2023

Graph Learning on Multi-Modality Medical Data to Generate Clinical Predictions

Justin Jiang

Graph Learning on Multi-Modality Medical Data to Generate Clinical Predictions

Justin Jiang

Weiqing Gu, Advisor

Jina Kim, Reader

Department of Mathematics

May, 2023

Abstract

There exist petabytes of data pertaining to medical visits – everything from blood pressure recordings, X-rays, and doctor's notes. Electronic health records (EHRs) organize this data into databases, providing an exciting opportunity for machine learning researchers to dive deeper into analyzing human health. There already exist machine learning models that aim to expedite the process of hospital visits; for example, summary models can digest a patient's medical history and highlight certain parts of their past that merit attention. The current frontier of medical machine learning is combining the various formats of data to generate a clinical prediction – much like a medical professional would. Challenges exist in accessing data, ethical AI, and the inequality that is inherent in our medical system.

This thesis explores the frontier of medical machine learning, particularly how graph learning is used to predict diseases using multi-modality data.

Contents

List of Figures

List of Tables

Acknowledgments

To my parents, grandparents, and family, who have made countless sacrifices that have allowed me to stand where I am today. They are a constant source of inspiration in my life – 妈妈, 爸爸, 阿爷, 阿妈, 外公, 外婆 – to whom I owe most of my success to.

To my friends, who make my life full. This year of college has been amazing because of various friends that I have had the absolute pleasure of spending my time with.

To Professor Gu, who has believed in me and supported me for all of these years. This thesis would not have been possible without her relentless efforts.

Chapter 1

Introduction

The field of health care presents an exciting new source of data for machine learning researchers, as electronic health records (EHR) become more and more prevalent for record keeping in hospitals around the world. These digitized health records provide a wealth of data for the curious machine learning researcher, but obstacles arise in the form of the lack of public benchmarks and inconsistent data handling.

1.1 Objectives

The objective of this thesis is to find the frontier of medical machine learning and apply the latest, state-of-the-art methods to improve existing ones. Specifically, the goal is to build a model that ingests data of different formats – multi-modality data – to predict a clinical output. Further areas of interest include the synthetic generation of unbiased medical data, which tackles multiple ethical obstacles in addition to being a challenging machine learning problem.

1.2 Medical background

As this thesis is an applied machine learning problem, there must be some description on the medical background to give color to the problem at hand. Thus, we will give a brief introduction on data recording, privacy laws, and the ethical issues around using machine learning in medicine.

1.2.1 Electronic health records

The electronic health record (EHR) is a longitudinal electronic record of patient health information generated by one or more encounters in any care delivery setting. EHRs are shared across different health care settings and provide a more complete view of a patient's health history, which has the ability to improve the quality and safety of care and can help to reduce health care costs(Gunter and Terry, 2005).

The widespread adoption of EHRs in the United States is a key goal of the Health Information Technology for Economic and Clinical Health (HITECH) Act, part of the American Recovery and Reinvestment Act of 2009. The HITECH Act provides financial incentives to eligible professionals and hospitals that adopt and meaningfully use certified EHR technology(HHS, 2009).

The use of EHRs has been shown to improve the quality of care; a study published in 2011 found that there was a statistically significant increase in quality of diabetes care between hospitals that kept EHRs versus hospitals that did not(Cebul et al., 2011). In their discussion, the authors mention that in their analysis, they corrected for various biases that exisit in the medical world, including socioeconomic and insured vs uninsured, but also that they might not have corrected for less obvious biases. In later sections, we discuss the problem of inequality in medicine and how we plan to handle these issues.

EHRs are not controversy-free, however: inputting data into a poorly-designed EHR program wastes valuable physician time, data corruption, whether accidental or purposeful, leads to further problems, and privacy concerns are all drawbacks of having a centralized electronic database of personal medical history. For example, the organization the ceritfies American hospitals for operation found that a quarter of the errors that caused a wrong prescription stemmed from a electronic error(**?**).

Regardless of the clinical efficacy of using electronic health records in the medical field, the existence of database is certainty good news for the machine learning researcher, who now can leverage modern, data-dependent techniques to century-old problems.

1.2.2 Machine learning in medicine

Computer-aided diagnostics was first recorded in the 1960s when scientists used early computers to help diagnose blood diseases. Since then, even

as compute power has grown exponentially, the rate of growth in medical machine learning has lagged behind (although, as we argued in Section 1.2.1, the recent adaptation of EHRs has changed the trajectory of growth.).

In the modern era, researchers have begun applying new machine learning methods to several big problems in medicine. In an article published in Nature in 2021, May claims that there are eight different ways that machine learning is applied in a medical context:

Purpose	Description
Reconstructing diseases	Simulate complex diseases and how they interact with new medicines
Hypothesis testing	Create complex statistical models that allow rigorous statistical analysis from incomplete data
Patient recruitment	Find ideal patients for new drug trials by analyzing demographics
Larger datasets	Collate and collect patient data for further analysis
Diagnostic tools	Create diagnostic tools to increase physician efficiency
Prognostic tools	Physicians can use tools to predict prognosis of patients
Patient monitoring	Automated monitoring tools can decrease the need of constant human monitoring
Checks and balances	More research means more conferences, peer reviewers, and a wider reach

Table 1.1 Different ways to apply machine learning in medicine

This thesis will mainly focus on building a model that will serve as the predictive part of a diagnostic or prognostic tool.

Next, we give a few examples of how predictive tools powered by machine learning have been used in the world.

Example: COVID prognosis tools

At the height of the COVID-19 pandemic, many hospitals ran out of Intensive Care Unit(ICU) beds due to the overwhelming volume of patients. As bed space grew scarce, hospitals made difficult choices in deciding who to admit

and who to turn away. Ustebay et al. developed a prognostic model that predicted three outcomes:

1. The need for intensive care
2. Intubation – does the patient need to be put on a ventilator, which were also scarce.
3. Mortality risk

Their aim with this model is to help clinicians make a more informed choice that is rooted in data. To that end, in this era of black box algorithms, they used a class of model known for its interpretability: the random forest.

Their training data came from 11,712 patients admitted to a Istanbul hospital for COVID-related reasons between April 2020 and February 2021 and consisted of several different modalities of data:

Type of Data	Examples
Numerical	age, temperature, heart rate, blood pressure
Categorical	gender, disease history
Time series	electrocardiogram, blood tests

Table 1.2 The different modalities of data that was used as well as examples of each modality

The authors found that the models they created had statistically significant predictive power according to commonly used metrics – an important first step in creating an end-to-end tool that can be used in a hospital.

However, the authors qualify their results, urging caution before acceptance of their model for several reasons:

1. The data was collected from one hospital in Turkey, which is not a representative sample of the population by any means.
2. When there was missing or incomplete data, the authors consulted medical professionals, then artificially filled in the missing data when they deemed necessary. This non-systematic approach could have introduced biases that were further enhanced by the model.
3. There was no validation of the model done on external data, which decreases the quality of the model.

As one can see, this study is an example of the demonstrated need for further research into the ethical considerations of medical data and the practical considerations that follow (such as filling in missing values).

Example: Automatic brain tumor recognition

Magnetic Resonance Images (MRI) are a type of imaging technique used to take high quality images in an non-invasive manner. MRIs and other similar imaging techniques are often used to diagnose brain tumors – to accomplish this, radiologists or other highly specialized clinicians must look at imagery created by these methods. Doing this is an incredibly laborious, experience-dependent, and error-prone task (Fun fact: in 2022, radiologists are the seventh highest paid speciality in medicine(Newitt)).

Recent advancement in neural networks have sparked research into medical image recognition, where models are trained to identify specific parts of images – such as the aforementioned task – with a better accuracy and more consistency across many patients.

In 2017, Bahadure et al. created a model that used Support Vector Machines (a popular type of model in 2017, but now slightly outdated in 2022) to analyze a patient's MRI images and reported the following model metrics: At the time, these results were cutting-edge, but recent results have

Evaluation Metric	Value
Accuracy	96.51%
Specificity	94.2%
Sensitivity	97.72%

Table 1.3 Model metrics from Bahadure et al. (2017)

only pushed the boundary further.

1.2.3 Data

The crux of any machine learning and data science problem is data – usually the more data that is used in training the model, the more reliable the model will be when it encounters unseen data (This is a board claim, and it needs to be qualified in a later section.).

As mentioned before, there are some challenges in accessing high-quality medical data. First, the mere existence of a database that is representative of the population is questionable; most databases, including the generally

accepted benchmark dataset, come from one hospital. Next, most medical data is anonymized due to privacy law, which strips away potentially valuable information. Data that is available for research is often an incomplete record, either due to physician error or the fact that a person visits multiple medical facilities and so creates multiple, often disjoint, electronic health records.

Furthermore, research-quality data is often blocked behind user agreements, which require the completion of a training course and test before access is granted.

Privacy Concerns

Medical data is protected by a variety of federal laws, including the Health Insurance Portability and Accountability Act (HIPAA), the Federal Trade Commission Act (FTCA), and the Health Information Technology for Economic and Clinical Health (HITECH) Act which amended HIPAA in 2009.

These laws were enacted well before the prevalence of electronic health records – the FTCA was signed into law by President Wilson in 1914! In the modern era, applications and smart wearable devices (Apple Watches, Fitbit) can track everything from sleep to resting heart rate to menstrual cycles. Apps are able to sell de-identified data back to a user's employers or to third-party advertisers(Bari (2019)).

Na et al. found that it is quite simple to reconstruct and identify users in data that has been de-identified and anonymized, which once again brings this issue into the medical machine learning field.

There are two sides in the debate to solve this issue. One is to completely anonymize the data: randomly scrambling data values, altering X-rays and MRIs, altering ages and genders. This completely destorys the value of the data in any research context; it is essentially just random data. The other side is to not anonymize the data at all. The reasoning is that if full randomization is impractical, then there is no reason to anonymize the data because techniques to re-identify the data are trivial.

We currently sit somewhere in the middle, where data is slightly anonymized but not enough – any user of the MIMIC datasets has to agree to not attempt to identify patients.

MIMIC

The Medical Information Mart for Intensive Care (MIMIC) database was created by a group of researchers at MIT to be a public resource. This

database also provides a preprocessed and thus standardized starting point for many researchers, while respecting the aforementioned federal privacy laws.

The MIMIC-III dataset is the third iteration of MIMIC, which includes deidentified health-related data from 40,000 patients who were admitted to the critical care units of the Beth Israel Deaconess Medical Center in Boston between 2001 and 2012(Johnson et al. (2016)).

Below is a data description chart that highlight some of the columns in the MIMIC-III dataset (a more through discussion of the data will follow in a later section):

Data	Description
Vital signs	Sampled once an hour, including heart rate and blood pressure
Lab results	The test results from blood work, stool samples, biopsies
Caregiver notes	Written notes about the patient from any caregiver during their stay
Medications	Any medications that the patient is currently on, including ones that were prescribed during the stay
Imaging results	Any image data, i.e. X-ray, MRI
Mortality	The state of the patient after being discharged from the ICU

Table 1.4 Some of the important features of the MIMIC-III dataset

The dataset totals 6.2 gigabytes of data and requires credentialed access on Physionet, an online medical database that researchers use to publish their data and find other datasets to research.

Compared to the data we saw in Section 1.2.2, the MIMIC dataset at least spans a decade of time and covers tens of thousands of patients, but it still only comes from one care center.

1.3 Machine Learning

As this is an introductory chapter, we will not go in-depth into the machine learning models that we plan to use.

