 dimensional data is used, which does abandon valuable timing information

2. The classes, which were used during class-conditioning are not fully distinguishable by a complex neural network, as presented in 5.2.2

3. The different sites (cities) who participated in the measurement use differently set up machines, resulting in differences in the obtained samples, which might be even more significant, than the class label itself

4. The anatomical nature of the samples is not negligible

5.2.1 Concerns regarding popular evaluation metrics

Since the scope of the task was primarily on the generation of images, it comes as straightforward to test the results against the widely acclaimed metrics, such as Inception Score [31] or FID [37].

For these metrics, the training of an inception network on the available fMRI training samples was necessary, in order for the network to be useful in the evaluation of the generated samples. This training was in fact a classification task, where the two classes in question were the ASD and non-ASD features of the ABIDE samples. The network used was a based on the inception network depicted on Figure 5.1, with some additional inflation to 3 dimensional convolutional kernels, making it suitable for the ABIDE samples (checkl, hogy le van e irva, hogy 3ds az adat). As opposed to traditional inception networks, it contains only two output neurons for the two classes.

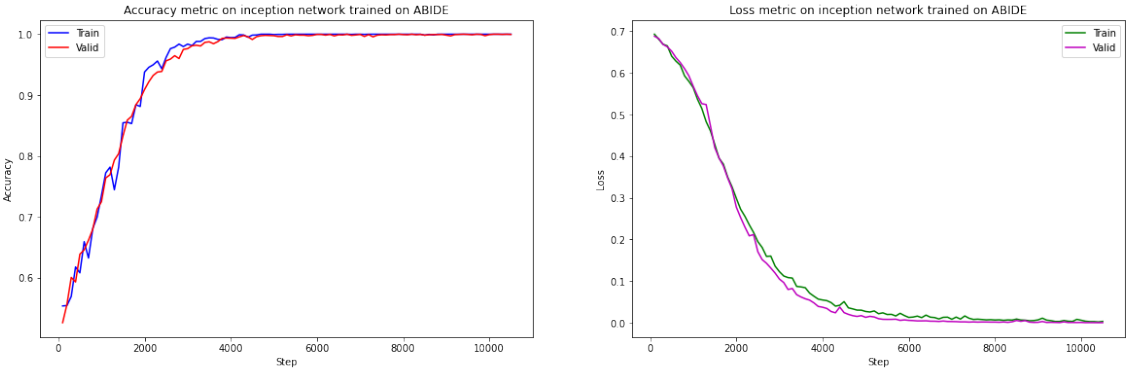

Figure 5.2: Inception network in training

As shown in Figure 5.2, the training converged quickly, raising interests in its generalization capabilities.

After extracting weights from the network in different phases of the training - mainly to check how the generalization evolves - the evaluation of the predictions on the test dataset took place, with the help of common metrics, such as accuracy, precision and recall. On Table 5.1, one can see that even after a few - in this case, 40 - epochs, the network was able to perfectly classify the images into one of the classes.

Epoch	Accuracy (%)	Precision (%)	Recall (%)
40	99.5	99.5	100
140	100	100	100

Table 5.1: Inception generalization capability

Such results were neither planned nor hoped for, thus a deeper understanding of the underlying problem was needed. Other articles about neural network based classification of fMRI - more specifically, ABIDE - samples achieved accuracy values in the region of 60-70% [17]. The key findings from those papers were that

1. in the case of the ABIDE samples, site - or even - subject related differencies in the data might play a bigger role than the ASD, or non-ASD feature (machine setup, anatomy (detailed in datasets)).

2. Thus, when constructing the different - train, validation and test - splits, it is important to have one subject only in one split, otherwise, from the network's point of view, it might seem like the same sample is present in multiple splits.
3. As input to the network - in the case of fMRI samples - using the region of interest (ROI) timeseries instead of raw images might work better on the long run.

In my case, since the training of the denoising diffusion network was already done, having to incorporate subject or ROI related information was not feasible, thus, using the inception network based metrics, such as FID or inception score, are for future use-cases.

