Bachelor's thesis

Bachelor's Programme in Computer Science

Multimodal Machine Learning and Data Fusion in Medical Diagnosis

Juuso Saavalainen

June 10, 2024

Faculty of Science

University of Helsinki

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty

Faculty of Science

Koulutusohjelma — Utbildningsprogram — Study programme

Bachelor’s Programme in Computer Science

Tekijä — Författare — Author

Juuso Saavalainen

Työn nimi — Arbetets titel — Title

Multimodal Machine Learning and Data Fusion in Medical Diagnosis

Ohjaajat — Handledare — Supervisors

Prof. Ville Mustonen, Ph.D. student Teemu Kuosmanen

Työn laji — Arbetets art — Level

Bachelor’s thesis

Aika — Datum — Month and year

June 10, 2024

Sivumäärä — Sidoantal — Number of pages

27 pages, 2 appendix pages

Tiivistelmä — Referat — Abstract

Traditionally machine learning models are build to take use of single modality. Medical diagnosis is rarely based on single source of information. Multimodal machine learning introduces techniques like data fusion that allow models to process and use multiple modalities. This thesis examines these models in a context of medical diagnosing.

Multiple recent literature reviews suggest potential improvements in accuracy with multimodal machine learning compared to the models utilizing single modality in the same task. However, most of the research in this area is found employing retrospective data and lack prospective validation.

The thesis follows a structure that aims to first provide overview of the basics in machine and deep learning. Then it moves to examine data fusion and medical studies employing multimodal machine learning.

ACM Computing Classification System (CCS)
Applied computing → Life and medical sciences → Health informatics
Computing methodologies → Machine learning → Machine learning approaches → Neural networks

Avainsanat — Nyckelord — Keywords

Deep learning, Multimodal learning, Data fusion

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Tiivistelmä

Nyky-yhteiskunnassa kerätään ja tallennetaan tietoa ennennäkemättömin määrin. Tämä tarjoaa merkittäviä mahdollisuuksia koneoppimiselle (Machine Learning) ja tekoälylle (Artificial Intelligence). Koneoppimisen sovellukset ulottuvat lähes kaikialle, terveydenhuolto ja lääketiede mukaan lukien. Neuroverkkojen ja syväoppimismallien kehittyminen on mahdollistanut monimutkaisten ja laajojen tietomassojen analysoinnin. Lääkärit hyödyntävät usein diagnosoinnin tukena useita tietolähteitä kuten kuvia, mittaustuloksia, ja oireita. Diagnoisointiin tai taudin luokan tunnistamiseen kuuluu siten usein monien eri lähteiden datan arvioimista yhtenäisenä kokonaisuutena. Radiologiassa syväoppimista on käytetty esimerkiksi kuvien segmentointiin ja poikkeavuuksien tunnistamiseen (Lee et al., 2017). Lupaavista tuloksista huolimatta tekoälyyn pohjautuvien järjestelmien käyttö terveydenhuollossa on vielä olematonta.

Tämän tutkielman tarkoituksena on selvittää ja tutkia miten useita modaliteettäjä hyödynnytään diagnosinnin tukena, mihin ne pohjautuvat sekä mitä ongelmia niiden käyttöön liittyy. Erityisesti tutkielma keskittyy useiden modaliteettien käyttöön liittyvään datafuusioon, eli useiden modaliteettien yhteen sovittamiseen koneoppimis arkkitehtuureissa. Modaliteetillä tarkoitetaan tietotyyppiä kuten kuvaa tai ääntä. Tämän lisäksi tutkielma käsittelee neuroverkkoja ja syväoppimista perusteiden kautta rakentaen pohjan kompleksisempien kokonaisuuksien ja datafuusion käsittelyyn.

Ohjattu oppiminen (Supervised learning) on yksi koneoppimisen osa-alue, jossa olemassa olevan datan avulla pyritään muodostamaan malli, joka pystyy esimerkiksi luokittelemaan sille syötettävää dataa. Ohjattuun oppimiseen tarvitaan siis dataa joka on jo luokiteltua, tätä kutsutaan koulutus dataksi. Lineaarinen regressio on hyvä selkeä esimerkki joka soveltuu yksinkertaisten ongelmien ratkaisemiseen. Datan ulottuvuuksien ja kompleksisuuden lisääntyessä edistyneemmät algoritmit, kuten neuroverkot, toimivat paremmin.

Neuroverkon rakenne koostuu yksinkertaisimmillaan solmuista, jotka sijaitsevat verkon eri kerroksissa. Kokonaan yhdistetyssä neuroverkossa jokainen solmu yhdistyy jokaiseen seuraavan kerroksen solmuun. Solmujen väliselle yhteydelle sijoittuu paino sekä jokaiseen solmuun harha. Neuroverkon kouluttamisella tarkoitetaankin näiden painojen ja harhojen säätelyä niin, että verkko tuottaa halutun tuloksen. Neuroverkon ensimmäinen kerros jota usein kutsutaan syötekerrokseksi (Input layer) ottaa esimerkiksi kuvan pikselien ar-

voja solmujen arvoiksi. Kuva, joka koostuu 20x20 pikselistä, tuottaa täten 400 solmun kokoisen syötekerroksen josta jokainen solmu yhdistyy jokaiseen seuraavan kerroksen solmuun. Neuroverkon viimeistä kerrosta kutsutaan tulostekerrokseksi (Output layer) ja sen solmujen määrä määräytyy esimerkiksi luokittelutehtävissä luokiteltavien kohteiden mukaan. Ensimmäisen ja viimeisen kerroksen koko on siis valmiiksi määritelty. Kerroksia jotka sijaitsevat syöte ja tuloste kerroksen välissä kutsutaan piilokerroksiksi (Hidden layer). Piilokerrosten määrää ja kokoa ei ole valmiiksi määritelty, syväoppimisesta puhuttaessa viitataan verkkoihin joissa piilokerroksia on useita.

Neuroverkkon kouluttamisella tarkoitetaan prosessia jossa verkon harhat ja painot optimoidaan. Koulutukseen käytettävä data voidaan jakaa testi- ja koulutusdataan jolloin verkon suoriutumista voidaan arvioida datalla, jota ei ole käytetty sen kouluttamiseen. Eteenpäinsyöttö algoritmin avulla verkon syöte viedään kerroksien läpi aina tuloste kerrokselle asti. Yksittäisen solmun aktivaatio voidaan laskea siihen kytkettyjen solmujen painolla skaalattujen arvojen summana, johon lisätään solmun harha. Tätä summaa voidaan kutsua solmun lineaariseksi muunnokseksi, jotta aktivaatio saadaan laskettua tähän sovitetaan aktivaatiofunktio. Aktivaatiofunktion avulla lineaarisuus saadaan rikottua. Tämä mahdollistaa epälineaaristen suhteiden mallintamisen. Samalla kaavalla voidaan edetä tuloskerrokselle asti. Tulostekerroksen aktivaatiofunktio poikkeaa usein muiden kerroksien aktivaatiofunktiosta, koska tämän kerroksen arvot halutaan kuvata todennäköisyyksinä ja täten niiden summan on oltava 1. Viimeisen kerroksen solmujen ja syötteen tunnisteen (label) erotusta kutsutaan kustannusfunktioksi. Takaisinvirtausalgortimi hyödyntää tätä ja sen tehtävä on laskea jokaisen painon ja harhan osittaisderivaatta suhteessa kustannusfunktioon. Näitä arvoja voidaan kutsua myös gradienteiksi. Gradientti suhteessa kustannusfunktioon kertoo suunnan, joka kasvattaa sen arvoa nopeimmin. Siirtämällä painoja ja harhoja kohti negatiivistä gradienttiä voidaan kustannusfunktion arvo minimoida. Neuroverkkoja on kehitetty moneen tarkoitukseen. Useat näistä kuten konvoluutio ja takisinkytkentäneuroverkot perustuvat samojen algoritmien ja koulutustekniikoiden käyttöön kuin kokonaan kytketyssä eteenpäinsyöttö neuroverkossa.

Multimodaalinen koneoppiminen (Multimodal machine learning) pyrkii hyödyntämään useita modaliteettejä saman aikaisesti. Multimodaalista koneoppimista voidaan hyödyntää monissa eri käyttötarkoituksissa kuten generatiivisessä tekoälyssä. Lääketieteellisessä diagnosoinnissa multimodaalisia koneoppimista voidaan hyödyntää esimerkiksi tunnistamaan tauti. Esimerkiksi röntgenkuvien avulla voidaan kouluttaa malli tunnistamaan nivelrikko (Tiulpin et al., 2018). Malleja, jotka pohjautuvat vain yhden modaliteetin kuten kuvan käyttöön, kutsutaan unimodaalisiksi (Unimodal) malleiksi. Röntgenkuvien lisäksi mallille voidaan syöttää myös muita modaliteettejä, jotka ovat relevantteja ongelman kannalta. Esimerkiksi lääketieteellisten kuvien lisäksi voidaan käyttää potilaskertomuksia, mittaus- tai testi tuloksia, tai geneettisiä tietoja. Datafuusio käsittelee useiden modaliteettien yhdistelemistä (Baltrušaitis et al., 2019). Datafuusiomenetelmät voidaan karkeasti jakaa kolmeen luokkaan: aikaiseen , myöhäiseen ja keskivaiheen fuusioon. Mallille syötettävät modaliteetit ovat lähtökohtaisesti erillisiä ja rakenteellisesti toisistaan poikkeavia. Fuusiomenetelmien etuliite kuvaa arkkitehtuurissa fuusion vaiheen ajankohtaa suhteessa koneoppimisalgoritmiin tai -malliin. Aikaisella fuusiolla tarkoitetaan täten siis menetelmää, jossa modaliteetit yhdistetään yhtenäiseksi ennen mallille syöttämistä. Myöhäinen fuusio taas on menetelmä, jossa jokainen modaliteetti käsitellään erillisenä ja jokainen syötetään omaan malliin. Näiden tuloksia yhdistetään ennen lopullista päätöksentekoa. Keskivaiheen fuusiolla tarkoitetaan yhdistämistä, joka tapahtuu mallin sisällä. Kaikista näistä myös löytyy variaatioita, joissa esimerkiksi jokin modaliteetti käsitellään eri tavalla.