Generally, machine learning models try to learn the relationship between input and output. There are a spectrum of models, from simple ones like linear regressions that can only capture a linear relationship between input and output to far more complex models like neural networks that can capture almost any type of relationship between in and out.

For now, we name the models we will use in later sections as well as how we plan to approach this problem.

As will be seen in a later literature review section, the current cutting edge is applying graph neural networks to multi-modality data, and so this is what we will attempt to do as well, albeit with our own flavor.

1.4 Ethical concerns

As a citizen of this world, it is hard not to see the inequality that is prevalent all around us. As someone in a position of power and privilege and the ability to affect the world, I must ensure that I do not contribute more to inequality through this research.

There are immense inequalities and biases in the American medical system: uninsured people have access to far less care, race and gender biases that are prevalent in society also get reflected in medicine, and the difference in the life expectancy of individuals from different socioeconomic backgrounds is telling of the quality of care that money buys. A lot of this inequality is reflected in the data, which I need to ensure – by means described later – does not get further reflected by the models I train.

1.4.1 Synthetic Data Generation

In recent years, the need to address inequality in medicine has become increasingly more pressing. While individuals from marginalized backgrounds are receiving inadequate access to healthcare, technology is opening up new possibilities to solve this unequal distribution of care. Synthetic medical data, or SMD, can be utilized to reduce inequality in medicine by providing more accurate and equitable access to patient data and healthcare services.

SMD is a type of computer-generated data that mimics the characteristics of real-world data. It is produced through algorithms which can generate data sets with the same characteristics as real-world data, such as patient demographics, treatments and outcomes. SMD can be used to help identify

disparities in health care, such as access to care, quality of care and cost of care. It can also help to identify emerging healthcare trends and better understand the needs of populations that suffer from unequal access to healthcare.

The use of SMD can help improve the delivery of healthcare services by providing more accurate and equitable access to patient data. By generating data sets that are more reflective of the real-world, SMD can provide a more accurate picture of the patient population, including their demographic information and health conditions. This data can then be used to better inform healthcare decisions, such as which treatments and services should be provided and to which individuals. In turn, this can help healthcare providers provide more equitable access to care and reduce disparities in health care.

However, synthetic data is very much new, and with anything this powerful, it requires investigation – which we will do in a later section of this paper.

1.4.2 Commercialization

Recently, companies have started to commercialize research, creating tools from the models that researchers pioneered. GNS Healthcare, an American healthcare company, recently released Gemini, a computer program that simulates multiple myeloma to study drug progression, something they call "The in silico Patient™", as seen in Image 1.1.

Thus, as researchers, we have to be aware of the current commercialization efforts of our work.

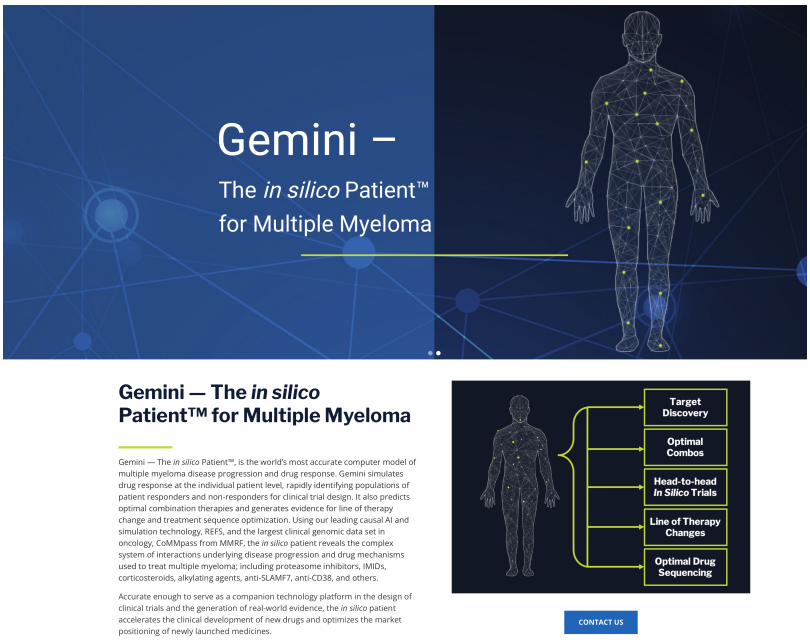

Figure 1.1 A screenshot from the website of GNS Healthcare, taken Dec 2022

Chapter 2

Problem Statement

In this chapter, I will go over the overarching research themes of this thesis then introduce the goals of this research.

2.1 Statement

Medical machine learning has taken off in the last few years, as recent advances in machine learning have enabled researchers to apply powerful new models to the field. One subset of models – graph learning – has advanced so far and that has become so popular in recent years (2.1) that the number of papers about medical machine learning that use graph learning has grown over the last five years (see 2.2).

This thesis explores how new advances in graph learning are being applied to tackle long standing problems in medical machine learning, such as mesh manifolds derived from MRI images and incorporating multi-modality data. Then, original work is presented in training a graph model on the MIMIC-III dataset in a multi-class prediction problem: disease prediction.

This work involves adapting a model from a recent paper and exploring how changes to the model architecture and loss functions change the results.

Finally, this thesis describes future work in this subspace of the graph learning and medical machine learning field of study. The next section contains a succinct description of the goals of this thesis.

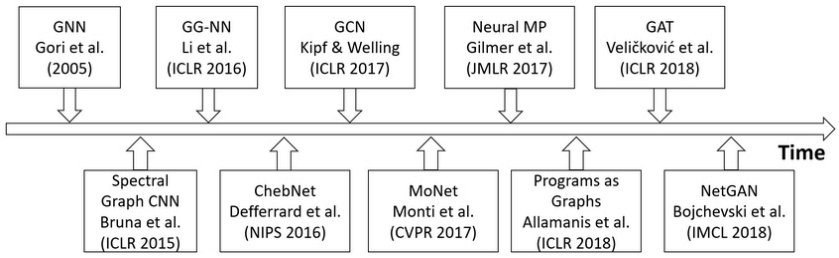

Figure 2.1 A brief timeline of big developments in graph learning. Taken from Xia et al. (2021).

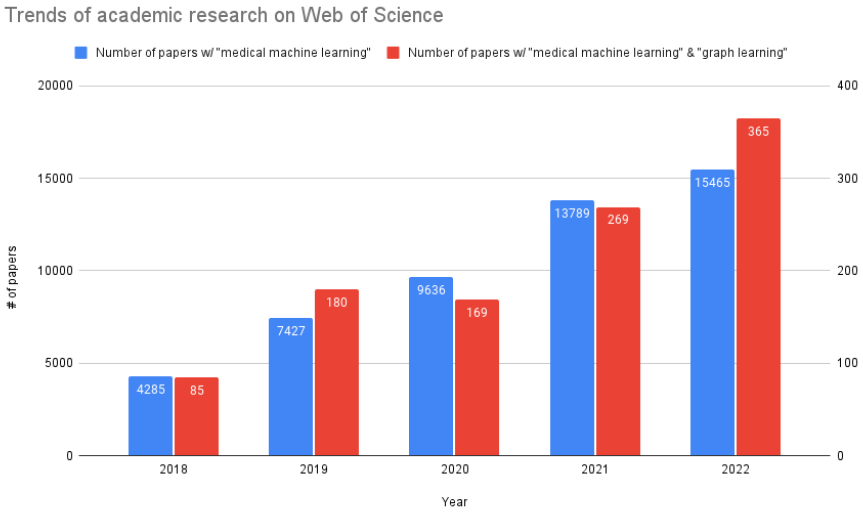

Figure 2.2 The number of papers on medical machine learning has steadily grown over the last five years, similar to the growth of medical machine learning papers that mention graph learning. Taken from a search on Web of Science

2.2 Goals

This section outlines the overarching topics that this thesis covers:

- **Review of current literature** A review of current methods that are popular in literature as well as a few in-depth analysis of several papers that will be important to my own research.
- **Research and Results** The research described in later sections of this thesis consists of adapting an existing model to work on a different data set with a different predictive target. Challenges that arise from this include creating a new data pre-processing pipeline and handling sparsity in the target variable.
- **Closing Remarks** Finally, the last act of this thesis includes typical ending materials, including a conclusion, a future works section, and relevant appendicies.

Chapter 3

Background

3.1 Medicine

This section is a non-technical introduction to give background to the medical terms needed to understand the aims of this mathematics thesis.

3.1.1 Disease Prediction

Disease prediction has a long and rich history, dating back to the early days of medicine. The roots of disease prediction can be traced back to the times of Hippocrates (*c.* 460 - *c.* 370 BC), who is considered the father of modern medicine. Hippocrates emphasized the importance of observing symptoms and making predictions based on these observations. In the centuries that followed, advancements in medical knowledge and technology have significantly improved our ability to predict diseases. In this subsection, we briefly discuss the milestones in the history of disease prediction.

1. **18th century:** The foundation of modern epidemiology was laid by John Graunt in the 17th century, and the discipline matured in the 18th century with the work of James Lind and Edward Jenner. These early epidemiologists used observational data to infer relationships between environmental factors and the occurrence of diseases.

2. **19th century:** In the mid-1800s, the French physician Pierre Charles Alexandre Louis pioneered the use of statistical analysis in medicine. He introduced the concept of numerical analysis to compare the outcomes of different treatments, laying the groundwork for evidence-based medicine.

3. **20th century:** The development of modern-day disease prediction models started in the 20th century. In the 1950s, the Framingham Heart Study, a longitudinal study investigating the risk factors for cardiovascular disease, initiated the era of risk prediction modeling in medicine.

   The Framingham Heart Study stands out as a pioneering and groundbreaking research initiative that has greatly advanced our understanding of the epidemiology of cardiovascular disease. Initiated in 1948 by the U.S. Public Health Service in collaboration with Boston University, the study was designed to investigate the factors that contribute to the development of heart disease and stroke. The town of Framingham, Massachusetts, was chosen as the study location as it was considered representative of the general U.S. population at the time.

   The initial cohort of the study included 5,209 adult participants, aged between 30 and 62, who were followed prospectively for their health outcomes. The researchers collected detailed information on various factors, including demographic characteristics, lifestyle habits, and medical history, to understand their impact on the development of cardiovascular disease. The study participants were examined and monitored every two years, resulting in a wealth of longitudinal data that has been invaluable for epidemiological analysis. Over the years, the Framingham Heart Study expanded to include the offspring of the original participants and their spouses (Offspring cohort), and later, the third-generation cohort.

   The Framingham Heart Study has made several important contributions to our knowledge of cardiovascular disease risk factors. It was the first study to identify the major risk factors for heart disease, namely high blood pressure, high blood cholesterol, smoking, obesity, diabetes, and physical inactivity. In the early 1960s, the Framingham study introduced the concept of risk factors, which has since become an essential part of preventive medicine. Moreover, the study led to the development of the Framingham Risk Score, a widely used tool for estimating the 10-year risk of developing coronary heart disease. This risk prediction model has been fundamental in guiding medical professionals to identify individuals at high risk and manage their conditions more effectively. With the help of the Framingham Heart Study, medical professionals have established policies and guidelines for reducing the burden of cardiovascular disease worldwide. In

conclusion, the Framingham Heart Study has had a profound and lasting impact on the field of disease prediction and has become a model for many subsequent longitudinal studies in medicine.

This study led to the development of the Framingham Risk Score, which became the foundation for many contemporary risk prediction models.