Non-inception based metrics Metrics, e.g. PSNR, SSIM or LPIPS can be however calculated on the generated samples without the need for an external feature extractor. For this experiment, I evaluated three different datasets against the training split of the ABIDE samples. The first was its corresponding test set, meaning it contained real samples with same characteristics. A noise dataset was formulated, in order to represent some kind of upper - or in other cases lower - bound for the metric. This dataset was sampled from a Gaussian normal distribution. The third set consisted of the generated samples from the ddpm pipeline. All the inputs were normalized to [-1,1] and in image dimension, meaning a shape of (61 × 48 × 56).

Dataset	**PSNR**	**SSIM**	**LPIPS**
Ground Truth	20.3	0.71	0.087
Noise	7.07	0.005	0.76
Generated	19.86	0.74	0.12

Table 5.2: fMRI data metrics

5.2.2 Contrastive Encoder

After the initial difficulties regarding ASD-based image classification, I have delved into other options to find a valuable evaluation metric, on which the fidelity of the generated images can be tested. Recently, i have came across a paradigm called Contrastive Learning, which showed promising approaches and ideas.

Contrastive Learning The primary objective of contrastive learning is to utilize a neural network to construct an embedding space that effectively separates similar and dissimilar sample pairs. This is achieved by optimizing a contrastive loss function, which is designed to maximize the similarity between "positive" pairs (e.g., augmented versions of the same data point) and minimize it between "negative" pairs (e.g., different data points or classes). The form of this loss function can be adapted based on the specific requirements of the task. This learning approach is versatile, applicable in both supervised[22] and unsupervised[13] settings.

Having sampled N random samples $\{x_k\}_{k=1}^{N}$ from the original dataset, one can construct the training dataset required for the contrastive framework, which features the proposed positive and negative pairs, in the following way:

- Create two random augmented[1] versions of x_k, here denoted as $\tilde{x}_{k1}$ and $\tilde{x}_{k2}$

[1] In my experiments, I used random flipping, cropping and affine transformation as augmentation on the fMRI data

- Assemble the dataset out of the augmented pairs, so that the training dataset will consist of $2N$ pairs
- If labels are available, make sure that $\tilde{y}_{k1} = \tilde{y}_{k2} = y_k$

For unsupervised framework, the following objective function was used.

$$\mathcal{L}_{\text{self}} = -\sum_{i \in I} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_j(i)/\tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a/\tau)} \tag{5.4}$$

Where $i \in I \equiv \{1, \dots 2N\}$ denotes the index of an arbitrary augmented sample, $j(i)$ is the index of the other augmented sample originating from the same source sample, $\mathbf{z}$ is the output of the neural network (presented in 5.2.2), the $\cdot$ symbol denotes the inner (dot) product, τ is a scalar temperature parameter, and $A(i) \equiv I \setminus \{i\}$. The index i is called the anchor, index $j(i)$ is called the positive, and the other $2(N-1)$ indices ($k \in A(i) \setminus \{j(i)\}$) are called the negatives. For each anchor i, there is 1 positive pair and $2N-2$ negative pairs.

For a supersived setup, the number of positive pairs can - and will - exceed only 1, so in order to incorporate this information into the loss function, it is rephrased as

$$\mathcal{L}_{\text{sup}} = -\sum_{i \in I} \frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p/\tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a/\tau)} \tag{5.5}$$

Here, $P(i) \equiv \{p \in A(i) : \tilde{y}_p = \tilde{y}_i\}$ is the set of indices of all positives in the training dataset distinct from i, and $|P(i)|$ is its cardinality.

Through my experiments, I have applied both supervised, and unsupervised contrastive learning on the ABIDE dataset, in the latter case neglecting the normally present ASD, non-ASD labels. My conducted trials were

1. Supervised training with ASD, non-ASD labels using $\mathcal{L}_{\text{sup}}$
2. Unsupervised training with $\mathcal{L}_{\text{self}}$

Encoder In the contrastive representation learning domain, the choice of the neural network can greatly affect the results of the process. One common architecture choice is the usage of an encoder model, which is designed to transform raw data into a lower-dimensional, compressed representation, often referred to as an “embedding”. The primary goal of an encoder is to capture the essential features or characteristics of the input data in this compressed form, which - in the domain of contrastive learning - is in fact, very useful.

I opted for an variational autoencoder based encoder network (presented in [28], used for latent denoising generation), which takes 3 dimensional input vectors - images - and transforms those, into a smaller, but still 3 dimensional embedding. I placed an additional projection MLP[2] (projection head) on top of the encoder's output, to create a smaller, flattened version of the embedding, which is helpful in loss convergence during training, but is discarded during testing.