Fuusiomenetelmän toimivuuteen vaikuttaa vahvasti ongelma, jota pyritään ratkaisemaan. Lisäksi siihen vaikuttaa datan määrä ja laatu. Aikaisen fuusion menetelmässä voidaan esimerkiksi hyödyntää valmiiksi koulutettuja malleja tai autoenkoodereita (Auto encoder) modaliteettien kuvaamiseksi yhtenäiseksi vektoriksi. Myöhäisen fuusion kohdalla taas voidaan hyödyntää unimodaaleja malleja, joiden koulutus ja käyttö eivät ole sidoksissa muihin modaliteetteihin. Myöhäisen fuusiota hyödyntävissä kokonaisuuksissa voidaan myös puuttuviin modaliteetteihin reagoida, koska yksikään osa ei ole suoraan riippuvainen muista modaliteeteistä. Keskivaiheen fuusio hyödyntää neuroverkkoja ja poiketen muista menetelmistä, se voidaan kouluttaa yhtenäisenä mallina ja takaisinvirtausalgoritmiä hyödyntäen aina tuloksesta alkuperäisiin syötteisiin asti. Neuroverkkojen hyödyntäminen vaatii enemmän dataa kuin muut menetelmät ja tekee siten mallin kouluttamisesta vaikempaa. Tieteellisessä yhteisössä ei ole konsensusta siitä, mikä menetelmistä on tehokkain. Multimodaaliseen koneoppimiseen liittyvät keskeiset haasteet voidaan jakaa viiteen luokkaan (Baltrušaitis et al., 2019):

1. *Esittäminen (Representation)* miten useat modaliteetit esitetään merkityksellisesti?

2. *Kääntäminen (Translation)* miten data määritetään modaliteetista (A) -> (B)?

3. *Kohdistaminen (Alignment)* mitkä osat yhdestä modaliteetista vastaavat suoraan toista?

4. *Fuusio (Fusion)* miten modaliteetit yhdistetään?

5. *Yhteisoppiminen (Co-learning)* miten tietoa voidaan jakaa eri modaliteettien välillä?

Multimodaaliseen koneoppimiseen pohjautuvien mallien tehokkuuden arviointia vaikeuttaa ratkaisujen vahva riippuvuus käsiteltävästä ongelmasta ja saatavilla olevasta datasta. Monet tutkimukset käyttävät valmiiksi kerättyä dataa, joka ei ole julkisesti saatavilla. Multimodaalista koneoppimista täsmälääketieteessä tarkasteleva kirjallisuuskatsaus (Kline et al., 2022), osoitti usean modaliteetin nostavan mallien tarkkuutta keskimäärin 6.4% verrattaen yhden modaliteetin malleihin. Katsaukseen päätyneissä artikkeleissä suurimpina ongelmina nousivat pienet otoskoot, luokkien erisuuruus ja retrospektiivinen data. Lisäksi useat tutkimukset käyttivät vain yhden sairaanhoitopiirin dataa. Kliinisen validaation sekä mallien yleistyvyyden arviointiin ei siis tämän katsauksen perusteella pystytä. Huomioitavaa on kuitenkin multimodaalisten mallien tarkkuuden paraneminen verratten vain yhden modaliteetin hyödyntämiseen. Katsaukseen valituista papereissa lääketieteellisistä alueista neurologia ja syöpä olivat eniten edustettuina. Sähköisten terveystiedot ja kuvantamisdata olivat eniten edustettu yhdistelmä modaliteettejä.

Tekoälyn soveltaminen lääketieteellisessä kliinisessä kontekstissa on laaja ongelma. Multimodaalisen datan käyttöön pohjautuvat mallit ovat antaneet lupaavaa näyttöä mahdollisesta tarkkuuden paranemisesta verrattaessa yhden modaliteetin malleihin. Koneoppimiseen pohjautuvien järjestelmien läpinäkyvyydellä ja päätösten selitettävyydellä on merkittävä rooli, kun ajatellaan potentiaalista kliinistä käyttöä.

Contents

1 Introduction

Diagnosing diseases or conditions is a typical task for physicians working in healthcare. Physicians use relevant data that describe the medical condition of the patient. The data needed depends heavily on the context but can be simplified as using the current and historical data available to identify the disease or condition. Additional data can be gathered from medical imaging or lab tests (Institute of Medicine and National Academies of Sciences, Engineering, and Medicine, 2015). The complexity of the diagnosing process varies case by case but can be seen as a classification problem in the end.

Machine learning has been used across different industries to solve classification problems, and deep learning has shown potential for complex classification tasks. The capability of the machine learning model is highly dependent on the data that it is trained with. Deep learning has been successfully used for image segmentation and classification in radiology (Lee et al., 2017). In medicine, data with multiple modalities namely electronic health records (EHR) and medical images are used together to gain a better understanding of the patient during the diagnostic process. However, machine learning models traditionally expect data from a single modality. Multimodal machine learning (MML) addresses the issue by introducing data fusion for multiple different sources of modalities during the training process (Baltrušaitis et al., 2019).

This thesis provides a basic view of the concepts in multimodal data fusion. This is achieved by introducing basic machine and deep learning concepts and then moving to data fusion. These concepts are then reviewed through recent studies utilizing the multimodal approach in medical contexts. This thesis tries to identify the potential benefits and challenges that can be found from using multiple modalities compared to single-modality approaches.

Research Questions:

1. What are the basic principles of multimodal machine learning and data fusion?
2. How do recent studies employing multimodal machine learning techniques in medical contexts demonstrate the potential advantages and challenges compared to single-modality approaches?

2 Machine learning methods

This thesis focuses on supervised machine learning models used in classification problems. Some non-supervised techniques to reduce dimensionality are also introduced. Supervised learning is a type of machine learning in witch the model is trained with labeled data. Formally, a training dataset with n samples can be described as a set of feature vectors X and a corresponding set of labels Y:

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \ , Y = \{y_1, y_2, \ldots, y_n\}$$

where each feature vector $\mathbf{x}_i$ is a d-dimensional vector: $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$. The d indicates the number of features per sample. The goal in classification is to create a model that can assign a correct label y to unseen feature vector $\mathbf{x}_i \notin X$.

This chapter follows mainly the ideas presented in a book: *Deep Learning* (Goodfellow et al., 2016). In this chapter, we define some of the fundamental algorithms and methods that are necessary to understand before moving into more complex (MML) models that utilize many data modalities simultaneously.

2.1 Artificial neural networks and deep learning

Deep learning, or artificial neural networks with multiple layers, has improved performance in multiple machine learning domains like computer vision and natural language processing (LeCun et al., 2015). The ability to automatically discover abstract representations from data allows complex functions to be learned (LeCun et al., 2015).

The basic structure of the neural network is a set of layers with an input layer, n hidden layers, and an output layer. Components of these layers are nodes, also referred to as neurons. The idea for the node and the whole fundamental structure of what later evolved to artificial neural networks, is a perceptron that was introduced by (Rosenblatt, 1958). Research for artificial neural networks was already hot in 1980 due to the discovery of key algorithms such as backpropagation (Karhunen et al., 2015). Interest was declining until deep learning was introduced by Hinton et al., 2006, combined with increasing computing power and data amounts (LeCun et al., 2015). This can be seen as the starting point for modern deep learning architectures.

At the input level of a multilayer perceptron (MLP), nodes represent the features that are input to the network. Similarly, the output layer is a set of nodes representing the possible labels in the dataset. For example, identifying a single number when we have 20x20 pixels defining a picture and labels that correspond to that number (1-9), the neural network for this task would have an input layer with 400 nodes and an output layer with 9 nodes. Hidden layers are the layers between the input and output layers. The number of nodes in each hidden layer, and the number of hidden layers are not fixed, like input and output layers. In a fully connected feedforward network (FFNN), all of the nodes per layer are connected with some set of weights and biases to each node in the next layer. In these kinds of networks, the number of weights between layers 1 and 2 is $n_1 \times n_2$, where n represents the number of nodes per layer. Each node, excluding the input layer, also has some bias associated with it.

2.1.1 Forward propagation

Forward propagation defines the actions that have to be made when feeding data from the input layer through the network. This is done to produce the output layer, which predicts the label corresponding to the input data. The first step in the data flow through the network is to calculate the linear transformation for the nodes in the next layer. With n input features in FFNN, the linear transformation for a single neuron in the first hidden layer can be defined as

$$z = \sum_{i=1}^{n} (w_i \times x_i) + bias$$

where w represents weight and x a value associated with a node in the input layer or activation in the previous layer. Activation of a node is then calculated by applying the activation function to the z. Dataflow through a single node is illustrated in Figure 2.1. The activation function is applied to break the linearity. There are many options when choosing an activation function,

$$\text{Sigmoid}(z) = \frac{1}{1 + e^{-z}} \text{ and } \text{RelU}(z) = \begin{cases} x \text{ if } z > 0 \\ 0 \text{ otherwise} \end{cases}$$

are common, many others are based on these with some modification. For the output layer, a representation that can be viewed as probabilities is wanted. The sum of the activations in the output layer is 1 and each node can get a value in the range [0,1]. This is achieved by applying a softmax activation function to the linear transformation of each node in

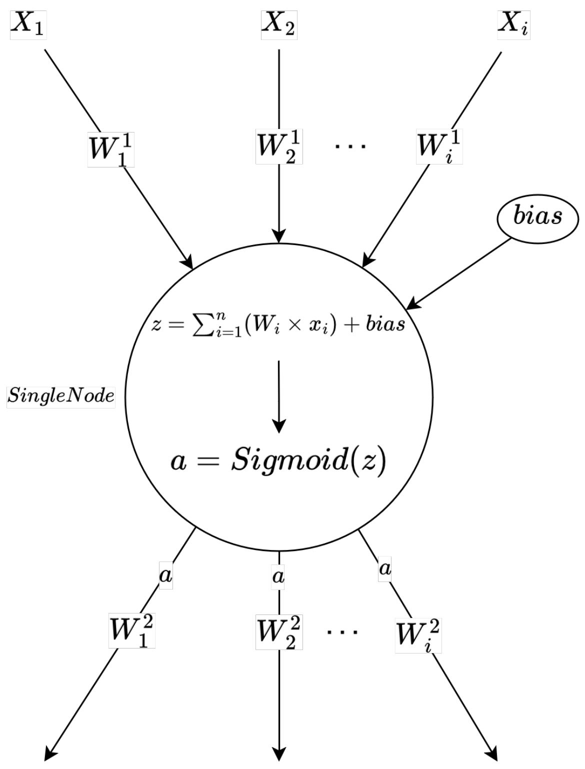


Figure 2.1: Forward propagation zoomed in single node

the output layer. With n nodes in the output layer softmax of the linear transformations $z = \{z_1, z_2, \dots, z_n\}$ can be calculated as:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$

Forward propagation can be seen as an algorithm that produces the output vector containing the label probabilities with a given input vector X based on the weights and biases of the neural network by feeding the data through the network and calculating the activations for each layer iteratively. The structure of a node or neuron and the idea of activation can be seen influenced by human brains (Rosenblatt, 1958). Forward propagation is sometimes misunderstood as part of backpropagation, but it is an own separate algorithm.