4. **21st century:** The rise of computer technology and the increasing availability of electronic health records transformed the field of disease prediction. Machine learning techniques became increasingly popular in the 1990s, with researchers applying algorithms such as artificial neural networks, decision trees, and support vector machines to medical data for predicting diseases. We will go into further detail about contemporary techniques in the later sections.

In the 21st century, the rapid growth of computational power and the availability of big data have empowered researchers to develop more sophisticated and accurate disease prediction models. Graph Machine Learning, the focus of this thesis, represents one of the most promising and cutting-edge approaches in this domain. It leverages the inherent relational structure present in many medical data sets to model complex relationships and make accurate predictions. In the following sections, we delve deeper into the fundamentals of Graph Machine Learning and its medical applications, particularly in disease prediction.

3.1.2 Electronic Health Records

Electronic Health Records (EHRs) have revolutionized the way healthcare professionals and researchers collect, store, and analyze patient data. The advent of EHRs has led to improved coordination of care, increased efficiency, and more accurate disease prediction models. In this subsection, we will explore the history of EHRs and discuss recent advancements that have shaped their current state.

Early History:

The concept of EHRs dates back to the 1960s when researchers began developing computerized systems to manage patient information. However, the adoption of EHRs was initially slow due to technological limitations

and high costs. One of the earliest and most influential EHR systems is the Computer-Stored Ambulatory Record (COSTAR), which was developed at Massachusetts General Hospital in the early 1970s. COSTAR provided a foundation for subsequent EHR systems by demonstrating the potential benefits of computerized patient data management.

Increased Adoption and Standardization:

The widespread adoption of EHRs began in the late 1990s and early 2000s, spurred by advancements in computer technology and the internet. The need for standardization in EHR systems became evident, leading to the development of Health Level 7 (HL7), a set of international standards for the exchange, integration, sharing, and retrieval of electronic health information. These standards played a crucial role in promoting interoperability and data exchange among different EHR systems.

Government Incentives and Regulations:

The adoption of EHRs gained momentum in the late 2000s due to government incentives and regulations. In the United States, the Health Information Technology for Economic and Clinical Health (HITECH) Act, enacted in 2009, provided financial incentives for healthcare providers to adopt EHR systems. Around the same time, several countries, including the United Kingdom, Canada, and Australia, also implemented national EHR initiatives to encourage the adoption of EHR systems among healthcare providers.

In recent years, the widespread use of EHRs has given rise to several advancements that have further improved the functionality and effectiveness of these systems. Some notable recent advancements are included in Table 3.1

The history of Electronic Health Records is characterized by the incremental development and adoption of computerized systems to manage patient data. The evolution of EHRs has been driven by advancements in technology, standardization, and government incentives. In recent years, the integration of wearable devices, the application of AI and ML techniques, and the improvement of interoperability have emerged as major trends that will likely shape the future of EHRs and their role in disease prediction and health care.

Advancement	Description
Integration with Wearable Devices	With the proliferation of wearable health devices, such as fitness trackers and smartwatches, EHR systems are now being integrated with these devices, enabling real-time monitoring and data collection. This integration provides healthcare professionals with more comprehensive and up-to-date patient data, ultimately improving patient outcomes and facilitating personalized medicine.
Application of Machine Learning	The vast amount of data available in EHRs has opened the door for the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques. Researchers are leveraging AI and ML algorithms to develop more accurate disease prediction models, optimize treatment planning, and identify potential outbreaks of infectious diseases based on EHR data.
Interoperability and Health Information Exchange	Efforts to improve the interoperability of EHR systems have led to the development of health information exchange (HIE) networks, which facilitate the secure and efficient sharing of patient data between different healthcare providers. These networks have significantly improved the coordination of care, especially in cases where patients receive treatment from multiple providers or facilities.

Table 3.1 Recent advancements in EHRs

3.1.3 Data Concerns: Ethical Implications

The increasing availability and use of health data, including Electronic Health Records and data from wearable devices, present both opportunities and challenges in medicine. While these data sources have substantially contributed to advancements in diagnostics, treatment, and predictive analytics, they also raise concerns about data privacy, security, and potential misuse for unethical purposes. In this subsection, we explore some of the critical data concerns in medicine, focusing on the potential for data misuse and the associated ethical implications.

Data Privacy

Ensuring the privacy of patients' medical data is paramount, as the disclosure of sensitive health information can lead to both immediate and long-term harm. Patients may be hesitant to share certain information with healthcare providers if they believe their data is not adequately protected, which can ultimately impact their quality of care. The misuse of health data for unauthorized purposes, such as marketing, can also breach patients' privacy and erode trust in the healthcare system.

Data Security

The storage and transfer of health data must be secure to prevent unauthorized access and potential misuse. Healthcare providers and researchers are responsible for implementing robust security measures to safeguard patient data from cyber threats, such as hacking, data breaches, and ransomware attacks. The consequences of data breaches can be severe, as they may expose sensitive patient information, lead to financial losses, and damage the reputation of the affected organization.

Discrimination and Bias

The misuse of health data for unethical purposes can result in discrimination and bias against certain individuals or groups. For example, insurance companies may use health data to deny coverage or charge higher premiums to individuals with pre-existing conditions or certain genetic predispositions. Additionally, biases present in health data, such as under-representation of certain minorities, can lead to the development of biased prediction models or treatment algorithms, perpetuating health disparities.

Exploitation of Data

Medical data may be misused for financial gain or to further goals that are not aligned with the best interests of patients. For example, pharmaceutical companies might use health data to selectively target patients for marketing purposes, promoting unnecessary or harmful treatments. The commercialization of health data also raises ethical concerns about patient consent, data ownership, and the equitable distribution of benefits derived from the data.

The increasing use of health data in medicine brings to the fore various concerns regarding data privacy, security, and potential misuse for unethical purposes. Healthcare providers, researchers, and policymakers must address these concerns by implementing appropriate safeguards, promoting transparency, and adhering to ethical guidelines and regulations. Ensuring the responsible use of health data is critical for maintaining public trust and realizing the full potential of health data in improving patient care and outcomes.

3.2 Machine Learning

Machine Learning (ML) focuses on the development of algorithms that can learn from and make predictions or decisions based on data. Neural networks, inspired by the biological neural networks of the human brain, are a class of ML models that have demonstrated great success in a wide range of applications. In this section, we provide an overview of the fundamentals of machine learning, with a particular emphasis on neural networks and their relevance to medical applications.

3.2.1 Basics of Machine Learning

Machine Learning can be broadly divided into three categories: supervised learning, unsupervised learning, and reinforcement learning.

Supervised Learning

In supervised learning, the algorithm is trained on a labeled dataset, where each input data point is associated with a corresponding output or label. The main goal of supervised learning is to learn a mapping from inputs to outputs that can be used to make predictions on previously unseen data. Common supervised learning tasks include regression (predicting

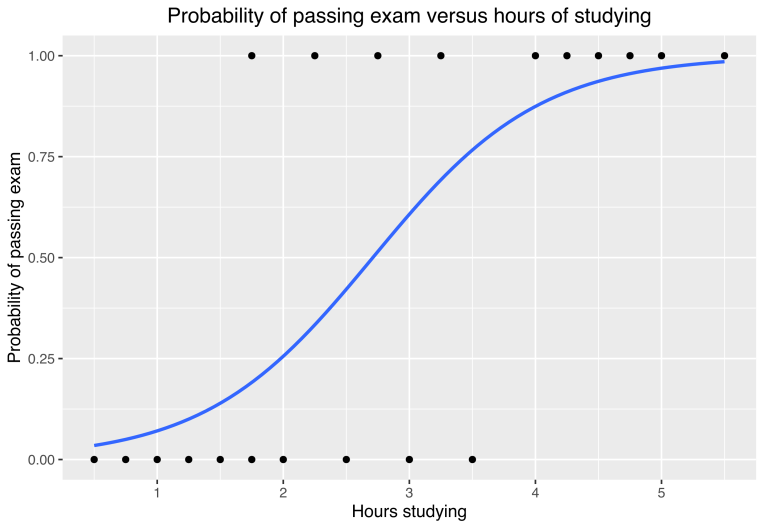

Figure 3.1 A univariate logistic regression, taken from wik (2023)

a continuous-valued output) and classification (predicting discrete class labels) as seen in Figure 3.1.

Logistic Regression

Logistic regression is a supervised machine learning algorithm used for binary classification problems. It aims to model the relationship between a set of input features and a binary target variable by estimating the probability that an observation belongs to a particular class. Logistic regression is a linear model that uses the logistic function as its activation function to ensure that the output probabilities lie between 0 and 1.

The Logistic Function

The logistic function, also known as the sigmoid function, is used to map any input value to a value between 0 and 1. The logistic function is defined as:

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \tag{3.1}$$

where z is the input to the logistic function. The logistic function maps positive values of z to probabilities greater than 0.5 and negative values of z

to probabilities less than 0.5.

The Logistic Regression Model

Given a dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i \in \mathbb{R}^p$ is a feature vector and $y_i \in \{0, 1\}$ is the corresponding binary target variable, logistic regression models the probability of y_i being 1, given x_i, as follows:

$$P(y_i = 1|x_i) = \sigma(w^T x_i + b) = \frac{1}{1 + \exp(-(w^T x_i + b))} \tag{3.2}$$

where $w \in \mathbb{R}^p$ is the weight vector, $b \in \mathbb{R}$ is the bias term, and $w^T x_i + b$ is the linear combination of the input features. The probability of y_i being 0 is given by:

$$P(y_i = 0|x_i) = 1 - P(y_i = 1|x_i) = 1 - \sigma(w^T x_i + b) \tag{3.3}$$

Maximum Likelihood Estimation

The parameters of the logistic regression model, w and b, are estimated using maximum likelihood estimation (MLE). The likelihood function for logistic regression is defined as the product of the probabilities of the target variables, given the input features and the model parameters:

$$L(w, b|D) = \prod_{i=1}^{n} P(y_i|x_i) = \prod_{i=1}^{n} \left(\sigma(w^T x_i + b)\right)^{y_i} \left(1 - \sigma(w^T x_i + b)\right)^{1-y_i} \tag{3.4}$$

To find the maximum likelihood estimates of the parameters, we maximize the log-likelihood function:

$$l(w, b|D) = \log L(w, b|D) = \sum_{i=1}^{n} \left[y_i \log \sigma(w^T x_i + b) + (1 - y_i) \log \left(1 - \sigma(w^T x_i + b)\right) \right] \tag{3.5}$$

Model Training

Since there is no closed-form solution for the MLE of the logistic regression parameters, we use numerical optimization algorithms, such as gradient descent, to find the optimal w and b that maximize the log-likelihood

function. The gradient of the log-likelihood function with respect to the parameters is given by:

$$\frac{\partial l}{\partial w} = \sum_{i=1}^{n} \left[y_i - \sigma(w^T x_i + b) \right] x_i \tag{3.6}$$

$$\frac{\partial l}{\partial b} = \sum_{i=1}^{n} \left[y_i - \sigma(w^T x_i + b) \right] \tag{3.7}$$

Model Evaluation

Once the logistic regression model is trained, it can be used to predict the binary class labels of new data points. The predicted class label $\hat{y}$ can be computed by thresholding the predicted probability at 0.5:

$$\hat{y} = \begin{cases} 1, & \text{if } \sigma(w^T x + b) \geq 0.5 \\ 0, & \text{if } \sigma(w^T x + b) < 0.5 \end{cases} \tag{3.8}$$

The performance of the logistic regression model can be evaluated using various classification metrics, such as accuracy, precision, recall, F1 score, and the area under the receiver operating characteristic (ROC) curve.