[2] Multi Layer Perceptron

Experiments The previously mentioned two trials were both performed on the ABIDE dataset, using the second type of data-splitting technique detailed in 3.1.1. The input data was augmented to have a shape of ($56 \times 56 \times 56$) (CxHxW) and was fed into the network with a batch size of 40 - after augmentation. The encoder network produced an output of size ($56 \times 56 \times 56$) with the projection head resizing it to a length of 128. The training was carried out on a NVIDIA 3090 with 24 gigabytes of VRAM.

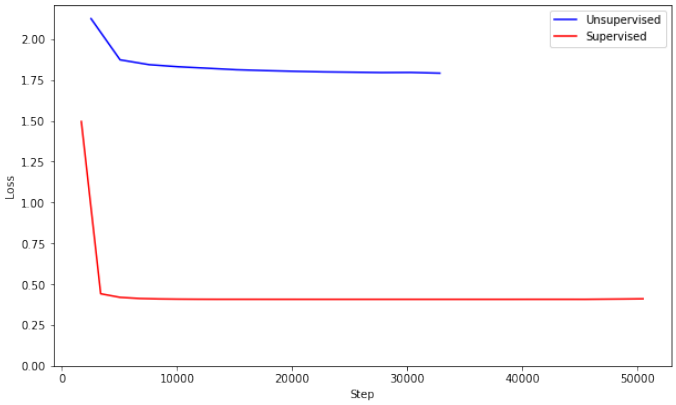

Figure 5.3: Supervised, and unsupervised loss of the contrastive learning algorithm

The representation ability of a contrastive encoder can be validated through a number of methods. One of those is by the reduction of the dimensionality of the encoder output - in this case ($14 \times 14 \times 14$) - to an arbitrary number, in most cases 2. By doing this, the embeddings of the tested dataset can be represented on a plot, with the two dimensions being the X and Y value.

t-Distributed Stochastic Neighbor Embedding (t-SNE)[38], is an algorithm designed for dimensionality reduction and visualization of high-dimensional data, and is particularly effective at preserving local structures and revealing clusters. I chose this, since the number of sites and subjects in the data is huge, thus hoping for the t-SNE to represent these connections in the embeddings.

After training, contradictory to Figure 5.3 - where it seems, that the supervised setup might work better, given that its loss function converged to better optima - the testing and the usage of the t-SNE algorithm reveals that the unsupervised learning setup was able to seperate the input fMRI samples in the latent space, while the supervised setup could not.

Figure 5.4 reveals the output of the t-SNE algorithm applied on the test fMRI samples, highlighting the classes (ASD vs non-ASD feature) of the data. This plot reinforces that the statement - made in 5.2.1 - is true, namely that the most prominent features in this dataset are not the ASD vs non-ASD nature, rather the differences or site - or subject - level.

Following this, extracting the site feature from the test dataset was done, and Figure 5.5 clearly shows that on site-level, the embeddings do separate more precisely. Selecting two distinct sites, namely Caltech (California Institute of Technology) and Yale, Figure 5.6 depicts the embeddings of the subjects at each site, having no overlap between each other. This means, that without the need for any kind of previous information about the classes or the sites, this contrastive learning approach is able to “classify” the samples.