2.1.2 Backpropagation

The mathematical concept of backpropagation was first introduced as a reverse mode of automatic differentiation without direct use of neural networks (Linnainmaa, 1970).

Later as a part of training the neural networks referred to as backpropagation. With forward propagation, we can produce outputs of the neural network. Backpropagation is an algorithm that is used to find the impact of each weight and bias term on the prediction error. Prediction error, also known as the loss or cost function, defines the disparity between the predicted and desired output. As the correct label is often presented as a single numerical value it needs to be converted into a similar representation as the output of the network. This can be done with one-hot encoding. A single sample from a dataset with 9 possible classes: $\hat{y} = \{0.12_1, 0.02_2, \ldots, 0.33_9\}$ where the correct label is 2, can be one-hot encoded to $y = \{0_1, 1_2, \ldots, 0_9\}$. There are multiple ways of calculating the difference or loss between the correct label and the predicted one. One of them is cross-entropy which is suitable for multiclass classification.

$$\mathrm{CE}(y, \hat{y}) = -\sum_{l}^{L} y_l \log(\hat{y}_l)$$

With backpropagation, we calculate the gradients of the loss function with respect to weights and bias terms. Backpropagation is sometimes misunderstood as the algorithm that optimizes the network (Goodfellow et al., 2016). The gradients can be thought of as a partial derivatives in a vector. We calculate these gradients for each layer moving backward from the output towards the input layer by layer. This is done because, with this approach we can use the chain rule to effectively use the already calculated values from previous layers to reduce the computation time. Backpropagation in FFNN multiclass classification with two input features, two hidden layers, and three output classes is illustrated in 2.2, where L is used as the notation for the loss.

2.1.3 Stochastic gradient descent

Stochastic gradient descent (SGD) is an algorithm to optimize the neural network to minimal loss using forward and backpropagation. Some researchers have suggested that SGD may be the most commonly used algorithm to optimize neural networks (Dean et al., 2012). Weights and biases are essentially the values that we want to optimize, so the network output loss is minimized. Most of the deep learning algorithms are based on SGD (Goodfellow et al., 2016), so it can be seen as a fundamental part of deep learning.

The optimization for weight and biases with SGD happens after every sample. A sample is forward propagated through the network to produce predictions. Then the predictions are used to calculate loss and backpropagation is used to find the gradients to weights and

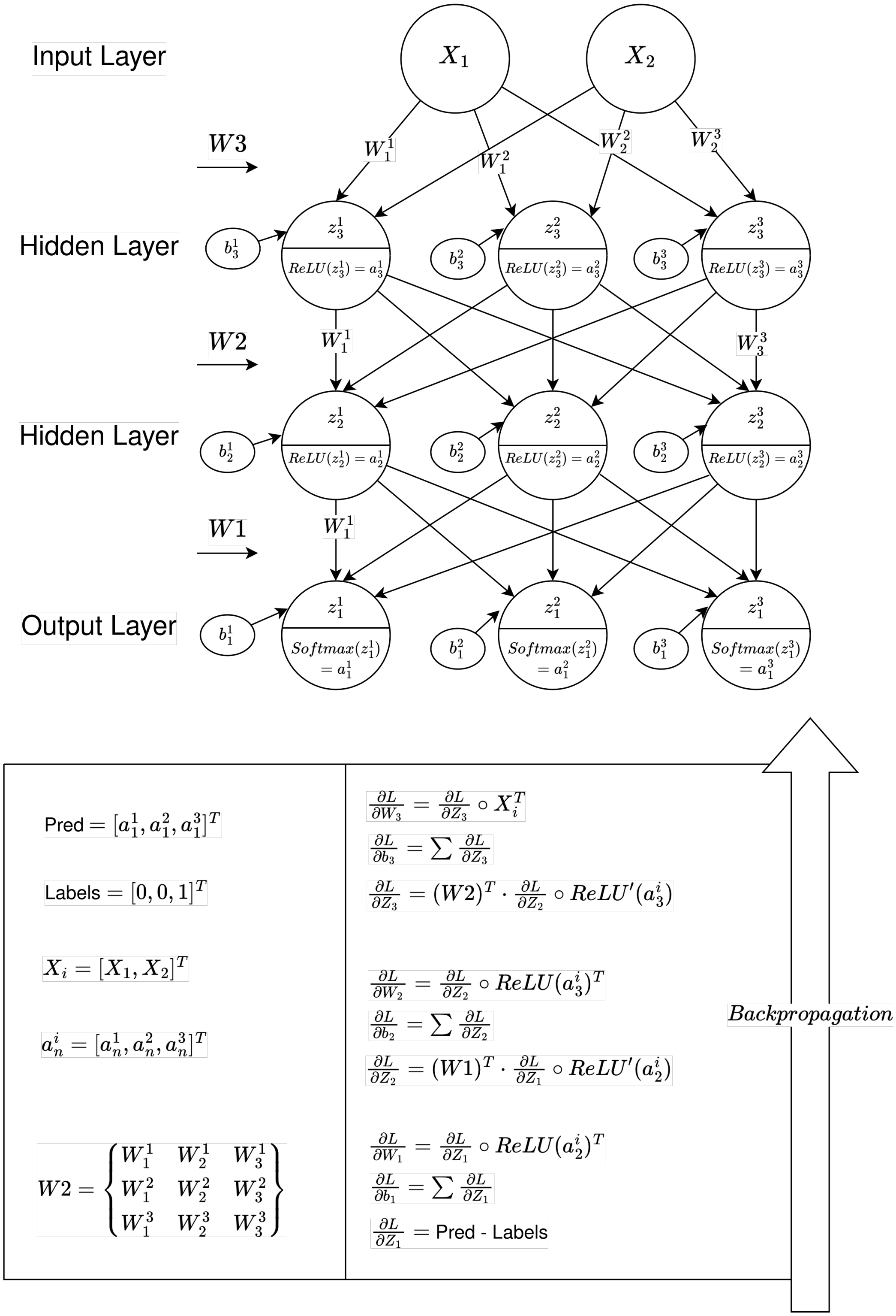

Input Layer
X_1
X_2
$W3$
W_1^1
W_1^2
W_2^2
W_2^3
Hidden Layer
b_3^1
z_3^1
$ReLU(z_3^1) = a_3^1$
b_3^2
z_3^2
$ReLU(z_3^2) = a_3^2$
b_3^3
z_3^3
$ReLU(z_3^3) = a_3^3$
$W2$
W_1^1
W_3^3
Hidden Layer
b_2^1
z_2^1
$ReLU(z_2^1) = a_2^1$
b_2^2
z_2^2
$ReLU(z_2^2) = a_2^2$
b_2^3
z_2^3
$ReLU(z_2^3) = a_2^3$
$W1$
W_1^1
Output Layer
b_1^1
z_1^1
$Softmax(z_1^1) = a_1^1$
b_1^2
z_1^2
$Softmax(z_1^2) = a_1^2$
b_1^3
z_1^3
$Softmax(z_1^3) = a_1^3$
Pred $= [a_1^1, a_1^2, a_1^3]^T$
Labels $= [0, 0, 1]^T$
$X_i = [X_1, X_2]^T$
$a_n^i = [a_n^1, a_n^2, a_n^3]^T$
$W2 = \begin{Bmatrix} W_1^1 & W_2^1 & W_3^1 \\ W_1^2 & W_2^2 & W_3^2 \\ W_1^3 & W_2^3 & W_3^3 \end{Bmatrix}$
$\frac{\partial L}{\partial W_3} = \frac{\partial L}{\partial Z_3} \circ X_i^T$
$\frac{\partial L}{\partial b_3} = \sum \frac{\partial L}{\partial Z_3}$
$\frac{\partial L}{\partial Z_3} = (W2)^T \cdot \frac{\partial L}{\partial Z_2} \circ ReLU'(a_3^i)$
$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial Z_2} \circ ReLU(a_3^i)^T$
$\frac{\partial L}{\partial b_2} = \sum \frac{\partial L}{\partial Z_2}$
$\frac{\partial L}{\partial Z_2} = (W1)^T \cdot \frac{\partial L}{\partial Z_1} \circ ReLU'(a_2^i)$
$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial Z_1} \circ ReLU(a_2^i)^T$
$\frac{\partial L}{\partial b_1} = \sum \frac{\partial L}{\partial Z_1}$
$\frac{\partial L}{\partial Z_1}$ = Pred - Labels
$Backpropagation$

Figure 2.2: Backpropagation

biases. These gradients are then used to tune the weights (W_i) and biases: (b_i)

$$W_i = W_i - \alpha \frac{\partial L}{\partial W_i}$$

$$b_i = b_i - \alpha \frac{\partial L}{\partial b_i}$$

Here α is some scalar that we use to define the step size towards the direction defined by the negative gradient. The gradient points in the fastest increasing direction, so by using a negative gradient, these parameters can be adjusted toward the direction that decreases the loss function. By repeating the steps defined above for multiple samples multiple times, we can find the local minima for the loss. By optimizing the weights and biases to perfection, we train the network to produce minimal error with the training data. This leads to a problem called overfitting. Overfitting happens when we get flawless outputs with the training data but can't generalize to unseen data (Ying, 2019). The goal of the training is to optimize the network to produce correct labels with unlabeled data, so by overfitting, we fail to achieve this goal.