Unsupervised Learning

In unsupervised learning, the algorithm is trained on an unlabeled dataset, and the objective is to uncover hidden patterns or structures within the data. Typical unsupervised learning tasks include clustering (grouping similar data points together) and dimensionality reduction (reducing the number of features while preserving the structure of the data).

K-means Clustering

K-means clustering is an unsupervised machine learning algorithm used for partitioning a dataset into k distinct clusters, where each data point belongs to the cluster with the nearest mean. The mean of each cluster is also known as its centroid. The main objective of the algorithm is to minimize the within-cluster sum of squares (WCSS), defined as:

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \tag{3.9}$$

where k is the number of clusters, C_i represents the i-th cluster, x is a data point in the i-th cluster, and μ_i is the centroid of the i-th cluster.

The k-means clustering algorithm consists of the following steps:

1. Initialize the centroids randomly by selecting k data points from the dataset.
2. Assign each data point to the nearest centroid, forming k clusters.
3. Update the centroids by calculating the mean of all data points within each cluster.
4. Repeat steps 2 and 3 until the centroids no longer change or a predefined stopping criterion is met.

It is important to note that the k-means algorithm is sensitive to the initial placement of centroids. In practice, it may be necessary to run the algorithm multiple times with different initializations and choose the result with the lowest WCSS.

The choice of k is a crucial factor in determining the quality of clustering. A common technique for selecting the optimal k is the Elbow method, in which the WCSS is calculated for different values of k and plotted against k. The optimal value of k is typically the one at which the plot shows an "elbow" or a sharp decrease in the WCSS.

K-means clustering is widely used in various fields such as image segmentation, customer segmentation, document clustering, and anomaly detection. Its simplicity and scalability make it a popular choice for clustering tasks with large datasets. However, k-means has some limitations, such as sensitivity to initial conditions, inability to handle categorical features, and the requirement to specify k beforehand.

Manifold Learning

Manifold learning is a class of unsupervised machine learning techniques that aim to discover the low-dimensional structure, called a manifold, embedded in a high-dimensional dataset. This is motivated by the observation that real-world high-dimensional data often lie on or near a low-dimensional manifold, capturing the essential structure and relationships between data points. Manifold learning methods seek to exploit this low-dimensional structure to perform dimensionality reduction, visualization, and clustering tasks.

Mathematically, a manifold $\mathcal{M}$ is a topological space that locally resembles a Euclidean space. Let $X \subset \mathbb{R}^D$ be a D-dimensional dataset, and suppose that the dataset lies on a d-dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^D$, where $d \ll D$. The central problem of manifold learning is to learn a mapping:

$$\phi : X \to \mathcal{M} \tag{3.10}$$

which projects the high-dimensional data points in X onto the low-dimensional manifold $\mathcal{M}$. This mapping must preserve the intrinsic geometric properties and relationships between data points in the original space.

Various manifold learning techniques have been proposed, each with its unique approach for preserving different aspects of the data's geometry. Some popular manifold learning methods include:

- Principal Component Analysis (PCA): PCA is a linear dimensionality reduction technique that projects the data onto the directions of maximum variance while preserving the Euclidean distances between data points. However, PCA cannot capture the non-linear structure of manifolds.

- Isomap: Isomap is an extension of PCA that preserves the geodesic distances between data points on the manifold. It constructs a neighborhood graph based on the k-nearest neighbors and estimates the geodesic distances between data points using shortest path algorithms. Then, it applies classical multidimensional scaling (MDS) to the geodesic distance matrix to obtain the low-dimensional embedding.

- Locally Linear Embedding (LLE): LLE reconstructs the local linear relationships between data points in the low-dimensional space. For each data point, LLE computes the weights that best reconstruct the point from its neighbors in the high-dimensional space. Then, it finds a low-dimensional embedding that preserves these reconstruction weights.

- t-Distributed Stochastic Neighbor Embedding (t-SNE): t-SNE is a non-linear dimensionality reduction technique that focuses on preserving the probability distribution of pairwise distances between data points in the high-dimensional and low-dimensional spaces. It minimizes the

Kullback-Leibler divergence between the two probability distributions using gradient descent.

Manifold learning methods have been widely applied in various fields, including image and speech recognition, genomics, neuroscience, and visualization of high-dimensional data. However, these methods have their limitations, such as sensitivity to noise and hyperparameters, difficulty in handling out-of-sample data points, and computational complexity for large datasets.

t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear dimensionality reduction technique that is particularly suitable for visualizing high-dimensional data in two or three dimensions. It was introduced by Laurens van der Maaten and Geoffrey Hinton in 2008. The main idea of t-SNE is to preserve the pairwise similarities between data points in the high-dimensional space while mapping them to a lower-dimensional space.

To achieve this, t-SNE defines a probability distribution over pairs of high-dimensional data points in such a way that similar data points have a high probability of being picked, and dissimilar data points have a low probability. Similarly, it defines a probability distribution over pairs of low-dimensional data points. The main goal of t-SNE is to minimize the divergence between these two probability distributions.

High-dimensional pairwise similarities

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ of high-dimensional data points, t-SNE first computes the pairwise conditional probabilities $p_{j|i}$:

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{3.11}$$

where σ_i^2 is the variance of the Gaussian that is centered on data point x_i. The value of σ_i is chosen for each data point such that the perplexity of the conditional distribution P_i matches a predefined perplexity value. The pairwise similarity between data points x_i and x_j is then defined as the symmetrized version of the conditional probabilities:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{3.12}$$

Low-dimensional pairwise similarities

Next, t-SNE defines a probability distribution over pairs of low-dimensional data points $Y = \{y_1, y_2, \ldots, y_n\}$ using a Student's t-distribution with one degree of freedom (also known as the Cauchy distribution):

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}} \tag{3.13}$$

Minimization of Kullback-Leibler divergence

The objective of t-SNE is to minimize the divergence between the high-dimensional pairwise similarities p_{ij} and the low-dimensional pairwise similarities q_{ij}. The Kullback-Leibler (KL) divergence is employed as the measure of divergence, and the minimization problem can be defined as follows:

$$C(Y) = KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3.14}$$

The KL divergence is minimized using gradient descent. The gradient of the KL divergence with respect to y_i is given by:

$$\frac{\delta C}{\delta y_i} = 4 \sum_{j \neq i} (p_{ij} - q_{ij})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1} \tag{3.15}$$

Optimization

The optimization process of t-SNE can be computationally expensive, particularly for large datasets. To mitigate this issue, the Barnes-Hut algorithm is employed to approximate the gradients, reducing the computational complexity from $O(n^2)$ to $O(n \log n)$. Also, early exaggeration and momentum-based optimization methods are used to improve the convergence of the algorithm.

Applications and limitations

t-SNE is widely used for visualizing high-dimensional data, such as gene expression data, images, speech signals, and text corpora. It has shown to be effective in revealing the underlying structure and clusters in the data.

However, t-SNE has some limitations:

- The algorithm is sensitive to hyperparameters, such as the perplexity and the learning rate.
- It does not provide a unique solution, as the random initialization of the low-dimensional data points can lead to different embeddings across multiple runs.
- The computational complexity and memory requirements can be challenging for large datasets, despite the use of approximation algorithms.
- It does not provide a straightforward way to incorporate new data points or perform out-of-sample extensions.

Reinforcement Learning

Reinforcement learning involves training an algorithm (known as an agent) to learn an optimal policy for making decisions based on the feedback received from the environment. The agent learns from trial and error, receiving rewards or penalties for each action it takes, and aims to maximize the cumulative reward over time.

3.2.2 Intro to Neural Networks

Neural networks, also known as artificial neural networks (ANNs), are a class of machine learning models inspired by the structure and functioning of the human brain. ANNs consist of interconnected nodes, or artificial neurons, that are organized in layers. The simplest form of a neural network is the single-layer perceptron, which consists of an input layer and an output layer. However, most practical applications utilize more complex architectures, known as deep neural networks (DNNs), that consist of multiple hidden layers between the input and output layers.

The power of neural networks lies in their ability to approximate complex, nonlinear relationships between input and output variables. During the training process, the weights of the connections between the neurons are adjusted so as to minimize a defined loss function representing the difference between the predicted outputs and the actual labels. The optimization is typically performed using gradient-based techniques, such as stochastic gradient descent (SGD) or its variants.

A neural network usually consists of these components seen in Table 3.2; a typical neural network can be seen in Figure 3.4.

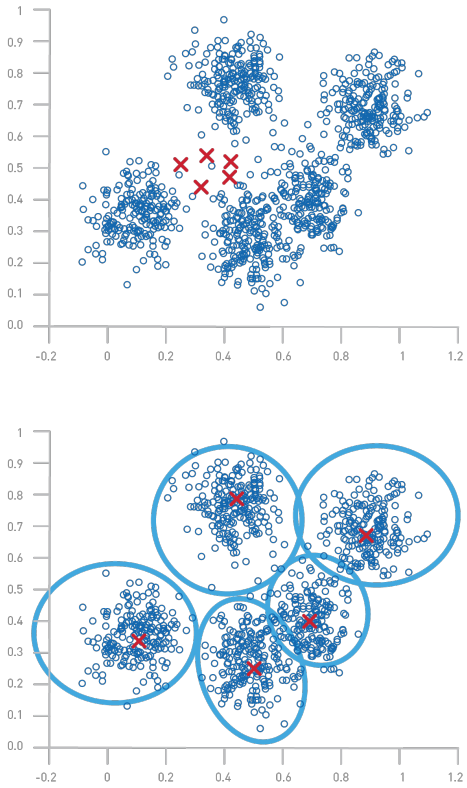

Figure 3.2 A visualized example of k-means clustering, taken from here. The image on the bottom shows the five distinct clusters that form after iterations of the algorithm.

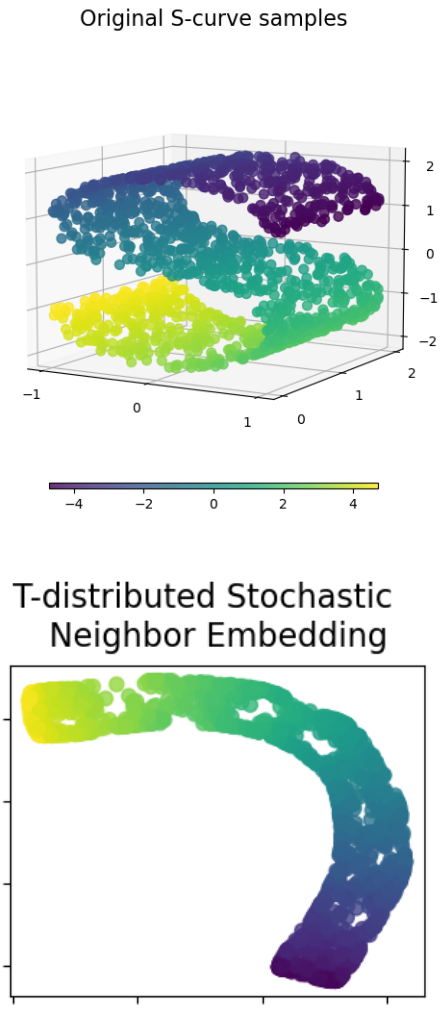

Figure 3.3 *Top*: A plot of a 3-D S-curve, where local neighborhoods are labelled differently, as show by the colors. *Bottom*: A visualization of the result of t-SNE; note how the original local neighborhoods are preserved after dimensionality reduction. Both images taken from scikit-learn.