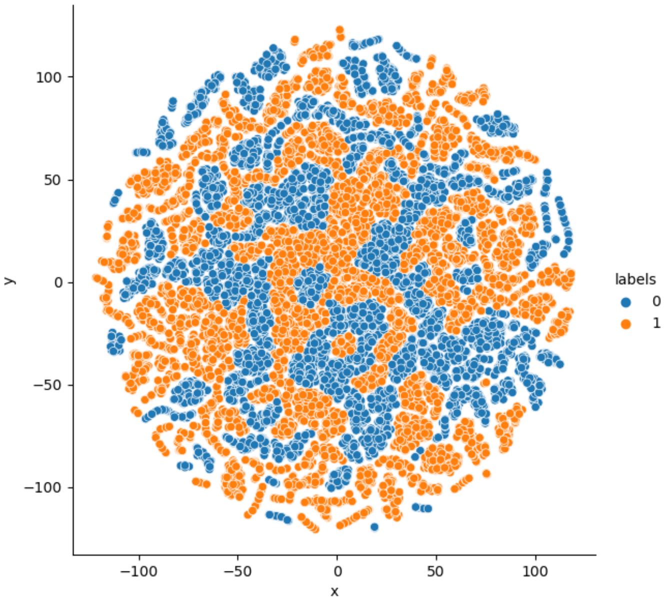

Figure 5.4: Classwise highlight of the contrastive embeddings using t-SNE

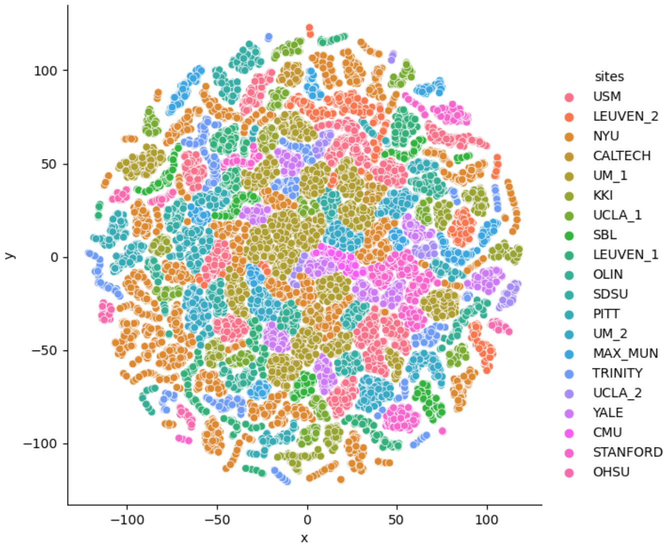

Figure 5.5: Site-wise highlight of the contrastive embeddings using t-SNE

These two dimensional latent space vectors were also acquired from the generated fMRI samples, and from an arbitrary number of noise samples[3], and plotted on the same figure. On Figure 5.7 both the generated samples, and the noise samples form a “subject” cluster, meaning they exhibit characteristics which previously were not fed into the encoder.

It does not come as a surprise, that while the embeddings of the noise samples are concentrated strictly in one place, the embeddings from the generated samples do overlap with some other subjects and sites, meaning that the generation did pick up some subject-specific features.

[3] sampled from Gaussian normal distribution

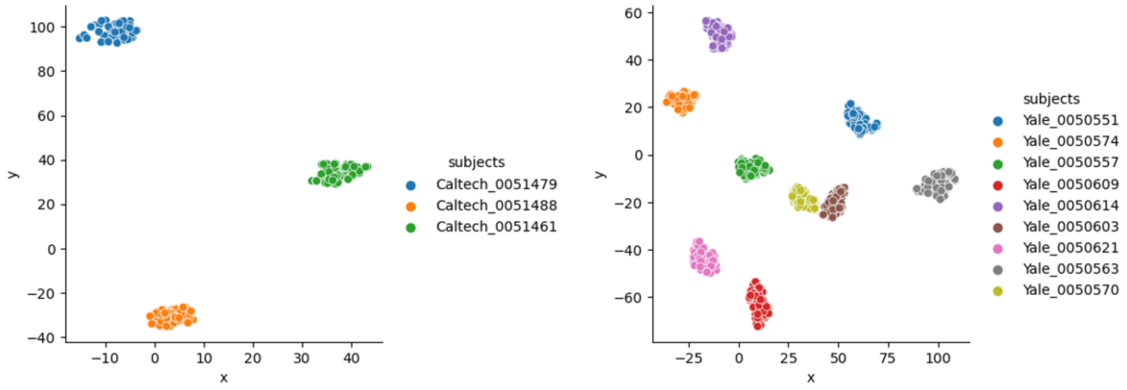

Figure 5.6: Subjects on Caltech and on Yale site

With the class-conditioned generated samples not fully aligned with their corresponding real classes, it is right to say that conditioning on ASD and non-ASD classes are not the best features to choose in this ABIDE dataset, rather the conditioning on ROI or subject/site is advised.