Training data is commonly split into testing and training data (Goodfellow et al., 2016). During the training and splitting the data is shuffled. Test data gives us a portion of the data that we don't use during the training and can be used for testing as unseen data to the network. This helps to identify the overfitting and shows more accurately how the network performs outside the data it has learned from. Additional parameters to the network that are used in training are called hyperparameters. Learning rate or α is a hyperparameter. The training loop goes through the whole training dataset multiple times during the optimization, "epoch" is a hyperparameter that defines how many times we crawl through the entire training data. Training can also be set to a halt when test data accuracy starts rising while the training data accuracy still decreases, this is considered an early stop (Ying, 2019). Penalties to the loss function can be set to motivate lesser weights to be used. Other techniques like dropout can be added, more data can be gathered or noise added (Ying, 2019). Dropout is a technique where random nodes are deleted during the training. Ultimately the optimal hyperparameters and techniques are case-specific but generally avoiding common known problems is helpful.

Training may be extended, particularly with large deep networks, due to the iterative nature of SGD optimizing parameters after processing each sample. A similar algorithm is batch gradient descent, also known as batch learning. This method uses the entire training data before updating the parameters. The batch method finds the true gradient instead of an estimate as in SGD but is slower and SGD often produces better results

(Lecun et al., 2000). Mini batch gradient descent is a method where training is done via batches. Batch size is an additional hyperparameter that defines the batch size. This method splits the training data in each epoch into a mini-batch consisting of multiple samples that are then all forward propagated through the network simultaneously. Then the backpropagation is done similarly to all the predictions once, and the average gradients are used to update the weights.

2.1.4 Types of neural networks

Based on the simple neural network many architectures have been built to suit specific domains or problems. They expand or add to the basic idea of neural networks and utilize similar algorithms in training.

Convolutional neural networks (CNN) introduce the concepts of convolution and pooling. CNNs are designed to handle data that has a grid-like topology and are therefore suitable for image-related tasks (Goodfellow et al., 2016). The simplified concepts of pooling and convolution for a 6x6x1 image are illustrated in Figure 2.3. Values in the kernel are considered learnable weights that are tuned similarly to regular weights connecting neural networks. Multiple kernels can be used in a single convolutional layer, and these layers can be vertically stacked to capture higher dimensional features. Pooling and convolutional layers can be feature extraction layers that eventually are flattened and fed into a regular fully connected neural network that produces the output. Stride defines how much the kernel is moved when it is applied. A 3x3 kernel applied with stride = 1 to 6x6x1 would mean 16 dot products in total. If we had three-channel pictures, for example RGB, a kernel would have the same depth. Pooling layers calculate the maximum or average of the kernel-sized region that is similarly slid over the grid. Additional noise by flipping or zooming the images and padding to the borders are common methods that are applied to CNNs to reduce overfitting. CNNs can significantly reduce the number of parameters in the network compared to FFNNs (O'Shea and Nash, 2015) and capture features of the image effectively. The training follows similar steps and is based on the same fundamental algorithms as described with MLP.

Recurrent neural networks (RNN) are another commonly known architecture. RNNs are used with sequential data, like stock prices, weather history, or natural language. FFNNs processes all the inputs simultaneously, while RNNs processes sequential data by applying each input in order to the model (Schuster and Paliwal, 1997). This allows the use of

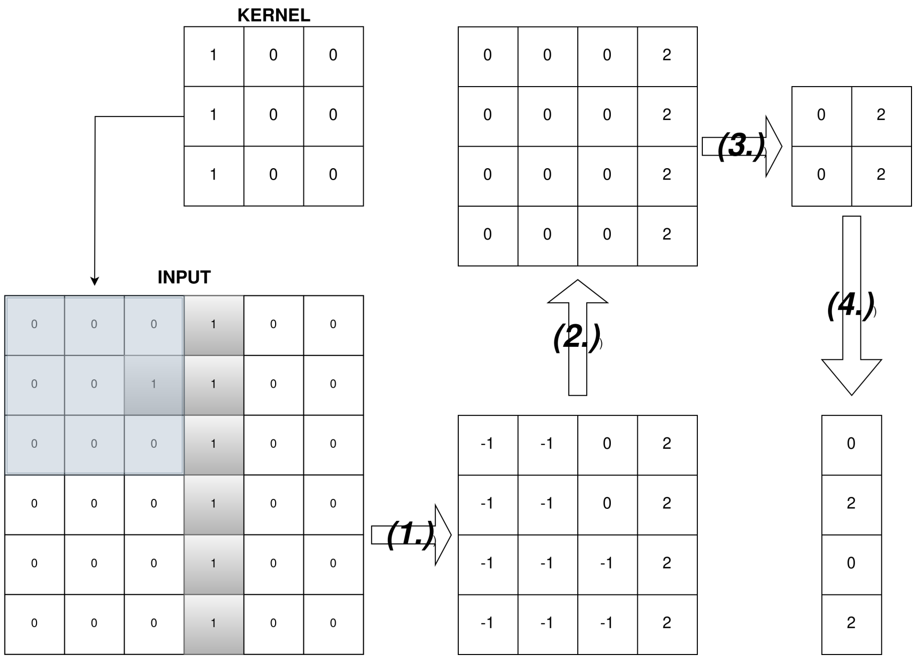


Figure 2.3: Key concepts of feature extraction layers in CNNs: (1.) Convolution with single 3x3 kernel(stride(1)) and bias[-1], (2.) Elementwise ReLU activation for the 4x4 feature map, (3.) Max-pooling (stride(2)), (4.) Flattening 2x2 to 4x1

recurrent connections that can carry information from the previous part of the sequence to the next part. Suppose we have a sequence of some signal of integers in sequence $X = \{4_{t1}, 5_{t2}, 6_{t3}\}$. The first input would be 4, and the recurrent connection M set to 0, by feedforwarding this to network, we would then have some updated M_1 that could affect the next input of the sequence. The architecture shares the weight and biases so it can be easily unrolled as the only changing part is the input and the incoming value from the recurrent connection as M_i. The unrolling and vanilla RNN are illustrated in A.2. These blocks can be stacked horizontally to make the network deeper. The unrolling and the shared parameters also allow flexibility on inputs as the blocks can be unrolled according to the input sequence length. The use of activation functions, weights, and biases in the operations and connections behave similarly to the MLP. Training is done similarly with forward propagation, backpropagation, and gradient descent with some modifications to address the recurrent aspects (LeCun et al., 2015). Unrolling long sequence inputs shows the main problems in RNNs, vanishing gradients, and the inability to retrain information in the memory (Hochreiter, 1998). Vanishing or exploding gradient is a problem where, during the backpropagation the gradients either vanish or explode, making the optimization difficult.

Long short-term memory network (LSTM) is a modification of RNNs introduced to address the vanishing gradient problem and use the memory aspect more effectively (Hochreiter and Schmidhuber, 1997). Many modifications built on top of the idea of vanilla RNN:s exist but the LSTM is one of the well-known ones. The LSTMs behave similarly to vanilla RNNs, a single block is illustrated in A.2. LSTM block has 3 gates: forget gate, input gate, and output gate (Gers et al., 1999). These gates define the amount of memory that is kept from memory and input. Activation functions such as *Sigmoid* which maps input to [0,1], and *Tanh* that maps the input to [-1,1] are used to scale these gates. Memory is also known as long-term memory and input as short-term memory. Similarly to vanilla RNN:s, a single block takes inputs sequentially but has 2 recurrent connections coming out of each block. These recurrent connections are not tied up with weights and therefore handle the vanishing gradient problem better.

RNNs can produce multiple and single outputs, and they can also be connected to a regular feedforward network. The main advantage of these architectures is their ability to use previous parts of sequences to influence the next parts with memory. One currently trendy and powerful architecture is a Transformer, introduced by Vaswani et al., 2023, which outperforms RNNs in multiple tasks. The Transformer model is built on top of the concepts of RNNs, but the specifics of the new things that this model introduces, such as the attention mechanism, will not be discussed in the scope of this thesis.

2.2 Dimensionality reduction methods

Neural networks and deep learning in general are known to need large amounts of training data. Datasets containing hundreds or more features can be described as high dimensional and are found in multiple domains such as medicine (Reddy et al., 2020). Dimensionality reduction methods aim to reduce the dimensionality and redundancy by mapping the data into a lower dimensionality space. These methods can be used for preprocessing the data in multiple machine learning classification architectures that deal with high dimensional data.

Principal component analysis (PCA) is a well-known linear dimensionality reduction method. PCA finds the lower dimensional representation of the data with the maximum amount of variance explained (Van Der Maaten et al., 2009). PCA starts with the normalization of the data and then computes the covariance matrix of the data to find eigenvectors and eigenvalues for each feature. The eigenvalue represents the order of the principal compo-

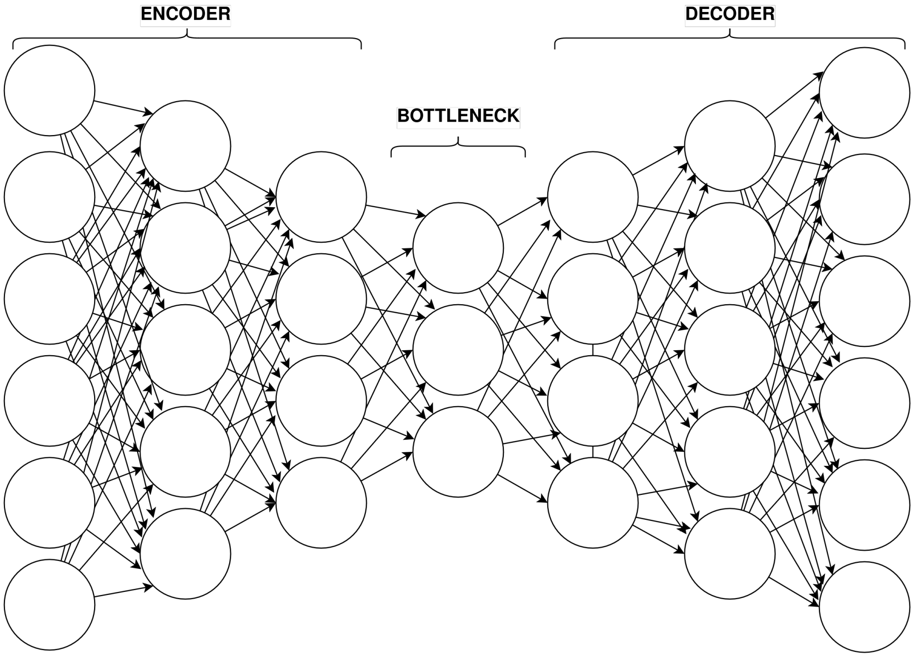


Figure 2.4: Illustration of the MAE structure

nents descending from the largest to the smallest. Eigenvalues also indicate the amount of variance that the component can capture. Eigenvectors can then be used to project the data into the wanted amount of these components.