Component	Description
Node	Nodes are the building blocks of neural networks; modeled after neurons in human brains, nodes are where computation takes place in a neural network.
Layer	Layers organize nodes into separate strata where nodes pass information between each layer. The topmost layer of a neural network is called the *input layer*, and the last layer is called the *output layer*. Layers between these two layers are called *hidden layers*.
Input	Each node has an input; for nodes in the input layer, this input is from the data, while nodes in the other layers are outputs from other nodes.
Weights	Weights are applied to a node's inputs to either amplify or dampen the signal of that node; mathematically, weights are usually multiplied with the input and the node outputs this result.
Activation function	An activation function changes the output of the node; usually, neural networks use a non-linear activation function to enable the model to learn non-linear relationships. A popular non-linear activation function is the rectified linear unit as seen in 3.16: $$f(x) = x^+ = \begin{cases} x & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.16)$$

Table 3.2 Common parts of a neural network.

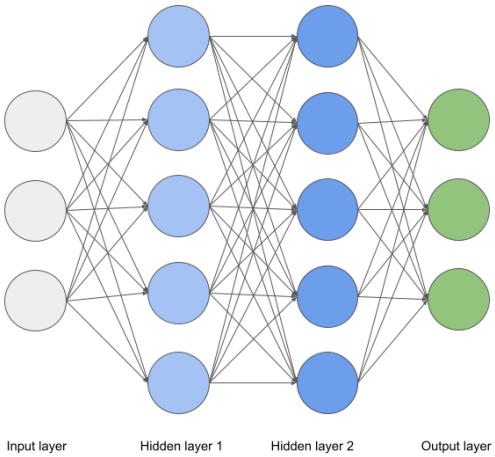

Figure 3.4 A feed-forward neural network, where connections move "down" the layers.

Math Behind a Neural Network

Here, we will briefly go into the mathematics behind a neural network.

To start, we will describe the simplest neural network, which consists of n inputs to one node with one output. This is called a perceptron, as described in Rosenblatt (1958).

First we define our set of inputs as $X = \{X_1 \dots X_n\}$ and our corresponding set of weights $W = \{w_1 \dots w_n\}$. We add a bias term b to the dot product of X and W, as seen in Equation 3.17. Note the similarity of this equation to a linear function.

$$Z = X \cdot W + b \tag{3.17}$$

The output Z in 3.17 would be a linear output which would make our model just a linear model, and so a non-linear "activation function" is applied to Z to introduce non-linearity. Examples of non-linear activation functions include 3.16 and 3.18

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \hat{y} \tag{3.18}$$

The output we get from our activation function $\hat{y}$ is the prediction from our network.

Our goal is to "train" our model so that $\hat{y}$ is as close to y as possible given our input data; mathematically speaking, we want to reduce the loss of $\hat{y} - y$. To do so, we choose a loss function, a function that approximates the difference between the model's prediction $\hat{y}$ and the ground truth y; in our example, we will use a simple loss function C: mean square error as defined in Equation 3.19.

$$C = MSE = \frac{1}{n}\sum^{n}(y_i - \hat{y}_i)^2 \tag{3.19}$$

Since we tune our model's output by adjusting the weights W, we take the partial derivative of C with respect to W. We save the computation of this derivative for the appendix (see A.1) and show the result in Equation 3.20.

$$\frac{\partial C}{\partial w_i} = \frac{2}{n}\sum(y - \hat{y}) \cdot \sigma(z) \cdot (1 - \sigma(z)) \cdot x_i \tag{3.20}$$

Finally, we're able to update the weights in our model with gradient descent as seen in Equation 3.21:

$$w_i = w_i - (\alpha \cdot \frac{\partial C}{\partial w_i}) \tag{3.21}$$

where $alpha$ is a user chosen "hyperparameter". In our simple model, we update w_i iteratively until the loss from the model given the training inputs reaches convergence.

Neural networks are composed of layers consisting of nodes that function similarly to how the perceptron works. Ultimately, a neural network is a powerful tool that can detect trends in non-linear datasets, albeit with a few drawbacks as described in Section 3.2.4.

3.2.3 Neural Networks in Medical Applications

Neural networks have been widely applied in various medical domains due to their capacity to learn complex relationships from large and high-dimensional datasets. Some prominent applications of neural networks in medicine include:

Disease Prediction:

Neural networks have been employed to develop prediction models for various diseases, such as diabetes, cancer, and cardiovascular disease, based

on patient data from EHRs, medical imaging, and genomic information.

Medical Image Analysis:

Convolutional Neural Networks (CNNs), a specialized type of neural network designed for processing grid-like data such as images, have demonstrated remarkable success in medical image analysis tasks, including segmentation, classification, and object detection.

Natural Language Processing:

Recurrent Neural Networks (RNNs) and Transformer-based models, which are capable of processing sequential data, have been utilized for various natural language processing tasks in medicine, such as the extraction of medical information from unstructured clinical text or the generation of medical reports from imaging data.

3.2.4 Drawbacks of Neural Networks

3.2.5 Graphs

A graph is a versatile and widely used mathematical structure for representing pairwise relationships between objects or entities in various domains. In this subsection, we provide an overview of the basic graph structure and discuss how graphs can be used to model real-world data.

Graph Structure

A graph G can be formally defined as an ordered pair $G = (V, E)$, where:

- V is a set of nodes (or vertices), representing the entities or objects in the graph. Each node $v \in V$ can be associated with an index or label, as well as a set of attributes or features describing the entity.
- $E \subseteq V \times V$ is a set of edges, representing the relationships or connections between the nodes. Each edge $e \in E$ can be represented as an ordered or unordered pair of nodes, (u, v), indicating a connection between nodes u and v. Edges can also be associated with weights or attributes that represent the strength or characteristics of the relationship.

Graphs can be either directed or undirected, depending on the nature of the relationships they represent. In directed graphs, edges have a direction, and the relationships are considered asymmetric (e.g., following someone on social media). In contrast, undirected graphs represent symmetric relationships, where an edge connecting two nodes implies a bidirectional relationship (e.g., friendship in a social network).

Modeling Real-World Data with Graphs

Graphs can be used to represent a wide range of real-world data in various domains. The key advantage of using graphs to model data is their ability to capture complex relationships between entities naturally and intuitively. In Table 3.3 are some examples of how graphs can model real-world data in different contexts.

Graph Neural Networks

Graph Neural Networks (GNNs) are a class of deep learning models specifically designed to handle graph-structured data. Graphs are a versatile data structure that can represent complex relationships between entities, such as social networks, citation networks, molecular structures, and transportation systems. Traditional deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are not well-suited for processing graph-structured data due to their grid-like structure and the irregularity of graphs. This has led to the development of GNNs as a powerful tool to learn meaningful representations of nodes, edges, and entire graphs.

Graph Representation

A graph G consists of a set of nodes $V = \{v_1, v_2, \ldots, v_n\}$ and a set of edges $E = \{(v_i, v_j) | 1 \leq i, j \leq n\}$. Each node $v_i \in V$ can be associated with a feature vector $x_i \in \mathbb{R}^d$, and each edge $(v_i, v_j) \in E$ can be associated with an edge feature or a scalar weight. A graph can be represented as an adjacency matrix $A \in \mathbb{R}^{n \times n}$, where the entry A_{ij} is the weight of the edge between nodes v_i and v_j, or 0 if there is no edge. The node features can be represented as a feature matrix $X \in \mathbb{R}^{n \times d}$.

Example	Application of Graphs
Social Networks	In a social network, nodes represent individuals, and edges represent relationships or interactions between them, such as friendship, communication, or collaboration. Graph-based models can be used to analyze the network structure and identify communities, influential individuals, or potential connections.
Biological Networks	In biological systems, graphs can model various types of interactions, such as gene-gene or protein-protein interactions, metabolic pathways, or cellular signaling networks. Nodes represent genes, proteins, or other biomolecules, and edges represent functional associations or physical interactions between them. Graph-based analyses can help uncover important biomolecules, functional modules, or disease-related pathways.
Transportation Networks	In transportation systems, graphs can model the connectivity of roads, railways, or flight routes. Nodes represent cities, airports, or intersections, and edges represent roads, railway tracks, or flight connections between them. Graph-based algorithms can be utilized to optimize routing, traffic flow, or infrastructure planning.
Medical Data	Graphs can model various types of medical data, such as patient similarity networks, medical knowledge graphs, or brain connectivity networks. For example, in a patient similarity network, nodes represent patients, and edges encode similarities based on clinical, genetic, or imaging data. Graph-based models can be employed to identify disease subtypes, predict patient outcomes, or recommend personalized treatment strategies.

Table 3.3 A list of real-world applications of graphs.

Message Passing Framework

The core idea behind GNNs is to learn node representations by aggregating the information from the local neighborhood of each node in the graph. This is achieved through a message-passing framework that consists of the following steps:

1. Each node sends a message to its neighbors. The message is a function of the node's features and the edge features.
2. Each node aggregates the received messages from its neighbors.
3. Each node updates its features based on the aggregated messages.
4. The message passing process is repeated for a fixed number of iterations or until convergence.

Graph Convolutional Networks (GCNs)

Graph Convolutional Networks (GCNs) are a popular class of GNNs that extend the concept of convolutions from grid-like data to graph-structured data. In a GCN, the feature update rule for each node is defined by a convolution operation on its local neighborhood. The key equation for a GCN layer is given by:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}) \tag{3.22}$$

where $H^{(l)} \in \mathbb{R}^{n \times d_l}$ is the feature matrix at layer l, $W^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ is the learnable weight matrix for layer l, $\sigma(\cdot)$ is an activation function, such as ReLU, and $\hat{A}$ is the normalized adjacency matrix with added self-connections:

$$\hat{A} = D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}} \tag{3.23}$$

where D is the diagonal degree matrix of $A + I$, and I is the identity matrix.

Graph Pooling and Readout Functions

To obtain a fixed-size representation of an entire graph or perform down-sampling for deeper architectures, graph pooling layers can be employed. Some popular graph pooling methods include global mean, max or sum pooling, and more advanced techniques such as DiffPool and Graph U-Net.

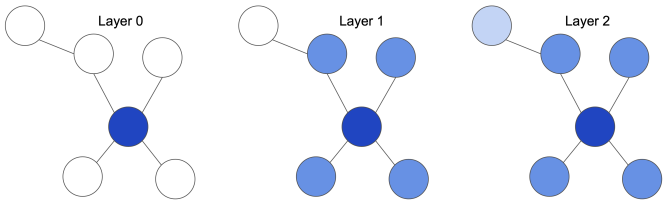

Figure 3.5 Layer pooling in a GNN

The readout function is used to obtain a fixed-size representation of an entire graph, which can be further processed or used for graph-level predictions. Common readout functions include component-wise mean, max, or sum of node features. In Figure 3.5, a node is propagating its update to the rest of the network.

A message passing layer is a "layer" of the neural network that maps a graph G to an updated graph G', usually by updating node values. Below is the equation of an update to a layer:

$$h_u = \phi(x_u, \Pi_{v \in N_u} \psi(x_u, x_v, e_{uv})) \tag{3.24}$$

where x_u is the feature set of node u, N_u is the neighbourhood, Π is an argument-invariant aggregation function (mean, sum, product), ϕ is the update function, and ψ is the message function.

Chapter 4

Data

Medical data is an essential resource for the development and implementation of machine learning models in healthcare applications. The use of machine learning in medicine has the potential to improve patient outcomes, lower costs, and advance research. In this section, we will discuss different types of medical data, challenges faced when using medical data for machine learning, and best practices to optimize model performance.