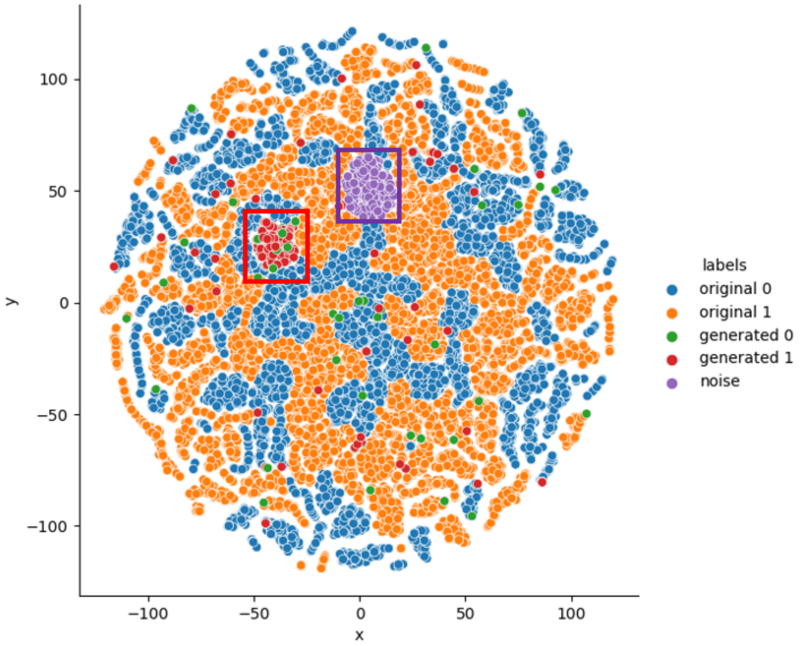

Figure 5.7: Generated and noise samples on the plot

Certainly, the objective here was also to be able to measure the generating power of the denoising neural network, however even without that, this experiment was more than useful regarding the future directions.

5.2.3 SVM

Support Vector Machines - as mentioned - are also present in the evaluation of the generated samples. They gained prominence for their robustness and efficacy in both classification and regression tasks. In my case, after all the preceding metrics, the goal was also to check the class-conditional generating power of the DPM.

Learning from previous failures - namely that, there are underlying artifacts in the data, related to subjects and sites - the training set only consisted on samples from one site. This chosen site was the one with the most samples in it, NYU.

I conducted two trainings with multiple SVM kernels[4]. These trainings featured 3000 real fMRI samples, and 500 generated samples, flattened out to a 163968-long vector (originally $61 \times 48 \times 56$).

1. Training on real fMRI images, testing on real images, and generated images
2. Training on real and generated images, testing on real images. This approach is to "measure" the augmentation power of the generated samples.

In both cases, three different SVM kernels were used, linear, polynomial and rbf kernels. The predicting power of the machines are tested on different metrics, such as accuracy, F1 score, specificity[5] and an additional confusion matrix is also present for visualization purposes.

Dataset	**Accuracy (%)**	**F1 Score (%)**	**Specificity (%)**
Real test	66.0	75.6	31.7
Generated test	55.8	58.4	50.0

Table 5.3: First SVM tested on real and generated data

Table 5.3 contains the results from the first training, in every case the best performing SVM was selected. In both cases, the SVM with the best results was the polynomial kernel based machine, with a parameter set detailed in Table 5.4. The term C corresponds to an inverse-regularization term, while degree is the degree of the polynomial kernel.

Kernel	**C**	**Degree**
poly	100	3

Table 5.4: Parameter set of the best performing SVM

Surprisingly, the model achieves better specificity on the generated dataset than on the real dataset. This means, that in the case of real dataset, the false positive ratio is more significant (also shown on Figure 5.8). The cause of this is unknown at the momemt, worth checking - but important to mention, that the number of class labels were balanced during training and testing, so label-imbalance, which is typically the root cause of this issue, is not present here. For accuracy and F1 score, the results are better on the real test dataset - as one would imagine.

Dataset	**Accuracy (%)**	**F1 Score (%)**	**Specificity (%)**
Real test	82.4	85.1	75.9

Table 5.5: Second, augmented SVM tested on real data

In the second case, after augmenting the dataset used in the first training, the results presented in Table 5.5 exceeded its corresponding metric scores shown in the first training, by a considerable margin.