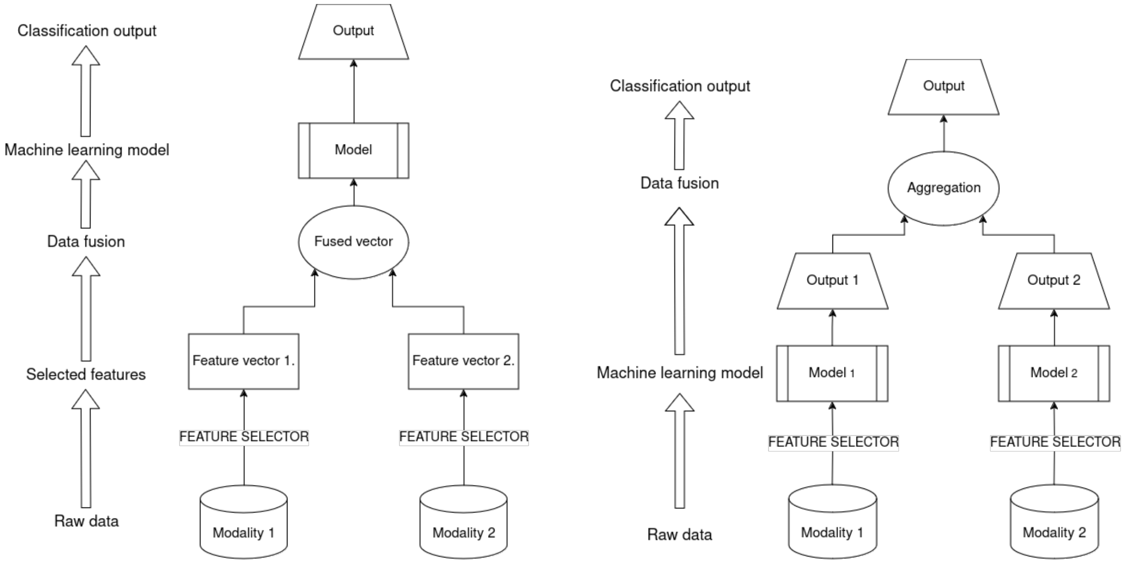
Autoencoders (AE) can also be used as dimensionality reduction methods. Autoencoders are FFNNs that aim to learn the input from reduced dimensions. This is done by forcing the data into lower dimensions and reconstructing the input from there. Multiple variations of autoencoders exist, such as multilayer autoencoders (MAE), which are simply networks with more than one hidden layer. Autoencoders hold an odd number of hidden layers and can be described with an encoder and a decoder that have shared weights (Van Der Maaten et al., 2009). The bottleneck is the middle layer in the network that captures the lower dimensional representation of the input. An illustration of MAE with six hidden layers, an input layer size of 6, and a bottleneck size of 3 is found in 2.4. The encoder can be thought of as the layers before the middle layer in the network that reduces the input data to the bottleneck. The bottleneck is the layer in the middle that holds fewer nodes than there are features in the data, forcing the compression of the data. The decoder is then taught as layers after the bottleneck reconstructing the input from the bottleneck. By training this kind of network we can learn a lower-dimension representation of the data by using the data from the bottleneck layer that is forced to produce it.

3 Multimodal data fusion

The motivation for using multiple modalities is the additional information that could be gained (Ramachandram and Taylor, 2017). Different medical imaging methods may give more detailed information when used together, whereas tabular data may give additional information that can't be gathered from images alone. Building a machine learning model to solve a problem with data from multiple modalities eventually leads to an issue of using the different modalities together. Data fusion combines multiple modalities into a single vector or outcome, depending on the approach. One definition of data fusion is the analysis of several data sets such that different data sets can interact and inform each other (Lahat et al., 2015). Different diseases benefit more from certain approaches to data fusion. For example, fusion approaches that enable cross-modal interactions are essential for certain diseases, whereas others may not benefit as much from such interactions. The nature of the problem and data that is available influences the approach's effectiveness. This chapter aims to provide an overview of the common fusion architectures and difficulties associated with data fusion. Fusion strategies can be used for more than two modalities, but for minimum complexity, the approaches are presented with two modalities. Unimodal models are referred to when classifying with a single modality.

3.1 Early, intermediate, and late fusion

Early fusion is considered when the input modalities are fused into a single representation vector before feeding into the model. Early fusion uses only one model, and the difference compared to unimodal models is the input that is fused. Concatenating is one of the strategies to fuse data within early fusion (Stahlschmidt et al., 2022). Concatenation of $\vec{x} = [1, 2, 3]$ and $\vec{y} = [4, 5, 6],$ is $\vec{x} \oplus \vec{y} = [1, 2, 3, 4, 5, 6]$. Feature extraction can help the fusing but it is seen as preprocessing not as a part of the model. Finding a common subspace for data with varying dimensions and removing correlations between modalities is important for successful early fusion (Ramachandram and Taylor, 2017). Early fusion approaches can not propagate loss back to the original input features which is a key difference compared to intermediate fusion methods. Early fusion can utilize any supervised machine learning algorithm. Simplified architecture for an early fusion model is illustrated in 3.1 (a).



(a) Early fusion (b) Late fusion

Figure 3.1: Early and late fusion

Using a separate model for each modality where each model outputs a prediction and combining these predictions to produce the final output is considered as late fusion. Probabilities from the independent model outputs are aggregated to produce the final output. The aggregation method varies based on the modalities and the application, for example, averaging and voting-based approaches can be used (Huang et al., 2020). Outputs of independent models can also be used as input to another model that makes the final prediction. Late fusion approaches make minimal use of the combined effects of modalities compared to early and intermediate approaches since each prediction is done independently. Simplified architecture for late fusion is illustrated in 3.1 (b).

Intermediate fusion is an approach where the architecture uses multiple networks together. The model can be separated into multiple unimodal neural networks that towards the end are combined in shared layers. The input to the model is the data from the modalities and the extracted features from independent models are combined into the shared representation layer that in the end produces the output of the model. Figure 3.2 shows a simplified structure of the intermediate fusion. Fusion between the modalities can also be done in different stages allowing flexibility and PCA or stacked autoencoders can be used to improve the representation in shared layers (Ramachandram and Taylor, 2017). Using neural networks in feature extraction and after the fusion allows the loss to be propagated back to the input modality level. This allows better feature extraction since the inde-

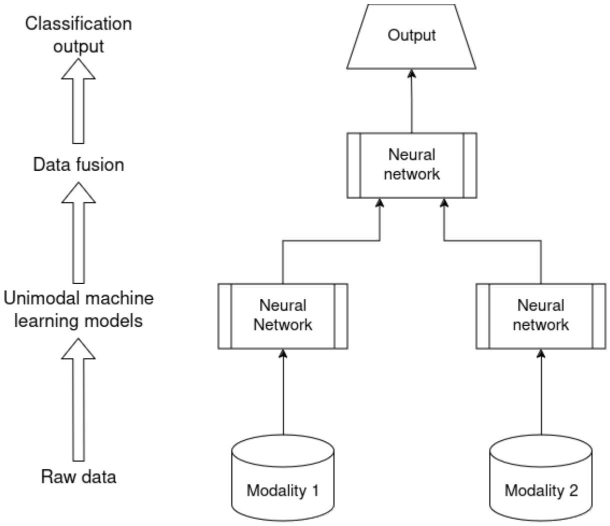


Figure 3.2: Intermediate fusion

pendent models are also optimized during the training. Intermediate fusion makes use of cross-modal interactions like early fusion and modality-specific models, the same as in late fusion approaches.

Early, intermediate, and late fusion can be seen as the baseline for the fusion approaches. While no single approach is proven to be supreme, studies often evaluate and combine different approaches and methods suitable for the given task. Each of the approaches has advantages and disadvantages. It have been suggested that various unimodal and multimodal approaches should be evaluated when building multimodal models (Huang et al., 2020), since unimodal models can be used in multimodal approaches and the most suitable fusion approach varies and depends on the problem. Considering the structure of the data fusion approaches, late fusion is suitable in situations where the modalities contain independent aspects of the data. Early fusion can make use of the cross-modal interactions. However, to make use of cross-modal interactions successfully, it might require specific preprocessing techniques. Late fusion architecture can adapt to missing modalities, offering ease when dealing with datasets where data contains different sets of modalities per sample. Independent unimodal models in the late fusion approach also offer easy addition of new modalities as the training can be done only for the additional unimodal models. Early fusion architectures may need retraining of the whole model if new modalities are added. Similarly, when the training is done with data where all modalities are present in each sample, handling missing modalities poses an issue. Relying on one model only reduces complexity compared to models relying on multiple models. Intermediate fusion offers parts from both of these with additional complexity.