4.1 Data Description

4.1.1 MIMIC

The Medical Information Mart for Intensive Care (MIMIC) dataset is a large, publicly available collection of electronic health records (EHRs) from the intensive care unit (ICU) patients at the Beth Israel Deaconess Medical Center in Boston, Massachusetts. The MIMIC database is widely used by researchers worldwide for developing machine learning models and conducting clinical research in critical care.

Overview of the MIMIC Dataset

The MIMIC dataset includes de-identified data from over 60,000 ICU admissions, encompassing more than 40,000 unique patients. The dataset spans a period of more than a decade, from 2001 to 2012. The MIMIC database contains various types of medical data, including:

- Demographic information

- Vital signs
- Laboratory test results
- Medications
- Clinical notes
- Radiology reports
- Procedure codes

The availability of a wide range of data types within the MIMIC dataset allows researchers to model and analyze various aspects of patient care in the ICU setting.

Applications of the MIMIC Dataset in Machine Learning Research

The MIMIC dataset has been used extensively in machine learning research to develop models for various clinical tasks, such as:

1. Mortality prediction: Predicting patient outcomes, such as in-hospital mortality and ICU readmission risk.
2. Disease diagnosis: Identifying and diagnosing various diseases using EHR data.
3. Treatment recommendation: Developing personalized treatment plans based on individual patient conditions.
4. Phenotyping: Automatically identifying patient subgroups based on shared characteristics or disease patterns.

These applications have contributed significantly to improving patient care and treatment decisions in the ICU setting.

Challenges in Using the MIMIC Dataset

Despite its many advantages, using the MIMIC dataset for machine learning research also presents several challenges:

1. **Data preprocessing:** Medical data can be noisy, incomplete, and inconsistent, requiring extensive preprocessing to ensure data quality and accuracy.

2. **Data sparsity:** Due to the nature of critical care and the variability of patient conditions, some clinical events or observations may be sparse in the dataset.

3. **Temporal dependencies:** Many clinical variables have temporal dependencies that require specialized techniques to model and analyze.

4. **Data de-identification:** The MIMIC dataset is de-identified, which may limit researchers' ability to investigate certain aspects of patient care or validate their findings on external datasets.

Despite these challenges, the MIMIC dataset remains a valuable resource for medical machine learning research due to its extensive collection of diverse, real-world ICU data.

4.2 General Model Training Background

4.2.1 Handling Overfitting and Validation Techniques

Overfitting occurs when a machine learning model learns the noise and idiosyncrasies present in the training data, leading to poor generalization on unseen data. It is crucial to address overfitting to ensure the model performs well on real-world data, maintaining a balance between bias and variance. This section discusses various techniques to handle overfitting, as well as validation techniques to assess the model's performance.

Regularization

Regularization is a popular approach to prevent overfitting by adding a penalty term to the model's loss function. This penalty discourages the model from learning overly complex functions and promotes simpler models, improving generalization. Common regularization techniques include L1 and L2 regularization:

- **L1 Regularization**: A penalty term proportional to the absolute value of the model parameters (also known as Lasso regularization) is added to the loss function. L1 regularization encourages sparse solutions, effectively performing feature selection.

- **L2 Regularization**: A penalty term proportional to the square of the model parameters (also known as Ridge regularization or weight

decay) is added to the loss function. L2 regularization helps prevent large parameter values, reducing the complexity of the model.

Data Augmentation

Data augmentation involves creating new training examples by applying various transformations to the original data, thereby expanding the training set. This helps the model generalize better by learning invariant features across the augmented data. Some common data augmentation techniques include:

- For image data: rotation, scaling, flipping, translation, and color jittering.
- For text data: synonym replacement, random deletion, random swapping, and adversarial misspelling.
- For time-series data: time-warping, noise injection, and scaling.

Early Stopping

Early stopping is a technique in which the training process is halted when the model's performance on a validation set starts to degrade, typically indicating that the model is beginning to overfit. By monitoring a validation metric such as loss, early stopping prevents the model from overfitting while still allowing it to learn from the training data.

Dropout

Dropout is a regularization technique specifically designed for deep learning models. During training, dropout randomly sets a fraction of the neurons' activations to zero at each iteration. This prevents the model from relying too heavily on any single neuron, promoting the learning of redundant and more robust representations.

Validation Techniques

Validation techniques are essential for assessing the model's performance and generalization ability. They help in model selection, hyperparameter tuning, and avoiding overfitting. Some common validation techniques include:

- **Hold-out Validation**: The dataset is split into a training set and a validation set. The model is trained on the training set and evaluated on the validation set. The disadvantage of this method is that the model's performance can be sensitive to the choice of the validation set.
- **K-fold Cross-Validation**: The dataset is divided into k equal-sized folds. The model is trained and evaluated k times, each time using a different fold as the validation set and the remaining $(k-1)$ folds as the training set. The model's performance is calculated as the average performance across the k iterations. K-fold cross-validation provides a more reliable estimate of the model's performance compared to hold-out validation.
- **Leave-One-Out Cross-Validation**: This is a special case of K-fold cross-validation, where k is equal to the number of data points in the dataset. The model is trained and evaluated for each data point, using the remaining data points as the training set. This method provides the most accurate performance estimate but can be computationally expensive for large datasets.

Properly handling overfitting and employing validation techniques are crucial steps in the development of machine learning models. By applying these techniques, the model's performance can be optimized, and its generalization to new, unseen data can be ensured.

4.2.2 K-fold Cross-Validation

K-fold cross-validation is a widely used validation technique that provides a more reliable estimate of a machine learning model's performance on unseen data. It helps to assess the model's generalization ability, making it useful for model selection, hyperparameter tuning, and avoiding overfitting. In this subsection, we provide a detailed description of the k-fold cross-validation process and its advantages and disadvantages.

The K-fold Cross-Validation Process

The k-fold cross-validation process consists of the following steps:

1. Shuffle the dataset randomly.

2. Split the dataset into k equal-sized folds. Each fold should ideally have an equal distribution of the target variable to ensure a representative evaluation.
3. For each fold $i \in \{1, 2, \ldots, k\}$:
   (a) Set fold i aside as the validation set, and use the remaining $(k-1)$ folds as the training set.
   (b) Train the model on the training set.
   (c) Evaluate the model on the validation set, and record the performance metric of interest (e.g., accuracy, F1 score, mean squared error, etc.).
4. Calculate the average performance metric across the k iterations to obtain the final performance estimate.

The choice of k depends on the size of the dataset and the computational resources available. A larger k results in a more accurate performance estimate but increases the computation time. A common choice for k is 5 or 10.

Advantages of K-fold Cross-Validation

K-fold cross-validation has several advantages:

- It provides a more reliable performance estimate compared to hold-out validation, as it uses the entire dataset for both training and evaluation.
- It reduces the impact of the validation set's choice on the performance estimate, as each data point contributes to the evaluation exactly once.
- It helps to mitigate the risk of overfitting, as the evaluation is based on multiple training sets, avoiding over-optimization for a single train-validation split.

Disadvantages of K-fold Cross-Validation

Despite its benefits, k-fold cross-validation has some disadvantages:

- It can be computationally expensive, especially for large datasets and deep learning models, as it requires training and evaluating the model k times.

- The performance estimate can still be sensitive to the initial random shuffling of the dataset, particularly for small datasets. Repeated k-fold cross-validation, where the entire process is performed multiple times with different random shuffles, can alleviate this issue.
- If the dataset has a significant class imbalance, stratified k-fold cross-validation, which preserves the class distribution in each fold, should be used to ensure representative evaluation.

Despite its limitations, k-fold cross-validation remains a popular and widely used validation technique in machine learning, providing a more reliable assessment of a model's generalization ability and helping to avoid overfitting.

Chapter 5

Current Research

5.1 Literature review

In this section, I'm going to include the literature reviews that I have been doing throughout the two semesters for all the thesis topics that I iterated through.

Graph-Representation of Patient Data: a Systematic Literature Review

Schrodt et al. (2020)

This systematic literature review found that most research on graphs representing patient data focuses on temporal relations, often represented by the connection among laboratory data points. However, there is potential for using graph theoretical algorithms to develop decision support systems for diagnosis, medication or therapy of patients using similarity measurements or different kinds of analysis.

Modeling electronic health record data using an end-to-end knowledge-graph-informed topic model

Zou et al. (2022)

The authors present Graph ATtention-Embedded Topic Model (GAT-ETM), an end-to-end taxonomy-knowledge-graph-based multimodal embedded topic model. GAT-ETM distills latent disease topics from EHR data by learning the embedding from a constructed medical knowledge graph. They applied GAT-ETM to a large-scale EHR dataset consisting of over 1 million patients. They evaluated its performance based on topic quality,

drug imputation, and disease diagnosis prediction. GAT-ETM demonstrated superior performance over the alternative methods on all tasks. Moreover, GAT-ETM learned clinically meaningful graph-informed embedding of the EHR codes and discovered interpretable and accurate patient representations for patient stratification and drug recommendations.

GAT-ETM is a tool for extracting clinical knowledge from electronic health records. It is based on a medical knowledge graph, and learns an embedding from this graph. This embedding is then used to distill latent disease topics from the EHR data. GAT-ETM was applied to a large dataset of over 1 million patients, and was found to outperform alternative methods on all tasks evaluated (topic quality, drug imputation, and disease diagnosis prediction).

Temporal Graph Networks for Deep Learning on Dynamic Graphs

Rossi et al. (2020)

The authors present Temporal Graph Networks (TGNs), a generic, efficient framework for deep learning on dynamic graphs represented as sequences of timed events. TGNs can be used for both transductive and inductive prediction tasks. We show that several previous models for learning on dynamic graphs can be cast as specific instances of our framework. We perform a detailed ablation study of different components of our framework and devise the best configuration that achieves state-of-the-art performance on several transductive and inductive prediction tasks for dynamic graphs.

High-throughput phenotyping with temporal sequences

Estiri et al. (2020)

In this study, the authors develop a high-throughput phenotyping method that leverages temporal sequential patterns from EHRs. They find that this results in superior classification performance across all 10 phenotypes compared with the standard representations in electronic phenotyping. The algorithm's classification performance was superior or similar to the performance of previously published electronic phenotyping algorithms.

Where in the Brain is Depression?

Pandya et al. (2012)

This paper is a summary of how depression manifests itself physically inside the human brain – the paper focuses on describing how Major Depressive Disorder (MDD) affects the brain and how we currently understand it. First, since most people reading this are not neuroscientists, then we start with a description of how the brain is structured: it can be divided into three parts:

1. **Prefontal neocortex:** this is involved in higher cognitive processes and regulation of emotions
2. **Limbic brain:** Otherwise known as the mammalian brain, this reigon of the brain is involved with basal emotions, such as fight or flight and procreation.
3. **Reptilian complex:** Comprised of the basal ganglia and the brain stem, the reptilian complex controls motor function and social communication.

Studies have repeatedly found that an analysis of the structure of the brain is an indicative signal in predicting depression.

Geometric models of brain white matter for microstructure imaging with diffusion MRI

Scholz et al. (2014)

This paper uses a mesh to construct a model of the brain. It uses an algorithm called "marching cubes" to reconstruct a surface given MRI data. I think this would be a good starting point for my own thesis, such that I use a similar reconstruction approach to rebuild a brain. Then, I can use various methods from differential geometry to classify each brain.

Towards a Mathematical Model of the Brain

Young (2020)

This paper uses a network of differential equations to model the neurons of the cerebral cortex. Although this paper is not exactly what we want to do, I think it is still important because it is an attempt to model the brain in a mathematical way.