[4] the scikit-learn python package was used

[5] Specificity = $\frac{\text{TN}}{\text{TN+FP}}, F1 = 2 \times \frac{TP}{2 \times TP + FP + FN}$

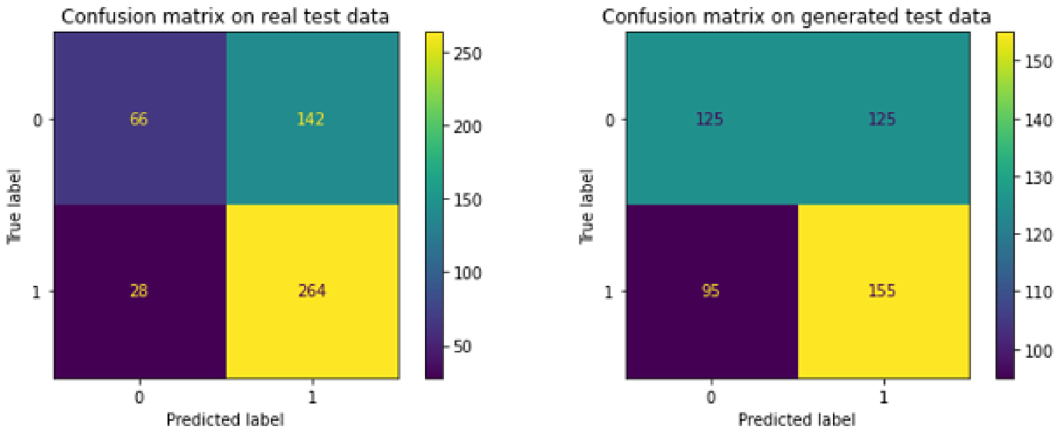

Figure 5.8

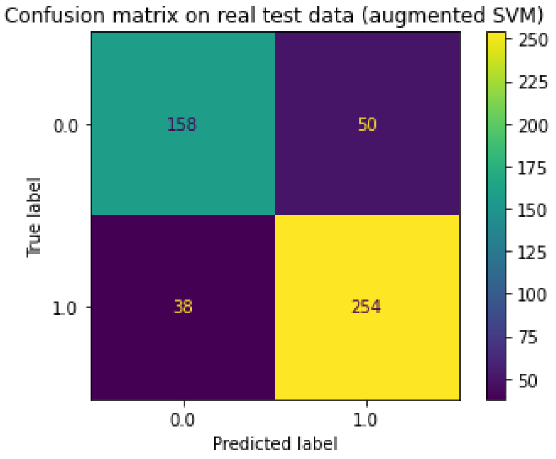

Figure 5.9

This improvement across all the evaluated metrics (as well as in Figure 5.9) shows that the augmentation power of the generated samples is significant. This can be further optimized by incorporating more stable conditioning.

Chapter 6

Conclusions

As demonstrated, my efforts have yielded successful generation of synthetic fMRI samples in both two and three dimensions. These samples have been rigorously evaluated against established metrics, delivering encouraging outcomes in the augmented SVM assessment. Moreover, this work has shown potential ideas for development that have not been previously reported in the literature.

Research in this field is ongoing, and throughout the experimental process, various questions and ideas have emerged regarding possible enhancements in both the generation and evaluation stages. These ideas include:

- Extending the generation to 4-dimensional domain, including the timing information in the process as well
- Introducing reconstruction guidance-based sampling, to enhance the fidelity of the generated samples [20]
- Condition the generation on more prominent features in the ABIDE dataset, namely on subject or site level
- In addition to the aforementioned evaluation metrics, site/subject-based training of the Inception network might produce valuable results, facilitating the application of FID and IS metrics.

Acknowledgements

I would like to express my gratitude to all those who have contributed to the completion of this project. I am thankful to my supervisors, Dr. Luca Szegletes and Szabolcs Torma for their guidance, support, and valuable insights throughout the research process. Their expertise and dedication have been instrumental in shaping the direction of this work.

Furthermore, I would like to thank my family and friends for their support and encouragement during the course of this thesis. Their love and belief in my abilities have been a constant source of motivation.

Bibliography

[1] Sketch of how to record an electroencephalogram. https://nghenhansu.edu.vn/electroenc-1695617280630685/.

[2] From elbo to ddpm. https://jaketae.github.io/study/elbo/.

[3] fmri research. https://pstnet.com/product_category/fmri-research/, .

[4] All about functional magnetic resonance imaging (fmri). https://psychcentral.com/lib/what-is-functional-magnetic-resonance-imaging-fmri, .

[5] Bold and brain activity. https://mriquestions.com/does-boldbrain-activity.html.