Literature review reported that fusion with medical images and clinical data using multimodal fusion generally increased accuracy (1.2–27.7%) and Area Under Receiver Operating Characteristic Curve (AUROC[*]) of (0.02–0.16) compared to models using a single modality for the same task (Huang et al., 2020). However, no single fusion strategy performed consistently optimally over all domains. In another review, multimodal data fusion was reported to increase the accuracy over unimodal approaches with a 6.4% mean improvement in Area Under Curve (AUC[*]) (Kline et al., 2022). The measurements of increased accuracy with multimodal data fusion do not mean that these models would be beneficial for clinical settings and real-world applications. End-to-end applications and prospective studies are needed to validate the relevancy of these models.

3.2 Challenges with multimodal machine learning

Challenges within multimodal machine learning have been outlined as follows (Baltrušaitis et al., 2019):

1. *Representation*: how multiple modalities are meaningfully represented.

2. *Translation*: involves the mapping of data from one modality (A) to another (B).

3. *Alignment*: which parts of one modality directly correspond to another.

4. *Fusion*: how to combine data from multiple modalities, particularly when some data may be missing or certain modalities are more informative than others.

5. *Co-learning*: how to transfer knowledge between different modalities.

All of these are important technical aspects for the successful multimodal machine learning model. It is important to notice that the classification problem itself and the modalities affect heavily the issues that may arise. On top of the five key challenges related to multimodal machine learning and the context-specific ones, traditional problems in machine learning can be similarly found in multimodal architectures.

3.2.1 Explainability

When machine learning is applied to critical fields such as medicine where the usage might have an impact directly on patient health it is crucial to have explainability on the mod-

[*] AUC and AUROC are used interchangeably and noted here as they appear in the original paper. More about AUC and AUROC in A

els (Hassija et al., 2023). Models that use deep learning are often thought of as black boxes and lack explainability. Multimodal machine learning models can have multiple of these employed. Machine learning models and especially deep learning models carry data in abstract representations that might be hard to understand. Similarly, many data fusion methods with feature extractions and shared representations hold data in abstract formats. To understand a decision that is made by the ML model additional metrics or methods can be applied. The eXplainable Artificial Intelligence (XAI) can be seen as a field that aims to produce these methods (Adadi and Berrada, 2018). An example of this is seen in Shapley values that have been used for measuring the contribution from different modalities (Soenksen et al., 2022). Explainability and justification of classification decisions in the models utilized is one of the key challenges to when machine learning models are utilized in medical context.

3.2.2 Data in medical context

The medical field offers vast amounts of datasets that derive from retrospective studies. Biobanks, national registries, and open-source datasets are sources of data in multimodal medical studies. Datasets such as MIMIC (A. Johnson et al., 2023) that are open source are used as a benchmark to compare models. The prospective studies to validate models built with retrospective data are an important part of understanding the true utility and real-world performance of these systems (Kelly et al., 2019). A specific dataset is needed to apply multimodal machine learning to certain diseases. Privacy in the medical context also limits the data available. In a recent review, small sample sizes and imbalanced samples were reported as common limitations to studies employing multimodal machine learning models (Kline et al., 2022). Small sample sizes and high dimensionality combined increase the sparsity of the data which can also lead to a curse of dimensionality (Altman and Krzywinski, 2018). The curse of dimensionality appears in problems like overfitting. While the curse of dimensionality and various problems associated with are well-known in machine learning, assessing them is a necessary step when building multimodal machine learning models. Choosing the correct architecture is an essential part of building a multimodal machine-learning model. Comprehensive evaluation requires the building of multiple models. Conducting high-quality research in this area can be challenging because it involves complex computational models, requires validation studies, and needs expertise from various fields.

4 Applications of multimodal machine learning

A recent literature review and highlighted the increased interest in multimodal machine learning (MML) research (Barua et al., 2023). In theory, MML approaches can be applied to any domain or problem that links to multiple modalities already solved with machine learning. While the concept of data fusion is not new, deep learning has enabled many possibilities in this field. Particularly within the medical context, data fusion, and MML are a relatively new and currently researched area. Applications of MML appear in recent research papers, but not yet in clinical environments.

4.1 Applications in medicine

The medical domain provides many problems that could benefit from MML approach. Biobanks, health centers, registers, and research provide large amounts of different sets of modalities. Literature review ranging from years 2011–2021 of MML in health for diagnosis/prognosis tasks found most papers were from neurology and oncology domains (Kline et al., 2022). Recent Finish article: "The mathematics and statistical models in predicting treatment response in cancer" pointed out, that one of the biggest challenges is the efficient integration of different modalities to achieve the best predictive accuracy (Miihkinen et al., 2024). Biomedical data opens opportunities for a wide amount of other tasks than prognosis and diagnosing. Potential use cases could be in drug discovery, remote monitoring, and many other applications in healthcare (Acosta et al., 2022). This section focuses on the studies on prognosis and diagnosis as downstream tasks and showcases the use of 2, 3, and 3+ modalities in their respective order. All of these showcased studies are retrospective. It's important to note that usage of machine learning in medical context

	Classification task	Modalities	Fusion approach
Cahan et al., 2023	PE risk	2	early,late,intermediate
Venugopalan et al., 2021	AD, MCI, controls	3	early,late,intermediate
Soenksen et al., 2022	Multiple	2-4 (2-11)	early

does not necessarily aim to remove the work of physicians but may help them for example to identify those patients that need specific treatment immediately. Similarly, MML can help to identify the possibility of some diseases earlier.

Cahan et al., 2023 applied MML methods to predict the risk stratification of pulmonary embolism (PE). Evaluation and comparison were done by building multiple models with different fusion strategies in this retrospective study. Three encoders were built to be used as encoders in multimodal settings and as unimodal baseline models for image and EHR modalities. Previously, Cahan et al., 2022 introduced CNN-based Stacked Attention Network (SANet), which was used as the main imaging encoder, but also compared to the second baseline image encoder, Swin UNETR Transformer (Tang et al., 2022), in intermediate fusion architectures. Futhermore, TabNet (Arik and Pfister, 2020), a neural network architecture for tabular data was used as a baseline encoder for EHR modality. For downstream tasks, XGBoost or TabNet was used. Additionally, for early and intermediate fusion architectures, dimensionality reduction methods were evaluated for the fusion layer after concatenation. The bilinear attention network (BAN) (Kim et al., 2018) was chosen as an additional layer to further enhance the fusion. The best-performing architecture was an intermediate fusion with SANet and TabNet encoders followed by concatenation and BAN for the fusion layer before the last TabNet as a downstream classifier. This architecture significantly increased AUC over the unimodal baseline models showing the possible value gained by using multimodal architecture. Identifying PE early and proper treatment can decrease the deaths caused by this disease (Cahan et al., 2023).

Venugopalan et al., 2021 presented MML approach to classify Alzheimer's disease (AD), mild cognitive disorders (MCI), and controls. Imaging (MRI), genetic (SNP), and clinical (EHR) modalities were used and evaluated in different combinations. The Alzheimer's Disease Neuroimaging Initiative (ADNI) (Mueller et al., 2005) was used as a data source for this study. The backbone of the Deep MML architecture was built using feature extraction for each modality. MRI features extracted with 3D CNN. SNP and EHR data features were similarly extracted with stacked denoising auto-encoders. Each of these models was trained independently. Extracted features are then used as input after concatenation for the classifier layer making it an early fusion approach. Other deep fusion approaches were not evaluated. K-nearest neighbors (kNN), support vector machines (SVM), random forests, and decision trees were compared as downstream classifiers. This model was compared to shallow models with early and late fusion approaches. Both shallow models employed decision trees and the late version used majority voting. A comparison between models using single modalities was done to highlight the performance of deep models over shallow ones. Deep unimodal approaches outperformed kNN, decision trees, random forests, and SVM approaches with SNP and MRI modalities. Decision trees and random forests achieved similar performance with deep approaches in EHR modality. Similarly, the two shallow models were compared to deep models with kNN, SVM, decision trees, and random forests as downstream classifiers in all possible combinations of modalities. All deep approaches performed better compared to shallow models in every combination of modalities excluding the MRI + SNP. This was explained with small sample sizes. This study demonstrated the possible benefits that can be gained with deep fusion approaches. Similar findings were found in review which stated that leveraging deep learning fusion consistently showed improvements in performance for Alzheimer's disease diagnosis (Huang et al., 2020). Identifying the AD or MCI early would be beneficial for patients and society as the costs decrease and the progression slowed (Rasmussen and Langerman, 2019).

Soenksen et al., 2022 introduced and evaluated a unified holistic AI in medicine (HAIM) framework. HAIM is built to handle nonfixed-size sets of modalities. Each modality is handled independently with feature extraction models that generate a set of embeddings that can then be used for multiple downstream tasks. This architecture can be modified with any set of modalities and feature extraction models to whatever state-of-the-art models are available. This is seen as early fusion architecture where the machine learning models act as feature extractors. The evaluation of the proposed framework was done with open-source data from MIMIC-IV (A. Johnson et al., 2023) and MIMIC-CXR-JPG (A. E. W. Johnson et al., 2019). Images, tabular data, natural language, and time series were used as modalities with 11 sources. For images pre-trained Densenet121 (Cohen et al., 2021) was used as a feature extractor, similarly for natural language pre-trained Clinical BERT (Alsentzer et al., 2019) was used. The tasks for evaluation included 10 different chest diagnoses, length-of-stay, and 48h-mortality. XGBoost was used as the final classifier for all of these downstream tasks. Shapley values were reported to analyze the benefits gained from different modalities and models with different combinations of data sources and modalities compared to single-source models. Vision data was found to contribute most to performance in chest diagnostic tasks, while length-of-stay and 48-hour mortality time-series were identified as the most relevant modalities. Multimodal approaches consistently outperformed the single-source models in AUROC with an average improvement of 9-28% across all downstream tasks.

4.2 Applications in other fields

Multimodal machine learning can be found in a wide variety of applications and research from multimedia, robotics, and human-computer interaction (Liang et al., 2023). While each field approaches problems from a different angle, the solution can benefit many. Autonomous vehicles could benefit from using multiple modalities but must also handle the data quickly in real-time. Prakash et al., 2021 introduced a model called TransFuser that utilizes multiple fusion layers between convolutional layers in feature extractors for LiDAR and front-facing cameras. Interestingly the latest versions of widely popular large language models are introduced as multimodal models as they can out and input multiple modalities (OpenAI et al., 2024 Team et al., 2023).