Cortical Surface-Based Analysis: II: Inflation, Flattening, and a Surface-Based Coordinate System

Fischl et al. (1999)

This paper was one of the first papers to try to model the brain mathematically. It describes ways to transform parts of the brain, "inflating" certain surfaces so that activity may be studied. They also developed transformations between various dimensions. I think that this paper is a very good starting point to what we want to do.

Exploring folding patterns of infant cerebral cortex based on multi-view curvature features: Methods and applications

Duan et al. (2019)

I think that this paper might be one of the most important papers to read. It covers how to use a similarity network to fuse all similarity matrices to compare the folding patterns of infant brains. Then, they cluster the dataset and reveal that each unique cluster has different expressed phenotypes.

5.2 Reserach Basis

I finally settled on (Zheng et al., 2022) to use as the theoretical basis of my thesis.

Below is an extended summary of that paper:

- Multimodal biomedical data can provide more complementary information about the patient's condition than single modal data, facilitating a more reliable diagnosis
- A number of approaches based on multi-modal learning have been proposed for disease prediction, but there are still several common issues with these methods. Graph-based methods, particularly graph convolutional networks (GCNs), have been applied in various biomedical applications and disease prediction field.
- Most existing graph-based methods construct the patient relationship graph from existing multi-modal features through pre-defined similarity measures, then apply GCNs to aggregate patient features over local neighborhoods to give the prediction results. However, these methods fail to effectively mine the intrinsic information of each modality.

- The concatenation and intra-modal attention mechanisms used in previous studies are hard to capture the latent inter-modal correlation, which may cause the learned representation to be biased towards a single modality.
- Existing single-graph based methods and multi-graph based methods construct the graph through hand-designed similarity measures, which are difficult to generalize to downstream tasks. A better approach is to learn a graph in an adaptive way.
- A population graph can be constructed for disease diagnosis by treating the patient set as a node set and the connections between each pair of nodes as edge sets. A well defined adjacency matrix is associated with the edge sets.

In the next section, I describe how I adapted their framework to create a predictve model from MIMIC-III data.

Chapter 6

Method

6.1 Data Preprocessing

In order to develop accurate and reliable machine learning models for medical applications, data preprocessing is a crucial step. This section outlines a framework designed to handle multi-modal medical data, ensuring that the input data is clean, consistent, and suitable for model training.

6.1.1 Overview of the Data Preprocessing Framework

The proposed data preprocessing framework consists of several stages, as illustrated in Figure 6.1:

1. Data Ingestion
2. Data integration
3. Data cleaning
4. Feature extraction
5. Feature selection and transformation

6.1.2 Data Acquisition

The first stage of the framework involves acquiring medical data from various sources. These sources may include electronic health records (EHRs), medical imaging data, laboratory test results, and clinical notes. The acquired data

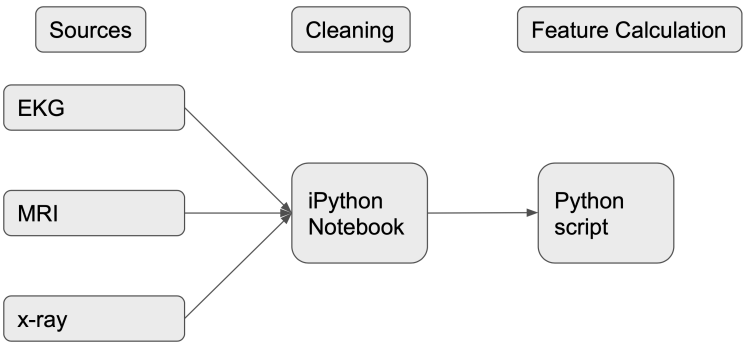

Figure 6.1 Data preprocessing framework for multi-modal medical data

should be stored in a consistent and structured format to facilitate subsequent processing steps.

6.1.3 Data Integration

As medical data often originates from multiple sources, it is essential to integrate the diverse datasets into a single unified dataset. This process may involve:

- Matching and merging records pertaining to the same patient across different sources.
- Aligning timestamps between different modalities to account for temporal dependencies.
- Handling inconsistencies between different sources or modalities.

Data integration ensures that all relevant information is consolidated into a single dataset, offering a comprehensive view of each patient's medical history.

6.1.4 Data Cleaning

Medical data can be noisy, incomplete, or inconsistent, which may adversely affect the performance of machine learning models. Data cleaning involves identifying and correcting these issues, such as:

- Handling missing values by imputation or deletion.
- Removing duplicate records or outliers.
- Correcting data entry errors and inconsistencies.

Data cleaning ensures that the input data is of high quality and suitable for model training.

6.1.5 Feature Extraction

Multi-modal medical data often contains a wealth of information that can be used to derive new features for machine learning models. Feature extraction involves transforming raw data into a set of meaningful features that capture relevant patterns and relationships. For example:

- Extracting statistical features, such as mean, variance, or trend from time-series data.
- Extracting text-based features from clinical notes using natural language processing techniques.
- Extracting image-based features from medical imaging data using deep learning techniques.

Feature extraction helps in reducing dimensionality and capturing complex patterns in the data.

6.1.6 Feature Selection and Transformation

After extracting features, it is essential to select the most relevant ones for model training. This process may involve:

- Univariate feature selection methods, such as correlation coefficients or mutual information.
- Multivariate feature selection methods, such as recursive feature elimination or LASSO regularization.

Furthermore, transforming the selected features using techniques like normalization or standardization can help improve model performance by ensuring that all features have comparable scales.

In summary, a well-designed data preprocessing framework is essential for handling multi-modal medical data and ensuring optimal performance of machine learning models in healthcare applications. The proposed framework addresses various challenges associated with multi-modal medical data and provides a systematic approach for preprocessing diverse types of medical datasets.

6.2 Model Framework

I constructed a model framework that is similar to (Zheng et al., 2022), which is shown in Figure 6.2.

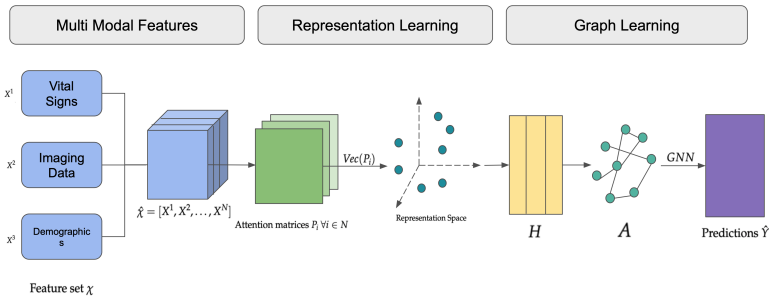

Figure 6.2 The pipeline framework

First, for each modality of data, we create a unique feature set $X = \{x_1, x_2, \ldots, x_m\}$ that is derived from the specialities of the data. This feature matrix X is high-dimensional and quite sparse.

Trying various dimensionality reduction techniques, I then pushed this feature set into a low dimensional space χ, where the distance between any pair of features represents the similarity between the two.

Treating each of the m modalities of data as a a feature, and the metric distance between any two features i, j in χ as the edge, we can construct a graph G. To vary the connectivity of G, we introduce a hyperparameter θ that when:

$$D(x_i, x_j) < \theta \implies e_{i,j} = 0$$

.

As input into the graph neural network, we provide the adjacency matrix A of G.

Finally, we use the insurance billing codes that is reported at the end of each patient's ICU stay as the prediction variable.

Chapter 7

Results

7.1 Results

Here we discuss how changing the manifold learning algorithm and varying the hyperparameter θ changed our results.

7.1.1 Changing manifold learning algorithms

In Section 3.2.1, I discussed the theoretical underpinnings of a few dimensionalality reduction techniques. I experimented with the following three:

1. Principal component analysis
2. Isomap
3. t-distributed stochastic neighbor embedding

In Figure 7.1, I graph the cosine similarity between the feature set and a random set of patient samples after dimensionality reduction.

After many rounds of hyperparameter searching and tuning, I achieved an accuracy of 77% on the validation set.

Figure 7.2 shows the final model's train and test loss having a bit of trouble converging towards the end. This osciliation implies that the model training did not go that smoothly.

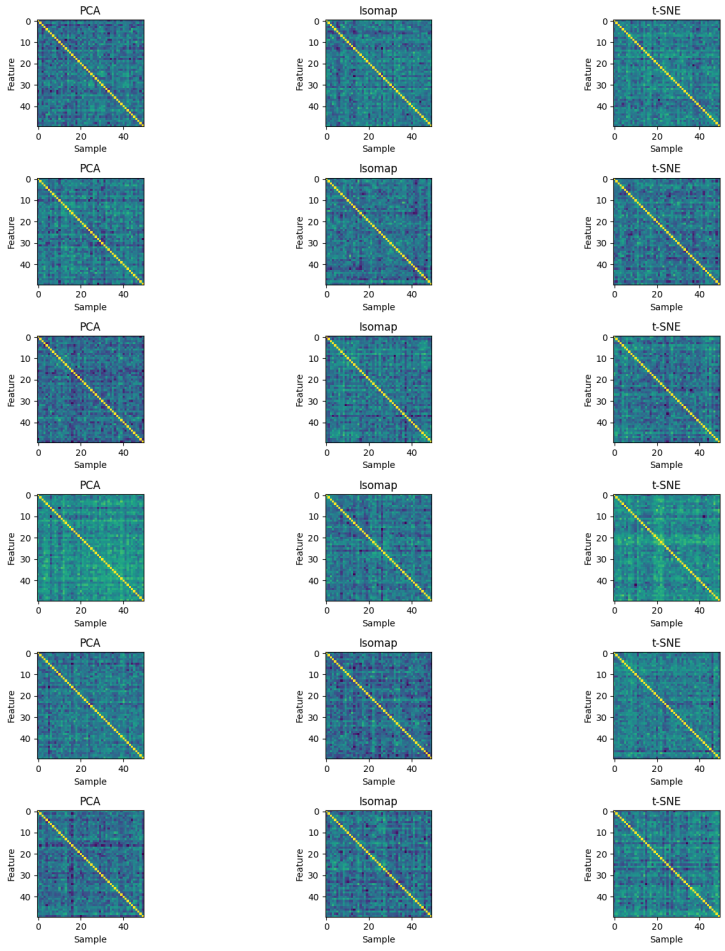

Figure 7.1 Similarity matrix between a random 50 samples compared to the m sized feature set, with reduction technique varied as well as learning rate.

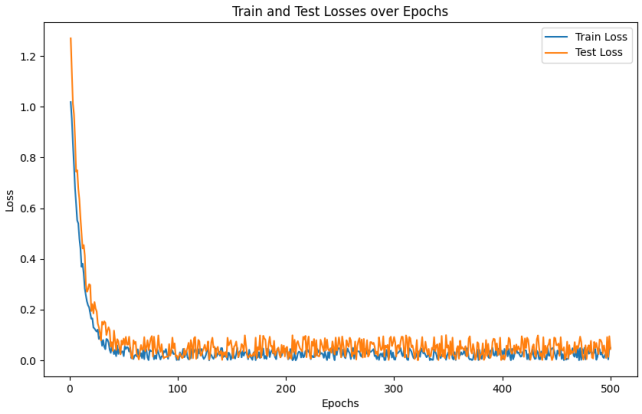

Figure 7.2 Train test loss after search for optimal hyperparameter.

Chapter 8

Conclusion

8.1 Closing remarks

In recent years, medical machine learning has made significant strides in shaping the future of healthcare. By leveraging the power of computational algorithms and large-scale data, machine learning models have been developed to assist clinicians in diagnostic processes, treatment planning, and patient monitoring. The advancements in this field have the potential to greatly benefit society by improving patient outcomes, reducing healthcare costs, and streamlining clinical workflows.