[6] Mini mri. https://hackaday.io/project/187854-minimri.

[7] Magnetic resonance imaging (mri) of the brain and spine: Basics. https://case.edu/med/neurology/NR/MRI%20Basics.htm, .

[8] Magnetic resonance, a peer-reviewed, critical introduction, chapter twenty-one, facts and figures. https://www.magnetic-resonance.org/ch/21-01.html, .

[9] Braviz v2 a web interactive toolkit for optimization and support of fmri and clinical data analysis, img 6. https://repositorio.uniandes.edu.co/server/api/core/bitstreams/08ee8735-ca6e-4590-8865-909c12690897/content.

[10] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1):53, Mar 2021. ISSN 2196-1115. DOI: 10.1186/s40537-021-00444-8. URL https://doi.org/10.1186/s40537-021-00444-8.

[11] David Calhas and Rui Henriques. Eeg to fmri synthesis benefits from attentional graphs of electrode relationships, 2022.

[12] Ka Chan, C. Lenard, and Terence Mills. An introduction to markov chains. 12 2012. DOI: 10.13140/2.1.1833.8248.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[14] Eric T. Chou and John A. Carrino. chapter 10 - magnetic resonance imaging. In Steven D. Waldman and Joseph I. Bloch, editors, *Pain Management*, pages 106–117. W.B. Saunders, Philadelphia, 2007. ISBN 978-0-7216-0334-6. DOI: https://doi.org/10.1016/B978-0-7216-0334-6.50014-5. URL https://www.sciencedirect.com/science/article/pii/B9780721603346500145.

[15] Cameron Craddock, Yassine Benhajali, Carlton Chu, Francois Chouinard, Alan Evans, András Jakab, Budhachandra Singh Khundrakpam, John David Lewis, Qingyang Li, Michael Milham, et al. The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives. *Frontiers in Neuroinformatics*, 7(27):5, 2013.

[16] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[17] Anibal Sólon Heinsfeld, Alexandre Rosa Franco, R. Cameron Craddock, Augusto Buchweitz, and Felipe Meneguzzi. Identification of autism spectrum disorder using deep learning and the abide dataset. *NeuroImage: Clinical*, 17:16–23, 2018. ISSN 2213-1582. DOI: https://doi.org/10.1016/j.nicl.2017.08.017. URL https://www.sciencedirect.com/science/article/pii/S2213158217302073.

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.

[21] Oliver C. Ibe. 10 - diffusion processes. In Oliver C. Ibe, editor, *Markov Processes for Stochastic Modeling (Second Edition)*, pages 295–327. Elsevier, Oxford, second edition edition, 2013. ISBN 978-0-12-407795-9. DOI: https://doi.org/10.1016/B978-0-12-407795-9.00010-4. URL https://www.sciencedirect.com/science/article/pii/B9780124077959000104.

[22] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2021.

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[24] J L Lancaster, M G Woldorff, L M Parsons, M Liotti, C S Freitas, L Rainey, P V Kochunov, D Nickerson, S A Mikiten, and P T Fox. Automated talairach atlas labels for functional brain mapping. *Hum Brain Mapp*, 10(3):120–131, July 2000.

[25] M H Lee, C D Smyser, and J S Shimony. Resting-state fMRI: a review of methods and clinical applications. *AJNR Am J Neuroradiol*, 34(10):1866–1872, August 2012.

[26] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015.

[27] Marius-Constantin Popescu, Valentina Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8, 07 2009.

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[30] Thomas E. Nichols Russell A. Poldrack, Jeanette A. Mumford. *Handbook of Functional MRI Data Analysis*. Cambridge University Press, 2011.

[31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016.

[32] Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview, 2019.

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

[34] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[36] Szabolcs Torma and Dr. Luca Szegletes. Brain signal generation and data augmentation with a single-step diffusion probabilistic model, 2023. URL `https://openreview.net/forum?id=wo0Q5Hb1o0F`.

[37] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric challenges, 2019.

[38] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL `http://www.jmlr.org/papers/v9/vandermaaten08a.html`.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[40] Daniel Zahner and Evangelia Micheli-Tzanakou. 2 artificial neural networks: Definitions, methods, applications. *Supervised and Unsupervised Pattern Recognition: Feature Extraction and Computational Intelligence*, 04 2000. DOI: `10.1201/9781420049770.ch2`.