5 Conclusions

In this thesis, we have identified some of the fundamental concepts related to multimodal machine learning and deep learning. We have clarified the differences and identified the common approaches to data fusion. Furthermore, the challenges related to data fusion are presented with a focus on medical diagnosis. Recent studies show the utilization of different data fusion approaches combined with varying machine learning methods. In general, the research questions set for this thesis are answered, as we successfully described the basic problems and methods related to the large and still evolving field of multimodal machine learning in medicine.

Multimodal machine learning has shown potential in medical diagnosing tasks. While many problems remain unsolved, potential benefits have been demonstrated in many retrospective original studies and comprehensive reviews. Research on data fusion and multimodal aspects is expected to continue, and prospective interdisciplinary studies, that allow comparison between similar approaches are needed to validate current findings. Synthetic Data and Explainable Artificial Intelligence could provide important aspects for development and research in this field. In the medical context, research on unimodal models is also important as models that improve on a single modality can then be utilized in MML architectures. Recent advances in deep learning methods such as Transformer architecture might also provide additional tools for data fusion.

The use of artificial intelligence in the thesis

Research rabbit was used to find papers relevant to the topic -Link

Grammarly was used for spelling and grammar corrections -Link

Bibliography

Acosta, J. N., Falcone, G. J., Rajpurkar, P., and Topol, E. J. (2022). "Multimodal biomedical AI". In: *Nature Medicine* 28.9, pp. 1773–1784.

Adadi, A. and Berrada, M. (2018). "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6, pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052.

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., and McDermott, M. B. A. (2019). *Publicly Available Clinical BERT Embeddings*. arXiv: 1904.03323 [cs.CL].

Altman, N. and Krzywinski, M. (May 2018). "The curse(s) of dimensionality". In: *Nature Methods* 15. DOI: 10.1038/s41592-018-0019-x.

Arik, S. O. and Pfister, T. (2020). *TabNet: Attentive Interpretable Tabular Learning*. arXiv: 1908.07442 [cs.LG].

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2019). "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2, pp. 423–443. DOI: 10.1109/TPAMI.2018.2798607.

Barua, A., Ahmed, M. U., and Begum, S. (2023). "A Systematic Literature Review on Multimodal Machine Learning: Applications, Challenges, Gaps and Future Directions". In: *IEEE Access* 11, pp. 14804–14831. DOI: 10.1109/ACCESS.2023.3243854.

Cahan, N., Klang, E., Marom, E., Soffer, S., Barash, Y., Burshtein, E., Konen, E., and Greenspan, H. (May 2023). "Multimodal fusion models for pulmonary embolism mortality prediction". In: *Scientific Reports* 13. DOI: 10.1038/s41598-023-34303-8.

Cahan, N., Marom, E., Soffer, S., Barash, Y., Konen, E., Klang, E., and Greenspan, H. (Apr. 2022). "Weakly Supervised Attention Model for RV Strain Classification from volumetric CTPA Scans". In: *Computer Methods and Programs in Biomedicine* 220, p. 106815. DOI: 10.1016/j.cmpb.2022.106815.

Cohen, J. P., Viviano, J. D., Bertin, P., Morrison, P., Torabian, P., Guarrera, M., Lungren, M. P., Chaudhari, A., Brooks, R., Hashir, M., and Bertrand, H. (2021). *TorchXRayVision: A library of chest X-ray datasets and models*. arXiv: 2111.00595 [eess.IV].

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q., and Ng, A. (2012). "Large Scale Distributed Deep Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.

Burges, L. Bottou, and K. Weinberger. Vol. 25. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/6aca97005c68f1206823815f66102863-Paper.pdf.

Gers, F., Schmidhuber, J., and Cummins, F. (1999). “Learning to forget: continual prediction with LSTM”. In: *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*. Vol. 2, 850–855 vol.2. DOI: 10.1049/cp:19991218.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Hassija, V., Chamola, V., Mahapatra, A., Singal, A., Goel, D., Huang, K., Scardapane, S., Spinelli, I., Mahmud, M., and Hussain, A. (Aug. 2023). “Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence”. In: *Cognitive Computation* 16. DOI: 10.1007/s12559-023-10179-8.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). “A fast learning algorithm for deep belief nets”. In: *Neural computation* 18.7, pp. 1527–1554.

Hochreiter, S. (1998). “The vanishing gradient problem during learning recurrent neural nets and problem solutions”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02, pp. 107–116.

Hochreiter, S. and Schmidhuber, J. (Dec. 1997). “Long Short-term Memory”. In: *Neural computation* 9, pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

Huang, S.-C., Pareek, A., Seyyedi, S., Banerjee, I., and Lungren, M. (Dec. 2020). “Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines”. In: *npj Digital Medicine* 3. DOI: 10.1038/s41746-020-00341-z.

Institute of Medicine and National Academies of Sciences, Engineering, and Medicine (2015). *Improving Diagnosis in Health Care*. Ed. by E. P. Balogh, B. T. Miller, and J. R. Ball. Washington, DC: The National Academies Press. ISBN: 978-0-309-37769-0. DOI: 10.17226/21794. URL: https://nap.nationalacademies.org/catalog/21794/improving-diagnosis-in-health-care.

Johnson, A., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Pollard, T., Hao, S., Moody, B., Gow, B., Lehman, L.-w., Celi, L., and Mark, R. (Jan. 2023). “MIMIC-IV, a freely accessible electronic health record dataset”. In: *Scientific Data* 10, p. 1. DOI: 10.1038/s41597-022-01899-x.

Johnson, A. E. W., Pollard, T. J., Greenbaum, N. R., Lungren, M. P., Deng, C.-y., Peng, Y., Lu, Z., Mark, R. G., Berkowitz, S. J., and Horng, S. (2019). *MIMIC-CXR-JPG,* *a large publicly available database of labeled chest radiographs*. arXiv: 1901.07042 [cs.CV].

Karhunen, J., Raiko, T., and Cho, K. (2015). “Chapter 7 - Unsupervised deep learning: A short review”. In: *Advances in Independent Component Analysis and Learning Machines*. Ed. by E. Bingham, S. Kaski, J. Laaksonen, and J. Lampinen. Academic Press, pp. 125–142. ISBN: 978-0-12-802806-3. DOI: https://doi.org/10.1016/B978-0-12-802806-3.00007-5. URL: https://www.sciencedirect.com/science/article/pii/B9780128028063000075.

Kelly, C. J., Karthikesalingam, A., Suleyman, M., Corrado, G., and King, D. (2019). “Key challenges for delivering clinical impact with artificial intelligence”. In: *BMC medicine* 17, pp. 1–9.

Kim, J.-H., Jun, J., and Zhang, B.-T. (2018). *Bilinear Attention Networks*. arXiv: 1805.07932 [cs.CV].

Kline, A., Wang, H., Li, Y., Dennis, S., Hutch, M., Xu, Z., Wang, F., Cheng, F., and Luo, Y. (Nov. 2022). “Multimodal machine learning in precision health: A scoping review”. In: *npj Digital Medicine* 5. DOI: 10.1038/s41746-022-00712-8.

Lahat, D., Adali, T., and Jutten, C. (2015). “Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects”. In: *Proceedings of the IEEE* 103.9, pp. 1449–1477. DOI: 10.1109/JPROC.2015.2460697.

LeCun, Y., Bengio, Y., and Hinton, G. (May 2015). “Deep Learning”. In: *Nature* 521, pp. 436–44. DOI: 10.1038/nature14539.

Lecun, Y., Bottou, L., Orr, G., and Müller, K.-R. (Aug. 2000). “Efficient BackProp”. In.

Lee, J.-G., Jun, S., Cho, Y.-W., Lee, H., Kim, G. B., Seo, J. B., and Kim, N. (2017). “Deep learning in medical imaging: general overview”. In: *Korean journal of radiology* 18.4, p. 570.

Liang, P. P., Zadeh, A., and Morency, L.-P. (2023). *Foundations and Trends in Multimodal Machine Learning: Principles, Challenges, and Open Questions*. arXiv: 2209.03430 [cs.LG].

Linnainmaa, S. (1970). “ALGORITMIN KUMULATIIVINEN PYÖRISTYSVIRHE. YKSITTÄISTEN PYÖRISTYSVIRHEIDEN TAYLOR-KEHITELMÄNÄ”. MA thesis. Univ. Helsinki.

Miihkinen, M., Mars, N., and Aittokallio, T. (2024). “Matematiikka ja tilastolliset mallit syövän hoitovasteen ennustamisessa”. suomi. In: *Duodecim* 140.3. Vertaisarvioitu., pp. 206–213. ISSN: 0012-7183.

Mueller, S. G., Weiner, M. W., Thal, L. J., Petersen, R. C., Jack, C. R., Jagust, W., Trojanowski, J. Q., Toga, A. W., and Beckett, L. (2005). "Ways toward an early diagnosis in Alzheimer's disease: The Alzheimer's Disease Neuroimaging Initiative (ADNI)". In: *Alzheimer's & Dementia* 1.1, pp. 55–66. DOI: https://doi.org/10.1016/j.jalz.2005.06.003. eprint: https://alz-journals.onlinelibrary.wiley.com/doi/pdf/10.1016/j.jalz.2005.06.003. URL: https://alz-journals.onlinelibrary.wiley.com/doi/abs/10.1016/j.jalz.2005.06.003.

O'Shea, K. and Nash, R. (2015). *An Introduction to Convolutional Neural Networks*. arXiv: 1511.08458 [cs.NE].

Olah, C. (2015). *Understanding LSTM Networks*. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

OpenAI et al. (2024). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL].

Prakash, A., Chitta, K., and Geiger, A. (2021). "Multi-modal fusion transformer for end-to-end autonomous driving". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7077–7087.

Ramachandram, D. and Taylor, G. W. (2017). "Deep Multimodal Learning: A Survey on Recent Advances and Trends". In: *IEEE Signal Processing Magazine* 34.6, pp. 96–108. DOI: 10.1109/MSP.2017.2738401.