One notable advancement in medical machine learning is the development of deep learning models for medical imaging analysis. Convolutional Neural Networks (CNNs) have shown remarkable success in tasks such as image segmentation, object detection, and classification. These networks have been applied to a wide variety of medical imaging modalities, including X-rays, CT scans, and MRI images, enabling automated disease detection and diagnosis at levels comparable to or even surpassing human experts. For instance, deep learning models have demonstrated impressive results in detecting diabetic retinopathy from retinal images, predicting stroke outcomes from brain CT scans, and classifying skin cancer from dermatoscopic images.

Another area where machine learning has made substantial progress is natural language processing (NLP) applied to clinical notes and text data. NLP techniques enable the extraction of valuable insights from unstructured text data present in electronic health records (EHRs). These algorithms can identify disease-related entities, extract relationships between various

clinical concepts, and predict patient outcomes based on textual data. By harnessing these capabilities, clinicians can make more informed decisions about treatment plans while researchers can analyze vast amounts of clinical data to uncover new patterns or associations that might not be apparent through manual inspection.

In addition to diagnostics and treatment planning, machine learning has also made advancements in personalized medicine. By incorporating individual-specific information such as genetic profiles or patient demographics into predictive models, more tailored treatment recommendations can be devised for each patient. This approach is particularly relevant for complex diseases like cancer, where molecular subtypes can significantly impact treatment response. Machine learning models can help identify patient subgroups that might respond better to specific therapeutic interventions, enabling more precise and personalized care.

Furthermore, the integration of machine learning with wearable devices and remote monitoring technologies has opened new avenues for continuous patient monitoring and early intervention. Machine learning algorithms can analyze data from sensors embedded in wearable devices to identify potential health concerns or monitor chronic conditions such as diabetes, heart disease, or sleep disorders. This can facilitate timely interventions and improve patient adherence to treatment plans.

In conclusion, the recent advancements in medical machine learning hold great promise for transforming healthcare delivery and improving patient outcomes. By integrating these cutting-edge technologies into clinical practice, healthcare providers can harness the power of data to make more informed decisions, personalize treatment plans, and ultimately enhance the quality of care provided to patients. As machine learning techniques continue to mature and evolve, it is imperative for healthcare stakeholders to invest in research and development to fully realize the potential benefits these advancements can bring to society.

8.1.1 Contribution

I present a machine learning framework that can ingest publicly-available medical data and developed a graph learning architecture that can generate clinical predictions.

8.1.2 Qualifications

As with anything, I need to qualify these results. These results are obtained from a small, possibly unrepresentative, subset of the global population. To create a better model that is more inclusive, one needs more representative data to train on.

Bias and Inequality

There are well-documented and widespread inequalities in the American medical system, and those systematic biases could be unknowingly present in medical data. Other sources of bias include human error in data input at hospitals or the tendency of insurance companies and hospitals to over-bill, which is reflected in diagnostic codes.

Machine learning might soon revolutionize healthcare, but there still remains much work to be done. Insuring that new systems do not perpetuate the inequalities of current and past is integral in advancing medical machine learning.

8.1.3 Next steps

The next steps that I would take are:

1. Explore integrating datasets that came from different sources, which further extends "multi-modality data"
2. Experiment with different graph learning architectures
3. Better awknowledge and treat bias in data

8.2 Links

To find a PDF of this paper, various talks I gave over the academic year, as well as a conference poster, please visit this site. I can be contacted at jbjiang[AT]hmc.edu or jjiang990[AT]hmc.edu.

Appendix A

Appendix

A.1 Gradient Descent

Gradient descent is an optimization algorithm that is widely used in machine learning models to minimize a loss function. It is an iterative optimization algorithm that moves in the direction of the negative gradient of the function to find the local or global minimum. In this technical appendix, we will explain the concept of gradient descent, its various types, and its applications in machine learning.

Given a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, our goal is to find the minimum point of the function, i.e., a point $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$. In gradient descent, we start with an initial guess $x^{(0)}$ and iteratively update our guess as follows:

$$x^{(k+1)} = x^{(k)} - \alpha \nabla f(x^{(k)}) \tag{A.1}$$

where k is the iteration number, α is the learning rate, and $\nabla f(x^{(k)})$ is the gradient of the function f at the point $x^{(k)}$. The gradient is a vector that contains partial derivatives of the function with respect to each variable:

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right) \tag{A.2}$$

The learning rate α determines the step size in the direction of the negative gradient. If the learning rate is too small, the convergence to the minimum will be slow, while if it is too large, the algorithm might overshoot the minimum and diverge.

A.1.1 Batch Gradient Descent

In batch gradient descent, the whole dataset is used to calculate the gradient at each iteration. This can be computationally expensive for large datasets, but it provides a stable convergence to the minimum.

A.1.2 Stochastic Gradient Descent

In stochastic gradient descent (SGD), a random data point is chosen at each iteration to calculate the gradient. This makes the algorithm much faster and allows it to handle large datasets efficiently. However, the convergence of SGD is noisier and less stable compared to batch gradient descent.

A.1.3 Mini-batch Gradient Descent

Mini-batch gradient descent combines the best of both worlds by using a small random subset of the dataset, called a mini-batch, at each iteration to calculate the gradient. This provides a balance between the computational efficiency of SGD and the stability of batch gradient descent.

Gradient descent is a powerful optimization algorithm that is widely used in machine learning for finding the minimum of a loss function. By iteratively updating the parameters in the negative direction of the gradient, gradient descent can converge to the optimal parameters of a machine learning model. Its variants, such as stochastic and mini-batch gradient descent, provide efficient ways to handle large datasets and find a balance between convergence stability and computational efficiency.

Bibliography

2023. Logistic regression. *Wikipedia* URL https://en.wikipedia.org/wiki/Logistic_regression.

Bahadure, Nilesh Bhaskarrao, Arun Kumar Ray, and Har Pal Thethi. 2017. Image analysis for mri based brain tumor detection and feature extraction using biologically inspired bwt and svm. *International Journal of Biomedical Imaging* 2017:1–12. doi:10.1155/2017/9749108.

Bari, Lisa. 2019. Rethinking patient data privacy in the era of digital health. *Forefront Group* doi:10.1377/forefront.20191210.216658.

Cebul, Randall D., Thomas E. Love, Anil K. Jain, and Christopher J. Hebert. 2011. Electronic health records and quality of diabetes care. *New England Journal of Medicine* 365(9):825–833. doi:10.1056/nejmsa1102519.

Duan, Dingna, Shunren Xia, Islem Rekik, Yu Meng, Zhengwang Wu, Li Wang, Weili Lin, John H. Gilmore, Dinggang Shen, Gang Li, and et al. 2019. Exploring folding patterns of infant cerebral cortex based on multi-view curvature features: Methods and applications. *NeuroImage* 185:575–592. doi:10.1016/j.neuroimage.2018.08.041.

Estiri, Hossein, Zachary H Strasser, and Shawn N Murphy. 2020. High-throughput phenotyping with temporal sequences. *Journal of the American Medical Informatics Association* 28(4):772–781. doi:10.1093/jamia/ocaa288.

Fischl, Bruce, Martin I. Sereno, and Anders M. Dale. 1999. Cortical surface-based analysis. *NeuroImage* 9(2):195–207. doi:10.1006/nimg.1998.0396.

Gerber, Samuel, Tolga Tasdizen, Sarang Joshi, and Ross Whitaker. 2009. On the manifold structure of the space of brain images. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009* 305–312. doi:10.1007/978-3-642-04268-3_38.

Gunter, Tracy D, and Nicolas P Terry. 2005. The emergence of national electronic health record architectures in the united states and australia: Models, costs, and questions. *Journal of Medical Internet Research* 7(1). doi:10.2196/jmir.7.1.e3.

HHS. 2009. Hhs strengthens hipaa enforcement. URL https://wayback.archive-it.org/3926/20131018161347/http://www.hhs.gov/news/press/2009pres/10/20091030a.html.

Johnson, Alistair E.W., Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, Roger G. Mark, and et al. 2016. Mimic-iii, a freely accessible critical care database. *Scientific Data* 3(1). doi:10.1038/sdata.2016.35.

May, Mike. 2021. Eight ways machine learning is assisting medicine. *Nature Medicine* 27(1):2–3. doi:10.1038/s41591-020-01197-2.

Na, Liangyuan, Cong Yang, Chi-Cheng Lo, Fangyuan Zhao, Yoshimi Fukuoka, and Anil Aswani. 2018. Feasibility of reidentifying individuals in large national physical activity data sets from which protected health information has been removed with use of machine learning. *JAMA Network Open* 1(8). doi:10.1001/jamanetworkopen.2018.6040.

Newitt, Patsy. 2022. Highest-paying physician specialties in the us: 2022. URL https://www.beckersasc.com/asc-news/highest-paying-physician-specialties-in-the-us-2022.html.

Pandya, Mayur, Murat Altinay, Donald A. Malone, and Amit Anand. 2012. Where in the brain is depression? *Current Psychiatry Reports* 14(6):634–642. doi:10.1007/s11920-012-0322-7.

Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65(6):386–408. doi:10.1037/h0042519.

Rossi, Emanuele, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. 2020. Temporal graph networks for deep learning on dynamic graphs. *CoRR* abs/2006.10637. URL https://arxiv.org/abs/2006.10637. 2006.10637.

Rué-Queralt, Joan, Angus Stevner, Enzo Tagliazucchi, Helmut Laufs, Morten L. Kringelbach, Gustavo Deco, and Selen Atasoy. 2021. De-

coding brain states on the intrinsic manifold of human brain dynamics across wakefulness and sleep. *Communications Biology* 4(1). doi:10.1038/s42003-021-02369-7.

Scholz, Jan, Valentina Tomassini, and Heidi Johansen-Berg. 2014. Individual differences in white matter microstructure in the healthy brain. *Diffusion MRI* 301–316. doi:10.1016/b978-0-12-396460-1.00014-7.

Schrodt, Jens, Aleksei Dudchenko, Petra Knaup-Gregori, and Matthias Ganzinger. 2020. Graph-representation of patient data: A systematic literature review. *Journal of Medical Systems* 44(4). doi:10.1007/s10916-020-1538-4.

Ustebay, Serpil, Abdurrahman Sarmis, Gulsum Kubra Kaya, and Mark Sujan. 2022. A comparison of machine learning algorithms in predicting covid-19 prognostics. *Internal and Emergency Medicine* doi:10.1007/s11739-022-03101-x.

Xia, Feng, Sun Ke, Shuo Yu, Liangtian Wan, Shirui Pan, and Huan Liu. 2021. Graph learning: A survey doi:10.1109/TAI.2021.3076021.

Young, Lai-Sang. 2020. Towards a mathematical model of the brain. *Journal of Statistical Physics* 180(1-6):612–629. doi:10.1007/s10955-019-02483-1.

Zheng, Shuai, Zhenfeng Zhu, Zhizhe Liu, Zhenyu Guo, Yang Liu, Yuchen Yang, and Yao Zhao. 2022. Multi-modal graph learning for disease prediction 2203.05880.

Zou, Yuesong, Ahmad Pesaranghader, Ziyang Song, Aman Verma, David L. Buckeridge, and Yue Li. 2022. Modeling electronic health record data using an end-to-end knowledge-graph-informed topic model. *Scientific Reports* 12(1). doi:10.1038/s41598-022-22956-w.