Rasmussen, J. and Langerman, H. (2019). "Alzheimer's disease–why we need early diagnosis". In: *Degenerative neurological and neuromuscular disease*, pp. 123–130.

Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., and Baker, T. (2020). "Analysis of Dimensionality Reduction Techniques on Big Data". In: *IEEE Access* 8, pp. 54776–54788. DOI: 10.1109/ACCESS.2020.2980942.

Rosenblatt, F. (1958). "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65 6, pp. 386–408. URL: https://api.semanticscholar.org/CorpusID:12781225.

Schuster, M. and Paliwal, K. K. (1997). "Bidirectional recurrent neural networks". In: *IEEE transactions on Signal Processing* 45.11, pp. 2673–2681.

Soenksen, L., Ma, Y., Zeng, C., Boussioux, L., Carballo, K., Na, L., Wiberg, H., Li, M., Fuentes, I., and Bertsimas, D. (Sept. 2022). "Integrated multimodal artificial intelligence framework for healthcare applications". In: *npj Digital Medicine* 5. DOI: 10.1038/s41746-022-00689-4.

Stahlschmidt, S. R., Ulfenborg, B., and Synnergren, J. (Jan. 2022). "Multimodal deep learning for biomedical data fusion: a review". In: *Briefings in Bioinformatics* 23.2, bbab569. ISSN: 1477-4054. DOI: 10.1093/bib/bbab569. eprint: https://academic.

oup.com/bib/article-pdf/23/2/bbab569/42805085/bbab569.pdf. URL: https://doi.org/10.1093/bib/bbab569.

Tang, Y., Yang, D., Li, W., Roth, H., Landman, B., Xu, D., Nath, V., and Hatamizadeh, A. (2022). *Self-Supervised Pre-Training of Swin Transformers for 3D Medical Image Analysis*. arXiv: 2111.14791 [cs.CV].

Team, G. et al. (2023). *Gemini: A Family of Highly Capable Multimodal Models*. arXiv: 2312.11805 [cs.CL].

Tiulpin, A., Thevenot, J., Rahtu, E., Lehenkari, P., and Saarakkala, S. (2018). "Automatic knee osteoarthritis diagnosis from plain radiographs: a deep learning-based approach". In: *Scientific reports* 8.1, p. 1727.

Van Der Maaten, L., Postma, E. O., Herik, H. J. van den, et al. (2009). "Dimensionality reduction: A comparative review". In: *Journal of Machine Learning Research* 10.66-71, p. 13.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].

Venugopalan, J., Tong, L., Hassanzadeh, H. R., and Wang, M. (Feb. 2021). "Multimodal deep learning models for early detection of Alzheimer's disease stage". In: *Scientific Reports* 11, p. 3254. DOI: 10.1038/s41598-020-74399-w.

Ying, X. (Feb. 2019). "An Overview of Overfitting and its Solutions". In: *Journal of Physics: Conference Series* 1168, p. 022022. DOI: 10.1088/1742-6596/1168/2/022022.

Appendix A Additional figures

Area Under the Curve (AUC) measures the accuracy or performance of the model with different thresholds. To calculate ROC and area under curve (AUC) we need a True positive rate (TPR) and a False negative rate (FPR).

$TPR = \frac{TruePositive}{TruePositive+FalseNegative}$ $FPR = \frac{FalsePositive}{FalsePositive+TrueNegative}$

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

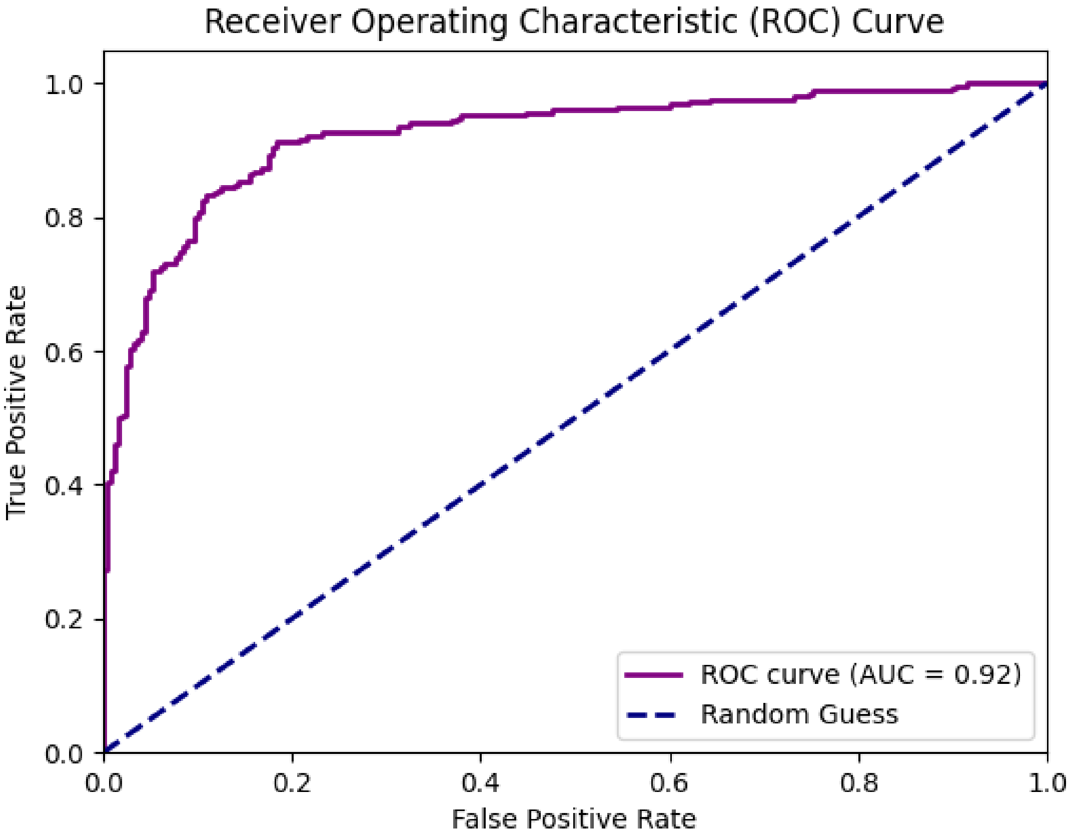


Figure A.1: AUC and ROC example

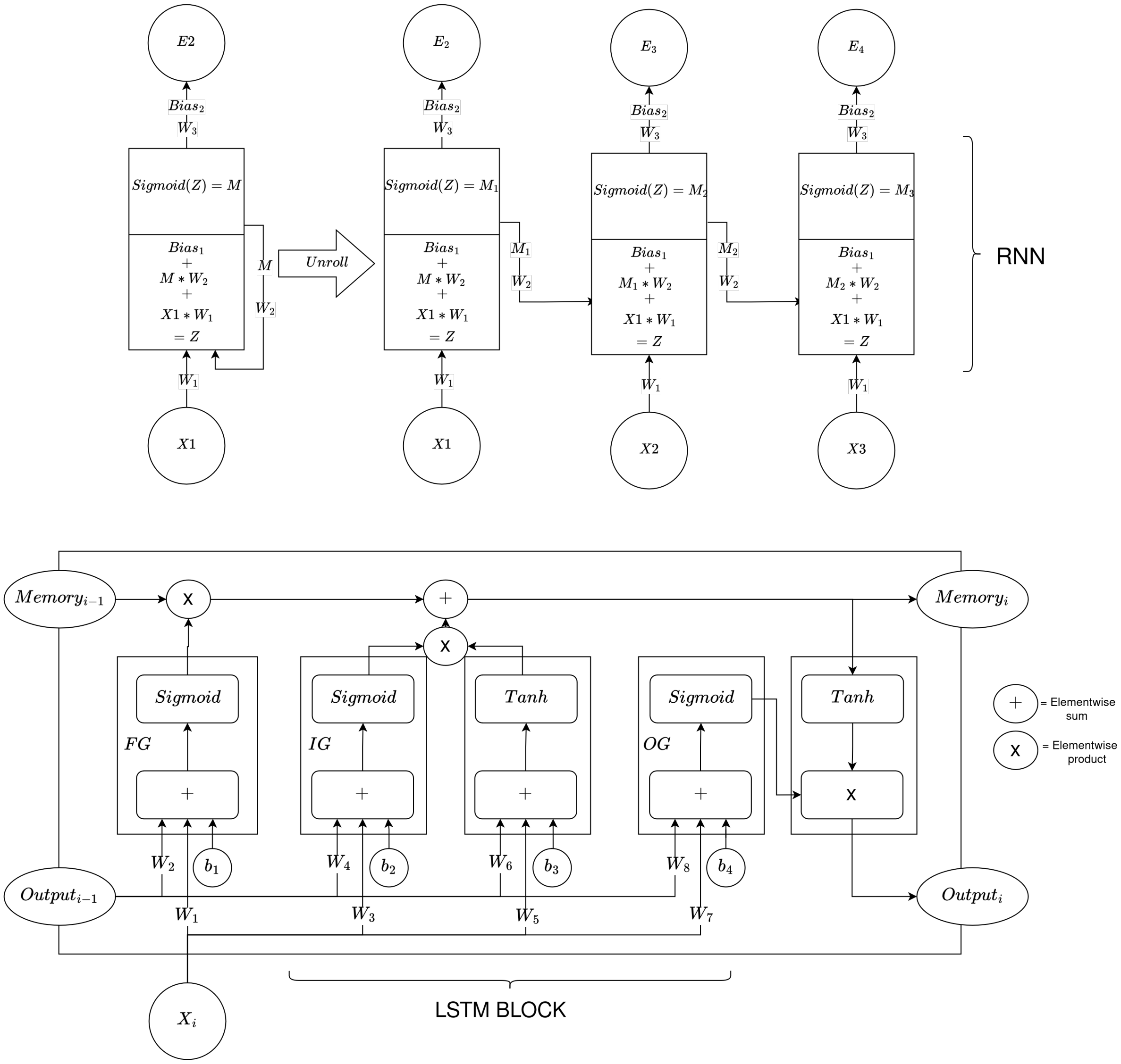


Figure A.2: Sturcture of RNN and LSTM illustrated [inspired: (Olah, 2015)]